A RESTAURANT OPERATIONS MOBILE BASED APPLICATION FOR PRIVATELY OWNED RESTAURANTS IN KENYA

Student No: 99914

GROUP A

An Information System Documentation Submitted to the School of Computing and Engineering Sciences in partial fulfilment of the requirements for the award of a degree in Bachelor of Business Information Technology

Date of Submission: January 2021

Declaration statement

I Joseph Nyariki declare that this project has not been submitted to any other University for the award of a Degree in Business Information Technology.

Student Name: _____

Sign: _____ Date:_____

Supervisor Name: _____

Sign: _____ Date: _____

ABSTRACT

Over the years, restaurants have provided an alternative to the daily dilemma of whether to cook food at home or eat outside and as a result making the restaurant industry an essential service provider all over the world. It can be said with certainty that many individuals and organizations who have ventured in the restaurant business have come out successful due to the demand in the sector. However, in modern day, restaurant owners are faced with many challenges to remain relevant and have an edge over their competitors. The main cause of the problems faced can be traced back to the traditional ways of operations, which involves a lot of paperwork within the restaurant. This pen and paper methods have become irrelevant resulting in ever-frustrated customers due to the congestion in waiting lines and unavailable waiters. Waiters and chefs on the other hand are becoming overwhelmed with the work, easily making order errors and producing low quality output. Furthermore, profit margins are on the decline forcing management to go back to the drawing board or even take extreme measures as closing down restaurants. The aim of the study is to develop an application that will facilitate transition of restaurant operations from analog to digital. The system will be structured and designed for execution on android operating devices. The main aim is to increase efficiency and enhance productivity subsequently introducing a possible unique point of sale for privately owned restaurants in Kenya. As a result, the proposal will review alternative systems currently in use, gaps in the systems and some of the possible development technologies that could revolutionize the restaurant industry.

Table of Contents

List of Figures

List of Abbreviations

IDE-Integrated Development Environment.

LCD-Liquid Crystal Display

POS-Point of Sale System.

RFID-Radio Frequency Identification.

Chapter 1. Introduction

1.1 Background Statement

In Both Modern and ancient times people turn to restaurants to either have their breakfast, lunch or dinner making the restaurant industry a critical and essential service provider all over the world. Over the years many individuals and Organizations have ventured into the business hence creating intense competition and while they all aim to gain from their business it is the quality of service, food and customer satisfaction that distinguishes their profit margins in the end. In Terms of Growth of the industry several factors have contributed including a rise in income levels, better hospitality and a variety of cuisines. According to National restaurant association the industry's share of the food dollar has risen from 25% during the ancient times 1955 to 51% in 2019. Households with annual income of at least $70,000 were responsible for 63% of total restaurant spending in 2018 ('The Restaurant Industry', 2019)

Most restaurants still operate using pen and paper methods with minimal automation or none at all. In an ideal restaurant scenario, a Customer enters the facility. He or she is greeted by an usher who often has a diagram of the restaurant floor to show status of the tables. Once settled a waiter tends to the customer by noting down the customer's order on a piece of carbon paper and delivers it to the kitchen for preparation. The waiter has to periodically check in with the kitchen to find out if order is ready. When food is ready , the piece of carbon paper is archives for record keeping by management .This technique of operation results in wastage of time and large amount of tab receipt (Marsic, 2007). Another disadvantage is that waiters have to carry around pads to take orders and have bills organized and synchronized with a particular table, which is overwhelming. Other of problems incurred in operations could be summarized as follows- inability to increase efficiency, productivity and increase marginal profits, Improper coordination of work activities and high operating costs, Inability to incorporate ever-changing menus.

1.2 Problem Statement

In the restaurant industry in order to be relevant and competitive your restaurant has to have the following ;suitable and flexible menu, a unique selling point , proper management ,well equipped and trained stuff and a reputable customer service or Customer experience(*Common Restaurant Problems and Solutions*, 2017). It is evident that the current techniques and methods of operation are gradually becoming incompetent making it difficult to adapt and overcome evolving changes while keeping the factors mentioned above constant. It is challenging and resource consuming to keep on creating new menus considering a menu changes overtimes with discovery of new cuisines. There is need for advanced inventory management to capture the restaurant's ever-growing data such as orders and menu items etc., which would save on operational costs and Time. The waiter-customer interaction is essential for a positive customer experience, hence there is need for efficiency to minimize order time, wait time and errors on customer's order & Billing information. With the current methods It is difficult for management to track activities without a transparent, accurate and reliable source of reference, hence there is inability to efficiently analyze sales and daily operations. Therefore, as a solution, an application with modules such as menu, where the customer can view menu items, select their preferred food of choice and place an order. On the other hand, the server-side (Kitchen) where the chef can modify menu, view incoming orders and notify customer on the status of their food.

1.3 Aim of the project

The main aim is to develop a system that will help restaurant personnel coordinate their activities and Improve the services rendered to achieve customer satisfaction, and for Management purposes such as track business growth, analyze sales and budget for the future.

1.4 Objectives

The specific objectives of the project included:

I). Increasing efficiency by minimizing turnaround time between waiting to be served and actually ordering

II). To implement an interactive e-menu

III). Develop a system with a logical flow that will make the ordering process efficient

IV). To develop a system that will be a unique selling point for a restaurant.

1.5 Justification

Automation is a process that makes work easier. The project embraces and acknowledges technology as a tool that has an impact on the basic factors that contribute to the success or failure of a business. Technology saves on time, in some cases it reduces costs incurred in production and mainly it enhances efficiency. Human beings look for efficiency in everything they do and that is what the project aims to achieve.

1.6 Scope /limitations of the study

The Project will focus on delivering a system that restaurant personnel and customers will interact with inside the restaurant. It entails having a database server to facilitate inventory management and a user interface with various access levels and different functionalities that will run on a mobile app. Effort will be put to deliver a security proof system with necessary encryption to ascertain the integrity and accuracy of data that the system will feed to the stakeholders. A sequential approach will be adopted throughout the development cycle.

Time will be a limiting factor in the early stages of software development cycle. The System aims to deliver a system that will enhance the work effort of the restaurant personnel hence it will not solve the problem of laziness and incompetence of restaurant staff.

Chapter 2. Literature Review

2.1 Introduction

In reference to Restaurant Automation, the main aim is to develop and establish a centralized system with broad functionality that will enable efficient operations in the restaurant space. In this Chapter current systems, techniques and methods of operations will be reviewed and critiqued. The various development technologies and frameworks that were used in the past and present will be discussed and possible breakthrough technologies will be reviewed.

2.2 Description of current restaurant operations

In most restaurants there is a common way of operations and personnel required in order to operate smoothly. We have *the Kitchen*, operated by the main chef and his support chefs. Their job in brief entails receiving orders and preparing the food as requested. Other additional duties such as deciding on cuisines to offer and preparing the menu also fall under their job description. On an ideal day the chefs are usually busy working hand in hand with the waiters receiving orders and dispatching the orders after preparation. This process is quite hectic and can get out of hand if the personnel are not organized.

The waiters. In most restaurants they are usually overwhelmed with the amount of work. They are tasked with receiving orders from customers, where they note down on pieces of carbon paper and deliver to the kitchen. Thereafter it is their responsibility to neatly write down the order on a bill carbon paper receipt book for payment purposes and avail them to the customers after or during their meal. Furthermore, it is their duty to deliver the food when ready from the kitchen to the awaiting customer.

The cashier. The job is to record and account for the payments. There are different ways in which this process is carried out. In most restaurants, the waiter prepares two copies of bill receipt book one for the kitchen and the other for cashier. The bill receipt for the cashier is delivered to him/her by the waiter with the collected payments from the customer. They are recorded on a computer and are archived for references. It is the cashier's job to generate a report for management review and analysis.

Management. They are the main stakeholders in restaurant and what they do is track all the activities in the restaurant on a day-to-day basis. They are responsible for providing the budget,

preparing payroll for each of their staff (in order to do this, they need to have a report on work done by each staff), deciding on what to serve on the menu after consultations with the chef and providing overall supervision to staff.

Finally, we have the customer, who upon entering a restaurant is assigned a table and provided with a menu by the waiter. After deciding on what to order, the customer calls the waiter to take the order. Thereafter when the waiter has already brought the food and the customer is done eating, the bill receipt is requested and payment is sorted.

2.2.1 Problems and challenges experienced with current methods of operation

There are problems and challenges experienced at all stakeholder levels mentioned above and therefore, there is need for eradication of manual paper pen methods as a result of the challenges to be mentioned below;

i) Kitchen Operations

Operations in the kitchen as mentioned earlier can easily become overwhelming and hectic without a form of organization. Incoming orders need to be attended to within a reasonable amount of time and delivered to the customer, for a chef it is tiresome to keep track of orders and the order in which they arrived especially when the orders come in form of carbon paper receipts. The chefs need to make sure they get the order with accuracy and speed in order to have a satisfied customer who would return in future. (Curreri, 2019). The menu is a component of the restaurant that keeps on changing, therefore preparing new menu booklets each time there is a change is resource consuming and inefficient.

ii) Waiter Operations

Waiters normally have the most work to do, they have to constantly move around the restaurant taking orders, keeping track of the orders and their respective tables and eventually returning the orders (Shrivastava, 2019). They also have to generate the bills from the orders they take, whereby they need to refer to the pieces of carbon paper which could become too much work making them inefficient.

iii) Managerial Operations

As the overall supervisors it is challenging to perform tasks such as keeping track of which employee is on duty and for how long they were on duty. As the manager of a restaurant, it is difficult to know how many tables a waiter has served and the amount of work the waiter has done and usually this is critical in deciding the employee's pay. Another issue with the current methods of operation is that it yields an overflow in paperwork and this slows down generation of important reports for analysis.

iv) Customer Experience

A good customer experience is every restaurant's core objective. The main challenge customers experience, is delays in the amount of time it takes before a waiter attends to him/her.

2.3 Review of Related works and Alternative Systems

Modern restaurant owners adopt the use of a tool referred to as Point-of- Sale system. This technology offers an alternative to the pen method and it inhibits overflow of paperwork. These All-in-one systems are considered the best in modern businesses, it combines all the basic hardware and software needed to serve customers while retaining accurate and genuine business records. Most restaurants in Kenya are familiar with the basic version of POS Systems, which are established using a terminal, cash drawer and a touch screen monitor. They are also designed with software to perform certain business tasks. Receipt printers and card readers are also included in the system. In some occasions some POS Systems are sold with additional components that include a printer for the kitchen, terminal for customer display and a wireless table -side order entry. Features mentioned above make it easy to accept payment from customers. This Point-of-sale systems can be designed and structured to suit a specific client and later programmed by end users to suit their needs (Kashima, 2010).

