### Design and Development of a Queue Management System for Soul Care Garden Hotel.

#### BY

**Isah**, Mubarak

#### BU/16C/IT/2360

FACULTY OF COMPUTER SCIENCE BAZE UNIVERSITY

#### ABUJA

May, 2020

**Design and Development of a Queue Management System for Soul Care Garden Hotel.**

## Thesis Submitted in Partial Fulfilment of the Requirement For the Degree of

B.Sc.

In

# Computer Science

## By

**Isah**, Mubarak To

## The Department of Computer Science Baze University, Abuja

May, 2021

###### DECLARATION

This is to certify that this Thesis/Report entitled [**Design and Development of a Queue Management System for Soul Care Garden Hotel**], which is submitted by [**Mubarak Isah** ] in partial fulfilment of the requirement for the award of degree for B.Sc. in Information Technology to the Department of Computer Science, Baze University Abuja, Nigeria, comprises of only my original work and due acknowledgement has been made in the text to all other materials used.

Date: 29/05/2021 Name of Student: Mubarak Isah

**APPROVED BY** ……………………………

**Head**

Department of Computer Science

###### CERTIFICATION

This is to certify that this Thesis/Report entitled **Design and Development of a Queue Management System for Soul Care Garden Hotel**, which is submitted by **Mubarak Isah** in partial fulfilment of the requirement for the award of the degree for B.Sc. in Computer science to the Department of Computer Science, Baze University Abuja, Nigeria is a record of the candidate’s work carried out by the candidate under my/our supervision. The matter embodied in this thesis is original and has not been submitted for the award of any other degree.

Date: Prof. Peter Ogedebe

###### APPROVAL

This is to certify that the research work, Dental Management System, and the subsequent preparation by Mubarak Isah with BU/16C/IT/2360 has been approved by the Department of Computer Science, Faculty of Computing and Applied Science, Baze University, Abuja, Nigeria.

By

Prof. Peter Ogedebe Date/Sign

1st Supervisor

Mr. Nasiru Aboki Date/Sign

2nd Supervisor

Dr. C. V. Uppin Date/Sign

Head, Department of Compuer Sciences

Prof. Mohammed Hammawa Baba Date/Sign

Dean, Faculty of Computing and Applied Science

Prof. Ahmed Baita Garko

External Examiner Date/Sign

###### DEDICATION

I dedicate this project to God Almighty my creator, my cherisher, my sustainer. I would also like to dedicate and thank all the people that supported me during the process of this project as well as during my stay in school. To my parents, who have supported me and have been my strongest pillar of support, to my aunt who has been there for me whenever I needed, to my grandmother, my Project supervisors for their guidance during the development of this project, lecturers, and my friends.

**ABSTRACT**

The proposed system is designed as an online solution to the long wait time and crowd with the pandemic we are facing now. The queue management system aims help service provider manage their customers efficiently. The system will be designed to ease customer flow management, which is useful for managers of the service provider. In addition, help provide the customer with clarity into hotel traffic and waiting time before leaving their destination. The app was developed using the waterfall methodology, was successfully tested, and produced desirable results. Development of the application will continue with for future update and further enhancements. Development Technology and Tools use in the building the Application is the Ionic 5 Angular7 with visual studio code and Firebase DB, which comprises of Typescript frame work, Ionic UI components and the reason for using this tools is that it is easy to learn, native Integration, hybrid application development and also works well with google API further more good for big data.

### TABLE OF CONTENTS

[**ABSTRACT 7**](#_bookmark1)

**CHAPTER 1: INTRODUCTION** ERROR! BOOKMARK NOT DEFINED.

* 1. OVERVIEW **ERROR! BOOKMARK NOT DEFINED.**
  2. BACKGROUND AND MOTIVATION **ERROR! BOOKMARK NOT DEFINED.**
  3. STATEMENT OF THE PROBLEM **ERROR! BOOKMARK NOT DEFINED.**
  4. AIM AND OBJECTIVES **ERROR! BOOKMARK NOT DEFINED.**
  5. SIGNIFICANCE OF THE PROJECT **ERROR! BOOKMARK NOT DEFINED.**
  6. PROJECT RISKS ASSESSMENT **ERROR! BOOKMARK NOT DEFINED.**
  7. SCOPE/PROJECT ORGANIZATION **ERROR! BOOKMARK NOT DEFINED.**

**CHAPTER 2: LITERATURE REVIEW** ERROR! BOOKMARK NOT DEFINED.

* 1. INTRODUCTION **ERROR! BOOKMARK NOT DEFINED.**
     1. SOULCARE GARDEN **ERROR! BOOKMARK NOT DEFINED.**
  2. HISTORICAL OVERVIEW **ERROR! BOOKMARK NOT DEFINED.**
  3. RELATED WORK **ERROR! BOOKMARK NOT DEFINED.**
  4. SUMMARY **ERROR! BOOKMARK NOT DEFINED.**

**CHAPTER 3: REQUIREMENTS, ANALYSIS, AND DESIGN** ERROR! BOOKMARK NOT DEFINED.

* 1. OVERVIEW **ERROR! BOOKMARK NOT DEFINED.**
     1. PROPOSED MODEL **ERROR! BOOKMARK NOT DEFINED.**
  2. METHODOLOGY **ERROR! BOOKMARK NOT DEFINED.**
     1. WATERFALL MODEL **ERROR! BOOKMARK NOT DEFINED.**
  3. REQUIREMENT ANALYSIS **ERROR! BOOKMARK NOT DEFINED.**
  4. REQUIREMENTS SPECIFICATIONS **ERROR! BOOKMARK NOT DEFINED.**
     1. *Functional Requirement Specifications* ***Error! Bookmark not defined.***
     2. *Non-Functional Requirement Specifications* ***Error! Bookmark not defined.***
  5. SYSTEM DESIGN **ERROR! BOOKMARK NOT DEFINED.**
     1. [*Application Architecture 38*](#_bookmark2)
     2. [*Use Case 41*](#_bookmark3)
     3. [*Activity Diagrams 43*](#_bookmark4)
     4. [*User Interface Design 43*](#_bookmark5)

**CHAPTER 4: IMPLEMENTATION AND TESTING** ERROR! BOOKMARK NOT DEFINED.

* 1. OVERVIEW **ERROR! BOOKMARK NOT DEFINED.**
  2. MAIN FEATURES **ERROR! BOOKMARK NOT DEFINED.**
  3. IMPLEMENTATION PROBLEMS **ERROR! BOOKMARK NOT DEFINED.**
  4. OVERCOMING IMPLEMENTATION PROBLEMS **ERROR! BOOKMARK NOT DEFINED.**
  5. USER GUIDE **ERROR! BOOKMARK NOT DEFINED.**
  6. *Testing* ***Error! Bookmark not defined.***
     1. *Test Plans* ***Error! Bookmark not defined.***
     2. *Test Report Summary* ***Error! Bookmark not defined.***

**CHAPTER 5: DISCUSSION, CONCLUSION, AND RECOMMENDATIONS..** ERROR! BOOKMARK NOT DEFINED.

* 1. OVERVIEW **ERROR! BOOKMARK NOT DEFINED.**
  2. OBJECTIVE ASSESSMENT **ERROR! BOOKMARK NOT DEFINED.**
  3. LIMITATIONS AND CHALLENGES **ERROR! BOOKMARK NOT DEFINED.**
  4. FUTURE ENHANCEMENTS **ERROR! BOOKMARK NOT DEFINED.**
  5. RECOMMENDATIONS **ERROR! BOOKMARK NOT DEFINED.**
  6. SUMMARY **ERROR! BOOKMARK NOT DEFINED.**

**REFERENCES** ERROR! BOOKMARK NOT DEFINED.

[**APPENDICES 1**](#_bookmark0)

### LIST OF TABLES

TABLE 1FUNCTIONAL REQUIREMENT SPECIFICATIONS **ERROR! BOOKMARK NOT**

**DEFINED.**

TABLE 2NON-FUNCTIONAL REQUIREMENT SPECIFICATIONS **ERROR! BOOKMARK**

**NOT DEFINED.**

### LIST OF FIGURES

FIGURE 1 USE CASE DIAGRAM **ERROR! BOOKMARK NOT DEFINED.**

FIGURE 2 ACTIVITY DIAGRAM **ERROR! BOOKMARK NOT DEFINED.**

FIGURE 1 ENTITY RELATIONSHIP DIAGRAM **ERROR! BOOKMARK NOT DEFINED.**

**LIST OF ABBREVIATIONS**

|  |  |
| --- | --- |
| CPU | Central Processing Unit |
| ERD | Entity Relationship Diagram |
| IT | Information Technology |
| DB | Database |
| UI | User Interface |
| QMS | Queue Management System |

# CHAPTER 1: INTRODUCTION

### Background of the study

In the world system, Nigeria to be specific, is one of the major country that is been faced with improper queue management. Queues (or waiting lines) help facilities or businesses provide service in an orderly fashion. Forming a queue being a social phenomenon, it is beneficial to the society if it can be managed so that both the unit that waits and the one that serves get the most benefit (Shukla and Shrivastar R.K., 2016). Efficient queue management is very essential anywhere in order to keep the people involved in the system to be on the move (Ajah, 2017). Queue Management has been an issue for a long time in many organizations especially in banks and government departments and is common to places that are overcrowded such as; markets, motor parks, banks, worship places, recreational centers, school and hotels. Due to the growing technology nowadays, it is not only important to provide convenience to the public consumers, but to help the person behind the counters to manage the existing queue so that the service will be delivered efficiently and effectively (Rushdi, 2013). The COVID-19 pandemic that has hit the Globe (world) made it mandatory to consider proper queue management in human daily activities,

organization, sects and so on. As overcrowded places can cause rapid spread of this disease. In the past six month of the country WHO declared a public health emergency of international concern, lots of countries went into a state of emergency leaving lots of establishment unmanned. In Nigeria today, we struggle to find resources to control the pandemic which has made the country to suffer setbacks. Some measures were set in place to control the spread of the COVID-19 pandemic by implementing social distancing, usage of a nose mask, and hand sanitizers, In other to manage the current state of the country, guidelines were given for some businesses to reopen by working short hours, controlling crowded environments which lead to the poor turnover for some businesses, especially markets, motor parks, banks, worship places, recreational centers, and hotels. This project work is focus on reducing the overcrowding places and aid social distancing by using virtual method to manage queue effectively. The proposed method uses mobile application software to implement a Queue Management System. This proposed method also help service provider sufficiently and manage customers. Queue Management software gives them the flexibility to help customers analyze the queue status virtually and the system puts the user in the queue and takes decisions on which customer to serve first and helps the customer receive appointment updates and alerts via their mobile devices.

### Problem of the statement

In our existing system all the booking of rooms in a hotel are done manually, so it was more time taking for an executive and customer to book, search rooms and collect payment in hotel. The need to build up this queue management system framework through inventive Development tremendously influenced the progression of this task. This task will plan and execute a reservation framework dependent on the queue management system. The system should be able to reserve a customer, initiate a checking for space available to take action of making a reservation for the

customer, if not send notification feedback to the customer through the application software and email. The expense of the establishment of any queue management system framework relies upon a few elements. To start with, the sort of customers wait time being minimize is of incredible thought the queue management system Provide customers with clarity into hotel traffic and wait times before they even step foot in your hotel, by doing so customers can search for the nearest 3 location with the shortest wait and can check-in virtually online, saving their spot in the queue, or schedule a future reservation.

### Aim and Objective of the study

The aim and the objectives (main function) of the project is;

1. To design and develop a queue management system for Soul Garden Hotel.
2. To analyse the method of Approach used in managing queue at soul care garden hotel.
3. To design a queue management system for soul care garden hotel
4. To develop and implement queue management system for soul care garden hotel.
5. To evaluate the performance, the queue management system.

### Motivation

The growing popularity and spread of smart phones and tablets have led to a change in how computer systems are designed. Technological developments have also made sure mobile devices are capable of helping in most sectors of our daily lives. Also the increase in COVID19 cases which has become a problem for customer service oriented companies Technology is a necessity in our lives and this project contributed to this factor whereby customers can receive appointment

updates and alerts sent to their phones. We have chosen the Android Operating System, as it is the most popular, ethical, user-controlled, operating system and it still is developing.

### Methodology

The project aims at designing an application to minimize the wait times of the user via an android application using a queue management system. The application is run on all android devices. The system will require mapping system, appointment, Customer Feedback, Alerting, identification, and internet connection. The Hotel Management System will allow remote access to hotel database only for customer after the authorization procedures. The customer could search, book rooms and give feedback.

### Significance of the project

The implementation of this project has the potential to benefit the academic field, the Local Society, Humanity, and the world as a whole. The knowledge to the academic field by providing safety to people being congested in a place with the pandemic the world is going through now, so far available solution to the pandemic is to minimize movement and control crowded areas. This project is not the first, but has more room for development and each part of the implementation of the project will be open for more development as technology advances. The contribution to the academic field is that doing things online help go back to our daily routine more safely and smartly, the pandemic is just another opportunity to help build new ways to cope in life. And it will give highlight for those who may develop interests to make a further study on similar topics.

### Project Risks Assessment

The project's risk assessment describes events that may be hard to control or slow down the workflow of this project. This section also includes precautions to be followed in case of such events.

**RISKS**

1. Inability to carry out research due to loss of hardware/software resources

* Be aware of and observe school IT security procedures
* Secure Android mobile phone when not in use.

1. Loss of work due to equipment failure /loss

* Weekly data backup to H drive and GIT HUB

1. Software availability (Unavailability of API’s)

* Alternative API’s will be checked for. Software requirements will be identified in good time for possible contentious API’s

1. Late delivery of hardware component

* Hardware requirements will be identified in good time to be able to order them in good time

1. Exams may halt work progress

* Accelerate workflow in order to finish before on time.

### Scope and limitation of the research

The project, which is in order of queue managing and controlling, is to manage the queue virtually using android application and sending feedbacks to the customer. The limitation of this project is the need for language translation and addition of voice translation for non-educated or unskilled customer.

CHAPTER 2 LITERATURE REVIEW

### Overview

This thesis is comprised of five chapters. The first Chapter provides an introductory knowledge of the project, as well as its scope and limitations. The historical background, review of past work and theory of the subject matter are covered in chapter two, chapter three describes the implementation, and tests, result and limitation are discussed in chapter four. Finally, a conclusion is reached in chapter five with recommendations for future work.

### Introduction.

This literature review was undertaken mainly to implement the use of Queue management systems, how they are structured, designed and implemented in many different organizations and how it can aid in solving queue management issue facing soul garden hotel (Ndung’u, 2018). To formulate another method which will proffer a suitable solution using the concept of queue management system. A queue management system is the organization of queues of people within a retail or public sector department. It can be either reactive through a system that can organize the existing queue or proactive through queue management statistics gathering system, so that the trends can be identified and anticipated. People that join queue in a standing line queue are direct to the next position by the system or be given issued with a ticket. With a ticketed system, customers are took out of the standing line queue, which can give comfort and less stress for the customers as well as their turns are not neglect. This queuing environment is an essential part of our daily lives and it is important for manufacturer to build the most cost-effective queuing solution (Uddin Md. et al, 2016).

### Review of past works

In controlling queue system many approaches or methods have been used in managing queue properly. Past works across this subject was reviewed, to achieve more functionality in the proposed method used in this project work. The past works review is as follows; in research (Ndung’u, 2018) A virtual queue management system was implemented using web application programming language (Such as; PHP, HTML, JavaScript, and CSS), MYSQL database for the webserver running the application, Wampserver. For web server, Firebase cloud storage for data exchange between the application and the web server, Android studio (API 7) for launching the application. The scope of the work is limited to only educated or skilled personnel’s alone. The research (Ndung’u, 2018) uses Intel Galileo gen-2, LCD (16\*2), PCF8574N, wire jumper for interconnection, trim pot 10k, DS1307RTC module, Plastic Container, 6\*6\*1 push button 4pin, screw PCB stand, Buzzer 6-12V, Resistor 10Kohms to implement an automated queue management system. This system has three Rules governing its operation, which are as follows; all service counters are multipurpose type, service counter only for customer choosing service A, each service has service counter. The system is restrictive to be able to real-world situations exactly (the theoretical solution may prove insufficiently informative to be useful). Then research (Hossain Md., 2011) is a microcontroller based electronic queue control system, which was implemented using 16F721C, a low-cost 8\_bit PIC microcontroller and entirely and entirely software controlled and these control programs have been developed using the PIC assembly language. This research work is not web based, so it requires the customer to be at the location. A smart queue management system for banking sector was designed in research (Jhala, 2012) with a small interface, easily accessible with smart phones for a queue management with SMS notification. It can be limited by network traffic. The research (Ngorsed, 2010) implementation

of Hospital Service Queue Management System with wireless approach was made, when customers, patients and stakeholder can access their queue remotely over the internet through a web application. In (Alias, 2007) research, a queue management system was designed and this was controlled by PIC 16F877, push button, decoder, latch, LED, seven segment, buzzer and voltage regulator circuit, the operational of this system 8 were programmed using the basic programing language. The research work (M.E. El-Nagger, 2010) described a methodology designed to support the decision-making process by developing seaport infrastructure to meet future demand. In order to determine an optimum number of berths at a seaport using queuing theory, the optimum number of berths that minimizes the total port costs can be decided. Then (Vasumathi.A, 2010) formulated a suitable simulation technique which will reduce idle time of servers and waiting time of customers for any bank having ATM facility. The research (S. A.

