# Letter of Transmittal

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

20th October 2019,

Illawarra Multicultural Services Inc.

17, Auburn Street,

Wollongong, NSW 2500.

Dear Mr. Moisés Espinoza and Mr. Hussam Hattar,

The information in this document herein is being presented to you to formally present the technical documentation of this project. Should you consider any of these unsuitable, please do not hesitate to contact us.

This report will detail the whether or not the requirements have been met, system design, project closeout and system installation information.

We hope you find this report satisfactory.

Sincerely,

Zaima, Shalinda, Xiaoxuan, Zhiyu, Dinushi of CSIT321,

University of Wollongong,

Northfields Avenue,

Wollongong, NSW 2500.

**Technical Report**

“I For Work”

Group code : csit321sp19-ims01

Group name : Static Void

Producers:

Zaima Khandaker

Shalinda Fernando

Xiaoxuan Mo

Zhiyu Tong

Dinushi Silva

Recipients:

Illawarra Multicultural Services Inc.

Date:

7th June 2020

# Executive Summary

The aim of this project was to develop an application that would allow people from refugee and migrant backgrounds, that are having difficulties in finding employment, to have a platform where small businesses, who provide services such as cleaning, lawn mowing, gardening, carpet laying and removal, etc., can register their services, allow potential customers to request for their service and match the customer with a service provider based on the needs of the customer. The application was to be developed in both iOS and Android providing all the functionalities that would be required to register on the app, log in and use the service request functionality.

The app has been made using a PHP backend, a PHPMyAdmin database and has security features that involve using PHP data objects to prevent SQL injection attacks and MD5 encryption to protect users’ passwords. The system has a functional Android version; however, the iOS version is yet to be completed along with some additional functionalities of the application. The user interface as also changed slightly to accommodate for the changes in the programming of the application. The reason for the changes include, time constraints, resource limitations and current events that involve the COVID-19 pandemic and the university transitioning to an online mode of delivery.

The project closeout includes the identified positive and negative lessons learned throughout the project, which involve a need for better risk management and better communication between the team members, these lessons will ensure any future projects completed by the team don’t repeat any mistakes. A post project review has also been conducted which states that the final product is fit for purpose but will require more work. The transition phase involves the product being handed over to the clients, before the handover, the remaining functionalities need to be added and more testing needs to be conducted and after handover the system will need to be maintained and regularly updated.

Lastly, the system can be installed on a user’s computer for testing and demonstration purposes and can be installed from the Google Play Store once it is available for download so that the application can be used by the general public and end users as intended.

# Introduction

This report describes the initial scope of the system and agreed upon main and additional requirements, whether or not the system has met these requirements and what are the reasons for any changes in the system.

It also outlines the current state of the project and what languages, tools and architecture were used to develop the system. Since the project is now over, the project closeout goes into detail about the positive and negative lessons learned during the project, the post project review, how the system will be transitioned to the clients and what needs to be done before and after the handover.

Lastly, the system installation plan gives an idea on how the system can be tested and demonstrated on a user’s computer and how it can be downloaded from the Google Play Store by the general public when it is available for use.

# System Requirements

This section outlines the initial agreed upon requirements of the system which includes its main functionalities, additional functionalities. It also the user interface expectations of the project and the intended scope of the entire system.

# System Scope

People from the refugee and migrant background are having problems in finding employment even for low skilled work due to hindrances such as low English proficiency and discrimination from employers; hence they are more likely to start a small business of their own. The project intends to provide such sole trader business and individuals a platform where they can register and offer their services and for potential customers to be able to find and book their services. This project involves developing an app for both Android and iOS that will allow small businesses, who provide services such as cleaning, lawn mowing, gardening, carpet laying and removal, etc., to register their services, allow potential customers to request for a service and match the customer with a service provider based on the needs of the customer. Customers will be able to input their required service, their location and price range and the app will send out a notification to relevant service providers who can then respond with an initial quote for the service, customers would be able to select the service provider they want and this will create an engagement between the customer and service provider.