2.3.1 Related works

A restaurant automation system that enhances efficiency through the use of electronic menus where the customer can communicate an Order to a central server which communicates to a Kitchen display and receives a message when order preparation has begun. The central server being also connected to a payment station which then generates a bill on the customers end.(Suthar, 2004)

The implementation of this kind of system would suit the current market, where everyone is always using and relying on gadgets. It is safe to say that over 80% of the activities we carry out as human beings involve a technological gadget. We use our phones to carry out most of our day-to-day tasks for example making calls and sending text messages, receiving and sending emails, performing payments transactions etc. Therefore, by having this kind of system where a customer does not need too much assistance in that the customer can navigate through the system and get the service and product required makes the world a much better place. The disadvantage that would arise in the migration to this kind of operational systems would be the initial cost incurred in acquiring the technological resources. Resources such as the LED screens for display, the servers and the tablets and mobile devices would be a strain for the management in terms of financial resources. Furthermore, there are risks and obligations that come with technological resources, their safety and maintenance are a key factor to consider.

A Radio frequency Identification system of restaurant automation. The system would require a serving device, ordering device, managing device, a RFID Customer and Dish tag. In this case the database will store frequent customer's personalized menu as well as the customer Information and customer's dish codes. The Ordering device, serving device and Managing device are all equipped with a display and a RFID reader. RFID dish tag is used as the code for the dish and RFID Customer tag used as the code for the customer. All the above are connected by an Ethernet (Wei et al., 2006,p. 1)

The use of RFID, personally is a sophisticated idea and would require massive resources to succeed in the restaurant business. Normally, we human beings want things that make life easier and tend to neglect sophisticated systems, applications and programs even though they might be

the solution to our problems. It is a very smart and logical idea but fortunately it is an invention that would take the restaurant industry by storm in the future.

2.3.2 Gaps in existing systems

Although the use of POS Systems has become very popular in the restaurant industry, they are still challenges experienced with the technology it comes with. The following are some of the interrupts that occur when the systems are in use;

i) Flexibility and Mobile Compatibility

As a manager or supervisor of a restaurant it's not always possible to be onsite to monitor day to day activities. Modern high-quality POS systems should be compatible with mobile devices to enable monitoring from any point at any given time.

ii) Usability issues

Some systems can be complex and difficult in terms of user interactions such that restaurant employees struggle to use them optimally. The learning curve eventually reflects on the productivity negatively. It is important to have a user-friendly system that does not require extensive training, which is a strain in a restaurants resource.

iii) Lack of customer Interaction

Customers are the core stakeholders in service-based businesses, therefore, it is significant to involve them in restaurant activities. POS systems lack technologies that can allow customers to view menu at the comfort of their phone, self-ordering and even give immediate feedback

2.4 Review of possible Development Technologies

In the attempts to embrace technology in the restaurant industry, here are some of the development technologies that could change the future;

a. Implementation and Development of an e-restaurant using RFID technologies and WLAN for customer centric service .In a quality high service system the customer should be centered in that customer identity is known and therefore favorite meals and expenditure records are recognized by the system so as to provide customer centric service(Wei et al., 2006,p. 1)

b. The use of high end portable devices to revolutionize a restaurant .In this case customers use and iPad to make orders real-time , eliminating completely the need for a waiter .The restaurant also uses the iPad to control the various displays projected on the restaurants screens .In five star restaurants that have a valet ,the iPad can be used notify the valet when the customer is ready for their car.(O'Grady, 2011,para. 2)

c. Robotic waiter. The robotics technology is gradually replacing the need for human labor throughout the world in various Industries. As a result of customer complaints in regards to congestion during peak hours, manual Order processing and unavailability of waiters a restaurant automation system has been proposed where a waiter robot is used. The Customer places an order through the electronic menu bar. The desired order is transmitted on a wireless network to the kitchen via menu bar which is based on the LCD, Keypad and Bluetooth module. The waiter robot then delivers food from the kitchen to the customer .(Asif et al., 2015,p. 14-15)

2.5 Conceptual framework

The proposed system will have two sides; the client side and the server side. The preferred device for administration is a tablet. The client side is where customers will be able to interact with the system. Each table will have a tablet with the application where the supervisor staff will login and have each tablet ready for use by customer. On the home screen the customer will navigate the menu and add menu items desired to the cart. Once the customer has made an order, the kitchen staff from the server side will be able to see the order with all the details and modify its status once

order is ready. Only an authorized user will be able to login to the server side where he/she can create, update, delete menu items and facilitate order management. Assuming each table will have a table name and table code which will be used as part of the credentials when the supervisor staff is registering tablets for use in the respective tables. The waiter will have a device where they can view the incoming orders and the status of each , where he/she can also modify the status when order is ready and has been picked from the kitchen and on its way to the customer's table.

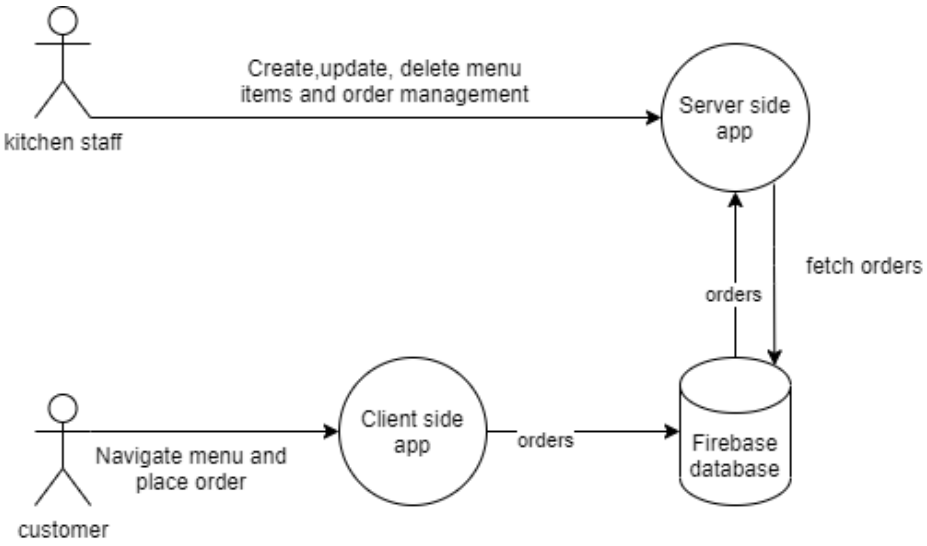


Figure 2-1 Conceptual diagram

Chapter 3. System Development Methodology

3.1 Introduction

This Chapter will review the methodology that will be used for the proposed restaurant system. The focus will be on the type of methodology to be adopted in the development, functional requirements and nonfunctional requirements.

A system development methodology is the process followed by an individual or organization in order to conduct all the necessary steps to analyze requirements, design, implement and maintain Information systems.(Benzzine, 2002) .

The Agile approach of development will suit the development of the proposed system since it advocates for evolutionary development, collaborative effort between the end user and developer and it encourages flexible responses to changes even at late stages of the development cycle. In our case it is important to work side by side with the end users to achieve the desired functionality because the system will play an important role in their productivity

3.2 Agile Software Development Methodology

The Agile approach software development methodology is based on four values which are communication, simplicity, courage and feedback. System analysts should always adopt these values regardless of the project.(Kendall & Kendall, 2011)

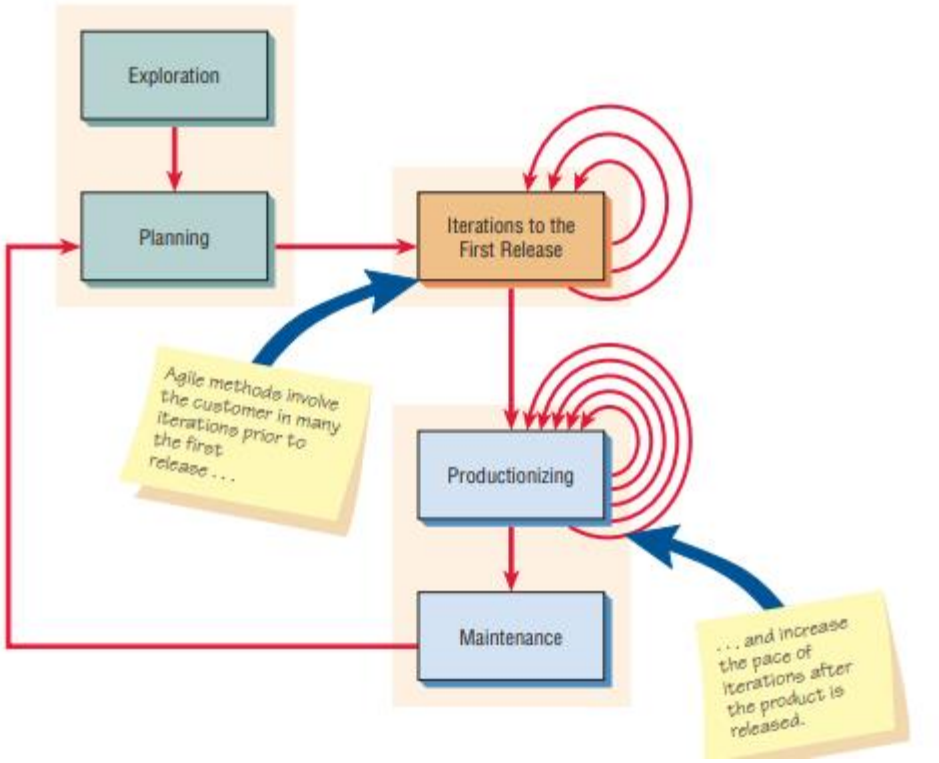


Figure 3-1 Diagram of agile approach as represented by(Kendall & Kendall, 2011)

3.2.1 Exploration

In this stage the environment is explored while potential technologies needed to build the system are examined. The software developer interacts with the customers gathering a draft of their requirements while ideally coming up with an estimate of the time required to complete the tasks discovered in this stage. At the end of this stage the developer has a concept of what is required.

3.2.2 Planning

This is where the end user and the system developer agree on the deliverables and the time it will take to produce a system with solutions to their identified problems. Upon proper understanding of the deliverables the developer and the end user work hand in hand where the developer has to design the simplest solution possible, introduce the system to the end user and get the feedback. It is from there that the developer adapts an initial design. There is need for a good relationship

between end user and developer for this stage to be successful since it is the end users' decisions that will identify the critical functionalities to be focused on by the developer.

3.2.3 Iterations

Iterations are cycles that involve testing the system with end user, getting feedback and implementing change where needed. In this stage the developer is free to manipulate the system functionalities and upon consulting the end user the developer can make adjustments to the predetermined schedule.

3.2.4 Productionizing

The product is released in this stage and feedback is immediate allowing the developer to add features for Improvement

3.2.5 Maintenance

After the sale of a product or service it is important to offer after sale service to the customer in order to enhance customer retention in your business. In system development it is mandatory for the developer to be available and ensure that a system runs smoothly after deployment to the market. In this stage the developer provides user support and at times he/she can consider additional of new features.

3.3 Analysis

System requirement analysis is the process of learning, identifying, determining and documenting end users' expectations and needs of the system to be designed that will solve a particular problem. The Process generates a software requirement document that captures the proposed systems' functionality, design constraints, performance and quality attributes.(Catanio, 2006)

3.3.1 Functional requirements

A functional requirement describes a system by specifying a function that the system must perform where a function is a combination of input, output and behavior that enable a specific component in a system to perform a task.

