DEVELOPMENT OF A MEAL ORDERING SYSTEM FOR MOUNTAIN TOP UNIVERSITY CAFETERIA

By

DADA IBUKUN OLUWASEMILORE

17010301015

A PROJECT SUBMITTED TO THE DEPARTMENT OF COMPUTER SCIENCE AND MATHEMATICS, COLLEGE OF BASIC AND APPLIED SCIENCES, IN PARTIAL FULLFILLMENT OF THE REQUIREMENTS FOR THE AWARD OF DEGREE OF BACHELOR OF SCIENCE IN COMPUTER SCIENCE

2021

DECLARATION

I hereby declare that this project has been written by me and is a record of my own research work. It has not been presented in any previous application for a higher degree of this or any other University. All citations and sources of information are clearly acknowledged by means of reference.

_____

BALOGUN, JEREMIAH ADEMOLA

_____

Date

CERTIFICATION

This is to certify that the content of this project entitled '**Development of a Meal Ordering System for Mountain Top University Cafeteria'** was prepared and submitted by **DADA IBUKUN OLUWASEMILORE** in partial fulfillment of the requirements for the degree of **BACHELOR OF SCIENCE IN COMPUTER SCIENCE.** The original research work was carried out by him under by supervision and is hereby accepted.

_____ (Signature and Date)

Dr B. J ADEMOLA

Supervisor

_____ (Signature and Date)

Dr. M.O. Adewole

Coordinator, Department of Computer Science and Mathematics

DEDICATION

This project is dedicated to God almighty for a successful completion of my BSc. in Computer Science.

ACKNOWLEDGMENTS

I owe my profound gratitude to God Almighty who gave the strength, wisdom and courage, divine help and provision to me from the beginning to the completion of this work. I express my gratitude to my major supervisor, Prof. P.A. Idowu for his guidance and support in ensuring the successful completion of this research. God bless you richly sir. I sincerely appreciate the Dean, College of Basic and Applied science for his fatherly advice, guidance and teachings. My heart felt gratitude goes to the Coordinator of the Department of Computer Science and Mathematics – Dr. Adewole and all other members of staff of the department of Computer Science: Mr. J. Balogun, Dr. (Mrs.) F.A. Kasali, Mr. I.O. Ebo and every other staff I failed to mention.

I will forever be grateful to my parents Mr. and Mrs. Dada, who sacrificed wealth, time and other resources for the sake of my success; and my siblings for their consecutive support and prayers.

Furthermore, I must appreciate the University chaplain and the chaplaincy unit for their love, care and prayer for me, and all Mountain Top University colleagues and friends for their prayers and support in one way or the other. God bless you all.

ABSTRACT

Over the year, people walk to restaurant to get their food while some walk to market to get ingredient for food, but in this modern day's people can easily order for what they desire or want. Nowadays, people do not have time to wait at restaurant, join queue for food, so they prefer to order for their food online which solves many problems such as overcrowding, long queue, long wait time etc. at restaurant.

The aim and objective of the system is to develop an information system that can be used by members (staff) of the university community to order for their meals in a convenient manner, identify the requirement of meal ordering system, specify the design of the system, implement the system and test the implemented system. To achieve this, the system was developed using one of Agile methodology (Extreme programming) which helped us to ascertain the steps needed to build the system.

The system developed helps staffs to order for food and get delivery of food at a precise location stated by them. Also, the system allows cafeteria food vendors to manage (add or remove) food on display screen and respond to already ordered food so as to notify the staff of the delivery time of the booked meal.

The result after ordering the food will appear on the vendor screen, who will send a confirmation message and prepare the food ordered. The system is the combination of Android and iOS and is built using flutter framework and firebase as the database.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER ONE

INTRODUCTION

1.1 Background to the Study

Meal is the food served and eaten especially at one of the customary, regular occasions for taking food during the day, as breakfast, lunch, or supper. Meal ordering is the process of making request for a food either from a restaurant or through the use of an electronic device. Meal ordering system is a program or an application that helps restaurants improve and control their restaurants. An online meal ordering system provides the ability of taking orders for food to be consumed by an individual and making sure that the food is delivered to the individual once done.

One of the most recent service providers that most fast-food companies in the Western world have adopted is the online meal ordering system, food is ordered online and delivered to customers using this manner. The online meal ordering system is one of the latest service providers most fast-food restaurants in the western world are adopting. With this method, food is ordered online and delivered to the customers. A meal booking system is a system designed to collect the food ordered by people and give them specific time when the food ordered will be available for collection. The online meal ordering system is a piece of software that is used to make the user's life easier.

When you don't feel like cooking or going out to eat, the online meal ordering system makes it possible for customers or users to acquire food quickly. (Vidyagiri, 2020).

Today, most people go through the stress of going to restaurants and eateries to get their food and in some cases, go through the stress of preparing food.

1.2 Statement of Problem

The system to be designed addresses various issues such as over-crowding. When many people arrive at the same time to get their food, there is risk of over-crowding which also contributes to long wait time. The long wait time is the time taking for an individual to get attended to, that is the time taken before been attended to and the time taken when attended to. Another issue of concern is based on the long queue formed during the process of ordering food. As more and more people arrive, the queue becomes long, which further discourage people from ordering food. It also removes the traditional means of going to purchase food at the restaurant.

1.3 Aim and Objectives

The aim of this study is to develop an information system that can be used by members of the university community to order for their meals in a convenient manner. The specific objectives are to:

i. identify the requirement of the online meal ordering system
ii. specify the design of the online meal ordering system
iii. implement the online meal ordering system
iv. test the implemented system

1.4 Proposed Methodology

In order to meet up with the aforementioned objectives of the study, the following methods will be adopted:

a. A review of literature will be done in order to identify and understand the existing system of an online ordering application.

b. The user and system requirements of the proposed system will be identified from the system users via informal interview
c. The system design will be specified using the unified modelling diagram such as sequence diagram, use case diagram, entity relationship diagram, class diagram, the data flow diagram, the activity diagram etc.
d. The database will be implemented using a no SQL database called Firebase and the front end will be implemented using Flutter.
e. The system will be tested component by component and then finally tested as a whole.

1.5 Significance of Study

Given the rapid advancement of computer technology in practically all areas of activity and its application in information systems, it is vital that all restaurants and food organizations must investigate online ordering systems to suit client needs as the manual method of acquiring food declines. Also, due to the current period we found ourselves, this system is far efficient because it helps to protect people from the deadly virus which spread fast in crowded places. From research, it was discovered that overcrowding can also lead to the increase in the spread of coronavirus. This project is set to benefit college restaurant or college cafeterias by removing the effect of overcrowding, reducing long queue and also reducing the wait time for getting food. It further provides a feedback system, where customers can give their various ideas and comment about the food.

1.6 Scope and Limitation

In this project, a university cafeteria system was developed, which allows staffs of the university to order for meal and receive it, this will reduce the number of customers

on the queue at the counter. It will also help reduce the work load on the employee serving or dishing food. Due to the time constraint, we will be focusing on designing and implementing an online food ordering system for university staffs only and we will not be implementing an online payment gateway.

1.7 Definition of Terms

i. Academic Staff: This comprises of staff members who hold an academic titles such as professor, associate professor, assistant professor, instructor, lecturer, or the equivalent of any of these academic positions .

ii. Customer: customers are the individuals and businesses that purchase goods and services from another business

iii. Firebase: Google's Firebase is a mobile app development platform that allows you to create, improve, and expand your app. It is also known as Backend-as-a-Service (Baas). It offers a wide range of tools and services to developers to help them create high-quality applications, expand their user base, and make income. It is based on Google's infrastructure. Firebase is a NoSQL database software component that saves data in JSON-like documents.

iv. Flutter: Google announced the arrival of Flutter, a free and open-source mobile UI framework, in May 2017. In a nutshell, it enables the development of a native mobile application using only one codebase. This implies you can design two distinct apps using a single programming language and codebase (for iOS and Android).

v. Food Ordering: the act or process of requesting for a particular meal or food at a restaurant.

vi. Meal: Any nutritive ingredient that an animal can process to provide energy and develop tissue.

vii. Menu: A list of dishes available at a restaurant or the food available or to be served in a restaurant or at a meal.

viii. Mobile Application: Mobile applications (sometimes referred to as mobile apps) are applications that are designed to operate on mobile applications and tablets. In contrast to desktop programs, mobile apps are moving away from integrated software systems.

ix. MySQL: MySQL is a relational database management system (RDBMS) based on Structured Query Language that is open source and supported by Oracle (SQL).

x. Online Food Ordering: The meal ordering system is a simple and easy way for consumers to buy food online without needing to visit a restaurant.

xi. Ordering System: They are a necessary component of inventory management.

xii. They are the programs that convert our projections, actual orders, safety stock, and order amounts into buy or production orders.

xiii. Ordering: The process or act of making an order for something

xiv. Restaurant: A restaurant or an eatery, is a business that prepares and serves food and drinks to customers. Meals are generally served and eaten on the premises, but many restaurants also offer take-out and food delivery services.

xv. Technology: The application of the knowledge and usage of tools (such as machines or utensils) and techniques to control one's environment. It is also the discipline dealing with the art or science of applying scientific knowledge to practical problems

xvi. Web Application: In contrast to computer-based software that runs locally on the device's operating system, a web application (or web application) is application software that operates on a web server (OS). Users access web applications using a web browser with an active network connection.

CHAPTER TWO

LITERATURE REVIEW

2.1 Information System

An information system (IS) is a collection of numerous pieces of technology that is used for the gathering, processing, storage, and distribution of information. (Techopedia, 2020). An information system consists of a number of technological components that are used to acquire, process, store, and disseminate information. (Piccolo & Pigni, 2018). Data is utilized to offer information, add to knowledge and create digital products that assist decision-making via the usage of information systems, which are comprised of components for data collecting, storage and processing. (Zwass, 2011).

People and organizations utilize data to gather, filter, process, generate, as well as disseminate information systems, which is an academic study of systems with a particular reference to information. Systems of hardware, software, and telecommunications networks that people develop and use to gather, produce, and disseminate valuable data are known as information systems. These systems are usually found in corporate contexts. Systematic collection, processing and storage of information is supported by information systems in an organization by coordinating decision-making, controlling and analyzing data and visualizing it. (Bourgeois, 2014).

2.1.1 History of information system

Despite the fact that IS has only been for five decades, it has done more than any other convention in history to grow its business and industry on the worldwide market. Today, the backbone of IS known as the World Wide Web, the Internet, or as a local area network in business, along with a list of buzzwords. EDI, EIS, ERP, SCM, and other hosts describe new ways to grow your business with IS. Despite the speed of information today, just 40 years ago the American business environment was

experiencing unprecedented postwar growth. During World Military II, much of the expertise of developing an economy was gained by organizing the country's industries to create effective war equipment.

Operations research was one field that grew out of the need to win the war (OR). At the end of the war, operating room related people were released from government work, unleashing disciplines unmatched with experience and high qualifications in history in business and industry, leaving the United States for 20 years. We have entered an era of prosperity and growth that continues. Year. In World War II also the first practical computer or Turing machine was born that was in charge of deciphering the German code and proactively warning the Allies of enemy movements. By today's standards, these early hands-on computers weren't very practical, at half a million dollars far less powerful than the calculators I bought today for less than ten dollars. Yet these early computers gave operation researchers the power they would, they needed to start simulating larger and more complex systems. This background to the early days of simulation, the operating room, and new technologies has led to the investigation of what is known as information systems.

The use of computers in business and industry generally began in the accounting department. This area was supposed to know better the use of numerical machines and the lack of understanding of the potential of databases as other business areas. At this time, many business schools have started to develop Management Information Systems (MIS) programs to meet the growing needs of IS managers. In the 1970s, more senior executives recognized the importance of IS and the flexibility it brought to the business. Telex has become the standard for information transfer and central computers have become the standard for creating databases. As it became clear that

data needed to be categorized and easily accessible, information-driven companies began to move mainframes from accounting control to their own departments.

Some of the technical adept managers of these new departments begin spending enormous amounts of money on systems and software at their own discretion, outspending all other departments without generating any returns for the company as IS gains autonomy and big budgets in companies. It was a difficult and dangerous decision for CEOs to move their companies to IS-based systems. Those who understood the systems tended to have their own objectives. Businesses' IS development was shaped by this turbulence. (University of Houston, 2000).

2.1.2 Types of information system

Information system can majorly be classified into three, they are:

i. Operational management: Workforces and employees use the Transaction Processing System (TPS), Office Automation System (OAS), and Knowledge Management System to streamline daily operations.
ii. Tactical management: Used by business unit managers, handles semi-structured data, and includes management information systems
iii. Strategic management: Unstructured data is handled via the decision support system and executive support system, which are used by executives.

The six major types of information systems corresponding to each organizational level (the four levels shown in above) are:

i. Transaction processing systems (TPS): ensures that all of the contractual, transactional, and customer relationship data is stored in a safe location and accessible to everyone who needs it. It serves the operational level of an organization.

ii. Knowledge work systems (KWS): A knowledge management system stores and extracts information to help users enhance their knowledge and optimize collaboration efforts to complete tasks.

iii. Office automation systems (OAS): An office automation system is a network of various tools, technologies, and people required to conduct clerical and managerial tasks. It used to serve the knowledge level of an organization.

iv. Decision-support systems (DSS): It processes data to assist in management decision-making. It stores and gathers the information required for management to take the proper actions at the correct time.

v. Management information systems (MIS): A management information system uses various transaction data from a TPS to help middle management optimize planning and decision-making. It serves the management level of the organization.

vi. Executive support systems (ESS): Executive support systems are similar to a DSS but are primarily used by executive leaders and owners to optimize decision-making. It serves the strategic level of an organization.

