

A Service Composition Platform in Cloud Computing using Mobile Devices for Smart Shopping

Submitted by

Lavender Tsongoro

Supervised by

Dr. Alexandru Murgu



This thesis is submitted in complete fulfillment of the academic requirements

for the degree of

Master of Science in Electrical Engineering

in the Faculty of Engineering and The Built Environment

University of Cape Town

2016

The copyright of this thesis vests in the author. No quotation from it or information derived from it is to be published without full acknowledgement of the source. The thesis is to be used for private study or non-commercial research purposes only.

Published by the University of Cape Town (UCT) in terms of the non-exclusive license granted to UCT by the author.

Approval

As the candidate's supervisor, I have approved this dissertation for submission.

Name: Dr. Alexandru Murgu.

Signed by candidate

Declaration

I declare that this thesis is my own work. Where collaboration with other people has taken place, or material generated by other researchers is included, the parties and/or materials are indicated in the acknowledgements or are explicitly stated with references as appropriate.

This work is being submitted for the Master of Science in Electrical Engineering at the University of Cape Town. It has not been submitted to any other university for any other degree or examination.

Signed

Lavender Tsongoro

Name

31.03.2016

Date

Dedication

In loving memory of my late parents Mr Sylvester Tsongoro and Mrs. Florence Tsongoro. I know that every step that I take towards success, you are up there in heaven cheering me on.

Abstract

The development of the Next Generation Networks (NGN) such as LTE, WiMax and 5G networks has resulted in the development of more diverse mobile services. Many voice and video services have been developed (e.g. Viber, Skype and WhatsApp). Social networking sites have also been developed (e.g. Facebook, Instagram and Twitter). Users of these services are increasingly expecting and demanding more complex services which have more capabilities that can improve their day to day business. Users want services that are reliable, fast and easy to use.

To effectively design and implement services, Service Oriented Architecture (SOA) principles are useful and some of the advantages of designing services using SOA principles are:

- Improved interoperability;
- Cross platform and cross application integration;
- Reusability;
- Service composition. [2]

Service composition has the advantage that customized services with more features can be developed by combining two or more basic services. In this research, SOA principles are used to design a cloud based Mobile Smart Shopping Service Platform. Canal Walk Shopping Mall, which is located in Cape Town, South Africa is used as a case study. Various mobile services are composed in order to solve the problem of getting information about the services provided by the shopping mall and also to show the available parking bays, which has become a major concern due to the rapid growth of the surrounding residential and business areas.

Performance measurements for the Smart Shopping service are then conducted to test its power consumption, memory usage, bandwidth usage and application timeline. Conclusions are drawn and recommendations for possible future development are then provided.

Key words: Service Composition, Cloud Computing, Service Oriented Architecture (SOA), Smart Shopping.

Acknowledgements

TO GOD ALMIGHTY. I would like to thank the Lord for helping me throughout this research project by giving me the wisdom to understand the concepts and the drive to go on.

TO THE UNIVERSITY OF CAPE TOWN: Massive thanks to my supervisor Dr. Alexandru Murgu for all his support and supervision throughout the whole research project. I am really grateful for your guidance. Thank you to my research group members for all their support.

TO MY FAMILY: I would like to give my utmost gratitude to my husband Denys Vera for all the love, encouragement and sacrifices that he had to do for me throughout this research project. Your support kept me going and I am forever grateful.

I would like to express my gratitude to my sister Lavinea Tsongoro and my brother Larry Tsongoro for their never ending support and motivation in all that I do. Thank you for always being there for me. I would also like to thank my lovely nieces, Atida Rylie Mugauri and Tanatswa Nicole Mugauri for all their warm hugs and cute little smiles that always made my day during tough times during the research project.

Table of Contents

| | |
|--|-----|
| Approval | i |
| Declaration | ii |
| Dedication | iii |
| Abstract | iv |
| Acknowledgements | v |
| List of Figures | ix |
| Glossary | xii |
| 1 Introduction | 1 |
| 1.1 Background | 1 |
| 1.2 Research Problems | 3 |
| 1.3 Research Objectives | 3 |
| 1.4 Research Questions | 4 |
| 1.5 Research Relevance | 4 |
| 1.6 Scope and Limitations | 4 |
| 1.7 Research Outline | 5 |
| 2 Literature review | 6 |
| 2.1 Introduction to services | 6 |
| 2.1.1 Types of Services | 7 |
| 2.1.2 E-services/ Information Services | 8 |
| 2.2 Cloud Computing | 8 |
| 2.2.1 Cloud Computing Architecture | 10 |
| 2.2.2 Cloud Services Delivery Models | 11 |
| 2.2.3 Service Delivery in Cloud Computing | 12 |
| 2.2.4 Cloud Computing Deployment Methods | 13 |
| 2.2.5 Advantages of Cloud Computing | 14 |
| 2.2.6 Challenges of Cloud Computing | 14 |
| 2.2.7 Cloud Computing Platforms and Technologies | 16 |
| 2.3 Mobile Cloud Computing | 17 |
| 2.3.1 Challenges in Mobile Cloud Computing | 18 |

| | | |
|-------|--|----|
| 2.4 | Service Oriented Architecture (SOA)..... | 19 |
| 2.4.1 | SOA Principles | 20 |
| 2.4.2 | How Services Communicate in SOA | 21 |
| 2.4.3 | Benefits of SOA..... | 22 |
| 2.4.4 | SOA Delivery Lifecycle | 24 |
| 2.5 | Service Composition..... | 25 |
| 2.5.1 | Design Characteristics of Composition Member Capabilities | 29 |
| 2.5.2 | Design Characteristics for Composition Controller Capabilities | 29 |
| 2.5.3 | Types of Research in Service Composition..... | 30 |
| 2.5.4 | Methods of dynamic service composition | 30 |
| 2.5.5 | Risks Associated with Service Composition | 33 |
| 2.6 | Smart Systems | 34 |
| 2.7 | Previous Studies Conducted on Related Topics | 35 |
| 2.7.1 | Issues and Current Solutions of MCC | 35 |
| 2.7.2 | Mobile Applications using MCC..... | 36 |
| 2.7.3 | Feasibility of Mobile Computation Offloading and Data Backups..... | 38 |
| 2.7.4 | Security and Privacy in MCC..... | 39 |
| 2.7.5 | Cloud based Middleware for Dynamic Service Composition..... | 41 |
| 2.7.6 | Quality of Service (QoS)-aware Composition Methods..... | 42 |
| 2.7.7 | Virtual Services in Cloud Computing | 42 |
| 2.7.8 | Service Composition and Testing for Cloud Computing | 43 |
| 2.7.9 | Applications of Smart Systems..... | 44 |
| 3 | Mobile Smart Shopping Service Design | 47 |
| 3.1 | Development Tools..... | 47 |
| 3.1.1 | Visual Studio 2015 | 47 |
| 3.1.2 | Microsoft Azure..... | 48 |
| 3.1.3 | Navicat..... | 51 |
| 3.1.4 | Postman | 52 |
| 3.2 | System Design Flow | 54 |
| 3.2.1 | Service Concept..... | 55 |
| 3.2.2 | Technological Architecture | 61 |
| 4 | Design Implementation | 68 |
| 4.1 | Presentation Layer Implementation..... | 69 |

| | |
|--|-----|
| 4.1.1 User Interface (UI)..... | 69 |
| 4.1.2 User Interface Logic Design..... | 74 |
| 4.1.3 The Smart Shopping Service UI layer..... | 76 |
| 4.2 Data Layer Implementation..... | 86 |
| 4.2.1 Deploying the Azure SQL Database..... | 87 |
| 4.2.2 Database Table Design..... | 87 |
| 4.2.3 Designing and Implementing API calls..... | 90 |
| 4.2.4 Testing API calls..... | 92 |
| 4.2.5 Data Access..... | 93 |
| 4.3 Business Layer Implementation..... | 94 |
| 5 System Performance Evaluation..... | 96 |
| 5.1 Mobile Service Performance Testing..... | 96 |
| 5.2 Service Testing Methods..... | 96 |
| 5.2.1 Mobile Phone Simulators..... | 96 |
| 5.2.2 Mobile Phone Emulators..... | 97 |
| 5.2.3 Physical Mobile Phone Device..... | 98 |
| 5.3 Performance tests using Emulator/Simulators..... | 99 |
| 5.4 Performance tests using real devices..... | 100 |
| 5.5 Smart Shopping Service System Performance Tests..... | 100 |
| 5.5.1 Application Timeline..... | 101 |
| 5.5.2 Memory usage..... | 102 |
| 5.5.3 Energy Consumption..... | 104 |
| 5.5.4 Bandwidth usage..... | 106 |
| 5.6 Performance Testing Conclusions..... | 110 |
| 6 Conclusions and Recommendations..... | 112 |
| 6.1 Overall Conclusions..... | 112 |
| 6.2 Recommendations for future work..... | 112 |
| Appendix A..... | 118 |
| Appendix B..... | 122 |

List of Figures

| | |
|--|----|
| Figure 2.1 Relationship between services, mobile services and mobile e-services [10]. | 8 |
| Figure 2.2 Users accessing cloud resources via a pay as you use basis. | 9 |
| Figure 2.3 Cloud computing architecture [1]. | 10 |
| Figure 2.4 Cloud Computing service delivery model [6]. | 12 |
| Figure 2.5 Characteristics of cloud computing [12]. | 16 |
| Figure 2.6 Internet access technologies for cloud computing [1]. | 17 |
| Figure 2.7 Communication of services in a SOA. | 22 |
| Figure 2.8 Phases of the SOA delivery lifecycle. | 24 |
| Figure 2.9 Service delivery process [10]. | 26 |
| Figure 2.10 Provision of composed services by the collaboration of organizations [10]. | 26 |
| Figure 2.11 Service Composition using SOA principles. | 27 |
| Figure 2.12 Service roles hierarchy. | 28 |
| Figure 2.13 Set up of the composite service interface [15]. | 31 |
| Figure 2.14 Stand - alone composite service method [15]. | 31 |
| Figure 2.15 First option for making a stand-alone composite service [15]. | 32 |
| Figure 2.16 More complex standalone composite service method [15]. | 32 |
| Figure 2.17 Second method for standalone composite method [15]. | 33 |
| Figure 2.18 Proposed architecture for dynamic service composition [23]. | 41 |
| Figure 2.19 Virtual services [25]. | 43 |
| Figure 3.1 Visual studio community 2015 start page. | 47 |
| Figure 3.2 Deployed Azure SQL database. | 50 |
| Figure 3.3 Deployed Azure mobile service. | 51 |
| Figure 3.4 Navicat graphic user interface. | 52 |
| Figure 3.5 Postman Graphic User Interface (GUI). | 53 |
| Figure 3.6 Postman URL header and key-value pair [37]. | 53 |
| Figure 3.7 Postman request builder and side bar [37]. | 54 |
| Figure 3.8 The three aspects of mobile service system design [11]. | 55 |
| Figure 3.9 Service Concept design process. | 56 |
| Figure 3.10 Factors affecting service acceptance [11]. | 60 |
| Figure 3.11 Mobile application architecture [38]. | 62 |
| Figure 3.12 The presentation layer. | 63 |
| Figure 3.13 Presentation layer design process. | 64 |

| | |
|--|----|
| Figure 3.14 Data layer design process..... | 65 |
| Figure 3.15 Shopping mall layout table definition..... | 66 |
| Figure 3.16 Parking scenarios table definition..... | 66 |
| Figure 3.17 Store categories table definition..... | 67 |
| Figure 3.18 Stores table definition..... | 67 |
| Figure 4.1 Design flow and implementation of the Smart Shopping Service System..... | 68 |
| Figure 4.2 Selection of the Hub template in Visual Studio Community 2015..... | 70 |
| Figure 4.3 The Hub app template [39]..... | 70 |
| Figure 4.4 User Interface design process..... | 72 |
| Figure 4.5 Hub page XAML..... | 73 |
| Figure 4.6 Loosely coupled XAML pages..... | 74 |
| Figure 4.7 UI Logic design process..... | 75 |
| Figure 4.8 Hub page code behind navigation methods..... | 76 |
| Figure 4.9 The Store Categories page and the Stores page..... | 77 |
| Figure 4.10 Store Details..... | 78 |
| Figure 4.11 Hub section 2..... | 78 |
| Figure 4.12 Lower level map..... | 78 |
| Figure 4.13 Upper level map..... | 78 |
| Figure 4.14 Trading Hours page (Hub section 3) and Parking Service page (Hub section 4)..... | 79 |
| Figure 4.15 Scenario 1..... | 80 |
| Figure 4.16 Scenario 2..... | 80 |
| Figure 4.17 Scenario 3..... | 81 |
| Figure 4.18 Scenario 4..... | 81 |
| Figure 4.19 Scenario 5..... | 81 |
| Figure 4.20 Scenario 6..... | 81 |
| Figure 4.21 Scenario 7..... | 82 |
| Figure 4.22 Scenario 8..... | 82 |
| Figure 4.23 Scenario 9..... | 82 |
| Figure 4.24 Scenario 10..... | 82 |
| Figure 4.25 Hub section 5..... | 83 |
| Figure 4.26 Installed external location application..... | 83 |
| Figure 4.27 Google Maps start page..... | 84 |
| Figure 4.28 Google Maps user start point..... | 84 |
| Figure 4.29 Google Maps destination point..... | 85 |

| | |
|---|-----|
| Figure 4.30 Here Drive+ start page..... | 85 |
| Figure 4.31 Here Drive+ user location | 85 |
| Figure 4.32 Here Drive+ destination point..... | 85 |
| Figure 4.33 Hub section 6..... | 86 |
| Figure 4.34 Deployed Azure SQL database. | 87 |
| Figure 4.35 Parking scenarios table..... | 88 |
| Figure 4.36 Store categories table..... | 88 |
| Figure 4.37 Stores table. | 89 |
| Figure 4.38 selecting the store names and their respective telephone numbers. | 89 |
| Figure 4.39 Selecting all the information from the Stores table..... | 90 |
| Figure 4.40 API calls. | 91 |
| Figure 4.41 SQL code for getstoresincategory API call..... | 91 |
| Figure 4.42 SQL code for getparkingscenarios API call..... | 92 |
| Figure 4.43 get shopping mall layout API call..... | 92 |
| Figure 4.44 Store Categories API call. | 93 |
| Figure 4.45 Data Model (red arrows) | 94 |
| Figure 4.46 Smart ShoppingAPI.cs class..... | 95 |
| Figure 5.1 Emulator 8.1 WVGA 4 inch 512MB start screen Hub Section 1 screen. | 97 |
| Figure 5.2 IpOverUsbSvc status change..... | 98 |
| Figure 5.3 Application timeline for mobile phone..... | 101 |
| Figure 5.4 Application timeline for mobile phone emulator..... | 102 |
| Figure 5.5 Memory usage for mobile phone..... | 103 |
| Figure 5.6 Memory usage for mobile phone emulator..... | 103 |
| Figure 5.7 Snapshot 1-3 for mobile phone..... | 104 |
| Figure 5.8 Snapshot 1-4 for mobile phone emulator..... | 104 |
| Figure 5.9 Power usage for the mobile phone..... | 105 |
| Figure 5.10 Power usage for mobile phone emulator..... | 105 |
| Figure 5.11 Connection of the mobile phone emulator to the virtual Ethernet..... | 107 |
| Figure 5.12 Bandwidth usage when loading Store Categories page..... | 108 |
| Figure 5.13 Bandwidth after loading Store Categories page..... | 108 |
| Figure 5.14 Bandwidth for loading the Banks, Forex and Financial Store Category list..... | 109 |
| Figure 5.15 Bandwidth after loading the Banks, Forex and Financial Store Category list... | 109 |

Glossary

- 1. Next Generation Networks (NGN):** Packet based networks that are used to provide services to users (e.g. telecommunication services).
- 2. Long Term Evolution (LTE):** 4G wireless communication standard which was developed by the 3rd Generation Partnership Project (3GPP).
- 3. World Wide Interoperability for Microwave Access (WiMAX):** 4G wireless communication standard which was put forward by the WiMax Forum.
- 4. Service Oriented Architecture (SOA):** This principle states that a collection of autonomous and standardised services can be used to solve a complex service delivery problem by having each of these basic services to solve parts of the problem.
- 5. Cloud Computing:** Enables network access to a pool of configurable resources for example applications, services, servers, storage and networks that are always available on demand and convenient.
- 6. Mobile cloud computing (MCC):** Mobile devices are used to access services that are hosted on the cloud. It is a combination of three technologies which are mobile computing, cloud computing and mobile internet.
- 7. Infrastructure-as-a-Service (IAAS):** Cloud infrastructure is delivered to the cloud user in the form of hardware, storage and/or networking.
- 8. Software-as-a-Service (SAAS):** Applications are provided to the cloud users on demand through a browser (e.g. metadata management, social networking, blogs and Wiki services).
- 9. Platform-as-a-service (PAAS):** Cloud users are given access to scalable runtime environments for deploying and executing applications by the use of programming languages and tools which are provided to users by the service provider.
- 10. Service Composition:** Customised services are developed by combining and linking basic services which may either be atomic or composite into a more complex service that has features which are not present in the individual services.
- 11. Smart Systems:** Make use of information in service infrastructures and aim to effectively use and analyse information, and for the surrounding physical and digital objects (e.g. machines, devices, sensors, businesses and people) to interact together and be involved in proactive service delivery.

1 Introduction

This chapter gives the introduction of this research. A background of the study is presented in section 1.1. Section 1.2 then presents the research problems which are then followed by the research objectives in section 1.3. The research questions are presented in section 1.4 and the relevance of this research is then presented in section 1.5. The scope and limitations of this research are presented in section 1.6 then finally, the research outline is presented in section 1.7.

1.1 Background

The use of Cloud Computing technology has resulted in the innovation of many new systems and applications which are used by people in their daily lives. Cloud Computing services such as cloud network resources, storage, runtime environments and applications are made accessible to cloud service users via either web browsers or web service Application Programming Interfaces (APIs). These services are scalable and are provided to users on demand and on a pay as you use basis with zero capital expenditure needed. [6]

The need to access services, applications and data at anytime and anywhere has been made possible by the mobile cloud computing platforms and these use mobile devices to access services which are hosted in the cloud. The use of mobile devices for data intensive applications has some limitations such as limited storage, low battery life and limited network bandwidth. Mobile cloud computing helps to alleviate these problems by allowing the data storage, processing and computing of the cloud service to be done in the cloud and not on the mobile device. [1]

The design and implementation of services can be effectively done by using the Service Oriented Architecture (SOA) principles [7], [2]. A collection of autonomous and standardised services are used to solve a complex service delivery problem by having each of these services solving parts of the problem [7].

These services must have the following features:

- Standardised;
- Loosely coupled;
- Reusable;
- Stateless;
- Discoverable;
- Composable;
- Autonomous;
- Abstracted - service contracts to contain only essential capabilities [7].

Service composition is the ultimate goal of SOA. It is a process of developing customised services by combining and linking basic services which may either be atomic or composite into a service that is more complex and has features that are not present in the individual services. [3], [4], [5]

In this research, Mobile Cloud Computing technologies are used to develop a Smart Shopping Mobile Service for Canal Walk Shopping Mall which is located in Century City in the town of Cape Town, South Africa. This service is accessed by the user via a mobile phone. The Smart Shopping service is developed by composing primary services in order to improve the shopping experience of users. Services that are composed in this system are as follows:

- Mall information service;
- Store category information service;
- Store information service;
- Parking service;
- Location service;
- Navigation direction from current location of user to Canal Walk Shopping Mall.

Canal Walk Shopping Mall is used as a case study because it is one of Africa's leading major regional retail shopping malls. It is also a tourist destination area that attracts thousands of both local and international visitors every year because of its spectacular architecture and spacious shopping mall that has over 400 shops that sell both local and international products. [8]

Canal Walk Shopping Mall is located at the epicentre of Century City, which is located 12 km from Cape Town City Centre and is a 250 hectares mixed use precinct with developments that have a value of more than ZAR 21 billion. Century City is also the home to a 900 seater conference centre, a football arena and churches, of which one of them is a mega church called Hillsong Church. It is also home to a day hospital, Ratanga Junction theme park, more than 500 businesses, hospitality apartments and 3000+ residential apartments with an estimated 50 000 people living and working in the area. [9]

Canal Walk Shopping Mall is also located about 5 minutes' drive away from one of Cape Town's high density residential areas called Jo Slovo where thousands of people reside and has a growing population.

1.2 Research Problems

- Large numbers of people visit Canal Walk Shopping Mall every day and motorists face problems when trying to locate the available parking bays in the shopping mall parking lot. The car parking system at Canal Walk only has a parking ticket issuing machine at the entrance of the shopping mall parking lot. This records the time spent by the shopper's car in the parking lot and this information is then used to calculate the cost of parking to be paid. Canal Walk parking system does not have any other parking information system to assist the drivers. Motorists have to drive around the parking lot in order to locate a free parking slot hence, there is great need for an intelligent system to assist motorists in the shopping mall parking lot by providing them with relevant information about the availability of parking bays.
- There is a need for a composite service delivery system that has information about the services provided by the shops and businesses located at Canal Walk Shopping Mall.
- There is also a need for a service that helps both local and international visitors to navigate to the shopping mall from their current location.

1.3 Research Objectives

- To propose and develop a composed smart shopping service delivery system that is hosted on a cloud computing platform and can be accessed by using a mobile phone. This system

should show relevant information about the shopping services provided by the shops and businesses located at Canal Walk Shopping Mall.

- To propose and develop a prototype of an intelligent mobile car parking service system which assists drivers with locating the available parking bays in the Canal Walk Shopping Mall parking lot. This mobile service must be hosted on a cloud computing platform.

1.4 Research Questions

- Can the Service Composition methods and mobile phone technologies be used to develop the information and location based service?
- How the Mobile Cloud Computing technologies can be used to develop a cloud based Smart Shopping Mobile Service System?
- What are the tools required to develop the cloud based information service system assisting with the car parking task?

1.5 Research Relevance

The motivations for this research project are as follows:

- Enhancing the capabilities of the car parking service, location-based service and the information dissemination service that is currently available at Canal Walk Shopping Mall.
- Assisting the Cape Town locals and tourists by providing them with relevant shopping information about Canal Walk Shopping Mall.

1.6 Scope and Limitations

- The developed composed service system in this study may only be installed on Windows Mobile Phones.
- This research presents a prototype of a parking lot plan with 38 parking bays which will be used as proof of concept.
- Due to the absence of the sensors that sense the presence or absence of a car in a parking bay at Canal Walk Shopping Mall, 10 scenarios of the state of the parking lot with regards

to parking bay availability are designed and the change of these states is based on a 100 seconds sampling strategy.

- The developed composed service system is an on-demand service where the user first has to install the application on their Windows Mobile Phone in order to use the service.

1.7 Research Outline

This research is organised as follows:

Chapter 2 provides the literature review on services, cloud computing technology, mobile cloud computing technology, SOA technology, principles of service composition technology and a study about smart systems is presented. Finally, a discussion on the previous studies that have been done by other researchers on the topics related to this study are presented.

Chapter 3 discusses the tools that were used to develop the Smart Shopping mobile service and how they were used to design the system.

Chapter 4 presents the implementation of the Smart Shopping mobile service.

Chapter 5 presents the system performance results obtained after testing the developed Smart Shopping service system and the analysis of the results.

Chapter 6 provides the conclusions and recommendations for future work.

2 Literature review

This chapter presents the literature on the aspects related to this research which are Services, Cloud Computing, Mobile Cloud Computing (MCC), Service Oriented Architecture (SOA), Service Composition and Smart Systems. Section 2.1 presents the aspects of services, their description, types and a further discussion on e-services and mobile e-services is then provided. Section 2.2 then presents the concept of Cloud Computing where the data for the Smart Shopping service is stored. The Cloud Computing architecture, deployment methods and cloud service delivery methods are also presented. The advantages and challenges of Cloud Computing are then presented. In section 2.3 the aspect of Mobile Cloud Computing is then presented. Section 2.4 presents the principles of SOA, its benefits and also its service delivery lifecycle. Section 2.5 then presents the aspect of Service Composition and the types of research that have been done on this area. Two Service Composition methods are presented and then the risks associated with Service Composition. In section 2.6, the aspect of Smart Systems and a discussion on its applications is provided. Finally, section 2.7 presents the previous studies that have been conducted on topics related to this research.

2.1 Introduction to services

Services are activities that are more or less intangible in nature and normally provided to customers by a service provider as a response to customer needs. These can either be physical resources, goods or systems. Services can be provided from business to business, business to customer, business to government, customer to customer, business to employee or within an organisation. [10]

Services should be user-context aware enabling the provision of personalised services to users by considering their demographic. This also enables the modelling of time varying user preferences (e.g. the type of channels that a user prefers to watch during the day might be different from those they might want to watch during the night) and the service must suggest the correct type of channel at any given time. [10]

2.1.1 Types of Services

Below is the explanation of location based services and mobile information services. In this thesis these services are combined to make a composed service system.

- **Location Based Services**

Information is automatically generated about the location of a mobile device using various methods such as Wi-Fi, GPS and cell tower triangulation method. Customised services can be delivered to a user, enabling the service delivery to be safe, convenient and productive. [11]

- **Mobile Information Services**

These provide information to the user via the use of a mobile device. Usefulness of a service is a measure of the benefits and costs of using the service and its value depends on the context of the user.

Table 2.1 presents the differences between services and physical goods.

Table 2-1 The difference between services and physical goods [10].

| Services | Physical goods |
|---|--|
| <ul style="list-style-type: none"> • Intangible • Heterogeneous • Production, distribution and consumption simultaneous process • Any activity or process • Core value produced in buyer-seller interactions • Customers participate in production • Cannot be kept in stock • No transfer of ownership | <ul style="list-style-type: none"> • Tangible • Homogeneous • Production and distribution separated from consumption • A thing • Core value produced in factory • Customers do not (normally) participate in production • Can be kept in stock • Transfer of ownership |

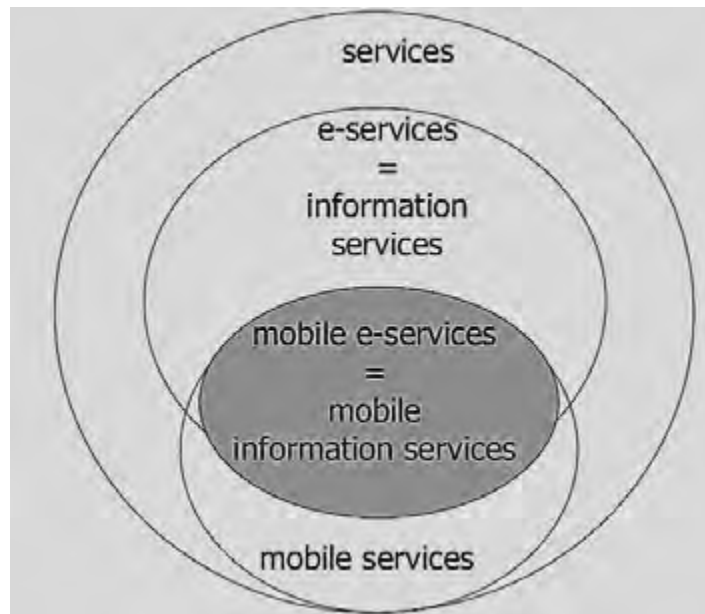


Figure 2.1 Relationship between services, mobile services and mobile e-services [10].

2.1.2 E-services/ Information Services

E-services are activities of intangible nature that are provided to customers by service providers via the internet [11]. Mobile e-services are e-services that are provided whilst the consumer is mobile via a telecommunications network that supports the internet channel between the customer and the service provider [11]. Figure 2.1 shows the relationship between services, e-services, mobile services and mobile e-services. E-services and mobile services are all types of services and mobile e-services are e-services provided by a mobile device [11]. In this research the developed Smart Shopping service is hosted on a cloud computing platform. The next section gives more details about cloud computing.

2.2 Cloud Computing

Cloud computing enables network access to highly available configurable resources, (e.g. applications, services, servers, storage and networks), on demand and conveniently. These services are provisioned rapidly, with minimal service provider interactions via the Internet, making it a fundamental part of the system. [6]

Virtualization technologies are used in cloud computing to effectively provide resources that are supported by data centres dynamically to the users and is used to partition the hardware resources such as CPU and memory. Cloud computing infrastructure, runtime environments, virtual hardware, storage and services can be accessed on a pay as you use basis with minimal upfront costs. This is illustrated in Figure 2.2. The resources can be scaled up or down according to the user's needs per given time. Using cloud computing services, a person's data and documents can be accessed anytime and anywhere using a device connected to the Internet. Cloud computing supports the innovation of new systems and applications which can be integrated into an existing system as it is both dynamic and flexible. [6]

The vision of cloud computing is to have an open global cloud computing market place where cloud service providers publish their services and cloud users are able to discover them by entering their service requests and cloud resources matching the request are suggested to the user for them to choose. This has the advantage that the revenue of cloud service providers may be increased as their services become more accessible to the users. Currently it is very difficult to move hosted cloud services from one vendor to another due to both legal and technological barriers that limit the cloud users to one vendor. [6]

Figure 2.2 shows how the cloud services users get access to cloud computing resources.

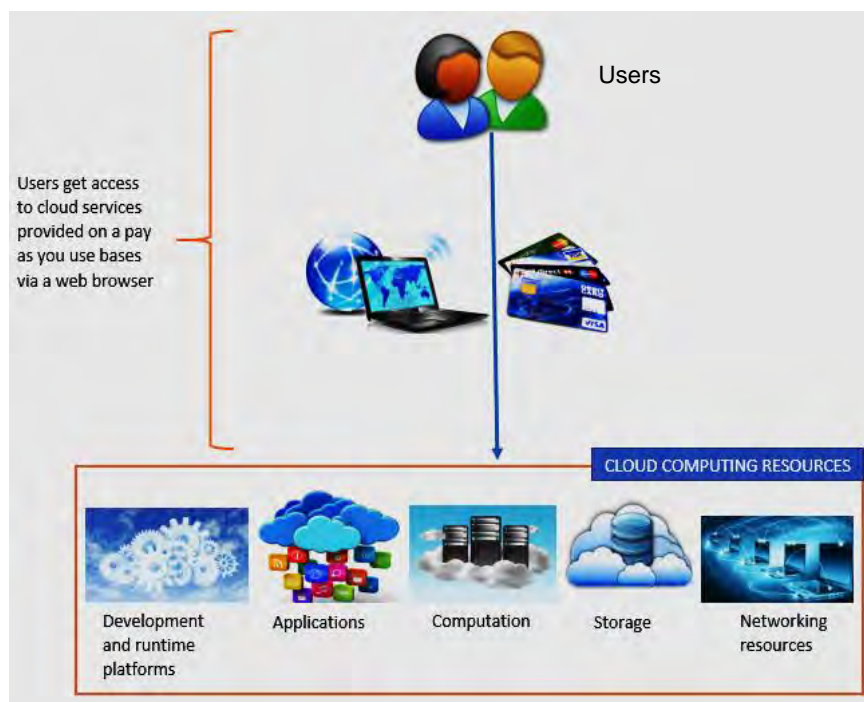


Figure 2.2 Users accessing cloud resources via a pay as you use basis.

2.2.1 Cloud Computing Architecture

The cloud computing architecture has four layers which are the application layer, platform layer, infrastructure layer and the hardware layer. Each of these layers is described below.

Figure 2.3 shows the cloud computing architecture.

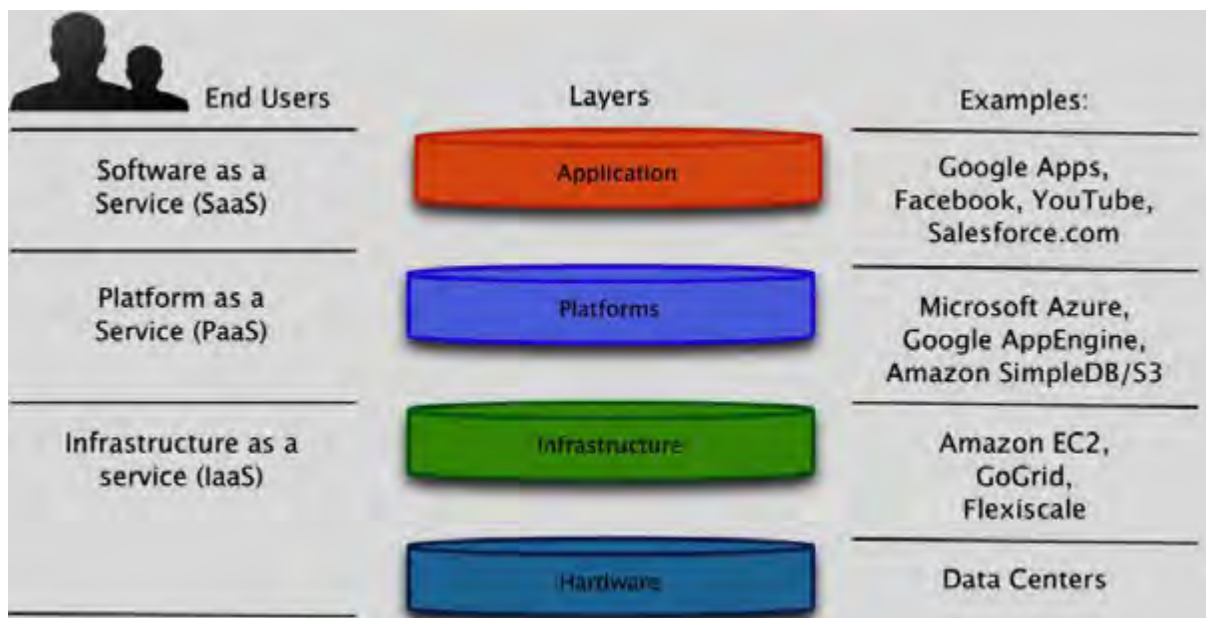


Figure 2.3 Cloud computing architecture [1].

Hardware Layer

These are the physical cloud resources for example physical servers, routers, cooling systems and switches, and these are deployed at the data centers. The hardware layer is responsible for the management of physical cloud resources. [1]

Infrastructure Layer

This is the virtualization layer where the cloud physical resources are partitioned and virtualized using virtualization technologies such as Xen, KVM and VMware for them to be made accessible to the users. [1]

Platform Layer

Made up of operating systems and application frameworks that are installed in the virtual environment on the cloud. Users can get access to them from any place where there is internet access. [1]

Application Layer

Made up of applications that are hosted on the cloud and can be auto scaled resulting in them having better availability, performance and lower application costs [1].

2.2.2 Cloud Services Delivery Models

Below, a description of the cloud services delivery models which are Infrastructure as a service, Platform as a Service and Software as a service is provided [6], [12].

Infrastructure as a Service (IAAS)

The user gets the cloud infrastructure in the form of hardware, storage and/or networking. Virtual networking offers services that manage networking among virtual instances and also their connectivity to the Internet or private networks. When a user does not want to invest in new computing infrastructure or when an organisation wants to rapidly expand their services, IAAS would be the best option to use. [6], [12]

Platform as a Service (PAAS)

Cloud users are given access to scalable runtime environments for deploying and executing applications by the use of programming languages and tools which are provided to users by the service provider. The user has control over the deployed applications and the application hosting environment configurations. The cloud platforms are mainly used in software development and it enables multiple developers to collaborate on a developmental project. The development and testing of the developed services should be automated. [6], [12]

Software as a Service (SAAS)

Applications are provided to the cloud users on demand through a browser. The example of cloud applications that can be accessed this way are metadata management, social networking, blogs and Wiki services. For this method the user does not have control of the underlying cloud infrastructure. [6], [12]

Figure 2.4 shows the cloud computing service delivery model.

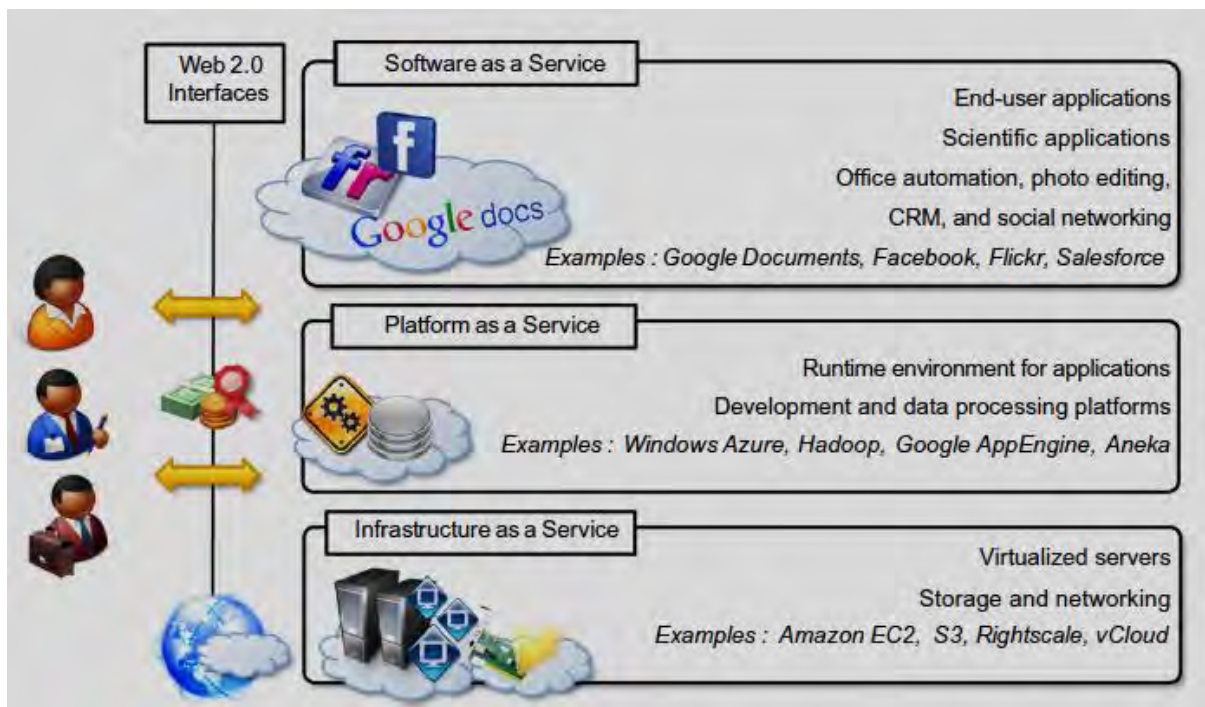


Figure 2.4 Cloud Computing service delivery model [6].

2.2.3 Service Delivery in Cloud Computing

- Cloud services are accessed via a non-proprietary web browser or a web services application programming interface (API);
- Zero capital expenditure is needed to get started;
- Services provided on a pay as you use basis;
- Services are scalable and device independent;
- Services provided to users on demand [6].

2.2.4 Cloud Computing Deployment Methods

There are four methods of deploying cloud computing and these are discussed below.

Public Clouds

Cloud services are sold to organizations or the general public on a subscription bases. Examples of these are OneDrive, ICloud and Google drive [6], [12].