# Interface and Usability Requirements

The application is targeted towards an audience that may not necessarily have extensive knowledge on technology hence the application provides users with an interface that is simple and easy to navigate. This also includes having support for multiple languages with the default language of the application being English and using icons and labels that are universal and easily recognisable. The usability of the application must be efficient in that it must not take more steps than required for a user to achieve a task and the application must also be easy to learn how to use and be easy to remember how to use once learned.

# Main Functionalities

The main functionalities of the application begin with login and create account functionalities for customers and simply login for service providers. Once logged in, customers will have a function that will allow them to make a service request by inputting details such as location, desired service, desired date and time, and price range. Once the request has been sent and replied to, customers will be able to view the replies and price quotes from service providers and select the one they want. Customers would also be able to communicate with service providers, make payments for services, end an active service, rate, review or report a service, view history of all completed services and browse through all services available in their area.

Once logged in, service providers will be able to view service requests sent by customers in their area and accept or decline them. If accepted, service providers will be able to enter a price quote and get notified whether or not the customer has selected them for the service. Additionally, service providers will be able to list and edit their services and prices on their ‘profile’ page, communicate with customers, end an active service, rate or report a customer and view history of all completed services.

# Additional Functionalities

The additional functionalities for this application describe the functionalities that are not essential for the application to function but are desired features that would enhance the quality and usability of the application and will be implemented if time allows. Customers will be given the option to add images along with their review and service providers will have a gallery with images of their work on their ‘profile’ page, a functionality to check if there are any services have not been ended during the estimated time frame given by the service providers and to notify the users to end or update the active service, support for multiple languages and a translation function that would allow users to translate any English text descriptions written by other users.

Lastly, an additional requirement of this project that does not involve the development of the application is that a service provider registration form needs to be created and integrated with the IMS website to allow individuals and small business to register to offer their services on the application by providing their details and documents.

# Project Summary

This section summarizes to what extent the system has met the requirement detailed in the above sections and which specific requirements have been met, the requirements had been given priorities and the system aimed to meet the highest priority requirements. The system had also undergone some changes from its original plan, the changes and reasons have also been outlined along with where the project stands now. Lastly, a traceability matrix lists the systems main functionalities that had been tested and the outcomes of the tests.

* 1. How the System Met Requirements

The requirements have been prioritized in accordance with the MoSCoW rules of Dynamic Systems Development Method. MoSCoW stands for Must have, Should have, Could have and Want to have.

The prioritization levels are as such:

* Must have: All features categorised in this group must be implemented and if they are not delivered, the system would simply not function.
* Should have: Features of this priority are important to the system but can be omitted if deadlines cannot be met.
* Could have: These features enhance the system with functional items which can easily be reassigned to a later timebox.
* Want to have: These features only serve a limited group of users and are of little value in terms of functionality.

For each of the requirements identified below, it is indicated whether the requirement has been met or not.

|  |  |  |
| --- | --- | --- |
| **Requirement** | | **Priority** |
| Customer creates account #2 | | Must have |
| **Description**: The register account use case allows potential users to create an account and gain access to the platform's services | | |
| **Has the requirement been met?** | **Yes** | |

|  |  |  |
| --- | --- | --- |
| **Requirement** | | **Priority** |
| User Login #1, #3 | | Must have |
| **Description**: The user login use case describes the process of customer or service provider accessing to his/her account to using the application. | | |
| **Has the requirement been met?** | **Yes** | |

|  |  |  |
| --- | --- | --- |
| **Requirement** | | **Priority** |
| Make a service request #31 | | Must have |
| **Description**: This use case describes the process how customers make a service request and choose the service provider. | | |
| **Has the requirement been met?** | **Yes** | |

|  |  |  |
| --- | --- | --- |
| **Requirement** | | **Priority** |
| View and Select from available service providers #33 | | Must have |
| **Description**: Customer selects a service provider from the list shown in the system | | |
| **Has the requirement been met?** | **Yes** | |

|  |  |  |
| --- | --- | --- |
| **Requirement** | | **Priority** |
| View Reviews #39 | | Should have |
| **Description**: Customer can see the service provider reviews | | |
| **Has the requirement been met?** | **Yes** | |

|  |  |  |
| --- | --- | --- |
| **Requirement** | | **Priority** |
| Send price quote #35 | | Must have |
| **Description**: Service provider send a price quote for the service requests from the customer | | |
| **Has the requirement been met?** | **Yes** | |