2.1.3 Components of information system

Every information system includes several key components: hardware, software, telecommunications, people and data. Hardware refers to the physical pieces of the information system; software is the programming that controls the information system; telecommunication transmits information through the system; humans manage and interact with the information system; and data is information stored within and processed by the system.

i. Hardware: The hardware component of an information system comprises the physical elements of the system. People can touch and feel pieces of hardware. These mechanisms, equipment and wiring allow systems like computers,

smartphones and tablets to function. Input and output devices are essential pieces of technology that allow humans to interact with computers and other information systems. Keyboards, mice, microphones and scanners are all examples of input devices. And output devices might include printers, monitors, speakers and sound and video cards. Pieces of hardware including microprocessors, hard drives, electric power supply units, and removable storage also allow computers to store and process data.

ii. Software: This is an intangible program that manages information system functions, such as input, output, processing, and storage. System software, such as MacOS and the Microsoft Windows operating system, provide the basis for running application software. Application software operates programs intended for specific purposes in information systems. For example, word processing applications are used to create and edit text documents. Graphical user interface (GUI) software is one of the most popular application software. It displays information stored on your computer and allows users to interact with your computer through digital graphics such as icons, buttons, and scroll bars instead of text-based commands. The software is either open source or closed source. The coding of open-source software is open to users and programmers, while closed source software is proprietary.

iii. Telecommunication: Communication systems connect computer networks and allow information to be transmitted over them. Telecommunications networks also allow computers and storage services to access information from the cloud. There are several ways that telecommunications networks can use to transmit information. Telephony, Internet and cable providers use coaxial and fiber optic cables to send data, video and audio messages.

A local area network (LAN) connects computers to create a computer network in a designated space, such as a school or home. A wide area network (WAN) is a collection of LANs that facilitates the exchange of data between large areas. Virtual private networks (VPNs) allow users to protect their privacy online by encrypting data on public networks. Microwaves and radio waves can also be used to transmit information over telecommunications networks.

iv. Data: Data is a raw intangible fact that other components of the information system store, transmit, analyze and process. Data is often stored as numerical facts and represents quantitative or qualitative information. Data can be stored in a database or data warehouse in the format that best suits the organization that uses it. The database contains a collection of data that can be queried or retrieved for a specific purpose. Databases allow users to perform basic operations like save and retrieve. Data warehouses, on the other hand, store data from multiple sources for analytical purposes. This allows users to evaluate the organization or its operations.

v. Human resources: Human resources are an important part of information systems. The human components of an information system include those who are qualified to influence and manipulate information system data, software, and processes. People involved in information systems may include business analysts, information security analysts, or systems analysts. Business analysts are working to improve the organization's operations and processes. The focus is often on improving efficiency and productivity or optimizing distribution. Information security analysts are committed to preventing data breaches and cybersecurity attacks. Systems analysts also use information technology to help organizations optimize the user experience of their programs.

2.2 Ordering information system

The OIS is a tool for reporting production orders, scheduled orders, and process orders used in shop floor control and process industries. As with the IOC logical database, evaluations take place via the initial order tables.

2.2.1 Online Ordering System

An ordering system can be defined as a simple and convenient way for people to purchase products online, without having to go to the organization. Internet access is required for this system to function. As a result of the internet, organizations and companies are able to communicate with their customers. Basically, Online Ordering Custom Made for Organization, Fast and Reliable Platform that does it all from taking order to providing delivery updates. This system is very profitable for organization who are into sales of different product. There are many software companies that can help people to build best solutions for their business (Haridwaj, 2020).

User's may place orders via a website or mobile application called Ordering System. Ordering product online is similar to online shopping – buying goods online. So, once user places an online order, organizations receive the order, starts processing it and delivers product to the user. An order is received/executed by a food cooperative or a meal delivery service, which then gathers and delivers food from a restaurant that the user specifies when making the order. Here in this case, restaurant provides a commission to the food cooperative on each order they provide (Hyderabad, 2019).

An online meal ordering system is software that enables restaurants to accept and handle online orders. An Online Food Ordering System's primary goal is to handle the information of an item's category, food, delivery address, order, and shopping cart. It keeps track of all the data pertaining to the Item Category, the Customer, the Shopping Cart, and the Item Category. A ready-to-eat product, such as a meal

prepared in a home or restaurant, or meal that has not been prepared specifically for immediate consumption, such as fresh vegetables from a farm or garden, or frozen meats, etc., can be included in this category.

Pizza Hut's pizza was the first internet food order placed in 1994. In the United States, the online meal ordering industry has grown, with 40 percent of U.S. people having ordered food online at least once. You may buy groceries online and have them delivered or picked up from a restaurant. The first online meal ordering business, World Wide Waiter (now known as waiter.com), was established in 1995. The site initially served just northern California, eventually extending to many more locations in the United States.

2.3 System Development Life Cycle (SDLC)

According to the System Development Life Cycle (SDLC), a project's phases are defined from the beginning to the end of the project. In the SDLC, conventional business processes are used while creating software applications. Planning, Requirements, Design, Build, Documentation, Testing, Deployment, and Maintenance are the usual six to eight stages. Depending on the project's scope, some project managers may combine, divide, or eliminate stages.

All software development initiatives should have these components as a foundation. In addition, the SDLC provides a method to assess and enhance the process of development. Using this method, each stage of the process may be analyzed in great detail. Companies are able to optimize efficiency at each step as a result. As computer power grows, software and developers are placed under more pressure. In order to remain competitive, companies must cut costs, provide software more quickly, and meet or exceed their customers' requirements. SDLC assists in achieving these goals

by finding inefficiencies and increased costs and correcting them so that operations function smoothly. (Jevtic, 2019).

A successful System Development Life Cycle (SDLC) should result in a high-quality system that satisfies customer expectations, is completed within the time and cost estimates, and operates effectively and efficiently in the existing and future Information Technology infrastructure. For the development of an information system, analysts utilize the SDLC. SDLC includes the following activities: Requirements, Design, Implementation, Testing, Deployment, Operations, and Maintenance

2.3.1 Phases of SDLC

The Systems Development Life Cycle is a systematic technique that explicitly divides the labor necessary to build a new or changed information system into phases, this phases are: Requirement analysis and planning, Feasibility study, Design, Coding, Testing, Deployment or installation, and Maintenance

i. Planning and requirement analysis

The most essential and fundamental step of the SDLC is the requirements analysis. Senior members of the team are responsible for this, and they include feedback from customers, sales divisions, market research, and specialists in the field. Use this information to design a fundamental project strategy and to perform economic, operational, and technological feasibility assessments. During the planning phase, quality assurance needs are determined and project hazards are identified. At this point, the breadth of the project and the problems, possibilities, and directions that prompted it become more apparent. Requirements teams must gather comprehensive and accurate criteria throughout the gathering step. Using this information, businesses may finalize the timetable for finishing the work of a system or program. This

research defines various technological methods that may be used in order to effectively complete a project with the least amount of risk possible.

ii. Feasibility study

Defining and documenting your software requirements is the next stage in the SDLC process after finishing the requirements analysis phase. A document called the "Software Requirements Specifications" (SRS) was used to guide this procedure. It encompasses everything that has to be planned and developed over the course of a project. There are five types of feasibility check, they are:

i. Economic: Can we finish the project on time and on budget?
ii. Legal: Is it possible to conduct this project in accordance with cyber legislation and other regulatory framework/compliant frameworks?
iii. Operation feasibility: Can we provide the operations that is expected of us by the client?
iv. Technical: It is necessary to determine whether the present computer system is capable of supporting the program.
v. Schedule: Determine whether the project can be completed within the given schedule or not time frame specified or not

iii. Design

A software and system design document is produced in this third phase based on the requirements specification document in the previous phase. System architecture may be defined using this information. This step of the model's creation serves as an input to the following phase. There are two types of design documents created during this phase: High-Level Design (HLD) and Low-Level Design (LLD).

iv. Coding

Code is the following step once the system design phase is complete. Here, the developer puts code in his preferred programming language and starts to construct the

whole system. As part of the coding process, various developers are given distinct responsibilities. In the software development life cycle process, this is the most time-consuming phase.

v. Testing

Once the software is complete and deployed in a test environment. Testing the functionality of the entire system is then carried out by the test team, so as to be certain that the entire application functions and perform according to the customer's requirements.

vi. Deployment

As soon as the software testing phase is complete and there are no remaining bugs or errors in the system, the final deployment process begins, and the product is formally released on the market. As part of an organization's business strategy, product deployment may occur in phases. An initial release in a restricted market sector may be followed by a real-world test (UAT- User acceptance testing). If there are any deployment issues, they will be addressed in accordance with the feedback given by the project manager.

vii. Maintenance

Customers begin utilizing the created system after it has been deployed, the following 3 activities occur:

i. Bug fixing - Some situations are not tested, and as a result, problems are found.
ii. Upgrade - Newer versions of the Software must be applied to the program in order to maintain maximum performance.
iii. Enhancement – New features are added into the system

The phase of SDLC is primarily concerned with ensuring that requirements are fulfilled and that the system continues to operate as specified in the first phase.

2.3.2 SDLC models

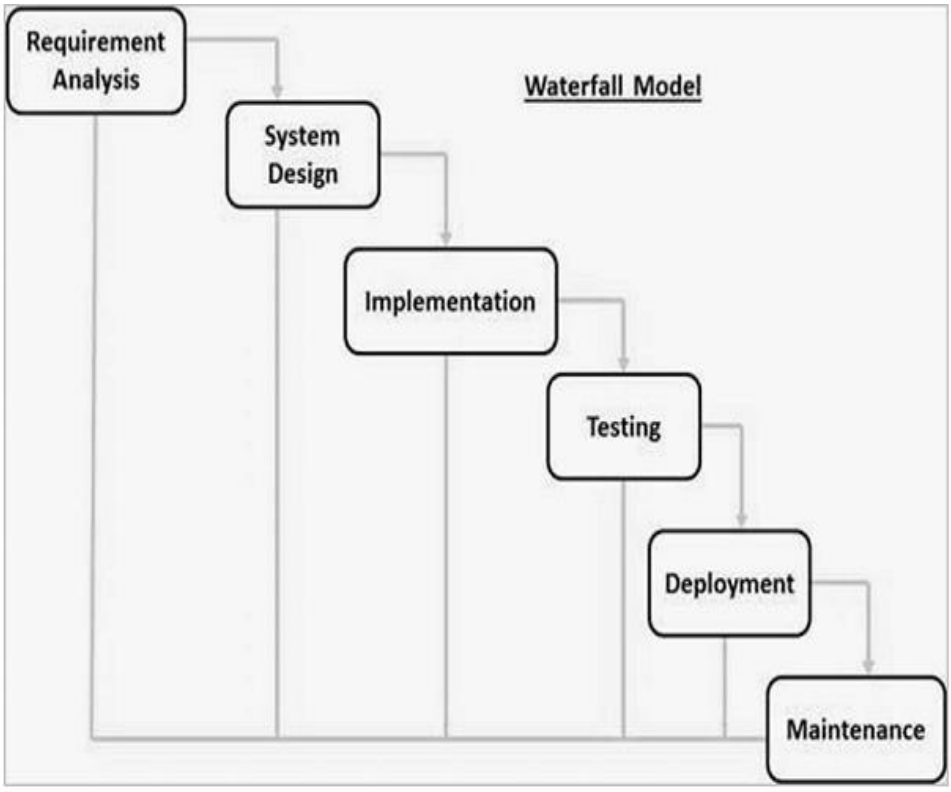
During the software development process, a variety of software development life cycle models are developed and designed. "Software Development Process Models" is another name for these models. Each process model has a unique set of procedures that must be followed to guarantee software development success.

a. Waterfall Model

The first process model presented was the waterfall model. Additionally, it is referred to as a sequential life cycle model. Use and understanding are extremely simple. There is no overlap between stages in the waterfall model. It depicts software development as a linear, sequential process. Each step of the development process, therefore, starts only after the preceding phase has been finished. As a result of this, the stages of the waterfall model do not overlap one another. The continuous phases of the waterfall model are:

- Requirements Gathering and Analysis: During this phase, all potential requirements for the system under development are gathered and recorded in the requirements specification document.
- System Design: This phase examines the first phase's requirements and prepares the system design for implementation. Hardware and software requirements are specified as well as a system's architectural design.
- Implementation: The system is initially created in a tiny program called a unit, and then incorporated into the next step once the system design is entered. Each item is designed and tested to ensure its functioning before it is released. In software development, this is known as a unit test.
- Integration and testing: Once all units created during the implementation phase have been tested, they are incorporated into the system as a whole. Failures and defects are tested on the whole system after it has been fully integrated.

- System Deployment - It's time to deploy or release the product to the market after functional and non-functional testing is complete.