Private/Enterprise Clouds

The cloud services are provided exclusively to a single client [1]. This cloud deployment method is mainly used by large organizations such as banks and the government for them to have a more private and secure cloud system because of the sensitive information they handle. The physical layer of the cloud infrastructure is set up by the organisation as a datacenter, cluster, an enterprise desktop grid or as a combination of the above. A virtualization technology such as Xen, VMware or KVM is then installed on the physical layer and this makes the foundation of the cloud system. This infrastructure is then controlled by virtualization management technologies to provide IAAS solutions. Private clouds are managed by either the organization itself or by third parties that can either be on or off premise. This method allows for greater control of the performance, security and reliability but has a major disadvantage that capital costs are required to set up this deployment method. [6], [12]

Community Cloud

The cloud infrastructure for this deployment method is shared by several organizations that have the same concerns for example security levels or compliance considerations. This method can be managed by the organizations themselves or a third party on or off site.

[6], [12]

Hybrid Cloud

It is the combination of two or more of the cloud deployment methods discussed above. These combined deployment methods continue to be unique entities and are bound together by

standardized or proprietary technologies that enables data and application portability. An example of this instance is an organization, which uses a private cloud but needs more cloud infrastructure than its current capabilities. It can start storing some of its less sensitive data on a public cloud. [6], [12]

2.2.5 Advantages of Cloud Computing

The advantages of using cloud computing are as follows:

- Reduced maintenance and operational costs;
- On demand access;
- Energy efficiency;
- System developers do not have to worry about the complexity of infrastructure management and scalability;
- Efficient resource allocation;
- Seamless creation and use of third party services;
- Due to the absence of upfront costs, smaller companies or individuals can now quickly translate their business ideas into results with little investment cost on the infrastructure or data storage;
- People can access their documents anytime, anywhere whilst using any device which is connected to the internet;
- Simplified application acceleration and scalability [12], [6].

2.2.6 Challenges of Cloud Computing

Service Unavailability

If a cloud service provider has difficulty or is not able to deliver the cloud services, a lot of cloud users will not be able to access or use their cloud services and this can have a large economic impact on companies that have businesses relying on the cloud services. Service level agreements (SLA) are used to state the availability of services. [6], [12]

Vendor Lock-In

Once a client starts using cloud services provided by a certain cloud vendor, it is difficult to move their data to another vendor. The National Institute of Standards and Technology (NIST) is attempting to address this problem. [6], [12]

Performance Unpredictability

Since there is competition for resources in shared cloud computing environments, applications with runtime behaviours that are affected by them running concurrently with other applications would be difficult to optimise. [6], [12]

Data Confidentiality and Auditability

Data encryption and authorisation can be used to prevent unauthorised people from gaining access to personal data of an individual or an organisation. Unavailability of these measures may result in private data being hacked and/or corrupted. [6], [12]

Data Transfer Bottlenecks

When using data intensive services and transferring large amounts of data, the network needs to be very fast so that the data transfer time is reduced. [6], [12]

Elasticity Requirements

Cloud computing services are elastic enabling their resources to be scaled up or down. Complex algorithms for controlling resource allocations and workload placements are needed. [6], [12]

Infrastructure Failure

This can occur due to power failures or natural causes such as lightning or electrical storms that can hit a data centre. This can be overcome by geographic dispersion of replicated data centres. [6], [12]

2.2.7 Cloud Computing Platforms and Technologies

The major cloud computing platforms that are on the market are as follows:

- Microsoft Azure;
- Google AppEngine;
- Amazon web services;
- Hadoop;
- Manjrasoft Aneka;
- Force.com and Salesforce.com [6].

Figure 2.5 shows the summary of the major characteristics of cloud computing.

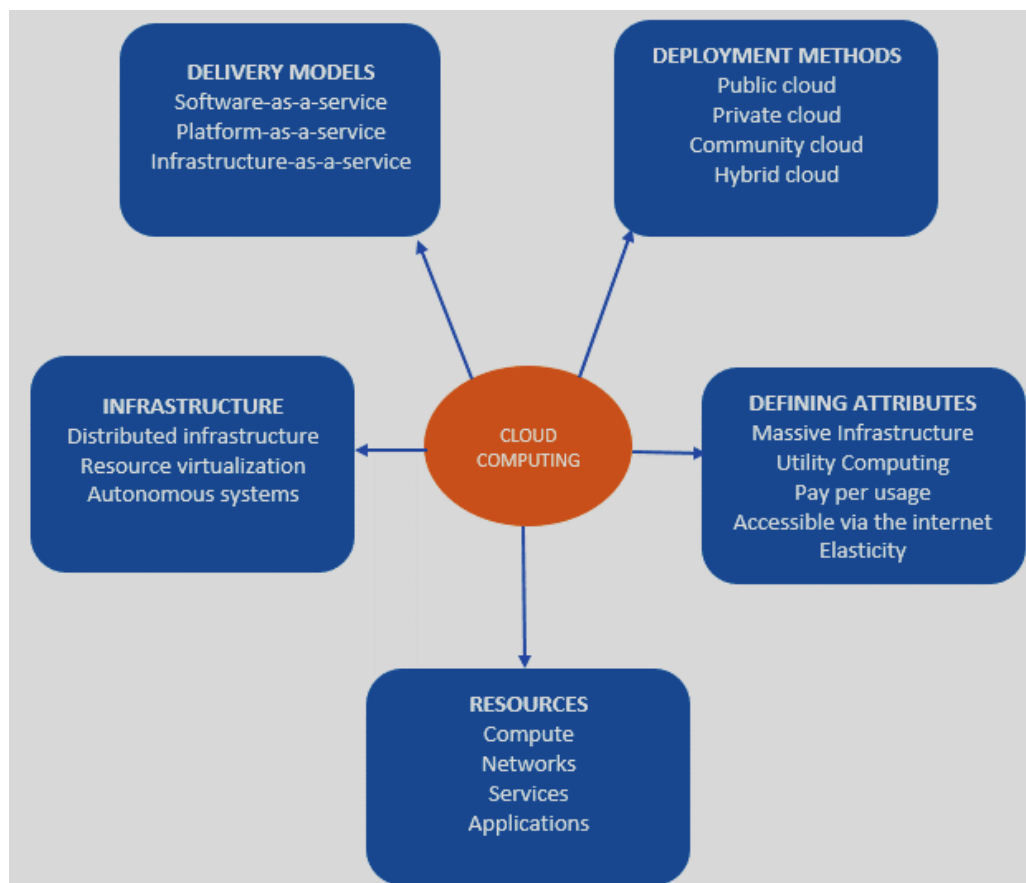


Figure 2.5 Characteristics of cloud computing [12].

The next section discusses the concept of Mobile Cloud Computing.

2.3 Mobile Cloud Computing

In Mobile Cloud Computing, mobile devices are used to gain access to services that are hosted on the cloud. It is a combination of three technologies which are mobile computing, cloud computing and mobile Internet. All the data storage, processing and computing for services using this technology is not done in the mobile device but in the cloud. This results in the reduction of the needed computing capabilities and required resources for mobile devices, enabling them to support rich media applications. The mobile devices act like thin clients that are used to access services. [1]

Mobile device capabilities are increasing at a rapid rate and there is growing need for mobile applications to help people in their day to day activities. Mobile application developers are taking advantage of cloud computing to help alleviate the limitations that mobile devices have such as limited storage, low battery life and limited network bandwidth. This has been enabled by the availability of mobile phone sensing modules (e. g. Global Positioning System (GPS) for getting location of a phone, navigation, image processing and natural language processing for example Cortana for Windows Mobile Phones and Siri for iPhones). [1]

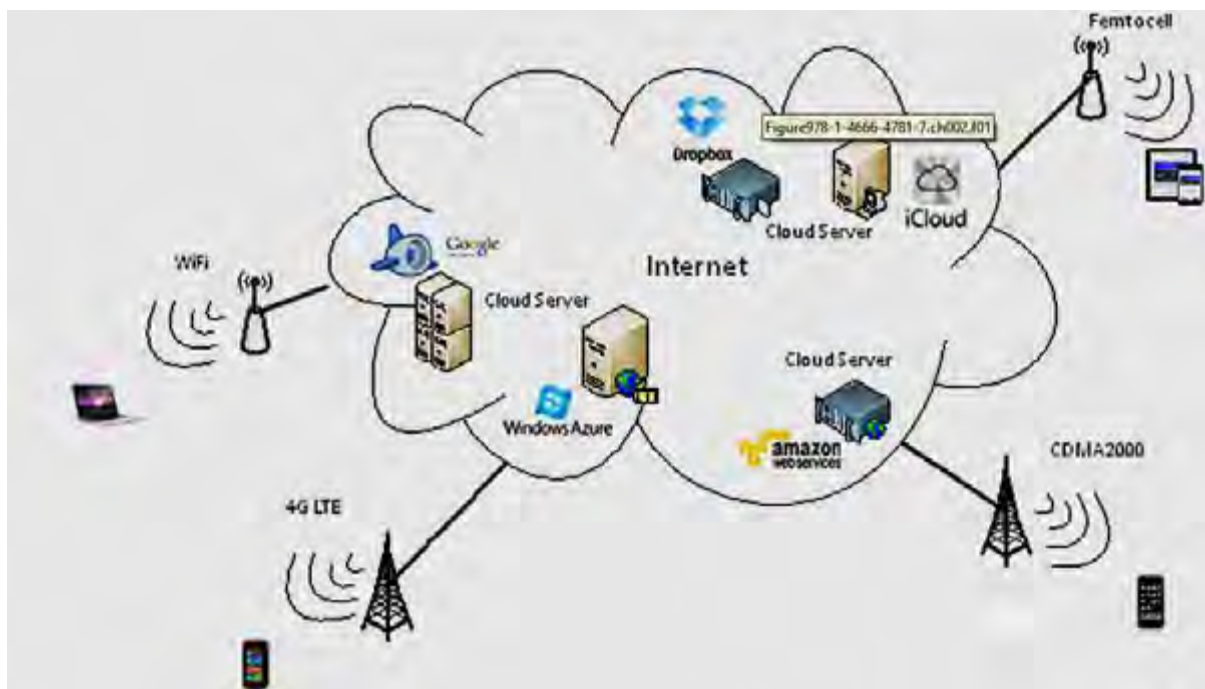


Figure 2.6 Internet access technologies for cloud computing [1].

Advancements of mobile communication technologies like Wireless Local Area Networks (WLAN), Wi-Fi, satellite communication and cellular technologies like LTE, WIMAX and 5G networks benefit the mobile cloud computing technology. Figure 2.6 shows the various internet access technologies that can be used to offload all the intensive computing and processing for mobile applications to the cloud. The choice on which wireless method to use is dependent on the capability of the mobile device and the strength of the wireless methods at a given time. [1]

2.3.1 Challenges in Mobile Cloud Computing

High Network Latency and Limited Bandwidth in Mobile Networks

High network latency and limited bandwidth constrains mobile networks. This affects the performance of mobile applications that require high processing capacity (e.g. online gaming) [6], [12], [1]. Transmission of large amounts of data with low bandwidth results high latency and unexpected error messages being displayed on the application [13]. Switching between high and low bandwidth may cause the application to freeze [13]. This affect the user's perception of mobile cloud computing services and may determine whether they will continue to use them or not [1].

Unavailability of Wireless Connection

Mobile cloud services /applications need internet connectivity to get data from the backend and this is provided via wireless networks whose availability varies depending on the user's location due to them being mobile. This is a major challenge because for mobile cloud applications to work effectively, a reliable, constant and speedy Internet connection is required [1].

Security

In Mobile Cloud Computing, data is transferred over an open air interface and this makes the data more susceptible to malicious attacks. Mobile devices have some challenges in that some of them may not have the capability to compute sophisticated security algorithms or to run sophisticated, computation intensive antivirus programs in order to improve the security [1].

Privacy

Privacy issues may result due to the distributed storage in cloud computing. These issues mainly affect public clouds which have a disadvantage that the files that are uploaded onto the cloud may be used by the service provider without the user being given notice ahead. [1]

According to Google's terms of service and I quote, "When you upload, submit, store, send or receive content to or through our Services, you give Google (and those we work with) a worldwide license to use, host, store, reproduce, modify, create derivative works (such as those resulting from translations, adaptations or other changes we make so that your content works better with our Services), communicate, publish, publicly perform, publicly display and distribute such content" [14]. Google also states that the rights that the users grant to Google on the usage of their uploaded files are only for the purpose of operating, promoting and improving Google's services and to develop new ones [14].

Guaranteed data security in Mobile Cloud Computing might be a challenge due to the cloud computing infrastructure that is distributed globally in different countries where legislations about privacy are different [1].

Mobile Devices Limitations

Mobile devices have limitations of memory, speed and battery life, especially for data intensive applications. The decision on whether mobile application data processing and computation is performed on the cloud or the mobile device is taken by considering the availability of the wireless network and the mobile device battery consumption rate. Energy consumption is related to the transmission rate and the channel conditions. If large amounts of data are transmitted over bad wireless network conditions, a large amount of energy is consumed from the battery hence the computations have to be done in the mobile device. [1]

2.4 Service Oriented Architecture (SOA)

Service Oriented Architecture is a way of using a distinct method that separates concerns. A large service need is solved by combining a collection of services that are autonomous and standardised, and each one of them will be solving part of the service need [2]. This method aims to enhance the efficiency, agility and productivity of the final service provided. In SOA, the capabilities of a service are published in a service contract and these services exist as

independent software programs designed to support SOA principles. Implementation of SOA can be done as a combination of technologies, APIs, products, supporting infrastructure extensions and various other parts. [2], [7]

2.4.1 SOA Principles

Below the principles of SOA are presented:

Standardised Service Contract

Service contracts are standardised documents that are comprised of a technical interface or service description documents that outline the capabilities or purpose of a service. These documents also have information about the service's functionality, how policies are asserted and attached and how data types and data models are defined. To improve the interpretability and overall predictability of services, service contracts should be granular, optimised and standardised. This also makes the end points of the services governable, reliable and consistent. [7]

Service Loose Coupling

In SOA, developed services, their service contracts, service customers and implementation should have reduced dependency and this enables them to be adaptively evolved overtime with little impact on each other. [7]

Service Abstraction

The service contract should only have essential information about the service and all the non-important and most of the underlying details of the service should be hidden. All the information about the service functionality, logic and technology must be hidden from all the other components of the system except the service itself. [7]

Service Reusability

For services to be interoperable with various systems, services should be generalised and reusable. The agility of businesses to meet new needs increases due to their ability to easily

respond to changes in the business environment by composing different services. Because service can be reused in other systems, the initial investment of making a service will yield big returns. [7]

Service Autonomy

Services should be able to control their underlying runtime execution environment in order to make the performance and behaviour of the service predictable when being reused and composed. This also increases the reliability of the runtime. [7]

Service Statelessness

Services should be stateless resulting in the management of state information being deferred. By doing this, the consumption of memory and system resources is minimised hence the stability of the service is increased. [7]

Service Discoverability

In service discovery a service registry is populated with service metadata for future reference. Service profile documents are written to supplement the service contract if the service registry does not exist. For a service to be reused, it has to be easily identified and interpreted. Services have communicative metadata that can be interpreted by both humans and software programs. This metadata is put on the service contracts and it clearly states the purpose and capabilities of a service. When a service discovery query is issued the service contract is referenced to. [7]

Service Composability

This is a form of service reuse and all the other SOA principles exist mainly to support service composition. It provides a medium by which the ultimate goal of SOA can be achieved. [7]

2.4.2 How Services Communicate in SOA

For services to be used by other services and programs, these entities have to interact and be aware of each other. They should be loosely coupled and these services exchange information

about each other via service descriptions. Services communicate via messages that are autonomous and self-governing. [2]

Figure 2.7 shows how services communicate in SOA systems

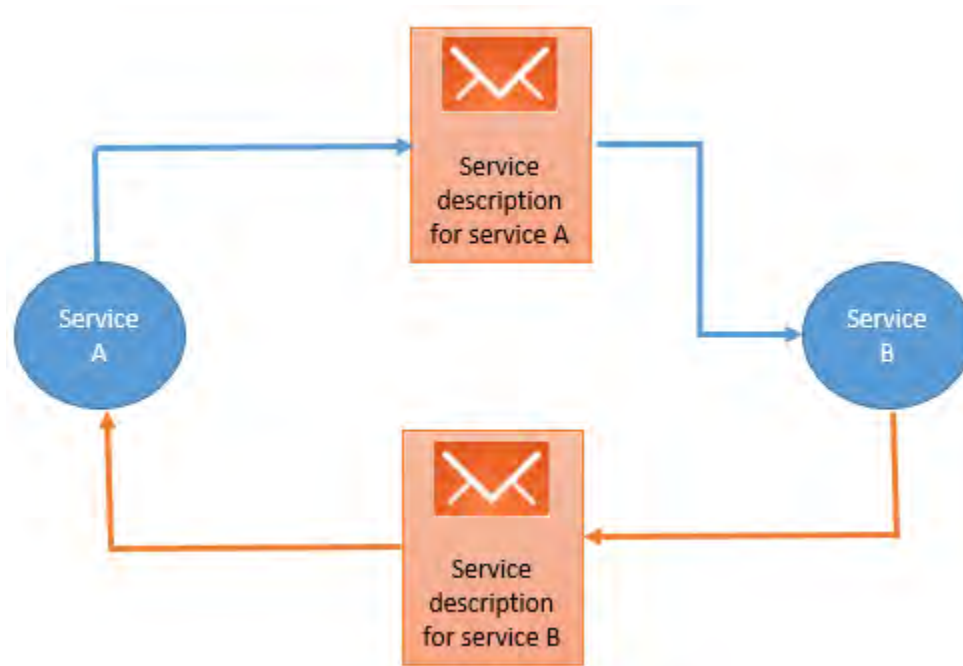


Figure 2.7 Communication of services in a SOA.

2.4.3 Benefits of SOA

Improved Integration and Interoperability

SOA reduces the cost and effort of either cross platform or cross application interoperability by the use of its vendor neutral communication frameworks, and highly standardised service descriptions and message structures. This allows for service interoperability. [2]

Service Reuse

Services are designed using standards in order for them to fulfil the immediate application level requirements. This enables them to support their reuse by other systems, hence investments into a service can be continuously leveraged as it can be used to build new service solutions. This reduces the cost and effort of building new service solutions. [2]

Service Composability

In order for service composition to occur, services must adhere to design standards that govern the allowable application level architecture extensions within each application environment. To perform service composition, only the knowledge of the component services or their service extensions is required, hence there is an advantage that a reduced skillset is needed. [2]

Legacy Investment Leveraging

Many legacy technologies now support service-oriented integration architectures and many organisations are now using technologies that deploy SOA. This results in the need to develop expensive point to point integration channels being eliminated. Leveraging of legacy systems lowers the cost and effort associated with the replacement of legacy systems and their integration to legacy and contemporary solutions in SOA. [2]

Focused Investments on Communication Infrastructure

SOA supports a common communications platform hence enterprise wide infrastructure can be evolved by investing in a single technology that is used for communication. This reduces the cost of scaling communications infrastructure. [2]

Use of Best of Breed Systems

In SOA, vendor neutral communications frameworks are used. Service designers are not bound to a single proprietary development or middleware framework hence they are free to choose the services that can be combined in order to accomplish a certain service task and also the methods on how to do so. This results in the quality of the services being produced being improved. [2]

Organisational Agility

Through proper design and standardisation, services can be reused and made interoperable by using a well-designed SOA system and this protects organisations from the impact of the evolution and change of service needs. This is important because a change in the needed services can be expensive, disruptive and damaging to an IT environment. To increase the

agility by which organisations respond to changes related to business processes or technological changes, organisations form loosely coupled relationships which allows them to evolve independently and adapt to changes imposed on them by each other as required. [2]

2.4.4 SOA Delivery Lifecycle

Figure 2.8 shows the different phases of the SOA delivery lifecycle. The SOA principles and characteristics considered during the first two phases of the lifecycle which are service oriented analysis and service oriented design [2].

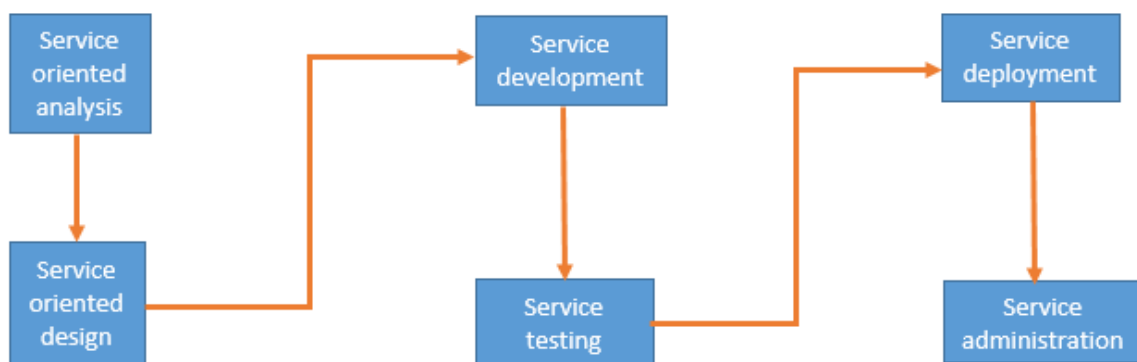


Figure 2.8 Phases of the SOA delivery lifecycle.

Service Oriented Analysis

The services needed to be built are identified and the parts of the solution logic where these services will be handled are outlined. [2]

Service Oriented Design

This stage is standards driven and it incorporates service oriented principles and industry conventions to determine how services can be constructed. [2]

Service Development

This stage deals with issues related to the development platform. The programming language and development environment are selected and these determine the physical form that the services will take. [2]

Service Testing

Before services are deployed into the production environment, they have to undergo rigorous performance tests to ensure that they perform up to standard and that the quality of the service is good. [2]

Service Deployment

During this stage the service interface and any middleware products are installed and configured into the production servers. In mobile services this would be the publishing of the service on the store or the installation of the service on a mobile device. [2]

Service Administration

Issues related to the management of the service like monitoring its usage and performance bottleneck detection are dealt with. [2]

2.5 Service Composition

Service Composition is the process of developing customised services by combining and linking basic services which may either be atomic or composite services into a more complex service that has features which are not present in the individual services [3], [4], [5].

Service Discovery architectures are used to discover services in a mobile environment [3]. A service provider publishes their services in a service registry. The service discovery method uses a template matching approach where the user requests are presented as request templates and through service discovery of the available services in the service registry, the services or resources matching these request templates are identified and then sent to the service requester [4].

If a service that is not pre-existent in the environment is requested, services that are already in the environment can be combined together by service composition methods in order to meet the user needs [3]. Figure 2.9 shows the service discovery process.

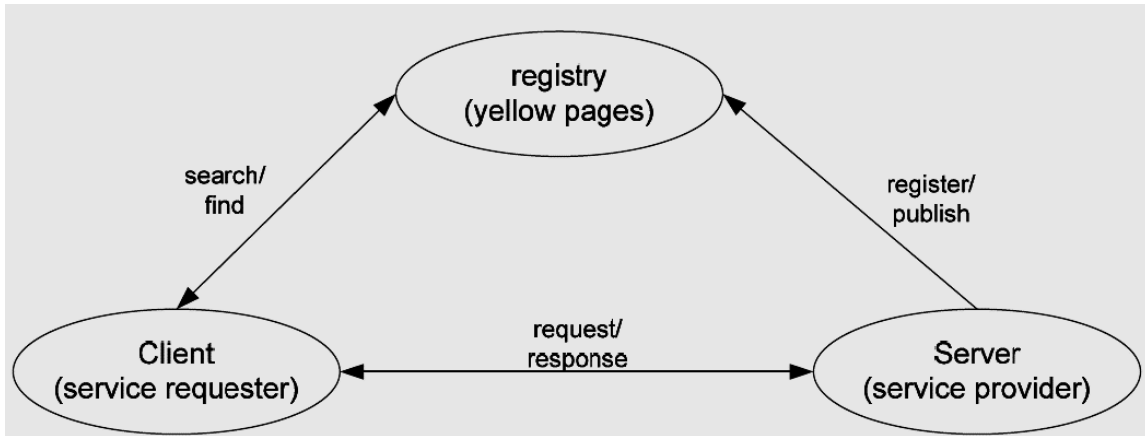


Figure 2.9 Service delivery process [10].

Composed services may be products from different organisations that will be required to cooperate during the composition process. These services may need to be transmitted to the user via a network infrastructure, which is also operated by other organisations. Figure 2.10 shows different organisations working together to provide a composed service to the user.

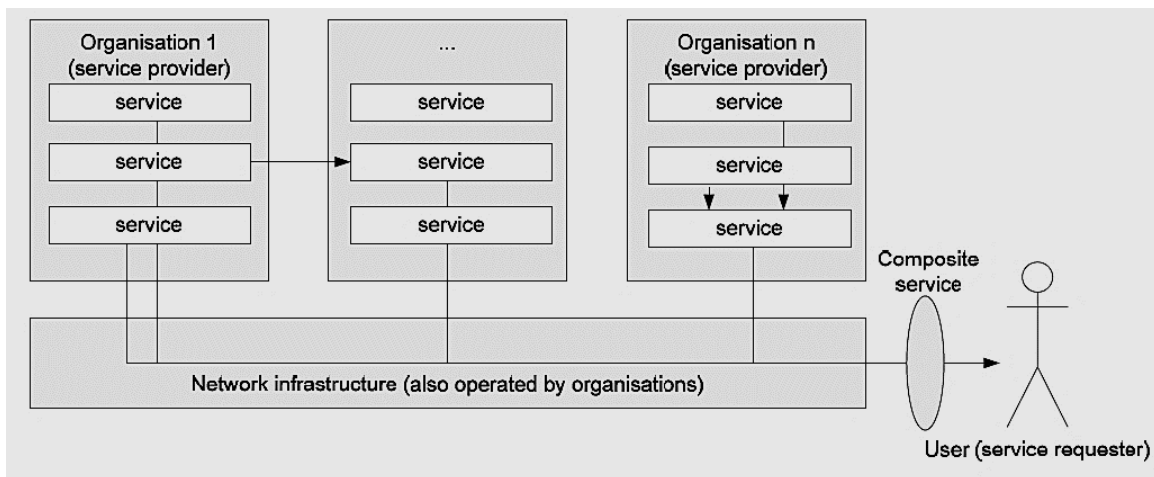


Figure 2.10 Provision of composed services by the collaboration of organizations [10].

Figure 2.11 shows how services are composed in systems using SOA principles.

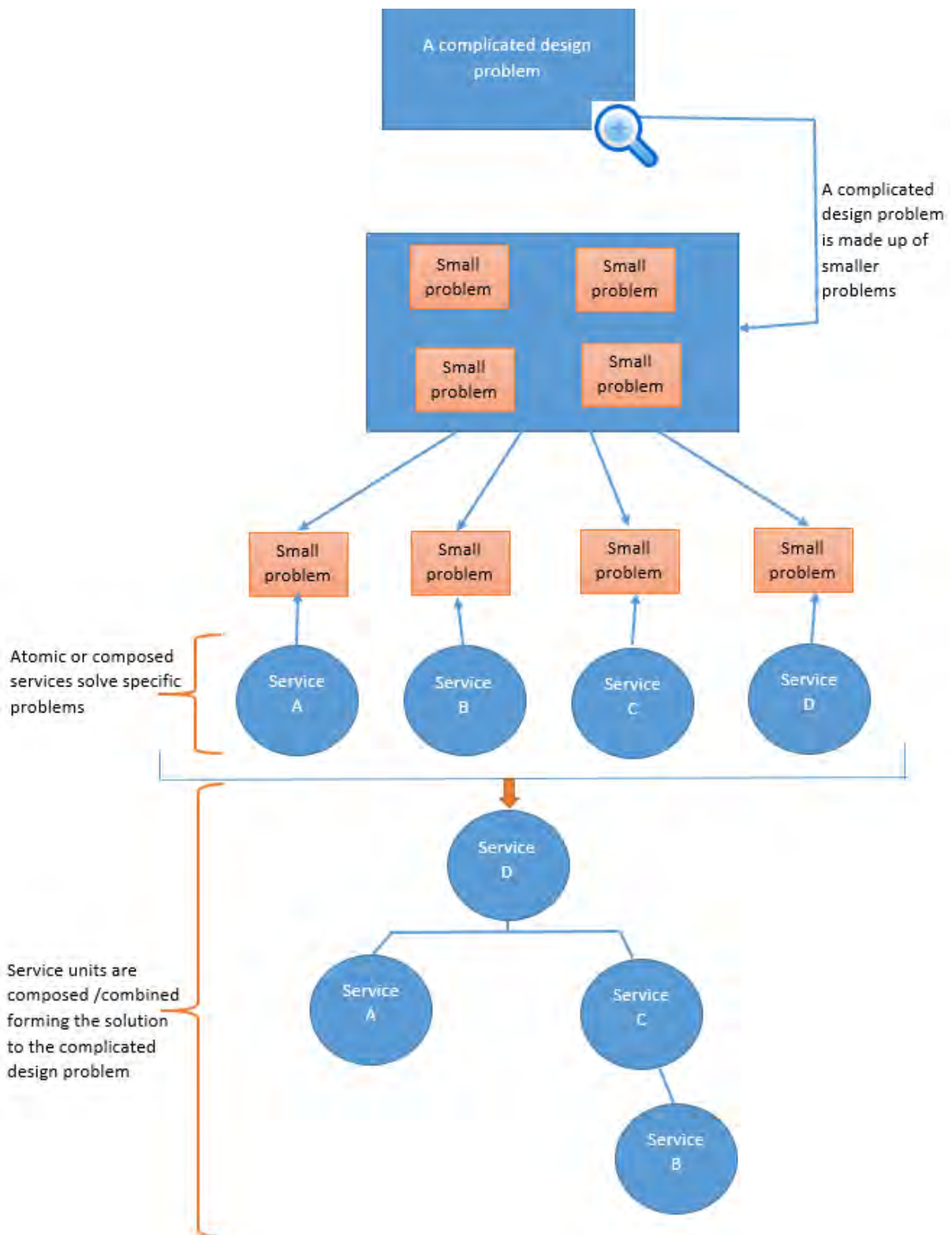


Figure 2.11 Service Composition using SOA principles.

A composed service system is comprised of services that can take up different roles depending on where they will be located in the overall composition configuration. The different roles that services can take up are either being members, controllers or sub-controllers. Designated

controllers will only have one role which is to control the composition of the other services. [7]

Figure 2.12 shows an example showing the various roles that services can have.

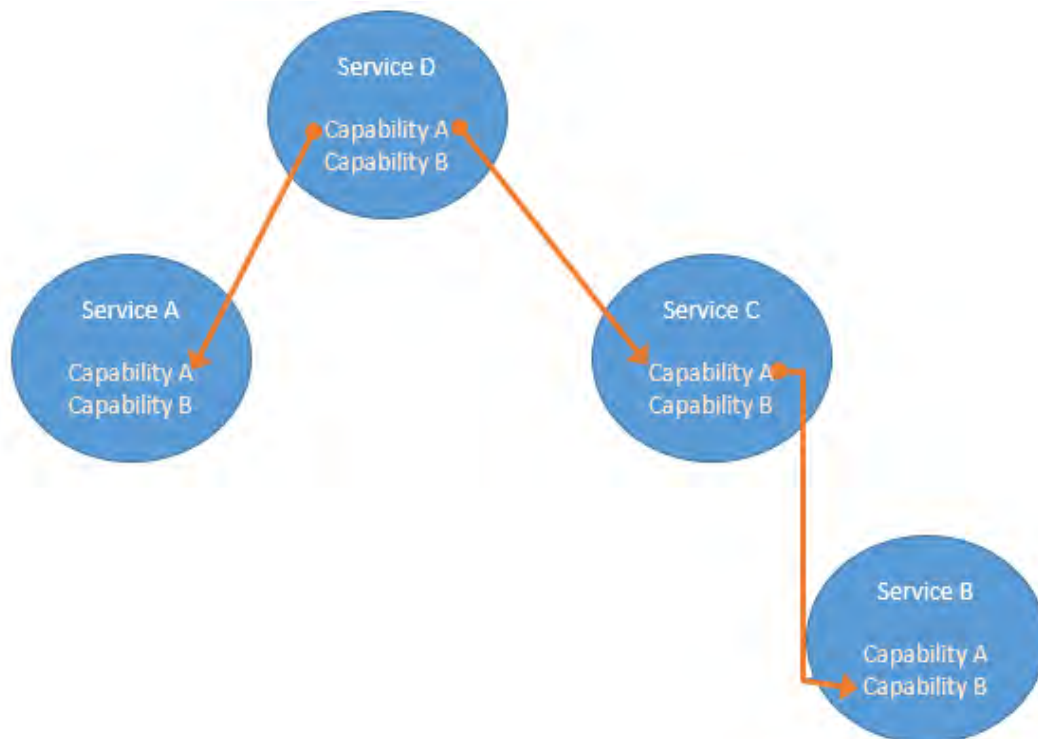


Figure 2.12 Service roles hierarchy.

Capability A of service D composes the capability A of service A and capability A of service C hence service D is the composition controller.

Capability A of service A does not compose any other service hence it is a composition member.

Capability A of service C is composed together with the capability A of service A and at the same time, the capability A of service C is composing capability B of service B hence it is a composition sub-controller.

Sometime an external program called a Composition Initiator may be connected to a composed service. This composition Initiator will be the one that initiates or triggers a composition but will not be part of the composed service. [7]

2.5.1 Design Characteristics of Composition Member Capabilities

Composition members have the following features:

- Reusability;
- High individual processing capabilities;
- Have flexible data contracts which can facilitate for different types of data exchange requirements, hence allowing the exchange of the same type of data at different levels of granularity;
- Runtime processing capabilities should be able to be optimised by the composed services in support of multiple, simultaneous compositions [7].

Some of the composition members will have to act as controllers or sub-controllers within different composition configurations.

2.5.2 Design Characteristics for Composition Controller Capabilities

Controllers are located at the top of the hierarchy and their capabilities contain and carry out logic that invokes the capabilities of other services. Services designated as controllers do not have the high performance demands placed onto composition members. [7]

Services that are controllers can be composed into a larger composed service but will not have reusability as their primary design consideration, even though the reuse principles are used where appropriate. Stateless is not emphasised as they may need to sometimes be statefull while the other composition members perform their duties in the overall composed service. [7]

To have an effective and efficient service composition, the service architecture has to be optimum. The runtime environment hosting the composed services should be scalable, reliable, should have dedicated clustered servers with fail over and should make use of mature runtime service technologies as required. [7]

2.5.3 Types of Research in Service Composition

- The first type of service composition research considers the architectures that support service composition. A declarative specification of a composite service is assumed and the tasks of service discovery, integration and service execution are performed. [3]
- The second type of research in service composition is one in which programming languages are used to formally specify and design both individual and composite services [3].

In this research, the second method of research was conducted.

2.5.4 Methods of dynamic service composition

There are two main methods of service composition and the choice on the best one to use is determined by the use and required efficiency of the composed service. After service composition is completed, the service has to be predictable and unanticipated behaviors must be minimized. The descriptions of the two main methods of service composition are presented next. [15]

Method 1: Composite Service Interface (CSI) method

For this method, two or more service components interact and combine into a single service via a service composition architecture during runtime. This method can be used when the combined service does not require high level of software performance. [15]

Disadvantages of this CSI

Lower operational performance and larger response time will result if the component services are distributed throughout the network and are not located on the same network node. It is very difficult to reuse the composed service since all the service components that make up the composed service are not held as a single entity. [15]

Advantages of CSI

Service composition is performed at a faster rate because the code of the component services is not moved or modified in any way in order for the new composite service to be made. [15]

Figure 2.13 shows how the Composite Service Interface is set up.

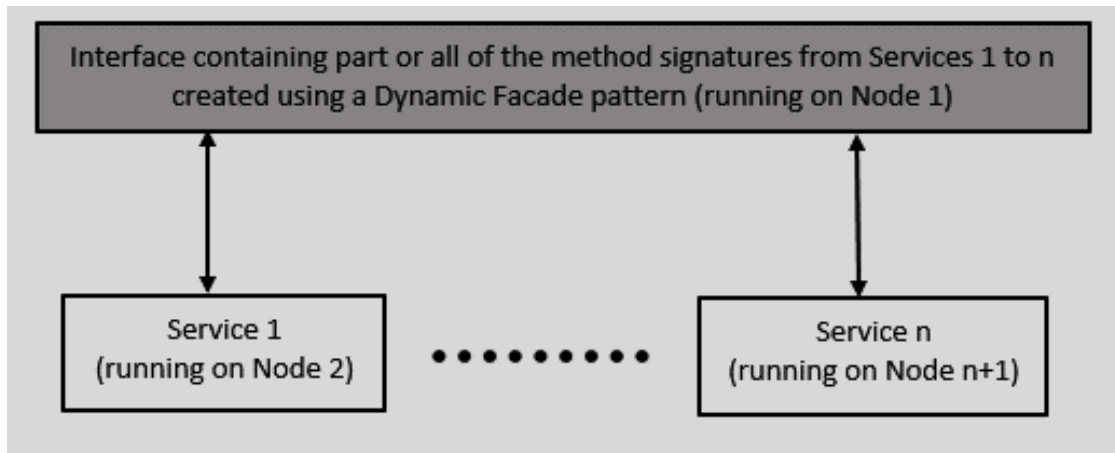


Figure 2.13 Set up of the composite service interface [15].

Method 2: Stand-Alone Composite Service method

For this second method, a new composite service is created with all its functionality contained in a single new component. This composite service contains all the basic set of operations for all the combined component services. This method is used when the performance of the composite service is more crucial. As shown in figure 2.14, all the code of the component services will be located on the same node hence the composite service has better performance. [15] Figure 2.14 shows the stand-alone composite service method [15].

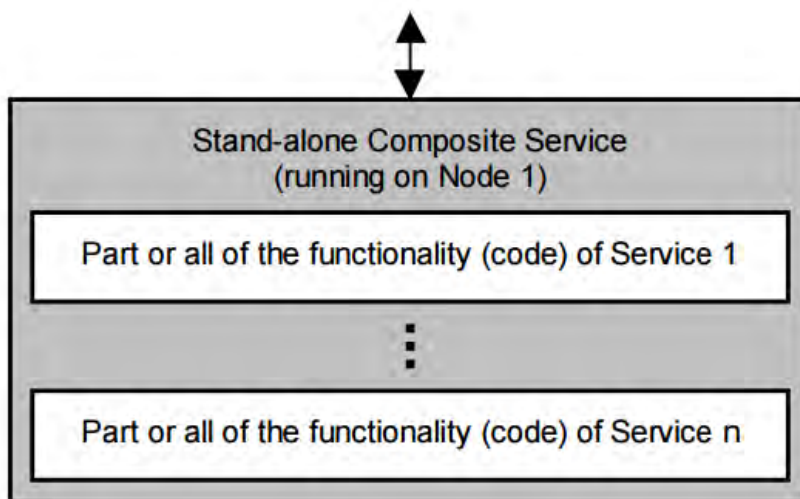


Figure 2.14 Stand - alone composite service method [15].