AL-Jumaily, 2011) have presented a new technique for queuing system called automatic queuing system. The proposed technique showed improvements in average waiting time. Also (Sundari, 2011) Have used the highly suitable modeling tool for MMC queueing model, the stochastic Birth-death Markov process and have eliminated the long waiting hours for customers. The (Patel, 2012) have discussed how queueing theory can be applied to a busy bank ATM and how the service should be improved so that the banks do not lose their customers. This research can help bank ATM to increase its QoS (Quality of Service), by anticipating, if there are many customers in the queue. The work at (S. K. Dhar, 2013) have discussed the application of queuing theory to the Bank ATM. This research can help bank ATM to increase its QoS (Quality of Service), by anticipating, if there are many customers in the queue. Because the bank can now estimate the number of customers waits in the queue and the number of customers going away each day. Then the research (S. Vijay, 2014) recommends changing the present queuing system

to alternate queuing system where the passengers do not need to wait so long. It was proved that this model of the queuing system is feasible and the results are effective and practical. Then, the research at (Ahsan, 2015) have proposed by shifting a server from day to night. From the performance, measurement of the proposed model the waiting time decreased expectedly at night shift than day shift. So this model can be used as an improvement technique for the service of the restaurant. In research we (Jhala, 2016) have described a methodology designed to support the decision-making process by the banks to meet the demand. In order to determine an optimum number of servers, queuing theory is applied. The Waiting and service Costs were determined with a view to determining the optimal service level.

### Basic features of a queue management system

1. Arrival of customers: It is a process of arrival for customers into the queue management System.

Classification of arrival of customers (Shastrakar, 2016) as Single line or multiple lines, Finite or infinite. Single customer or customers come in bulk. Arriving customers are totally under control or partially or no control, Deterministic or Probabilistic process, Empirical or a Theoretical Probability Distribution, Independent or conditionally dependent variables, Sometimes arrivals of customers is stationary.

1. Service Discipline: It works on the rule by which customers are selected from the queue for service. Rules are classified (Shastrakar, 2016) as: i. First-In First-Out (FIFO) ii. Last-In First out (LIFO) iii. Service for Random Order (SRO) iv. Priority Service. (PS) In queuing system (Alias, 20070, there are many types of queue model like SPF (Shortest Processed first), FIFO (First in First Out), SQ (Single Queue), Multiple Queues, diffuse Queue and Head of Queue. In (Alias, 2007) SPF, its more often used by the restaurant. SPF more accurately describes a model where

transactions of short expected length are dealt with sooner (as in a 10 items only queue in a supermarket). SPF can work well but is problematic if the customers do not perceive the right degree of fairness from the system. SPF works as (Md. Nasir Uddin et al, 2016) scheduling policy that choose lesser time execution to process first. In a supermarket, a specific paying counter only process transaction which customer buying goods less than 10 goods. SPF can work well because of its simplicity and minimizes the average amount of waiting time for each process. However, the setback is that it requires long time to complete if short process is continually add and customers do not perceive the right degree of fairness from the system. It is important to explain why customers are being serve in that order and ensure the understanding of customers to see the logic of this alternative approach. It is necessary to explain why customer are being served in an order other than simple First in First out (FIFO) and ensure that consumers understand and see the logic of this alternative approach.11 FIFO is the model of service provision (Alias, 2007) which is most fairly, where each customer is served in the order in which they registered for service. In service models where all customers think of themselves as equal, this model is important. The single queue is the familiar snake or corral queue format. Each person waiting is served in turn and the format of the queue discourages pushing in. It also provides visible reassurance to customers that they will be served in fairly and that the queue is progressing. For the multiple queues, this is the format that we have always seen at the supermarkets. Like a number of individual. Take-a-ticket queue models are example of diffuse queue. There is no formal queue line but customer register place in the process with a ticket (Mohamad Fazli Bin Alias, 2007). The head of queue, this is the place where the next person to be served waits in a single queue environment. It’s vital that they can see along the line of service position to avoid significant gaps in service provision (Shastrakar, 2016). With more than checkouts, this factor becomes increasingly significant. In order to maintain

fairness in the queuing system, there are the rules have been set for the Queue Management System concept: -  Queue must be fair.  Queues must be managed systematically and not allowed to descend into a free-for-all.  The process must include positive feedback of progress.  The process must be clearly identified; start & ends must be visible.  The perception of waiting time should be well managed.

##### Advantages of queue management system

In this system (Shastrakar, 2016). it provides many advantages to the customer service provider and the customer itself. The advantages are –

1. Staffs dealing with customers who have been well treated and get less stressed and this increases job satisfaction.
2. Staff can serve more customers per hour if those customers are fed to each counter efficiently.
3. Help the management by producing statistical reports on information such as arrival rates and patterns, waiting and service times, and default and reneging cases.
4. Enabling managers to monitor and set performance thresholds.
5. Customer will be treated with fairness and in more relaxed environment.
6. The queue flow will be smoother and increase the efficiency of queue management.
7. Nature of Customer As usual it is depending on the nature of arriving customers whether he is willingly accepted a waiting line or refuses it (Shastrakar, 2016). If the system is filled up to its capacity, then the arriving customer is naturally rejected. In some other cases if there is a rejection of the primary system, the customer accepted secondary system and ‘queue up’ in an informal

waiting line to enter in to the system. We note that there are mainly four items, which must be specified for any given Queuing System.

* 1. Balking: If the customer experiences that waiting time are very large as the queue is moving very slowly, the customers might balk and refuse to join the queue.
  2. Reneging: After joining the queue customer experience that it will take too much time to enter the system which is worthless then he customer reneges i.e. leaves the queue.
  3. Collusion: Several customers may cooperate and only one of them may stand in the queue to reduce the waiting time and buy the required service.
  4. Jockeying: If there is more number of queues, there is a way for customers to change the queue, which gives fast service than the other. In this process, the customer scans the lines for changing it.

##### Customer Flow Management

There are few manager (Md. Nasir Uddin et al, 2016) of service provider know what happens with their customers throughout the whole interaction process in real-time. If the supply and production managers know the flow of material and product in detail, why does not the head of operations know the flow of customers equally well? Is the flow of customers less important than the flow of material and products? Is the flow of customers more difficult to monitor and control? By knowing absolutely, the customers are a key success factor for any service provider that wants to be competitive. Therefore, Customer Flow Management (CFM) is managing the flow of customers and their experience from initial contact to final service delivery. The Customer Flow Management process consist of several phases including pre-arrival, arrival, queuing or waiting, serving, post

serving and managing customer service operation, the resulting framework for making informed business decisions and the frames the boundaries of CFM.

1. Pre-arrival CFM can start before the customers physically visits the shop or service center by implementing a method to book appointments before arrival. This reduces the time spent waiting by the customer and produces a positive impact on the customer’s service experience.
2. Arrival Customers need to be place in an appropriate queue on arrival. Customer Flow Management stresses the possibility of segmenting the customers in different queues rather than entering all customers in the same queue.
3. Queuing/waiting: Most customers will endure a period of waiting after queue entry. A balanced and controlled waiting period is the desired optimum results of any managers. No one (Md. Uddin et al, 2016) wants to have a completely empty waiting area as it reflected that you are overstaffed or impression of abandonment. Equally, too many customers waiting are simply as off-putting. In the case of a hospital or public service center, certain citizens might not accept it as they can demonstrate this during elections. CFM can help managers get the balance rightly by improving staff planning and by adding more flexibility to the process.
4. Serving: When calling the customer forward, staff can start preparations if the service chooses to identify and tracking customer’s history before the customer actually arrives at the service point.
5. Post-serving After a customer has been serve, a case handling function can continue to manage the case throughout its lifetime if needed and each step is document and process.
6. Managing Managers can use the gathered data in CFM process to evaluate the current

processes. Reports can be generating on employee-customer interactions, service times and customer wait times. Operational inefficiencies can be identified and addressed through process changes or training.

1. Service mechanism: The service mechanism is worked on the policy decided for the service facility for the customers who are serviced and leave the service system. Service mechanism follows single channel-single phase, single channel-multiphase, multichannel-single phase, and multichannel-multiphase.

F. Eisting Queue Management System There are many products (Md. Nasir Uddin et al, 2016) available on the market produced by many queue management Companies for delivering optimum customer service, which are Stand Alone Queue System and Centralized Control Queue System.

1. Stand Alone Queue System Stand Alone Queue System (SAQS) design based on First Come First Serve, FCFS queue model, where there is only one service counter operation. All customers will be managing at the single counter. This system operates by calling or displaying number in sequential or randomized order and the customers will be treating fairly. The SAQS is performing well in a single department, service operation environment such as clinic.
2. Advance Queue System Advance Queue System (AQS) based on SAQS design where additional service counters are add to give flexibility in queue system process. This system can support up to 32 service counters and additional of 60 counters. It also can provides useful queue features as well as comprehensive reports. Besides that, it allows manager of real-time monitoring status for the queue management analysis. AQS is performing well in bank, hospital or any organization that has multiple department service operation.
3. Centralized Control Queue System Centralized Control Queue System (CCQS) design is use for higher range of customers in different department. This system has the capability to support up to 20 departments which each department can have up to 32 service counters and 60 counters. CCQS is network compatible because each department has been located at different part of a building or even in geographical area. Thus, CCQS is connecting through LAN or Internet and it also provide real-time status monitoring.
   1. History of waiting lines theory

In Waiting Lines Theory (Shastrakar, 2016) a model is constructed and record the inter-arrival time of customers and time required service mechanism to complete the service. In 1909, Agner Krarup Erlang, father of Waiting Lines Theory, which is also called Queuing Theory, had its beginning in the research on the Waiting Line Theory. The first developers of Queuing Theory as applicable to the telephone industry were Tore Olaus and Erlang. Erlang experimented with fluctuating demand in telephone, traffic, later he published a report addressing the delays in automatic dialing equipment and its cost. Further, he was extended to problems that are more general and to business applications of the waiting lines. Engset’s formulations were not known until later because of the delay in publishing and traffic engineers to develop better systems first used Erlang’s model. Engset’s main work was not in Queuing Theory and traffic engineering and his contributions are not as well known. The Danish mathematician, Erlang developed models that accounted for callers that dropped due to frustration from waiting for an operator and those that were patient enough to wait for their call to be connected. Erlang (M/D/1) queuing model in 1917 and (M/D/K) queuing model in 1920.

### Characteristics of Queuing System

There are six items, which must be specified for any given queuing System.

1. Mean arrival time of customer, 2. Mean service time of server, μ 3. Customer’s behavior in the system 4. Capacity of the system 5. Number of service counters 6. The service rate is faster than arrival rate. Classification of Probabilistic Queuing Models
   1. Poisson-Exponential, Single server-Infinite population model (M/M/1:∞/FCFS)
   2. Poisson-Exponential, Single server-Finite population model (M/M/1: N/FCFS)
   3. Poisson-Exponential, Multiple server-Infinite population models (M/M/S: ∞/FCFS)
   4. Poisson-Exponential, Multiple server-Finite population model (M/M/S:N/FCFS) Other Queuing models are:
2. Poisson Arrivals and Erlang Service Distribution (M/ Ek / 1).
3. Poisson Arrivals and General Service Time Distribution (M /G/1)
4. Poisson Arrivals and Regular Service Time Distribution (M / D/1)
5. Constant Arrival Rate and Constant Service Rate (D/ D/1)

### 2.6 Summary

The literature review in this chapter proved that the implementation of and integration of QMS is severely underdeveloped. It shows that it is essential to have management systems present in- order to make this skill acquisition program more effective and efficient.

Therefore, it is vital to explore techniques that prove to offer an easy to use and reliable experience. Chapter 3 presents the requirement analysis and the methodology adopted in solving the problems highlighted in this chapter.

## CHAPTER 3: REQUIREMENTS, ANALYSIS, AND DESIGN

### INTRODUCTION

This chapter details the chosen methodology used in the project, and the approach chosen for the selected methodology. The tools and techniques used in the implementation of the project. The ethical considerations of this project are also observed in this chapter. The requirement analysis and specification of the project are also clearly stated in this chapter.

###### RESEARCH METHODOLOGY

The research methodology is consisting of different stages.

* + - The first stage involves that how many papers covers different areas where queuing system is used.
    - The second stage is concerned with establishing a classification scheme described.
    - The third stage involves distribution of papers over the years.

###### Search Strategy Search strategies and screening process follows the following steps: Sources of information

* IEEE Xplore ([http://ieeexplore.ieee.org](http://ieeexplore.ieee.org/))
* Science Direct ([www.sciencedirect.com](http://www.sciencedirect.com/))
* ACM Digital Library([www.acm.org/dl](http://www.acm.org/dl))
* WileyInterscience(<http://www.interscience.wiley.org/>)
* [www.ijstr.org](http://www.ijstr.org/)

###### STUDY SELECTION

The search engine in the above digital database hits quantity of studies, articles. Research papers posted by using journals, conference complaints and workshops are concept to be worthy and dependable. Keyword based totally search is hired to pick out the maximum relevant works. The keywords used are “Queuing machine”, “Banking”. The criteria used for exclusion of a research paper consist of unpublished papers, non-English papers, text-books, Master and Doctoral dissertations, non-peer-reviewed papers. In end result indicates the distribution of paper from 2000 to 2020. Table 1 suggests the described search method and variety of results received. From the returned studies, first beside the point research are excluded on the premise of title. Certain studies could not be envisioned from the identify, after which their summary is considered. If even abstract is not obvious then after analyzing the total textual content of papers, irrelevant research is excluded. In some library seek when large quantity of studies again then follows a few superior seek.

###### TABLE 3.1 SEARCH SELECTION

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  | Excluded | | |  | |
| S.  no. | E-source | Studies returned | based on title | based on abstract | based on full text | Keyword used | |
| 1 | [http://ieeexplore.ieee.org](http://ieeexplore.ieee.org/) | 83 | 107 | 4 |  | Queuing Banking | system, |
| 2 | [www.sciencedirect.com](http://www.sciencedirect.com/) | 473 | 1237 | 5 | 5 | Queuing Banking | system, |
| 3 | [http://dl.acm.org](http://dl.acm.org/) | 307 | 298 | 2 |  | Queuing system, Banking,  Customer management | |
| 4 | [www.interscience.wiley.com](http://www.interscience.wiley.com/) | 6 | 3 | 1 |  | Queuing Banking | system, |

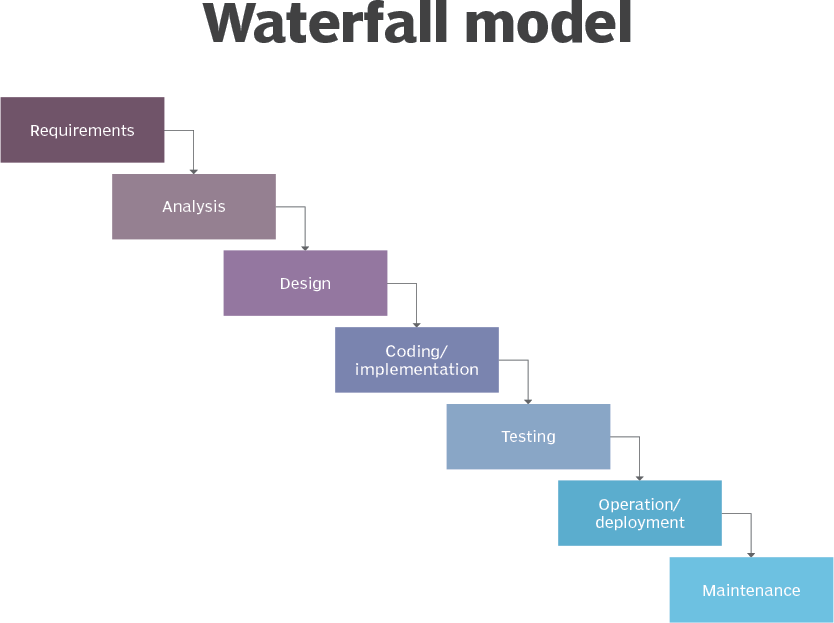
***Table 3.1 is showing the search selection***

###### PROPOSED MODEL

The software application to be developed will be run on android devices and can be downloaded through the Google Play Store. As stated before, the main function of this application is to design a queue management system for a hotel system. The Hotel queue management application will allow users to have access to the map location of the nearest branch to them and get the traffic status of the branch they intend to go with the user of an android smartphone and an iQueue application. Users will access to Soul care garden hotel through iQueue.