2.2: Waterfall Model (Tutorialspoint, 2005)

Table 2.1: Advantages and Disadvantages of the Waterfall Model (Melsatar, 2019)

S/N	Advantages	Disadvantages
1	Simple and straightforward to comprehend and apply	Until late in the life cycle, no functioning software is developed.
2	Because of the model's rigidity, it is simple to manage. Each phase includes its own set of deliverables and a review mechanism.	This is not an appropriate approach for complex, object-oriented programs.
3	One phase at a time is processed and completed.	Poor model for long-term projects.
4	Works effectively for smaller projects with well-defined criteria.	Not appropriate for projects with a moderate to high risk of change in requirements. As a result, the risk and uncertainty associated with this process model are significant.
5	Stages are well defined.	Measuring development within phases is tough.
6	Tasks are simple to organize.	Changes in requirements cannot be accommodated.
7	Both the process and the results are meticulously documented.	Changing the scope of a project during its life cycle might lead to its demise.

- Maintenance: It is not uncommon for customers to have difficulties. As a result of these problems, patches have been issued. Also, improved versions of the product have been produced.
- To bring about these modifications in your surroundings, maintenance is performed.

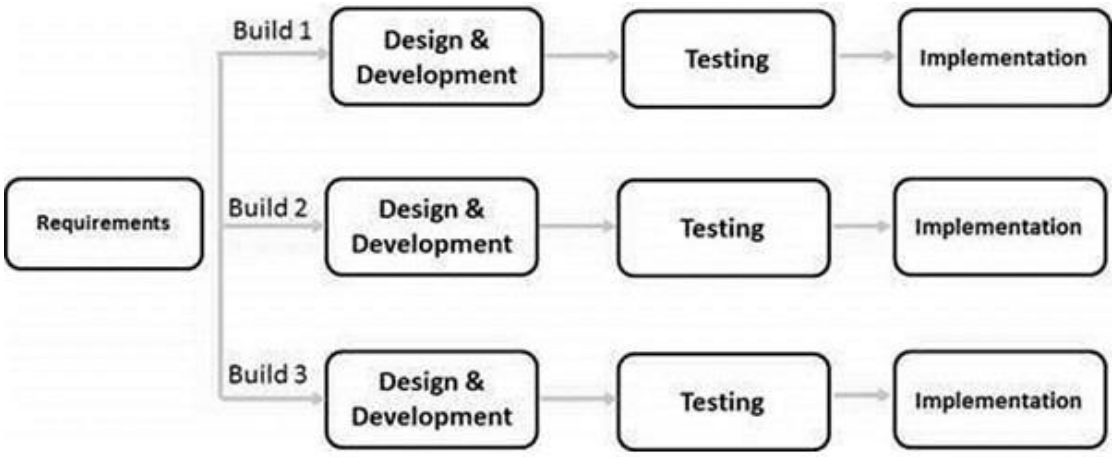
b. Iterative Model

In an iterative model, the iterative process starts with a basic implementation of a limited set of software requirements and then repeatedly expands the developing version until the whole system is in place and ready for deployment. The iterative life cycle model does not start with a full definition of the requirements. Instead, development starts by defining and developing just a part of the program, which is then evaluated to discover further needs.

This procedure is then repeated to produce a new version of the program at the conclusion of each iteration of the model. Each iteration makes design modifications and adds new functionality. The fundamental concept behind this approach is to build a system with a cycle of iterations (iterations) and tiny portions at a time (increments) (increments). The key to effectively utilizing the iterative lifecycle of software development is the thorough validation of requirements and the validation and testing of each software version against those criteria in each model cycle. As the program develops via a continuous cycle, it must be tested and expanded frequently to verify each version of the product.

c. Spiral Model

The spiral model has four phases. A software project iterates through these phases in an iteration called a spiral.



. 2.3: SDLC Iterative Model (Tutorialspoint, 2009)

Table 2.2: Advantages and Disadvantages of Iterative Models

S/n	Advantages	Disadvantages
1	Some features may be added fast and early in the life cycle.	More resources and attention from management are necessary.
2	The results are obtained early and on a regular basis.	Although the cost of modification is lower, it is not appropriate for changing requirements.
3	Parallel development is possible, and progress may be tracked.	Because not all requirements are acquired at the start of the complete life cycle, system architecture or design challenges may occur.
4	It is less expensive to adjust the scope/requirements.	Defining increments may need the specification of the whole system.
5	Testing and debugging are simple with smaller iterations.	Smaller tasks are not suited.
6	During iteration, risks are recognized and handled, and each iteration is a manageable milestone.	Management complexity is increasing.
7	Every increase results in the delivery of operational product and a reduction in the initial operating time.	The project's end date may not be known, which is a risk.
8	Each increment's issues, problems, and dangers can be used/applied to the following	The risk analysis phase is critical to the project's development.

	iteration.	

i. **Identification**: This phase begins with gathering business requirements in a basic spiral. In later spirals, as the product matures, system requirements, subsystem requirements, and unit requirements are identified during this phase. This phase also includes understanding system requirements through ongoing communication between customers and systems analysts. At the end of the spiral, the product is deployed to a specific market.

ii. **Design**: Starting with a simple spiral, collect business needs. System requirements, subsystem requirements, and unit requirements are defined during this part of the spiral, which occurs as the product develops in subsequent stages.

    During this phase, clients and systems analysts communicate often to understand system needs. An end-of-the-spiral deployment of the product occurs on a particular market

iii. **Build**: In each spiral, a genuine software product is created during the build phase. A first POC (proof-of-concept) is created at this phase of the baseline spiral in order to get input from consumers while evaluating and designing a product.

    Another spiral with better design specifications and requirements creates a functioning software model called a build, which is identified by version number. For the customer's input, these builds will be given to him.

iv. **Risk assessment and analysis**: at this stage, technical feasibility and management risks, such as cost overruns and schedule delays, are identified,

estimated, and monitored as part of the risk analysis. As part of the first iteration, customers assess the program once it has been built and give comments.

Iterations continue to be performed based on the customer's input as the software development process moves on to the next iteration. The spiral continues to iterate throughout the life of the program. Spiral models are extensively employed in the

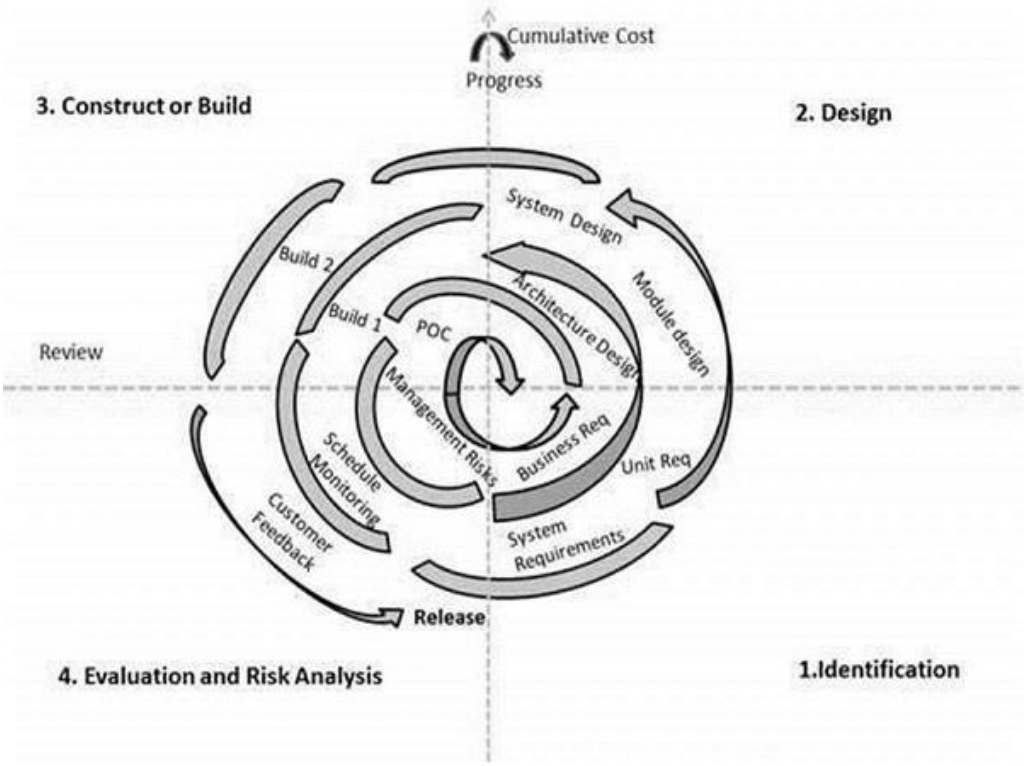


Figure 2.4: Spiral Model (Tutorialspoiint,2009)

Table 2.3: Advantages and Disadvantages of Spiral Model

S/n	**Advantages**	**Disadvantages**
1	It is possible to adapt changing needs.	Management is becoming increasingly difficult.
2	Prototypes can be used extensively.	The project's conclusion may not be known for some time.
3	Requirements may now be captured more precisely.	Not appropriate for low-risk or small-scale initiatives, and may be costly for such enterprises.
4	Users get a first look at the system.	The procedure is complicated.
5	Development may be broken into smaller portions, and dangerous aspects can be created early, allowing for better risk management.	The spiral might carry on indefinitely.

software business, since they are in sync with the naturally occurring development process of any product - i.e., learning as a product matures, with little risk to both the client and the development companies.

d. V Model

According to SDLC, processes operate in a V-shape. The verification and validation paradigm is also known as the verification model. Testing stages are associated with each development step in Paradigm V, which is an extension to the waterfall model. In other words, each step of the development cycle has a test phase that is closely linked to it. In this approach, the following step begins only when the preceding phase is completed. In model V, the development phase's test phases are scheduled in tandem with the development phase. Hence, there are two validation phases on each side of the "V". During the coding step, the two elements of the V are combined. There are three phases in the V model, they are: Verification Phases, Coding Phase and Validation Phase.

i. **Verification Phase:** There are several verification phases in the V Model, they include:

- Business requirements analysis: Customers' perspectives are taken into account at this initial step of product design. Here, you'll interact with your consumers in depth to determine their precise expectations and needs.
- System design: Designing a comprehensive system follows once you've established clear and precise product requirements.

  As part of the system design, the hardware and communication infrastructure of the device under development is outlined in depth.
- Architectural design: The architectural requirements are analyzed and developed during this phase of the project. Technical methods are usually offered in a variety of forms with the ultimate choice being determined by the combination of technical and financial feasibility
- Module layout: Each module is designed in depth at this phase. Low-level design is the term used for this (LLD). Compatibility with other modules of the system architecture and other external systems is critical to the design. An essential component of the design process is 'unit testing,' which identifies potential problems before they become major problems.

ii. **Coding Phase**: Code is written for a system module created during the design process. On the basis of architectural and system requirements, the optimal programming language is selected. Guidelines and standards govern the encoding process. After being subject to a number of code reviews, the final build is optimized for the optimum performance.

iii. **Validation Phase**: The various validation phases of the V model are described in detail below.

- Unit test: During the validation step, unit tests that were created during the module design phase are executed in code. Unit tests are code-level tests that help to eliminate errors early, but unit tests cannot find all the flaws.

- Integration test: Integration testing is a part of the architectural design step of the development process. A system's internal components are tested for cohabitation and communication via integration testing.
- System test: This is directly related to the design phase of the system. System tests verify the overall system's functioning and the communication between the system under development and external systems. Most hardware and software compatibility issues can be apparent while running this system test.

Table 2.4: Advantages and Disadvantages of V-Model

S/N	ADVANTAGES	DISADVANTAGES
1	This is a very disciplined strategy in which each Phase is performed one at a time.	Uncertainty and high risk.
2	Works effectively for smaller projects with well-defined criteria.	This is not an appropriate approach for complex, object-oriented programs.
3	Simple, straightforward, and simple to grasp and use.	Poor model for long-term projects.
4	The model's stiffness makes it simple to use. Each level includes a review method and stated deliverables.	Not appropriate for projects with a moderate to high risk of change in requirements.
5	This is a very disciplined strategy in which each Phase is performed one at a time.	It is tough to update a functionality once an application has entered the testing stage.
6	Works effectively for smaller projects with well-defined criteria.	Until late in the life cycle, no functioning software is developed.

- Entrance examination: Acceptance testing is linked to the business requirements analysis phase and consists of testing the product in the user environment. Acceptance testing identifies concerns with compatibility with other systems in the user's surroundings. It also detects non-functional faults in the real user environment, such as load and performance difficulties.

The application of the V-model is quite similar to that of the waterfall model, since both models are sequential. Before the project begins, the requirements must be crystal clear, since it is generally costly to make modifications afterwards. Medical development uses this approach since it is a highly regimented profession.

e. Big Bang Model

The Big Bang model is an SDLC model that does not follow a specific process. Development only starts with the money and effort required as input, and the output is developed software that may or may not meet customer requirements (Javatpoint, 2007). This Big Bang model does not follow processes or procedures and requires few plans. Even customers don't know exactly what they want and requirements are

implemented on the fly without much analysis. This model is generally used in small projects where the development team is very small. (Tutorialspoint, 2021)

The Big Bang model is about concentrating all possible resources on software development and coding, with little or no planning. Understanding and implementing requirements as needed. A software update may or may not be required to make the necessary modifications. Ideal for small projects where one or two developers collaborate, as well as academic and practical initiatives, this model is a good choice. Ideal for goods whose needs are unclear and whose release date is uncertain.