There are two ways of performing the stand-alone composite service method and these are presented next.

First Option

Figure 2.15 shows the first method for making a standalone composite service. The service components remain independent. Input is sent to the first component whose output is sent to the next component until the output of the last component is sent out as the output of the composed service. Rearranging the components will result in a different output. [15]

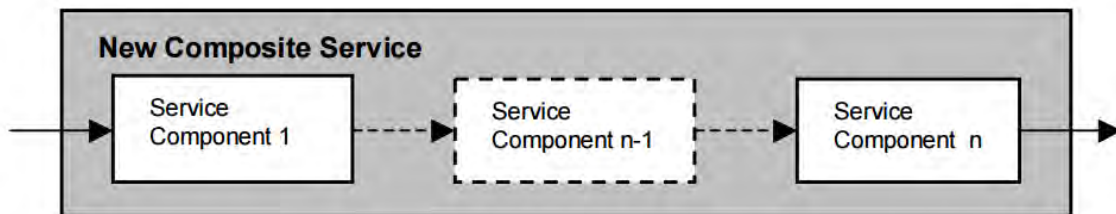


Figure 2.15 First option for making a stand-alone composite service [15].

Figure 2.16 shows a more complex version of the first option shown above whereby some of the output of a service component is feedback into that service component [15].

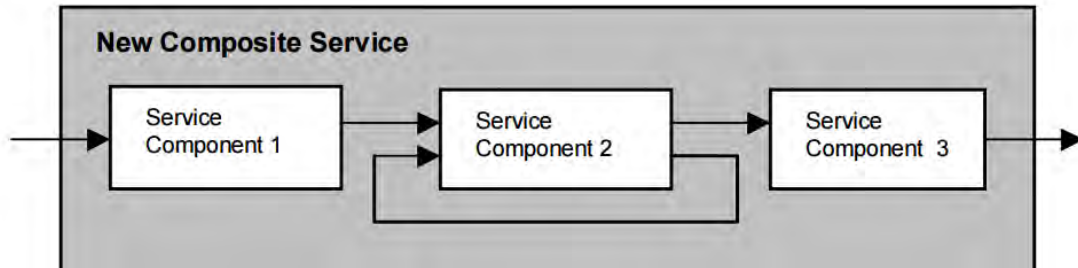


Figure 2.16 More complex standalone composite service method [15].

Second Option

Since some of the code in the component services are only useful for the component service and not the composite service, this method only makes use of the parts that are useful in the composite service and these are assembled together to make a new composite service. This new composite service is composed at runtime and has one body of code. The composite service must contain all the attributes of all the composed component services. Standalone service can be reused and easily composed with other services. They also execute the internal transmission of messages at a higher level of performance since all the code is executed at the

same location. [15]

Figure 2.17 shows the second method for making a standalone composite service [15].

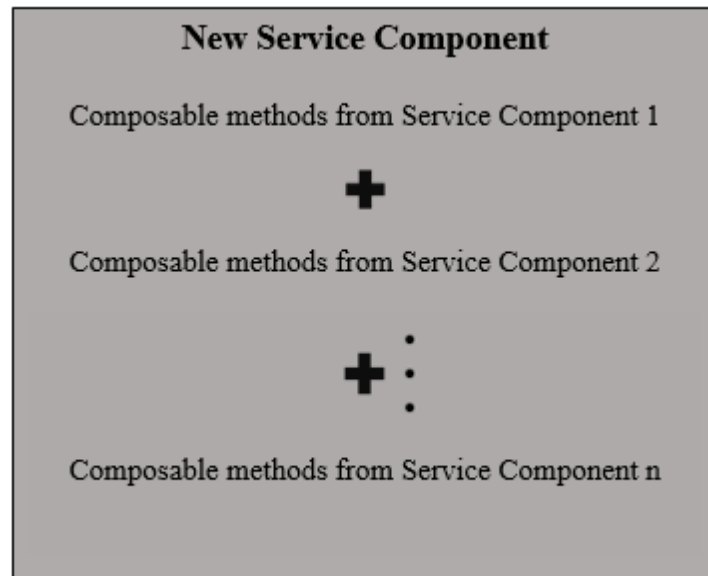


Figure 2.17 Second method for standalone composite method [15].

In this work, the second method of the Stand-Alone Composite Service method was used to compose the mobile services required to produce a mobile Smart Shopping service.

2.5.5 Risks Associated with Service Composition

Single Point Failure

If a service that has been composed in many other composed systems malfunctions or stops working, all the composed services for which the affected service is a part of will be affected. This can have serious impacts on an organization's business process. [7]

Composition Members as Performance Bottlenecks

Runtime performance of a composed service system is dependent on the collective response time of all the individual composition members in the system. Poor runtime performance, of a composition member results in the overall performance of the system being lowered. [7]

Governance Rigidity of Over-reuse

A service which has been over-reused in a business process, and is also composed into many other composed service systems can be very hard to change. This is because any design change to the logic of this service can affect the whole business process or the composed service systems for which it is part of. This risks them not functioning well. This problem makes the business process to be less adaptive and less responsive to change, which may result in the system being out of date. Expert design of service where effort, care and great insight is taken can minimize this problem. This ensures longevity of the service contract. Versions of services can be made when there is need to change a service in order to minimize this problem. [7]

2.6 Smart Systems

Smart systems have the ability to use various information in order to provide the relevant services to the user at anytime and anywhere. The use of smart systems has been motivated by the increased convergence and development of various technologies like cloud, data processing and networking that are now available at lower costs than previous years. This has opened more doors to many technological innovations and development of services. IBM started an initiative to develop a smart planet where various technologies are used to create smarter environments (e.g. smart homes, smart cities, smarter governance and smarter industries). [16]

Smart Systems are also known as intelligent systems and they make decisions based on information obtained from the environment by using various sensors which monitor, measure or analyse the surroundings. They also have the ability to use and share the vast amount of information that exist in the service infrastructure. Smart systems aim to effectively use and analyse information, and for the surrounding physical and digital objects such as machines, devices, sensors, businesses and people to interact together and be involved in proactive service delivery. [17]

Smart systems aim to enable services that can be delivered:

- Anyhow;
- Anywhere;
- Always.

These service attributes allow for the availability of information at the user's fingertips. Smart systems have the ability to behave as a digital personal assistant that will analyse the user context and usual behaviour at any given time and provide ways or solutions to help the user in their activities. [17]

There are different types of smart systems that can:

- Collect usage performance data to be used by service designers to produce more effective versions of the system in the future;
- Collect data, process it and present the information to the user in order to help them in making decisions, (e.g. in this work a parking system is designed to inform drivers about the available parking spaces in a shopping mall);
- Use collected data and take action without direct intervention of the user [18].

2.7 Previous Studies Conducted on Related Topics

The growing interest in the research aspects discussed in the above sections has resulted in many research groups doing intensive studies in these areas. In this last section, a literature review of some of the research work that has been done is presented. Firstly, the research work that has been done on MCC is presented. Research work done on Service Composition on the cloud is then presented. Finally, the research that has been done on various smart systems is presented.

2.7.1 Issues and Current Solutions of MCC

Lalit Kumar et al. [19], did a survey on Mobile Cloud Computing (MCC) and an overview of the definition, architecture and applications was presented. They went on to highlight three issues in MCC and present the existing solutions to them. The issues they discussed were:

Limitations of Mobile Devices

Mobile devices are resource constrained and have limited computing capabilities and energy resources. Lalit Kumar et al. [19] suggest that this problem can be reduced by duplicating mobile devices to the cloud using image processing and virtualisation for Data Intensive

Computing (DIC) and energy intensive computing to be processed. They also suggest having an effective elastic application division mechanism and migrating tasks from the terminal to the cloud. [19]

Quality of Communication Networks

Lalit Kumar et al. [19] stated that the data rate in MCC is constantly changing and the availability of the wireless network connection is inconsistent. The problem can be alleviated by upgrading the bandwidth of wireless connections and using regional data centres. The data centres and resources of internet service providers are normally far away from the end users and this problem can be overcome by deploying the application processing nodes at the edge of the cloud. This reduces the data delivery time. [19]

Division of Application Services

Applications have to be divided in order for data and compute intensive applications to be deployed on the cloud. This faces the challenges of data processing in the data centre and mobile devices, network handover delays and data delivery time. Lalit Kumar et al. [19] recommend the use of dynamically optimised application push in cloud and using the mobile terminals as thin clients. [19]

2.7.2 Mobile Applications using MCC

M. Reza Rahimi et al [20], presented the concepts of MCC, the technology's challenges, its opportunities and examples of applications that use the MCC technology in various domains and disciplines. Finally, open research issues to be studied were outlined. The examples of applications that use MCC that were presented are as follows:

Mobile Learning

Mobile technology is used to improve the knowledge, skills and behaviour of people through training, education or performance anytime and anywhere. The example given by the author of this application is the Mobile Learning System(MLE) which allows for the dynamic interaction between teachers and students and also between students. Based on the user's needs, context

and learning pattern, appropriate courses, teachers and classmates are assigned and recommended to the users. Power efficient content delivery mechanisms have to be used. M. Reza Rahimi et al [20] suggested the use of MCC to enhance power efficiency and content delivery. [20]

Mobile Commerce

Mobile commerce (m-commerce) applications include mobile transactions, payments, mobile tickets and advertising done by sending messages. M. Reza Rahimi et al [20] presents the two emerging classes of m-commerce applications and the first one uses free mobile applications for advertising (e.g. Pandora) which uses user context to advertise to local businesses. The second class presented are social networking and location based services (e.g. Yelp) where mobile users can share their experiences through images, multimedia and reviews. These types of applications can help both business owners and customers to provide better services and good and competitive deals. Due to the high computation and storage needs of these applications that require complex data processing and sharing, M. Reza Rahimi et al [20] suggested the use of Mobile Cloud Computing. [20]

Mobile Healthcare and Wellness

Treatment of chronic diseases depends on the continuous reports of the patient's symptoms and side effects, and this information can be obtained by using mobile health platforms that have the ability to collect relevant data and share it with physicians anywhere and at any time. Challenges of these applications are the real time processing and storing of large volumes of patient data, privacy and security of medical information. M. Reza Rahimi et al [20] presented an example of a context aware application using these principles called Mobile Cloud for Assisting Healthcare (MoCAsH). This application uses contextual data obtained from sensors, mobile devices and user profile information. Interaction with the system is done through a cloud portal that uses interfaces for communication and management of users, mobile agents and the cloud. [20]

Mobile Social Media

These mobile applications are made based on social networks such as Facebook which become the defacto means of entry login into the mobile applications. This bridges the gap between

virtual social communities and physical world information and services. The term Mobile Social Network (MSN) is used to refer to such applications. The use of location based services allows the users to get access to localised information like maps, driving directions, sharing data, search for point of interests and socialising with family and friends. M. Reza Rahimi et al [20] assumed that MCC will be the next generation platform that will be used for MSN applications by providing a scalable and elastic platform for information processing and storage. [20]

2.7.3 Feasibility of Mobile Computation Offloading and Data Backups

Marco V. Barbera [21] presented the problem of feasibility and cost of mobile computation offloading, and mobile software or data backups to the cloud with regards to bandwidth and energy consumption by using real life scenarios. Two types of software clones are made for real devices on the cloud and these are the off-clone which supports offloading and the back-clone for restoring the user's data and apps when needed. Eleven Android smart phones were used as primary mobile phones by some participants and these phones were cloned on the Amazon EC2 cloud. The results obtained from the experiment are as follows [21]:

- On average the smartphone users were connected to the Wi-Fi 50% of the time;
- On average 50% of the smartphone users did not have Wi-Fi coverage for 2 hours at most and this was likely to be during the time when the users were commuting between places with Wi-Fi for example between travelling from home to work. This meant that sync operations of the user's data on the devices would have to wait until Wi-Fi access was available to occur [21];
- Synchronising back-clones required 4 times less network traffic and about 3 Kilojoules less energy overhead than when synchronising off-clones;
- The difference in overhead during synchronising the two clones decreased drastically as the frequency with which the sync operations are done decreases. Syncing every 30 minutes is the reasonable time but increases the energy spent by the phone battery to do so with 11% of the battery being spent for the off-clones and 8% for the back-clones [21];
- Wi-Fi technology has lower overhead than 3G technology [21].

Marco V. Barbera [21] then concluded that continuous update of software clones in the cloud can be sustained in mobile cloud computing with reasonable overhead in terms of bandwidth and energy especially if the sync intervals are not too short.

2.7.4 Security and Privacy in MCC

Hui Suo et al [22], presented the security and privacy issues of the mobile terminal, analysis of the mobile network and the cloud. They also presented the current security and privacy approaches available. Hui Suo et al [22] said that mobile terminals have open operating systems, can support third party software and are able to connect to the internet anywhere and anytime. This poses problems and the authors described them as below:

Malware

Can be downloaded automatically along with other useful programs and applications. They can also be transmitted via the USB interface, 3G network, Bluetooth or MMS attachments which make their prevention difficult. Malware can get illegal access to a mobile user's personal information. This can lead to increased data flow resulting in the user paying more money for data usage to their network service provider. More complex anti-malware which detect and prevent malware is needed but this is limited by the resources in mobile phones. Hui Suo et al [22] then went on to recommend two main ways to combat malwares and the first one is deploying anti-malware software to the cloud which reduces resource consumption in the mobile device. Cloud AV is an example of an antivirus that is provided as an in-cloud network service. The second method is users to be careful of their behaviours. [22]

Software vulnerabilities

Mobile phone users manage their phones through mobile phone management software. This software manages files in the phone through content synchronization between the phone and the computer and File Transfer Protocol (FTP) is usually applied in this process. User name and password of FTP is transferred over the network and saved in configuration files in clear text. This makes it easy for malicious modification of personal data or its intentional delete due to illegal access of the phone, using FTP from computers in the same network. Application software is also relatively not rigorous hence intruders can attack the mobile phones via a bug

of the application software. The operating system manages and controls hardware and software resources and its software is complex and will have coding bugs which may be used to destroy the mobile phone by attackers. Mobile users also usually lack security awareness and may mis-operate hence there is need to detect and prevent anomalous behaviours of users. Hui Suo et al [22] suggested that these software problems can be prevented by timely updating the phone operating system by downloading and installing its patches or revamped versions. They can also be prevented by checking software integrity and legitimacy before installing any software. Improving the security awareness of users is also very important for them to avoid the installation of unauthorised software, not clicking unexplained links, being careful of receiving data from strange phones and closing the Bluetooth and Wi-Fi interfaces when not in use. [22]

Hui Suo et al [22] went on to suggest methods of protecting Mobile Network Security which are data encryption and researching the security protocols of the mobile network. They finally outlined some Mobile Cloud Security measures which are:

- **Protection and platform reliability** can be improved by the cloud providers integrating security technologies such as VPN technology, authentication, access control, encryption and backing up user's data on the cloud. These methods prevent Deny of Service (DOS) attacks and allow for the recovery of user's data in the case that they get attacked; [22]
- **Data encryption and key management.** Data need to be encrypted during storage and transmission. It should be stored as cipher text in the cloud hence effective analysis and processing of cipher texts should be done; [22]
- **Authentication and access control.** Authentication methods can either be user centric where user identification methods are used or behaviour authentication where the user's habits and behaviours are analysed and this prevents the risks of fraud in the mobile cloud. Access control methods can also be used for cloud access; [22]
- **Privacy protection.** Governments and international technology organisations developed privacy protection plans, strategies and laws. An example is the Platform for Privacy Preferences (P3P) which was announced by the WWW consortium as an agreement on personal data privacy protection between network service providers and this has gained usage in internet sites. [22]

In the next section, a discussion about service composition in cloud computing is done.

2.7.5 Cloud based Middleware for Dynamic Service Composition

Jiehan Zhou et al [23] proposed a Cloud based Middleware for Dynamic Service Composition (CM4SC) into a conventional cloud architecture to allow for automatic composition planning, service discovery and service composition. This middleware was implemented on the Windows Azure cloud platform in order to support on-demand dynamic web service composition. This middleware layer is placed between the application layer and the platform layer of the cloud architecture. [23]

Figure 2.18 shows the proposed cloud architecture for dynamic service composition.

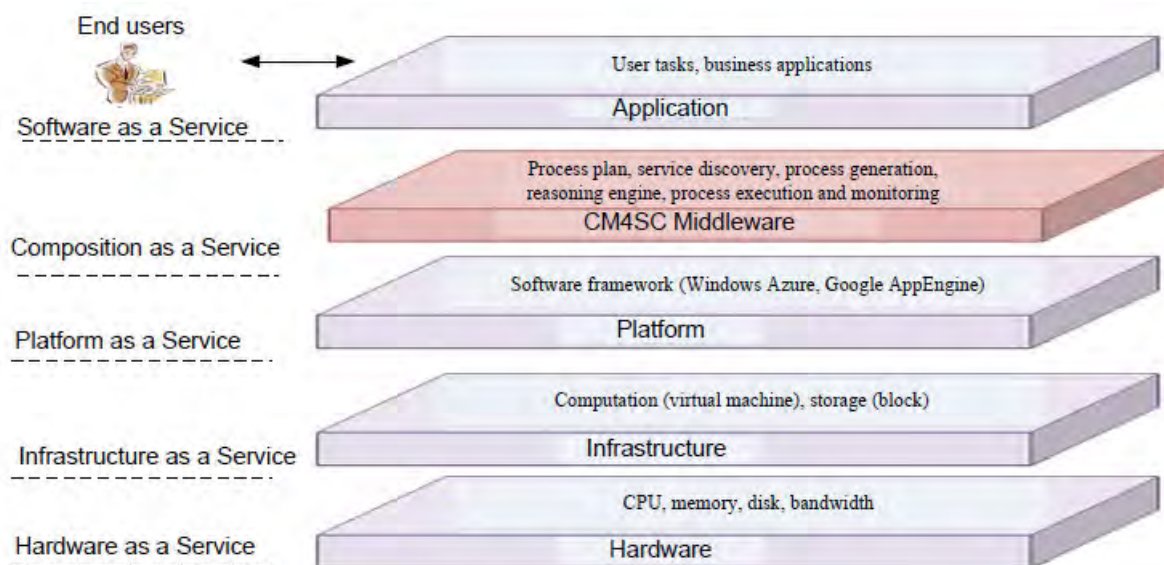


Figure 2.18 Proposed architecture for dynamic service composition [23].

Jiehan Zhou et al [23] realised that the basic components for conventional service composition are similar to that for cloud service composition but the latter accelerates rapid dynamic service composition, increases availability and scalability of service composition with more guarantee than the conventional methods. Jiehan Zhou et al [23] also demonstrated that existing cloud computing environments offer the tools needed to implement CM4SC for example, they used Windows Azure and Windows Communication Foundation (WCF) to implement cloud services and Windows Workflow Foundation (WF) was used to describe the internal flow, states and transitions of a task. [23]

2.7.6 Quality of Service (QoS)-aware Composition Methods

Due to the limited number of web services in cloud platforms, it is challenging to compose services in order to meet the many application requirements of users. This is because only part of the needed services for composition will be in the cloud and the rest will have to be invoked outside the cloud in a cross platform way. Lianyong Qi et al [24] investigated a QoS-aware composition method for supporting cross platform service invocation in cloud environment. They suggested a decision making method to determine whether a QoS aware Web Service Composition (WSC) problem has a QoS qualified composite solution to avoid unnecessary time cost. A QoS- aware service composition method called Local Optimization and Enumeration Method (LOEM) is then proposed, aimed to improve the efficiency of web service composition when many composite solutions are available. This method could also help to build flexible and scalable cloud platforms. [24]

2.7.7 Virtual Services in Cloud Computing

Jichen Fu et al [25] proposed the concept of virtual services which are used to bridge semantic gaps in the event of a service composition failure, enabling the composition process to continue. Virtual services only exist conceptually with no physical counterparts. One of the common techniques used in service composition is the Artificial Intelligent (AI) planning-based web service composition technique in which web services are modelled as AI planning actions. A web service composition problem is transformed into an AI planning problem [25]. This method has a problem that when services are being composed to meet a need and the majority of the services are available except maybe one of them which may for example be unavailable, or would have stopped working, the service composition process will return a failure although the majority of the services were available. [25]

Figure 2.19 shows the concept of virtual services. These are denoted as dotted squares and they belong to a virtual cloud that is outside of the clouds provided by cloud service providers (e.g. Amazon, Microsoft and Google). Jichen Fu et al [25] state that conceptually, virtual services are treated similarly to physical services and when being used they are removed from the virtual cloud and put into a real cloud. Jichen Fu et al [25] also discusses the system model of the virtual services. Their algorithm is then identified and a Graphplan-based algorithm is

presented. Finally, improvements are discussed, a conclusion and future recommendations are given. [25]

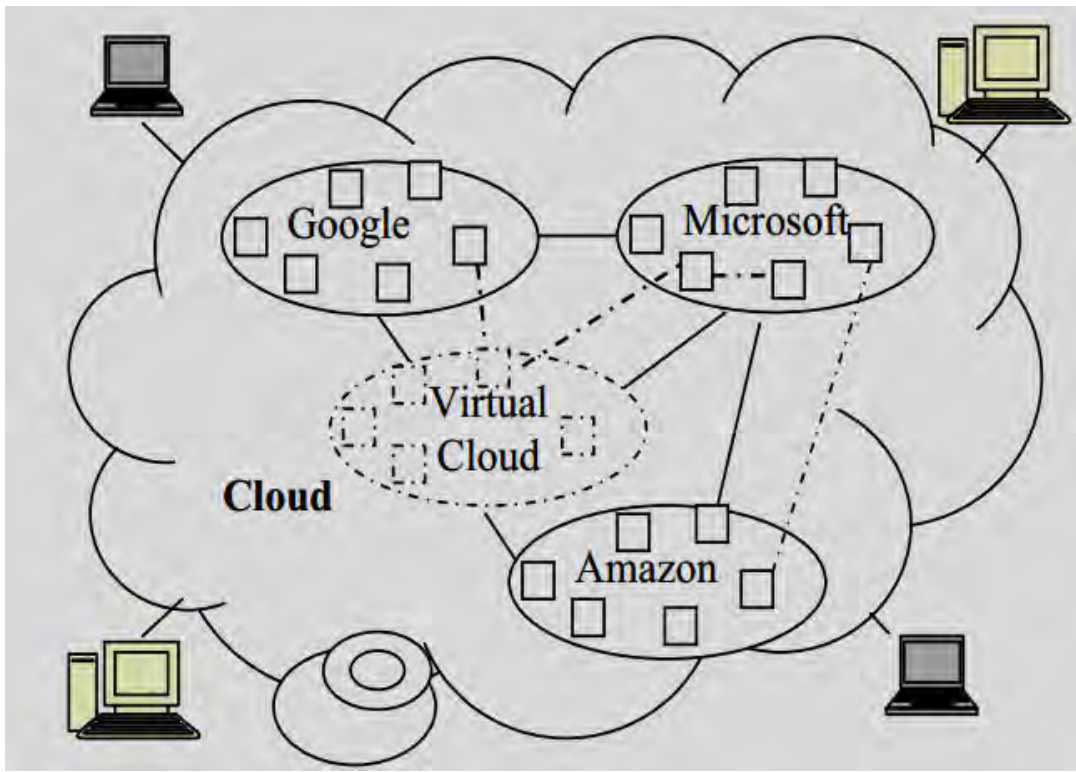


Figure 2.19 Virtual services [25].

2.7.8 Service Composition and Testing for Cloud Computing

In traditional service-oriented computing, service composition and testing are distinct processes. The service interface, which describes the input/output information of a service is used by developers to perform service composition. Developers may want to have more control in selecting service implementations hence Wei-Tek Tsai et al [26] proposes service injection as a way for developers to designate specific service implementations. The combinations that can be made using the services published in the cloud for service composition are too many and testing them is very difficult. They propose a process for testing these composed services that is based on service level MapReduce. This method makes it easy to test composed services that may exist in large numbers in the cloud architecture. [26]

Below is a discussion about smart systems that can be developed using composed services.

2.7.9 Applications of Smart Systems

Smart Utilities

Smart grids for monitoring the consumption of utilities like power and water adapt to the changes in supply and demand of the resource in order to balance them. They are also predictive which enable the operators to plan and operate the state of the systems in advance. These systems engage with customers through smart meters and the grids have self-optimisation capabilities to maximise the efficiency of the operation. [18], [17], [27]

Smart Cities

Smart cities are urban developments where various digital information and telecommunication technologies are integrated in order to efficiently manage the city. This makes the city's networks and services like transport, water supply, waste disposal more efficient. This will result in the needs of the city's inhabitants to be met more effectively. By 2050 more than 70% of the world's population will be living in cities and this was stated in the UN World Population Prospects report [23]. This raises the need for effective management of resources and for having smart cities. [18], [17], [27]

Smart Transportation

The availability of sensors and monitors on the roads or monitoring traffic congestion in the roads by the means of the satellites is being used to develop smart transport systems. This information is sent to a central data centre where it is analysed and then sent to the users. This information can be used to re-route the traffic flow to a less congested road hence the motorists avoid delays hence saving the drivers time. [18], [17], [27]

Smart Shopping

Smart shopping systems are used to collect information that is relevant in assisting a user of the system to enjoy their shopping in an effective manner. This research is concerned with a smart shopping system to assist shoppers at Canal Walk Shopping Mall in Cape Town. [18], [17], [27]

Vinh-Tiep Nguyen et al [28] proposed a smart shopping assistant system which uses mobile devices to identify products in stores then give the customers information related to the respective products. This information will be obtained from online commerce systems and also social media [28].

Chia-Chen Chen [29] investigated a context aware Intelligent Shopping-aid Sensing System which uses sensors and Radio Frequency Identification (RFID) for the recognition, collection and delivery of customer contexts. Analysis of this information provided instant and personalised information to the customers about products. This information can also be used to promote products and online marketing. [29]

Further research was done by C Tian et al [30], where a smart virtual shopping centre is developed that allows for the intelligent agent browsing of goods, smart search and product recommendation capabilities. Analysis of these shopping processes and behaviours of people in real shopping environments can be used to obtain desired product information. [30]

Jin Liu et al [31] proposed a location technology-based shopping service system. This system gives customers in a shopping mall information about products, the user's current location in the mall and gives them a shopping route and navigation information for choices in a shopping list. [31]

More research about smart shopping was done by Phani Bhushan. Avadanam and Nuthan Seegehalli Hanumanthappa [32] where the authors proposed a smart shopping framework where a retailer provides displays of their products in the display portion of the shop but keeps the products in the inventory department of the shop. A shopper scans the products they want to buy and sends this information to the retailer via a wireless receiver. The retailer then sends the selected goods to the checkout part of the shop where the customer pays for them. [32]

Chia-Chen Chen [33] used sensor technologies to develop a smart assistant for home furniture shopping which enables customers to easily locate products, confirm their detailed information and provide real time recommendations related to the customer's needs. [33]

In this research, a Smart Shopping mobile service for Canal Walk Shopping Mall is developed. It is made by composing different services in order to improve the shopping experience of people at the shopping mall. The services composed by this system are:

- Mall information service;
- Store category information service;
- Store information service;
- Parking service;
- Location service;
- Navigation from current location of user to Canal Walk Shopping Mall.

In the next chapter, the tools that were used to develop the Smart shopping service and the design process is described.

3 Mobile Smart Shopping Service Design

In the following chapter, a description of the tools that are used to design the Smart Shopping service and where they were used in the design process is outlined. A discussion of the design process is also done.

3.1 Development Tools

In order to design the Smart Shopping service system using the concepts described in the previous chapters and sections, a number of developmental tools were used and these are Visual studio, Microsoft Azure, Navicat and Postman.

3.1.1 Visual Studio 2015

Visual Studio 2015 is used in this research to develop and design the Presentation Layer, Business Logic Layer and the Data Access classes for the Smart Shopping service.

Figure 3.1 shows the start page for Visual Studio Community 2015.

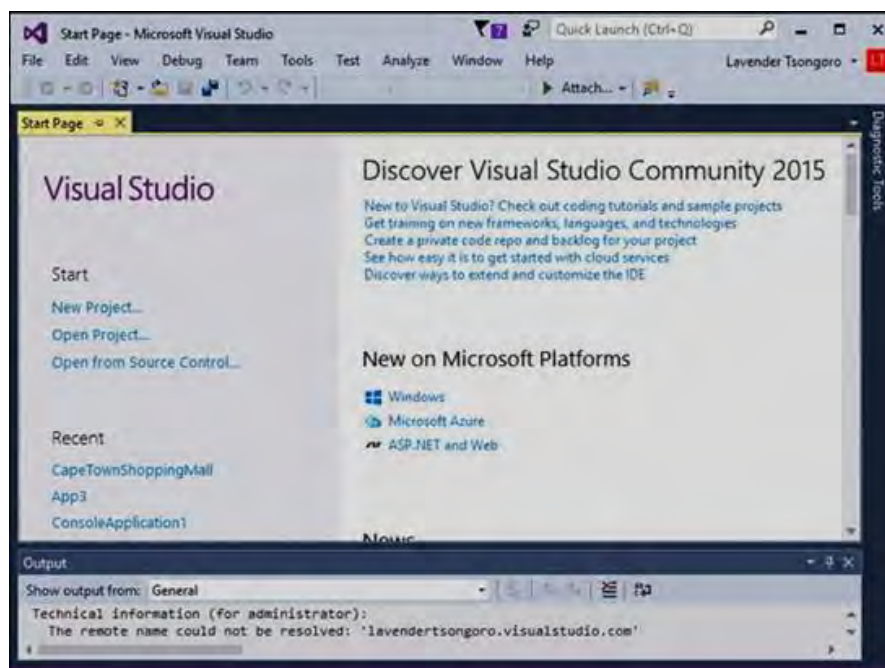


Figure 3.1 Visual studio community 2015 start page.

Visual Studio is an Integrated Development Environment (IDE) that is used to develop mobile applications for Windows, Android and IOS. It can also be used to develop desktop applications, web applications and cloud services. [34] Visual Studio also supports different languages (e.g. C Sharp (C#), Visual Basic (VB), HTML, Javascript, Python, C++ and F#). [34], [35]

3.1.2 Microsoft Azure

The data for the Smart Shopping service developed in this research is obtained from Canal Walk Shopping Mall's website and is then hosted on a cloud computing platform called Microsoft Azure which is Microsoft's cloud computing platform. This platform can be used to do the following:

Computations

Windows and Linux virtual machines can be provisioned on this cloud and highly available and infinitely scalable cloud applications and APIs can be created. Its service fabric can be used to produce some applications that are always on, scalable and distributed, and by using its RemoteApp, the client apps can run on any device. Large scale parallel and batch compute jobs can also be done on this cloud platform. [36]

Web and Mobile

Windows Azure's App Service enables the creation of web and mobile applications that can run on any platform and device. Back ends for mobile applications can be created in this platform and the Power Apps can be used to provide a platform for mobile applications. Web applications can also be created using this cloud platform. Using this platform, cloud APIs can be built and published securely at scale and push notifications can be sent to millions of devices using the Notification Hubs. The Azure Mobile engagement can be used to monetize, retain and maximise app usage. [36]

Data and Storage

The Microsoft Azure cloud platform can be used for storing blobs, tables, queues, files and disks and these are highly available and scalable. Hybrid clouds can also be created using StorSimple and these are mainly for enterprises, which reduces costs and improve data security. Relational SQL databases can also be created and Azure's DocumentDB can be used to manage NoSQL document databases. Windows Azure's Redis Cache can be used to build applications that are fast, scalable, have low latency and high throughput. Windows Azure's data storage also provides search and warehouse as a service. The SQL server stretch database can be used to dynamically stretch SQL server databases to Azure. [36]

Networking

Virtual private networks that can be optionally connected to on premise datacentres and private network fibre connections to azure can be built using this cloud platform. The traffic manager can be used to route incoming traffic for high performance and availability and highly available network performances can be delivered using Azure's Load balancer. Domain Name Servers (DNS) can be hosted on Azure and the VPN gateway can be used to establish cross premises connectivity that is secure. Azure's Application gateway can be used to build scalable websites that have delivery control and load balancing. [36]

Other Services

Microsoft's Azure cloud platform can be used for data analytics, deploying Internet of things (IoT) and supporting media services and Content Delivery Networks (CDN). It can also be used for hybrid integration where the enterprise and cloud can be seamlessly integrated and also cloud recovery, protection and back up. The service bus is also available to connect across private and public cloud environments. This cloud platform has the capability to be used for identity and access management, billing, security and provides some developer services. [36]

Redundancy

Microsoft Azure has data centres in Asia, North and South America, Europe and Australia. When deploying Azure's cloud services, a user can choose where they would want to store their data. To further ensure the availability of a service, a user can store their data in different

data centres, which ensures that if anything happens to one of the data centres, the other one will be able to still provide the service to the user. [36] In this research no information is deployed onto another redundant data centre because to do this additional costs would have been added.

In this research, a SQL database is created on Microsoft Azure and is deployed on a data centre in Northern Europe as shown in figure 3.2. The detailed description on how this database is set up is shown in appendix B.

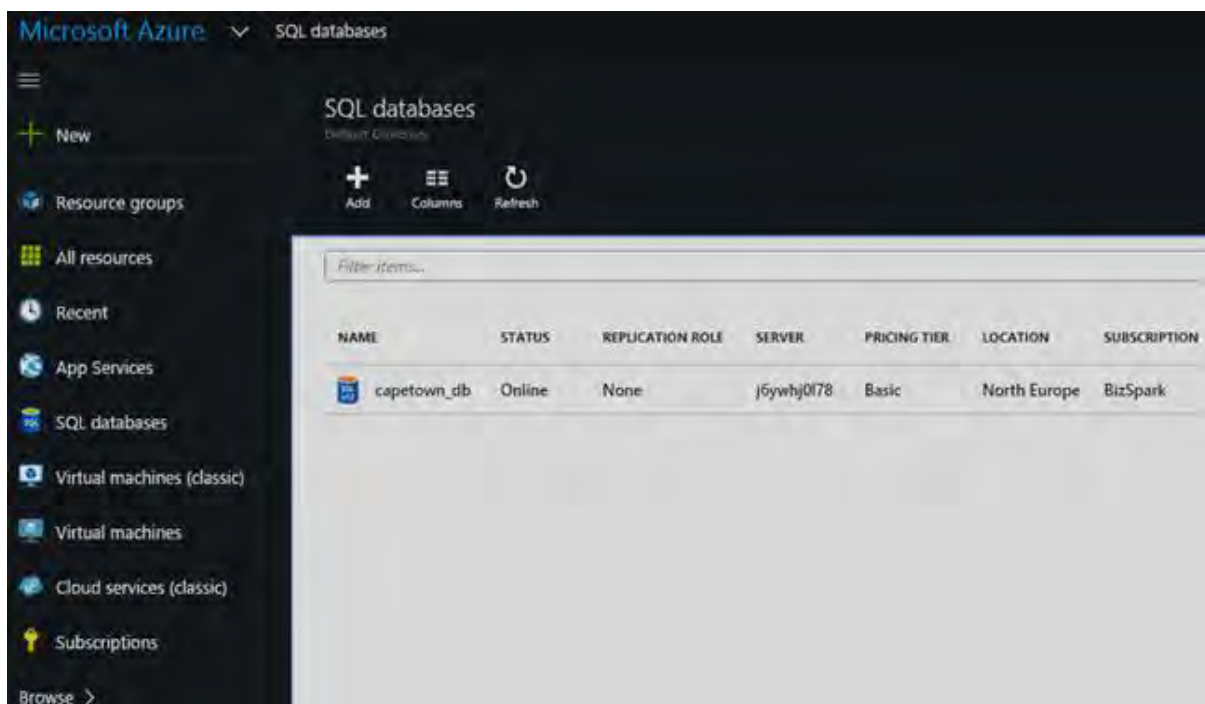


Figure 3.2 Deployed Azure SQL database.

Microsoft Azure, is also used to create an Azure Mobile Service and this is deployed onto a server in Northern Europe as shown in Figure 3.3. This enabled the backend of the Smart Shopping mobile application to be hosted on the cloud. A detailed description of the setup procedures is shown in appendix B.

In this research, Azure Mobile Service is used to implement Application Programming Interface (API) calls. API calls are a set of instructions that send HTTP request methods to a database and these can either get, post, patch, put or delete data from there [37]. In this research

API calls are used to get data from the Azure SQL database. The data is then pushed to the Smart Shopping mobile application.

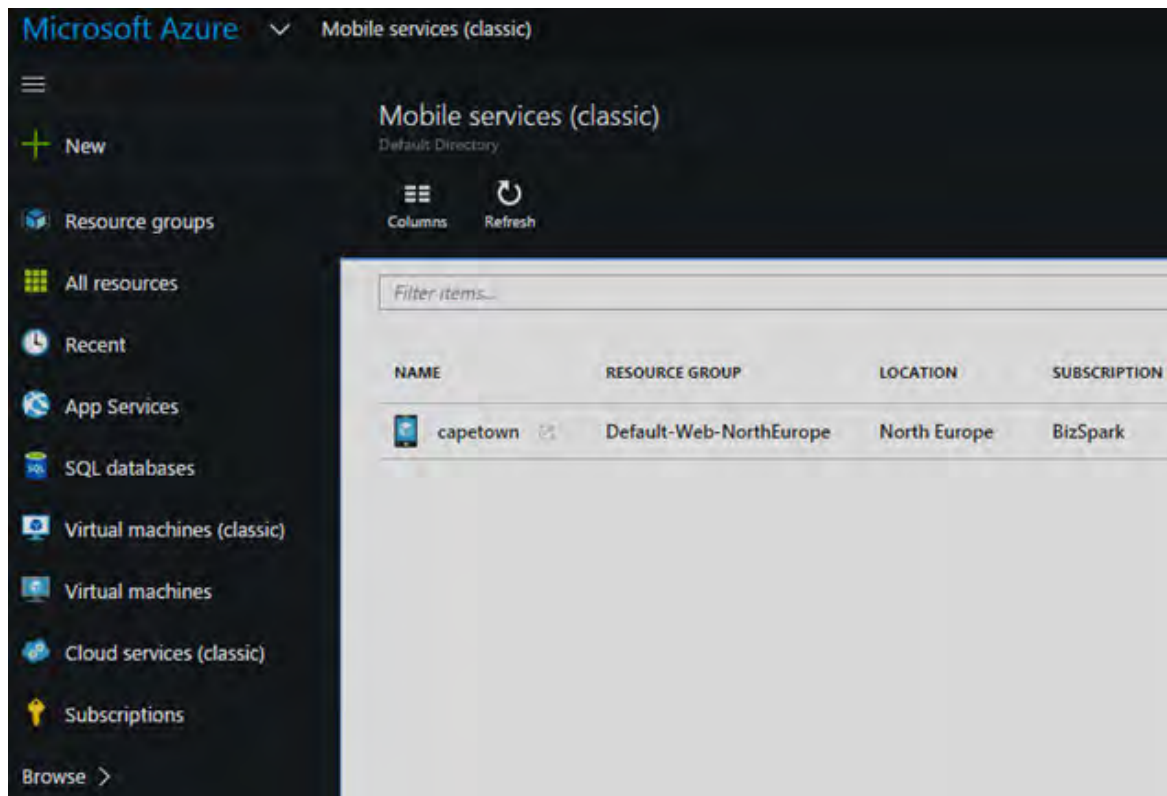


Figure 3.3 Deployed Azure mobile service.