###### METHODOLOGY

The best methodology for the development of this project would be the waterfall model, Waterfall model is preferable as it has provision for changes and the changes can be implemented in the maintenance phase. This is because the waterfall model is simple and easy to understand and use for the developer and the other users. This model also allows for early design changes and places emphasis on requirement and design before writing any single line of code which ensures minimal time wastage and effort in design changes. It is really easy to use, handle, and best for projects of a such as this one. The waterfall model consists of seven developmental stages that are executed in a sequential order, which implies that a preceding stage needs to be completed before moving on to the next phase of development. The figure 3.1 below represents the waterfall model along with all its seven stages.



***Figure 3.1 system Methodology***

The seven developmental stages in Waterfall model are:

1. **Requirement Gathering and analysis:** This phase involved performing a feasibility study and ascertaining all possible requirement of the queue management system.
2. **System Design:** This phase involves getting specifications, functional and non-functional requirements of both software and hardware required to develop the system. This step also involved defining the system architecture.
3. **Implementation/Coding:** With outcome from system design of the developed system prototype, the system was developed in small programs called units, which were pushed to the next stage to be tested.
4. **Integration and Testing:** All the units developed in the implementation were integrated into the system, isolated and fully tested. Once verified to be working, the entire system was tested to detect any undetected failures or faults.
5. **Operation/Deployment of system:** Once testing was done, the developed system was deployed and tested with live customer data.
6. **Maintenance.** Maintenance of the developed system was to be done in continuum to ascertain that all functional and nonfunctional requirements were reached and any changes to the software and hardware would not affect the customer environment.

###### SYSTEM DESIGN

System design is an important part of any system that is worth implementing. In this section, we shall look into the hardware, functional and non-functional system requirements, data aspect and process design for the Queue Management System (QMS).

###### HARDWARE REQUIREMENTS

Consist of hardware requirements to be met in order to successfully run the system:

###### Desktop Computer

To be used for design and development of the system. The computer had these specifications:

* + - * Intel® Core(TM)i3-4000MCPU@ 2.40GHz 2.40GHz
      * 4.00GB RAM
      * 500 GB Hard Disk Drive
      * Windows 10 pro.

###### Tablet or Smart Phone

It was to be used for registering and login in users on mobile, joining nearby queues and sending notifications in real time.

###### SOFTWARE REQUIREMENTS

To successfully run the system, there are a number of software requirements had to be met which were:

1. **Operating System:** Windows 10 Pro versions of OS (x64 bit) was used for the project work.
2. **Database Management System:** A database to store the details of various room numbers, the available ones and booking space. The following data based was to be used for the web server and the mobile-based application.
3. **MySQL Database:** For the Web server running the application.
4. **Android Ionic:** For the running the android application.
5. **Laravel:** Has been used as a web server.
6. **Android Studio:** At least API 7 or above.
7. **Programming Languages:** PHP, HTML and CSS

The mobile application is written with ionic. Ionic Framework is an open source UI toolkit for building performant, high-quality mobile using web technologies such as; HTML, CSS, and JavaScript with integration's for popular front-end frameworks like Angular, React, and Vue. The mobile application use API (Restful) as server for a means of communication with database for storage and exchanging data. That Representational state transfer API (RESTful) is written with laravel. Laravel is a free, open-source PHP framework, created by Taylor Otwell and intended for the development of web applications following the model–view–controller architectural pattern and based on Symfony.

###### REQUIREMENT ANALYSIS

A breakdown of the requirements of the mobile application and the classification of each requirement falls under.

###### FUNCTIONAL REQUIREMENT SPECIFICATIONS

The functional requirements of a system are the descriptions of what the system should do the services that it provides and the limitations on its operation. These requirements reflect the needs of customers for a system that serves a certain purpose such as status check for availability, queue reservation, or requirement elicitation. Having the functional requirements in mind, we define use- cases for better guiding the interfaces development from a user perspective and proceed with a general use-case diagram for the overall picture.

###### TABLE 3.2: FUNCTIONAL REQUIREMENT SPECIFICATIONS

***Table 3.2 is showing the functional requirements of the proposed application, iQueue. The***

***requirement numbers and descriptions have been shown as written above.***

###### NON-FUNCTIONAL REQUIREMENT SPECIFICATIONS

A non-functional requirement is a description of how the system should behave under certain conditions or input from the user.

|  |  |
| --- | --- |
| **Non-Function Requirements** | **Non-Function Category** |
| The Queue Management system must be web accessible, and run on the cloud | Deployment |
| The email alert must go out within 5 minutes of the ticket being created or closed | Operability |
| The email must have a reliability of 99.99% that it will be sent out | Performance / Reliability |
| The Queue Management system man must support both English and Hausa interfaces | Localization |

###### Table 3.3: Non-Functional Requirement Specifications

|  |  |
| --- | --- |
| **Req. No.** | **Description** |
| R-101 | When launched, the application shall stay running unless there is an  intentional shutdown of the application or the platform. |
| R-102 | The application will have a Face book, Google account signup by default. |

|  |  |
| --- | --- |
| R-103 | The application will display a message if the connection to iQueue fails. |
| R-104 | The application will boot within 5 seconds. |
| R-105 | The iQueue will remain on unless the user turns it off. |
| R-106 | The iQueue will boot into a command line and run the queue  management process. |

***Table 3.3 is showing the non-functional requirements of the proposed application iQueue. The requirement numbers and descriptions have been shown as written above.***

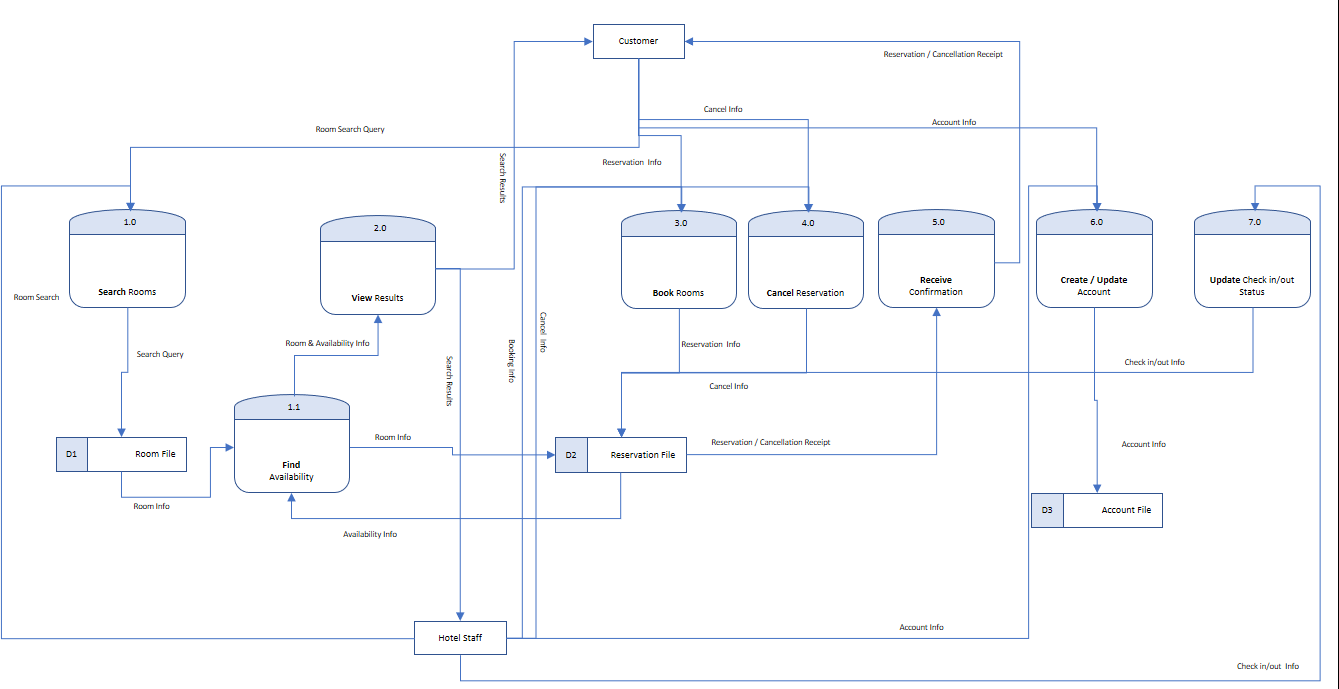
###### SYSTEM DESIGN

This system is comprised of three major parts, The Queue Management System, the Reservation System, and the Search Engine Optimization will guide you to achieve the reservation and securing you a spot at the hotel and it gives you feedback which will send out an email with the confirmation that your reservation has been secured.

###### SYSTEM/APPLICATION ARTIFACTS

This section shows the various System and Application artifacts used in the development of this application.

* + 1. APPLICATION ARCHITECTURE

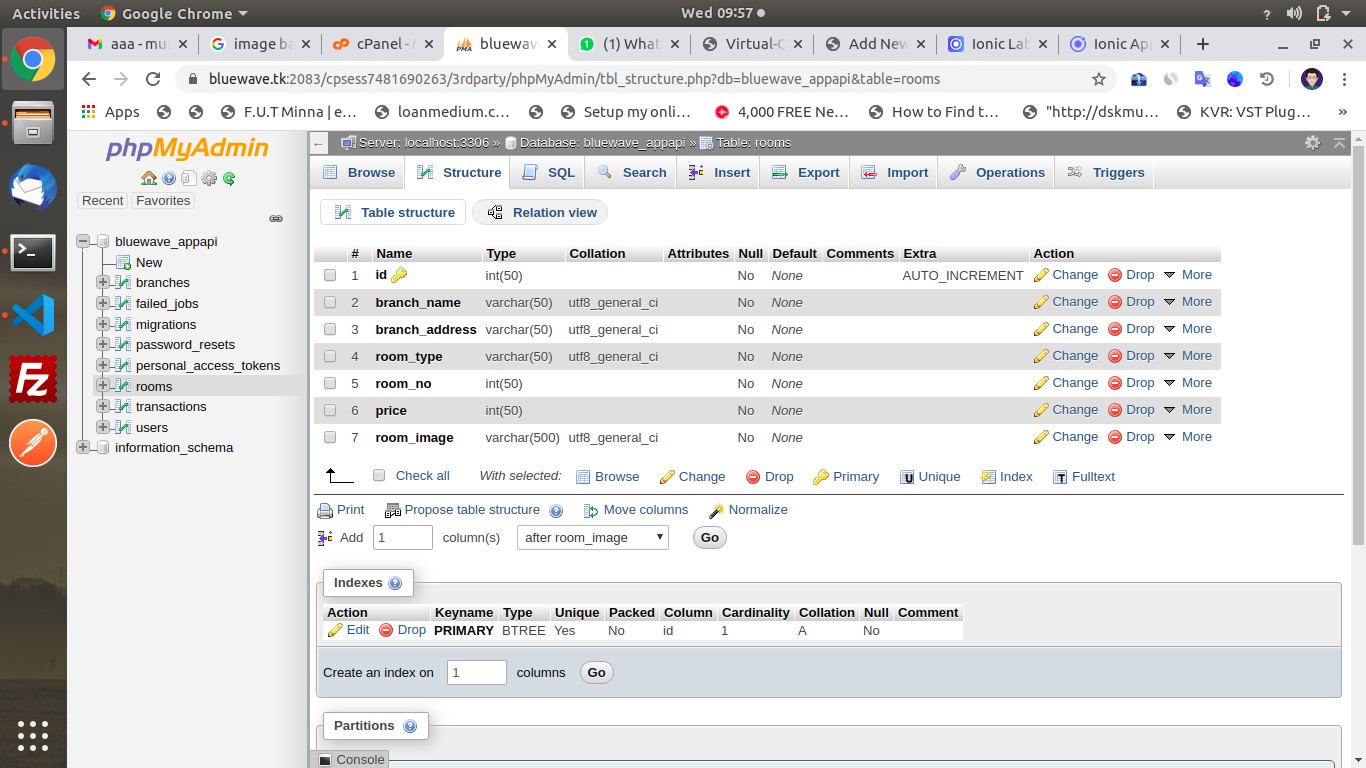


*Figure 3.2: Application architecture for Home Page.*

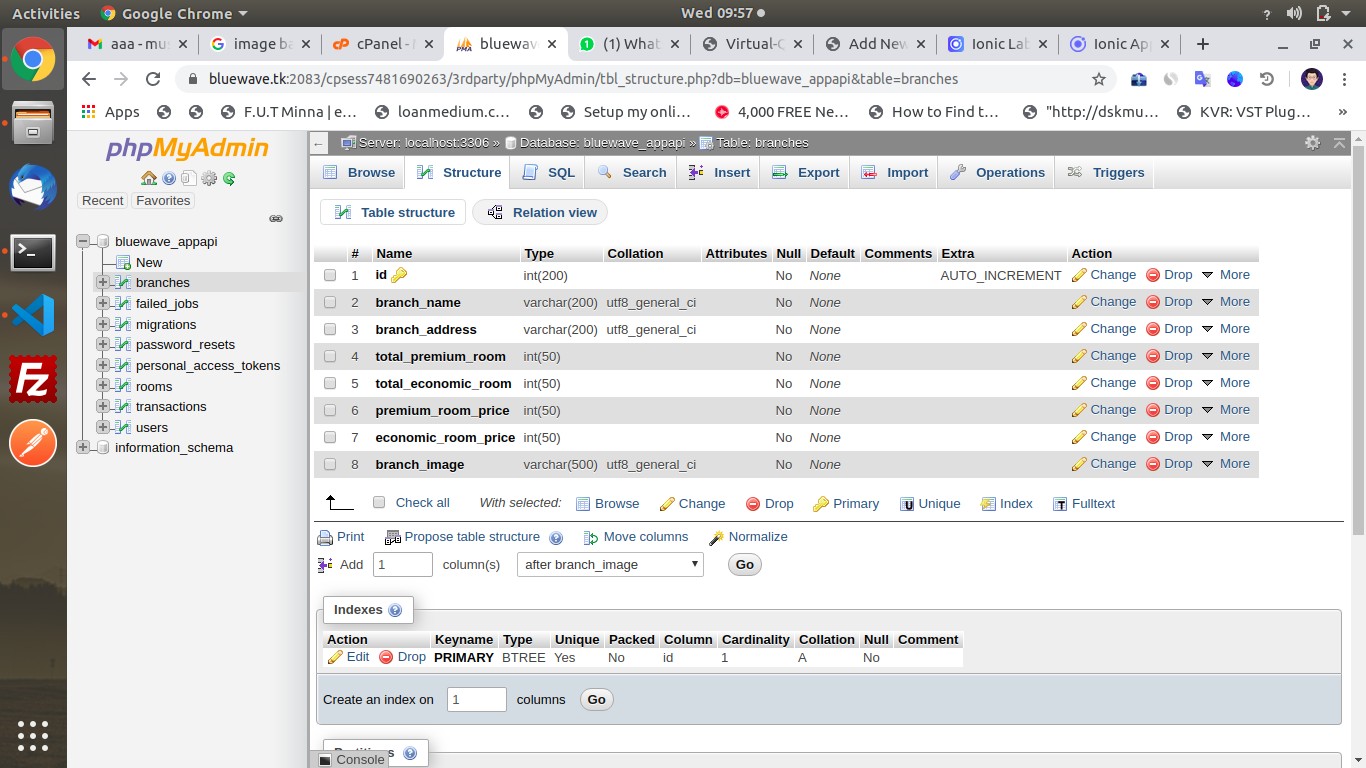
The figure above shows a general overview of the application architecture of the proposed application/system.