The proposed system will have the following functionalities;

i). Menu modification- From the server-side an authenticated user will be able to create, update, delete menu items which will subsequently reflect on the client side.

ii). Menu navigation- The customer can navigate from the menu category into the foods, view the food description and add chosen item(s) to cart with ease.

iii). Push Notifications- The client side will receive notifications whenever the respective order is altered from the kitchen end.

iv). Order management – A user form the client side will be able to place an order and view the order under the orders in the navigation drawer. From the server side an authenticated user will be able to view the order and order details and modify the order status once order is ready.

v). The system will have the ability to generate a report of requests made from the database

3.3.2 Non-functional requirements

Non-functional requirements describe the properties that the system must have. They define the quality attribute of a system and ensure that usability and effectiveness of the entire system.

The non-functional requirements include;

i) Audit trail- Orders will be captured independently including date and time.

ii) Customers will need to be in the restaurant to login to the system. Table login credentials will be changed periodically by administrator in order to prevent a customer from logging in from remote location

iii)Orders placed from the client side cannot be altered once modified

iv). Portability and compatibility. The system will run smoothly on all android devices, which is convenient especially for the waiter who will be moving around the restaurant a lot.

v). Critical restaurant data will be secure as the system will incorporate data encryption

3.4 System Design

Defining the structure of interacting components and how they fit each other is what the activity system design entails. Collectively system design is the process where a software engineer defines the structure of a system in terms of modules, architecture, data and interfaces based on the requirements.

3.4.1 Object Oriented Analysis and Design

The Object-Oriented approach of analysis and design is used to facilitate the establishment of systems that are ever changing rapidly in response to dynamic business environments(Kendall & Kendall, 2011). It focuses on object in a particular system in our case the objects being customers, orders, menu items etc. This makes this the best approach for our desired restaurant system.

The approach uses Unified modelling Language to break down the system into a use case model. UML describes software both behaviorally and structurally with graphical notation. Other graphical design modelling languages that will be used for the proposed system include;

i). Data flow diagram -Will be used to show flow of the restaurant data within the system.

ii). Flow chart- a diagram that will be used to demonstrate step by step how a particular problem will be solved

3.5 System development tools and techniques

Android Studio will be used as the IDE for developing the proposed system. Android studio is an Integrated development environment for googles' android Operating system. Android studio is of the best applications for making an android app. It is open source and provides native UI. It uses Java as the programming language, which is class based and Object-oriented.

Firebase database server will be used. Firebase is a google mobile application development platform that assists in building, improving and growing your app. The firebase real-time database is appropriate to use in the proposed system since it allows all the application clients to share instance of the database and automatically receive updates with latest data.

3.6 Methods used to test System

To test the functional and non-functional system requirements black box software testing method will be adopted. It will be the best method to use because the end users of the proposed system are only interested on the behavioral function of the system based on requirement specification. The method does not require user to have knowledge of programming in order to test and it also saves on time.

3.7 Domain of Execution

Th system will be built to run on mobile application this is because android devices such as phones and tablets are portable and easy to use making it convenient for our end user. The use of mobile devices as medium of execution will make transition from traditional paper pen method smooth and relatively cost effective.

3.8 Proposed modules and System Architecture

Modules are components of a system that within them have functionalities. The proposed system will have the following modules;

i) Cart- The customer can add menu items, where the total price and quantity of menu items selected will be displayed

ii) Orders- This module is where according to the access levels the end user can view orders, place orders and modify orders.

iii) Menu -The module will have a scroll view of the restaurant's menu and according to level access end user can view, create, update, delete menu items.

Chapter 4. System Analysis, Design and Architecture

4.1 Introduction

The chapter will highlight the approaches that were employed and provide a list of the identified system requirements (functional and non-functional) during the system analysis phase.

4.2 Requirements Gathering

This involved gathering information about the key aspects of daily operations in a restaurant. In an attempt to gather qualitative system requirements, interviews and questionnaires were administered to an experienced restaurant staff in one of the fast-food restaurants around. Furthermore, an IT information systems expert gave insight on the approaches and the best way to build the system.

Interaction and consultation with the restaurant expert through one-on-one interviews provided crucial first-hand information on what is actually important in the restaurant business process. It provided a clear structure on how process flow is and how the system structure would look like.

The collective feedback and requirements from both the experts were logical flow of the operation in the system, proper communication between the client-side and server-side, accountability and traceability of orders made in the system, report generation for analysis and references.

4.3 System Requirements

4.3.1 Functional Requirements

The system's functional requirements are as listed in the table below.

FR1	The system should allow authorized staff to sign in into client-side.
FR2	The system should allow staff to create, update and delete menu items.
FR3	The system should allow staff to view incoming orders as received from client-side.
FR4	The system should allow staff to modify status of orders.
FR5	The system should allow customer to view menu category, menu item and food description.
FR6	The system should allow customer to add items to cart and place an order.
FR7	The system should allow customer to view the order made and current status of the order.
FR8	The system should allow user to request for password through a secure code in case user forgot password.

4.3.2 Non-Functional requirements

The system's non-functional requirements are as listed in the table below

NFR1	The system should allow user at both client and server-side to navigate easily through the navigation drawer
NFR2	The system should allow only users with authorized staff ID to login to server side
NFR3	The system should show a timestamp on orders made

4.4 System Architecture

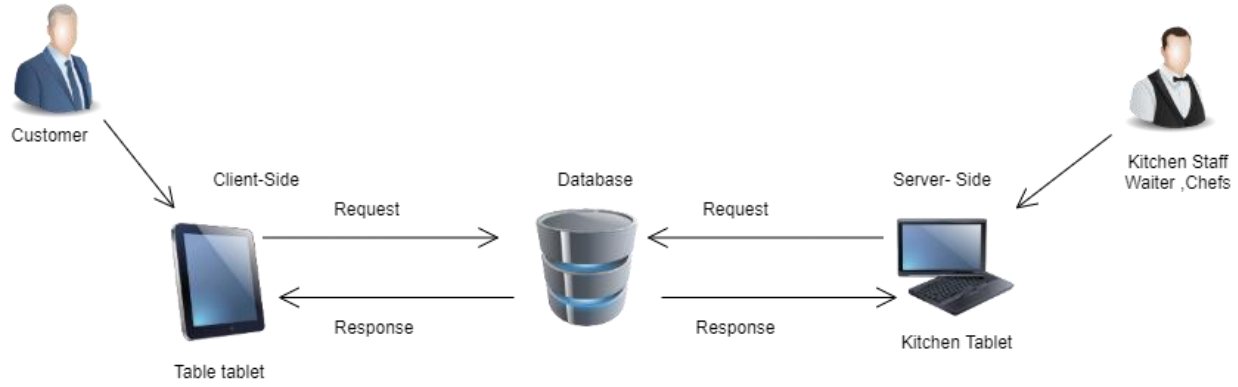


Figure 4-1 System Architecture

The system architecture compromises of Client side which is accessed through a tablet. This is where the customer navigates through the menu items and places an order. Order details are stored in a request in the database. The Kitchen Staff can access the system through the Kitchen tablet on server-side, where they can manage the menu and manage orders.

4.5 System Designs

4.5.1 Use Case Diagram

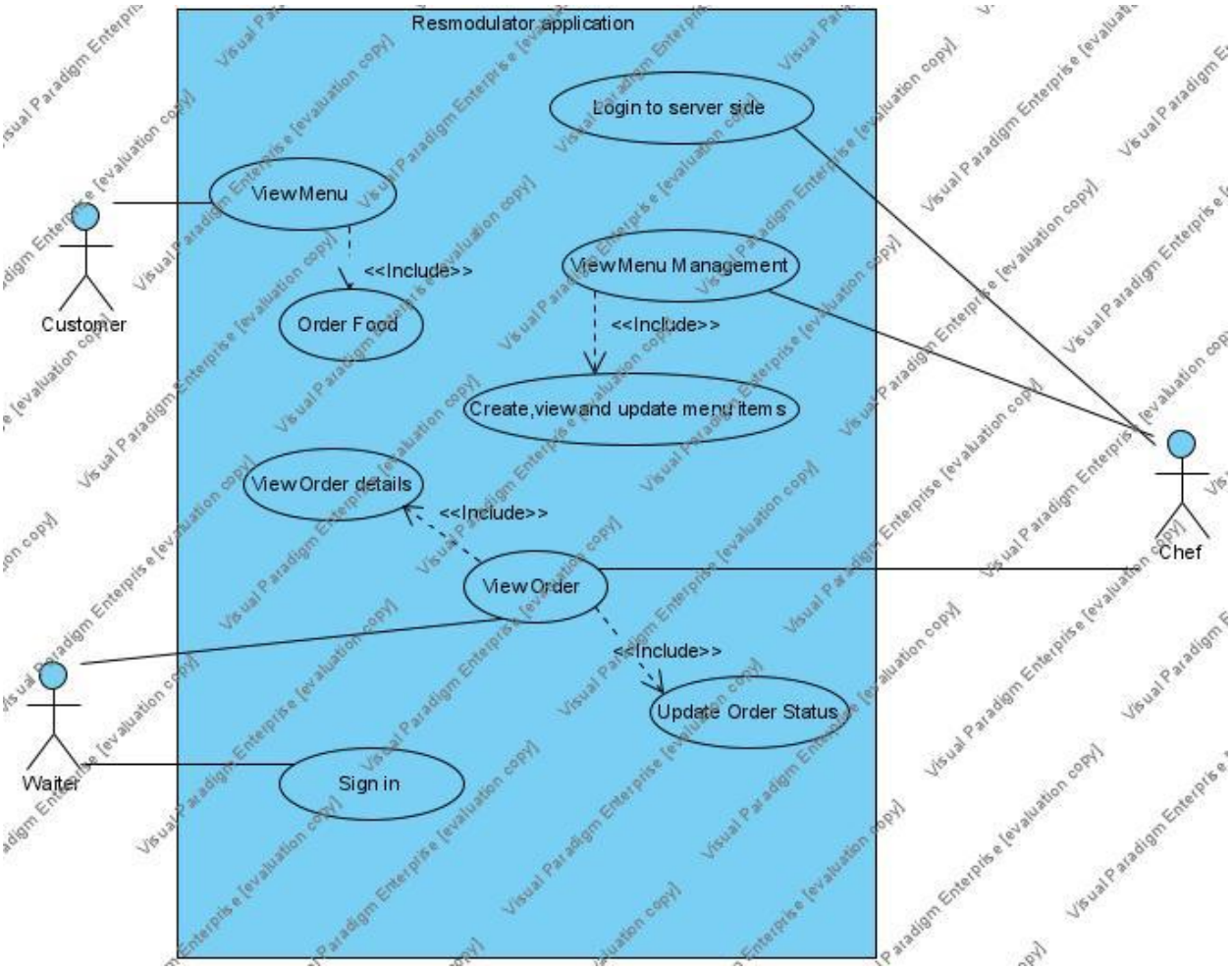


Figure 4-2 Use Case Diagram

The Use case above shows how the users of the application will interact with the system