3.1.3 Navicat

A database management tool called Navicat is then used to manage and manipulate the data that was stored on the SQL Azure database. Navicat first had to connect to the Azure SQL database before any management of the data could be done. Connection to the database is made using the Secure Socket Shell (SSH) and Hypertext Transfer Protocol (HTTP) connections which prevent the data's security from being compromised when the servers are being accessed [37]. Navicat also allows for data to be imported or exported in many formats such as MS Access, MS Excel, XML, HTML, TXT and CVS [37].

This tool can be used to connect to different types of databases such as MySQL, MariaDB, SQL Server, SQL Azure, Oracle, PostgreSQL and SQLite and it supports many of their functionality such as tables, stored procedures, events, triggers, functions and views.

Data in the databases tables can be edited, deleted and created using this tool. Its data modelling tool can be used to make conceptual, logical and physical data models. [37]



Figure 3.4 Navicat graphic user interface.

Appendix B shows how Navicat is connected to the Azure SQL database. Figure 3.4 shows the Navicat Graphic User Interface (GUI). It also shows the connection that Navicat made with the Azure SQL database called capetown_db that had been deployed on the Microsoft Azure cloud platform in this research. The figure also shows the tables that were made and stored on the database.

3.1.4 Postman

Before the API calls are implemented on the Azure Mobile Service, Postman is used to test them. This was done to make sure that the correct information is obtained from the database when the API calls are made on the Azure Mobile Service.

Postman is a software that is used to test and document APIs. HTTP requests can be created and sent using Postman and these can be stored to be sent again later. API calls that are related can be organised into Postman collections and this makes them much more easy to test and makes the workflow much more integrated. Figure 3.5 shows the GUI for postman software. As shown in the figure by the drop down menu, Postman can be used to test API calls that can get, post, put, patch, delete, copy data from a database. [37] Four parts of the HTTP requests are the URL, header, body and method. Below is a brief description of the header and the URL. [37]

URL

The URL has an input field where the URL that is obtained from Azure Mobile Services together with the API address of the call to be made is input. It also has a Key and value that must be entered. These are shown in Figure 3.6.

Header

The header also has a place where a key and a value has to be entered [37].

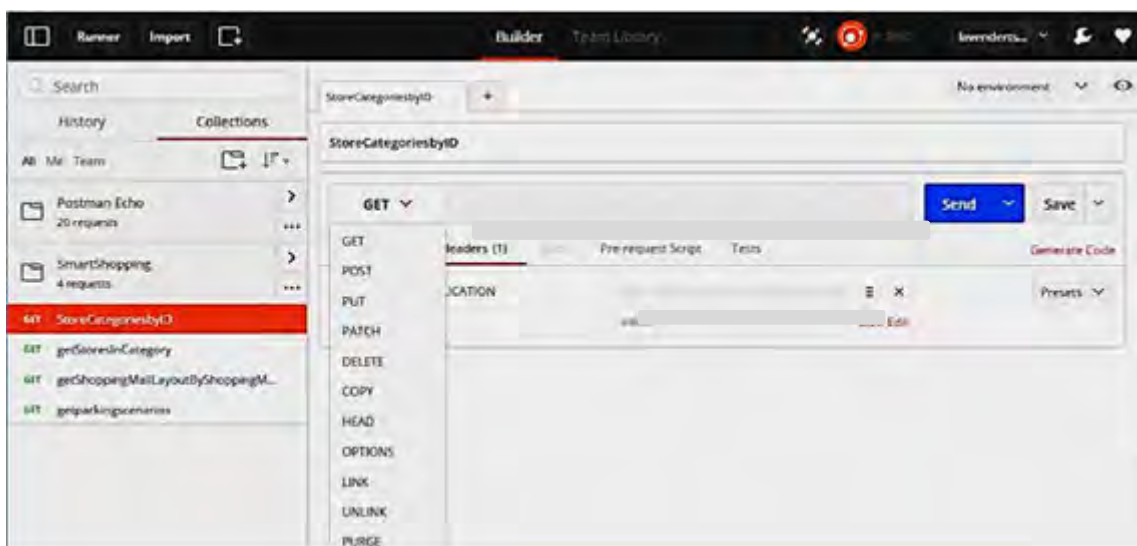


Figure 3.5 Postman Graphic User Interface (GUI).



Figure 3.6 Postman URL header and key-value pair [37].

As in Figure 3.7, the postman interface is divided into two parts which are the request builder where any kind of requests can be made and the side bar which shows the history of the API

calls that would have been made and saved [37]. Figure 3.5 side bar shows the API calls that were made and tested in this research.

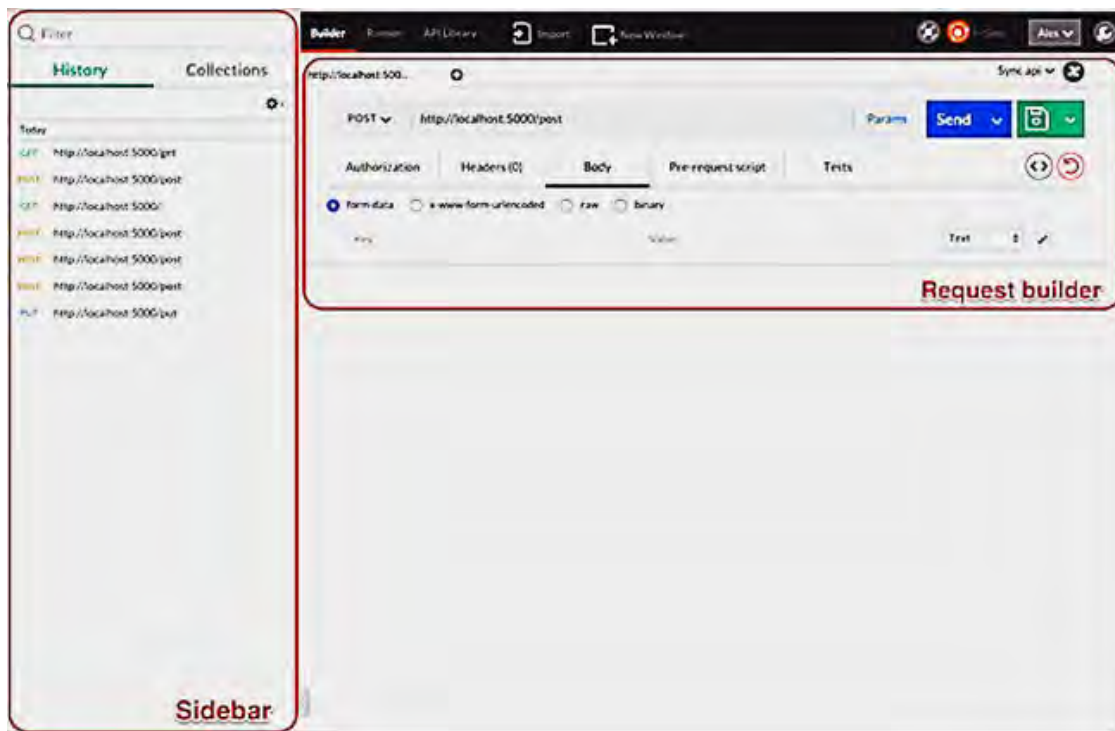


Figure 3.7 Postman request builder and side bar [37].

The explanation of how Postman was used is presented in the implementation chapter.

3.2 System Design Flow

Mobile services are developed as a response to new technologies or changes in an organization that demand new service systems to be implemented [11]. In this research the type of mobile service system that is developed is a mobile application. When designing mobile service systems, three major aspects have to be taken into account and these are:

- Service concept;
- Organisational network;
- Technical architecture [11].

This research only focuses on the service concept and the technical architecture.

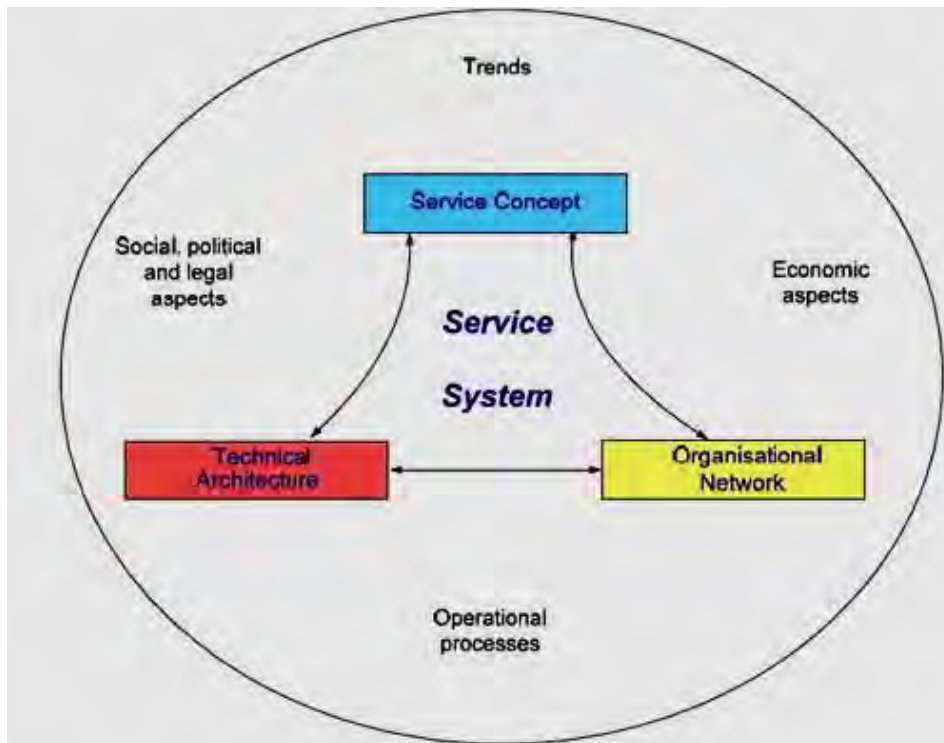


Figure 3.8 The three aspects of mobile service system design [11].

Figure 3.8 shows these three aspects and their interactions. The service concept and the technical architecture are affected by social, political and legal aspects, the service concept and the organisational network are affected by economic aspects and finally the technical architecture and the organisational network are affected by operational processes [11].

The organisational network provides the coordination between resources provided by organisations involved in enabling the delivery of a service and also the revenue models between them [11].

3.2.1 Service Concept

When designing a mobile application/service system, the service concept has to be considered first. The service concept describes how the services provides value to the users. Figure 3.9 shows the process to be followed when determining the service concept during the first stage of designing a composed service. The following segments described the process in more detail.

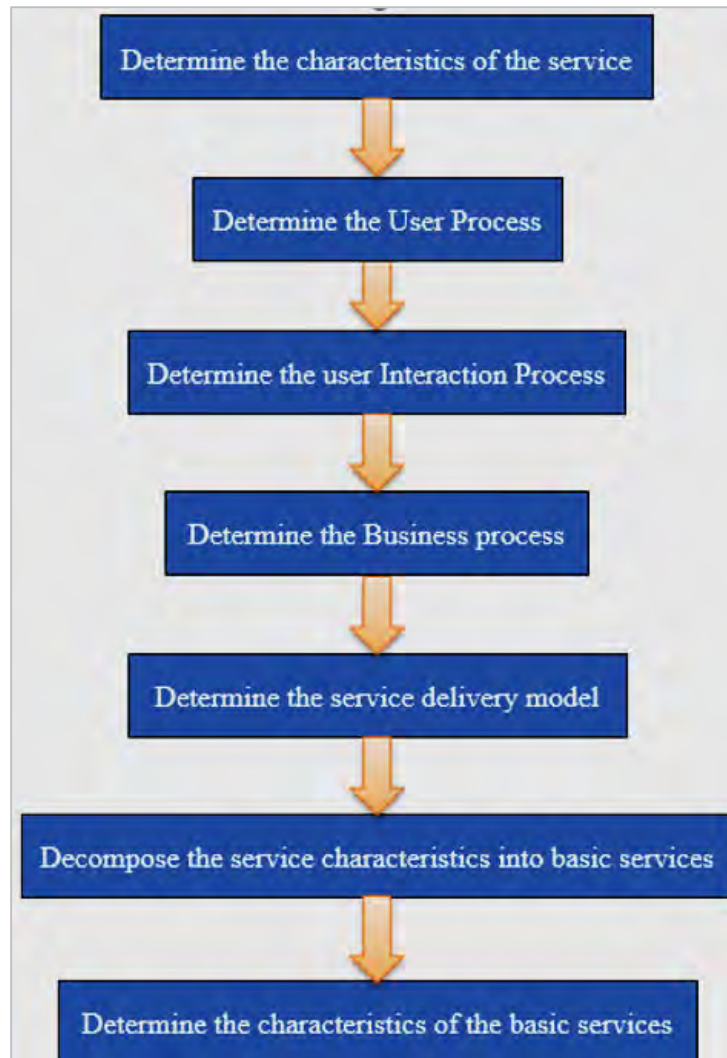


Figure 3.9 Service Concept design process.

3.2.1.1 Characteristics of the Service

The service characteristics describes the functionality that the service must have. The characteristics of the developed Smart Shopping service are as follows:

- Name of shopping mall and operating hours;
- Location of shopping mall;
- Direction of shopping mall from current location;
- Store categories;
- User may choose store/restaurant name;
- Information about the stores in the shopping mall;
- Number of free/available parking bays;

- Available parking bay numbers;
- Real time update of occupation of parking bays;
- Map of shopping mall indicating the location of the shops.

3.2.1.2 User Process

This process refers to the user needs and how they will use the service. When designing a service, these factors affecting the users must be analysed and designed:

- Roles of the user with respect on what the user needs to do in order to use the service;
- Context in which the service is used;
- Usage behaviour of the user that needs to be supported by the service;
- Dependence of the service on the usage of the service by other users;
- Sometimes the user's activities need to be redesigned for them to effectively use the service. This normally occurs when the user's way of doing things cannot be supported by the service [11].

For the Smart Shopping service, the user first needs to download and install the service to their Windows Mobile Phone before they can start to use it. The user's device needs to connect to the internet for them to have the best user experience.

3.2.1.3 Interaction Process

This is the method by which the service is delivered to the user. It is affected by the technological capabilities of the device [11].

For the Smart shopping service, the user interacts with the service via a mobile phone device.

3.2.1.4 Business Process

This process refers to what the service provider needs to do in order to deliver the service to the users, and be able to satisfy their requests and needs. The service provider needs to gather the right data which should be presented in a user interface that the user would enjoy viewing when using the service. The service designer must guarantee continued functionality of the

service regardless of the state of the other components or dependences of the service, (e.g. external links becoming unavailable). User preferences need to be stored, enabling personalised service delivery. [11]

The following business processes are considered for the Smart Shopping service:

- To ensure accuracy of the data shown on the service, the data is obtained from the Canal Walk Shopping Mall website;
- The data is stored on a reliable Microsoft Azure SQL database.

3.2.1.5 Service Delivery Models

There are three models in which services are delivered to users as follows:

- **Pull category:** Enables the users to pull information at any time for example services that provide information about the weather condition of a town;
- **Push category:** Services are pushed to the user depending on whether they fit a certain criterion or user context that will make them potential customers (e.g. age or location of the user);
- **In between categories:** The user first has to subscribe to a service and the service provider can then push information to the user depending on the user's characteristics [11].

The Smart Shopping service uses the pull service model.

3.2.1.6 Decomposed Services Characteristics

The characteristics of the Smart Shopping service can be decomposed into the following services and their characteristics:

a. Mall Information Service

- Name;
- Map of the mall (mall layout);
- Operating hours;
- Contact details.

b. Store Category Information Service

- Store category name, (e.g. the store category called Clothing will have all the names of the shops that sell clothes).

c. Store Information Service

- Name;
- Store number;
- Telephone number;
- Store image;
- Store category name;
- Level on which the store is located (This can either be Lower level – LL or Upper level-UL).

d. Parking Services

- Map of the parking lot (Designed by in this research, and to be used as a prototype due to unavailability of the real maps of the mall);
- Available parking bay numbers;
- Unavailable parking bay numbers.

e. Location Services

- Current location of the user;
- Shopping mall location.

f. Navigation to Canal Walk Shopping Mall.

3.2.1.7 Factors Affecting the Use and Acceptance of New Services

Some of the factors affecting the use and acceptance of new services are as follows:

- Perceived usefulness;
- Social aspects;
- Behavioural beliefs;
- Perceived ease of use;

- Reliability;
- Cost;
- Responsiveness;
- Trust [11].

Figure 3.10 shows the factors that affect service acceptance.

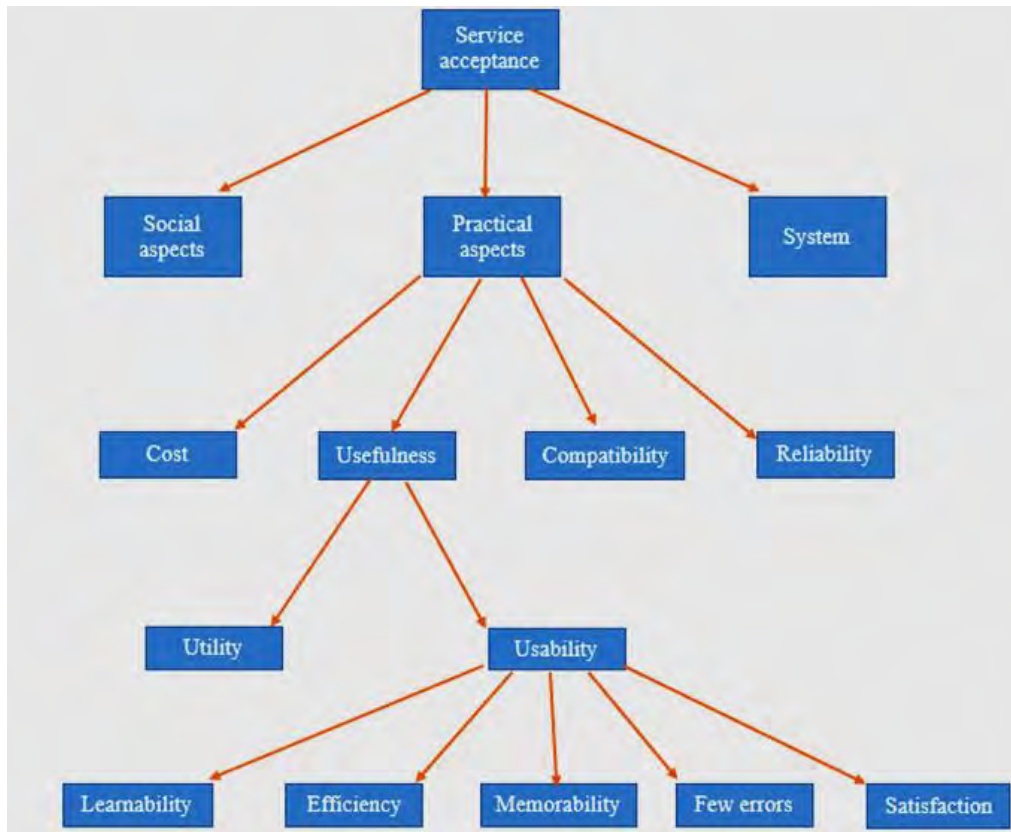


Figure 3.10 Factors affecting service acceptance [11].

The main reason why a person may start to use a new service is the perceived usefulness and capabilities of the service. This perception is affected by how the user interface of the service looks and the way the user will interact with the service. The user interface must be designed to make the user perceive the service as easy to use and useful. A user may also decide to use a service depending on the frequency they will use it, the cost of using the service and the skills needed to learn how to use the service. [11]

Acceptance of a service is also affected by *social aspects* which is how the society views the service and how popular the service is. A service can also be delivered to a user as part of a

technology (for example, all Windows smart phone users must have a Microsoft email account and all Android smart phone users must have a Google email account in order for them to maximise the capabilities of their mobile phones). *Practical aspects* also affect the acceptance of a service and this is affected by the cost, compatibility, reliability and usefulness of the service. The usefulness of a service is a measure of how the service meets the user needs. It is determined by the *utility of the service* which is a measure of the required functionality and also the usability. *Usability* is determined by the learnability of the service. It is also a measure of how efficient and easy to remember the use of the service is. The user satisfaction and errors present in the service also affects its usability. [11]

3.2.2 Technological Architecture

The three-layer architecture is used when designing and developing a mobile application [11], [38]. The architecture is described in detail below and how it is used in this project is explained.

Three-Layer Architecture

Mobile applications have an architecture that is divided into three layers called the Presentation Layer, the Data Layer and the Business Logic Layer. Layering separates the different functionality of the application into different areas of concern. This allows a team of developers to work on the same project jointly on different layers which may need different programming languages. The layers are loosely coupled. This is an advantage that application programmers with certain programming skills will only work in their area of expertise. [11], [38]

Loose coupling of the layers allows for the evolution of the applications as new user needs and opportunities arise due to that it is possible for each layer to be modified without affecting the other layers. This also allows for separate testing of the different layers of the application. [11], [38]

Figure 3.11 shows the mobile application architecture [38].

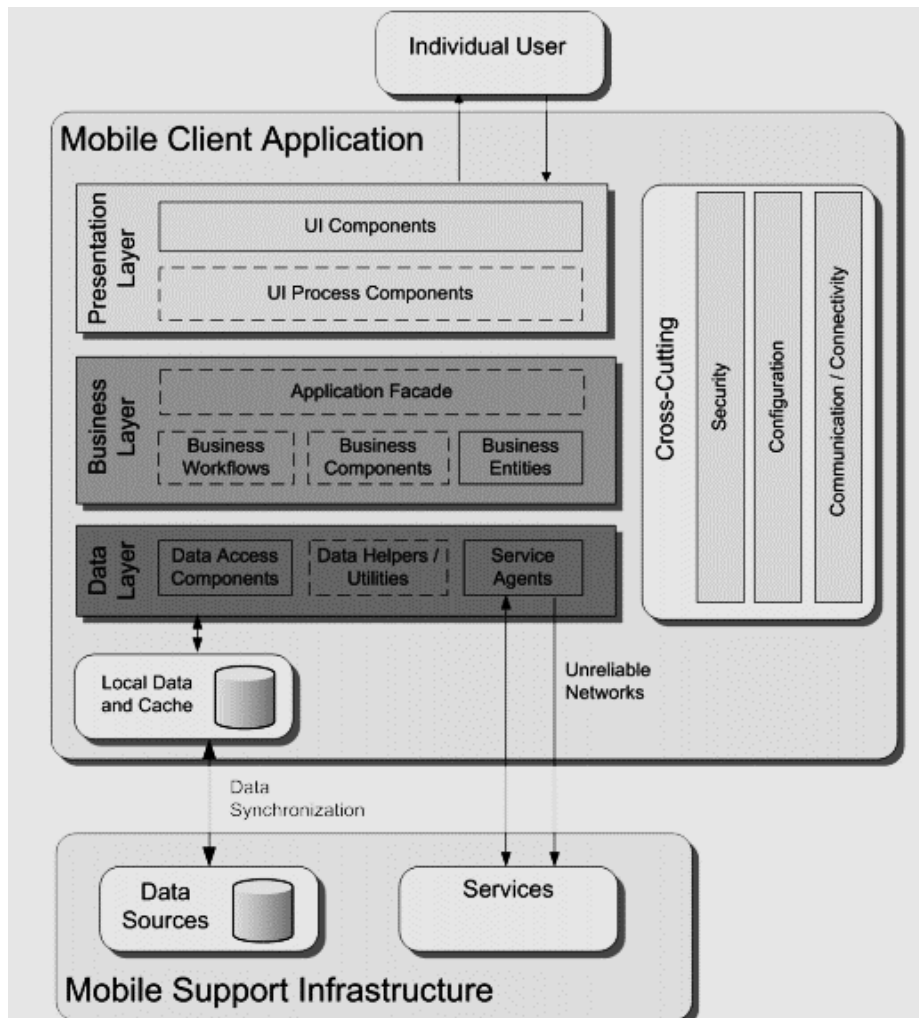


Figure 3.11 Mobile application architecture [38].

3.2.2.1 Presentation Layer

The presentation layer is also known as the *View* and has components that implement and display the User Interface (UI) and also manages the user interactions with a service. It has controls for user input and display and also components that organise user interactions. The programming work provides the graphic user interface (GUI), application specific entry forms and interactive windows. [11], [38]

Presentation Layer Components

The components of the presentation layer are as follows and are shown in Figure 3.12.

UI Components

The UI components are a means by which a user can interact with the mobile application when using the mobile phone and this will be the mobile phone display screen. They render and format data for users and they also acquire and validate data input by the user. In this research, the UI components are referred to as the User interface (UI). [11], [38]

User Process Components

They synchronise and orchestrate user interactions [11]. In this research, the UI process components are referred to as the UI Logic or the Code behind.

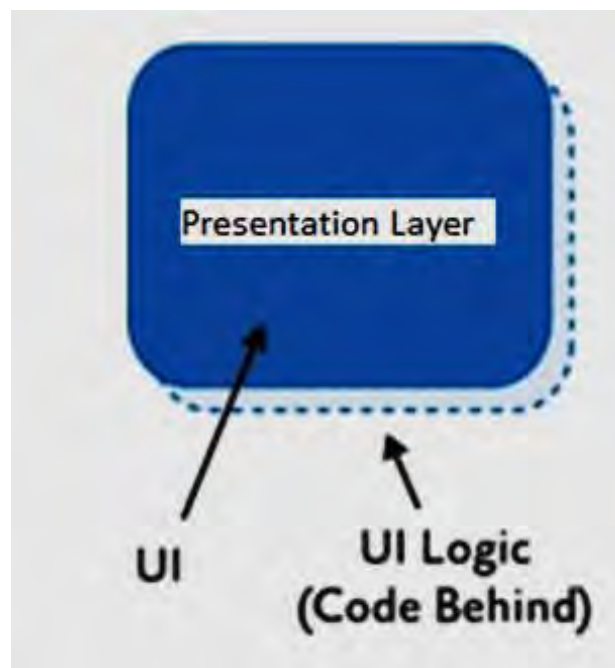


Figure 3.12 The presentation layer.

Figure 3.12 shows the presentation layer components which are the UI and the UI logic.

Presentation Layer design process

The second step that is done in this mobile application design, after determining the Service Concept is the design of the presentation layer. Figure 3.13 shows the process that need to be followed when designing the presentation layer.

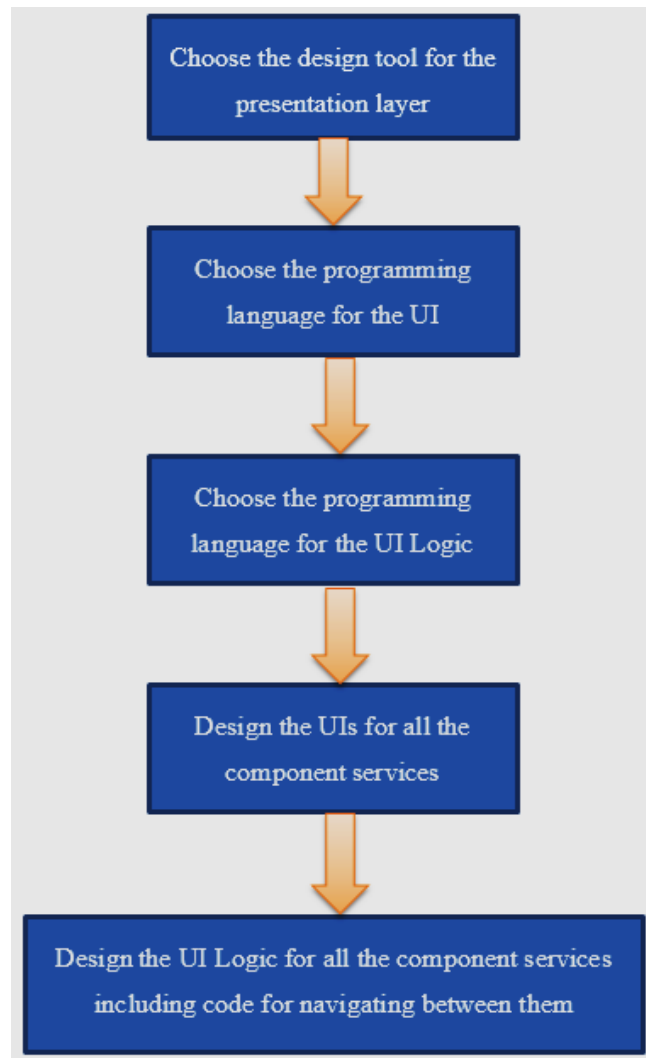


Figure 3.13 Presentation layer design process.

In this research the tool that is used to design the presentation layer is Visual Studio 2015 Community. XAML programming language is used to design the UI and C# programming language is used to design the UI Logic. The detailed discussion on how the UI Logics and the UIs for all the basic services to be composed are designed and how the navigation between them is designed for the Smart Shopping service is presented in the next chapter.

3.2.2.2 Data Layer

The data layer manages how data is stored, accessed and manipulated. The components of the data layer are as follows:

Data Access Logic Components

These abstract the necessary logic required to access the underlying data that will be stored in data bases. This contains the properties of the data to be used in the application, obtained from the data base. [38]

Data Helpers/ Utilities

These are functions and utilities that are used to maximise data access performance, data manipulation and transformation. [38]

Service Agents

Used to convert between formats of services when for example a business component in an application uses a functionality that is exposed by another external service, service agents are used for communication between the two services. [38]

Data Layer Design Process

The third step that is conducted in this mobile application design is the data layer design process which is shown in Figure 3.14.

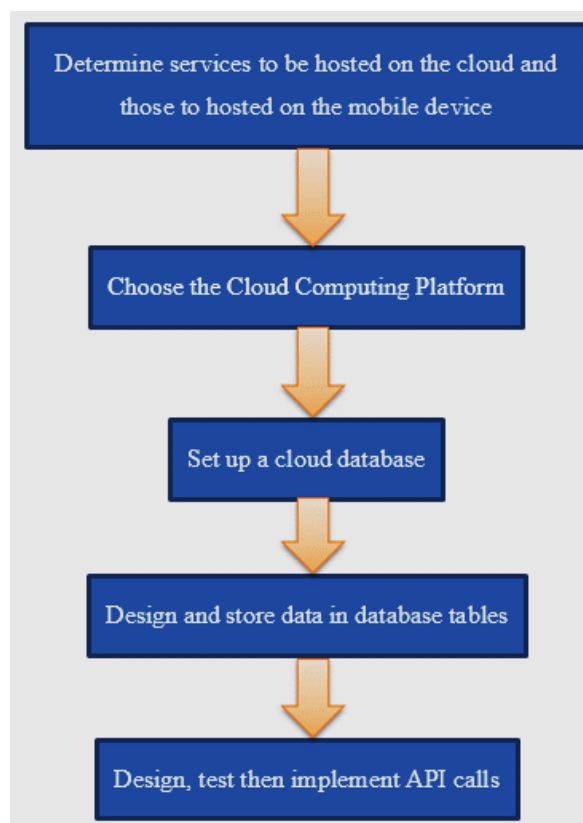


Figure 3.14 Data layer design process.

Due to the limitations of mobile devices, it is of high importance to determine which of the data computations, storage and processing is done on the mobile device or on the cloud. For the Smart Shopping service, the data about Canal Walk Shopping mall’s contact details and trading hours are stored on the mobile device as these required no computations and the data remains constant. The cloud computing platform that is used is Microsoft Azure cloud platform. An Azure SQL database is set up on the cloud platform. Data tables for the shopping mall layout, store categories, stores, trading hours and parking scenarios are made in the Azure SQL database, and data is stored in them. Figure 3.15 shows the table definition for the shopping mall layout table.

| Name | Type |
|----------------------|--------------|
| ShoppingMallLayoutID | int |
| ShoppingMallID | int |
| LayoutLevel | varchar(MAX) |
| MallLayout | varchar(MAX) |

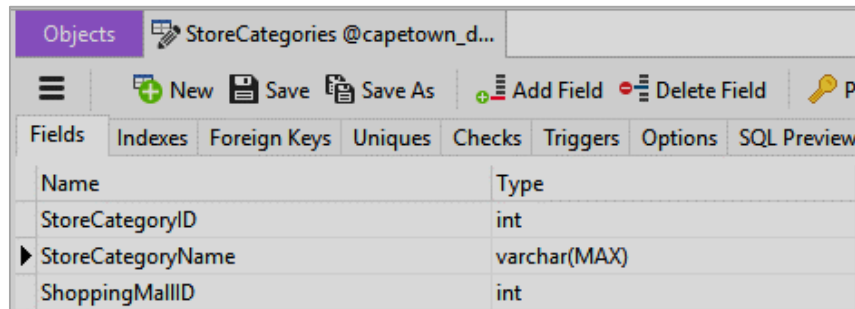
Figure 3.15 Shopping mall layout table definition.

| Name | Type |
|--------------|------|
| Scenario | int |
| ParkingBay1 | int |
| ParkingBay2 | int |
| ParkingBay3 | int |
| ParkingBay4 | int |
| ParkingBay5 | int |
| ParkingBay6 | int |
| ParkingBay7 | int |
| ParkingBay8 | int |
| ParkingBay9 | int |
| ParkingBay10 | int |
| ParkingBay11 | int |
| ParkingBay12 | int |

Figure 3.16 Parking scenarios table definition

Figure 3.16 shows part of the table definition for the parking scenarios table. The table has a column for the scenarios and 38 other columns from parking bay 1 to parking bay 38.

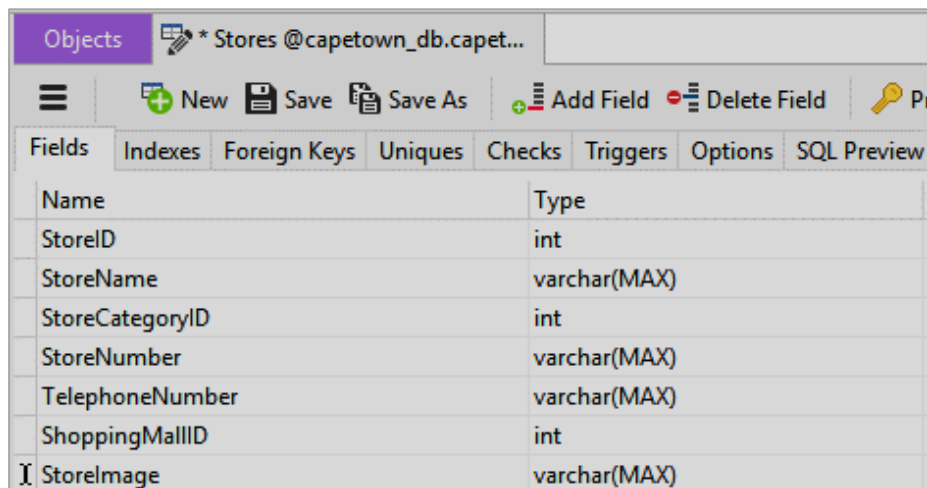
Figure 3.17 shows the table definitions for the store categories table.



| Name | Type |
|-------------------|--------------|
| StoreCategoryID | int |
| StoreCategoryName | varchar(MAX) |
| ShoppingMallID | int |

Figure 3.17 Store categories table definition

Figure 3.18 shows the table definitions for the stores table.



| Name | Type |
|-----------------|--------------|
| StoreID | int |
| StoreName | varchar(MAX) |
| StoreCategoryID | int |
| StoreNumber | varchar(MAX) |
| TelephoneNumber | varchar(MAX) |
| ShoppingMallID | int |
| StoreImage | varchar(MAX) |

Figure 3.18 Stores table definition

The next step in the data layer design process is the design, implementation and testing of the Application Programming Interface (API) calls. A software called Postman is used to test the API calls after they were implemented on Azure Mobile Services. The details on how this is done is given in the next chapter. The next chapter also describes how the data is accessed and how the Business Layer was implemented is presented.

The next chapter describes how the Smart Shopping service is implemented.

4 Design Implementation

This chapter describes how the Smart Shopping service is implemented. Firstly, the implementation of the presentation layer is presented, followed by that of the data layer. Finally, the business layer implementation is discussed. Figure 4.1 below shows the summary of how the Smart Shopping service system is designed and implemented using the tools.

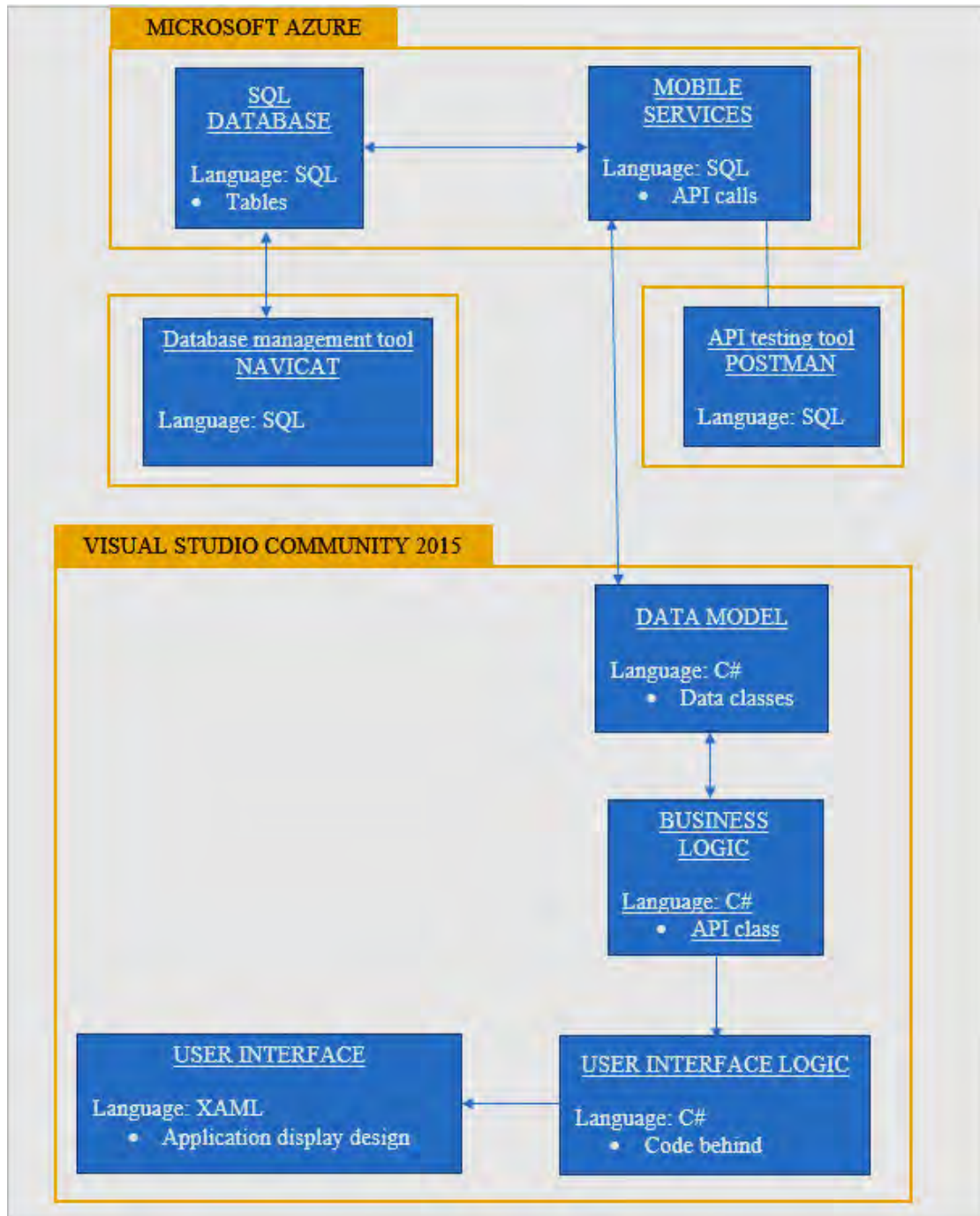


Figure 4.1 Design flow and implementation of the Smart Shopping Service System

Visual Studio Community 2015 is used to design and develop the presentation layer which is made up of the user interface and the user interface logic. This tool is also used to develop the business logic layer and the data model. Microsoft Azure is used to design a SQL database on which data was saved, and was stored in the cloud. Azure mobile service is used for sending API calls with requests of the data needed by the app from the database. This database is then pushed to the data model. The business logic class is then used to connect the presentation layer to the data model and data source. Data binding is used to connect the user interface to the data. A detailed description on how the design process is done is presented in the following sections.