|  |  |  |
| --- | --- | --- |
| **Requirement** | | **Priority** |
| See service requests #34 | | Must have |
| **Description**: Service providers want to see the service requests made in their area | | |
| **Has the requirement been met?** | **Yes** | |

|  |  |  |
| --- | --- | --- |
| **Requirement** | | **Priority** |
| End service #36 | | Must have |
| **Description**: Service providers and customers want to end a service once it is complete and proceed to payment. | | |
| **Has the requirement been met?** | **Yes** | |

|  |  |  |
| --- | --- | --- |
| **Requirement** | | **Priority** |
| View Transaction history #21, #25 | | Must have |
| **Description**: The user can view his/her transaction history and its detail. | | |
| **Has the requirement been met?** | **Yes** | |

|  |  |  |
| --- | --- | --- |
| **Requirement** | | **Priority** |
| Reset password #22, #26 | | Should have |
| **Description**: User can reset his/her password in the system. | | |
| **Has the requirement been met?** | **Yes** | |

|  |  |  |
| --- | --- | --- |
| **Requirement** | | **Priority** |
| Report user/service provider #24, #28 | | Must have |
| **Description**: Customer or service provider are able to report each other for any issue, and Admin views reports to take an appropriate action | | |
| **Has the requirement been met?** | **Yes** | |

|  |  |  |
| --- | --- | --- |
| **Requirement** | | **Priority** |
| Manage account #23, #27 | | Must have |
| **Description**: This use case describes the process of making changes in user’s account information and detail. | | |
| **Has the requirement been met?** | **Yes** | |

|  |  |  |
| --- | --- | --- |
| **Requirement** | | **Priority** |
| Translate #29 | | Could have |
| **Description**: This use case describes the process of making changes in application language. | | |
| **Has the requirement been met?** | **No** | |

|  |  |  |
| --- | --- | --- |
| **Requirement** | | **Priority** |
| Provide review #37 | | Must have |
| **Description**: Interface for writing reviews concerning service providers | | |
| **Has the requirement been met?** | **Yes** | |

|  |  |  |
| --- | --- | --- |
| **Requirement** | | **Priority** |
| Make Payment #105 | | Should have |
| **Description**: The register account use case allows potential users to create an account and gain access to the platform's services | | |
| **Has the requirement been met?** | **No** | |

|  |  |  |
| --- | --- | --- |
| **Requirement** | | **Priority** |
| Provide images with reviews | | Should have |
| **Description**: The customer may want to add images of completed services along with their reviews | | |
| **Has the requirement been met?** | **No** | |

|  |  |  |
| --- | --- | --- |
| **Requirement** | | **Priority** |
| Have gallery of images | | Should have |
| **Description**: Service providers can have a gallery of images on their profile showcasing their work to help attract customers | | |
| **Has the requirement been met?** | **No** | |

|  |  |  |
| --- | --- | --- |
| **Requirement** | | **Priority** |
| Translate descriptions | | Could have |
| **Description**: an option to translate service descriptions written by customers to the users preferred language for better understanding | | |
| **Has the requirement been met?** | **No** | |

|  |  |  |
| --- | --- | --- |
| **Requirement** | | **Priority** |
| Service not ended notification | | Could have |
| **Description**: A notification will be sent to users if they haven’t ended an active service by the estimated completion date. | | |
| **Has the requirement been met?** | **No** | |

|  |  |  |
| --- | --- | --- |
| **Requirement** | | **Priority** |
| Generate password #41 | | Could have |
| **Description**: An admin can generate password in database for qualified service provider | | |
| **Has the requirement been met?** | **Yes** | |

|  |  |  |
| --- | --- | --- |
| **Requirement** | | **Priority** |
| Ban/Unban Customers/Service providers #42 | | Should have |
| **Description**: An admin can ban or unban a customer/service provider account in response to reports | | |
| **Has the requirement been met?** | **No** | |

|  |  |  |
| --- | --- | --- |
| **Requirement** | | **Priority** |
| View active services #44 | | Should have |
| **Description**: Admins have access to a record of all currently active services | | |
| **Has the requirement been met?** | **Yes** | |