###### DATABASE DESIGN TABLES

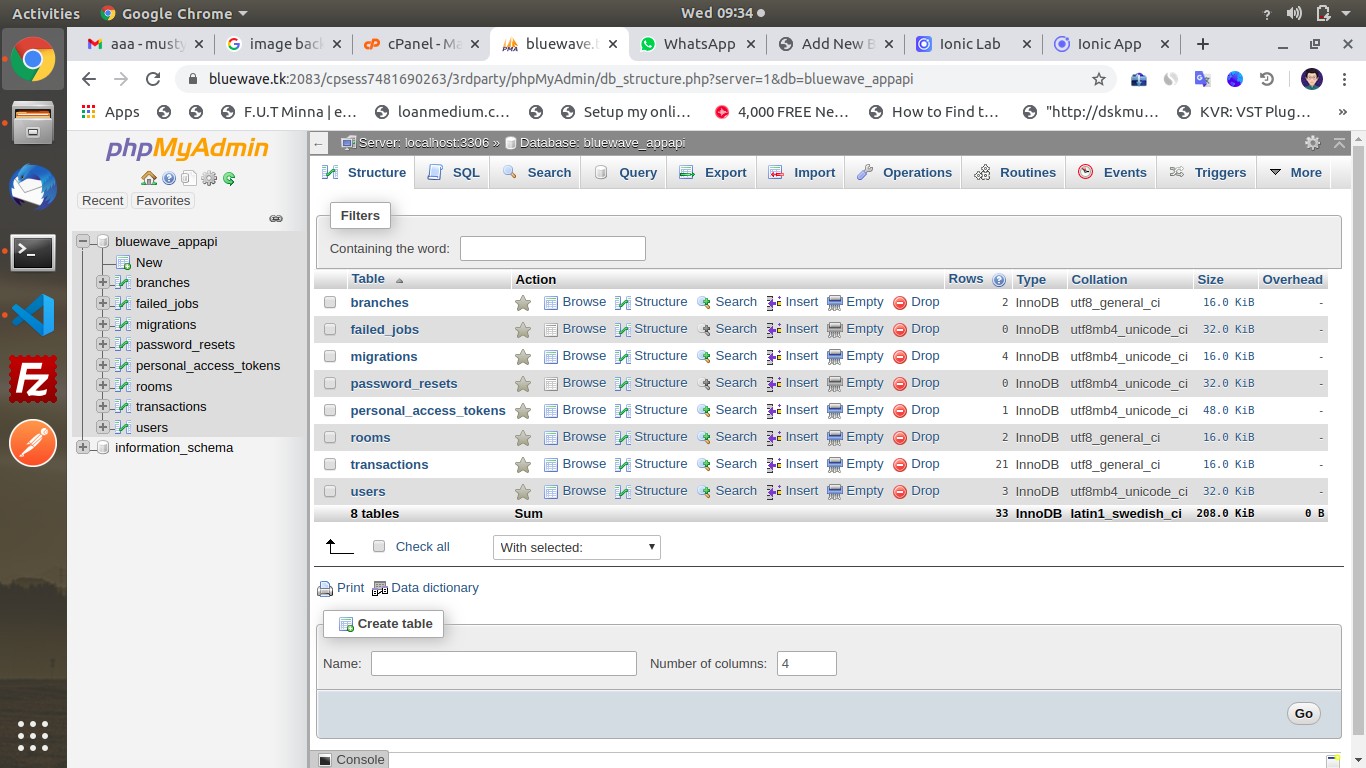
**Users Table as shown in table 3.4 below;**



###### Company Table as shown in table 3.5 below;



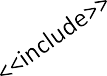
**Passwords Reset Table as shown in table 3.6 below;**



USE CASE

A Use Case diagram depicts the interaction between the users and the system. It shows the functions of the system from the user’s point of view and the various actions the user as the actor carries out.

***Figure 3.3: Use case diagram for the proposed system, showing an interaction between user and system.***



1. **Search**

Rooms

Verify Room

2. **View**

Results

3. **Book**

Rooms

4. **Cancel**

Reservation

Customer

Hotel Staff

5. **Receive**

Confirmation

6. **Create /**

7. **Update**

Check-in/out

* 1. ACTIVITY DIAGRAMS

An activity diagram is a model that shows the process of a task or action from a use case.

Reservation System

ADMIN

EXECUTIVE

CUSTOMER

Login Admin

Login Executive

Register

Customer

Register for

Executive

Book Rooms

Login Customer

Manage

Records

Take Payment, if

payment is cash

Check Availability

of Room

Manage

Reports

Monitoring Rooms & Customers

Book Room

Create Rooms

& Offers

Add Customer details

and also update

Make Payment

Provide rules &

Regulations

View reporotsf

customer & booked rooms

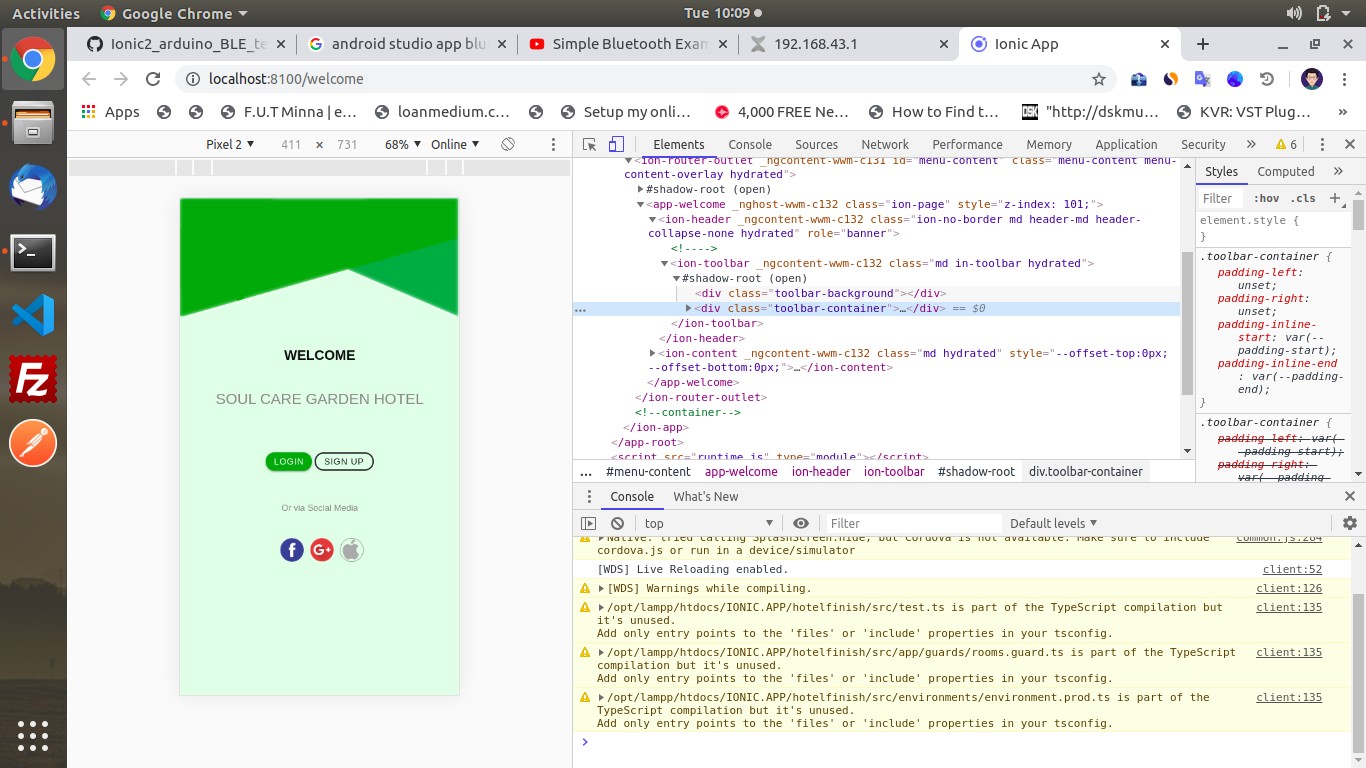
Fill Feedback

form

***Figure 3.4: Activity Diagram for the reservation.***

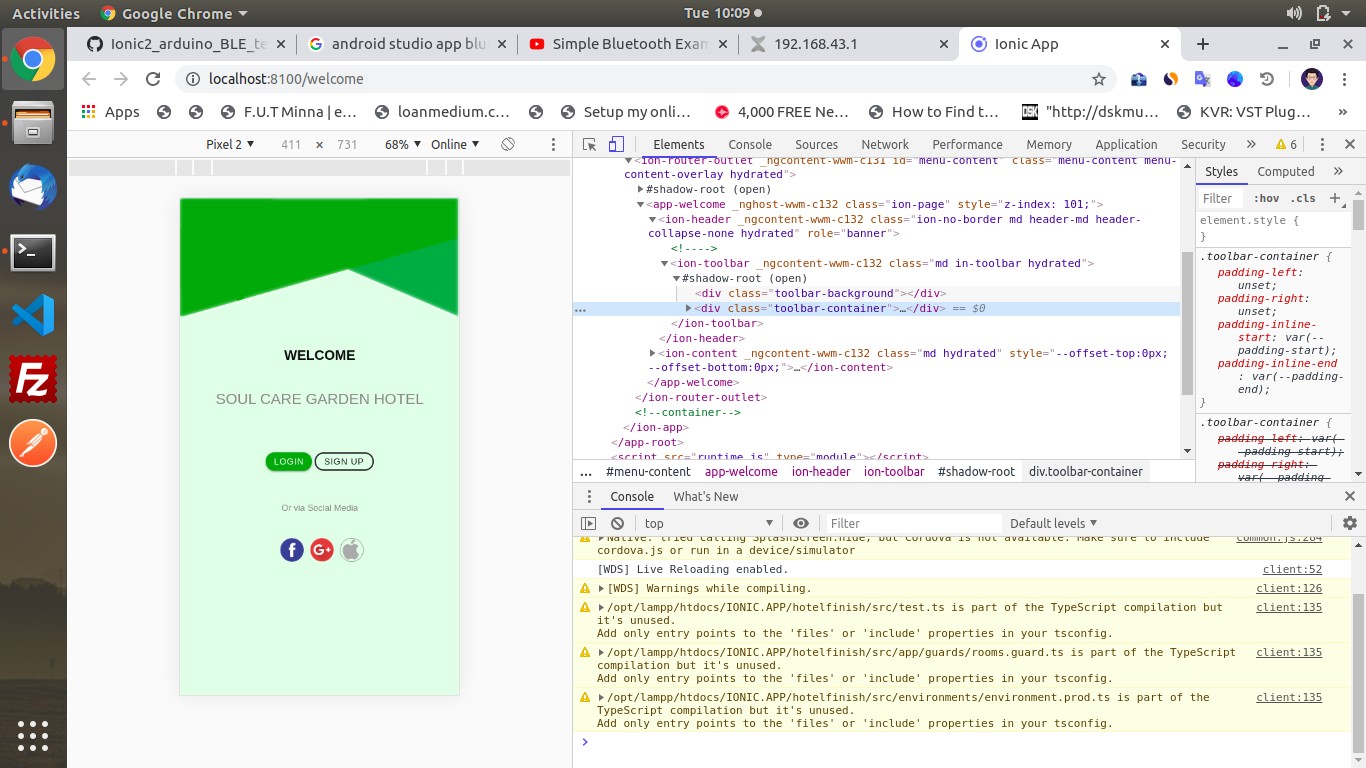
USER INTERFACE DESIGN

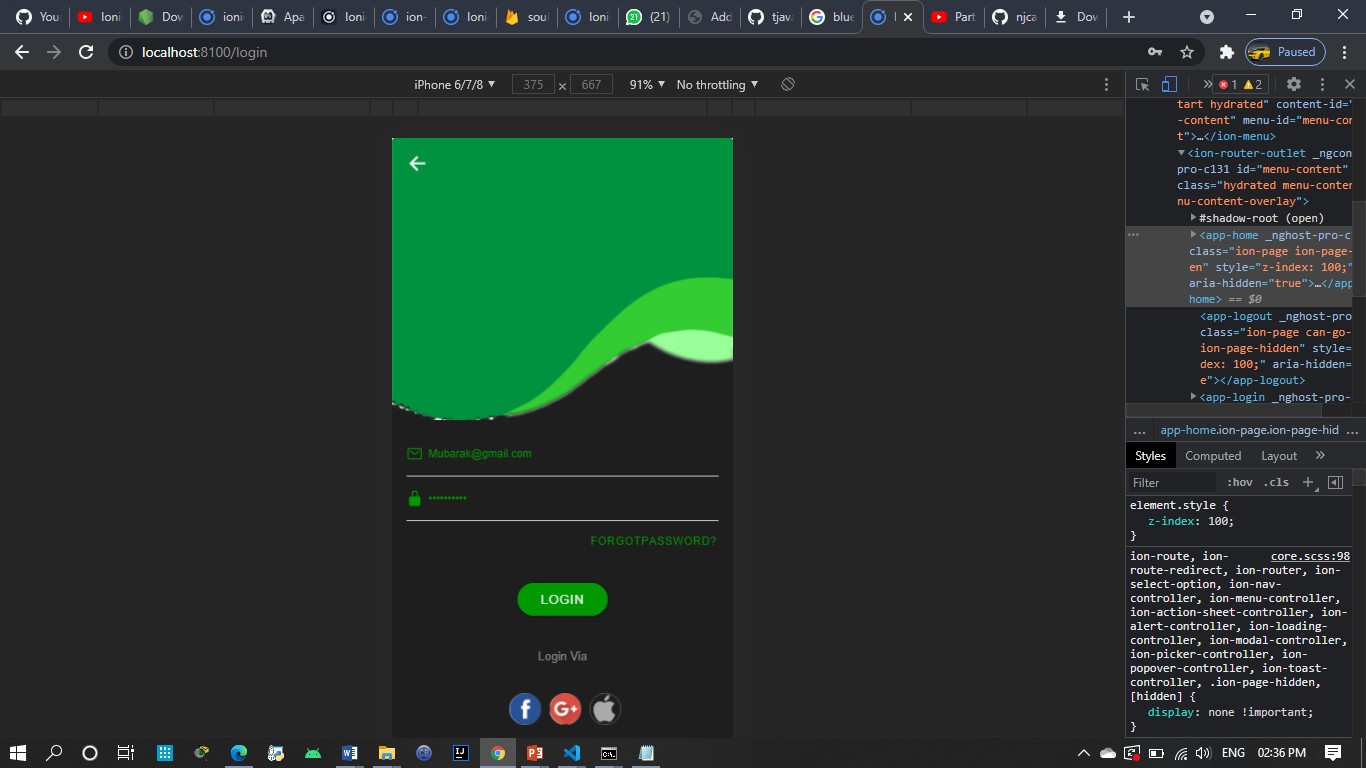
***Figures 3.6: A first view of the user interface of the Proposed Application.***