4.5.2 Sequence Diagram

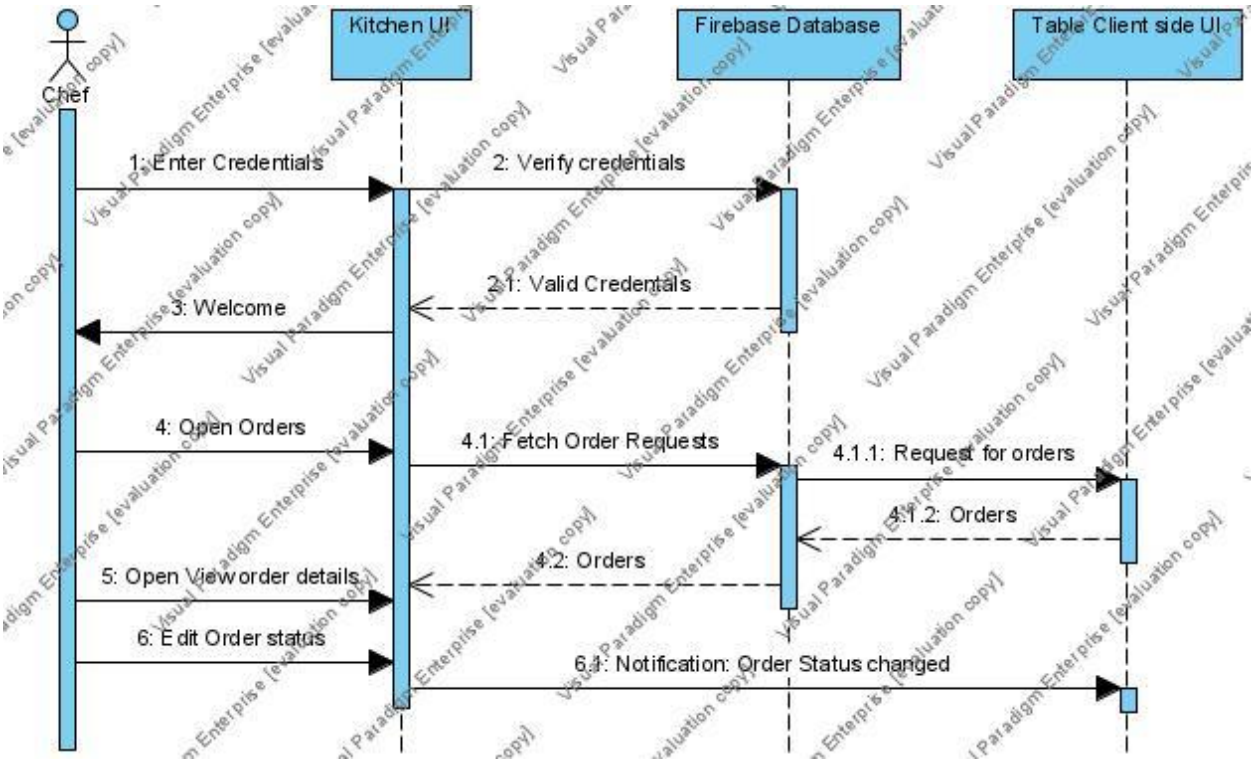


Figure 4-3 Chef Sequence diagram

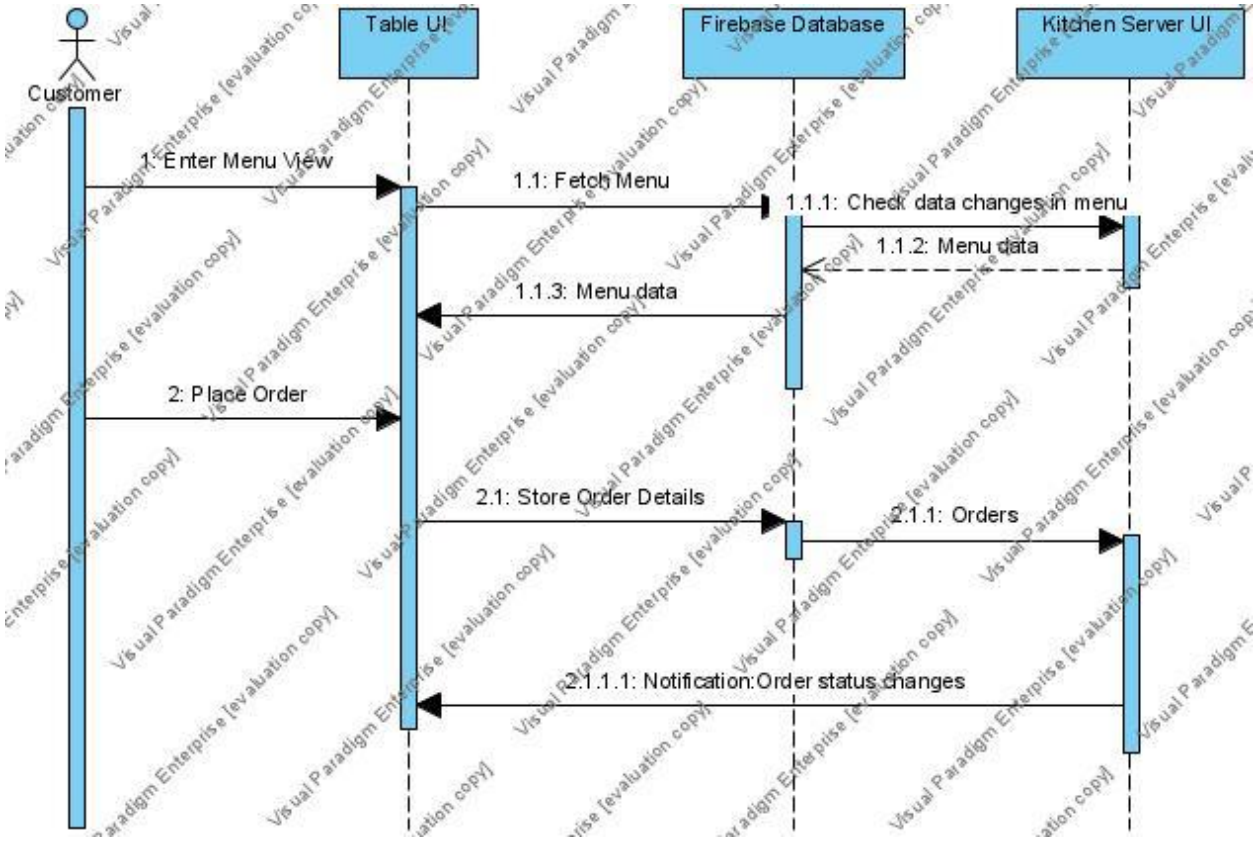


Figure 4-4 Customer Sequence Diagram

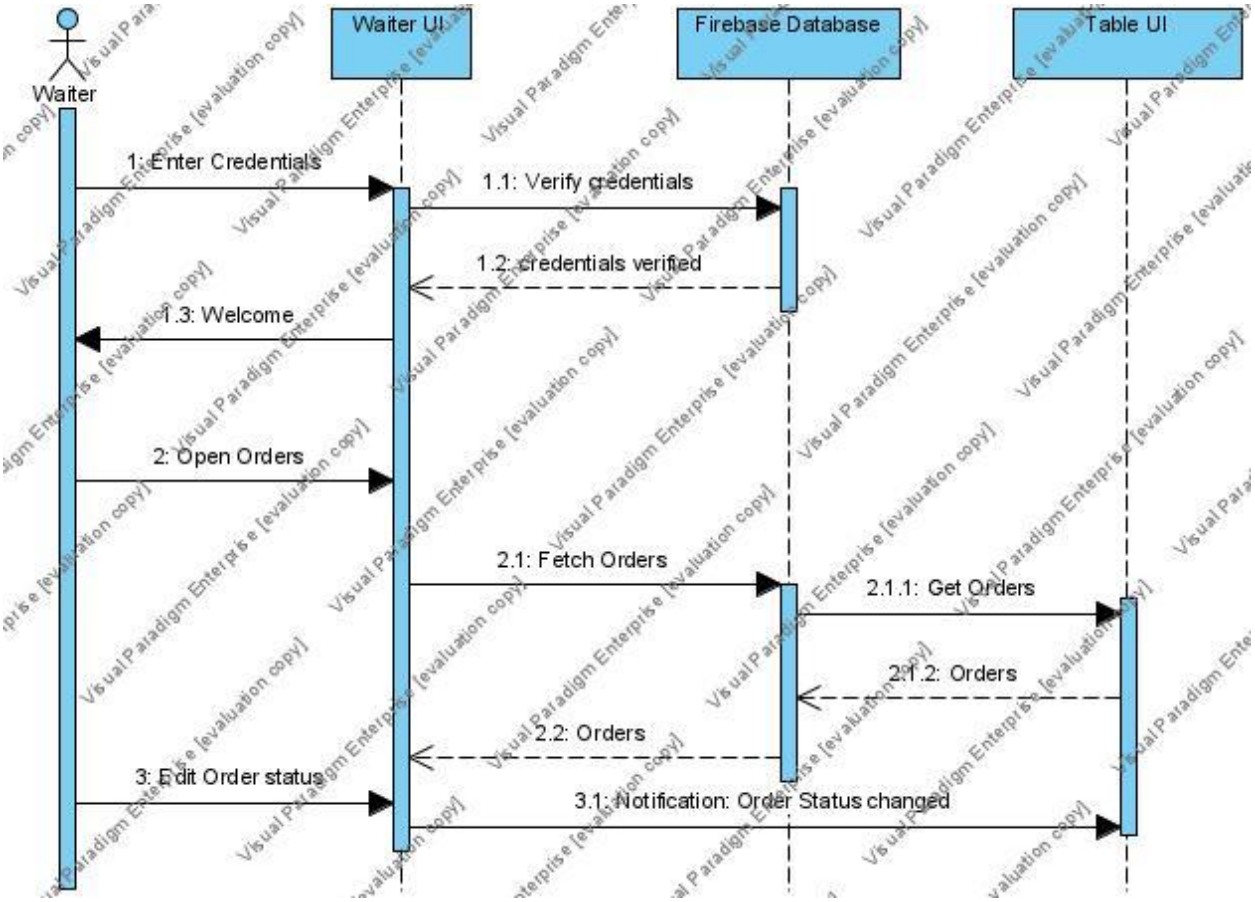


Figure 4-5Waiter Sequence Diagram

4.5.3 Class Diagram

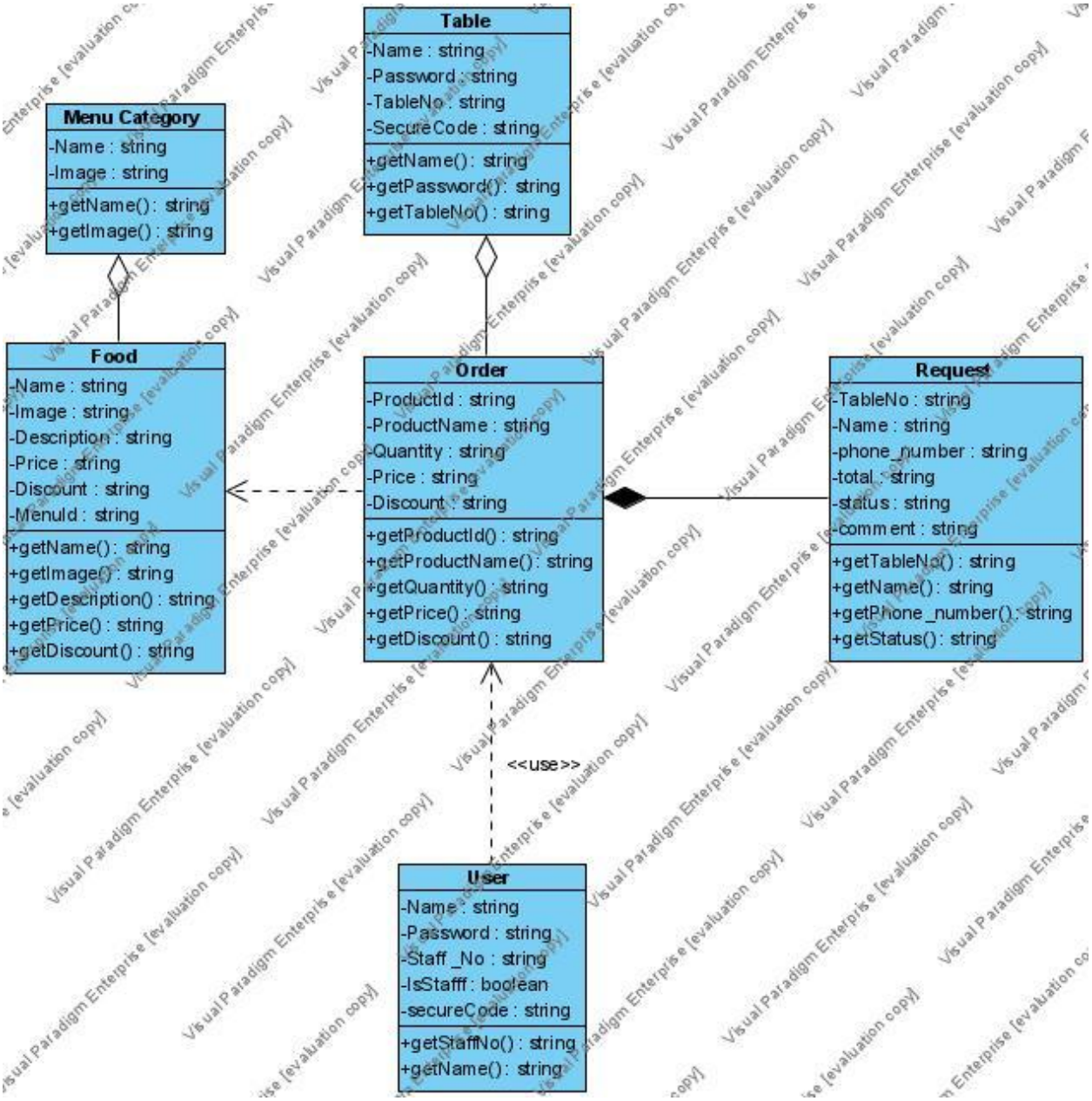


Figure 4-6Class diagram

4.5.4 Entity Relation Diagram

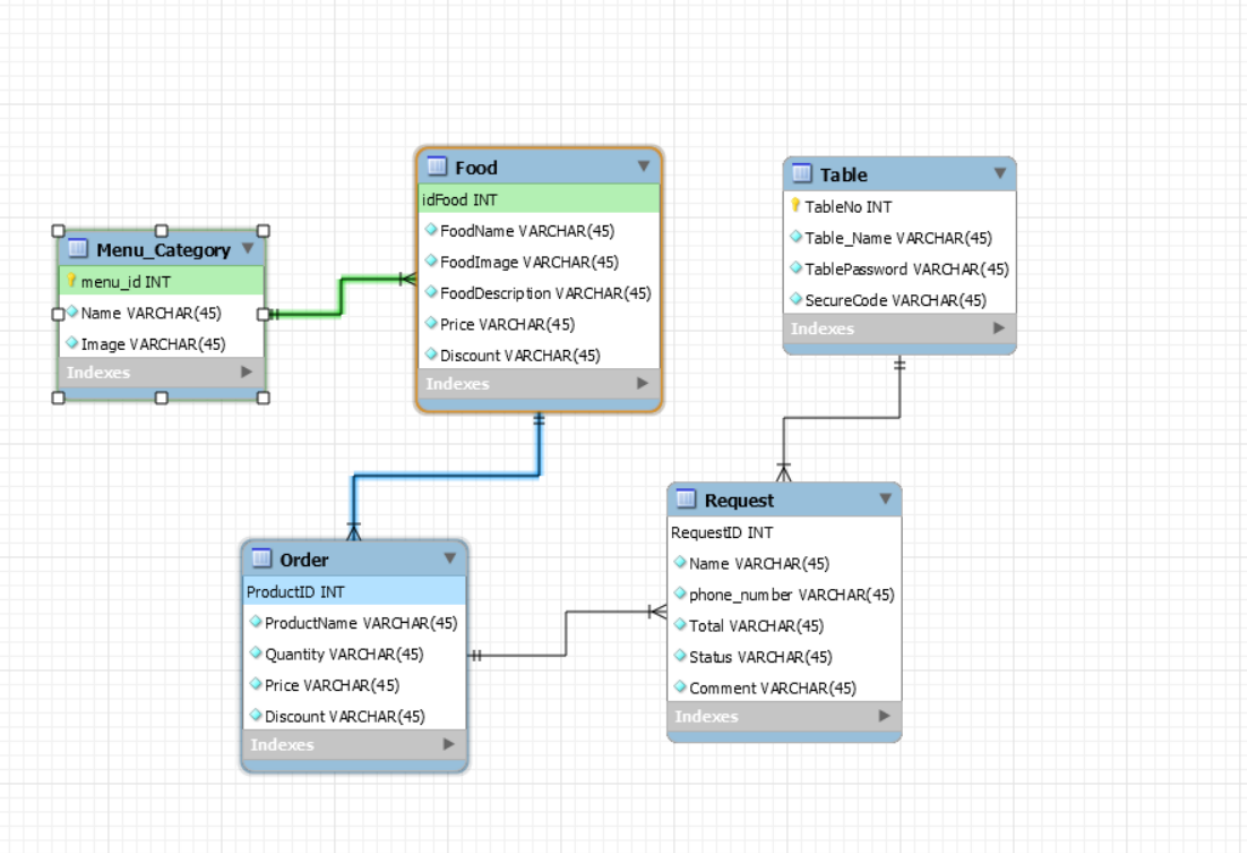


Figure 4-7 Entity relation diagram

Chapter 5. System Implementation and Testing

5.1 Introduction

In this chapter for the purpose of system development it will cover more on what the system actually entails, how the system was developed and the role of various sections of the system. Additionally, it will focus on testing, where the system's functional requirements are compared with the actual functionality of the system. The chapter aims to provide assurance that the system's objectives were met as defined earlier in chapter 1.

5.2 System Implementation

The MVC architecture formed the foundation in which the system's construction and development began. The Model-View-Controller is an architectural plan that divides an application into three main logical components which handle specific development aspects of the application. The Model deals with all the data-related logic that the user interacts with, the view is used for User Interface logic and the controller acts as an interface between Model and the view.