|  |  |  |
| --- | --- | --- |
| **Requirement** | | **Priority** |
| View customer/service provider reports #43 | | Should have |
| **Description**: Admins have access to a record of all reports made on customers and service providers | | |
| **Has the requirement been met?** | **Yes** | |

|  |  |  |
| --- | --- | --- |
| **Requirement** | | **Priority** |
| Generate earnings report #45 | | Want to have |
| **Description**: Use case generate reports about earning made from POD users, number of services, customers and service providers | | |
| **Has the requirement been met?** | **No** | |

|  |  |  |
| --- | --- | --- |
| **Requirement** | | **Priority** |
| Check if service providers are members of IMS | | Want to have |
| **Description**: A functionality to check if a registering service provider is a member of IMS. | | |
| **Has the requirement been met?** | **No** | |

* 1. Changes and Reasoning

The system had gone through some changes during development depending on factors that include time limits, problems with limited resources, group member inconsistencies and current events. The main change in the system is to do with the iOS version of the product, originally the system was to be made in Android and iOS however due to limited resources that are required for iOS development this could not be completed, the resources were procured later in the development stages which did not provide enough time for the iOS version to be completed which is why the development team decided to focus on the Android version to ensure at least one version of the system is fully functional.

The initially intended user interface design has also changed, the development team had found that some features needed to be coded in a certain way that did not fit the first iterations of the UI design, consequently the design had to be changed on some parts of the system.

The recent COVID-19 outbreak had misplaced many of the group members, some unable to return to Australia from China for a while and some having to leave Wollongong to get away from the highly populated metropolitan area for their own safety, this made it difficult for the group to meet as a whole, collaborate and communicate effectively.

* 1. Current Project State

This project was a large undertaking and could not be completed as initially intended. The Android version is functional and can be used to make service requests, accept them and use the main functionalities that is required for the application to work, some additional functionalities and stretch goals have not been completed, all of which are outlined in the traceability matrix below. The iOS version is not complete and functional and will require more work.

* 1. Traceability Matrix

|  |  |  |  |
| --- | --- | --- | --- |
| Requirement No | Requirement description | Test | Completion Status |
| 2 | Creates account | passed | completed |
| 1,3 | User login | passed | completed |
| 31 | Make a service request | passed | completed |
| 33 | View and select from available service providers | passed | completed |
| 39 | View reviews | passed | completed |
| 35 | Send price quote | passed | completed |
| 34 | See service requests | passed | completed |
| 36 | End service | passed | completed |
| 21,25 | View Transaction history | passed | completed |
| 22,26 | Reset password | passed | completed |
| 24,28 | Report user/service provider | passed | completed |
| 23,27 | Manage account | passed | completed |
| 29 | Translate | N/A | incomplete |
| 37 | Provide review | passed | completed |
| 105 | Make Payment | N/A | incomplete |
|  | Provide images with reviews | N/A | incomplete |
|  | Have gallery of images | N/A | incomplete |
|  | Service not ended notification | N/A | Incomplete |
| 41 | Generate password | passed | completed |
| 42 | Ban/Unban Customers/Service providers | N/A | incomplete |
| 44 | View active services | passed | completed |
| 43 | View customer/service provider reports | passed | completed |

1. System Design

The database designed for the system is relatively concise, all the relevant attributes are stored separately so that it is easy to retrieve and manage for maintenance personnel. We use PHP for the back end and Java for the entire system, PHPMyAdmin is used for the database, PDO (PHP Data Objects), that are an abstraction layer for databases, non-emulated prepared statements have been used to prevent the possibility of an SQL injection attacks. MD5 password encryption has also been used to secure the accounts of users.

Shared preferences, which are objects that point to a file containing key-value pairs, have been used to store username and password security token pairs. Once a user tries to log in, the hashed security token is decrypted and checked to see if the user exists in the database. Once a user logs in to the application, they will remain logged in until they press the log out button or clear cache.

Different classes in the code have been categorised by using different prefixes. The classes beginning with A belongs to Admin, the classes beginning with C belongs to Customer, the classes beginning with SP belongs to Service Provider, and the classes beginning with CM belongs to Common. The program files include text documents with this information as well.