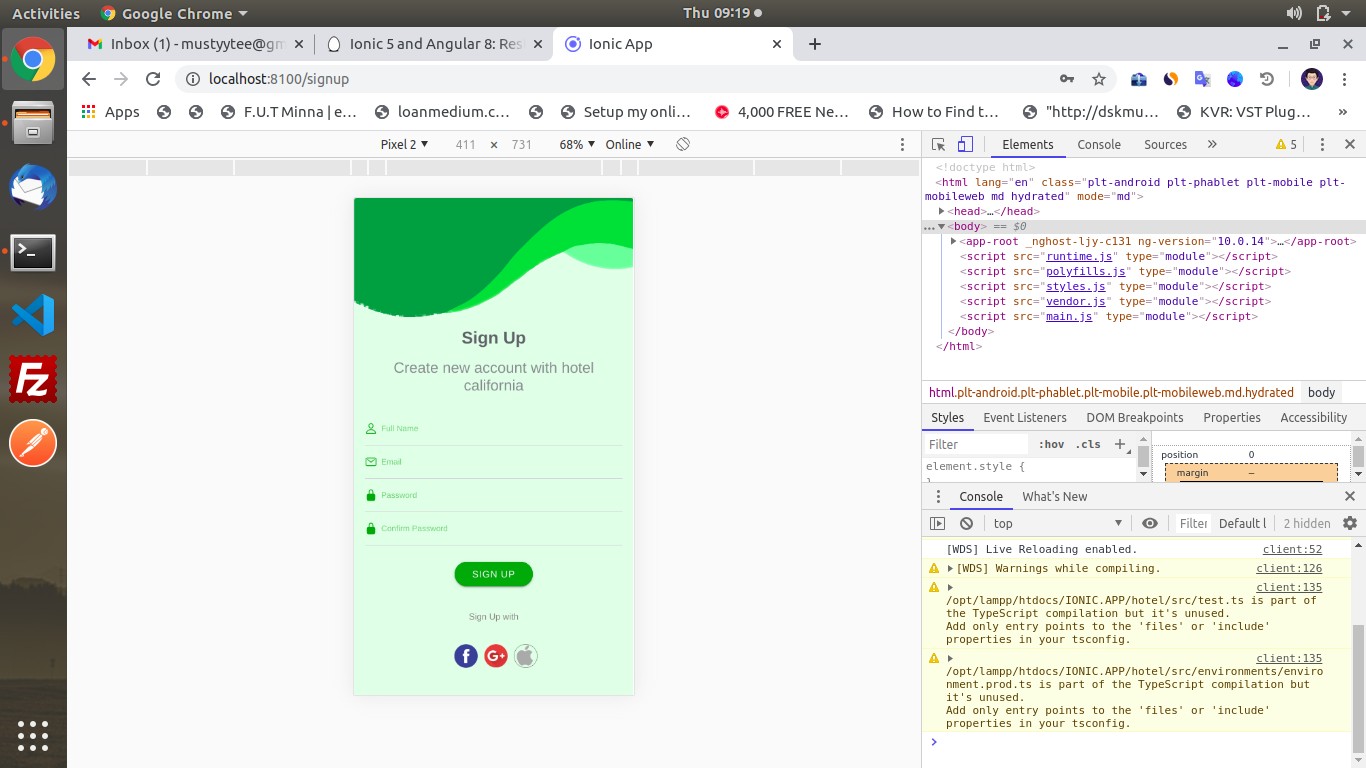
###### INPUT DESIGN

**Login Page**

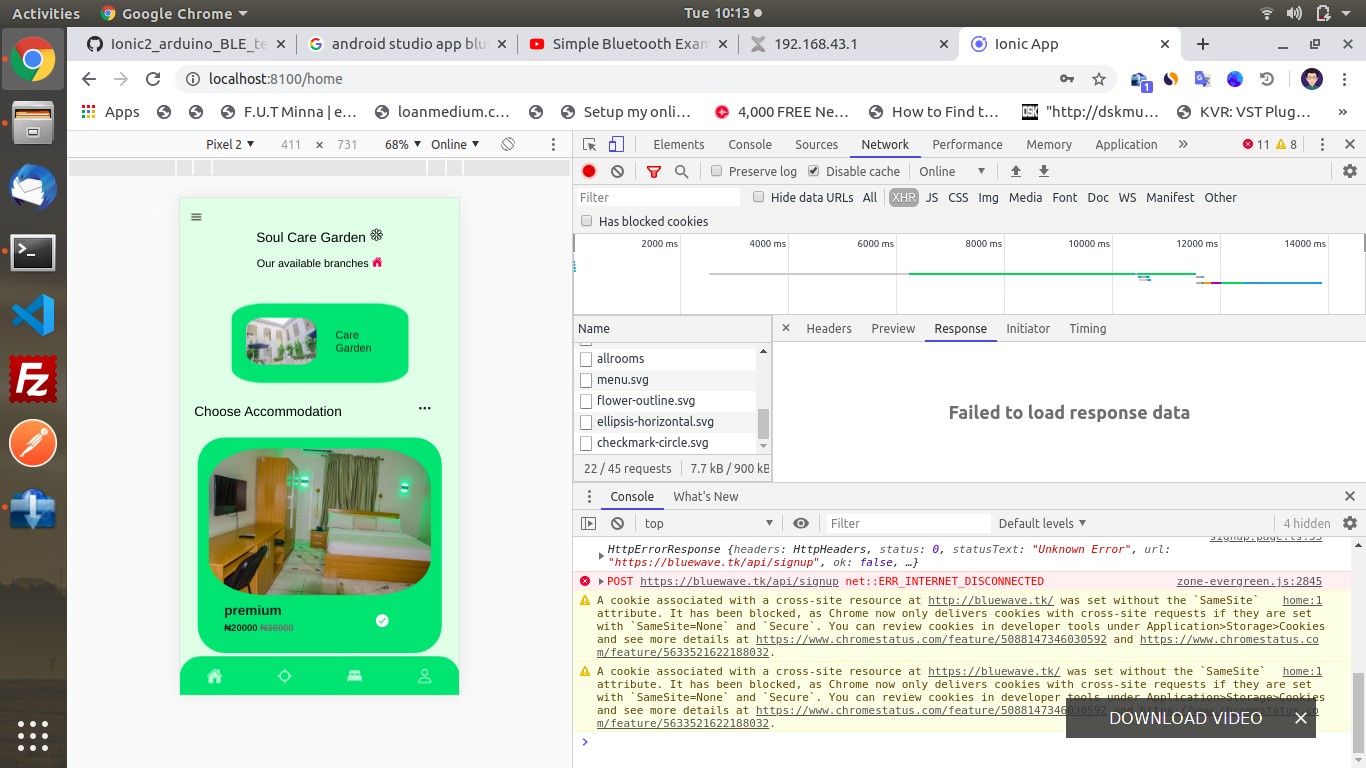


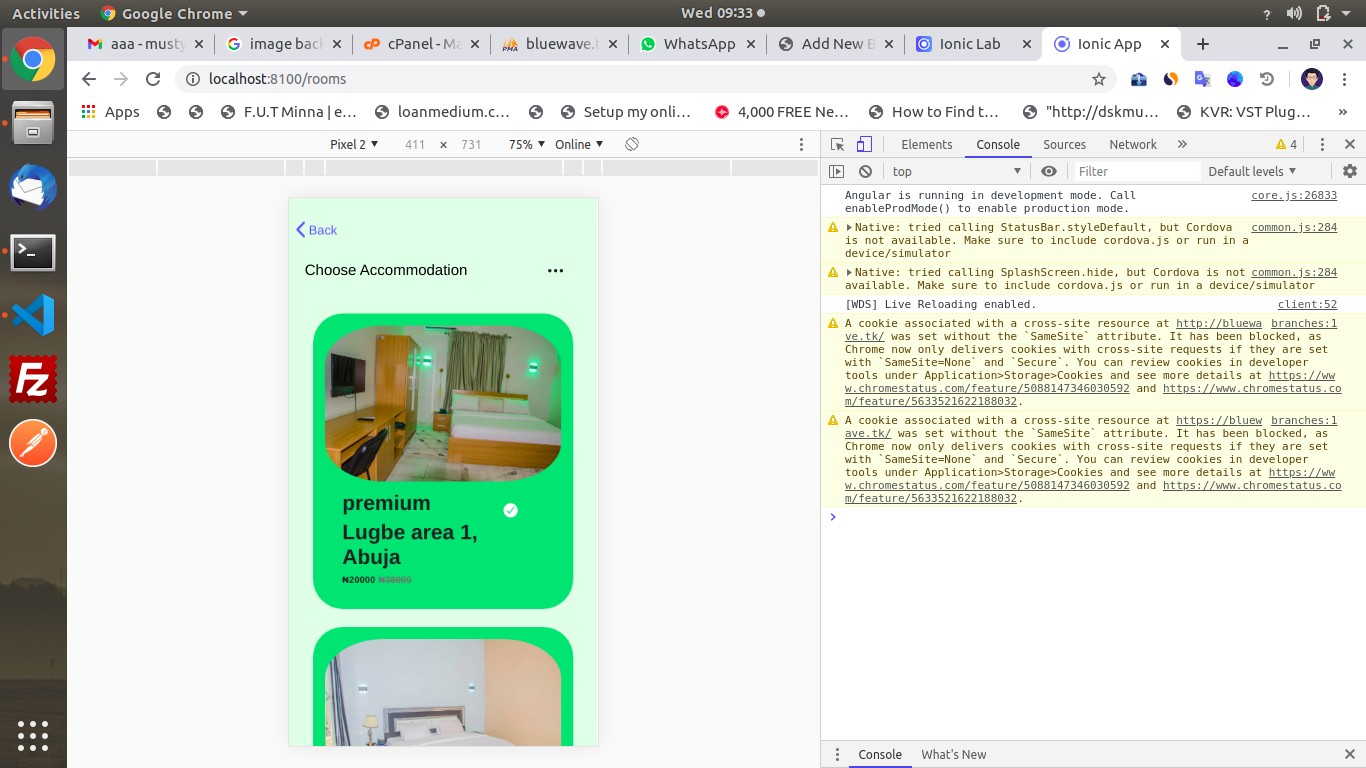


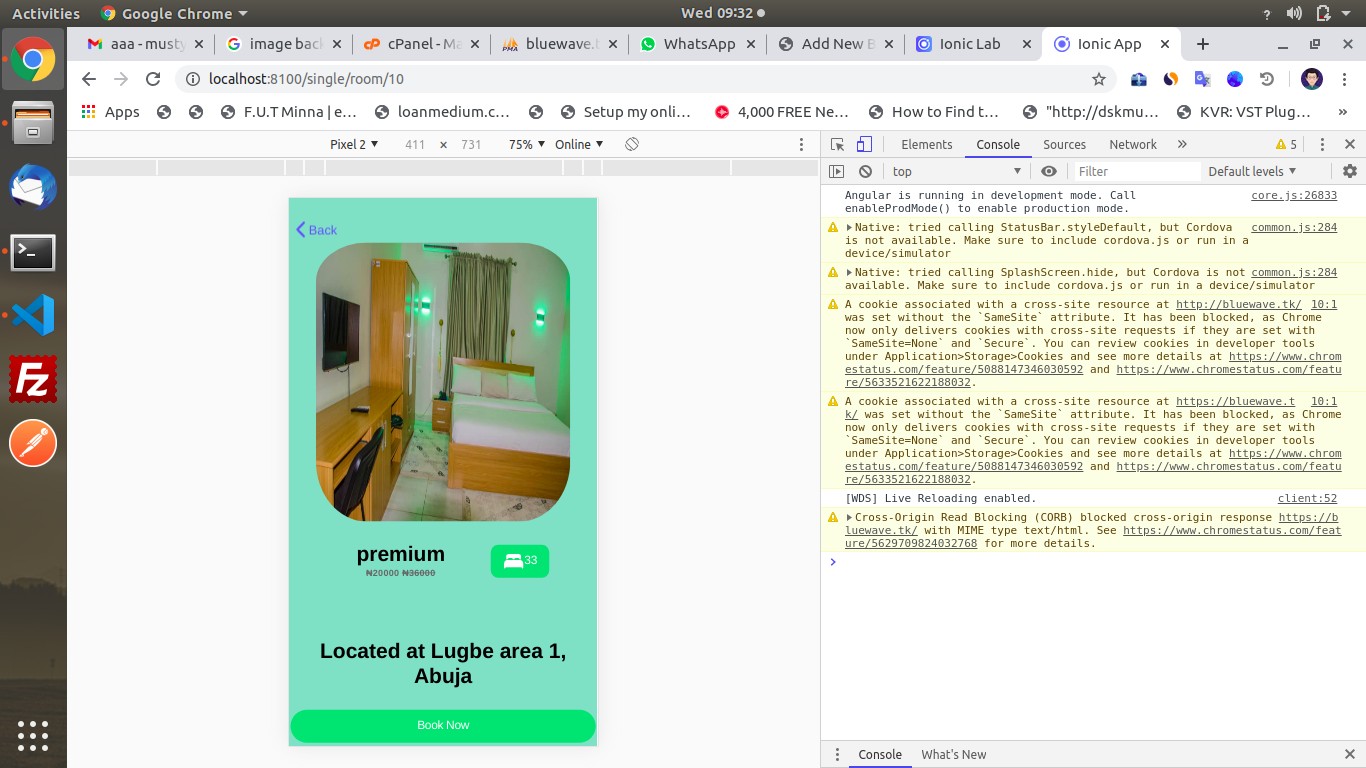
###### Registration account page



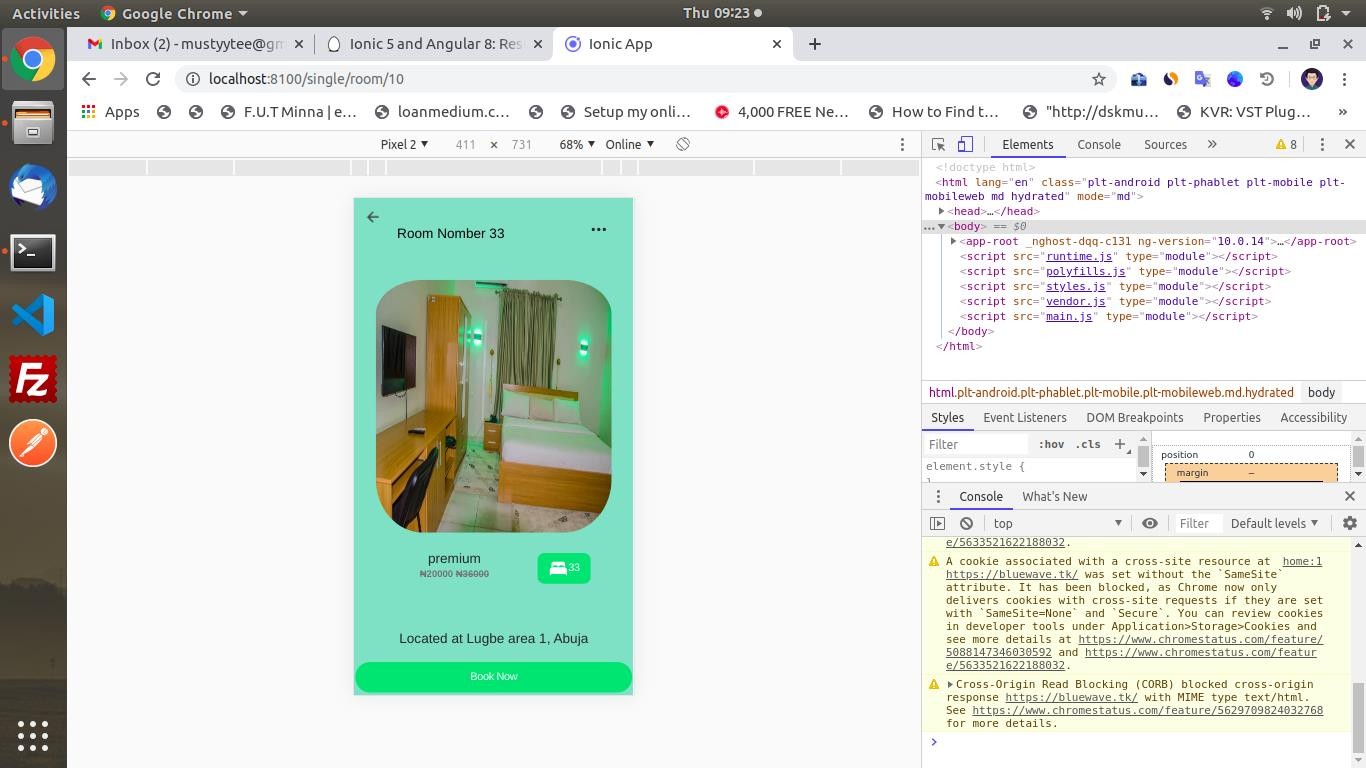
**Available Services**



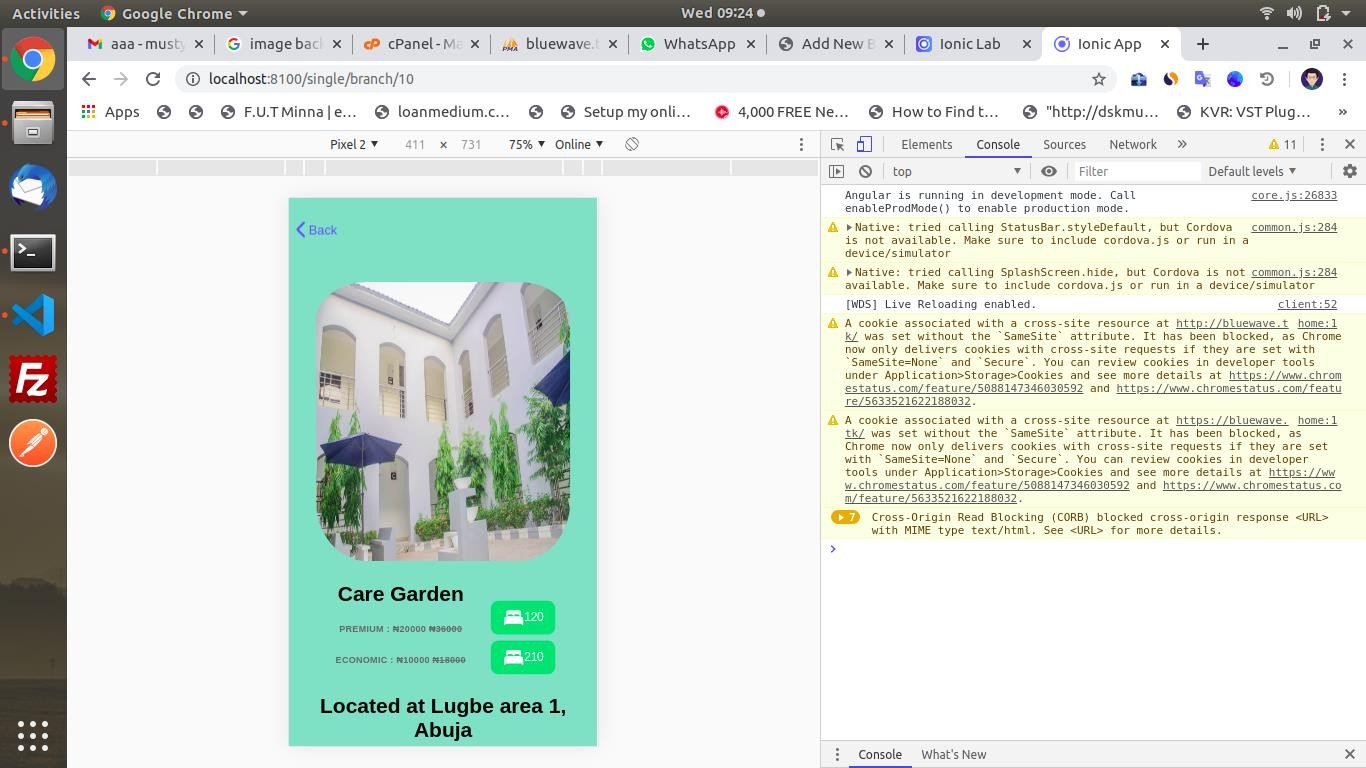




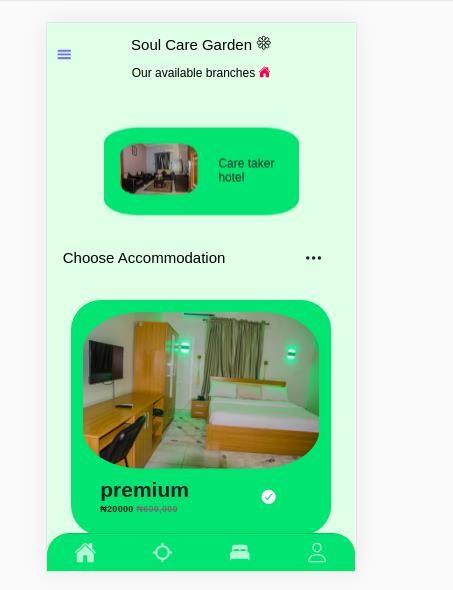
###### Room reservation page



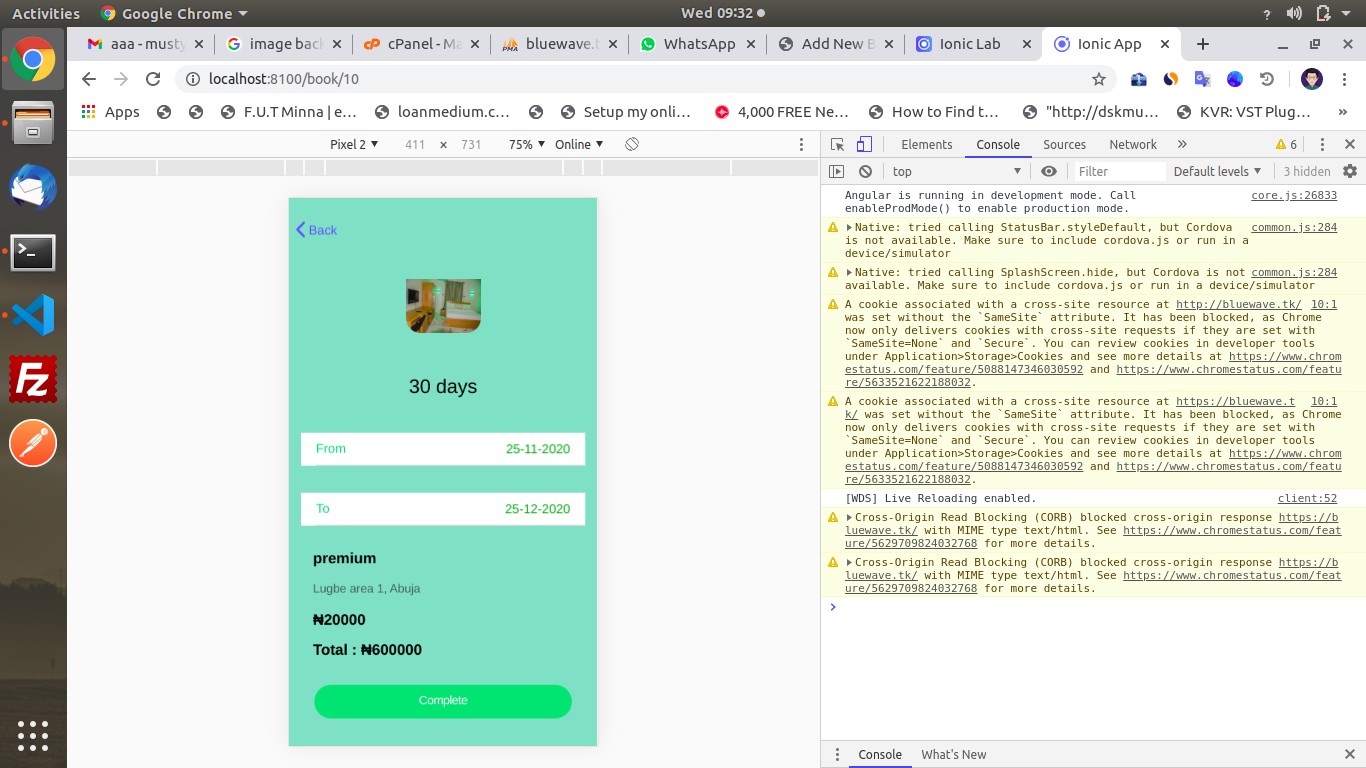
**Environment view**



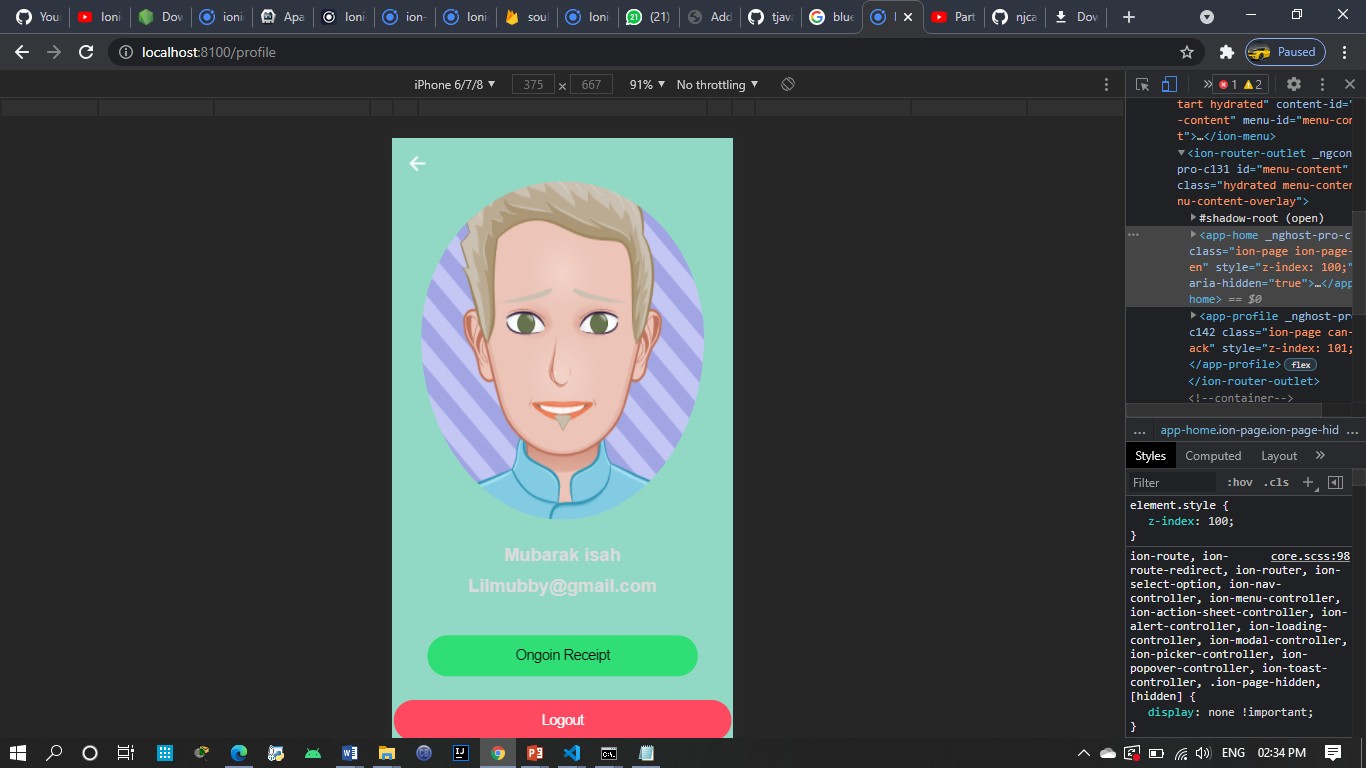
###### Android Application Screens



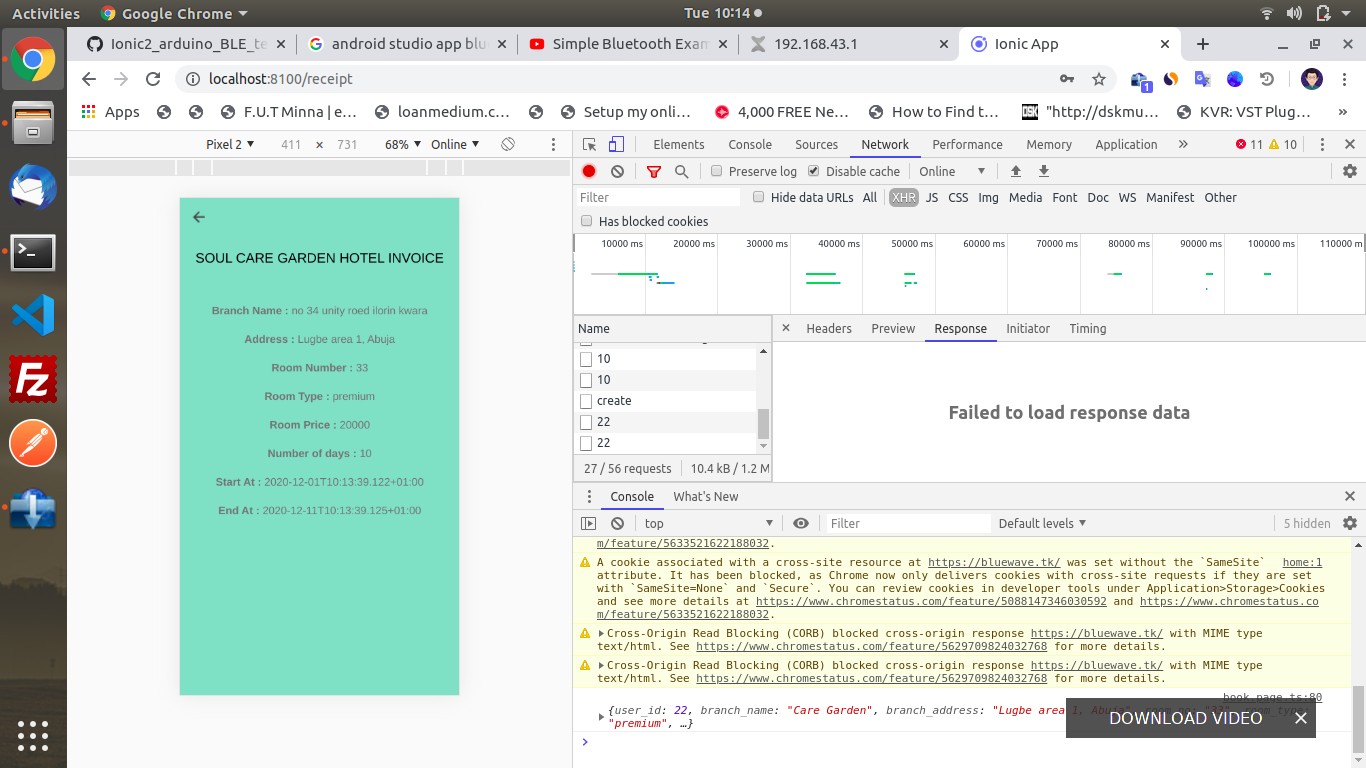
**Booking of Room**



###### Successful Booking



**Notification (Receipt page)**



### Summary

In this chapter the requirement analysis and design, functionalities and requirement gathering techniques used in the design of this system. The Data was deliberately examined and Requirement Gathering Techniques utilized in building the system

## CHAPTER 4: IMPLEMENTATION AND TESTING

###### INTRODUCTION

This chapter is dedicated to going over the main features of the application and going through details of its implementation, and Identifying any problems experienced during the implementation. After problems have been Identified details on how and why the problems occurred will be provided, as well as how said problems were overcome. The chapter will also go through the testing plan by identifying all major components and report on their behavior. We will also go through the test and error report summary and look at corrections for those errors. Finally, The user guide will be provided and then the contents of this chapter will be summarized.

* 1. **Testing**

Testing is one of the most important parts of a software development life cycle. It helps to provide verification and validation on whether all of our functional requirements are met. This is done by generating test data and feeding it to the application to monitor the behavior. This helps to identify the vulnerabilities and limitations of our application. If a feature fails to meet expectations, it will be isolated from other components and examine.

The following sub-chapters go through features to be tested and documents the performance of each component. A test report will also be provided towards the end.

* + 1. **Tests Plans (for Unit Testing, Integration Testing, and System Testing)**

Below is the test plan for “QMS”.

* + - 1. **Test Identifier:**

###### TEST LEVEL: Master Test Plan AUTHOR’S NAME: Mubarak Isah

**AUTHOR’S CONTACT:** [lilmubby@gmail.com](mailto:lilmubby@gmail.com)

* + - 1. Features Not To Be Tested
         * External APIs
         * Internet protocols
      2. Approach

by using the built-in debugger for visual studio code, we’ll be able to stop the app at each breakpoint to examine the app’s data flow. By using the “console.log” function at the end of each method, we’ll be able to see exactly what each method is returning.

* + - 1. Test Deliverables

The deliverables of this test plan are:

* + - * + Test cases
        + Test report
        + Traceability matrix
        + Test results
        + Error report
    1. Test Suite (for Unit Testing, Integration Testing, and System Testing)

*Test case TC-001(User Login)*

|  |  |
| --- | --- |
| Test Suit ID | R-116 |
| Test Case ID | TC-001 |
| Test case summary | To ensures, users can log in |
| Related Requirement | R-116 |
| Prerequisite | * The web app must be opened in a browser * Uninterrupted internet connection must be available * Must be a registered |
| Test Procedure | 1. Navigate to the site using URL 2. login |
| Test Data | User password |
| Expected Result | The user should be logged and navigated to the dashboard |
| Actual Result | The application logged in the user |
| Status | Test case passes |
| Remarks | The test was successful |
| Created by | Mubarak Isah |
| Date Created | 12th November 2020 |
| Executed by | Mubarak Isah |
| Date of Execution | 5th march 2020 |
| Test Environment | Hardware: DELL Laptop  Software: Browser – google chrome |

Table 3.1 Test Suite Performed for Login

Test case TC-003(Logout)

|  |  |
| --- | --- |
| Test Suit ID | R-112 |
| Test Case ID | TC-002 |
| Test case summary | This test case is designed to make sure application logout user |
| Related Requirement | R-112 |
| Prerequisite | * must be a logged-in user * must be on the user’s dashboard * must click on the log out button |
| Test Procedure | 1. Login 2. Click logout |
| Test Data | Logout |
| Expected Result | The user should be logged out redirected to the login page |
| Actual Result | The user is logged out |
| Status | Test case passes |
| Remarks | The test was successful |
| Created by | Mubarak Isah |
| Date Created | 13th November 2020 |
| Executed by | Mubarak Isah |
| Date of Execution | 14th November 2020 |
| Test Environment | Hardware: DELL Laptop  Software: Browser – google chrome |

Table 3.2 Test Suite Performed for Login

Test case TC-003(Role-based access control)

|  |  |
| --- | --- |
| Test Suit ID | R-113 |
| Test Case ID | TC-003 |
| Test case summary | This test case ensures that the only the correct user is allowed to access a route |
| Related Requirement | R-113 |
| Prerequisite | * The web app must be opened in a browser * Uninterrupted internet connection must be available * Must be a registered |
| Test Procedure | 1. Open sited 2. Attempt to log in with different user types and observe the behavior |
| Test Data | login |
| Expected Result | The application should only allow the right user access to a particular route and display an alert for unauthorized users. i.e. users cannot access the  admin dashboard |
| Actual Result | An alert was displayed for unauthorized users and denied them access |
| Status | Test case passes |
| Remarks | The test was successful |
| Created by | Mubarak isah |
| Date Created | 17th November 2020 |
| Executed by | Mubarak Isah |
| Date of Execution | 17th November 2020 |
| Test Environment | Hardware: DELL Laptop  Software: Browser – google chrome |