4.1 Presentation Layer Implementation

As described in the previous chapter, the presentation layer is comprised of the User Interface and the User Interface Logic. The implementation of these parts are described in the following two sections.

4.1.1 User Interface (UI)

In this research the User interface is designed using Visual Studio. This tool has many templates that can be used to design the Windows Phone application user interface. Templates are there to ensure that applications that are designed are consistent to the Microsoft design pattern. Some of the examples of these are:

- Blank App;
- Hub app;
- Pivot app.

Figure 4.2 shows the Visual Studio templates that can be used for Window Phone application development. The Hub app template in Visual Studio Community 2015, which is selected in Figure 4.2, is used to design the presentation layer in this research.

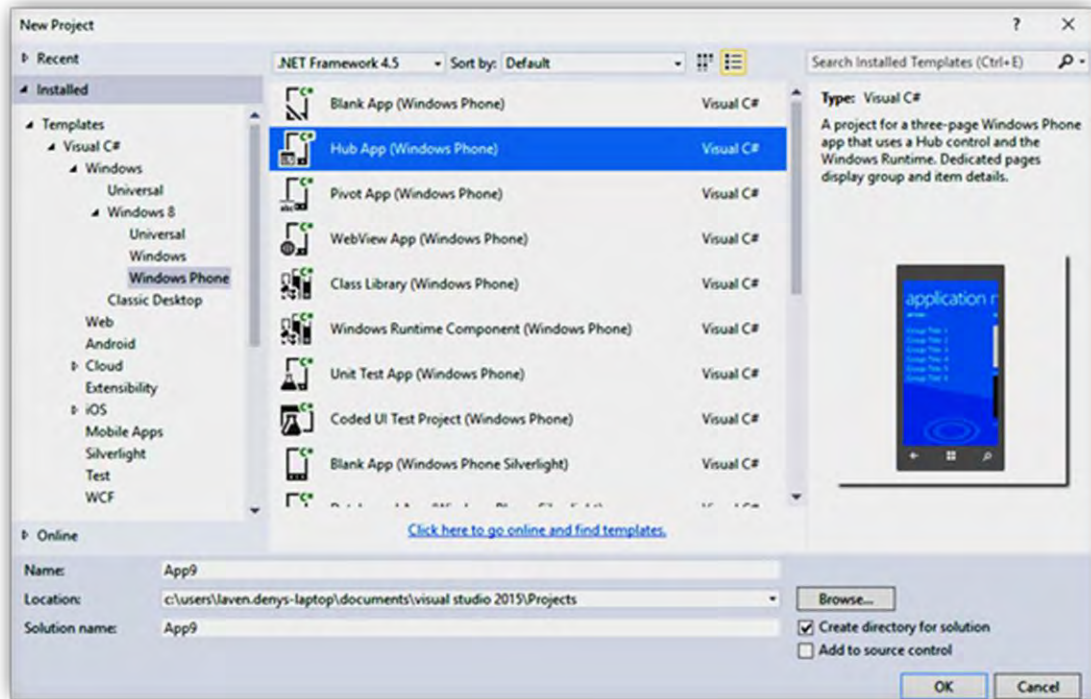


Figure 4.2 Selection of the Hub template in Visual Studio Community 2015.

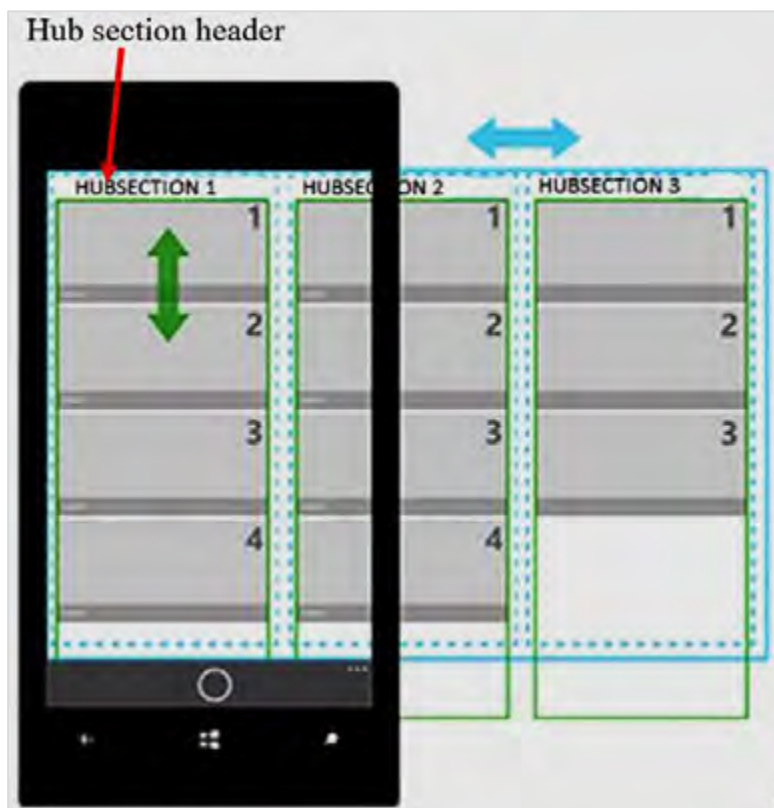


Figure 4.3 The Hub app template [39].

The Hub App Template

The Hub template consists of a collection of Hub sections and each section has its own Data Template and data source. It makes it easy to present the mobile application data in a meaningful and aesthetically pleasant way. The content that can be presented in the form of lists, text, videos, links and images, with each data type having a unique layout and interaction model. On mobile devices, one hub section is displayed on the phone screen at a given time and also a small part of the next hub section is displayed as shown in Figures 4.11, 4.14 and 4.33. This makes the user of the application aware that there is more information coming on the next section. The hub page also has a hub header where the application name can be written. This hub header stretches across all the hub sections as shown in Figures 4.9, 4.11 and 4.14. [39]

In this research, Hub page section one, which has content about the categories of the stores at Canal Walk Shopping Mall is the first page that is displayed on the mobile phone screen, just after the splash screen. As shown in Figure 4.3 by the blue arrow, navigation from one section to the next section of the hub template can be done by swiping horizontally. If there is more information on a hub section that cannot all be displayed on the mobile phone screen, scrolling vertically the sections can be enabled in order to view all the information on the section as shown by the green arrow in Figure 4.3. Each item on the section can contain more detailed information about it and some of the ways of getting this information is by clicking or tapping, depending on the application design. [40], [39]

In this research, all the services that are composed to build the Smart Shopping mobile application are designed on hub sections. The information that is presented on the user interface is obtained from the Canal Walk Shopping Mall website and then stored on the Microsoft Azure cloud. A photo of Canal Walk Shopping Mall is used as the background picture of the user interface.

4.1.1.1 User Interface Design and Implementation Process

The UI design process is as shown in Figure 4.4. A detailed explanation of how the whole process is done is presented in the following sections

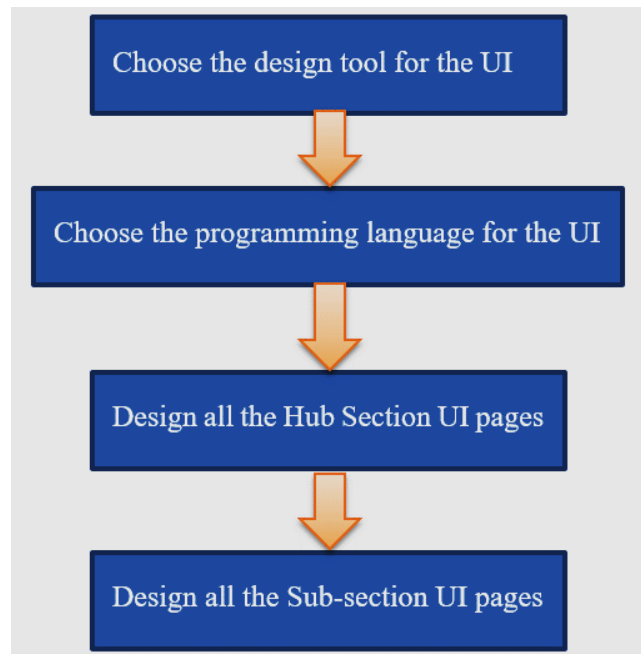


Figure 4.4 User Interface design process.

4.1.1.2 UI Design Tool

Visual Studio Community 2015 is used to design the UI.

4.1.1.3 User Interface (UI) Programming Language

The programming language that is used to design the UI (i.e. all the screens to be displayed on the mobile phone when using the application) is called the Extensible Application Markup Language (XAML).

4.1.1.4 Hub Section UI page Design

In this research six hub sections were implemented. These hub sections were as follows:

- Stores Categories (Hub section 1);
- Mall layout (Hub section 2);
- Trading hours (Hub section 3);
- Parking (Hub section 4);
- Directions (Hub section 5);
- Contact details hub section page (Hub section 6).

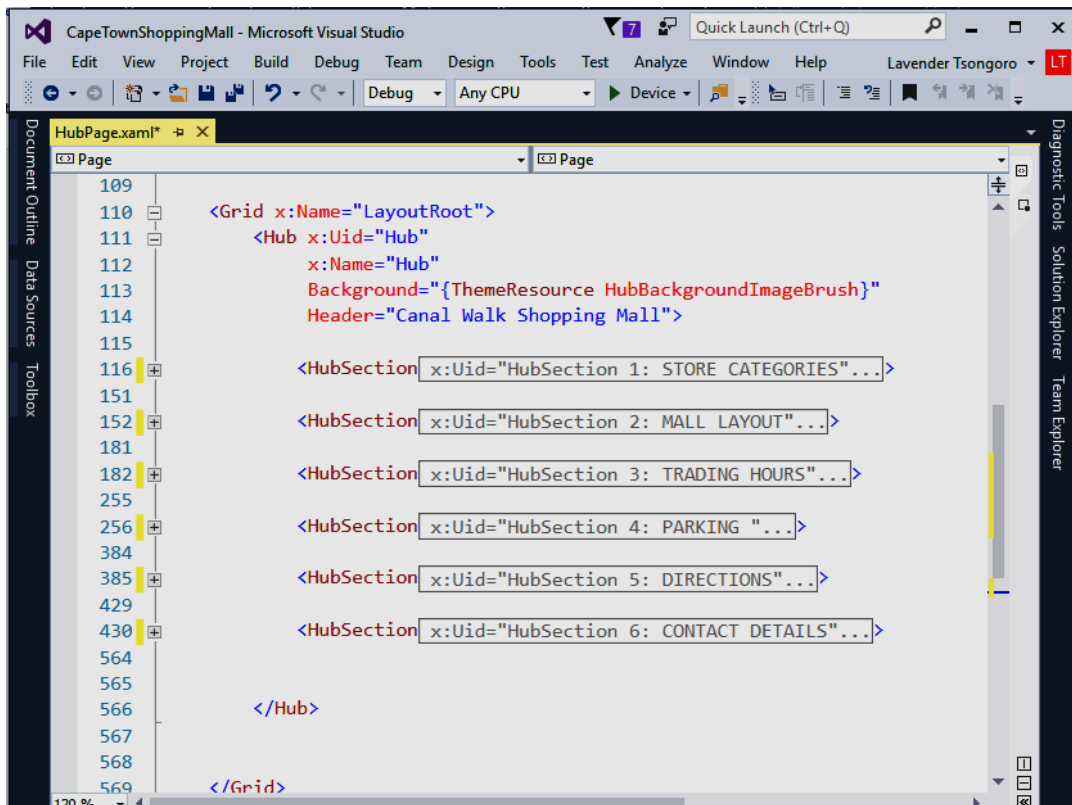


Figure 4.5 Hub page XAML.

Figure 4.5 shows the hub page XAML with the six hub sections. Each hub section shown above when expanded has some XAML code which designs all the features that will be displayed on that display screen of the phone for that hub section page. For example, hub section 1 and hub section 2 contains the XAML code as shown in the attached software document sections 1 and 2 respectively.

4.1.1.5 Hub Sub-section UI page design

SOA principles are used in the design process in which the hub page and all the other pages which are navigated to from the hub pages (i.e. sub-section pages) are loosely coupled. They have their own individual XAML code that are on separate XAML pages in Visual Studio as shown in Figure 4.6. An example of this is as follows:

When navigating from Hub section 1 (Stores categories), the display screen goes to the Sub-section display screen which shows the list of stores for the selected store category. The XAML code for this sub-section (stores) is shown in the attached software document section 3.

When navigating from Hub section 2 (Mall layout), the display screen goes to the Sub-section display screen which shows the maps of the shopping mall. The XAML code for this sub-section is as shown in the attached software document, section 4.

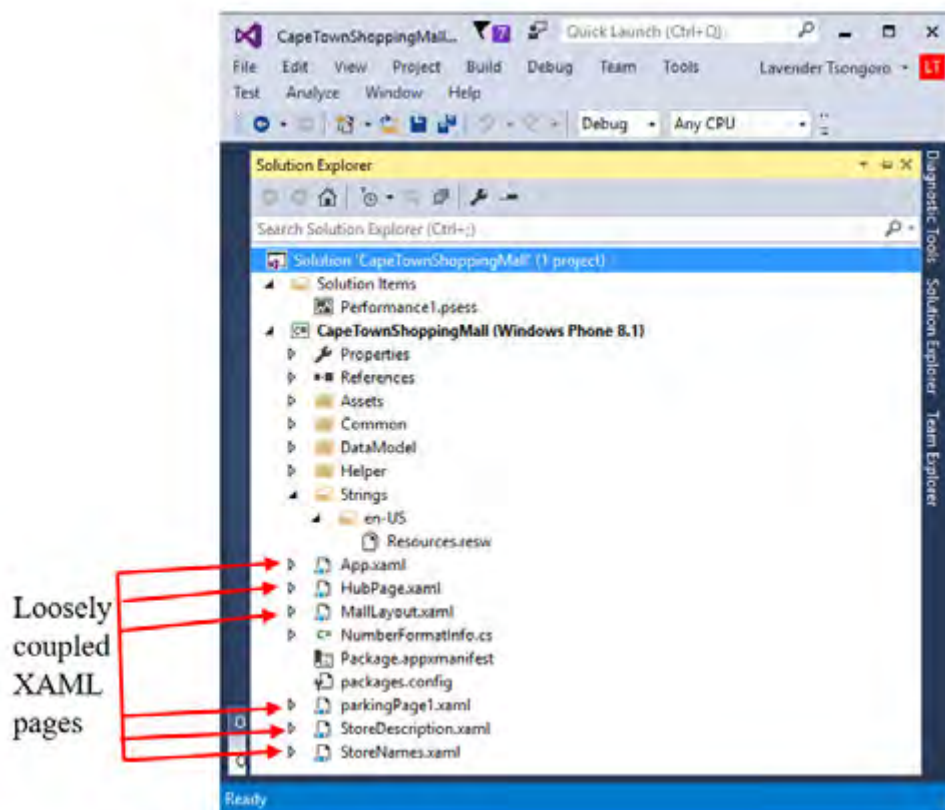


Figure 4.6 Loosely coupled XAML pages.

4.1.2 User Interface Logic Design

The UI Logic is also known as the code behind. The code behind has methods that are used to describe the behaviour of all the components that are presented on the display screens of the pages.

4.1.2.1 User Interface Logic Design and Implementation Process

Figure 4.7 shows the User Interface logic design process. It consists of four steps. The detailed explanation of how all these steps were done in the Smart Shopping service system design process is presented in the following sections.

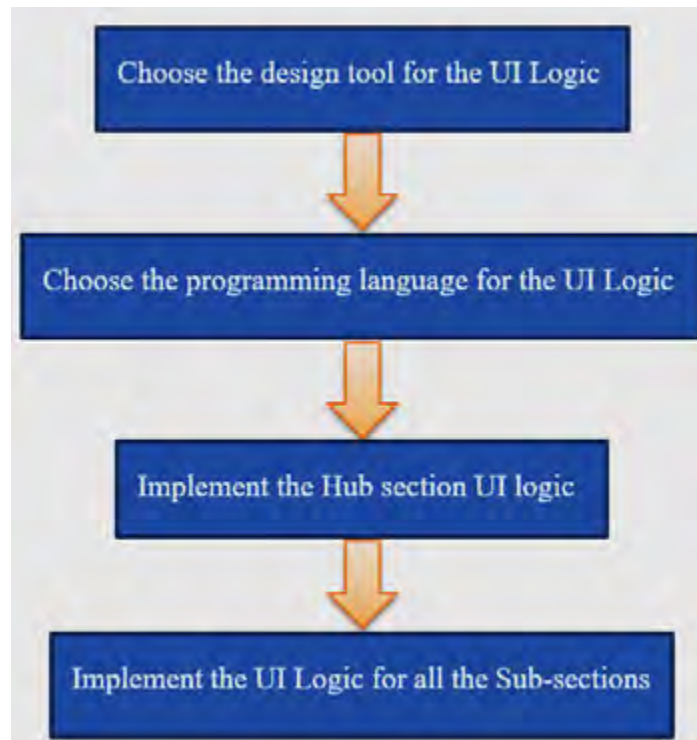


Figure 4.7 UI Logic design process.

4.1.2.2 UI Logic design tool

Visual studio Community 2015 is used to design the UI logic for the Smart Shopping service.

4.1.2.3 User Interface Logic Programming Language

The programming language that is used in the code behind is called C Sharp (C#). Using SOA principles, each one of the XAML pages for the Smart Shopping service has its own corresponding code behind which is loosely coupled from all the other code behinds and they are programmed using C#.

4.1.2.4 Hub Section UI Logic

The hub page has its own code behind. The C# code behind for the hub sections is shown in the attached software document, section 5.

4.1.2.5 Sub sections UI Logic

All the Sub-section pages which are navigated to from the hub page also have their own individual code behind for example, when navigating from the Hub section 1 (store categories), the screen goes to the frame with a list of store names for the selected store category. The C# code behind for this sub-section is shown in the attached software document section 6. When navigating from the sub-section with the list of store names in a category, after a category is selected, the screen goes to the frame with the information for the selected store. The C# code behind for this sub-section is shown in the attached software document, section 7.

In the code behind there are also methods that describe the navigation between pages. For example, as shown in Figure 4.8, the first method states that when the button on hub section 2 (mall layout page) is tapped, the frame will navigate from this section (Figure 4.11) to the mall layout pivot page shown in Figures 4.12 and 4.13. The second method states that when any of the store categories in hub section one is clicked, the frame will navigate from this page to the page with the list of stores for the selected categories as shown in Figure 4.9.

```
240 private void mallLayoutButton_Click(object sender, RoutedEventArgs e)
241     {
242         Frame.Navigate(typeof(mallLayoutPivot));
243     }
244
245
246 private void ShoppingCategoriesListView_ItemClick(object sender, ItemClickEventArgs e)
247     {
248         var item = ((StoreCategories)e.ClickedItem);
249         Frame.Navigate(typeof(StoreNames), item);
250     }
251 }
```

Figure 4.8 Hub page code behind navigation methods.

Other programmers preferred to place most of the C# code in the code behind into another class called the View Model [41].

4.1.3 The Smart Shopping Service UI layer

The design of the six hub sections for the Smart Shopping service is discussed in the following sections.

4.1.3.1 Hub Section One

In this section the **Store Category Information Service** is presented and this is displayed as a list of the store categories found in Canal Walk Shopping Mall as shown in Figure 4.9. This list can be scrolled up and down. Clicking on any of the store categories navigates to the Stores page. This displays a list of all the stores in the selected store category (e.g. clicking on the *Clothing* category goes to a page showing the list of stores in that category as shown in Figure 4.9) This list can be scrolled up and down.

Clicking on any of the stores on the list navigates to the page presenting the **Store Information Service** which displays the details of the selected store (e.g. Figure 4.10 shows the details of a selected clothing store called Ackermans). The details that will be shown are the store name, store telephone number, the picture of the store and the store number which also indicates the level on which the store can be found in the mall. Navigating back from the store details page will go back to the store list page and navigating back from there will go back to hub section one.

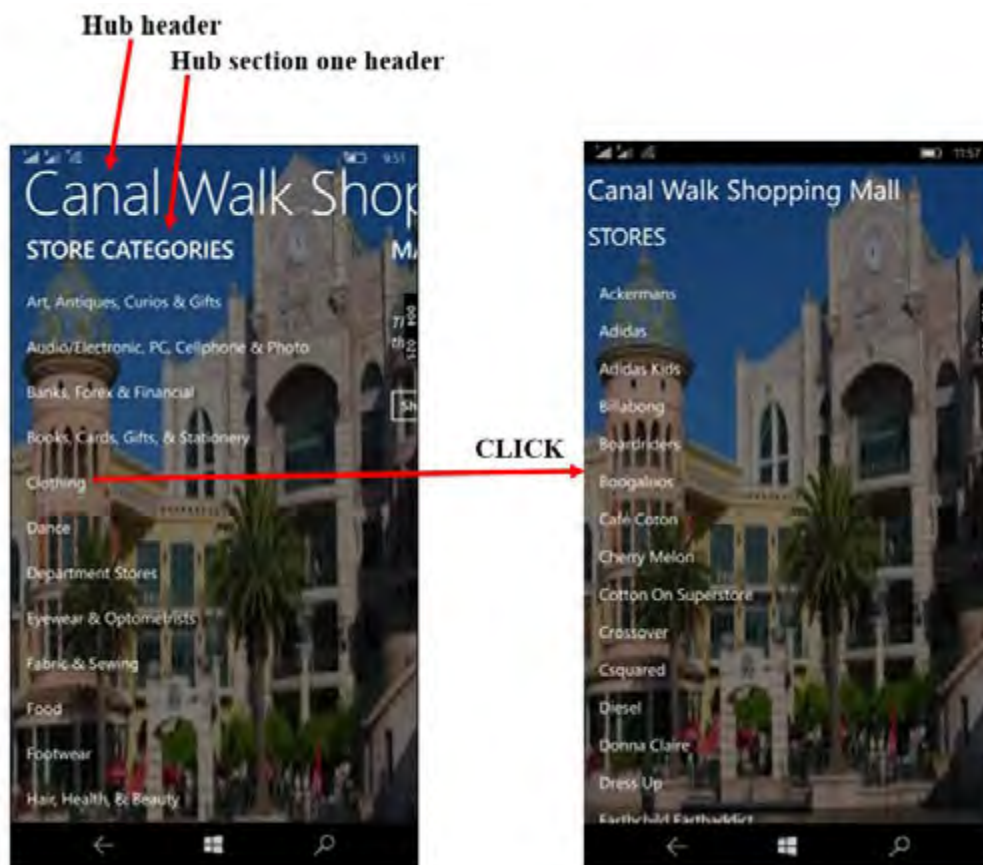


Figure 4.9 The Store Categories page and the Stores page.



Figure 4.10 Store details.

Hub section two header Hub header

Button

Preview of the next Hub section



Figure 4.11 Hub section 2.



Figure 4.12 Lower level map.



Figure 4.13 Upper level map

4.1.3.2 Hub Section Two

This section presents part of the **Mall Information Service** with the shopping mall layout information, which is given as shown in Figure 4.11. Tapping the button on this section navigates to a pivot page with the layout of Canal Walk Shopping Mall. As shown in Figure 4.12, the map of the lower level is shown. Swiping this page will then show the map of the upper level as shown in Figure 4.13. Scrolling on the maps moves them and enables other parts of the map to be displayed on the screen. Navigating back from the pivot page will go back to hub section two.

4.1.3.3 Hub Section Three

This section also presents part of the **Mall Information Service**. The trading hours of the stores at the shopping mall are displayed as shown in Figure 4.14, on the trading hours page.

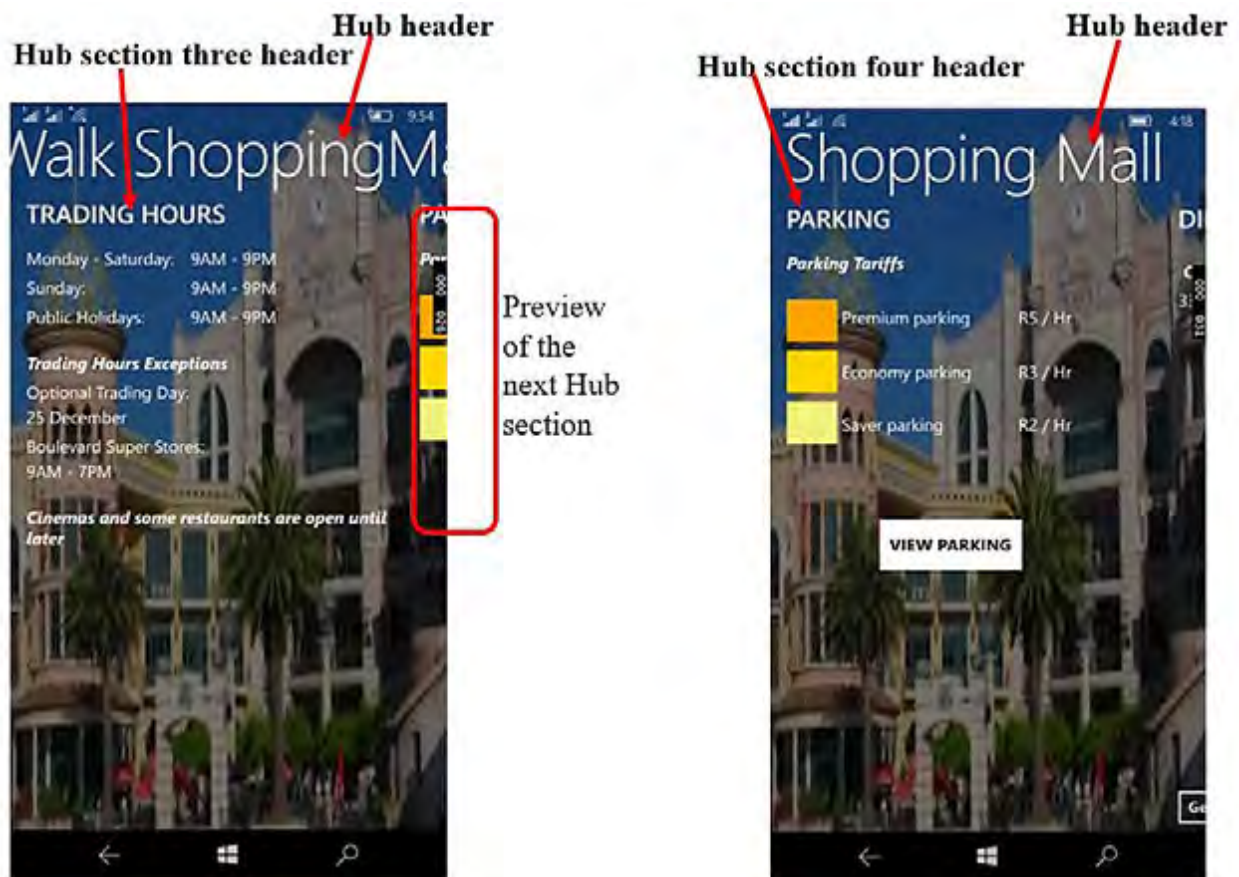


Figure 4.14 Trading Hours page (Hub section 3) and Parking Service page (Hub section 4).

4.1.3.4 Hub Section Four

On this hub section, the **Parking Service** is presented. As in Figure 4.14 parking page, the parking tariffs for premium, economy and saver parking are displayed on this page. Premium parking is pricier because these parking bays are located close to the exit of the parking lot followed by economy parking which has bays located in the middle section of the parking lot. Saver parking is the cheapest and has parking bays that are located much further away from the exit of the parking lot. Selecting a parking bay that is closer to the exit has these advantages:

- Fast exit from the parking lot;
- During peak hours when the parking lot is busy, the driver doesn't waste a lot of time trying to drive out the parking lot.

This hub section has a button which when clicked, will navigate to another page which shows the layout of the parking lot which was designed in this research and their respective states at 100 seconds sampling rate. Figures 4.15 to 4.24 show the states of the parking bays for the 10 scenarios that were designed. These screen shots only show part of the parking lot because all 38 of them can not fit on the mobile phone screen. For the user to view all the parking bays, the screen can be scrolled vertically up or down. The designs of the state of the ten scenarios are given in more detail in Appendix A.



Figure 4.15 Scenario 1.

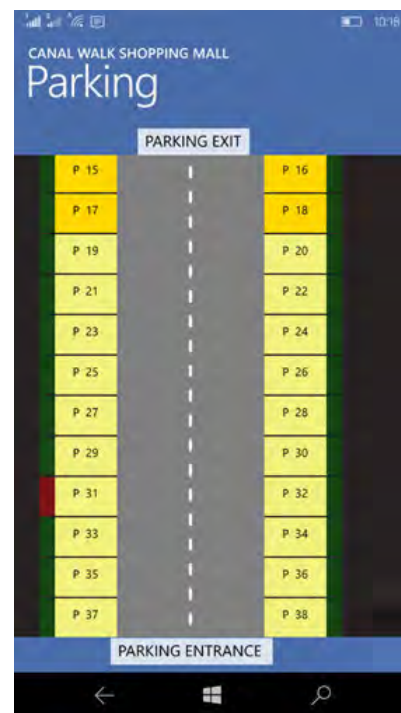


Figure 4.16 Scenario 2.

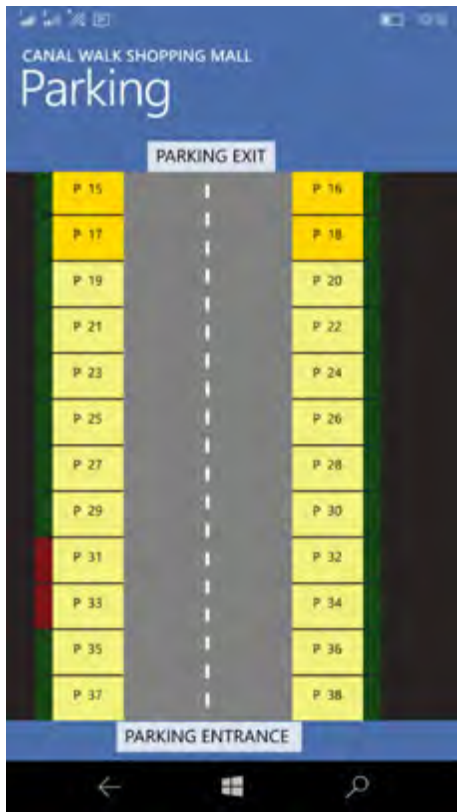


Figure 4.17 Scenario 3.



Figure 4.18 Scenario 4.

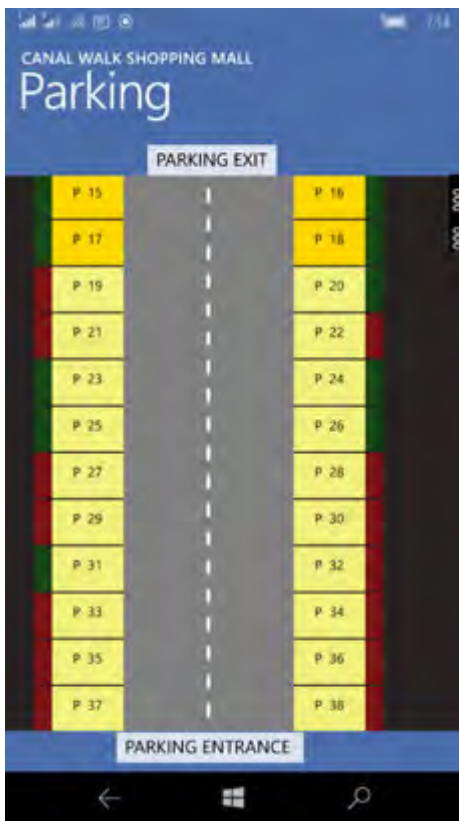


Figure 4.19 Scenario 5.



Figure 4.20 Scenario 6.



Figure 4.21 Scenario 7.



Figure 4.22 Scenario 8.

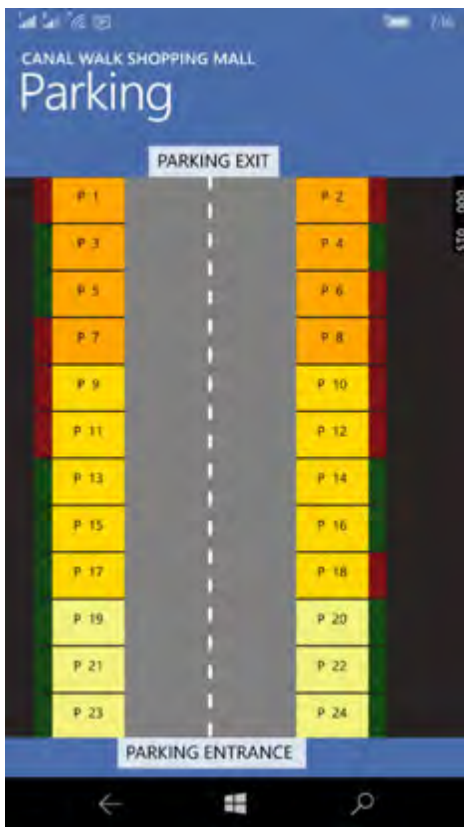


Figure 4.23 Scenario 9.



Figure 4.24 Scenario 10.

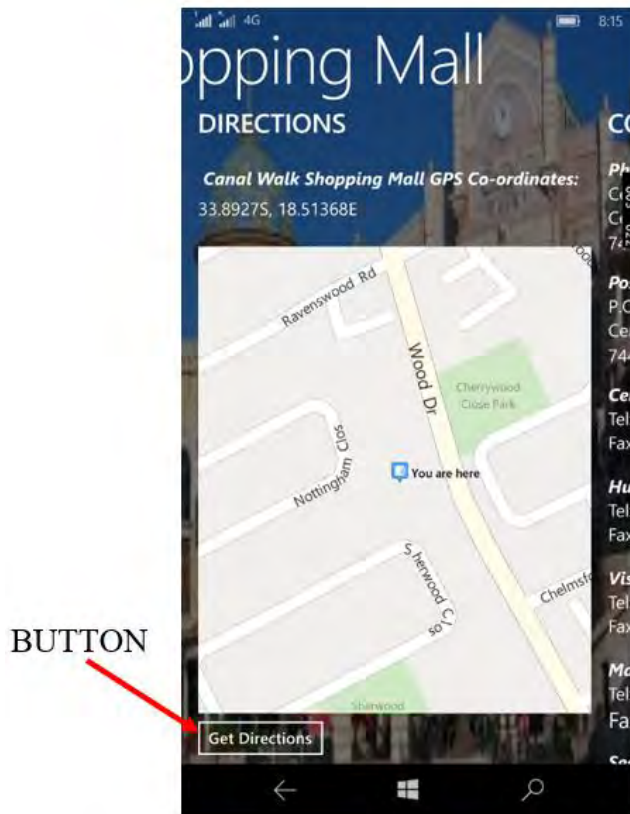


Figure 4.25 Hub section 5.

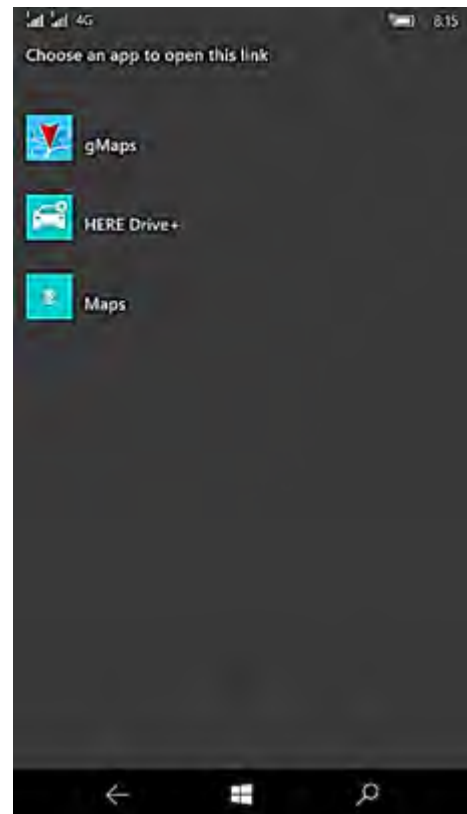


Figure 4.26 Installed external location application.

4.1.3.5 Hub Section Five

This hub section presents the **Location Service**. The current location of the user is displayed on this page. Figure 4.25 shows the location where the author of this research was when the Smart Shopping service application was tested. Tapping on the button on this page will navigate to a frame with a list of external location applications that would have been installed by the user on the mobile phone. In this research three external location applications were installed on the mobile phone and these are Google Maps, Here drive+ and Maps, and these are presented on the page navigated to as shown in Figure 4.26. This shows that the Smart Shopping service developed in this research used SOA and service composition principles discussed in chapter 2, making it reusable, composable and interoperable with other external services that are independent to the Smart Shopping service and that were not developed in this research.

If the icon for Google Maps is tapped, the Google map application will be opened as shown on Figure 4.27. The route for getting from the user's current location to Canal Walk Shopping

Mall is then displayed. Figure 4.28 shows the starting point which is the current location of the user, which is the same as the one which would have been shown in hub section 5. Figure 4.29 shows the destination point which is Canal Walk Shopping Mall. Tapping on the back button on this figure will make the display screen navigate back to hub section 5 page.