* 1. Information Architecture

Model-View-Controller

The MVC pattern, which means Model-View-Controller pattern is a design pattern in which we connect the data model with the view through an interface that is called controller. The controller directly manipulates the data in its given model. That means the frontend and the backend of the App are completely decoupled from one and another.

Each controller is designed to receive data and send back appropriate information according to the data received.

We chose to use the MVC pattern because it allows multiple developers to work on different parts of the project at the same time, without impacting on anyone’s working. Also, when a user requests a URL, it populates a model that the user can work on. If he decides to modify it, the model will be returned back to the controller on some request and the controller can extract data from that model.

Model-View-View Model

The Model-View-View Model pattern creates a controller that allows the view to manipulate the data model. It’s the simplest way to connect a disjointed model and a view. Unlike the MVC, which is a controller, the MVVM acts as a binder that binds between the view and the model. Thus, it is designed to provide a direct communication between the view and the model.

Architecture Design

After the evaluation of our non-functional requirements namely usability and availability, and also taking into consideration our preference for a client-server solution, we decided to use a distributed, client-server design pattern. For our system we will be using a thin-client, two tier client server architecture.

We selected a client-server architecture for two main reasons, to meet the customers preferences for the system, and to better fit within our non-functional requirements.

Two tier client server architecture gives us the ability to hide and protect certain data and information on the storage and processing server, while also limiting who can access data on the App through the requirement of usernames and passwords. On the other hand, client-server architecture allows us to provide fast and easy access to data for those who need it and when they need it

As previously stated, we have selected to run a thin-client, two tier client server architecture. We have elected to use this server as the main website, IOS, Android App and database server allowing for storage, and processing to be complete on the server and the clients devices. The App is then accessible through any device with an internet connection and a web browser. This also makes accessing information very easy and very fast as all the client needs to do is display the App and the processing and storage will be complete on the main server.

1. Project Closeout

Once a project is finished, some tasks still need to be completed by the project manager that include identifying all the positive and negative lessons learned throughout the project, why some things went wrong, what can be learned from those events and how it can be improved in the future. A post project review is also conducted to determine if the system is fit for purpose and whether or not the project followed the initial plan. The transition phase goes into detail about how the system will be handed over to the clients and what needs to be done before and after the handover is complete.

* 1. Lessons Learned

Many valuable lessons had been learned by the team throughout the entire duration of the project, the most significant of which would be risk management. The team had done their risk management based on common issues that arise when managing and developing such a project, but the team had not planned for an event as unlikely as a global pandemic and their usual university routine being transitioned fully to online learning. The team had a difficult time coping with such an event and would ensure that in any future project, to plan for even the most unlikely events to guarantee that the development of a system does not stagnate or come to a complete halt.

The team had planned many elaborate functionalities for the system but have learned it is important to keep the minimum requirements of a project more realistically aligned to the abilities of the team members so that all requirements and expectations are met in the given time frame. Effective communication and collaboration are imperative in the success of a project and could have been improved throughout the development process.

Nevertheless, some lessons learned have been of a positive nature as well, the team now is able to develop mobile applications in different languages for different operating systems and have also gained knowledge on carefully crafting an effective user interface for the target audience. Secondly, a project is not only about the coding and development of the system, the importance of each piece of documentation that goes along with a project was identified, since detailed and constructive documentation ensures that a project is organised. All lessons learned during this project will remain with each team member so that any future project they take part in is fruitful and their performance in new team endeavours is productive.

* 1. Post Project Review

The post project review is made to determine whether or not the system fully solves the problem it was designed to solve, if and how the system can be taken be taken further and how the lessons learned can be applied to future projects.

It has been determined that the system does meet the main requirements identified early on in the project with the exception of some additional functionalities and the iOS version of the system. The error rate of the system is low, and the system does function as intended and is fit for the purpose it was made. The system is easy to use, the user manual is available in multiple languages to accommodate users with low English proficiency and can be well adjusted to meet future operating demands. If the system is found to have bugs in the future, the team must ensure that they are fixed, and ongoing management of the system is required to patch and update it as frequently as required for security and performance reasons.