Table 2.5: Advantages and Disadvantages of Big Bang Model

S/N	ADVANTAGES	DISADVANTAGES
1	This is a pretty basic model.	There is a high level of risk and unpredictability.
2	There is little to no preparation necessary.	This is not an appropriate approach for complex, object-oriented programs.
3	Simple to handle	Poor model for long-term projects.
4	There are very little resources required.	If needs are misinterpreted, it may be quite costly.

f. Agile Model

The Agile SDLC model is a hybrid of an iterative and incremental process model that emphasizes process flexibility and customer satisfaction through the quick delivery of usable software solutions. Agile methodologies divide the product into smaller incremental builds. These builds are supplied in an incremental fashion. Each iteration typically lasts between one and three weeks. Each iteration requires a cross-functional team to engage in many areas at the same time, such as planning, requirements analysis, design, coding, unit testing, and acceptance testing. The work product is visible to the client and key stakeholders at the conclusion of the iteration. (Tutorialspoint, 2021)

Each project, according to the agile model, must be handled differently, and current techniques must be modified to better meet project objectives. In Agile, jobs are split into time periods (short spans of time) that offer particular functionality for the launch of a product or service. Each iteration is based on an iterative process that results in a

fully functional software development. In terms of functionality, each build is incremental. It has all the features you need. Agile software development began in the early stages of software development, and its flexibility and adaptability made it popular over time.

g. Rapid Application Development

The Rapid Application Development (RAD) model is based on prototyping and iterative development and does not include any specific plans (The Economic Times, 2021). In addition to the actual software production process, the product's development process requires extensive planning. Customer needs are collected via workshops or focus groups, prototypes are tested early utilizing iterative ideas, existing prototypes (components) are reused, and the process is continued.

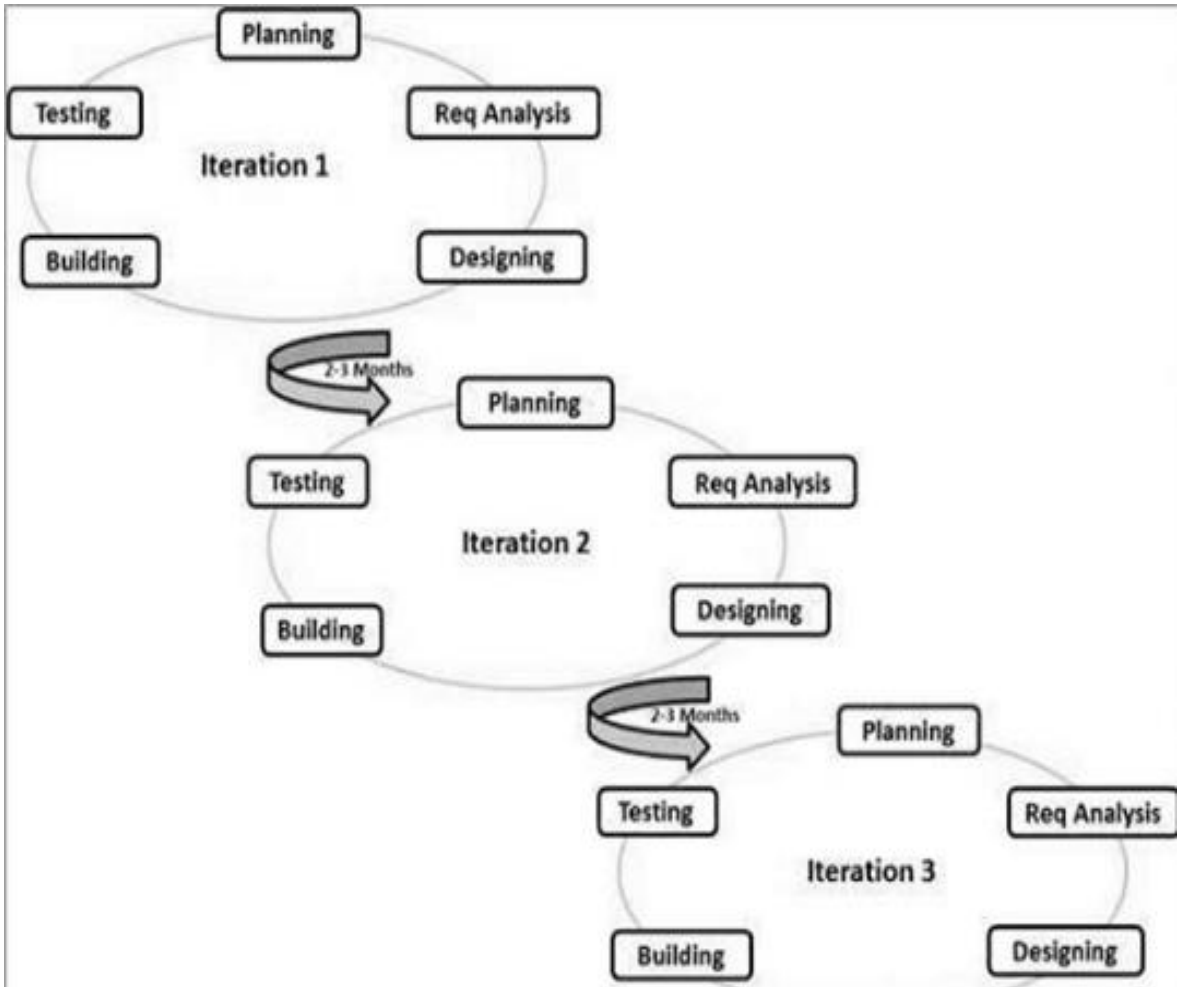


Table 2.6: Advantages and Disadvantages of Agile Model

S/N	ADVANTAGES	DISADVANTAGES
1	Is a very realistic approach to software development and easy to manage.	Not suitable for handling complex dependencies.
2	Promotes teamwork and cross training.	Greater risk of sustainability, maintainability and scalability.
3	Functionality can be developed rapidly and demonstrated.	Overall planning, agile leaders, and agile project management practices are essential and won't work without them.
4	Resource requirements are minimum.	Tight delivery controls determine scope, capabilities, and delivery times.
5	Suitable for fixed or changing	If the customer is unclear, the team

	requirements and flexible for developer use	may go in the wrong direction because it relies heavily on customer interaction.

Targeted integration and rapid delivery are the primary goals. As a software development approach, fast application development puts a high priority on quick prototyping and utilizes little planning. It is functionally identical to a product component. Rapid Application Development (RAD) involves the simultaneous development of prototypes of several functional modules that are then combined to produce a full product. In the absence of comprehensive planning, it is simple to integrate modifications in the development process. As part of the RAD project, a small team of developers, domain experts, client representatives and other IT resources work on components or prototypes in phases. This approach is iterative and incremental.

In order for this approach to be successful, it must be possible to reuse the prototype. The RAD approach divides the steps of analysis, design, build, and testing into short, iterative development cycles. The following are the various phases of the RAD Model:

i. Business Modelling: In terms of information flow and dissemination, the product's business strategy has been developed to take advantage of various business channels. An in-depth examination of your business's information needs will reveal how it is obtained, how and when it is processed, and what variables contribute to a smooth flow of information.

ii. Data Modelling: During the business modeling phase, the information collected is evaluated and processed in order to create a collection of business essential data items. All data sets have characteristics that have been discovered and described. The business model establishes and defines the connections between these data items in great detail.

iii. Process Modelling: To accomplish a specific business goal, data object sets established during the data modeling phase are converted into business information flows specified by the business model. It is at this phase that a

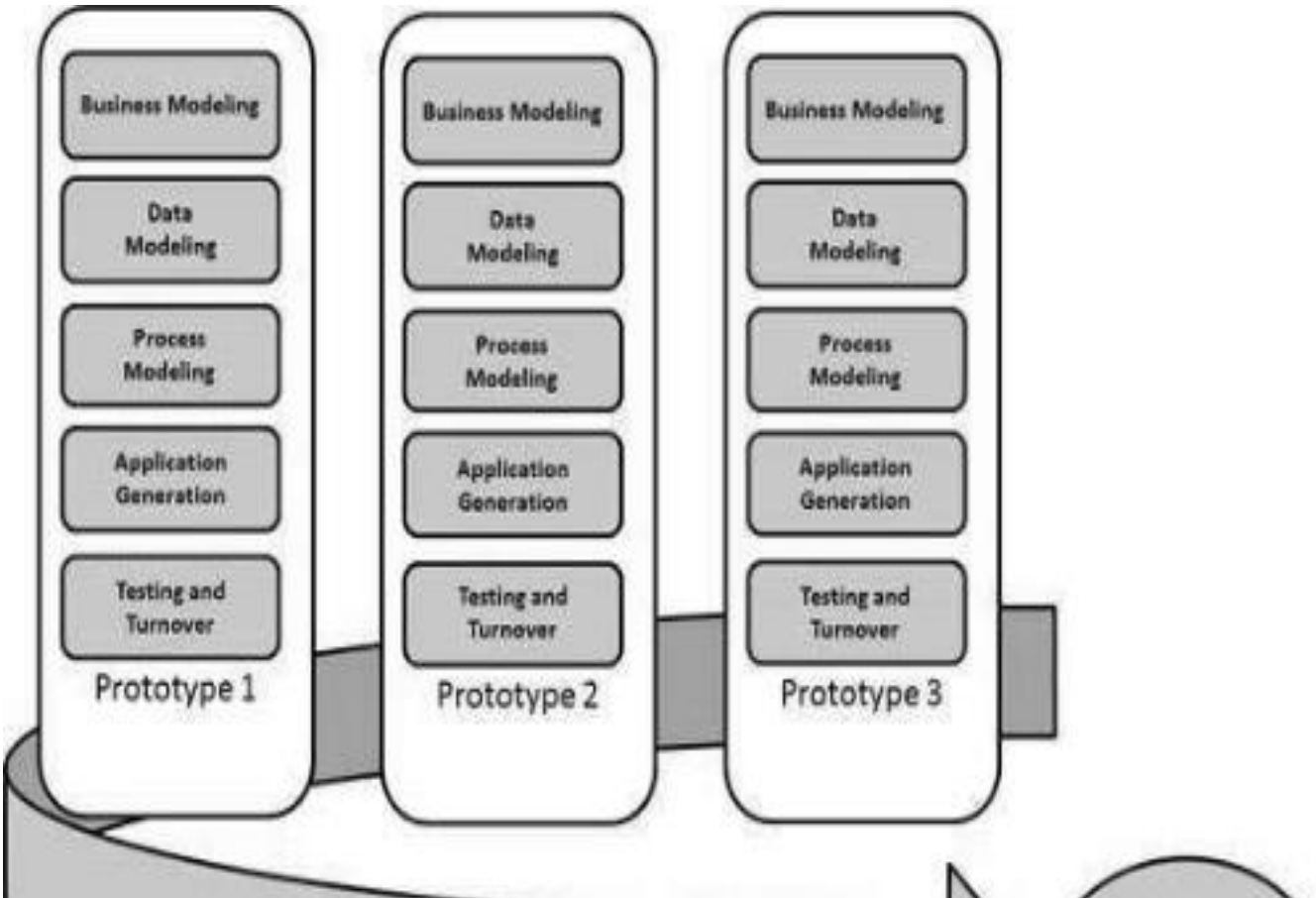


Table 2.7: Advantages and Disadvantages of Rapid Application Development

S/N	ADVANTAGES	DISADVANTAGES
1	Progress can be measured.	Requires highly skilled developers/designers.
2	Reduced development time.	High dependency on Modelling skills.
3	Increases reusability of components.	Management complexity is more.
4	Quick initial reviews occur.	Suitable for systems that are component based and scalable.
5	Encourages customer feedback.	Requires user involvement throughout the life cycle.
6	Productivity with fewer people in a short time.	Suitable for project requiring shorter development times.

process model for altering or adding to a collection of data items is developed and specified.

The procedure for adding, removing, retrieving, or changing data items is described in detail.

iv. Application Generation: The actual system is built and coding is done by transforming the process and data model into actual prototypes using automated tools.

v. Testing and Turnover: When testing the RAD model, each iteration of the prototype is carried out separately. A complete test coverage is required for the data flow and interaction between all components.

Most of the programming components have previously been tested, which reduces the chance of significant issues occurring during the project's

implementation. The RAD model can be successfully used to projects that have obvious modularization. RAD may fail if the project cannot be split down into modules.

h. Software prototype

Building software application prototypes that demonstrate the capabilities of the product under development but may not have the exact logic of the original software is referred to as software prototyping. Because it allows you to grasp client needs in the early phases of development, software prototyping is becoming increasingly popular as a software development approach. This allows software designers and developers to obtain useful input from consumers while also understanding exactly what is expected of the product under development. A prototype is a working model of software with limited functionality. Prototypes don't always have the exact logic used in real-world software applications, but they are extra effort that must be accounted for in effort estimates. Prototyping is used to allow users to evaluate developer suggestions and test them before deployment. It also helps you understand requirements that are user-specific and may not have been taken into account by the developer when designing the product. (Girish, 2021). The following are approaches to design software prototype: Basic Requirement Identification, Initial prototype development, Review of the Prototype, and Revise and Enhance the Prototype.