5.2.1 Model

Initially dummy data was obtained from an open-source repository known as meal DB, which holds data about food and their corresponding image and the data structured using JSON Online Object Generator. As a result, the structure of classes category and food were accurately defined using notepad and thereafter the data imported into Firebase Realtime database.

Figure 5-1 Snapshot of dummy data used

5.2.2 View

Android studio provided all the necessary libraries and packages that was needed to create an interactive user-friendly interface. Below are snapshot images of the various activities that the user interacts with both from the client-side and server-side.

Figure 5-2Client-Side Home Menu

Once logged in this is where the customer can navigate and choose what category of food, he or she wants food from.

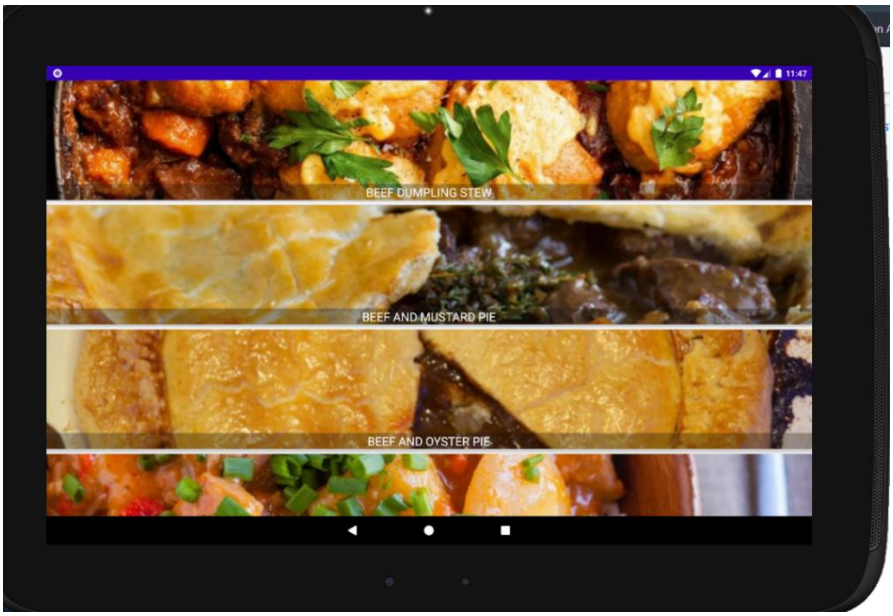


Figure 5-3Client-Side Foods scroll view

Once user has selected a category for example the image above shows foods under the category meat.

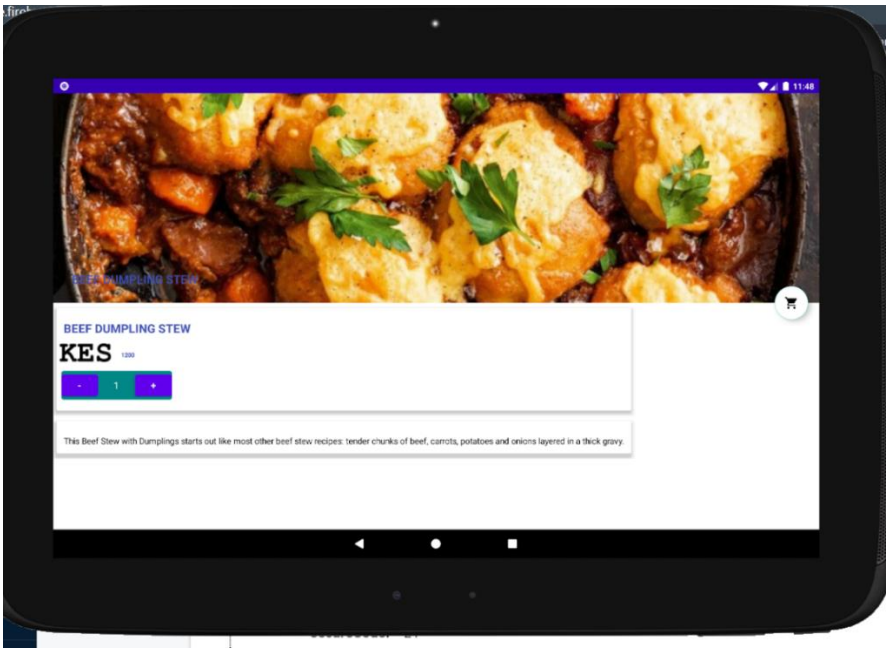


Figure 5-4Client-Side Food Description

Upon making a decision on what food to eat, this is where the user can see the price and description and eventually can add food to cart.