The end result of the system does not fully match the original plan and expectation, but future iterations can ensure that it is equivalent to the initial plan. The clients of this project had the desire to launch the application to the general public for use; however, more work will need to be done for that to be possible which the team is willing to do to warrant the satisfaction of the clients. The system would require more testing, a larger database, a more refined user interface, the addition of the functionalities that it currently does not have and the completed development of the iOS version.

* 1. Transition Plan

Since the project has come to an end, it is time to hand over the system to the clients; however, before this transition can be made, there are some tasks that need to be completed. The system requires more testing that would emulate its usage by real customers and service provides to ensure all users can complete their intended tasks on the application. The system requires further development in terms of some additional functionalities that will make the application more attractive to users and will give it an equal standing to its competition application (AirTasker). Moreover, the system needs to be uploaded to the App Store so that end users are able to download and use it, the system currently only has a completed Android version which can be used but it will not reach the entire targeted audience until the iOS version is completed as well.

The development team intends to complete these tasks before handover to the clients to ensure their satisfaction and make certain that the product that is released with their brand name on it is high quality, preserving their good brand image.

After the handover, the application will need to be patched and updated often so that the security and performance is not compromised, if users find and report any bugs, those will also need to be rectified swiftly. The clients will also be handed over the database that goes with the application and admin privileges for the management of the application and its users.

1. System Installation Plan

The system installation plan outline two ways the system can be installed and use, one way is to use the system on your local computer, this method us suitable for testing, demonstration and to get a more technical overview of the program. The other way is to install the system from the Google Play app store once it has been submitted to be able to use the application for its intended purpose.

* 1. Local Computer

The system had been developed using Android Studio and a PHPMyAdmin database. To run the program on a computer, the program files, the database, Android Studio, a web server solution stack (WAMP Server, XAMPP Server) will be required. Firstly, a user must connect to their web server solution and import the database into PHPMyAdmin, then they can build and run the program on Android Studio. A user can either use the Android virtual device emulator to use the program or they can connect their Android mobile device to their computer using a USB cable and running the program on their device. It is recommended to run the program on an Android device to get the real feel of a mobile application. This method is suitable for testing and demonstration purposes since it will not allow the user to connect with a wide range of other real users that are offering services on the app.

* 1. App Store

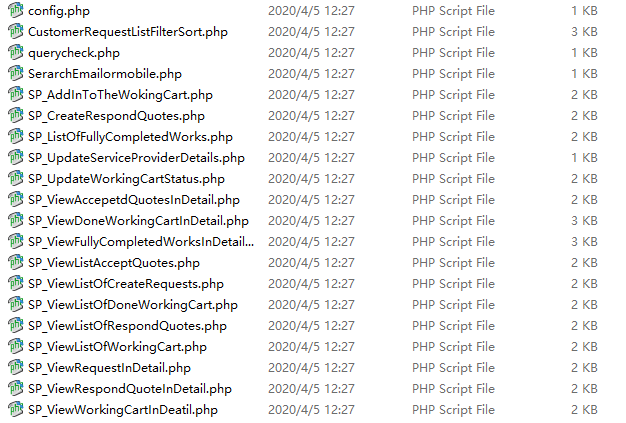
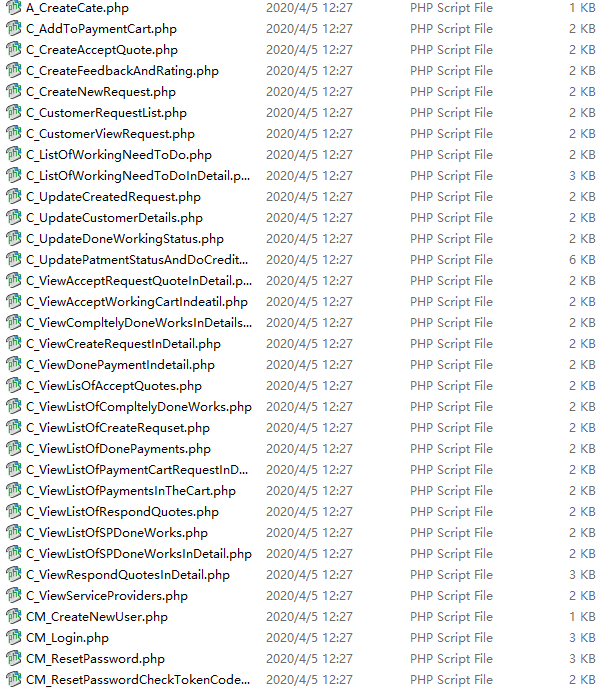
The system is to be submitted to the Google Play Store to allow anyone to be able to download it on their Android mobile device. The user would simply access the Google Play app on their phone, search for the application using its name “I For Work” and click ‘Install’. Once the installation is complete the user would be able to access the app to create an account and use its functionalities.