If the Here drive+ icon was pressed, Here drive+ external application will be opened as shown in Figure 4.30. The route for getting to Canal Walk Shopping Mall from the user's current location will then be displayed. Figure 4.31 shows the starting point which is the same as the one which would have been shown in hub section 5 and is the current location of the user. Figure 4.32 shows the destination point which is Canal Walk Shopping Mall. Pressing the back button on Figure 4.32 will navigate the display screen back to hub section 5 page.

When the external location application is selected, the user of the Smart Shopping service will get to have the full functionality of these external services. They will be able to change their settings depending on the capabilities of the external application, for example, they would be able to include voice navigation, change the unit used to measure the distance for example either kilometres or miles, change the way the map looks or turn on the speed limit warning.



Figure 4.27 Google Maps start page.

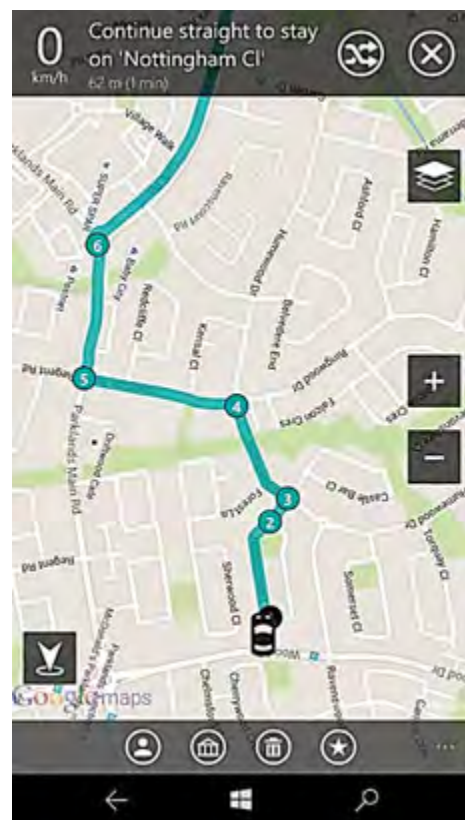


Figure 4.28 Google Maps user start point.

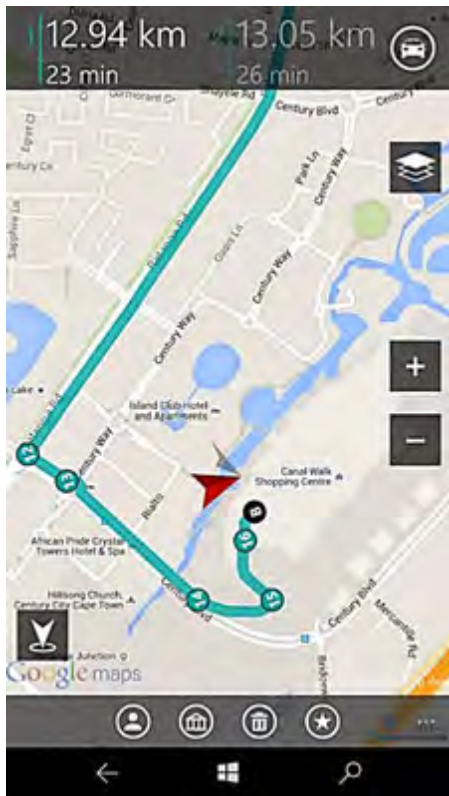


Figure 4.29 Google Maps destination point.



Figure 4.30 Here Drive+ start page.

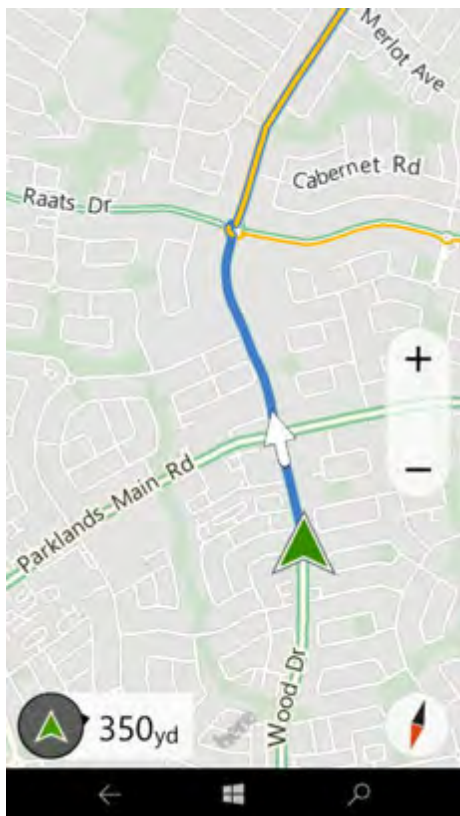


Figure 4.31 Here Drive+ user location

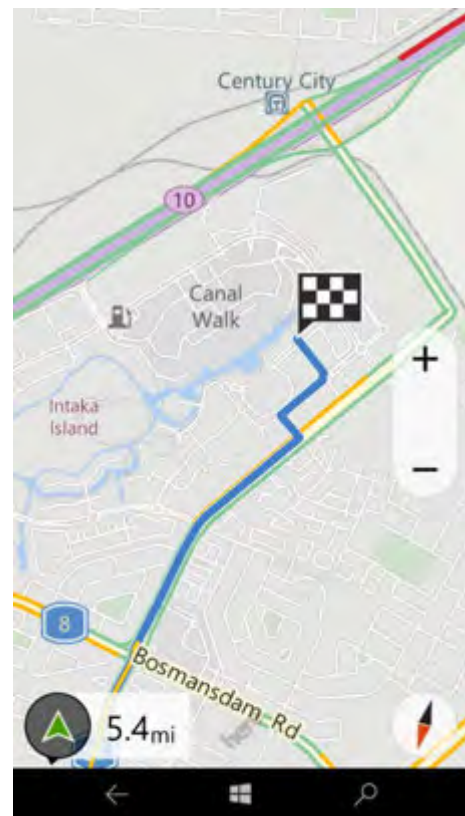


Figure 4.32 Here Drive+ destination point.

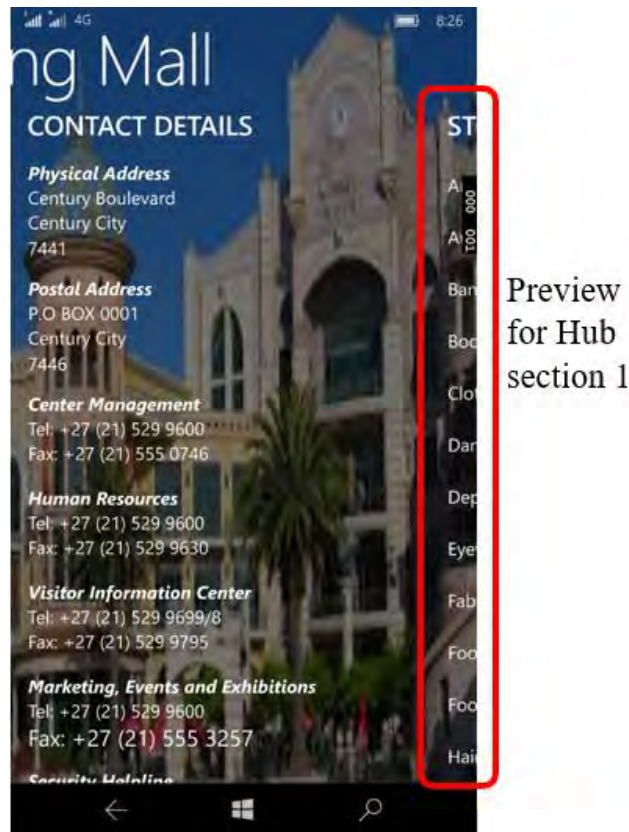


Figure 4.33 Hub section 6.

4.1.3.6 Hub Section Six

This is the last hub section on the developed Smart Shopping mobile application. It presents the last part of the **Mall Information Service** which are the contact details of the shopping mall as shown in Figure 4.33. This page is important to people wanting to get more information about the shopping mall for example if they are interested in starting a business there or ask some information from the visitor information centre. This page can be scrolled up or down in order to see all the information on the page. Swiping to the left of this page will navigate back to hub section one shown in Figure 4.9 store categories page.

4.2 Data Layer Implementation

The Data Layer design process described in the previous chapter in section 3.2.2.2 is used to implement the Data Layer. The first and the second step were covered in the afore mentioned section. The following sections present step three to the last one.

4.2.1 Deploying the Azure SQL Database

In this research, a SQL database is created on Microsoft Azure and is deployed on a data centre in Northern Europe as shown in Figure 4.34. The detailed description on how this database can be set up is shown in Appendix B.

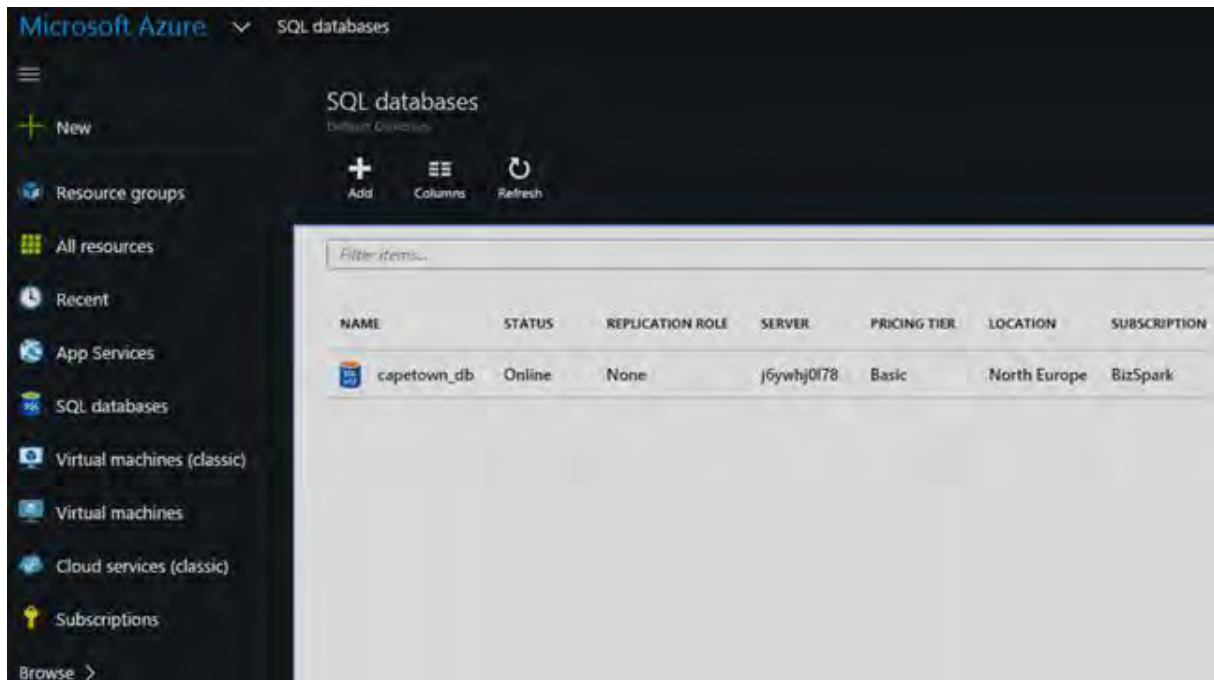


Figure 4.34 Deployed Azure SQL database.

4.2.2 Database Table Design

The database tables that were designed in this research are as shown below. These were stored on the Azure SQL database.

Parking Scenarios Table

Figure 4.35 shows the Parking Scenarios table that was stored on the database and the 10 scenarios that were designed by the researcher. The states of the parking bays in each scenario are also shown with the value 1 representing an available parking bay and the value 0 for an occupied parking bay.

| Scenario | ParkingBay1 | ParkingBay2 | ParkingBay3 | ParkingBay4 | ParkingBay5 | ParkingBay6 | ParkingBay7 |
|----------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 2 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 3 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 4 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 5 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 6 | 1 | 1 | 0 | 0 | 1 | 1 | 1 |
| 7 | 1 | 1 | 0 | 0 | 1 | 1 | 1 |
| 8 | 1 | 0 | 0 | 0 | 0 | 1 | 1 |
| 9 | 0 | 0 | 1 | 1 | 1 | 0 | 0 |
| 10 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

Figure 4.35 Parking scenarios table.

Store Categories Table

Figure 4.36 shows part of the Store Categories table and the data that was stored there.

| StoreCategoryID | StoreCategoryName | ShoppingMailID |
|-----------------|---|----------------|
| 1 | Art, Antiques, Curios & Gifts | 1 |
| 2 | Audio/Electronic, PC, Cellphone & Photo | 1 |
| 3 | Banks, Forex & Financial | 1 |
| 4 | Books, Cards, Gifts, & Stationery | 1 |
| 5 | Clothing | 1 |
| 6 | Dance | 1 |
| 7 | Department Stores | 1 |
| 8 | Eyewear & Optometrists | 1 |
| 9 | Fabric & Sewing | 1 |
| 10 | Food | 1 |
| 11 | Footwear | 1 |
| 12 | Hair, Health, & Beauty | 1 |
| 13 | Home, Decor, Interior & Furniture | 1 |
| 14 | Jewellery & Accessories | 1 |
| 15 | Luggage & Leather | 1 |
| 16 | Motor & Motorbikes | 1 |
| 17 | Movies, Music & Entertainment | 1 |
| 18 | Offices | 1 |
| 19 | Pet Shops | 1 |
| 20 | Services | 1 |

Figure 4.36 Store categories table.

Stores Table

Figure 4.37 shows part of the Stores table and the data that was stored there.

| StoreID | StoreName | StoreCategoryID | StoreNumber | TelephoneNumber | ShoppingMallID | StoreImage |
|---------|-----------------------|-----------------|-------------|-----------------|----------------|---------------------------------------|
| 1 | Big Blue | 1 | 28 LL | (021) 551 1566 | 1 | http://www.mallguide.co.za/uploadimag |
| 2 | Cape Arts & Crafts | 1 | 27 LL | (021) 555 3699 | 1 | http://www.mallguide.co.za/uploadimag |
| 3 | Chetson Art Gallery | 1 | 581 UL | (021) 551 5012 | 1 | http://www.mallguide.co.za/uploadimag |
| 4 | Malaikah Curios | 1 | A19 LL | 0827309730 | 1 | http://www.mallguide.co.za/uploadimag |
| 5 | Massai Mara Curios | 1 | A4 LL | (021) 555 3101 | 1 | http://www.mallguide.co.za/uploadimag |
| 6 | Mombasa | 1 | A3 LL | (021) 552 7974 | 1 | http://www.mallguide.co.za/uploadimag |
| 7 | SA Framing | 1 | 561 UL | (021) 552 3727 | 1 | http://www.mallguide.co.za/uploadimag |
| 8 | Altech Autopage | 2 | 15 LL | (021) 555 4100 | 1 | http://www.mallguide.co.za/uploadimag |
| 9 | Audiolens | 2 | 74 LL | (021) 552 0980 | 1 | http://www.mallguide.co.za/uploadimag |
| 10 | Cell C | 2 | C5 UL | (021) 555 1676 | 1 | http://www.mallguide.co.za/uploadimag |
| 11 | Cellucity | 2 | 273A LL | (021) 555 0038 | 1 | http://www.mallguide.co.za/uploadimag |
| 12 | Cellular Supreme | 2 | 84 LL | (021) 552 9567 | 1 | http://www.mallguide.co.za/uploadimag |
| 13 | Cellular Village | 2 | 633 UL | 0747341715 | 1 | http://www.mallguide.co.za/uploadimag |
| 14 | Computer Mania | 2 | 641 UL | (021) 555 3020 | 1 | http://www.mallguide.co.za/uploadimag |
| 15 | Dion Wired | 2 | 109 LL | (021) 528 5000 | 1 | http://www.mallguide.co.za/uploadimag |
| 16 | HI | 2 | 120 LL | (021) 552 2533 | 1 | http://www.mallguide.co.za/uploadimag |
| 17 | IFIX | 2 | 82 | (021) 551 0912 | 1 | http://www.mallguide.co.za/uploadimag |
| 18 | Incredible Connection | 2 | 407A UL | (021) 528 3400 | 1 | http://www.mallguide.co.za/uploadimag |
| 19 | Istore | 2 | 326 UL | (021) 003 0400 | 1 | http://www.mallguide.co.za/uploadimag |

Figure 4.37 Stores table.

4.2.2.1 Database Table Design Programming Language

The programming language that is used to, manipulate and edit the data that was stored in the Azure SQL database is called Structured Query Language (SQL) programming language.

```

1 SELECT StoreName, TelephoneNumber FROM capetown.Stores;

```

| StoreName | TelephoneNumber |
|---------------------|-----------------|
| Big Blue | (021) 551 1566 |
| Cape Arts & Crafts | (021) 555 3699 |
| Chetson Art Gallery | (021) 551 5012 |
| Malaikah Curios | 0827309730 |
| Massai Mara Curios | (021) 555 3101 |
| Mombasa | (021) 552 7974 |
| SA Framing | (021) 552 3727 |
| Altech Autopage | (021) 555 4100 |
| Audiolens | (021) 552 0980 |
| Cell C | (021) 555 1676 |
| Cellucity | (021) 555 0038 |
| Cellular Supreme | (021) 552 9567 |
| Cellular Village | 0747341715 |
| Computer Mania | (021) 555 3020 |

Figure 4.38 selecting the store names and their respective telephone numbers.

Figure 4.38 shows the SQL code that is used to retrieve the store names and their respective telephone numbers from the Stores table using the SQL code shown below:

```
SELECT StoreName, TelephoneNumber FROM capetown.Stores;
```

Figure 4.39 shows the SQL code that is used to retrieve all the information in the Stores table using the SQL code shown below:

```
SELECT * FROM capetown.Stores;
```

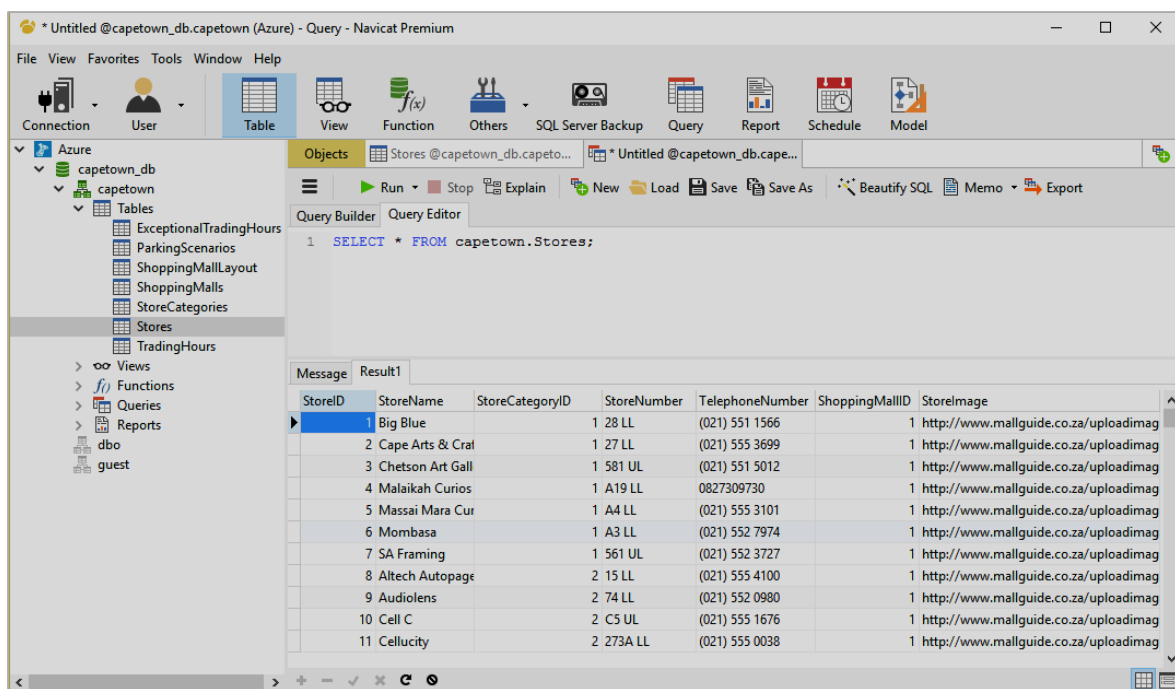


Figure 4.39 Selecting all the information from the Stores table.

4.2.3 Designing and Implementing API calls

As described in section 3.1.2, Microsoft Azure is used to create an Azure Mobile Service which is deployed onto a server in Northern Europe. Azure mobile service is used to implement API calls to get data from the Azure SQL database.

Figure 4.40 shows the API calls that are created. Figures 4.41 and 4.42 show the code that was written for the API call to get stores in a category and to get the parking scenarios, from the created Azure SQL database respectively.

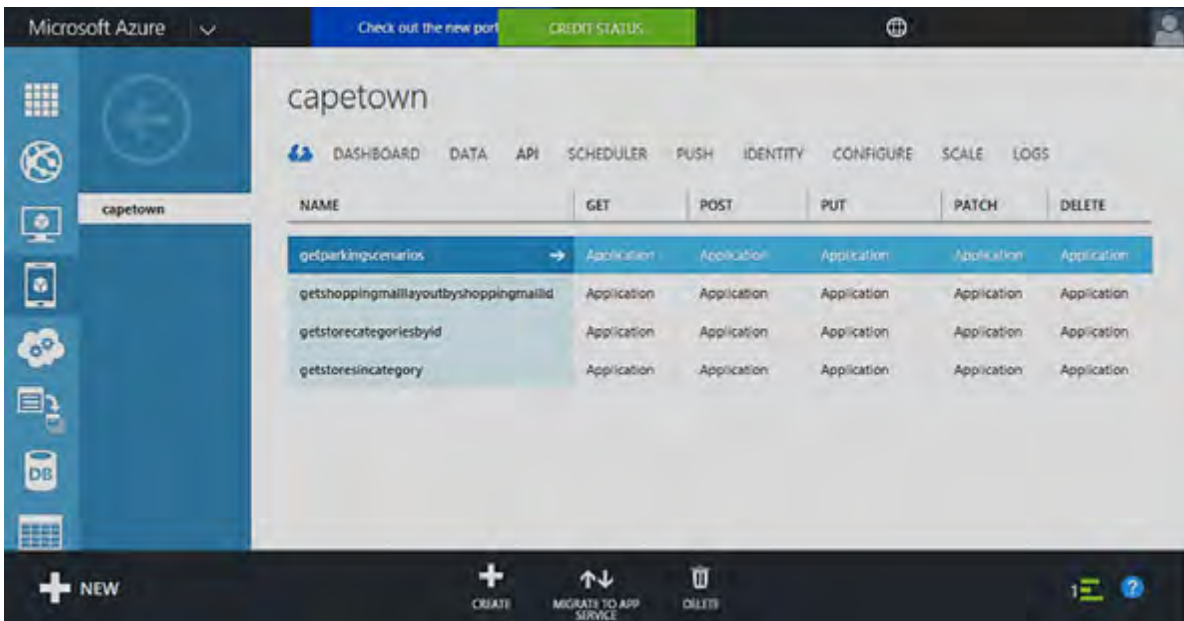


Figure 4.40 API calls.

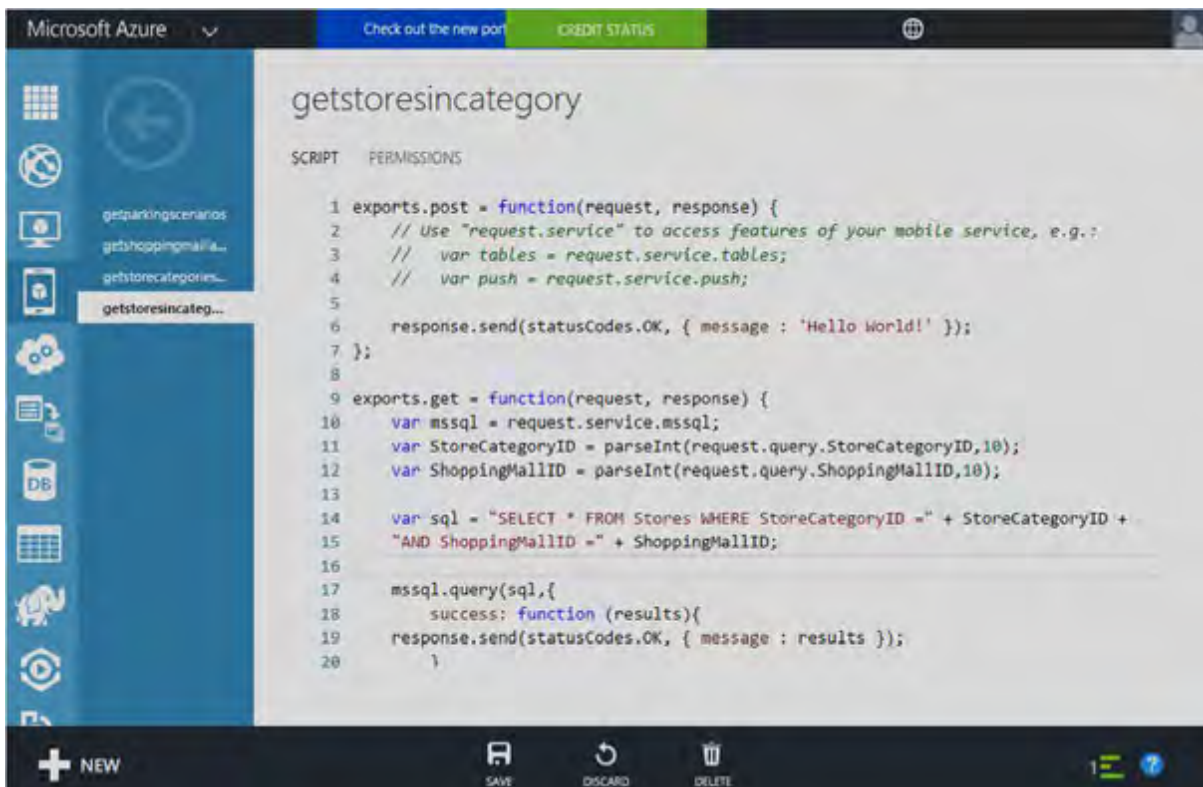


Figure 4.41 SQL code for getstoresincategory API call.

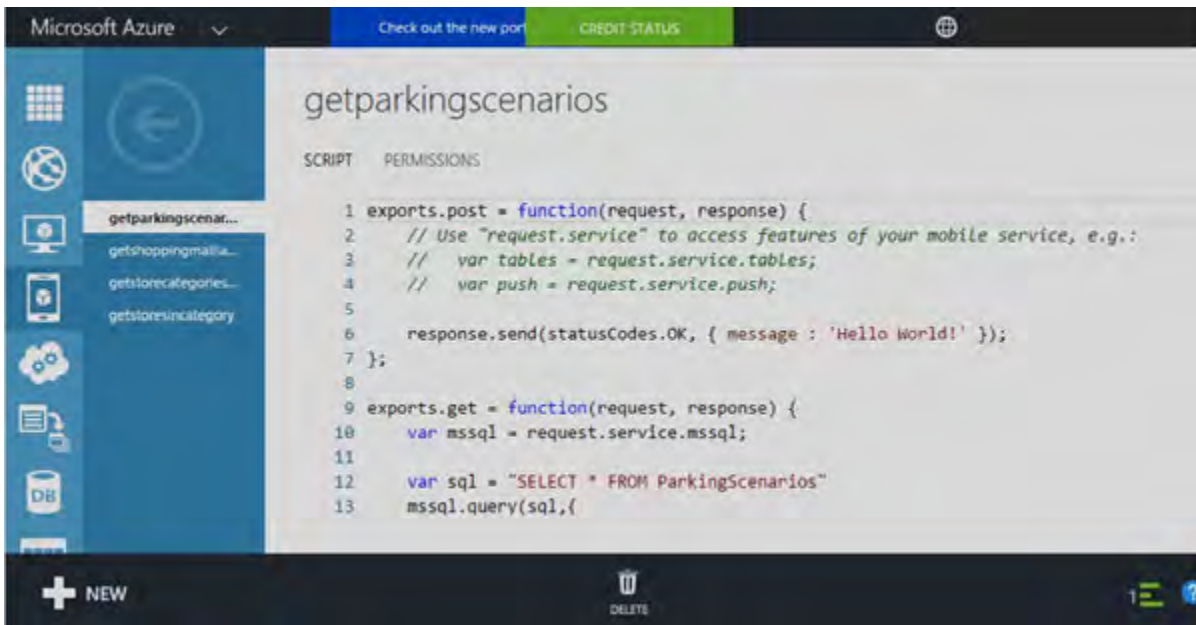


Figure 4.42 SQL code for getparkingscenarios API call.

4.2.4 Testing API calls

Postman was used to test the API calls that were implemented. Figure 4.43 shows the results obtained when the get Shopping Mall Layout API call was tested.

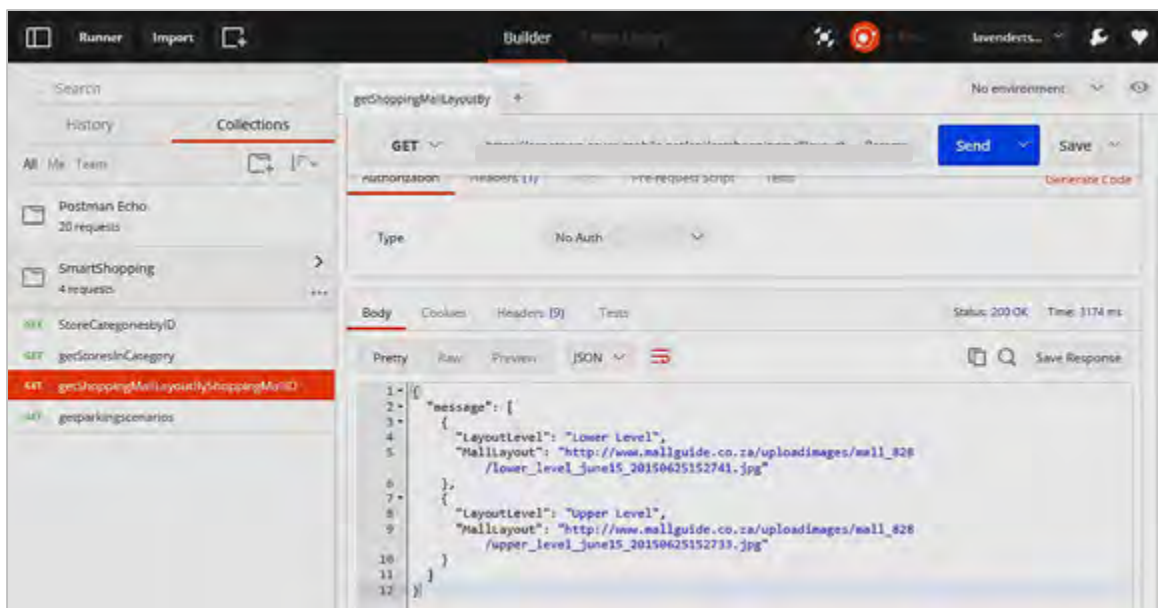


Figure 4.43 get shopping mall layout API call.

Figure 4.44 shows the results obtained when the Store Categories API call was tested. The results were a data output in the form of a JavaScript Object Notation (JSON) file.

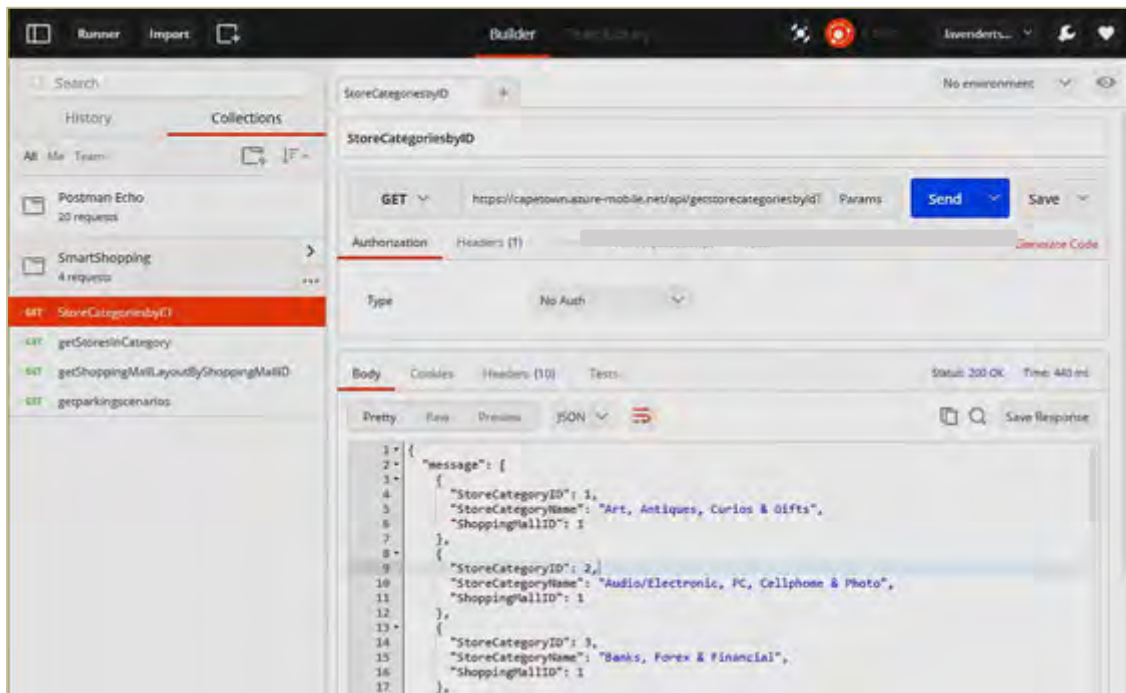


Figure 4.44 Store Categories API call.

4.2.5 Data Access

After the implementation of the API calls, the data needs to be accessed. This encapsulates all the required logic to retrieve data that would be needed to be displayed on the application from the data source, which in our case is the Azure SQL server database. Instances of data classes are to be made and these should have all the properties of the data needed for the application but these classes should not have any methods to implement the business logic. These data classes are also known as the Model. Data binding is used to connect the presentation layer/view to the model. [41]

Figure 4.45 shows the data model classes that were made and are represented by the red arrows.

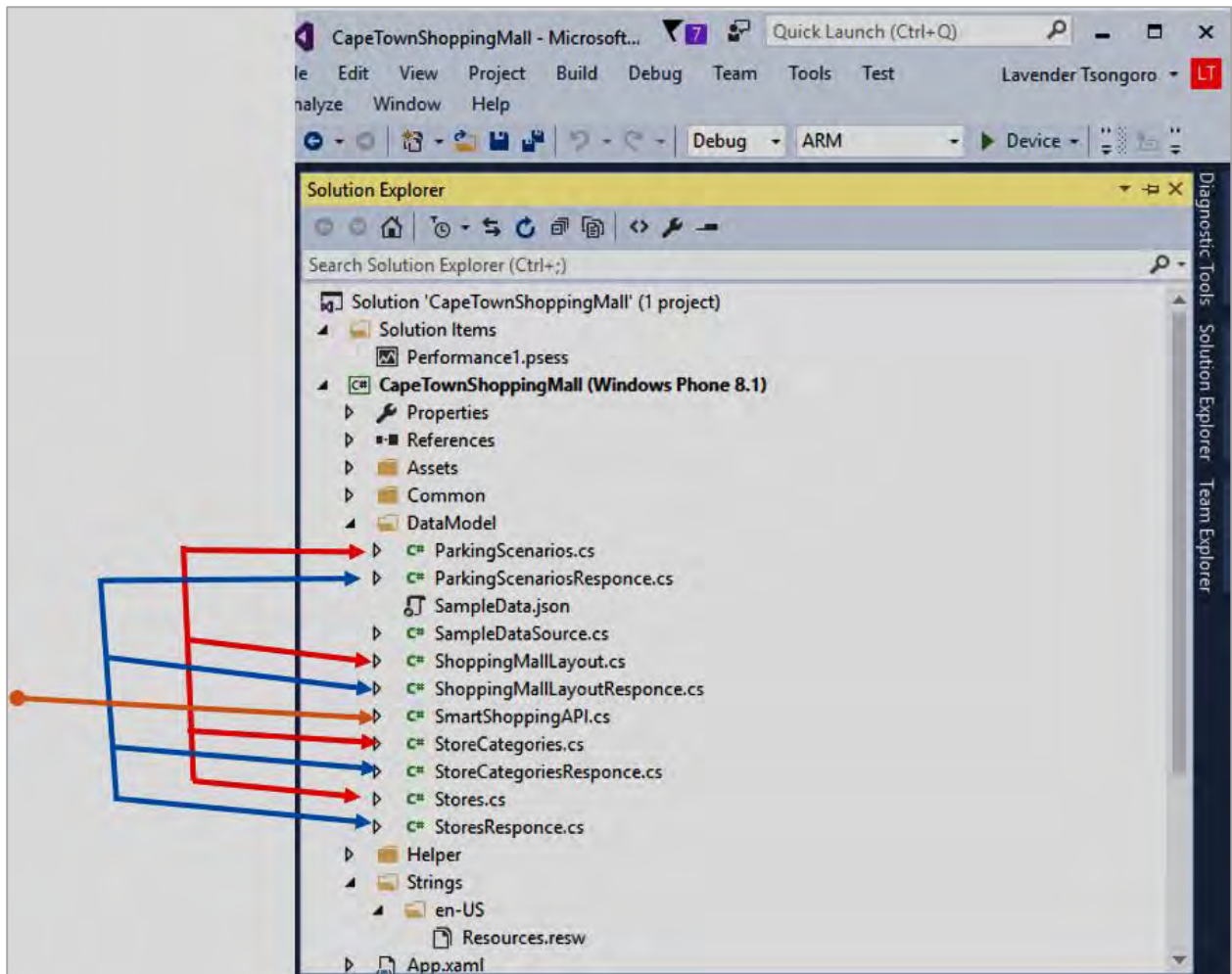
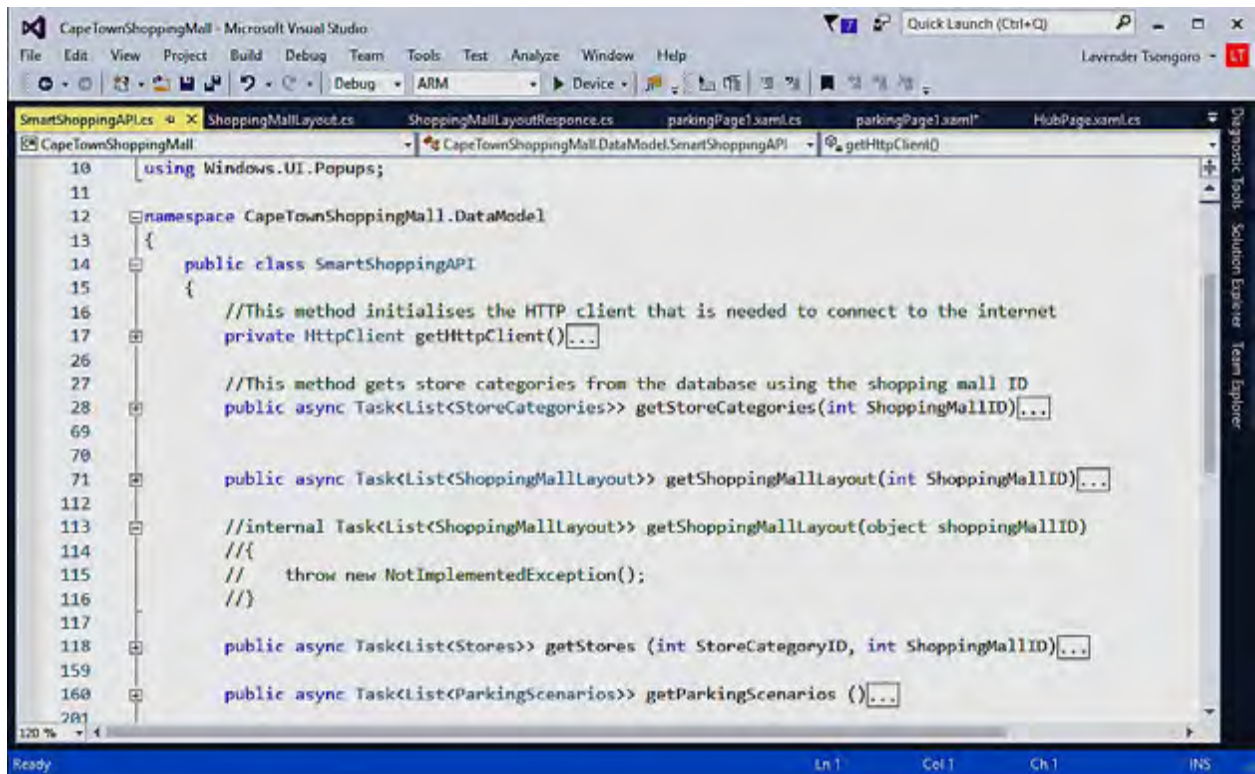


Figure 4.45 Data Model (red arrows)

4.3 Business Layer Implementation

The next step in designing a mobile application is to develop the business layer. This layer involves the application logic and business rules used to manage business processes [11]. The business layer may have an optional application facade which combines business processes into a single message based operation. This layer will also have business components for example the ones that process business rules, interact with data access components, process requests or serve for business rules application. Business components that are used to pass data between other components are called business entities and an example of the data that can be transferred between components are database entities such as tables and views. The business layer may also have business workflows which are multi-step business processes. [11], [38]

Errors and exceptions must be managed effectively in this layer because not doing so may result in the application being vulnerable to Denial of Service (DoS) attacks or sensitive information of the application being revealed. [38], [11]



```
10 using Windows.UI.Popups;
11
12 namespace CapeTownShoppingMall.DataModel
13 {
14     public class SmartShoppingAPI
15     {
16         //This method initialises the HTTP client that is needed to connect to the internet
17         private HttpClient getHttpClient()...
26
27         //This method gets store categories from the database using the shopping mall ID
28         public async Task<List<StoreCategories>> getStoreCategories(int ShoppingMallID)...
69
70
71         public async Task<List<ShoppingMallLayout>> getShoppingMallLayout(int ShoppingMallID)...
112
113         //internal Task<List<ShoppingMallLayout>> getShoppingMallLayout(object shoppingMallID)
114         //{
115             // throw new NotImplementedException();
116         }
117
118         public async Task<List<Stores>> getStores (int StoreCategoryID, int ShoppingMallID)...
159
160         public async Task<List<ParkingScenarios>> getParkingScenarios ()...
201
```

Figure 4.46 Smart ShoppingAPI.cs class

In this research the business layer is comprised of classes with properties and methods that were used to connect the data from the Azure SQL database to the View of the application. As show in Figure 4.45 by the blue arrows, these classes have properties that connect to the classes in the Model.