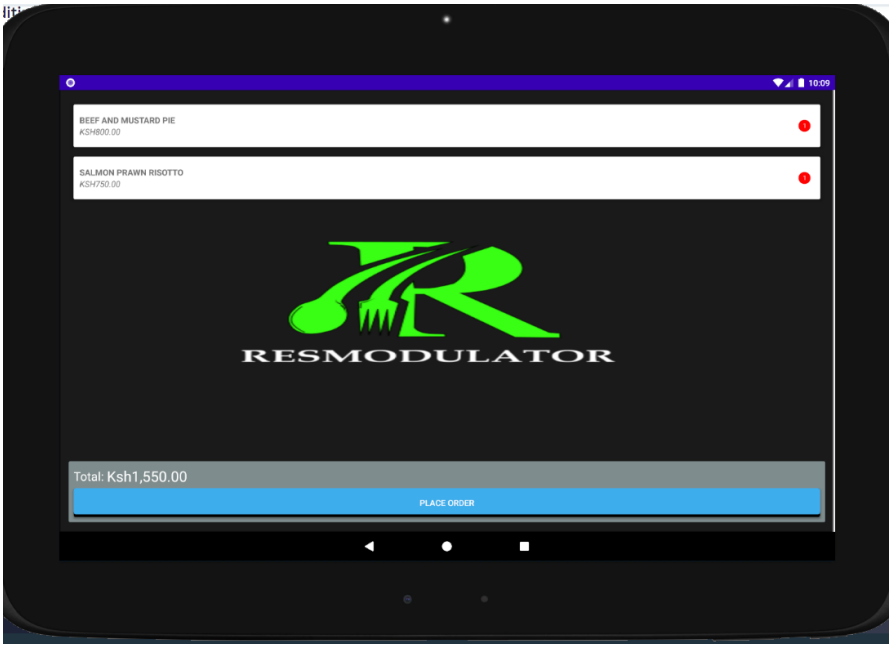


Figure 5-5 Client-side cart view

Once items have been added to cart the customer can navigate to the cart module where he will see all the items chosen and total price

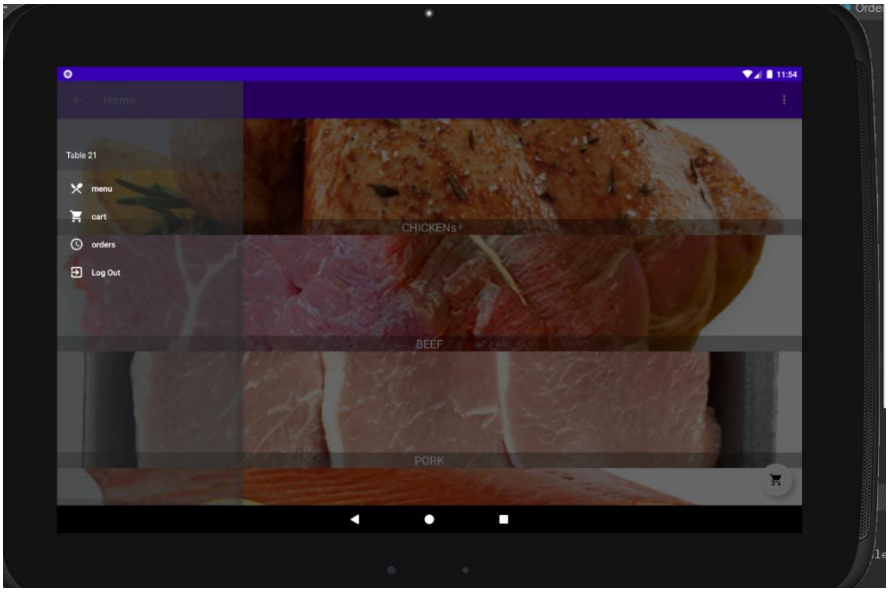


Figure 5-6Client-Side Navigation Drawer

After placing an order, the customer can use the navigation drawer to navigate to orders where her will see his or her order.

Figure 5-7 Client-side order module

Inside the orders module the customer can see the order made and keep track of the status when altered by kitchen staff.

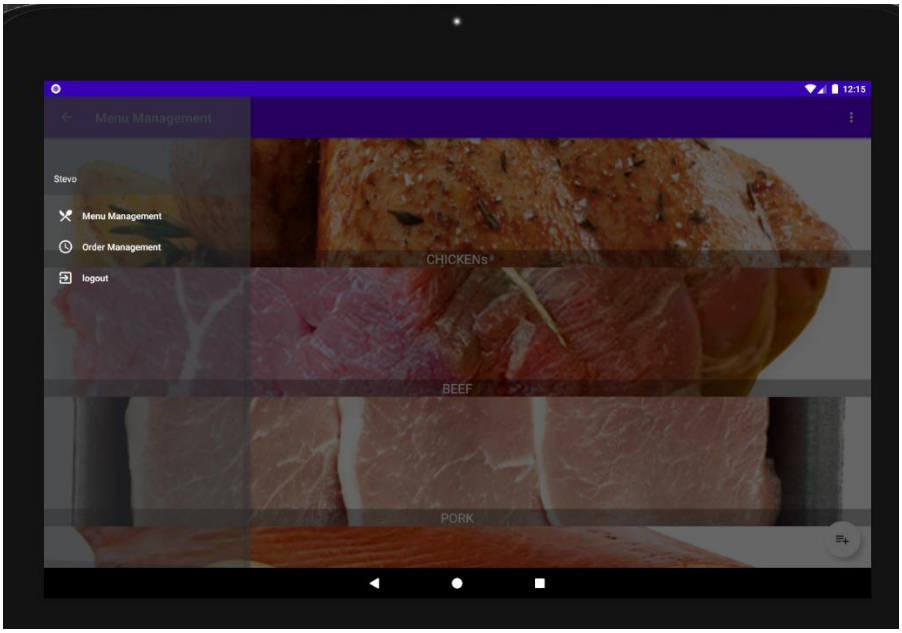


Figure 5-8 Server-Side Navigation drawer

Once a kitchen staff logs in successfully this is the home page view where he can manage menu items and manage orders.

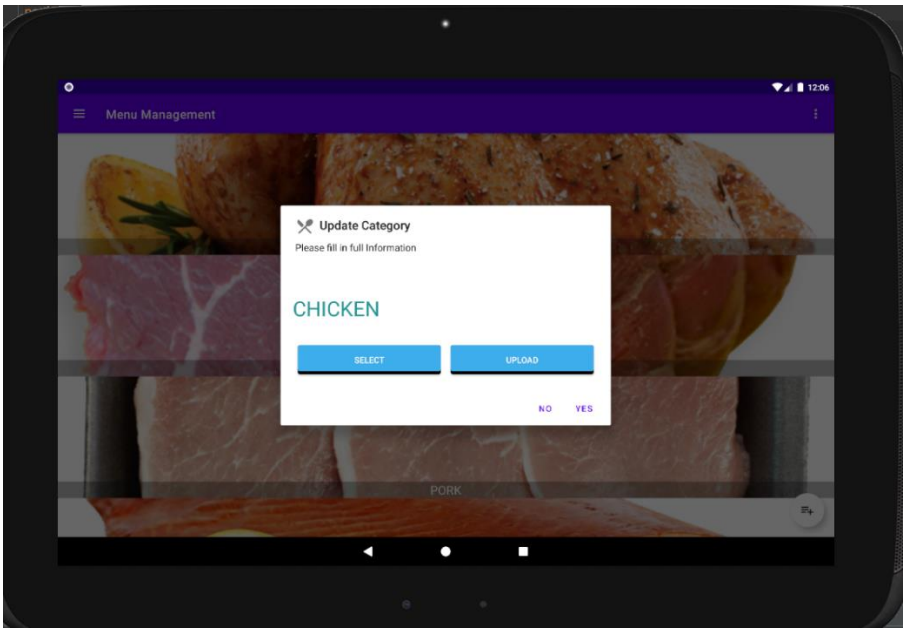


Figure 5-9 Server-side add category

The staff can add new categories into his menu with the dialog. The dialog prompts user to enter a name and afterwards select an image from device directory and upload it into the menu.

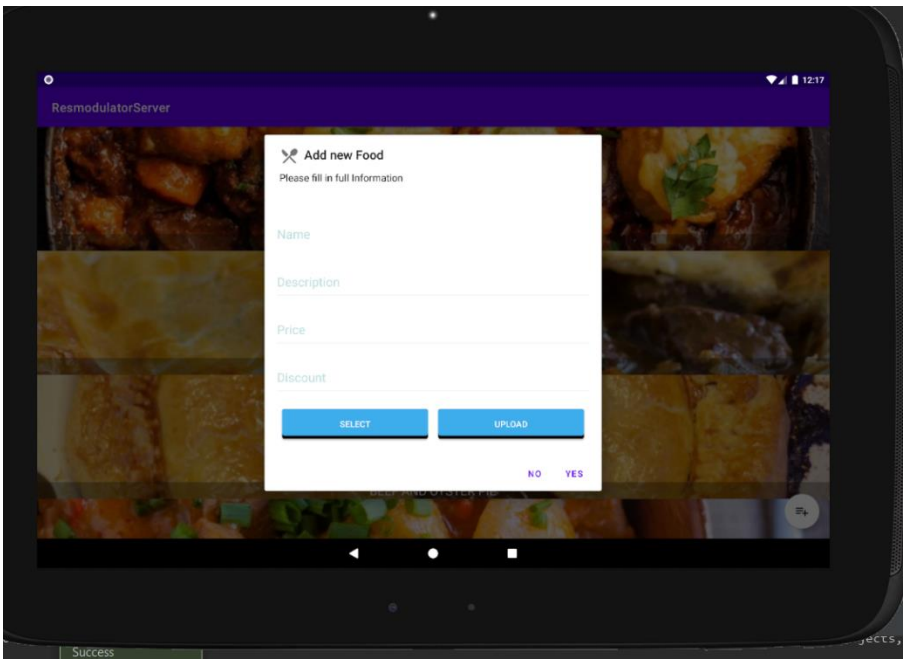


Figure 5-10Server-Side add new foods

The user from the server side can proceed to add food items, where he will be prompted to enter the food name, description, price and discount. Just like when adding menu category, he will select an image and upload it into the menu.