1. Conclusion

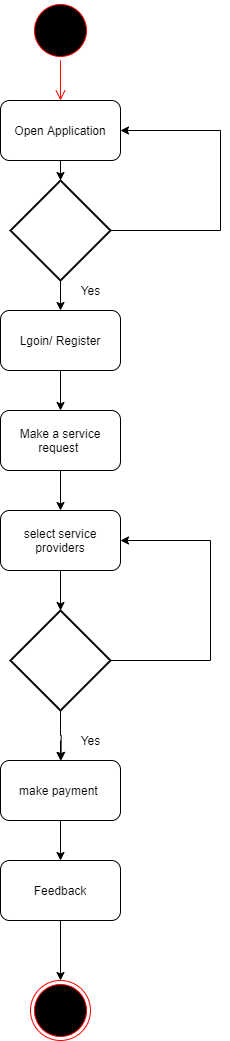
This project has come to an end with a complete part of the application that functions as intended and is able to support users creating accounts, logging in, using the service request functionalities and viewing their transaction history. The system still requires the completion of its iOS counterpart along with some additional functionalities. The occurrence of the COVID-19 global pandemic caused some issues in the successful flow of the project; hence, important lessons have been learned throughout the project life cycle, better risk management and communication being among the most vital. The system requires some tweaks and further refining, after which it will be ready to be handed over to the clients and used by the general public.

1. Appendix

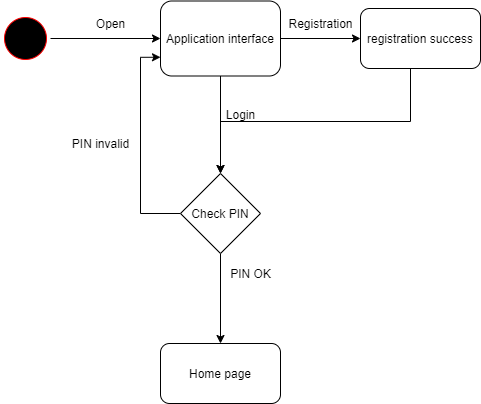
PhpMyAdmin Database



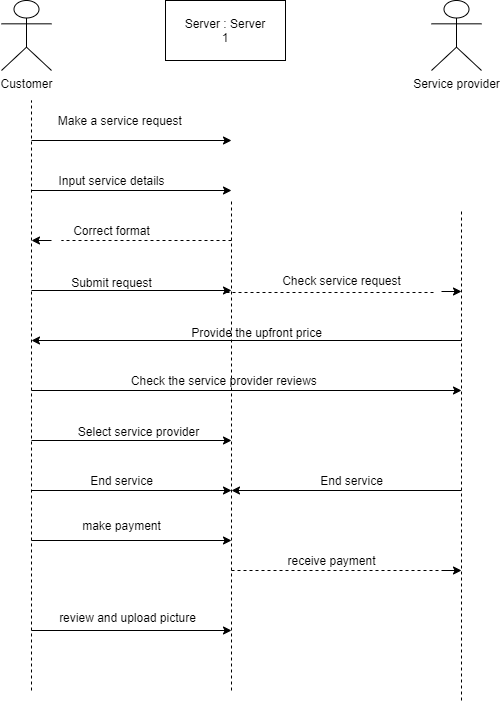
### General system design（Activity diagram）



### Login and Registration（State diagram）



### Make service request (Sequence diagram)



### Manage account (Sequence diagram)