There are different types of software prototypes used in the industry, the following are the major software prototyping types used widely

i. Throwaway/Rapid Prototyping: Disposable prototyping is known as fast-end or closed-end prototyping. This type of prototyping requires little effort with a minimum requirements analysis for prototyping. This is followed by a complete rewrite of the real system, based on the user's actual needs.

ii. Evolutionary prototypes: It is based on the construction of a functioning prototype that has limited functionality, which is known as "evolutionary prototype". It will be used as the foundation for all future prototypes and as the basis for the whole system. Through the use of evolutionary prototypes, well-understood needs are included into the prototype and added as they become clear.

iii. Incremental prototypes: Incremental prototyping is the creation of multiple functional prototypes of different subsystems and the integration of all available prototypes into a complete system.

iv. Extreme prototypes: Extreme prototyping is used in the domain of web development. It consists of three consecutive phases. First, the basic prototype, including all existing pages, is displayed in HTML format. Then use the prototype service layer to simulate data processing. Finally, the service is developed and incorporated into the final prototype. This is known as extreme prototyping, and it is designed to bring attention to the second step of the process. In this phase, a fully

Table 2.8: Advantages and Disadvantages of Software Prototype

S/N	ADVANTAGES	DISADVATAGES
1	User participation in the product has increased even before its adoption.	Because of the prototype's reliance, there is a risk of poor requirement analysis.
2	Users have a better grasp of the system being created since a functioning model of the system is shown.	Users may become perplexed when navigating prototypes and live systems.

3	Reduces time and expense by detecting faults much earlier.	In practice, this process may enhance the system's complexity when the scope of the system expands beyond the original goals.
4	Users may provide input more quickly, which leads to better solutions.	Even if it is not technically viable, developers may attempt to utilize previous prototypes to construct the actual system.
5	Missing functionality is quickly spotted.	If prototypes are not carefully managed, the effort expended may be excessive.

functional user interface is developed with little consideration of the actual service.

Systems having a high degree of user involvement, such as online systems, are best developed using software prototypes. Prototyping may be particularly successful in systems where users must fill out forms and travel through various screens before the data is processed. Many types of software that handle a large amount of information and have few user interfaces do not benefit from prototyping. For such projects, prototyping may be an added cost and time commitment.

i. Extreme programming

This model will be used in building the food order system. Kent Beck, Ward Cunningham, and Longer Freeze came up with Extreme Programming in 1999. Other contributors include Robert Martin and Martin Fowler. In the mid-1980s, Kent Beck and Ward Cunningham began partnering at Tektronix. Extreme Programming (XP) is an agile software development framework aimed at creating higher quality software and higher quality lives for development teams. XP is the most specific agile framework for proper engineering techniques in software development.

XP was designed and developed to meet the specific needs of small software development teams working with vague and changing requirements in a fast-paced environment. Extreme programming includes:

i. Create unit tests before scheduling and keep all tests running at all times. Unit testing is automated, eliminating defects early and reducing costs.
ii. Start with a design that is simple enough to code the functionality in question, and redesign as necessary.
iii. Pair programming, involves two programmers using a screen and a keyboard in turn. One confirms the output and provides the input while the other uses the keyboard.

Table 2.9: Advantages and Disadvantages of Extreme Programming

S/N	ADVANTAGES	DISADVANTAGES
1	Slipped schedules: Achievable development cycles and scheduling ensure timely delivery.	Another drawback of XP is that this method does not measure the quality assurance of your code. The initial code may be buggy.

2	Cancelled Projects: Ensure client transparency and resolve issues immediately by focusing on ongoing client engagement.	XP is not the best option if the programmers are geographically separated

iv. Integrate and test the entire system several times a day.

v. Quickly put a minimal work system into production and upgrade as needed.

vi. Keep your customers involved and get constant feedback.

2.4 Unified Modelling Language (UML)

The Unified Modeling Language (UML) is a standardized modeling language comprised of an integrated collection of diagrams that not only assists system and software engineers in specifying, visualizing, building, and documenting software system artifacts. UML is a set of engineering best practices that have shown to be effective in modeling big and complex systems. UML is a critical component of object-oriented software development as well as the software development process. To depict the design of software projects, UML typically use graphical notation (Visual Paradigm, 2002). Using UML as a basic modeling method, any real systems in today's complicated world may be represented. System architectural blueprints may be shown in a diagram using UML.

2.4.1 Conceptual model

Conceptual model are models made out of concept and their relationship. The first step taking before constructing. The conceptual model helps to understand entities in a system and how they interact with each other. When looking at a system, one can view a system from many viewpoints, for example a software development team will have an analyst, designer, coder, tester, quality assurance, customers and technical authors playing different roles in the development of a system. All this people are interested in different aspect of the system and require different level of details such as an analyst needs to understand the function and operation of the system before sending to the designer, a coder requires to understand the design of the system so as to convert to a low-level language, a technical writer needs to know the behaviour of the system and how the product function. To understand the conceptual model of UML, three major elements have to be mastered: the building blocks of UML, rules guiding connecting building blocks and common mechanism of UML

2.4.2 UML diagrams

In UML there are fourteen types of diagrams which are divided into 2 categories, they are: structure diagram and behavioral diagram.

a. Structure Diagram

Structure diagram represent the static aspect of the system and it part on different abstraction, implementation level and how related they are to each other. It lay emphasis on things that must be present in the system being modelled. Since this diagram represent the structure of the system, they are mostly used during documentation of the software architecture of the system. The structure diagram includes: class diagram, component diagram and object diagram.

i. **Class diagram:** A structure diagram depicting the structure of a system by displaying the system's classes, their properties, operations (or methods), and the connections among objects is referred to as a kind of static structure diagram.

ii. **Component diagram:** In software modeling, these functions are often used to simulate the appearance of the physical elements of object-oriented systems used for visualizing, defining, and documenting components and also to build executable systems via reverse and forward engineering.

iii. **Object diagram**: in order to explain the static structure of a system at a given moment, provide an account of its present state. To check if class diagrams are accurate, they may be utilized.

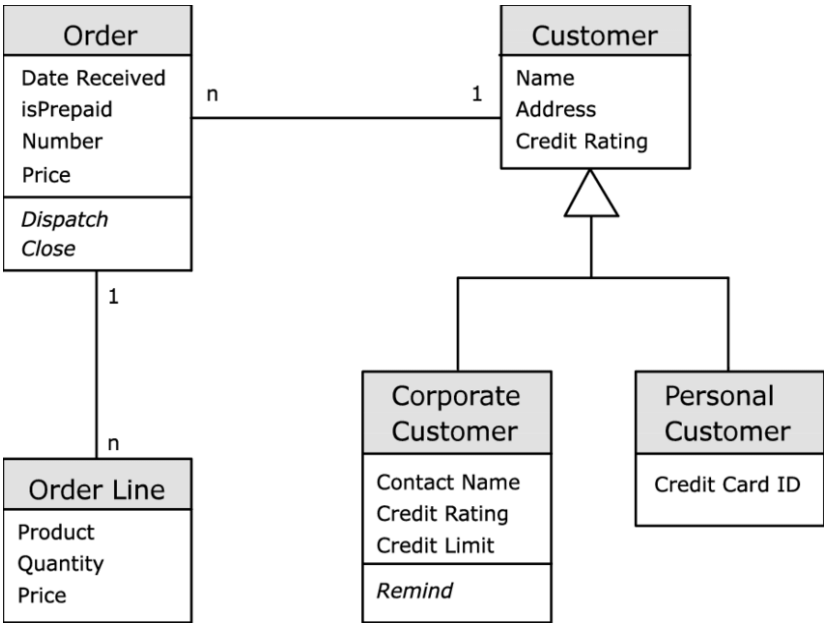


Figure 2.7: UML Class Diagram

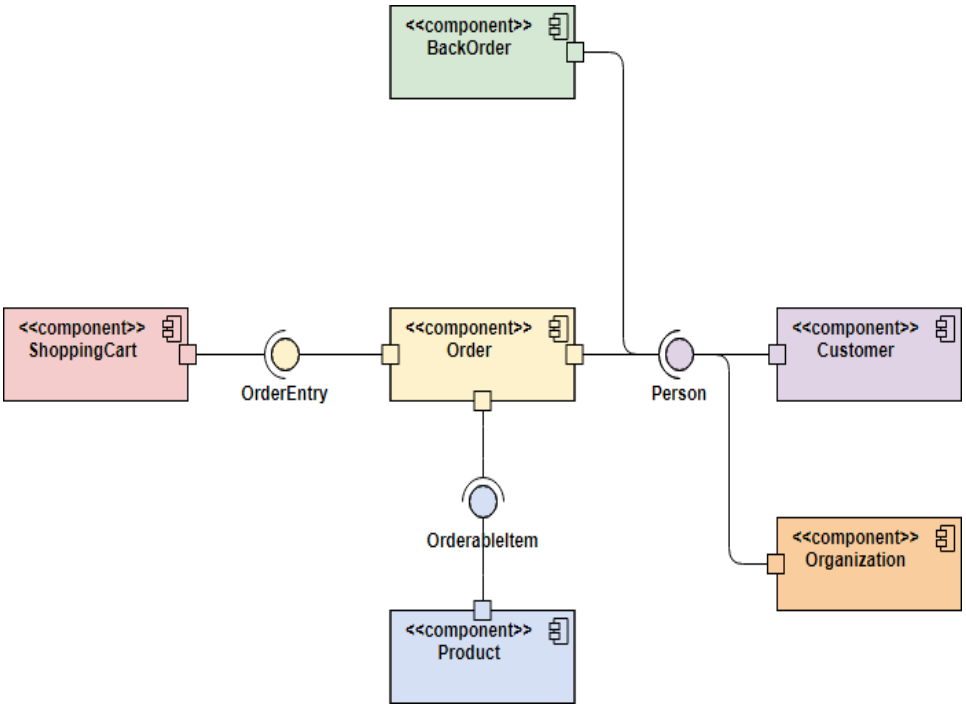


Figure 2.8: UML Component Diagram

iv. **Composite structure diagram**: A classifier's internal structure is shown, as well as its interactions with the environment through ports, or the behavior of a joint effort, for example, in this way.

v. **Package diagram:** Class diagrams are occasionally seen as a distinct approach by developers. In order to minimize dependencies between system components, they organize them in a certain way.

vi. **Deployment diagram**: as a result, it provides a static deployment perspective of an architecture.

vii. **Profile diagram:** In a profile package, a profile diagram is any diagram that is produced. The UML may be extended using profiles.

b. Behavior Diagram

These may be used to show the time-dependent components of a system, as well as their dynamic ideas and how they connect to one another.

i. **Activity diagram:** With support for choice, iteration and concurrency, the process is represented graphically.

ii. **Use case diagram**: User scenarios help you connect what you need from a system to how it meets those needs.

iii. **State machine diagram:** Using finite state transitions, it illustrates discrete behavior of a specified system component.

iv. **Sequence diagram**: It depicts the series of messages exchanged between the objects required to carry out the scenario's functionality.

v. **Communication diagram**: It displays the sequence of messages that must be sent between the objects in order for the scenario to operate properly.

vi. **Interaction overview diagram**: In this diagram, nodes may include additional interaction diagrams, which together form a flowchart.