Table 3.3 Test Suite Performed for role-based authorization

Test case TC-005(find branches)

|  |  |
| --- | --- |
| Test Suit ID | R-118 |
| Test Case ID | TC-005 |
| Test case summary | This test case is designed to make sure new users can find organizations by search |
| Related Requirement | R-118 |
| Prerequisite | * User must be logged in * User must be on Find organization page |
| Test Procedure | * Login * Navigate to find branches page * Type in the state you want to make reservation find your branch closest |
| Test Data | State name |
| Expected Result | The app should filter out organizations from the given state in the search  index |
| Actual Result | The app filters out organizations from the given state in the search index |
| Status | Test case passes |
| Remarks | The test was successful |
| Created by | Mubarak isah |
| Date Created | 18th November 2020 |
| Executed by | Mubarak Isah |
| Date of Execution | 20th November 2020 |
| Test Environment | Hardware: DELL Laptop  Software: Browser – google chrome |

Table 3.4 Test Suite Performed to find organizations

Test case TC-006(Customers/Users progress)

|  |  |
| --- | --- |
| Test Suit ID | R-119 |
| Test Case ID | TC-006 |
| Test case summary | This test case is designed to make sure the application sets the users progress and makes it visible to their supervisor and admin |
| Related Requirement | R-119 |
| Prerequisite | * Be logged it * Find branch * Start internship |
| Test Procedure | 1. Login in 2. Click on start reservation button on the dashboard |
| Test Data | Current date |
| Expected Result | The application should set the user’s start, expected end date, and store it in  the database. Then display it to the authorized users |
| Actual Result | The application sets the user’s start, expected end date, and then store it in  the database. Then display it to the authorized users |
| Status | Test case passes |
| Remarks | The test was successful |
| Created by | Mubarak Isah |
| Date Created | 22nd November 2020 |
| Executed by | Mubarak Isah |
| Date of Execution | 25th November 2020 |
| Test Environment | Hardware: DELL Laptop  Software: Browser – google chrome |

Table 3.5 Test Suite Performed to Users progress

#### Error Reports and Corrections

* Logical errors and bugs due to coding. This was corrected by revising coded and changing logic to fit each function.
* Insufficient permissions. This was due to user not being able to read or write from unauthorized routes. This is good because it shows that the back end did not allow unauthorized users.
* Runtime error. This was caused because of plugins installed on google chrome. It was corrected by disabling the plugins

##### Use Guide

The user guide provides new users instructions on how to get started and use the application. Once a new user has signed in, they are expected to set-up their profile, once that is done, they can now begin their reservation by navigating to the user dashboard and await further instruction from the admin. On the admin page, user details can be accessed by clicking on the displayed user. This then. Expands and reveals several functions that can be performed by clicking on the particular button. All buttons have a descriptive label that describes the function it performs.

###### DATABASE TESTING

The database systems used in this project are MySQL by Oracle Technologies. Database testing was done after system development to check whether the database was able to store the desired data as well as testing its integration with other components of the system. The developer used various categories of data to test the integrity of the database. First normal range data was used and the results were valid after processing. Extreme data was used to test whether the system could accommodate the extreme ranges. Further, exceptional data was used to test how the system would

respond when subjected to invalid data. Consider the following test data performed on the Users table. The two fields under test are Email and password.

##### Overcoming Implementation Problems

To overcome these problems and constrained the development of this application, all issues were isolated and examined. Most of the bugs were fixed by searching online for people who have experienced the same or similar issues and checking for the solution. If a specific is unique to and cannot be found, the best thing to do was to post the problem online a websites’ such as Stack- overflow and Git-hub, then within a few hours, Developers reply with a possible solution.

To solve the issue of role-based authorization preventing google user, the documentation for firebase was examined, this revealed a method called “New User” this returns a Boolean value that allows us to write functions depending on if it is true or false.

###### FIELD NAME DATA TYPE TEST DATA RESPONSE COMMENT

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **FIELD NAME** | **DATA TYPE** | **TEST DATA** | **RESPONSE** | **COMMENT** |
| **Email** | varchar | [admin@admin.com](mailto:admin@admin.com) | Accepted | Valid |
| **Email** | varchar | Admin.com | Rejected | Invalid |
| **Email** | Varchar | [admin@admin.com](mailto:admin@admin.com) (email already  exists) | Rejected | Valid |
| **Password** | Varchar | Password01 | Accepted | Valid |
| **Password** | Varchar | #PAssword (Confirmation  Error) | Rejected | Invalid |

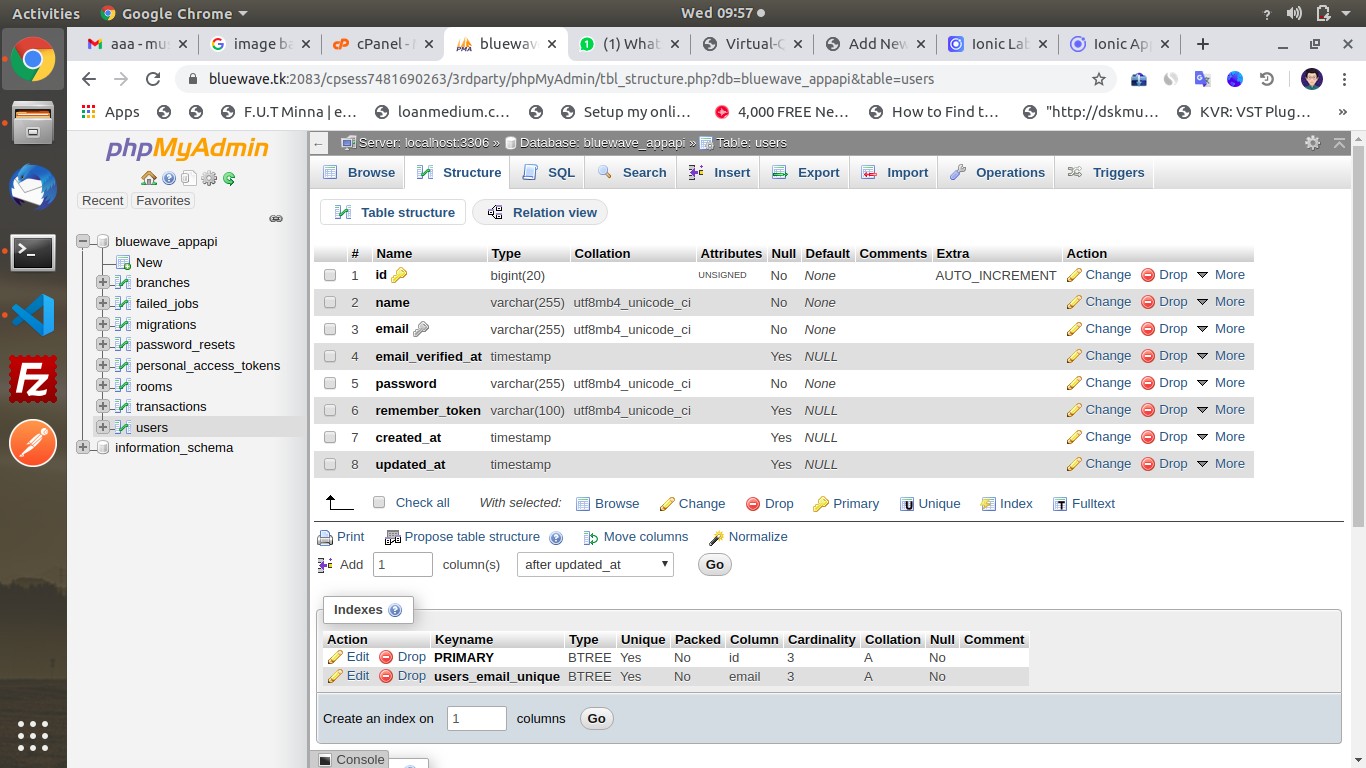
***Table4.1 Database testing***

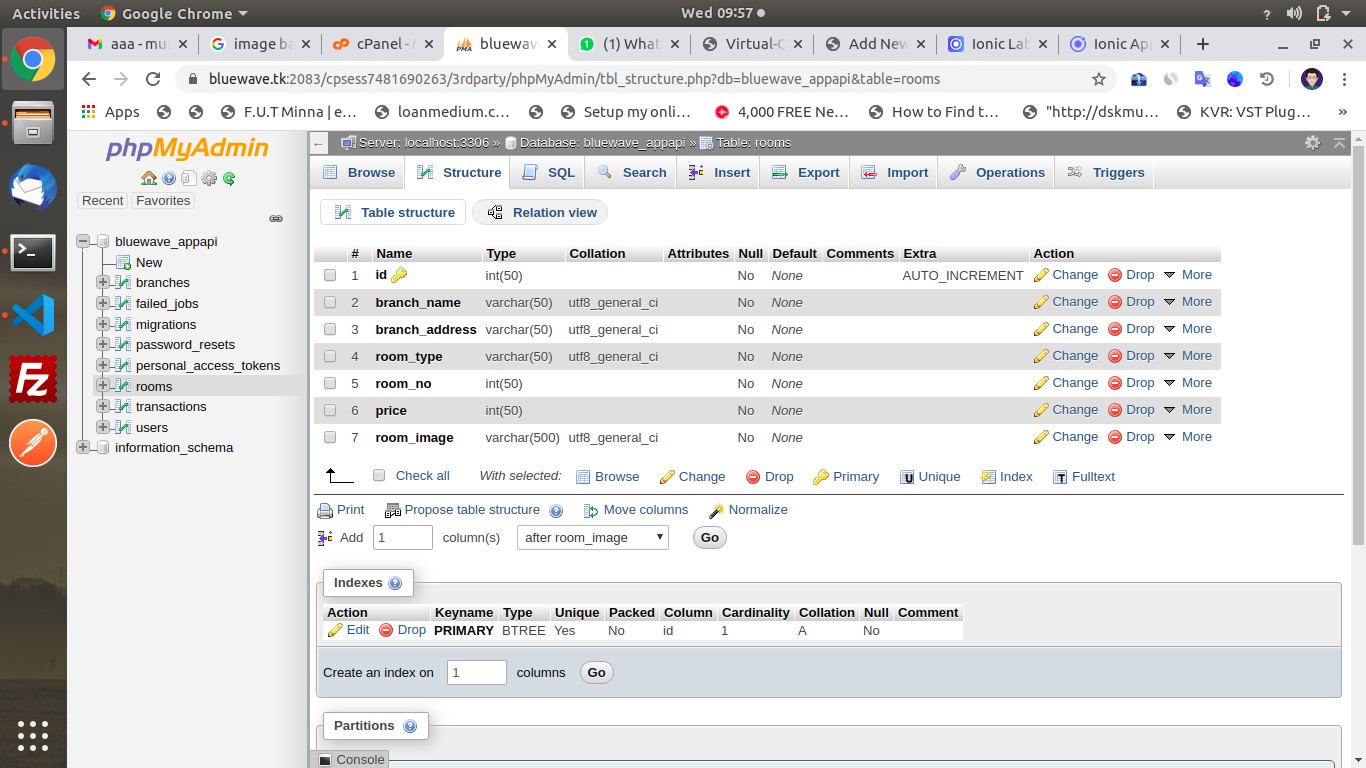
* 1. **UNIT TESTING**

Unit involves testing software with a small piece of source code. QMS is built on top of the Laravel php framework, thus has access to a custom TDD library. When performing tests, some assertions would be made, and the testing function would then assert if true or false.

Source code for unit testing were created by the developer as a part of software development. The following unit tests were performed to ascertain functionality.

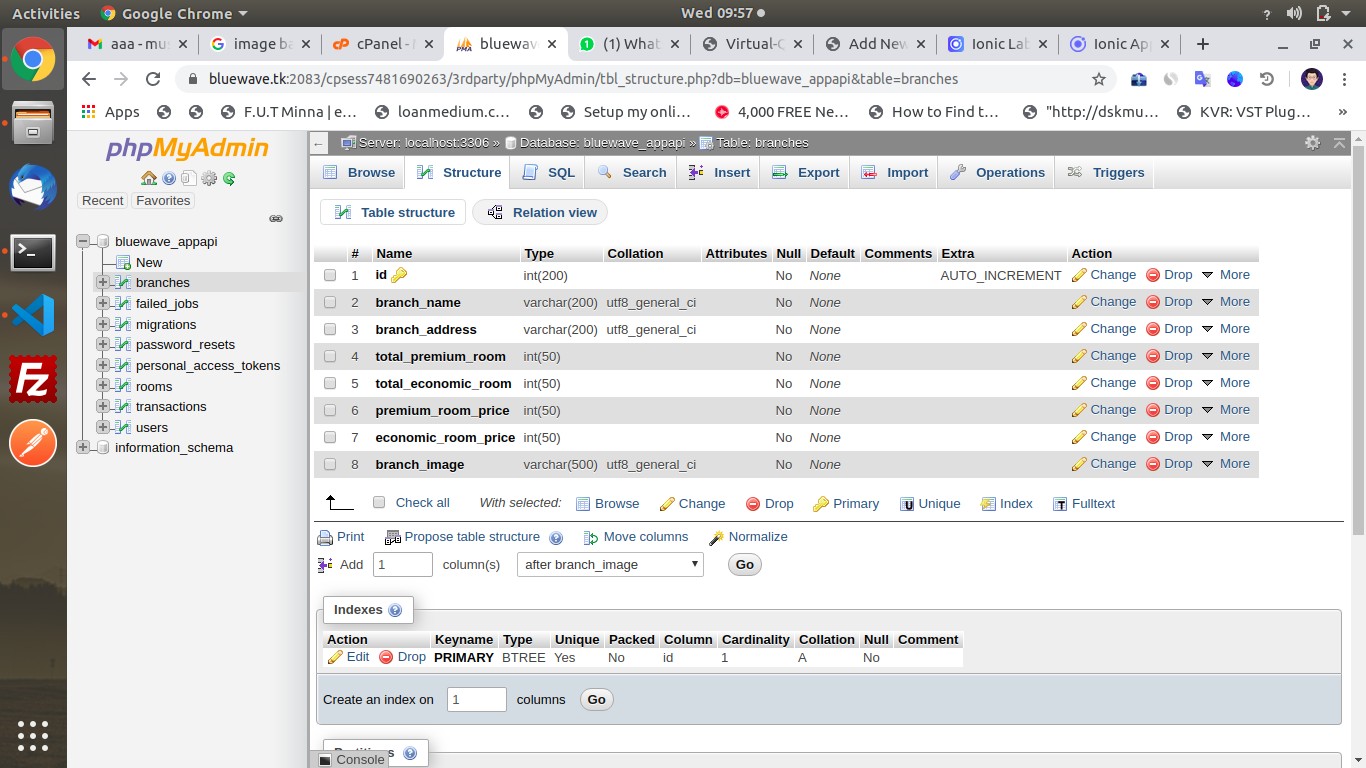
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TEST**  **CASES (TC#)** | **TEST NAME** | **TEST DESCRIPTION** | **SOFTWARE** | **TEST ENVIRONMENT** |
| **TC1** | Navigation Tests | This test verifies if the user is able to navigate  the site and access all | QMS | Windows Home Edition, 500GB HDD,  4GB RAM, Apache |
|  |  | URLs in the application.  Testing a login scenario |  | Server, PHP Engine,  MySQL Server |