Another class called SmartShoppingAPI.cs was also created and this had the methods used to connect to the database to get the application data needed as shown in Figure 4.45 by the orange arrow and also in figure 4.46. This data is then sent to the presentation layer to be displayed on the mobile phone's screen.

In the next chapter the performance of the Smart shopping service was tested and the results are presented.

5 System Performance Evaluation

In this chapter, the results regarding performance tests that were conducted on the Smart Shopping Service are presented. The analysis of these results is connected to the validation of the research problems and the degree of satisfaction of the overall system design.

5.1 Mobile Service Performance Testing

Mobile performance tests are software tests which are conducted to ensure that services are robust, stable, free of bugs and usable. These are important because mobile phone users expect applications to be reliable, fast, easy to use and that they should launch in under two seconds [13]. Average smart phone user checks their phones every six second and has about 40 applications installed on their device [13].

There are more than two million applications in the application stores of the major vendors and more than one application is normally available to perform the same task. Due to this massive competition between mobile applications, it is of high importance to develop a high performance mobile application. [13]

5.2 Service Testing Methods

Mobile services can either be tested on a physical mobile phone device, a mobile phone emulator or a mobile phone simulator. Emulators are run on a computer and get internet access by connecting to the Local Area Network (LAN) via a virtual Ethernet port [42]. A real mobile phone device gets internet access via a radio interface [42]. The differences between these three are discussed in the following sections.

5.2.1 Mobile Phone Simulators

A mobile simulator is a moderately complex software application that simulates a small subset of the real device's behaviours and hardware features. They attempt to duplicate the behaviour of the mobile device and are very similar to the simulated mobile phone platform. [13]

5.2.2 Mobile Phone Emulators

The mobile application is executed on a desktop computer after the application source code is compiled on the host computer. Mobile phone emulators are slower than mobile phone simulators. Most of the device specific hardware elements such as sensors (e.g. light sensors, proximity sensors, acceleration sensors and location sensors) or touch gestures cannot be emulated. Due to this reason, when the Smart Shopping service was run on the mobile phone emulator, the location service could not be enabled. Mobile phone emulators duplicate the entire architecture of the mobile device and are closer to the target platform. [13]

In this research, a Windows 8.1 Phone Emulator was used to test the performance of the Smart Shopping service that was developed. The Windows Phone Emulator is a desktop application that runs on Microsoft's virtualisation software called Hyper-V. This arrangement provides a virtualised environment that has similar performance to that of a physical device and enables testing and debugging of the mobile application code. [43]



Figure 5.1 Emulator 8.1 WVGA 4 inch 512MB start screen Hub Section 1 screen.

The emulator that was used in this research is the **Emulator 8.1 WVGA 4 inch 512MB** and has the following properties:

- Operating System: Windows 8.1 Mobile operating system;
- Available RAM: 512MB;
- Screen resolution of 800 x 480 pixels [43].

Figure 5.1 shows the start screen of the Emulator 8.1 WVGA 4 inch 512MB and also the Hub Section one screen for the Smart Shopping application displayed on the emulator.

5.2.3 Physical Mobile Phone Device

Physical mobile phones can be used to conduct performance tests of a mobile application and these provide the real user environment. Initially, during the testing process, before the application is installed on the mobile phone, the mobile phone has to be connected to the computer where the application source code is running, via a USB cable. The status of the **Windows Phone IP over USB Transport (IpOverUsbSvc)** has to be changed from stopped to running first. This is done by going to the computer's Task Manager, select Services, then select the IpOverUsbSvc and change its status from stopped to running as shown in Figure 5.2.

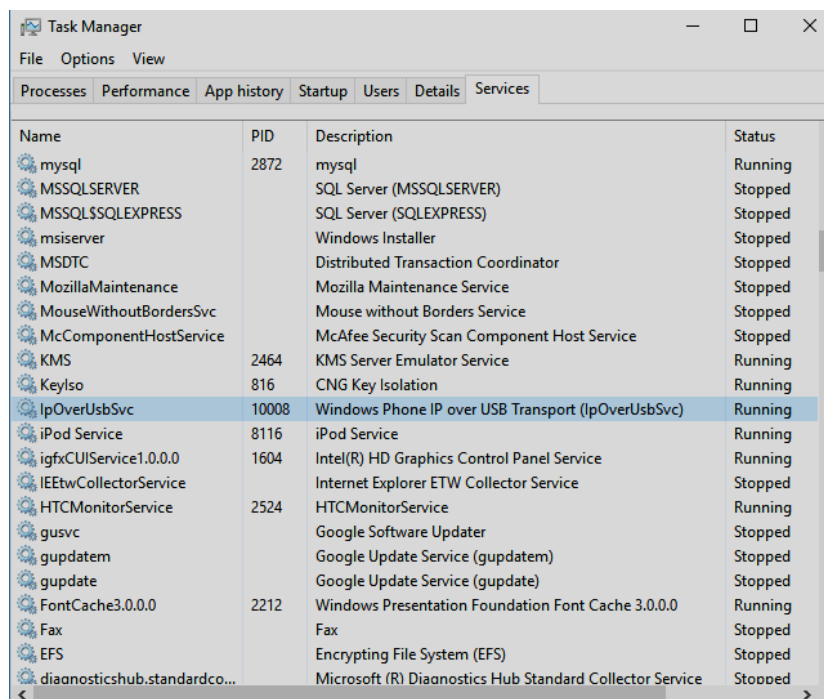


Figure 5.2 IpOverUsbSvc status change.

The physical device that is used to do the performance tests of the Smart Shopping application is a mobile phone having the following features:

- Device model: Nokia Lumia 640XL LTE Dual sim;
- Installed RAM: 1GB;
- Operating System: Windows 10 Mobile operating system;
- Screen resolution: 720x1280;
- CPU: Quad-core 1.2 GHz Cortex-A7.

The computer that is used during the service implementation is a Toshiba laptop running Windows 10 Professional operating system and has the following features:

- Processor: Intel® Core (TM) i5-3210M CPU @2.50GHz;
- Memory (RAM): 6GB;
- Operating System: Windows 10 Professional 64-bit OS, X64 based processor.

5.3 Performance tests using Emulator/Simulators

The advantages and disadvantages of testing applications using either mobile phone simulators or mobile phone emulators are as follows:

Advantages

- Emulators/simulators are cheap and come free with the mobile SDK installation;
- Simple to use [13].

Disadvantages

- Emulators/simulators are not the same as real mobile phone device;
- Emulators/simulators have a plain and simple mobile operating system;
- Emulators/simulators do not have the hardware sensors and interfaces such as camera, GPS;
- The data networks are not real;

- Emulators/simulators do not offer the same performance as real devices in terms of CPU, GPU (Graphics processing unit), memory or sensors [13].

5.4 Performance tests using real devices

Advantages

- Test conducted in real user environment hence results are closer to real user experience;
- There is access to real device hardware and software features;
- Real device shows real performance behaviour of the application;
- Tests are done in real data networks [13].

Disadvantages

- The cost of devices needs to be considered. When the device features are updated in newer versions of the phones, the latest devices have to be regularly bought to test if the app will run on the new features;
- Maintenance of the devices is time consuming [13].

5.5 Smart Shopping Service System Performance Tests

The performance measures that were used to test the Smart shopping service are as follows:

- Application timeline;
- Memory usage;
- Energy consumption;
- Bandwidth usage.

The Diagnostic Tool which is part of Visual Studio Community 2015 is used to conduct the performance tests. The results that were obtained from the tests are presented in the next sections:

5.5.1 Application Timeline

Application timeline shows the time spent in the application. This test is relevant to this research in that it shows how fast the application started and was loading the application pages onto the phone screen. This affects the user acceptance of the service as they are more likely to continue using applications that load faster. Figure 5.3 shows the application timeline results obtained when a mobile phone was used to test the application's performance and Figure 5.4 shows the results obtained when a mobile phone emulator was used to conduct the same tests. The application start-up time is the time taken by the application from the time it is launched to the time the first screen of the application is displayed on the screen.

The typical allowable start-up time is less than 5 seconds [45]. For the performance tests conducted in this research, the application *start-up time* for the mobile phone is 487.93 milliseconds and that for the mobile emulator is 1.37 seconds. *Parsing* the hub page for the Smart Shopping service using the mobile phone took 13.83 milliseconds and this took 50.76 milliseconds when using the mobile emulator for the tests. The results show that the mobile emulator is much slower than the mobile phone and that the start-up time of the Smart Shopping application is less than the typical recommended start up time of 5 seconds, hence it meets these requirements.

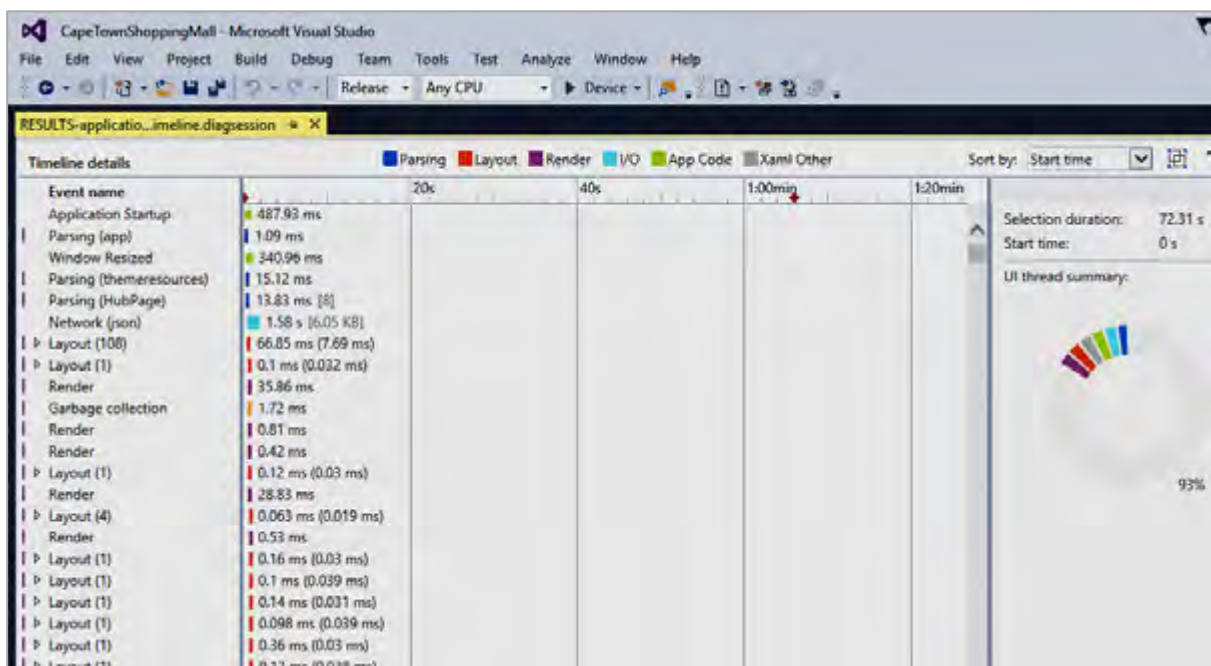


Figure 5.3 Application timeline for mobile phone.

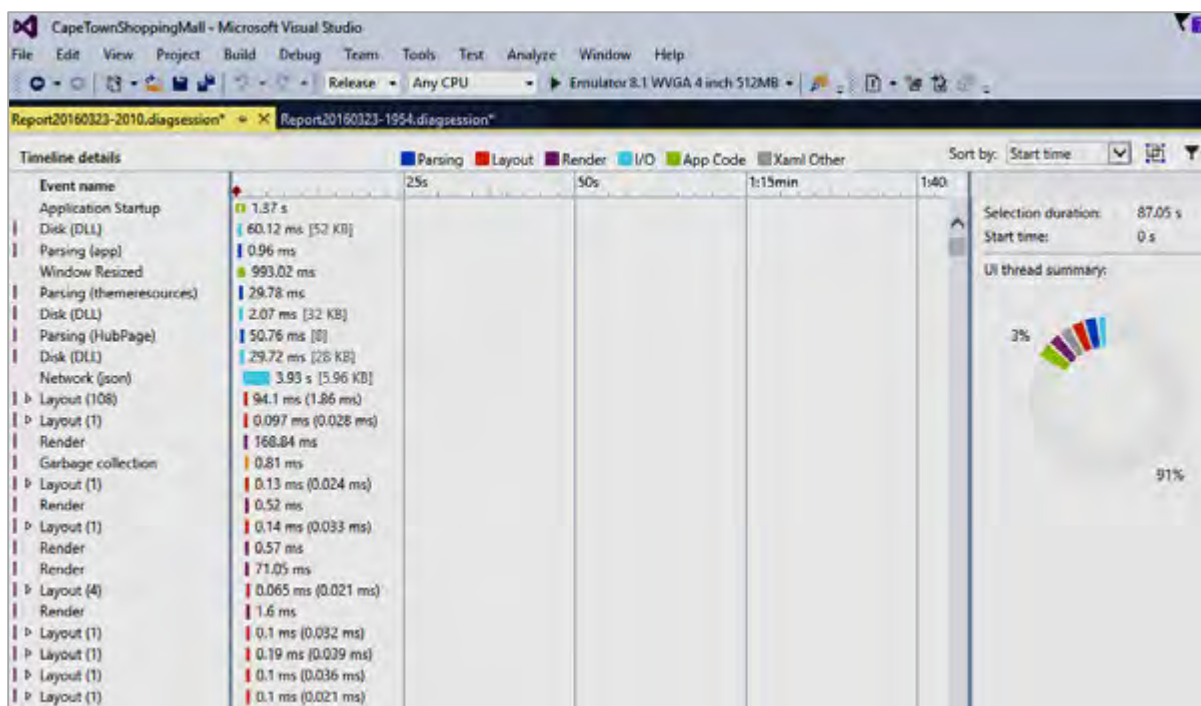


Figure 5.4 Application timeline for mobile phone emulator.

5.5.2 Memory usage

Memory usage is significant because if the memory consumption of an application is very high, the application may not be able to run on low end devices which normally have lower phone memory. To ensure that an application runs on more devices with different memory sizes, it is advised to design an application which runs on lower memory phones, which are typically phones with 512MB of memory and this ensures that they will definitely run on higher memory devices. Applications designed for devices with higher memory size will not run on devices with lower memory. Table 5.1 shows the capacity of the memory usage of application for physical mobile phones with different memory size and operating systems [46].

Table 5-1 Maximum application memory usage [46].

| Application OS | Lower Memory Phones | 1 GB Phones | 2 GB Phones |
|--------------------------------|---------------------|-------------|-------------|
| Windows Phone 8.0 applications | 180 MB | 380 MB | 780 MB |
| Windows Phone 8.1 applications | 185MB | 390 MB | 825 MB |

Applications developed for Windows phones devices with Mobile phone operating systems and memory as shown above must have a corresponding maximum memory usage as indicated in the table above. To reduce the strain on the mobile device memory, applications can be installed on an SD card which is inserted into the mobile phone device.

For the memory usage tests conducted in this research for the Smart shopping service, the maximum amount of phone memory that was used by the mobile phone during the application analysis session was 20.8MB as shown in Figure 5.5, which was less than that obtained for the mobile phone emulator which was 26.0 MB shown in Figure 5.6. With reference to Table 5.1, these results show that the Smart Shopping application will be able to function well on both low and high end devices without straining the memory of the mobile device.

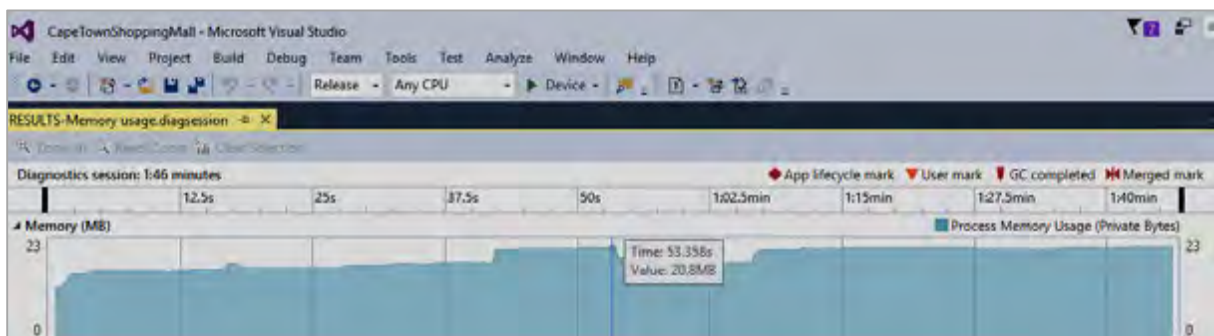


Figure 5.5 Memory usage for mobile phone.



Figure 5.6 Memory usage for mobile phone emulator.

During the application analysis session, snapshots were taken for a sample of three frames of the application and the corresponding memory used to load them was recorded as shown in Figure 5.7 and Figure 5.8.

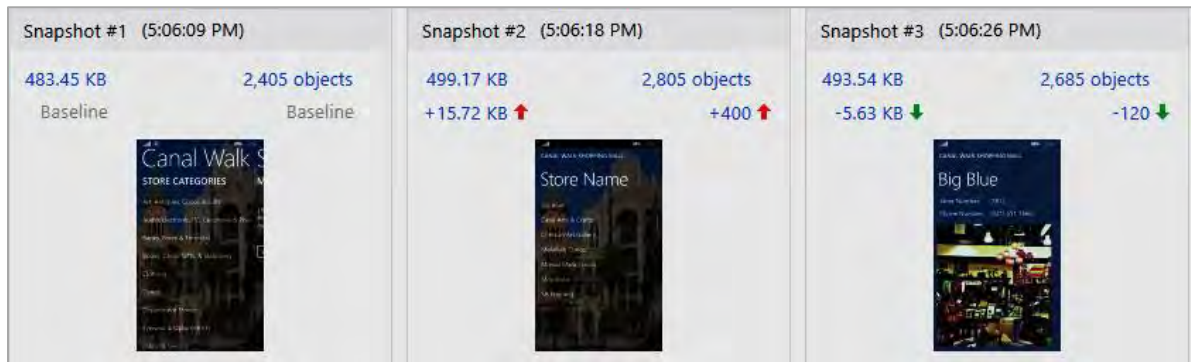


Figure 5.7 Snapshot 1-3 for mobile phone.

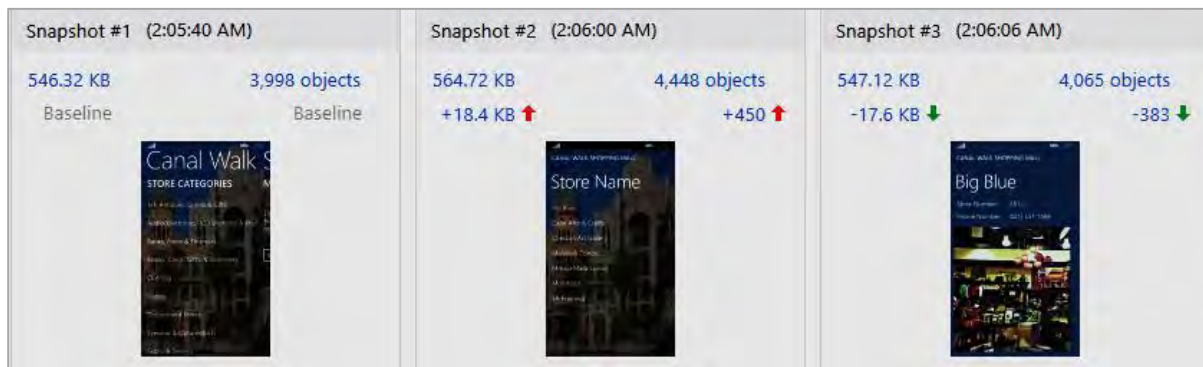


Figure 5.8 Snapshot 1-3 for mobile phone emulator.

Figures 5.7 and 5.8 show that to load the Store Categories frame using the mobile phone, 483.45KB of memory was used, which was lower than the 546KB of memory used when testing with the mobile phone emulator. Loading the Store Name frame used 499.17KB of memory when using the mobile phone device and this was lower than the 564.72KB of memory that was used when the mobile phone emulator was used in the test. Snapshot 3 shows that there was a decrease in the phone memory that was used when loading the Store Details frame for a store called Big Blue, which was a 5.63KB decrease for the mobile phone and 17.6KB decrease for the mobile emulator. 493.54KB of the mobile phone's memory was used to load this page, which was lower than the 547.12KB which was used when the mobile phone emulator was used in the tests.

5.5.3 Energy Consumption

This test is relevant to this research in that optimising mobile applications for energy efficiency means that the mobile phone battery will last longer hence increase user satisfaction. Figures 5.9 and Figure 5.10 show the power usage for the mobile phone and the mobile phone emulator

respectively. This performance test was conducted at a location where there was no cellular network coverage hence the only form of data transmission that was done was via the Wi-Fi. More battery energy was used when using the mobile phone emulator than when the mobile phone was used.

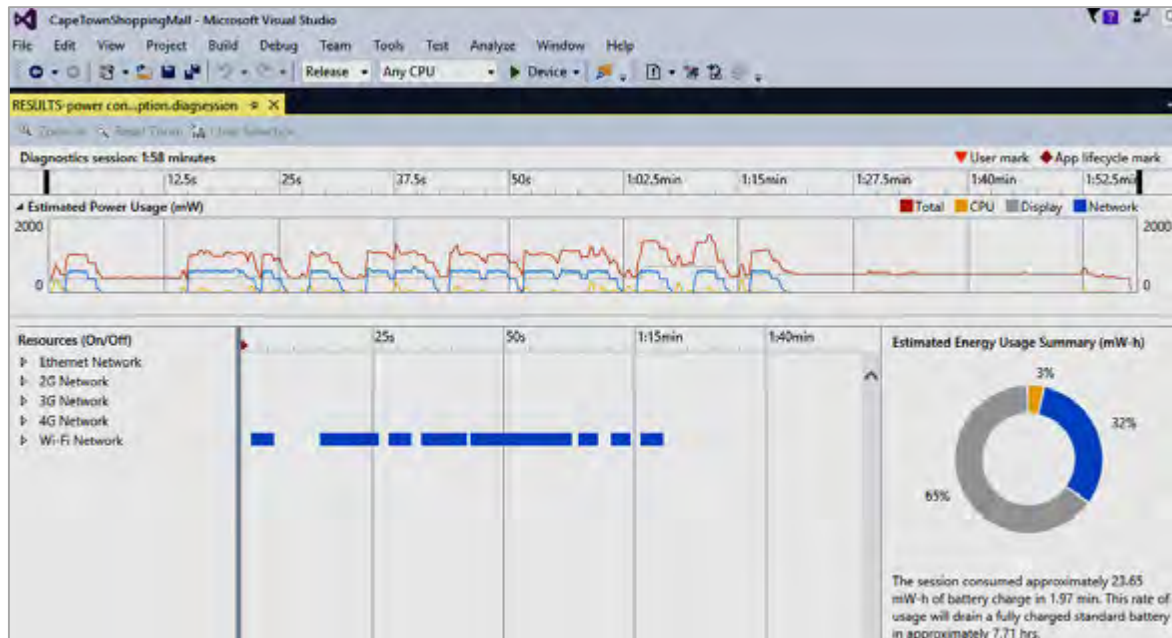


Figure 5.9 Power usage for the mobile phone.



Figure 5.10 Power usage for the mobile phone emulator.

This may have been because the mobile phone emulator took more time for it to load screens onto the display and was much slower. The CPU energy usage was the same for both cases. More energy was used for the display when the mobile phone was used than when the mobile

phone emulator was used. This may have been because the mobile phone had a bigger screen with larger screen resolution than that for the mobile phone emulator.

The energy consumption is the measure of the electrical energy used by the phone battery to run the application. Its power consumption is the rate at which the electrical energy is being consumed. The display power is dependent on the colours used in the application design. Generally, darker colours consume less power than brighter colours. Cellular networks use the battery's power during data transmission and also during the tail state when the radio interface remains in high power state after the data is finished to be transmitted. The idle state also consumes the battery power. [47]

Wi-Fi networks also use the battery power during data transmission. The tail state consumes less power and changes quickly to idle state with very little energy consumption. Generally, Wi-Fi uses less battery power than cellular networks when transmitting the same amount of data. [47] It is however very important to note that this is dependent on the radio transmission range and the signal quality. The greater the transmission range, the larger the amount of battery power that is used. Better signal quality will result in less consumption of battery power and as this value deteriorates, more battery power is consumed.

Applications with heavy computations use more battery power than those with less computations that require less processing. [47]

To reduce power usage by the display, light colours should be avoided when possible during the design of mobile applications. The application's computation tasks should also be reduced and all the heavy computations must be offloaded to a cloud server.

5.5.4 Bandwidth usage

Most mobile applications (e.g. the Smart Shopping application) require an always on network connectivity for effective service delivery. This has the disadvantage that some applications will continue running in the background resulting in high bandwidth usage by a user. Due to that the network service providers bill users according to their bandwidth usage, applications with high bandwidth usage may lead to high costs being incurred by the user when using the service, leading to bad user experience. [48]

Applications which are designed poorly result in high bandwidth usage and this may be due to the following factors:

- Continuous downloading or transmitting data at small intervals, either for housekeeping, refreshing, new connection to same address or pinging purposes results in high bandwidth usage;
- Applications that store data on cloud computing platforms (e.g. the Smart Shopping service) would need to download data when needed and optimising this can be done by pre-fetching the data at the start;
- Heavy data download in poor network conditions results in the application service being unresponsive or pages loading slowly. [48]

Bandwidth values also depend on the quality of the network to which a service is connected. The measurement of the bandwidth usage for the Smart Shopping service is relevant to this study in order to have a measure of how the service will behave when being used. The Windows phone emulator was connected to the internal virtual Ethernet port of the computer on which it was hosted as shown in Figure 5.11 which was captured from the network adapter settings of the network sharing centre in the computer's control panel.

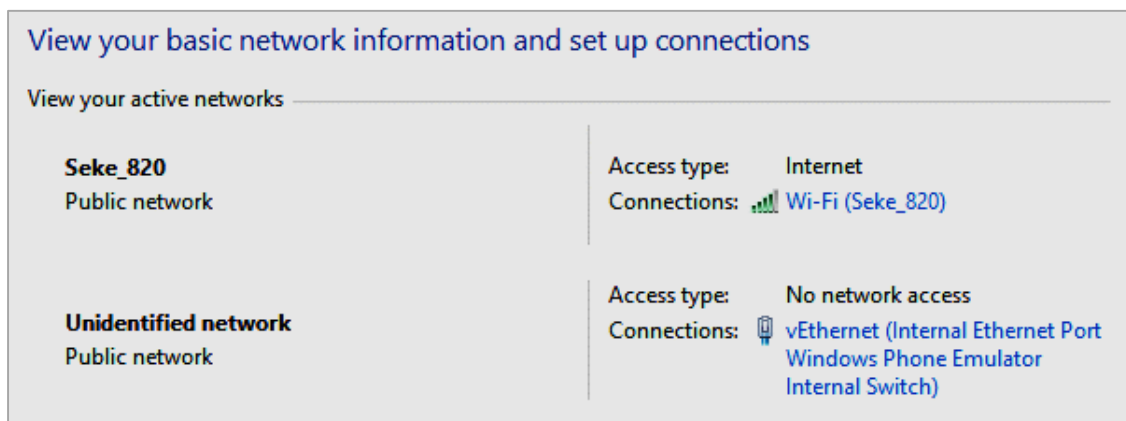


Figure 5.11 Connection of the Windows phone emulator to the virtual Ethernet.

The bandwidth was then tested for the data traffic of the Windows phone emulator being transmitted via the Hyper-V Ethernet adapter whilst the Smart shopping application was being used. Tests were done when the Store Categories page was being loaded as shown in Figure

5.12 and the rate at which the HTTP requests were being sent to the database to get the data was 32Kbps and the rate at which the data was being received was 40Kbps.

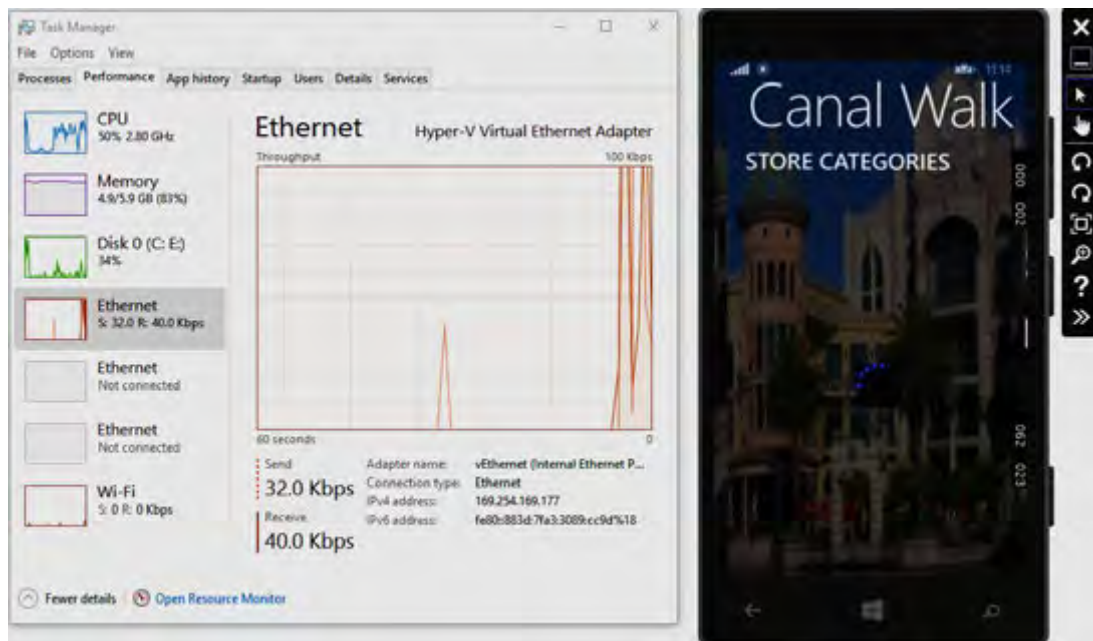


Figure 5.12 Bandwidth usage when loading the Store Categories page.

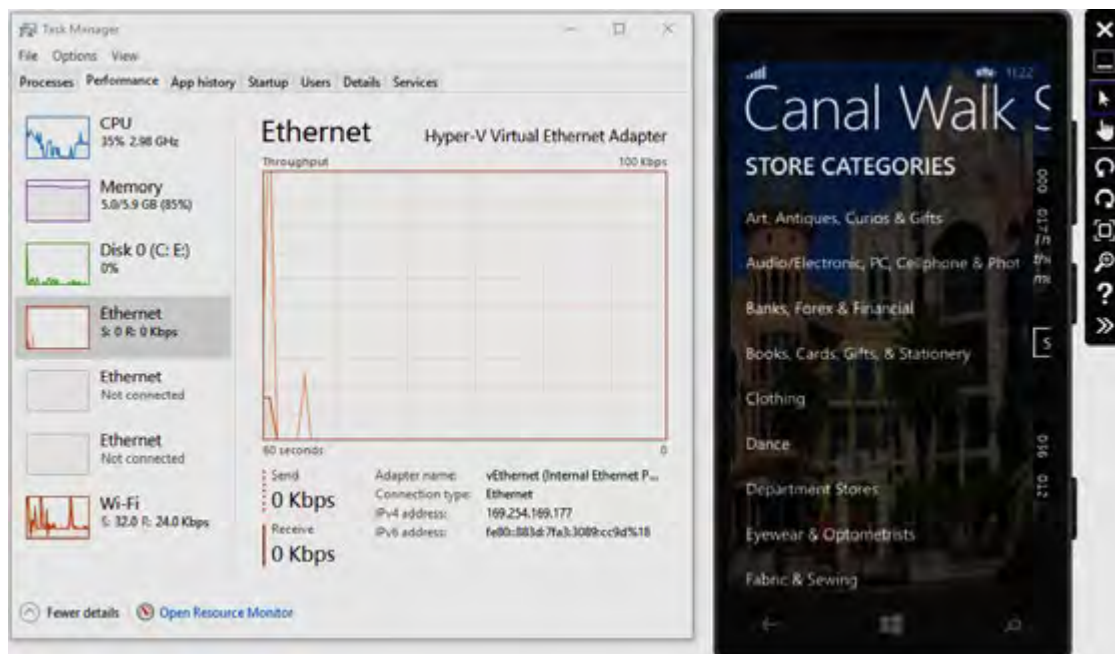


Figure 5.13 Bandwidth after loading Store Categories page.

After all the data had been received and the page was loaded, no data was either being received or sent as shown in Figure 5.13 with both upload and download data rates at 0Kbps.

The Store Category called *Banks, Forex and Financial* was then selected and bandwidth tests were done during the loading of the list of financial stores/banks in this category as shown in Figure 5.14. The rate at which HTTP requests were being sent to the database to get the relevant data was at 32Kbps and the rate at which the data was being received was also at 32Kbps.