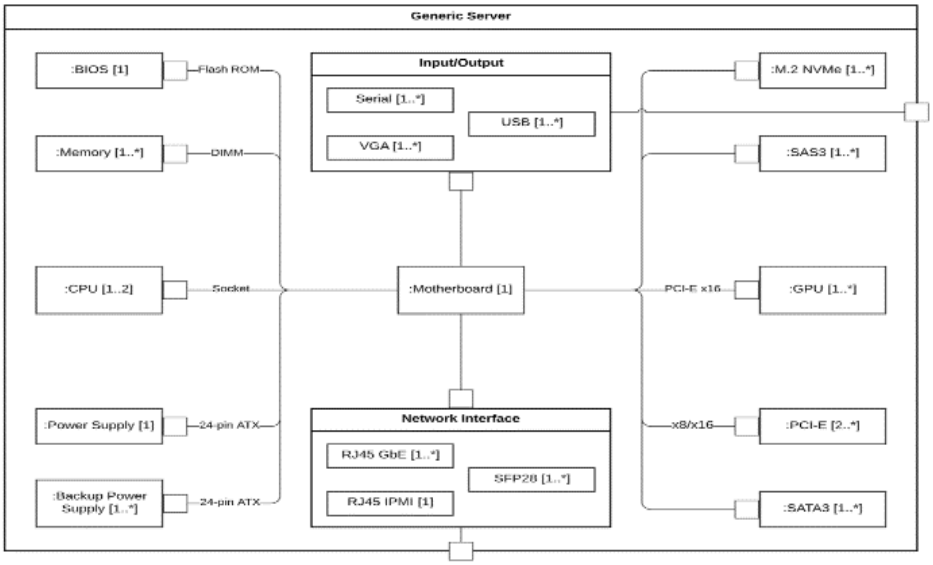


Fig 2.10: UML Composite Structure

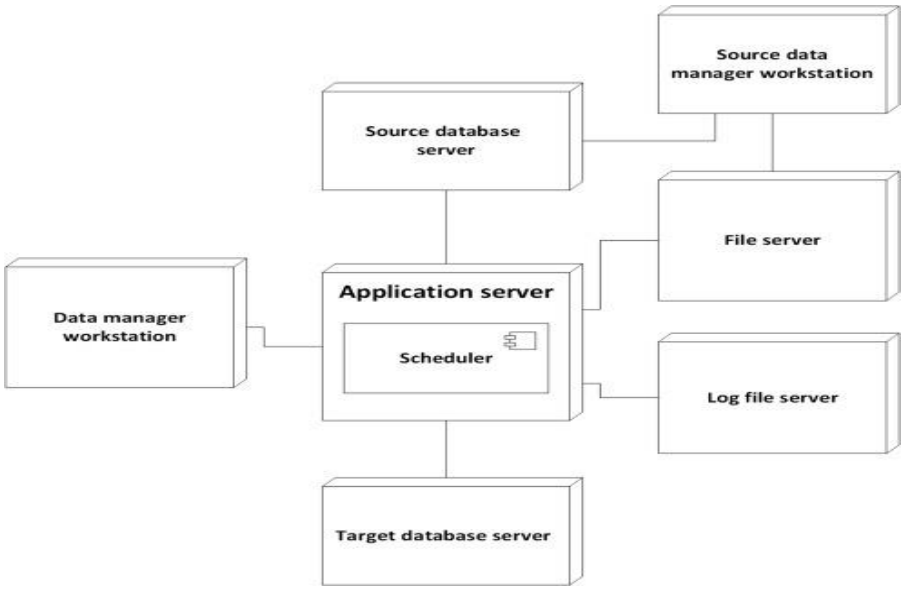


Fig 2.12: UML Deployment Diagram

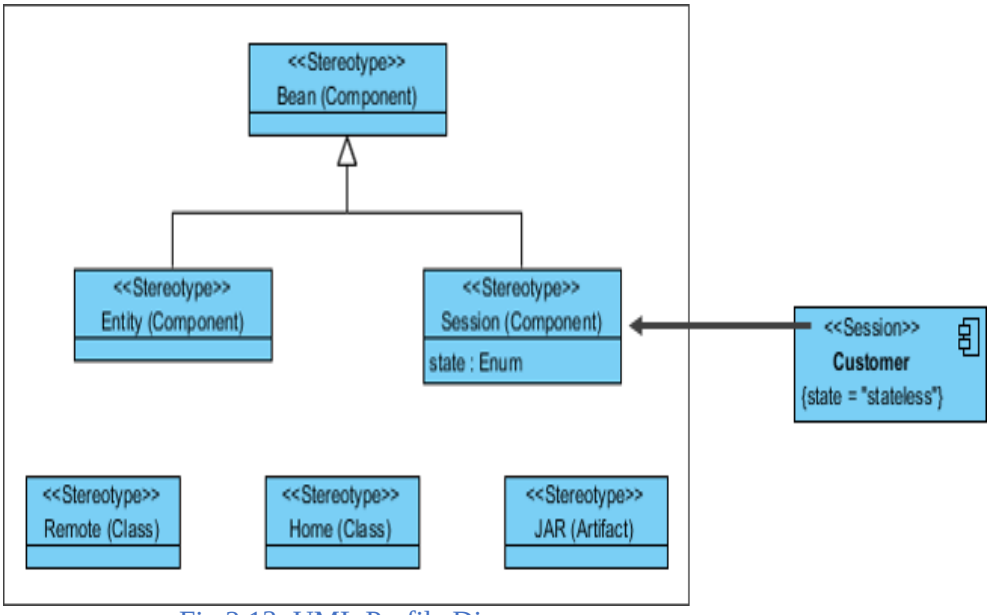


Fig 2.13: UML Profile Diagram

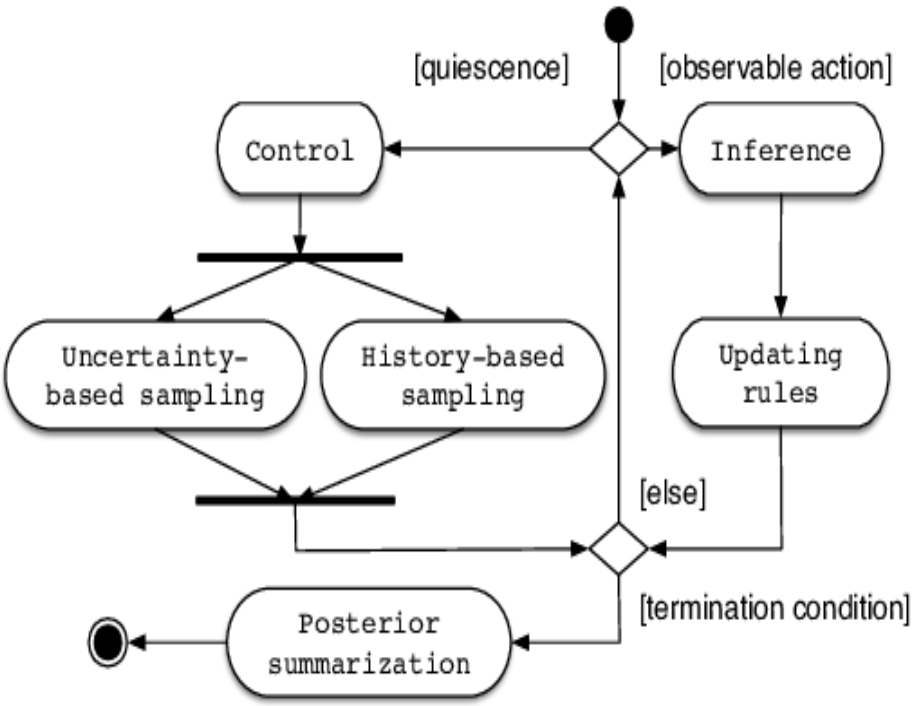


Fig 2.14: UML Activity Diagram

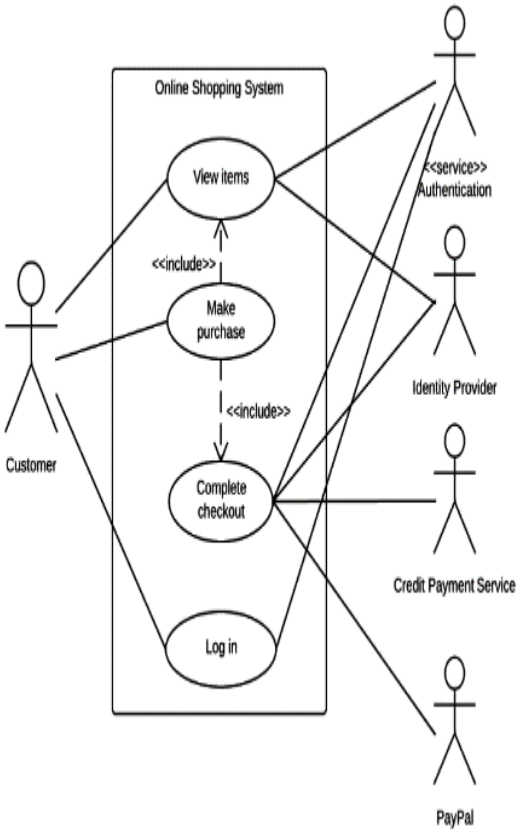


Fig 2.15: UML Use Case Diagram

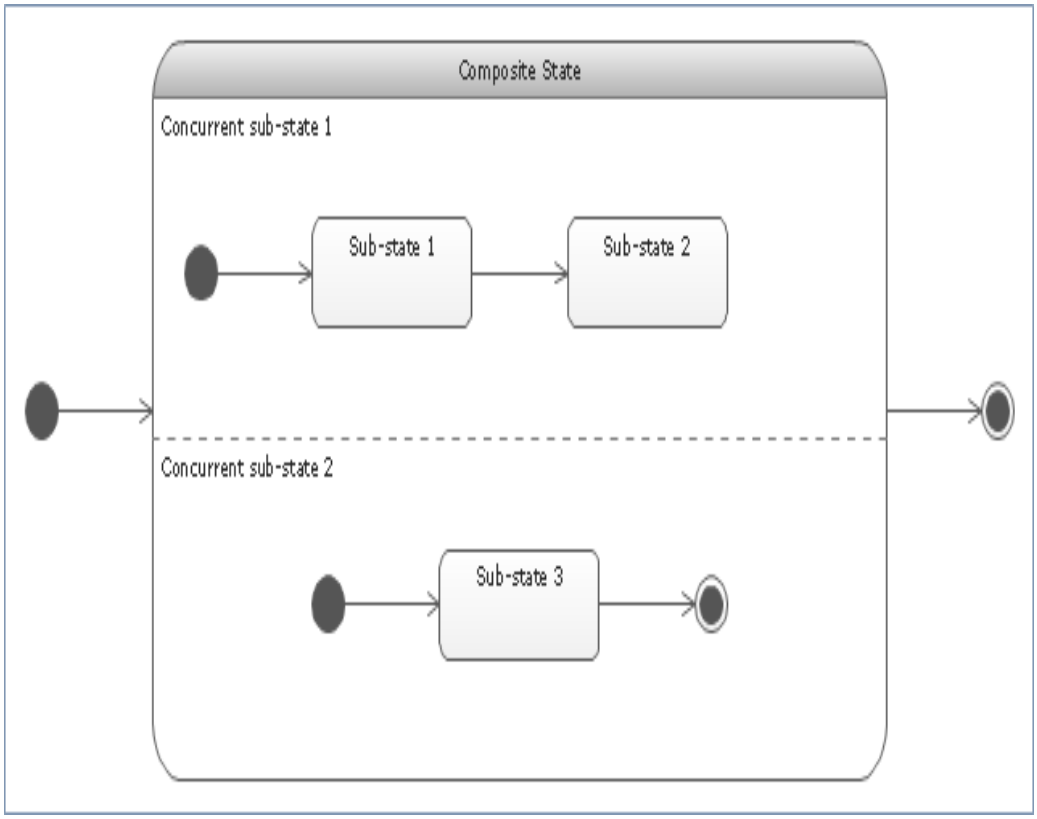


Fig 2.16: UML State Machine Diagram

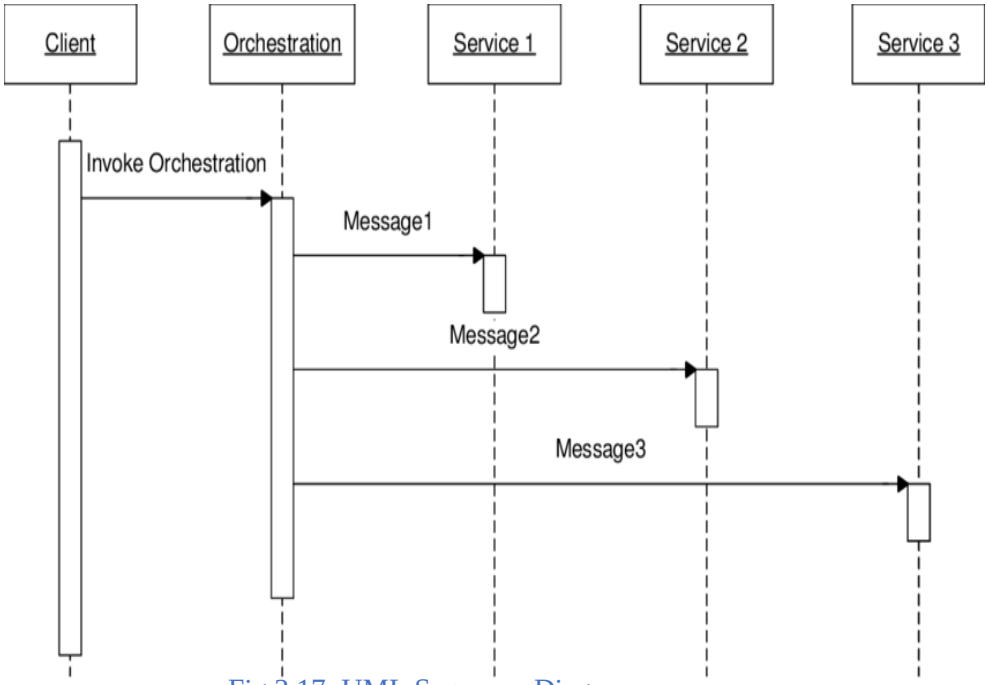


Fig 2.17: UML Sequence Diagram

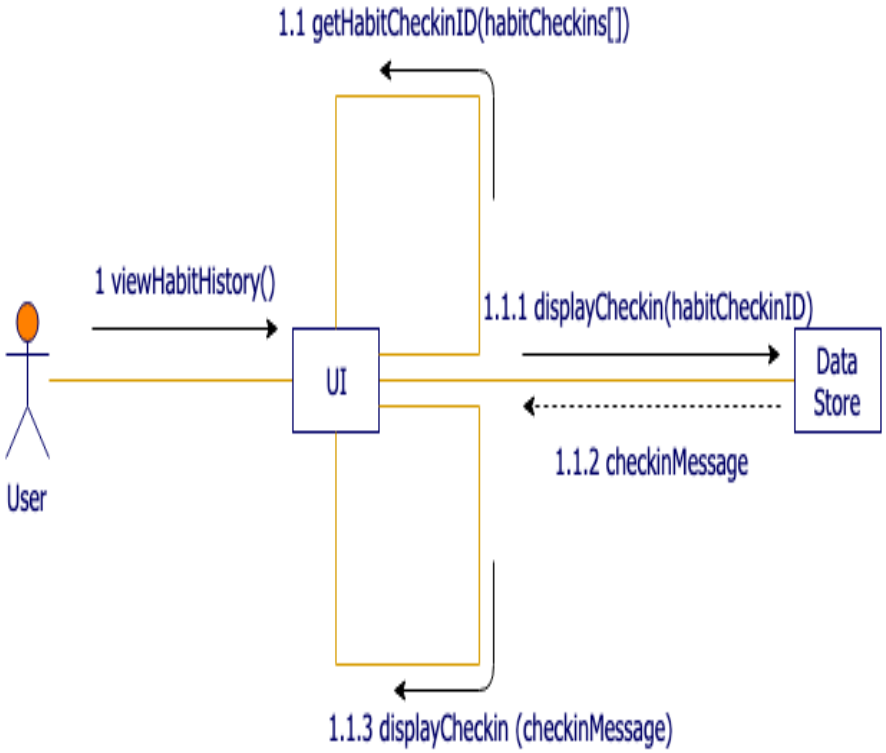


Fig 2.18: UML Communication Diagram

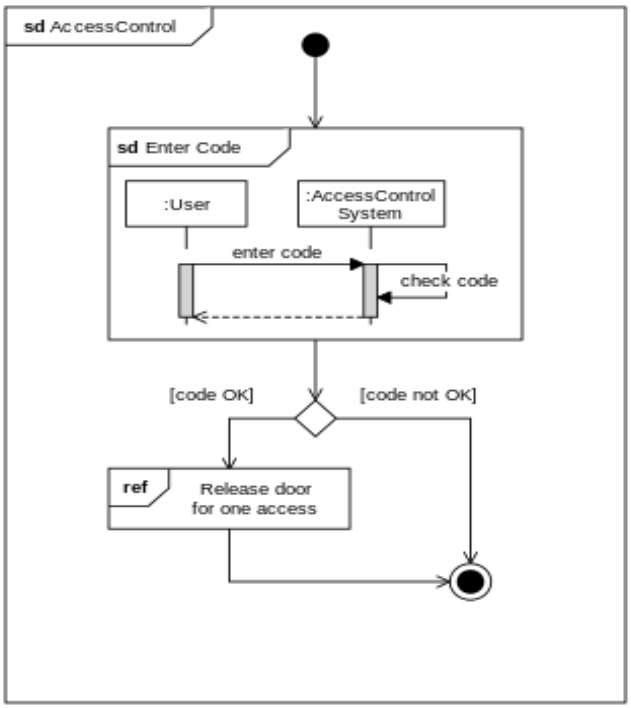


Fig 2.19: UML Interaction Overview Diagram

vii. **Timing diagram:** A timing diagram that focuses on circumstances that change inside and among timelines as they move along a linear time axis illustrates interactions.

2.4.2 UML diagram symbols

There are many types of UML diagrams, each with a slightly different set of symbols. Class diagrams is part of the most commonly used UML diagrams, and the symbols in class diagrams focus on defining class attributes. For example, there are active class and interface symbols. You can also split the class symbol to show the operations, attributes, and responsibilities of the class. Visibility of any class members are marked by notations of lines are also important symbols that indicate relationships between components.

Generalization and inheritance are indicated by unfilled arrows. Composition is shown in a filled diamond. Aggregation is displayed as an unfilled diamond. Dependencies are represented with a broken arrow lines. << >> is used to indicate the properties of that dependency. Package diagrams have symbols defining a package that look like a folder. Activity diagrams have status and activity symbols, including separate starting and ending status symbols. Control flows are typically indicated by arrows and object flows are indicated by dashed arrows.

2.5 System Development Tools

To understand the system developed, we are making use of a software framework called Flutter for the front end and firebase for the backend and the database.

2.5.1 Flutter

Google developed Flutter, a free and open-source UI software development kit. It provides a single codebase which enables you to build native mobile apps for Android,:

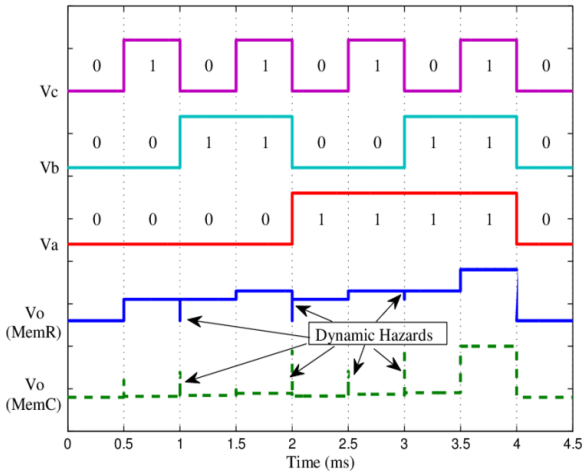


Fig 2.20: UML Timing Diagram

Figure 2.21a: UML Diagram Symbols

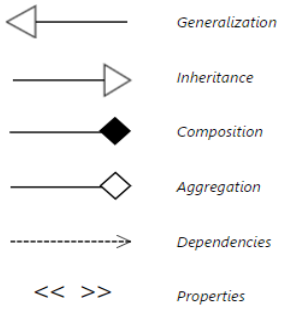


Figure 2.21b: UML Line Symbols

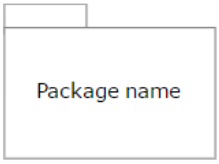


Figure 2.21c: UML Package Symbols

Figure 2.21d: UML State Symbols

iOS, Linux, Mac OS X, Windows, Google Fuchsia, and the web. Flutter consists of two important parts, they are:

a. Software Development Kit (SDK): It's a collection of software tools and programs that developers utilize to create apps for certain platforms. It tools will feature a variety of items that developers may utilize and incorporate into their own projects, such as libraries, documentation, code examples, workflows, and instructions. SDKs are software development kits that are tailored to certain platforms or programming languages.
b. A Framework (UI Library based on widgets): These are collection of UI elements such as text inputs, buttons, sliders which are reusable and can be personalized for one work.

Flutter uses DART programming language. Google developed dart in October 2011. It focuses more on front end development of mobile and web application. Dart is a typed object programming language and have similar syntax to JavaScript.

2.5.2 Framework architecture

The major component of flutter framework is:

a Dart Platform: Apps written on flutter are written in the Dart programming language, which has numerous sophisticated capabilities. The Dart virtual machine for Windows, macOS, and Linux is equipped with a timely runtime. This enables updates to the source files to be sent to an application that is running as it is being written and debugged using Flutter's Just In Time compilation technique. Changes to the source code of an app are reflected in the running app without requiring a reboot or a loss of state.

b Flutter Engine: Google's Skia graphics package is used for low-level rendering in Flutter's engine, which is mainly developed in C++. Android and iOS-specific SDKs may be used with the toolkit as well. As a portable runtime environment, the Flutter Engine may be used to host Flutter applications. Other features include animation and graphics as well as file and network I/O

and accessibility support. The Flutter Framework, which comprises of a reactive framework and a collection of platform, layout, and foundation components, is used by the majority of developers when interacting with Flutter.

c Foundation Library: The Foundation library, written in Dart, contains fundamental classes and methods for building Flutter apps, such as APIs for communicating with the engine.

d Design Specific Widget: There are two types of widgets in the Flutter framework, each with its own design language. Cupertino widgets utilize Apple's iOS Human Interface guidelines, whereas Material Design widgets use Google's equivalent design language.

2.5.3 Firebase

As a cloud-based NoSQL database, Firebase Realtime Database allows you store and sync data in real time across your users. The Google Cloud Platform app development platform Firebase began as a YC11 company and has evolved into a next-generation backend-as-a-service platform. As a result of Firebase, developers are better able to concentrate on creating greater user experiences for users. As a result, users don't need to maintain servers or create APIs since Firebase is their server, API, and datastore, and it's designed in a way that allows them to customize it to meet their specific requirements.

2.6 Related Works

Mayurkumar Patel (2015) developed an online meal ordering system for a restaurant utilizing JavaScript, JSP, HTML, and the Java Oracle Database The system built streamlines the ordering procedure for both consumers and restaurants, and it automates the whole order-taking process. Customers choose one or more items to