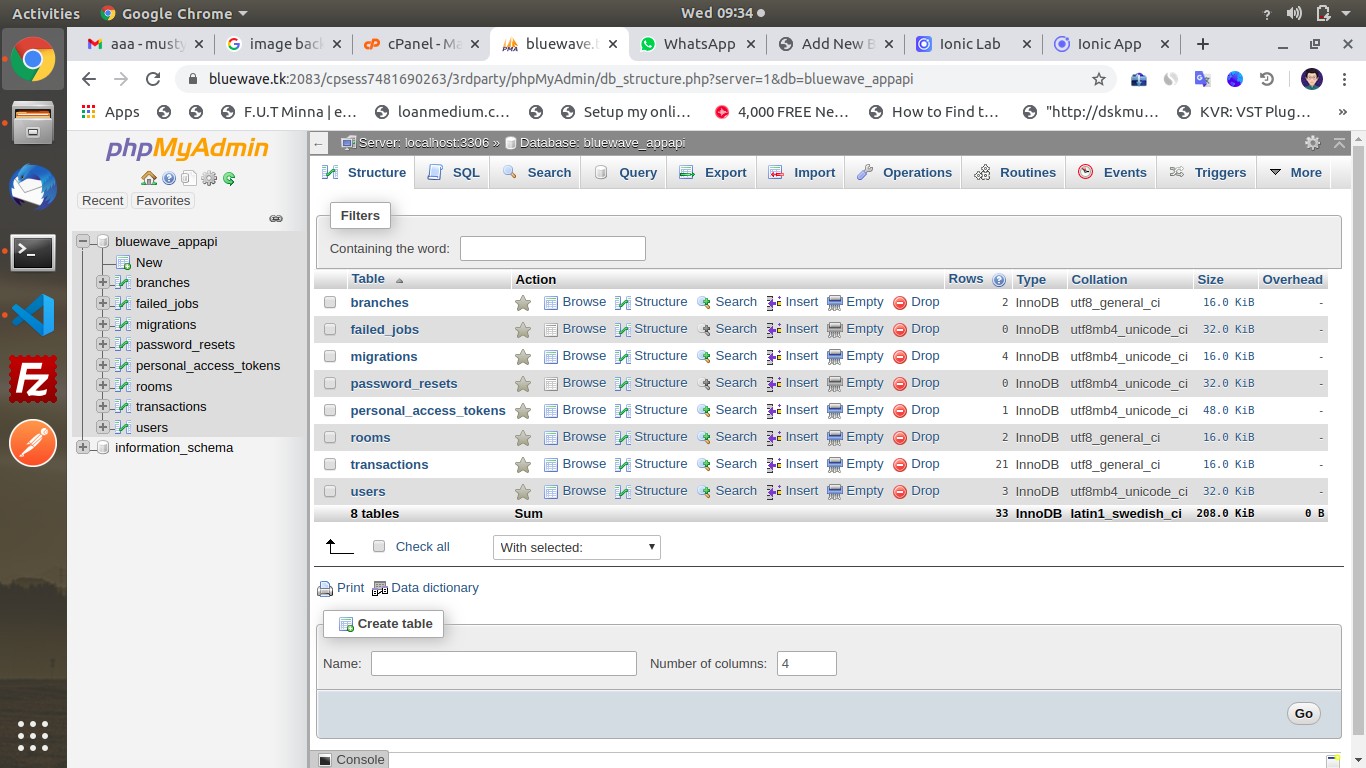




***Figure 4.1-unit test navigation***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TC2** | Authentication Tests | This test verifies the username and password to access QMS | QMS | Windows Home Edition, 500GB HDD, 4GB RAM, Apache Server, PHP Engine,  MySQL Server |





***Figure 4.2-unit test authentication***

INTERFACE TESTING

Interface testing was performed to evaluate if all units of the interface transferred control to each other. To perform the interface tests, the developer created a checklist that outlined all the functional requirements of the system and the various test cases to assess them.

|  |  |
| --- | --- |
|  | **Description** |
| **TR01** | Create an account for users of the application |
| **TR02** | The system services choices for users |
| **TR03** | The system shows branches to be selected  manually |

|  |  |
| --- | --- |
| **TR04** | The nearest branch is selected manually |
| **TR05**  **TR06** | The user generates a queue number  The system view ticket information |

***Table4.2Functional requirements***

###### TEST CASES

The table below shows how each of the functional requirements were assessed using Test Cases. TEST CASE 1 (TC1)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TEST**  **CAS**  **E** | **FUNCTIONAL TEST** | **TEST DESCRIPTION** | **SOFTWARE TEST** | **REQUIREMENT ENVIRONMENT** |
| TR01 | Login and account create | Verify and authenticate user using email and password. | QMS | Windows Home Edition, 500GB HDD, 4GB RAM,  Apache Server, PHP  Engine, MySQL |

***Table 4.3 Test Case 1***

###### TEST CASE 2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TEST**  **CAS**  **E** | **FUNCTIONAL TEST** | **TEST DESCRIPTION** | **SOFTWARE TEST** | **REQUIREMENT ENVIRONMENT** |
| TR02 | The system services | The system shows the available accommodation and their respective functions | QMS | Windows Home Edition, 500GB HDD, 4GB RAM,  Apache Server, PHP  Engine, MySQL Server |

***Table 4.4 Test Case 2***

###### TEST CASE 3

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TEST**  **CAS**  **E** | **FUNCTIONAL TEST** | **TEST DESCRIPTION** | **SOFTWARE TEST** | **REQUIREMENT ENVIRONMENT** |
| TR03 | Branches available | The system displays the available branches and their location. | QMS | Windows Home Edition, 500GB HDD, 4GB RAM,  Apache Server, PHP  Engine, MySQL Server. |

***Table 4.5 Test Case 3***

###### TEST CASE 4

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TEST**  **CAS**  **E** | **FUNCTIONAL TEST** | **TEST DESCRIPTION** | **SOFTWARE TEST** | **REQUIREMENT ENVIRONMENT** |
| TR04 | Nearest branch Location | The nearest branch is displayed and can be selected manually. | QMS | Windows Home Edition, 500GB HDD, 4GB RAM,  Apache Server, PHP  Engine, MySQL Server. |

***Table 4.6 Test Case 4***

###### TEST CASE 5

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TEST**  **CAS**  **E** | **FUNCTIONAL TEST** | **TEST DESCRIPTION** | **SOFTWARE TEST** | **REQUIREMENT ENVIRONMENT** |
| TR05 | Generation of a queue number | The system generate a queue number and it will be on the ticket or receipt. | QMS | Windows Home Edition, 500GB HDD, 4GB RAM,  Apache Server, PHP  Engine, MySQL Server. |

***Table 4.7 Test Case 5***

###### TEST CASE 6

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TEST**  **CAS**  **E** | **FUNCTIONAL TEST** | **TEST DESCRIPTION** | **SOFTWARE TEST** | **REQUIREMENT ENVIRONMENT** |
| TR06 | Notification/receipt or ticket  information | After completion or successful process, the system display ticket or receipt. | QMS | Windows Home Edition, 500GB HDD, 4GB RAM,  Apache Server, PHP  Engine, MySQL |

***Table 4.8 Test Case 6***

* + 1. USABILITY TESTING

The table below summarized tests that were performed to ascertain the usability and experience of users while interacting with the system.

***Table 4.4 Usability Testing***

**Element Output**

|  |  |
| --- | --- |
| **Flow from start to finish** | Yes |
| **Feedback from Actions performed** | Instant Feedback |
| **Tokens Received** | Received |
| **Seamless Navigation** | Yes |
| **Performance** | Optimal |
| **Failure or crashes** | None |
| **Runtime error messages** | None |
| **Slow or delayed loading** | Acceptable |

* + 1. INTEGRATION TESTING

The purpose of this testing was to check whether the various modules of the system are well integrated and working harmoniously. All the form modules were well connected with the database and processing of data was successfully done. The reports generated outputs successfully as expected from the database and in the correct formats.

* 1. Summary

This chapter demonstrated in clear detail, the main features of the applications and describes the techniques used to implement those features. The problems that occurred during the implementation were also identified and the chapter went over all of the ways method, which were used to overcome those problems. The chapter then goes through the test plan, procedures, and documents the results of those tests. Errors found during the test procedure were also documented and the corrections for those errors and a user guide were also provided. The next chapter will discuss the conclusions, reconditions, and future enhancements.

## CHAPTER 5: CONLUSION AND RECOMMENDATION

##### Introduction

This chapter is the final chapter of this thesis. In it, we will go over an objective assessment of the project as a whole in section 5.2. Then discuss the limitations and challenges faced during the development of this project in section 5.3. In section 5.4, we will also go over plans for future enhancements for this project. Finally, recommendations will be made, based on the findings of this project.

##### Objective Assessment

This project has gone according to plan, all of its functionalities have been implemented, and all functional requirements were satisfied. The project is designed and coded in such a way that any further modifications that are needed in the future can be easily implemented without affecting the functionality of the system. By building upon my foundation of object-oriented programming to develop such apps that solve our immediate issue in our current environment. This app has the potential to be used by every university in Nigeria as there is nothing like it that has been built so far in this country. In addition, it will serve as a benchmark by which everyone who develops a similar application will be judged.

##### Limitations and Challenges

During the development process, there was an outbreak of a deadly called COVID-19, this out- break spread worldwide and was declared as a national emergency when the first case was recorded in Nigeria. Due to this, a nationwide lockdown was announced which prevented civilian movements and the federal government announced that all educational institutions were to close immediately. Because of this. I and other students were unable to attend our weekly project assessment classes. This class was valued because it provided constructive feedback about this project, and also gave recommendations and ideas on how to solve bugs and errors and ideas on how to implement certain features. The lockdown also prevented interactions with other lecturers for outside counsel and opinions. However, communications with the assigned supervisors continued via formal online interactions like sending e-mails.

Demonstrations of the project were performed by screen recording the application and providing voice-over narrations.

##### Future Enhancements

The end of this project does not mark the end of the application’s developments. Future updates and improvements will be added. The design of the app will also change over time to fit modern UI design trends.

There are currently plans to deploy the application to mobile users. Ionic and Angular already provide us with the tools necessary for this. It uses capacitors, which is a technology that allows us to wrap our web application in its current state with a mobile frame, and deploy it to both android and iOS users. In addition, to make the mobile versions of the application run better, optimizations will be made to allow better native integration, meaning the app will have access to the device’s components such as the camera and fingerprint sensor or any other tool provided for biometric authentication.

The app will also be enhanced by adding quality of life improvements such as more search filters in the admin page, which will help them to sort user data as the application is used more and more by different departments and gains many more users.

Plans are also in place to evolve the application so that will allow major project supervision, as there is no implementation in place for that system. Finally, bug fixes will be done as real-world users discover them because they are free to interact with the application is an unexpected way, which can help highlight unnoticed problems or bugs.

##### Recommendations

Based on the findings of this project, it is clear that there is a serious lack of development in facilitating and automating our systems used for academic processes with that in mind, the following recommendations are made.

Nigerian universities that have a department dedicated to computer science and information technology should utilize the skill of staff and students in that department to provide technology solutions to any and all of the institution's underlying problem and update current systems to use modern computer technologies,

Students should also be able to identify issues in their immediate environment and seize the opportunity to develop meaningful apps for their community. Developing such a mentality will allow Nigerians to solve our problems one by one and eventually develop a technologically aided society.

##### Summary

This chapter concluded this project thesis. In it, we were able to give an objective assessment of the project itself. In addition, discussed limitations and challenges such as the nationwide lockdown, we discussed plans for future improvements such as native interactions, mobile app optimizations, and plans for evolving the application for major project supervision. Finally, we discussed the author's recommendation for academic institutions and Nigerians.

###### CONCLUSION

The aim of developing a queue management system for soul care gardens hotel is to make queue management easy for the organization, as the world is still suffering from the dilemma of COVID’19 pandemics; this queue management system is to enhance the social distancing rule set by the government. The system design uses android application that can be downloaded from play store or link. In the application, the input of the customers (registration and log in) are being taken, then shows the customer the available rooms and their prices, then the customer can book and pay without wasting time, after which the customer will come to the hotel and claim the room. The system can manage queue smartly without supervision. The system development was successful, all the interface of the project is functioning as required and the aim of the project was achieved.

**REFERENCES**

Bank ATM Bhavin Patel, Pravin Bhathawala; Case Study for Queuing Model; International Journal of Engineering Research and Applications (IJERA) Vol. 2, Issue 5(2012)

Damodharf Shastrakar, Sharad S Pokley, KD Patil, “Literature Review of Waiting Lines Theory and its Applications in Queuing Model” *In proc. International Journal of Engineering Research and Technology(IJERT) ISSN;2278-0181(2016).*

Dr. Ahmed S. A. AL-Jumaily, Dr. Huda K. T. AL-Jobori; Automatic Queuing Model for Banking Applications; International Journal of Advanced Computer Science and Applications, Vol. 2, No. 7, (2011)

M.E. El-Naggar; *Application of Queuing Theory to the container terminal at Alexandria seaport; Journal of Soil Science and Environmental Management*; Volume 1, Issue 4 (2010)

Manoon Ngorsed, Poonphon Suesaowaluk, “Hospital Service Queue Management System with Wireless Approach” Graduate School of e-learning in Information Communication Technology Assumption University, Bangkok Thailand.

Md. Belayat hossain, Md. Nahid Hossain, Md. Enamul Hoque Chowdhury and Md. Habibur Rehman, “Design and Development of microcontroller-based Electronic Queue Control System” DUJASE Vol. 2(1) 11-15 JULY 2011.

Md. Manjurul Ahsan, Md. Raisul Islam , Md. Ashikul Alam; Study of Queuing System of a Busy Restaurant and a Proposed Facilitate Queuing

Md. Nasir uddin, Mm Rashid, Mg Mistafa, Belayat H, Sm Salam, Na Nithe and DZ ahmed, “Automated Queue Management System” *In proc. Global Journal of Management and Business Research (GJMBR) ISSN; 2249-4588 and print ISSN; 0975-5853 Vol.16 issue 1 Version 1.0* 2016. Mohamad Fazli Bin Alias, “Queue Management System” Facaulty of Electrical and Electronic Engineering University Malaysia Pahang. November 2007.

Mrs. S.Maragatha Sundari, Dr. S.Srinivasan; M/M/C Queueing Model For Waiting Time of Customers In Bank Sectors; Int. J. of Mathematical Sciences and Applications, Vol. 1, No. 3, September (2011)

Ndung’u Michael Ngugi, Josphat karami Mwai, Mwangi Njomo, “Virtual Queue Management System” *In proc. International Journal of social science and information technology (IJSSI) ISSN; 2412-0294 Vol. 4 issue 5, August 2018.*

Nityangini Jhala and Pravin Bhathawala, “Smart Queue Management System for Banking Sector” *In proc. International Journal of Business Management and Economic Research. ISSN; 2349- 2333 Volume 3, Number 1 (2016). Pp. 29-34.*

Nityangini Jhala, Pravin Bhathawala; Application Of Queuing Theory In Banking Sector; IOSR Journal of Mathematics (IOSR-JM); Volume 12, Issue 2 Ver. II (Mar. - Apr. 2016)

S. K. Dhar, Tanzina Rahman; Case Study for Bank ATM Queuing Model; IOSR Journal of Mathematics; Volume 7, Issue 1 (2013)

S. Vijay Prasad and V. H. Badshah; *Alternate queuing system for tatkal railway reservation system*; Advances in Applied Science Research-(2014)

Vasumathi.A, Dhanavanthan P; Application of Simulation Technique in Queuing Model for ATM Facility; International Journal of Applied Engineering research, Volume 1, No 3, (2010)

**APPENDICES**

**Appendix A - Project Document**

The project documentation for the Design and Development of a Queue Management System for Soul Care Garden Hotel:

DETAILED PROJECT DOCUMENTATION

**Candidate Name:** Mubarak Isah

**Student Id No:** BU/16C/IT/2360

*Soul care App*

Design and Development of a Queue Management System for Soul Care Garden Hotel.

**Course of Study:** B.Sc. Software Engineering

*Background Study and Objectives*

The proposed system is designed as an online solution to the long wait time and crowd with the pandemic we are facing now. The queue management system aims help service provider manage their customers efficiently. The system will be designed to ease customer flow management, which is useful for managers of the service provider. In addition, help provide the customer with clarity into hotel traffic and waiting time before leaving their destination. The app was developed using the waterfall methodology, was successfully tested, and produced desirable results. Development of the application will continue with for future update and further enhancements.

**Appendix B – Source Codes**

import { Component, OnInit } from '@angular/core'; import { HttpClient } from '@angular/common/http'; import { Router } from '@angular/router';

import { AuthService } from './../../services/auth.service'; import { StorageService } from './../../services/storage.service'; import { AuthConstants } from '../../config/auth-constants'; import { ToastService } from './../../services/toast.service';

@Component({ selector: 'app-login',

templateUrl: './login.page.html', styleUrls: ['./login.page.scss'],

})

export class LoginPage implements OnInit {

public postData = { email: '',

password: ''

};

constructor(private http: HttpClient, private router: Router, private authService: AuthService, pri vate storageService: StorageService, private toastService: ToastService) { }

ngOnInit() {

}

validateInputs() {

let email = this.postData.email.trim();

let password = this.postData.password.trim(); return (

this.postData.email && this.postData.password && email.length > 0 && password.length > 0

);

}

loginAction() {

if (this.validateInputs()) { this.authService.login(this.postData).subscribe( (response: any) => {

if (response.user) {

// Storing the User data. this.storageService.store(AuthConstants.AUTH, response.user); this.router.navigate(['home']);

} else {

this.toastService.presentToast('incorrect password.');

}

},

(error: any) => { this.toastService.presentToast('Incorrect credentials.');

}

);

} else {

this.toastService.presentToast('Please complete the information above.');

}

}

}