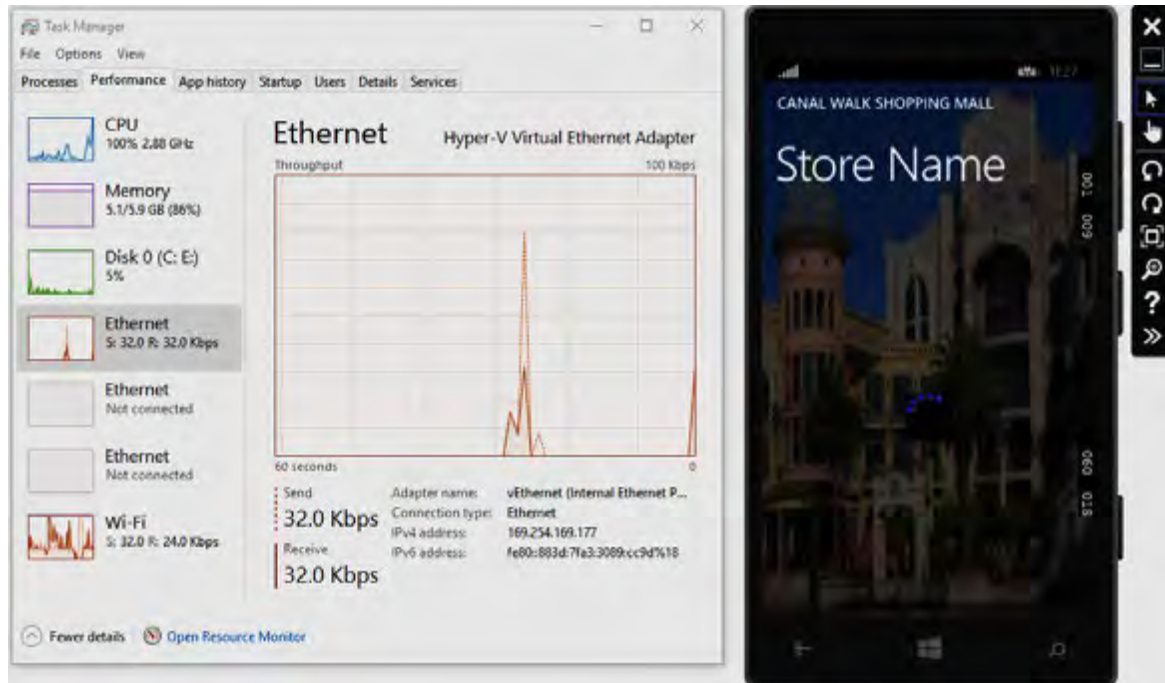


Figure 5.14 Bandwidth for loading the *Banks, Forex and Financial* Store Category list

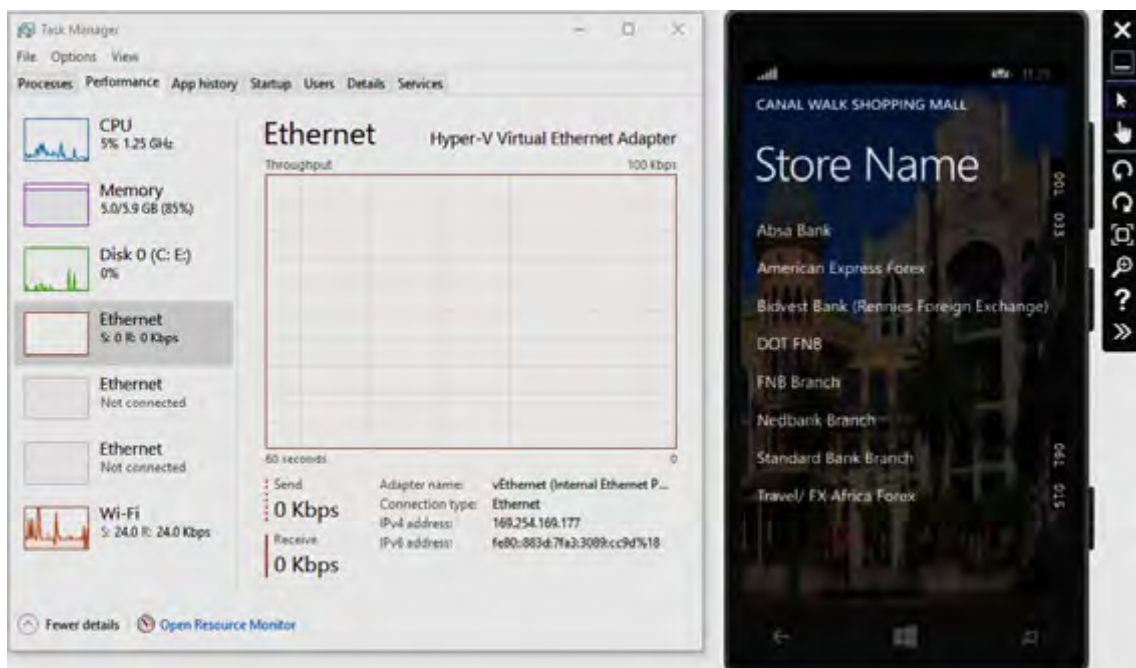


Figure 5.15 Bandwidth after loading the *Banks, Forex and Financial* Store Category list.

After all the data for the financial stores/banks in this list had been received and was loaded on the page, no data was either being received or sent as shown in Figure 5.15 with both upload and download data rates at 0Kbps.

The tests were not conducted on a physical mobile phone device because the tools that were available gave the overall bandwidth usage of all the applications that were installed on the mobile device and not for only the Smart Shopping service, which in this research is required.

5.6 Performance Testing Conclusions

The results obtained in the conducted performance tests showed that a real mobile device is the best method to be used to conduct performance tests for a mobile service than using mobile emulators. This is because of the following reasons:

- The mobile phone emulator took more time to start up than that for the physical mobile phone device and its performance was slower.
- The memory usage for the mobile phone emulator was higher than that for the physical mobile phone device.
- More battery energy was used by the mobile phone emulator during the data transmission than when the physical mobile phone device was used. More display energy is used up by devices with larger screens than with smaller ones.
- To reduce the energy consumption by the display screen, light colours must be avoided whenever possible for the mobile service design and this is due to that darker colours use less energy than light colours on the display screen.
- It is highly advised to do all data transmission via the Wi-Fi network as it consumes less energy than when the cellular network is used.

The overall performance of the Smart Shopping service met some of the requirements of standard Windows Phone applications and this was proven by the following results:

- The application *start-up time* for the Smart Shopping application 487.93 milliseconds when a physical mobile phone device was used and 1.37 seconds for the tests using a mobile phone emulator. *Parsing* the hub page for the Smart Shopping service using the mobile

phone took 13.83 milliseconds and 50.76 milliseconds when testing using the mobile phone emulator. This was lower than the typical allowable start-up time for a standard Windows phone application which is less than 5 seconds [45].

- The *maximum memory used* by the physical mobile phone device during the application analysis session was 20.8MB and that for the mobile phone emulator was 26.0 MB. By referring to Table 5.1, the Smart Shopping application used lower memory than the recommended capacity, hence it will be able to function well on both low and high end devices without straining the memory of the mobile device.
- The Smart shopping application stopped sending any data after the loading the data needed on a page as shown in Figure 5.13 and Figure 5.15 and no background data is transmitted hence this minimises the *bandwidth usage* hence conserves the mobile phone's battery.

In the next chapter, overall conclusions for this research are made and recommendations for future work to be done are presented.

6 Conclusions and Recommendations

This chapter presents the conclusions about this work and recommendations for future research to be done.

6.1 Overall Conclusions

This research delivers a solution on how to build a composed Smart Shopping service system that is hosted on a cloud computing platform and is accessed using a mobile phone. The research objectives were met and the research questions were answered as explained below:

- Service composition methods and mobile phone technologies were used to develop the composed Smart Shopping service system;
- Mobile cloud computing technologies were used to deliver the smart shopping mobile service;
- The tools that were used to develop the cloud based information service system assisting with the car parking task were the Microsoft Azure Cloud Computing Platform, Postman, Navicat and Visual Studio Community 2015.

6.2 Recommendations for future work

- ❖ Further work that can be done for this research is to design the Smart Shopping Service system for the Android and IOS mobile phone platforms. The data used for these additional platforms will be similar to that used in this research for the Windows platform. This can be modified by adding or removing some of the data in the SQL database, depending on the characteristics of the new application to be developed. When developing the Smart Shopping service for the Android platform, XML programming language is used to develop the user interface and Java programming language is used to develop the user interface logic and Business Layer. When this application is developed for the IOS platform, a Story Board is used to develop the user interface and Swift programming language is used to develop the user interface logic and the Business Layer.

- ❖ More services may be added to the composed service system in order to provide more services to be used. Examples of such services are as follows:
 - Car renting service;
 - Mobile phone online shopping;
 - Parking slot reservation.

- ❖ The actual parking lot layout for Canal Walk shopping mall must be used to develop the parking service of the Smart Shopping service.

- ❖ Sensors to be installed on all the parking bays and the data about the availability of parking spaces in the parking lot is to be sent to the database and then used in the development of the parking service of the Smart Shopping service.

References

- [1] J. J. P. Rodrigues, *Mobile networks and cloud computing convergence for progressive services and applications*, Information Science Reference, 2014.
- [2] T. Erl, *Service Oriented Architecture: concepts, technology and design*, Prentice Hall Press, 2005.
- [3] A. J. T. F. a. Y. Y. Dipanjan Chakraborty, "Service composition for mobile environments," *Journal on Mobile Network and Applications, Special Issue on mobile services*, vol. 10, no. 4, pp. 435-451, 2005.
- [4] M. K. a. B. A. S. Swaroop Kalasapur, "Dynamic service composition for pervasive computing," *IEEE transaction on parallel and distributed systems*, vol. 18, no. 7, pp. 907-918, 2007.
- [5] D. K. I. F. G. P. G. Kasitsaki, "Service Composition: State of the art and future challenges," in *Mibile and wireless communication summit*, Budapest, 2007.
- [6] Raikumar Buyya, *Mastering cloud computing*, Elservier Inc, 2013.
- [7] T. Erl, *SOA principles of service design*, Prentice Hall Press, 2007.
- [8] Canal Walk shopping Mall, "About canal walk," 2013. [Online]. Available: <http://www.canalwalk.co.za/about-canal-walk.htm>. [Accessed 23 February 2016].
- [9] century city, "Century city is booming with new development totalling R4,3 billion," 15 May 2014. [Online]. Available: <http://centurycity.co.za/news-articles/century-city-is-booming-with-new-development-totalling-r4-3-billion>. [Accessed 15 February 2016].
- [10] E. v. d. K. a. A. Verbraeck, *Designing mobile service systems*, Amsterdam: IOS Press, 2007.
- [11] E. v. d. Kar, *Designing mobile service systems*, Netherlands: IOS press, 2007.
- [12] D. Marinescu, *Cloud Computing: theory and practice*, Waltham: Elservier Inc, 2013.
- [13] D. Knott, *Hands-on mobile app testing*, Indiana: Pearson education Inc, 2015.
- [14] Google, "Google terms of service," 14 April 2014. [Online]. Available: <https://www.google.com/policies/terms/>. [Accessed 18 February 2016].
- [15] B. P. a. D. Mennie, "An architecture to support dynamic composition of service components," Carleton University, Ottawa, 2000.

- [16] IBM, "What is a smarter planet," IBM, [Online]. Available: <http://www.ibm.com/smarterplanet/us/en/overview/ideas/>. [Accessed 18 February 2016].
- [17] D. G. K. a. el., Internet of things, spaces, and next generation networking, Berlin: Springer, 2013.
- [18] T. r. a. o. Engineering, "Smart infrastructure: the future," The royal academy of engineering, London, 2012.
- [19] L. Kumar, "Mobile cloud computing," *International journal of research in information technology*, vol. 2, pp. 787-792, 2014.
- [20] M. R. Rahimi, "Mobile cloud computing: A survey, State of art and future directions," *Mobile networks and applications journal*, vol. 19, no. 2, pp. 133-143, 2014.
- [21] M. V. Barbera, "To offload or not to offload? The bandwidth and energy costs of mobile clou computing," in *IEEE Infocom*, 2013.
- [22] H. Suo, "Security and privacy in mobile cloud computing," in *9th International wireless communication and mobile cloud computing* , 2013.
- [23] J. Zhou, "Cloud architecture for dynamic service composition," *International journal of grid and high performance computing*, vol. 4, no. 2, pp. 17-31, 2012.
- [24] L. Qi, " A QoS aware composition method supporting cross platform service invocation in cloud environment," *Journal of computer and system sciences* , pp. 1316-1329, 2012.
- [25] J. Fu, "Virtual services in cloud computing," *IEEE 6th world congress on services*, pp. 464-472, 2010.
- [26] W.-T. Tsai, "An approach for service composition and testing for cloud computing," in *IEEE tenth international symposium on autonomous decentralised systems*, 2011.
- [27] I. s. association, "Standadisation: Living in the smart cities of the future: eWork, eMobility and connection to the smart grids," 04 April 2013. [Online]. Available: http://standardsinsight.com/ieee_international/smart_cities. [Accessed 29 February 2014].
- [28] N. Vinh-Tiep, "Smart shopping assistant: A multimedia and social media augmented system with mobile devices to enhance customer's experience and interactions," in *PACIS*, 2012.

- [29] C. -C. Chen, "Real-time smartphone sensing and recommendations towards context aware shopping," *Multimedia systems*, vol. 21, no. 1, pp. 61-72, 2015.
- [30] C. Tian, "Web based smart virtual shopping center and key strategy for implementation," *Journal of Shanghai Dianji University*, no. 5, pp. 253-257, 2010.
- [31] j. Lui, "Location technology-based mobile shopping service system," *International journal of hybrid information technology*, vol. 5, no. 4, 2012.
- [32] P. B. Avadhanam, "Wireless smart shopping framework". United States of America Patent US20130151335 A1, 13 June 2013.
- [33] T.-C. H. J. J. P. Chia-Chen Chen, "A smart assistant toward product-awareness shopping," in *Personal and ubiquitous computing*, Springer, 2014, pp. 339-349.
- [34] Microsoft Corporation, "Visual Studio," Microsoft Corporation, 2016. [Online]. Available: www.visualstudio.com. [Accessed 08 03 2016].
- [35] Lars Powers, Mike Shell, Microsoft Visual Studio 2015 Unleashed, Microsoft Corporation, 2015.
- [36] M. a. team, "Microsoft azure documentation," Microsoft corporation, 2016. [Online]. Available: <https://azure.microsoft.com/en-us/>. [Accessed 9 March 2016].
- [37] PremiumSoft TM CyberTech Ltd, "Products," PremiumSoft TM CyberTech Ltd, 2016. [Online]. Available: <http://www.navicat.com/products>. [Accessed 11 March 2016].
- [38] J.D Meier et al, "Microsoft mobile application architecture guide, Application architecture pocket guide series," Microsoft Corporation, 2008.
- [39] W. a. team, "Introducing the new hub control for heterogeneous data," Microsoft corporation, 5 February 2014. [Online]. Available: <https://blogs.windows.com/buildingapps/2014/02/05/introducing-the-new-hub-control-for-heterogeneous-data/>. [Accessed March 8 2016].
- [40] David Hale, "Hub control/pattern," Microsoft corporation, 25 February 2016. [Online]. Available: <https://msdn.microsoft.com/en-us/windows/uwp/controls-and-patterns/hub>. [Accessed 8 march 2016].
- [41] B. Tabor, "Understanding MVVM: ObservableCollection and INotifyChanged, Windows Phone 8.8 development for absolute beginners," Microsoft virtual academy, 24 june 2014. [Online]. Available: <https://mva.microsoft.com/en-US/training->

- courses/windows-phone-81-development-for-absolute-beginners-8375?l=AYmKjDIz_6104984382. [Accessed 7 march 2016].
- [42] V.-V. Helppi, "Rely only on real- Emulator vs devices.," 30 October 2013. [Online]. Available: <http://testdroid.com/news/rely-only-on-real-emulators-vs-devices>. [Accessed 20 December 2015].
- [43] M. V. Academy, "Working with the Wndows Phone 8.1 Emulator," Microsoft, 24 June 2014. [Online]. Available: https://mva.microsoft.com/en-US/training-courses/windows-phone-8-1-development-for-absolute-beginners-8375?l=aSNy9BIz_4304984382. [Accessed 02 February 2016].
- [44] C. M. Q. J. C. B. Arjun Anand, "A qualitative analysis of power consumption for location-aware applicationa on smart phones," in *Industrial Electronica, ISIE 2007. IEEE International Symposium on*, Vigo, 2007.
- [45] Jonathan Lyons, "Building high perfomance Windows phone apps," Windows, 25 January 2013. [Online]. Available: <https://blogs.windows.com/buildingapps/2013/01/25/building-high-performance-windows-phone-apps/>. [Accessed 14 December 2016].
- [46] M. Cooperation, "App memory limits for windows phone 8," Microsoft Cooperation, 2016. [Online]. Available: [https://msdn.microsoft.com/en-us/library/windows/apps/jj681682\(v=vs.105\).aspx](https://msdn.microsoft.com/en-us/library/windows/apps/jj681682(v=vs.105).aspx). [Accessed 31 March 2016].
- [47] Jonathan Lyons, "Optimising battery consumption of Windows phone applications," Windows, 17 January 2013. [Online]. Available: <https://blogs.windows.com/buildingapps/2013/01/17/optimizing-battery-consumption-of-windows-phone-applications/>. [Accessed 10 February 2016].
- [48] Rahul Bagaria, "Network monitoring for Windows Phone apps," Windows Cooperation, 20 December 2012. [Online]. Available: <https://blogs.windows.com/buildingapps/2012/12/20/network-monitoring-for-windows-phone-apps/>. [Accessed 31 March 2016].
- [49] IBM, "What is a smarter planet," IBM, [Online]. Available: <http://www.ibm.com/smarterplanet/us/en/overview/ideas/>. [Accessed 16 February 2016].

Appendix A

The parking Scenarios that were emulated in this research are as follows:

| SCENARIO 1 | |
|------------|---|
| P1 | G |
| P2 | G |
| P3 | G |
| P4 | G |
| P5 | G |
| P6 | G |
| P7 | G |
| P8 | G |
| P9 | G |
| P10 | G |
| P11 | G |
| P12 | G |
| P13 | G |
| P14 | G |
| P15 | G |
| P16 | G |
| P17 | G |
| P18 | G |
| P19 | G |
| P20 | G |
| P21 | G |
| P22 | G |
| P23 | G |
| P24 | G |
| P25 | G |
| P26 | G |
| P27 | G |
| P28 | G |
| P29 | G |
| P30 | G |
| P31 | G |
| P32 | G |
| P33 | G |
| P34 | G |
| P35 | G |
| P36 | G |
| P37 | G |
| P38 | G |

| SCENARIO 2 | |
|------------|---|
| P1 | G |
| P2 | G |
| P3 | G |
| P4 | G |
| P5 | G |
| P6 | G |
| P7 | G |
| P8 | G |
| P9 | G |
| P10 | G |
| P11 | G |
| P12 | G |
| P13 | G |
| P14 | G |
| P15 | G |
| P16 | G |
| P17 | G |
| P18 | G |
| P19 | G |
| P20 | G |
| P21 | G |
| P22 | G |
| P23 | G |
| P24 | G |
| P25 | G |
| P26 | G |
| P27 | G |
| P28 | G |
| P29 | G |
| P30 | G |
| P31 | |
| P32 | G |
| P33 | G |
| P34 | G |
| P35 | G |
| P36 | G |
| P37 | G |
| P38 | G |

| SCENARIO 3 | |
|------------|---|
| P1 | G |
| P2 | G |
| P3 | G |
| P4 | G |
| P5 | G |
| P6 | G |
| P7 | |
| P8 | G |
| P9 | G |
| P10 | G |
| P11 | G |
| P12 | G |
| P13 | G |
| P14 | G |
| P15 | G |
| P16 | G |
| P17 | G |
| P18 | G |
| P19 | G |
| P20 | G |
| P21 | G |
| P22 | G |
| P23 | G |
| P24 | G |
| P25 | G |
| P26 | G |
| P27 | G |
| P28 | G |
| P29 | G |
| P30 | G |
| P31 | |
| P32 | G |
| P33 | |
| P34 | G |
| P35 | G |
| P36 | G |
| P37 | G |
| P38 | G |

| SCENARIO 4 | |
|------------|---|
| P1 | G |
| P2 | G |
| P3 | G |
| P4 | G |
| P5 | G |
| P6 | G |
| P7 | |
| P8 | G |
| P9 | G |
| P10 | G |
| P11 | G |
| P12 | G |
| P13 | G |
| P14 | G |
| P15 | G |
| P16 | G |
| P17 | G |
| P18 | G |
| P19 | |
| P20 | G |
| P21 | |
| P22 | |
| P23 | G |
| P24 | G |
| P25 | G |
| P26 | G |
| P27 | |
| P28 | |
| P29 | G |
| P30 | G |
| P31 | |
| P32 | G |
| P33 | |
| P34 | G |
| P35 | G |
| P36 | G |
| P37 | G |
| P38 | G |

| SCENARIO 5 | |
|------------|---|
| P1 | G |
| P2 | G |
| P3 | G |
| P4 | G |
| P5 | G |
| P6 | G |
| P7 | |
| P8 | G |
| P9 | G |
| P10 | G |
| P11 | G |
| P12 | G |
| P13 | G |
| P14 | G |
| P15 | G |
| P16 | G |
| P17 | G |
| P18 | G |
| P19 | |
| P20 | G |
| P21 | |
| P22 | |
| P23 | G |
| P24 | G |
| P25 | G |
| P26 | G |
| P27 | |
| P28 | |
| P29 | |
| P30 | |
| P31 | G |
| P32 | |
| P33 | |
| P34 | |
| P35 | |
| P36 | |
| P37 | |
| P38 | |

| SCENARIO 6 | |
|------------|---|
| P1 | G |
| P2 | G |
| P3 | |
| P4 | |
| P5 | G |
| P6 | G |
| P7 | G |
| P8 | G |
| P9 | G |
| P10 | |
| P11 | |
| P12 | |
| P13 | |
| P14 | G |
| P15 | G |
| P16 | G |
| P17 | G |
| P18 | G |
| P19 | |
| P20 | |
| P21 | |
| P22 | |
| P23 | |
| P24 | |
| P25 | |
| P26 | |
| P27 | |
| P28 | |
| P29 | |
| P30 | |
| P31 | G |
| P32 | |
| P33 | |
| P34 | |
| P35 | |
| P36 | |
| P37 | |
| P38 | |

| SCENARIO 7 | |
|------------|---|
| P1 | G |
| P2 | G |
| P3 | |
| P4 | |
| P5 | G |
| P6 | G |
| P7 | G |
| P8 | G |
| P9 | |
| P10 | |
| P11 | |
| P12 | |
| P13 | |
| P14 | |
| P15 | |
| P16 | |
| P17 | |
| P18 | G |
| P19 | |
| P20 | |
| P21 | |
| P22 | |
| P23 | |
| P24 | |
| P25 | |
| P26 | |
| P27 | |
| P28 | |
| P29 | |
| P30 | |
| P31 | |
| P32 | |
| P33 | |
| P34 | G |
| P35 | G |
| P36 | G |
| P37 | G |
| P38 | G |

| SCENARIO 8 | |
|------------|---|
| P1 | G |
| P2 | |
| P3 | |
| P4 | |
| P5 | |
| P6 | G |
| P7 | G |
| P8 | G |
| P9 | |
| P10 | |
| P11 | |
| P12 | |
| P13 | |
| P14 | |
| P15 | |
| P16 | |
| P17 | |
| P18 | |
| P19 | |
| P20 | |
| P21 | |
| P22 | |
| P23 | |
| P24 | |
| P25 | |
| P26 | |
| P27 | |
| P28 | |
| P29 | |
| P30 | |
| P31 | |
| P32 | |
| P33 | |
| P34 | |
| P35 | |
| P36 | |
| P37 | |
| P38 | |

| SCENARIO 9 | |
|------------|---|
| P1 | |
| P2 | |
| P3 | G |
| P4 | G |
| P5 | G |
| P6 | |
| P7 | |
| P8 | |
| P9 | |
| P10 | |
| P11 | |
| P12 | |
| P13 | G |
| P14 | G |
| P15 | G |
| P16 | G |
| P17 | G |
| P18 | |
| P19 | G |
| P20 | G |
| P21 | G |
| P22 | G |
| P23 | G |
| P24 | G |
| P25 | G |
| P26 | G |
| P27 | G |
| P28 | G |
| P29 | G |
| P30 | G |
| P31 | |
| P32 | |
| P33 | |
| P34 | |
| P35 | |
| P36 | |
| P37 | |
| P38 | |

| SCENARIO 10 | |
|-------------|---|
| P1 | G |
| P2 | G |
| P3 | G |
| P4 | G |
| P5 | G |
| P6 | G |
| P7 | G |
| P8 | G |
| P9 | |
| P10 | |
| P11 | |
| P12 | |
| P13 | G |
| P14 | G |
| P15 | G |
| P16 | G |
| P17 | G |
| P18 | |
| P19 | G |
| P20 | G |
| P21 | G |
| P22 | G |
| P23 | G |
| P24 | G |
| P25 | G |
| P26 | G |
| P27 | G |
| P28 | G |
| P29 | G |
| P30 | G |
| P31 | |
| P32 | |
| P33 | |
| P34 | |
| P35 | |
| P36 | |
| P37 | |
| P38 | |

Appendix B

B1. Creating a SQL Azure database

Go to the Microsoft Azure website www.azure.com , go to the Portal and log on with your credentials getting you to an interface as shown in Figure B1 below. Click on the **+ NEW** icon.

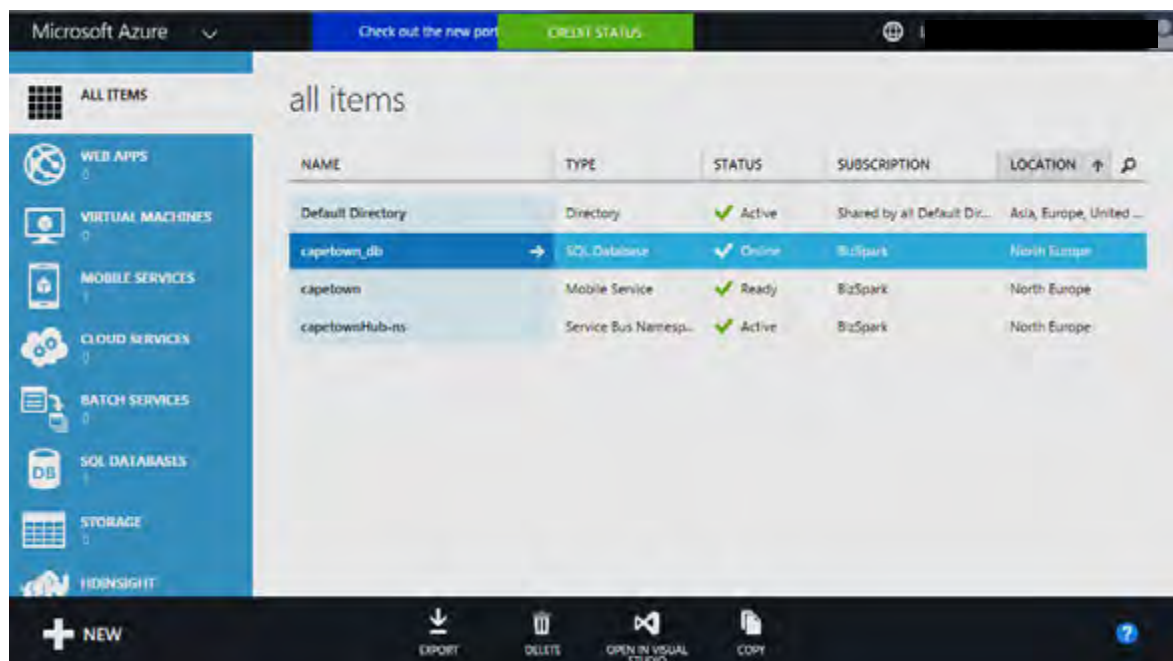


Figure B1 Azure portal interface

Click on Data services as shown in Figure B2. Select SQL database. Select quick create. Put a name that you want to call the database that you want to create, choose whether you want to store it on an existing server or you want to create a new one.

Select the region where you want the server to be sited. Type your login name and password. Finally click on the CREATE SQL DATABASE icon. After this step your SQL database will be created and stored in a server located in the region you would have selected.

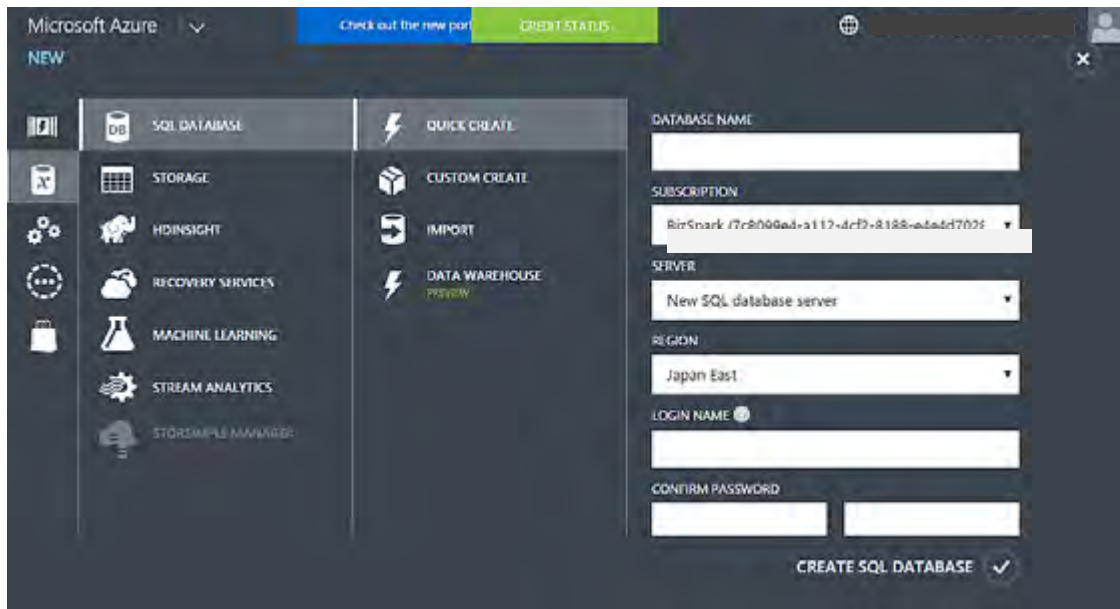


Figure B2 Setting up an Azure SQL Database.

B2. Creating an Azure mobile service

After logging on to the Azure website portal, Click on the **+ NEW** icon. Click on Compute. Select Mobile Services and then select Create as shown in Figure B3 below.

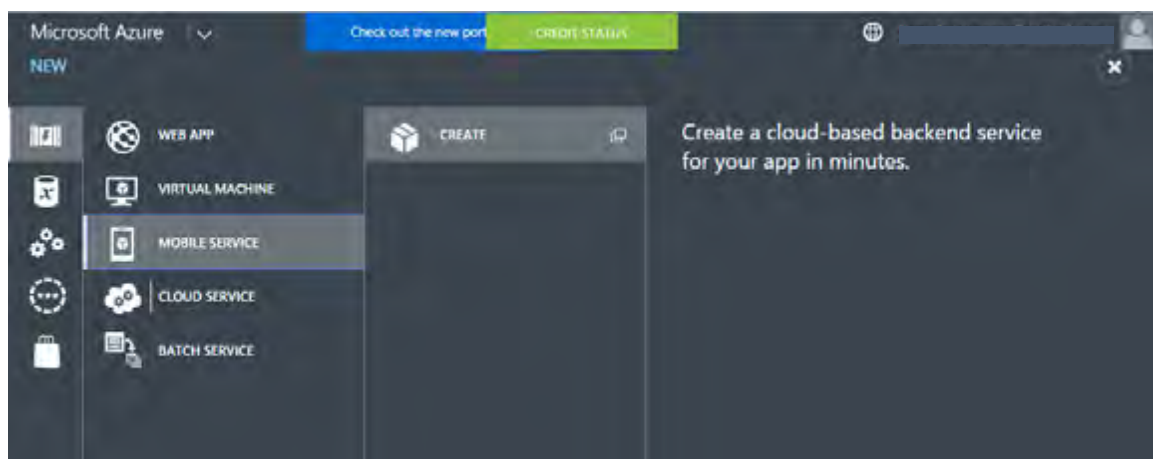


Figure B3 Setting up an Azure Mobile Service.

This will go to be page shown in Figure B4 below.

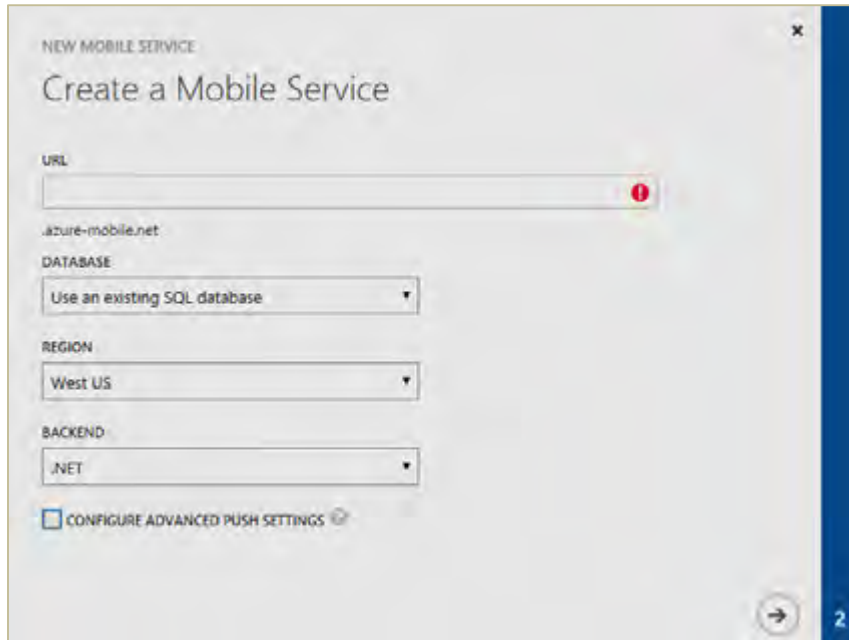


Figure B4 Azure mobile service details setup.

Type in the URL that you would want your mobile service to have. This will end with a **.azure-mobile.net**. Chose the database where you want to store the mobile service, the region where the database should be located and the type of backend required. This can either be .NET or JavaScript. Clicking on the arrow in Figure B4 will go to the page shown on Figure B5.

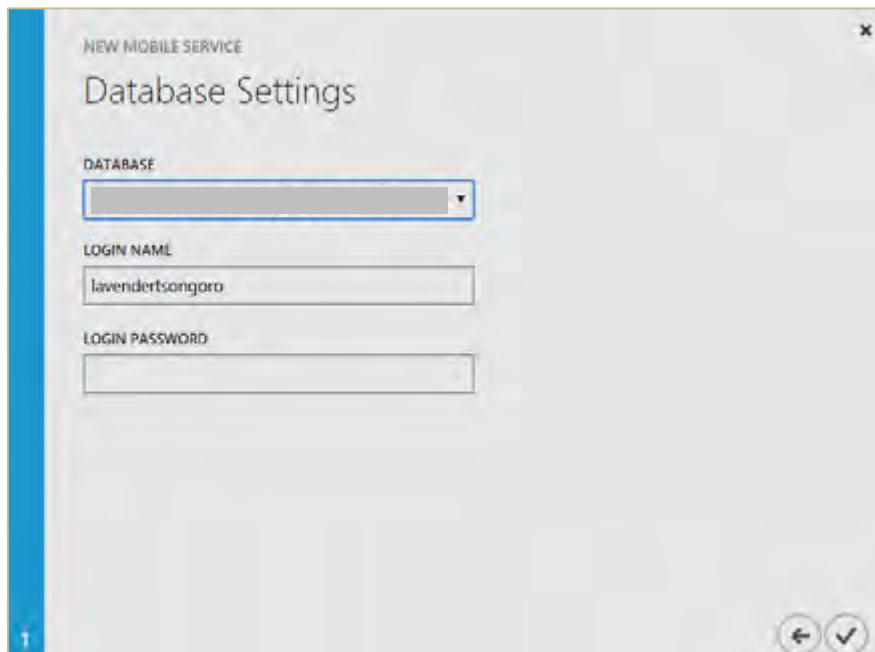


Figure B5 Mobile service user details.

Select the data base where you would want your mobile service to be stored then login with your username and password. Clicking on the icon with the tick then creates the mobile service.

B3. Connecting Navicat to Azure SQL database

Download the Navicat programming tool from www.navicat.com. This can either be the free version or you will have to buy one with more functionality.



Figure B6 Connecting to Azure SQL database from Navicat.

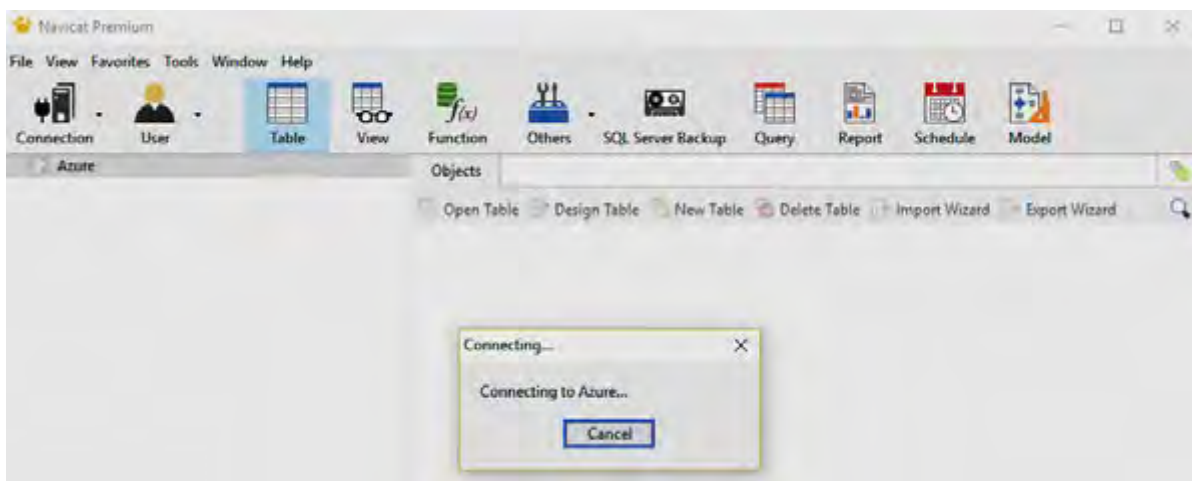


Figure B7 Navicat connection to Azure SQL database.

Put the details of your Azure database in order to connect, getting you to the interface shown on Figure B6. Click on Azure then click in Open Connection as shown below. Another interface will then pop up showing that Navicat is trying to connect to your Azure database as shown in Figure B7.

After Navicat connects with your database, all the tables and data stored there will then be shown on the interface as shown in Figure B8 below.

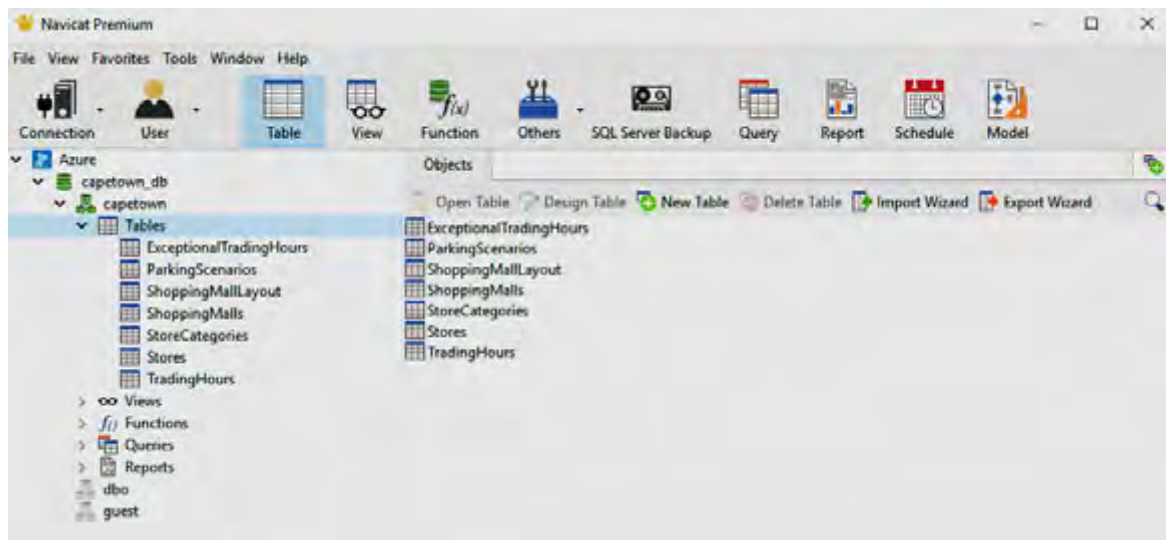


Figure B8 Tables in Azure SQL database shown in Navicat.