place an order which lands in the cart, customer view order details in cart before checking out and receives an order confirmation details. The methodology used in the system modelling is agile.

Adithya, Abhishek and Salma (2017), developed an online food ordering system by using Java (front end), MySQL (backend) and Android Studio(emulator) to develop a mobile application that is reliable, convenient and accurate in taking huge amount of orders at a time and automatically produces the bill.

Varsha, Priya, Snehal and Priyanka (2015), developed a customizable online food ordering system using a web-based application and a mobile based application to combine wireless technology and smart phone technology to automate food ordering system. The system minimizes imperfection in the conventional system by reducing the working of the restaurant, obtaining feed-back from customers and providing the restaurant a means of review of their service.

Muminur et al (2015), worked on a responsive online food ordering system for restaurants which helps customers to order foods online. The system developed aim to build web-based system to manage the online orders and view status. The system also tries to attract and acquire users trust by integrating social media platform such using Facebook API. Also, it was developed using Laravel 5.4 PHP framework for the back-end and a bootstrap for the responsive front-end. The developed system was modelled using a waterfall model.

Tan and Debashish (2018), worked on an Online Food System for University Students to Maintain Healthy Eating Habits which was developed using mobile application. The technology allows students to order meals using their online mobile application, gives ingredient and nutritional information for all menus, minimizing the motivation for students to miss a meal or eat something unhealthy, and assists them in avoiding

eating ingredients to which they are allergic. Moreso, the system was developed using Java Server Faces (JSF) and MySQL, the chosen methodology for the development was Kanban.

Shinde et al. (2014), developed an application for Android operating system-based smartphones where they used Java and SQLite. Compared to PDA-based ordering systems, this Android application processes orders faster, between the customer, restaurant manager, and kitchen, the application can play media.

Saeed et al. (2016), implemented an Android based smart restaurant system, which enables consumers locating free parking space and free table, ordering meals and paying bills via their cell phone. On the other side, the management can supervise the entire job correctly. Mobile and web apps, the Internet of Things (IoT), Near-Field Communications (NFC) sensors, and cloud computing are all part of this system.

This Smart Restaurant Management System has two interfaces for the two types of users in restaurants: An Android mobile app for customers and a web app for restaurant employees.

Gan (2002), developed an online fast-food restaurant ordering system that allows customers to place orders anytime at any place. The technology facilitates the management of client orders as well as the promotion of advertising. Using it, kitchen staff can see order information, management can manage fast food raw materials, and personnel can search for client delivery and profile information. When used during peak hours, this method helps to decrease queues, accelerate food preparation, and boost client numbers.

Dhore et al. (2014) proposed an Android-based system to automate the restaurant's ordering procedure, allowing one or more customers to order meals, reserve a seat,

and pay for it all from afar. Both of these works implemented functions that were almost identical.

Leong (2016) also developed, a system for the Android operating system was built, which can be used to place food orders online. In order to see and verify orders and payments, computers must be installed in the kitchen and at the manager's desk.

Table 2: Summary of Related Work

S/N	Author (Year)	Title	Scope	Methodology	Results	Remarks
1	Mayurkumar Patel (2015)	Online Food Order System For restaurants	Web based; JavaScript, JSP, HTML, Java Oracle Database	Agile	Successful implemented	The system simplifies ordering process for customers and restaurant
2	Adithya, Abhishek and Salma (2017)	online food ordering system	Web and mobile based; Java, MySQL	Agile	Successfully implemented	The system is reliable and accurate in taking huge amount of order and automatically calculating bills
3	Varsha, Priya, Snehal and Priyanka (2015)	Customizable online food ordering system	Web and mobile application	XP	Successfully implemented	
4	Muminur et al (2015),	responsive online food ordering system		Waterfall	Successfully implemented	
5	Tan and Debashish (2018)	Online Food System for University		Karban	Successfully implemented	
6	Saeed et al. (2016)	Android based smart restaurant system		XP	Successfully implemented	
7	Gan (Gan, 2002)	online fast-food restaurant ordering system			Successfully implemented	
8	Dhore et al. (2014)	Android based system to automate ordering process			Successfully implemented	

CHAPTER THREE

METHODOLOGY

3.1 Method of Identification of User and System Requirement

Although there are different methods of identification of user and system requirement, the main source of identification of user and system requirement used in this project are:

a Primary source: Primary source refers to the means of collecting data in which requirement or data were collected using an empirical approach such as informal survey and personal interview.

b Secondary source: Secondary Source of data were obtained from journals, library source, magazines and newspapers.

i. Informal survey: The informal survey is one of the methods used in the collection of user and system requirement. Informal survey involves observation of already existing system.

ii. Personal interview: It is a means of collecting data from a respondent through asking of questions directly from respondent and collecting of information with the aim of facilitating understanding. The interview was done between the researcher, university staff and cafeteria vendors. Reliable answers that help in understanding the requirement of the system and user were gotten based on the questions asked on the development of the system.