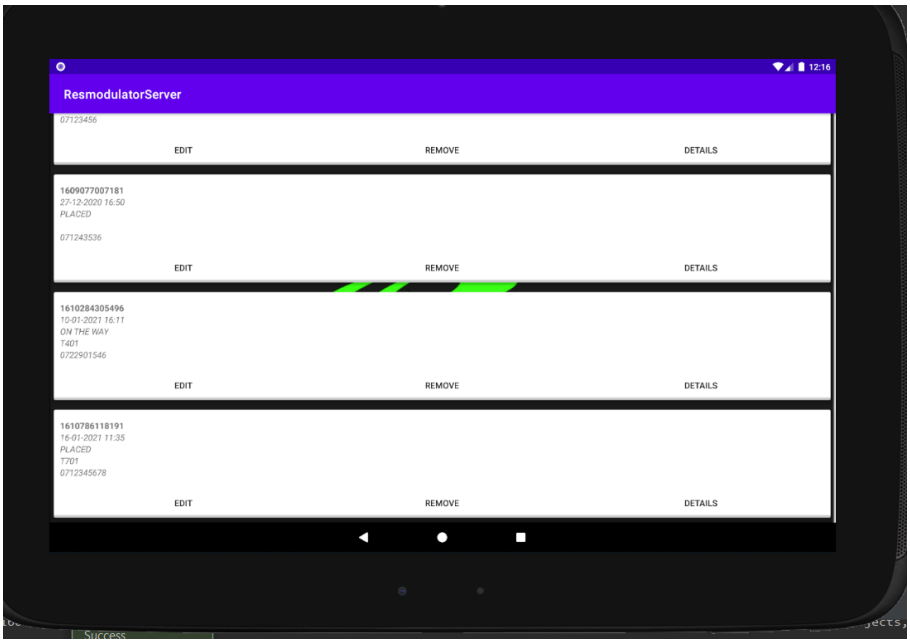


Figure 5-11Server-side Order Management

Inside the order management the user can see all the orders from the client-side, the date and time of order, the table number the order is coming from, phone number of the customer and the status of the order.

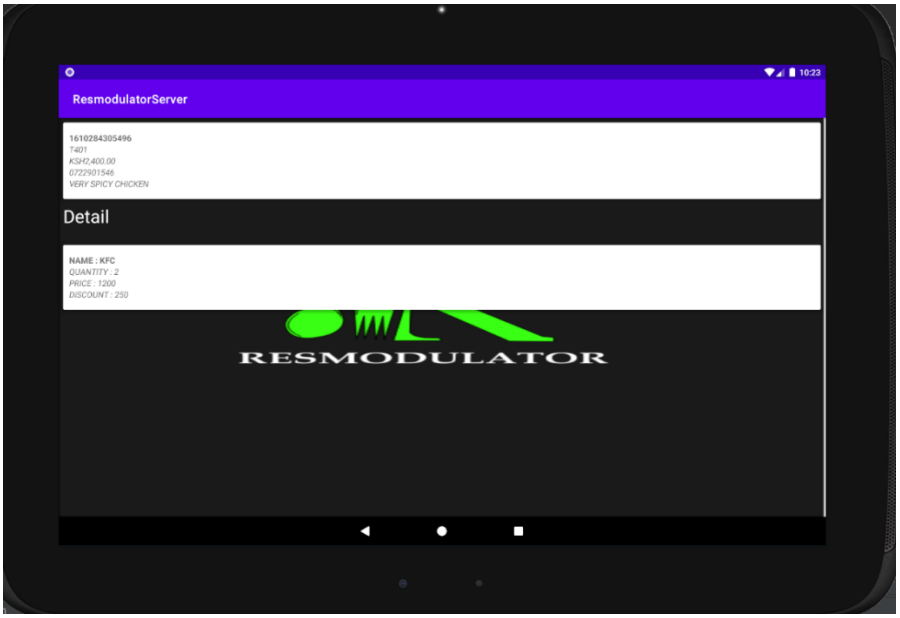


Figure 5-12Server-side Order details

When the kitchen staff wants to see the actual details of the actual order such as name of food and quantity he can click on details and view all the required details.

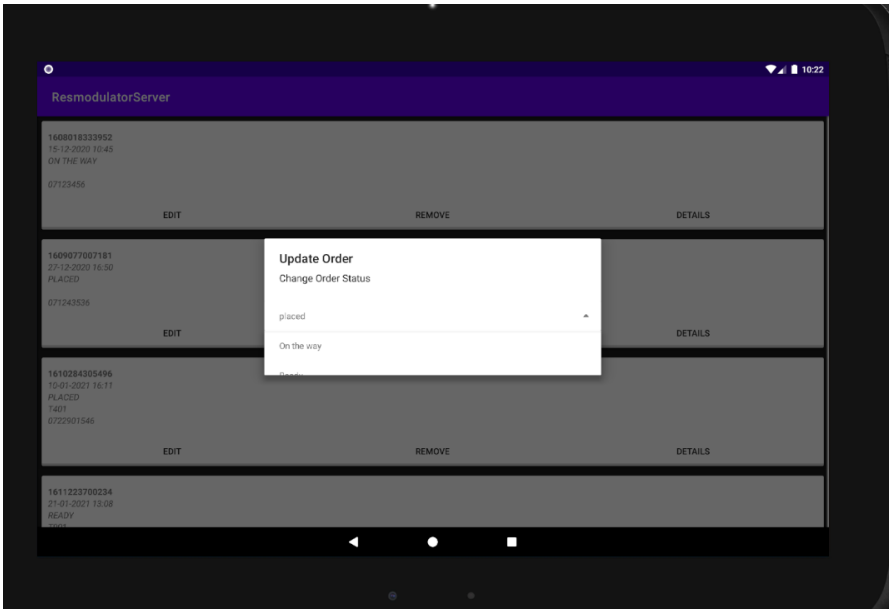


Figure 5-13Server-side update order

Once the order is ready the kitchen staff can then click edit and update the order status to ready, when the food is ready and the waiter can set it to on the way.

5.2.3 Controller

The system had to fulfill various functionalities and the logic behind was coded using Java. The figures below will show how the logic was implemented for the various functions within the system.

i) Notification Management- Firebase messaging was used to declare, implement and execute the notification service and a firebase Id Service class was used to create tokens which were used to distinguish between client-side and server-side.

```
@RequiresApi(api = Build.VERSION_CODES.O)
private void sendNotificationAPI26(RemoteMessage remoteMessage) {
    RemoteMessage.Notification notification = remoteMessage.getNotification();
    String title = notification.getTitle();
    String content = notification.getBody();

    Intent intent = new Intent( packageContext: this, OrderStatus.class);
    intent.putExtra(Common.STAFF_NO_TEXT, Common.currentUser.getStaffNo());
    intent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
    PendingIntent pendingIntent = PendingIntent.getActivity( context: this, requestCode: 0,intent,PendingIntent.FLAG_ONE_SHOT);
    Uri defaultSoundUri = RingtoneManager.getDefaultUri(RingtoneManager.TYPE_NOTIFICATION);

    NotificationHelper helper = new NotificationHelper( base: this);
    Notification.Builder builder = helper.getResModulatorNotification(title,content,pendingIntent,defaultSoundUri);
    // Generate  random Id  for notification to show all notification
    helper.getManager().notify(new Random().nextInt(),builder.build());

}

private void sendNotification(RemoteMessage remoteMessage) {
    RemoteMessage.Notification notification = remoteMessage.getNotification();
    Intent intent = new Intent( packageContext: this, MainActivity.class);
    intent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
    PendingIntent pendingIntent = PendingIntent.getActivity( context: this, requestCode: 0,intent,PendingIntent.FLAG_ONE_SHOT);
```

Figure 5-14 Creation of Notifications

ii) Menu Management- In order to enable staff to create, update or delete menu items, a dialog was created and inside the dialog edit texts were created, to enable input. Buttons were also created. A select button to enable user to select an image of a food item that he/she would like to add and another upload button that would be used to upload the image into the menu layout

```
private void showDialog() {
    AlertDialog.Builder alertDialog = new AlertDialog.Builder( context: Home.this);
    alertDialog.setTitle("Add new Menu Category");
    alertDialog.setMessage("Please fill in full Information");

    LayoutInflater inflater = this.getLayoutInflater();
    View add_menu_layout = inflater.inflate(R.layout.add_new_menu_layout, root: null);

    edtName = add_menu_layout.findViewById(R.id.edtName);
    btnSelect = add_menu_layout.findViewById(R.id.btnSelect);
    btnUpload = add_menu_layout.findViewById(R.id.btnUpload);

    // on click event for button
    btnSelect.setOnClickListener(new View.OnClickListener() {...});
    btnUpload.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) { uploadImage(); }
    });

    alertDialog.setView(add_menu_layout);
    alertDialog.setIcon(R.drawable.ic_baseline_restaurant_menu_24);

    // set the button
    alertDialog.setPositiveButton( text: "YES", new DialogInterface.OnClickListener() {...});
    alertDialog.setNegativeButton( text: "NO", new DialogInterface.OnClickListener() {...});
    alertDialog.show();
```

Figure 5-15 Creation of editable dialogs for menu management

iii) Loading Menu Items from Database- to be able to load the menu with the images and detailed information of the various menu items a view holder was created, which aided in storing the attributes of the menu items to be displayed in the interface. A load menu method was created which basically used data from the respective view holder to load the menu items into the layout.

```
private void loadMenu() {
    adapter = new FirebaseRecyclerAdapter<Category, MenuViewHolder>(
            Category.class,R.layout.menu_item,
            MenuViewHolder.class,categories
    ) {
        @Override
        protected void populateViewHolder(MenuViewHolder menuViewHolder, Category category, int i) {

            menuViewHolder.txtMenuName.setText(category.getName());
            Picasso.with(Home.this).load(category.getImage())
                    .into(menuViewHolder.imageView);

            menuViewHolder.setItemClickListener(new ItemClickListener() {
                @Override
                public void onClick(View view, int position, boolean isLongClick) {
                    //send category id and start new activity
                    Intent foodList = new Intent( packageContext: Home.this,FoodList.class);
                    foodList.putExtra( name: "CategoryId",adapter.getRef(position).getKey());
                    startActivity(foodList);

                }
            });
```

Figure 5-16 Loading Menu Items

iv) Order Management- Just like in menu management, a view holder class was created to store the order details once a customer places the order from the client-side. Once the data was in the view holder, a method to load orders was created, which retrieved the data from the order view holder and display the data in the server-side together with modified view such as edit button and details button. For the edit button an onclick method was created and a dialog created inside to enable user to alter status order. On the other hand, for the details button an Intent to redirect to an order details layout was created, where the user could view the actual order details such as the food that was ordered, price etc.

```
private void loadOrders() {
    adapter = new FirebaseRecyclerAdapter<Request, OrderViewHolder>(
            Request.class,
            R.layout.order_layout,
            OrderViewHolder.class,
            requests
    ) {
        @Override
        protected void populateViewHolder(OrderViewHolder orderViewHolder, Request request, int position) {

            orderViewHolder.txtOrderId.setText(adapter.getRef(position).getKey());
            orderViewHolder.txtOrderStatus.setText(Common.convertCodeToStatus(request.getStatus()));
            orderViewHolder.txtOrderPhone.setText(request.getPhone_number());
            orderViewHolder.txtTableNo.setText(request.getTableNo());
            orderViewHolder.txtOrderDate.setText(Common.getDate(Long.parseLong(adapter.getRef(position).getKey())));

            //new event button
            orderViewHolder.btnEdit.setOnClickListener(new View.OnClickListener() {
                @Override
                public void onClick(View v) {
                    showUpdateDialog(adapter.getRef(position).getKey(),adapter.getItem(position));
                }
            });
            orderViewHolder.btnRemove.setOnClickListener(new View.OnClickListener() {
                @Override
                public void onClick(View v) { deleteOrder(adapter.getRef(position).getKey()); }
            });
            orderViewHolder.btnDetails.setOnClickListener(new View.OnClickListener() {
                @Override
```

Figure 5-17Loading orders to server-side

v) Cart creation- an adapter was created to store all items added to cart by a particular user at any given time. Inside the adapter, a method called get item count was created to calculate the total price of each item given quantity and price of the item and display the total price of all items within the cart.

```
public CartAdapter(List<Order> listData, Context context) {
    this.listData = listData;
    this.context = context;
}

@NonNull
@Override
public CartViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {
    LayoutInflater inflater = LayoutInflater.from(context);
    View itemView =  inflater.inflate(R.layout.cart_layout,parent, attachToRoot: false);
    return new CartViewHolder(itemView);
}

@Override
public void onBindViewHolder(@NonNull CartViewHolder holder, int position) {

    TextDrawable drawable = TextDrawable.builder()
            .buildRound( s: ""+listData.get(position).getQuantity(), Color.RED);
    holder.img_cart_count.setImageDrawable(drawable);

    Locale locale = new Locale( language: "en",  country: "KE");
    NumberFormat fmt = NumberFormat.getCurrencyInstance(locale);
    int price = (Integer.parseInt(listData.get(position).getPrice()))*(Integer.parseInt(listData.get(position).getQuantity()));
    holder.txt_price.setText(fmt.format(price));
    holder.txt_cart_name.setText(listData.get(position).getProductName());

}
```

Figure 5-18Cart creation

vi) Loading food details from the database- Picasso an image downloading and caching android library was used to fetch the image URL from the database and display it in the interface. A food view holder class was created to hold the food information details and the attributes were then called inside a method called get food detail.

```
private void getDetailFood(String foodId) {

    foods.child(foodId).addValueEventListener(new ValueEventListener() {
        @Override
        public void onDataChange(DataSnapshot dataSnapshot) {
            currentFood = dataSnapshot.getValue(Food.class);
            // set food image
            Picasso.with(getBaseContext()).load(currentFood.getImage())
                    .into(food_image);

            collapsingToolbarLayout.setTitle(currentFood.getName());

            food_price.setText(currentFood.getPrice());

            food_name.setText(currentFood.getName());

            food_description.setText(currentFood.getDescription());

        }

        @Override
        public void onCancelled(DatabaseError databaseError) {

        }
    });
}
```

Figure 5-19 Fetching food data

5.3 System Testing

This section aims to focus on what the system does, mainly the focus will be on system testing and assessment on whether the system functions were successful.

Test ID	Related requirement	Inspection check	Pre- condition	Test data	Priority level
1	FR1	Does the system allow staff to register a table and sign in to client side?	A restaurant table must have a name and table no, to be used as credentials to sign in for customer, so that customer finds the app on home screen	Table name: Table 21 Password: *****	Medium
2	FR2	Can the user form the server side create, update and delete menu items?	Device must have images of the food item(s) that user wants to add.	Images of food	High
3	FR3	Does the system allow staff to view incoming orders from the client side?	Authorized user must be signed in and a customer must	Order data	High

			have placed an order		
4	FR4	Can the staff from the client side alter status of an order?	Authorized user must be signed in and a customer must have placed an order	Order data	High
5	FR5	Can the customer from the View the menu category, menu item and food description	A restaurant staff must have logged in to client side with the table credentials	Food data from the database	Medium
6	FR6	Can customers add items to cart?	A restaurant staff must have logged in to client side with the table credentials	Food data from the database	High
7	FR7	Can the customer view the order made and its status after placing the order	Customer must have placed an order.	Order data	Medium
8	FR8	Can a user request for his password if forgotten, using a secure code that was entered during registration	user must remember his or her secure code.	User data	Low

Chapter 6. Conclusions, Recommendations and Future Works.

6.1 Introduction

The aim of this chapter is to summarize in a discussion on the system deliverables, conclusions related to the deliverables and recommendations. In addition, the chapter will mention on additional modules that will enhance the system functions under future works.

6.2 Discussion

As mentioned in the problem statement in the first chapter the system aimed to embrace technology in the operations within a restaurant. The system in the end was able to incorporate an editable menu in which the user from the client-side can view menu category for example meat, fish, chicken. Upon selecting a category, the user can see all the various foods under that specific category. On each food user can see food details. In the end the system was able to deliver an attractive interactive menu where the customer can view and choose food from and on the other hand the chef can add new foods and update existing foods in the menu.

To facilitate the ordering process in a restaurant, the system delivered with an order management module where from the server-side application the kitchen staff can view the order with the details and subsequently alter the status of the order when the order is ready. From a browser the restaurant supervisor can see a list of all requests and generate Pdf reports from the list. In the end the system achieved a logical flow that a customer could easily navigate through from viewing menu items, adding the menu items to a cart and eventually placing an order and on the other hand the chef could view the incoming orders in detail.

6.3 Conclusion

In the process of implementing the system it was noted that various businesses have also adopted use of applications to facilitate their business process. An example is Mr. Price clothing store which has tablets installed in their checkout to process payments. This shows that technology is evolving and is crucial for an organization’s accuracy and efficiency.

6.4 Recommendation

It was noted that in order to establish the system on a restaurant, the restaurant needed to invest on technological resources such as tablets which are to be installed on each table. In addition, the restaurant table need to have names and numbers that would be used as part of the authentication functionality of the system. For security purposes the restaurant would need to change the table details once in a while.

6.5 Future works

The system in the beginning aimed to aid in restaurant operations which also include payments. The future work involves adding a payment module in which the system would be integrated with M-Pesa API to enable a customer to make the payment through his or her phone. The system would use the customer phone number to integrate with M-Pesa which would result in a pop-up on the customer phone requesting him to input his or her pin to process the payment. As a result of adding the payment module a proper reporting module would need to be developed to generate reports of payment information.

REFERENCES

Asif, M., Sabeel, M., & Mujeeb-ur-Rahman, K. Z. (2015). Waiter robot–solution to restaurant automation. *Proceedings of the 1st Student Multi Disciplinary Research Conference (MDSRC), At Wah, Pakistan*, 14–15.

Benzzine, A. (2002). *(System Development methodologies: A framework for comparison)*. http://www.umsl.edu/~sauterv/analysis/488_f02_papers/methodologies.html

Catanio, J. T. (2006). Requirements Analysis: A Review. In T. Sobh & K. Elleithy (Eds.), *Advances in Systems, Computing Sciences and Software Engineering* (pp. 411–418). Springer Netherlands. https://doi.org/10.1007/1-4020-5263-4_64

Common Restaurant Problems and Solutions. (2017, March 21). Glimpse Corp. https://www.glimpsecorp.com/restaurant-problems-and-solutions/

Kendall, K. E., & Kendall, J. E. (2011). *Systems analysis and design* (8th ed). Pearson Prentice Hall.

Marsic, I. (2007). *Software Engineering Project—Restaurant Automation*. https://www.ece.rutgers.edu/~marsic/books/SE/projects/Restaurant/

O'Grady, J. D. (2011). *IPads replace menus at Atlanta eatery*. ZDNet. https://www.zdnet.com/article/ipads-replace-menus-at-atlanta-eatery/

Suthar, Y. (2004). *Restaurant automation system*. Google Patents.

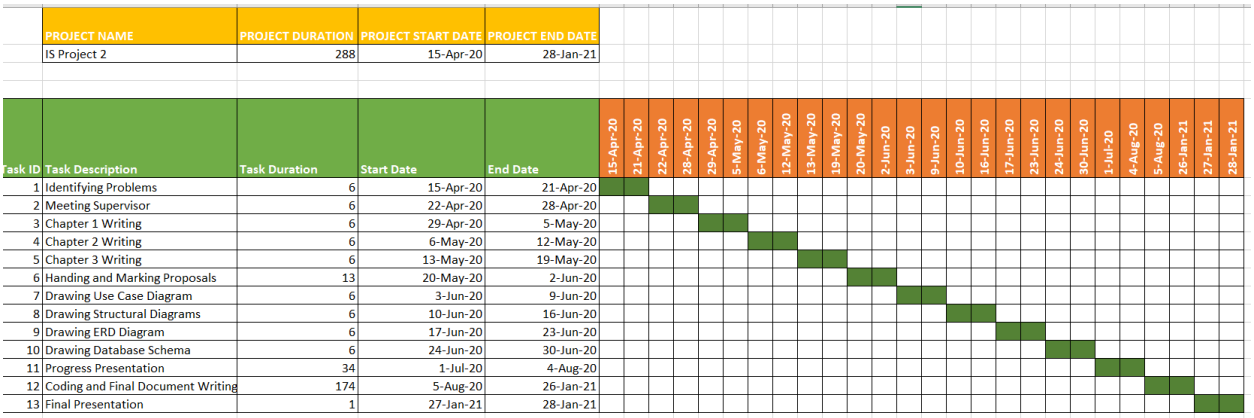
The Restaurant Industry: An Overview. (2019, September 26). *Market Realist*. https://marketrealist.com/2019/09/the-restaurant-industry-an-overview/

Wei, C., Yi-Tang, L., Jen-Chieh, W., & Xherry, C. (2006). *RFID system of restaurant automation*. Google Patents.

APPENDIX

PROJECT NAME	PROJECT DURATION	PROJECT START DATE	PROJECT END DATE
IS Project 2	288	15-Apr-20	28-Jan-21

Task ID	Task Description	Task Duration	Start Date	End Date
1	Identifying Problems	6	15-Apr-20	21-Apr-20
2	Meeting Supervisor	6	22-Apr-20	28-Apr-20
3	Chapter 1 Writing	6	29-Apr-20	5-May-20
4	Chapter 2 Writing	6	6-May-20	12-May-20
5	Chapter 3 Writing	6	13-May-20	19-May-20
6	Handing and Marking Proposals	13	20-May-20	2-Jun-20
7	Drawing Use Case Diagram	6	3-Jun-20	9-Jun-20
8	Drawing Structural Diagrams	6	10-Jun-20	16-Jun-20
9	Drawing ERD Diagram	6	17-Jun-20	23-Jun-20
10	Drawing Database Schema	6	24-Jun-20	30-Jun-20
11	Progress Presentation	34	1-Jul-20	4-Aug-20
12	Coding and Final Document Writing	174	5-Aug-20	26-Jan-21
13	Final Presentation	1	27-Jan-21	28-Jan-21



Project Gannt chart 1