3.1.1 Identification of system requirement

The system requirements are more specific descriptions of the software system's functions, services, and operational limitations. The functional specification, often known as a system requirements document, should specify exactly what is to be

implemented. The computer system is made up of different unit that work together to achieve a goal. The requirement of the implementation of the system are:

functional requirement of the system, non-functional requirement of the system. software requirement and hardware requirement.

a Functional Requirement

These are assertions about what services the system should give, how it should react to specific inputs, and how it should operate in specific scenarios. The functional requirement of the system developed are:

i. Staff and Vendors should be able to register and login to the system
ii. Staff and Vendors should be able to retrieve password if forgotten
iii. Staff should be able to view available food
iv. Staff should be able to select food item of their choice
v. Staff should be able to add selected food item to cart
vi. Staff should be able to remove food items from cart
vii. Staff should be able to give a full description of their location for delivery
viii. Staff should be able to cancel already made orders before checking out
ix. Vendors should be able to create(add), update and delete food items in food menu.
x. Vendors should be able to receive orders from customers
xi. Vendors should be able to respond to submitted order so that users can know if their order is confirmed.

b Non-Functional Requirement

These are limitation on the system's services or operations, such as time, development process, and standards-imposed constraints. The following are the non-functional requirement of the system

i. The system will display confirmation message to users within 10 seconds after the user submit the order

ii. The system will check if user information correlate with the one in the database before providing access to other functionality of the system

iii. The system will send received order details from users to the vendors within 10 seconds.

iv. The system will permit only registered vendors to create and edit menu

v. The system will permit vendors to view their own page and not others.

vi. During registration, the system will check if user id (Staff ID) and Vendor ID correlate with generated one's by the admin

c Software Requirement

For the effective and efficient implementation of the system, the following software were installed and used:

i. Operating System: Windows 10

ii. Visual Studio Code

iii. Android Studio Emulator

iv. Git Bash

d Hardware Requirement

For the successful implementation of the system, the hardware requirement used are:

i. 16 GB RAM

ii. 500GB Hard Drive

iii. Intel(R) Core (TM) i7 Processor

3.1.2 Identification of User Requirements

The food ordering system must provide certain requirement for the user. The users of the system are:

a. Staff

Staff can register, Staff can login, Staff can manage their account. Staff can view displayed food item, Staff can add food items to menu, Staff can remove food items

from menu, Staff can review their order before submission, and Staff can cancel their order before submission

b. Vendors

Vendors can register, Vendors can login, Vendors can add food item to be displayed, Vendors can remove displayed food item, and Vendors can manage menu list.

c. Admin

Registering Vendors, Removing Vendors, Registering Staffs, and Manage Users.

3.2 System Design Methods

3.2.1 Use case diagram

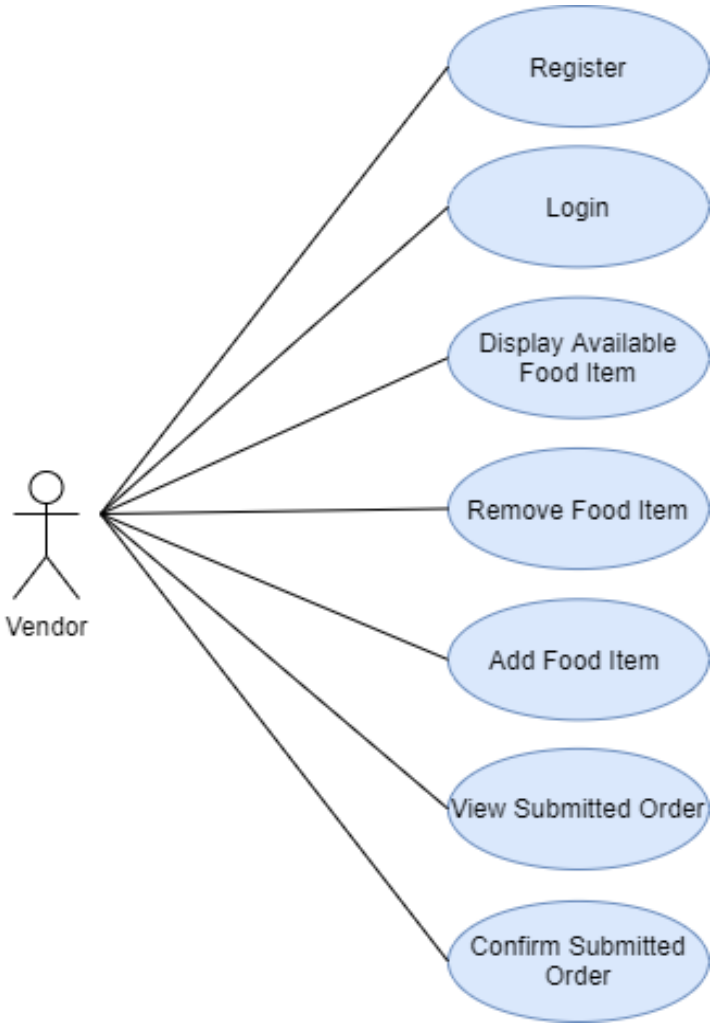


Fig. 3.1: Vendor Use Case Diagram

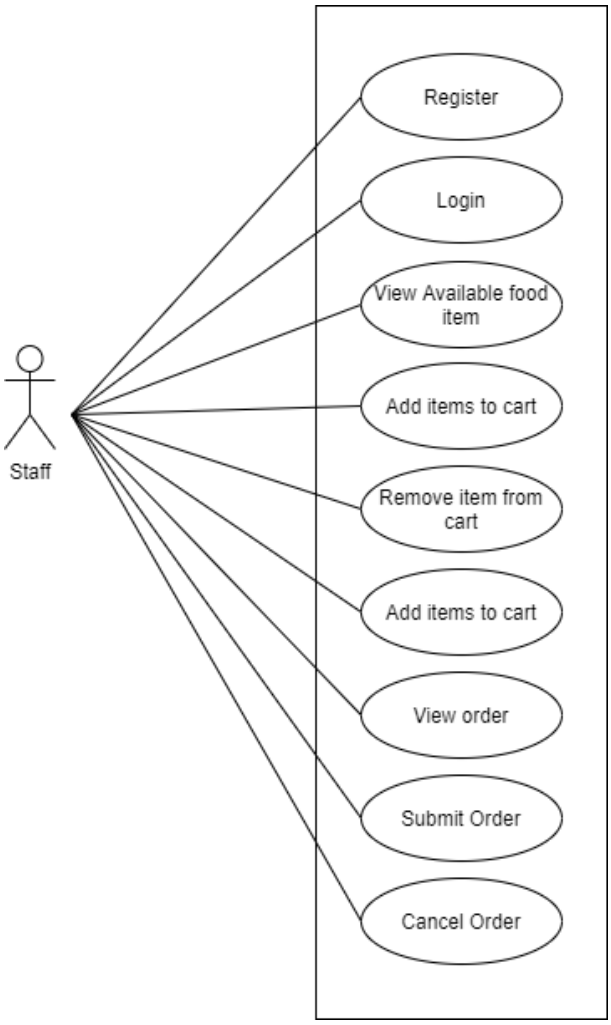


Fig. 3.2: Staff Use Case Diagram

3.2.2 Sequence diagram

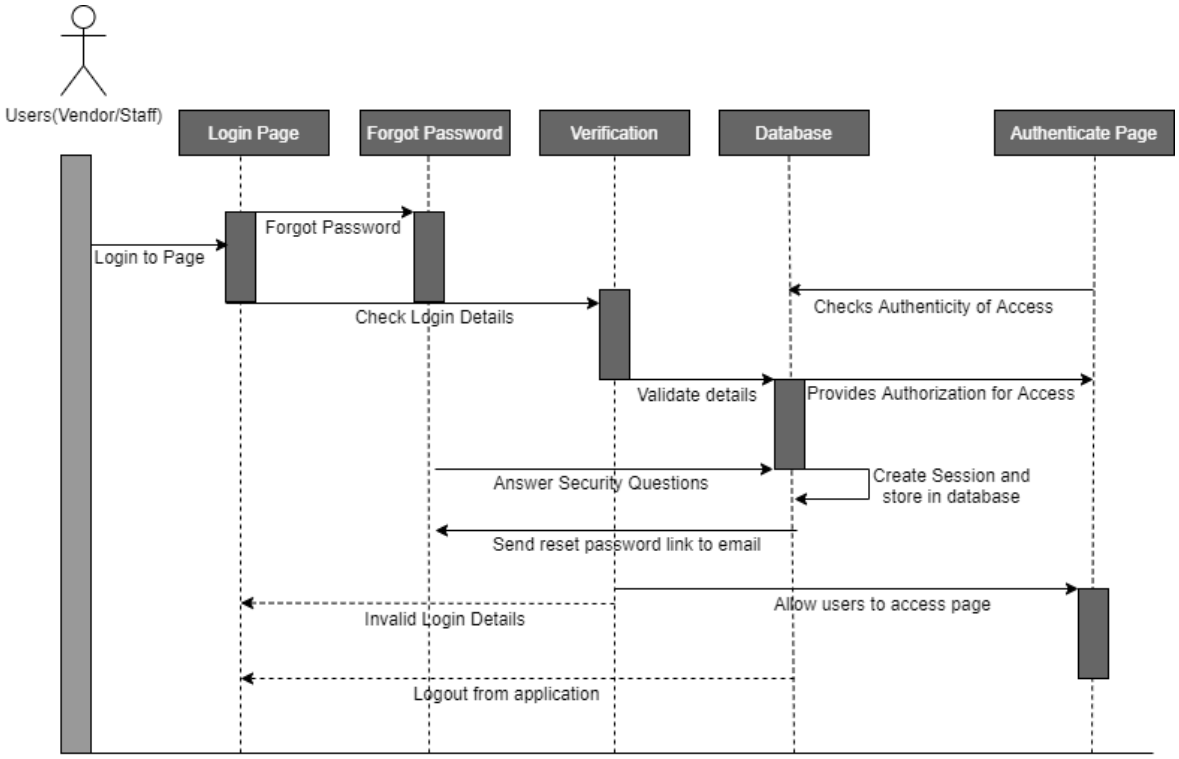


Fig. 3.3: Login Sequence Diagram

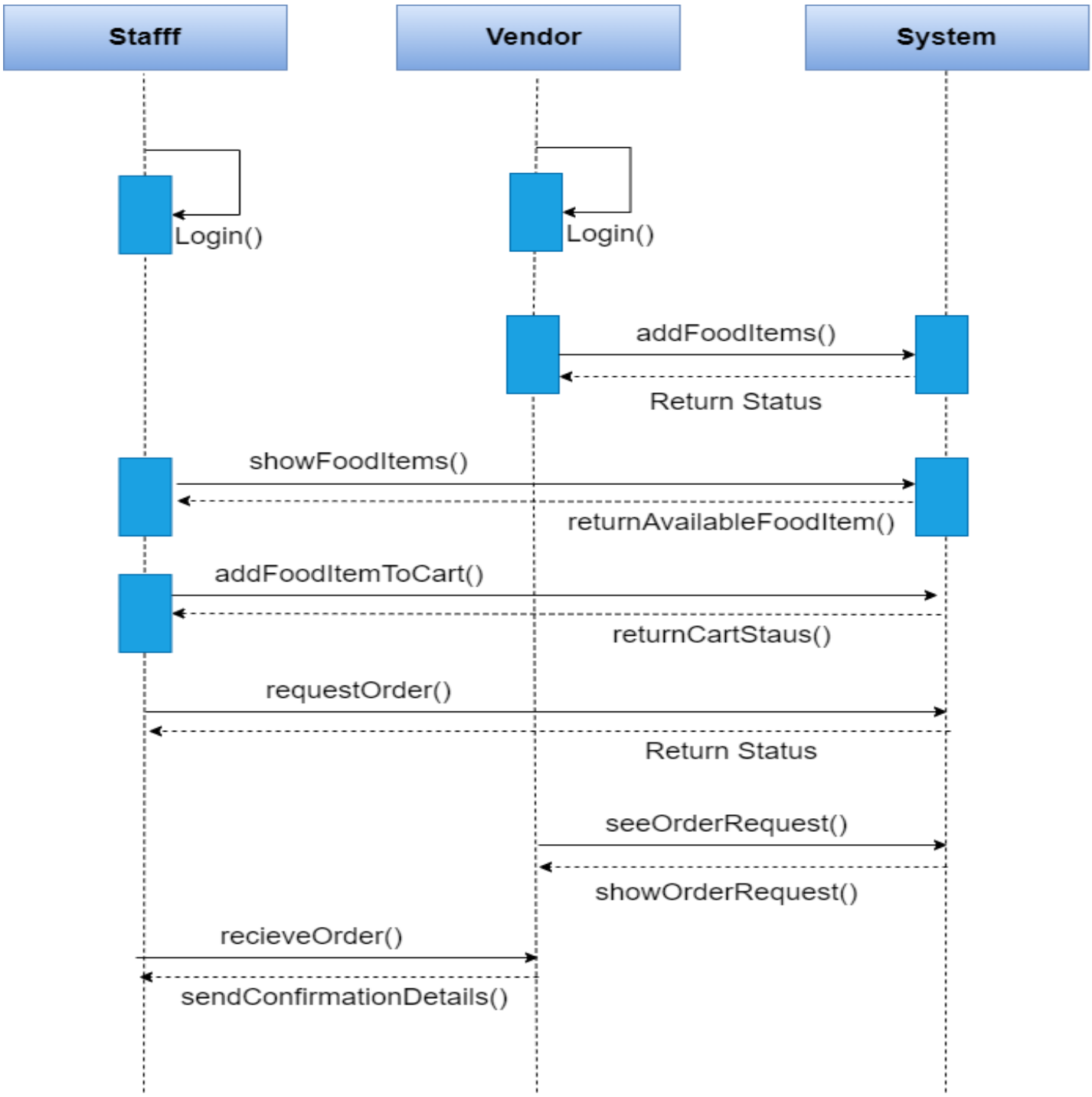


Fig. 3.4: Order Sequence Diagram

3.2.3 Activity diagram

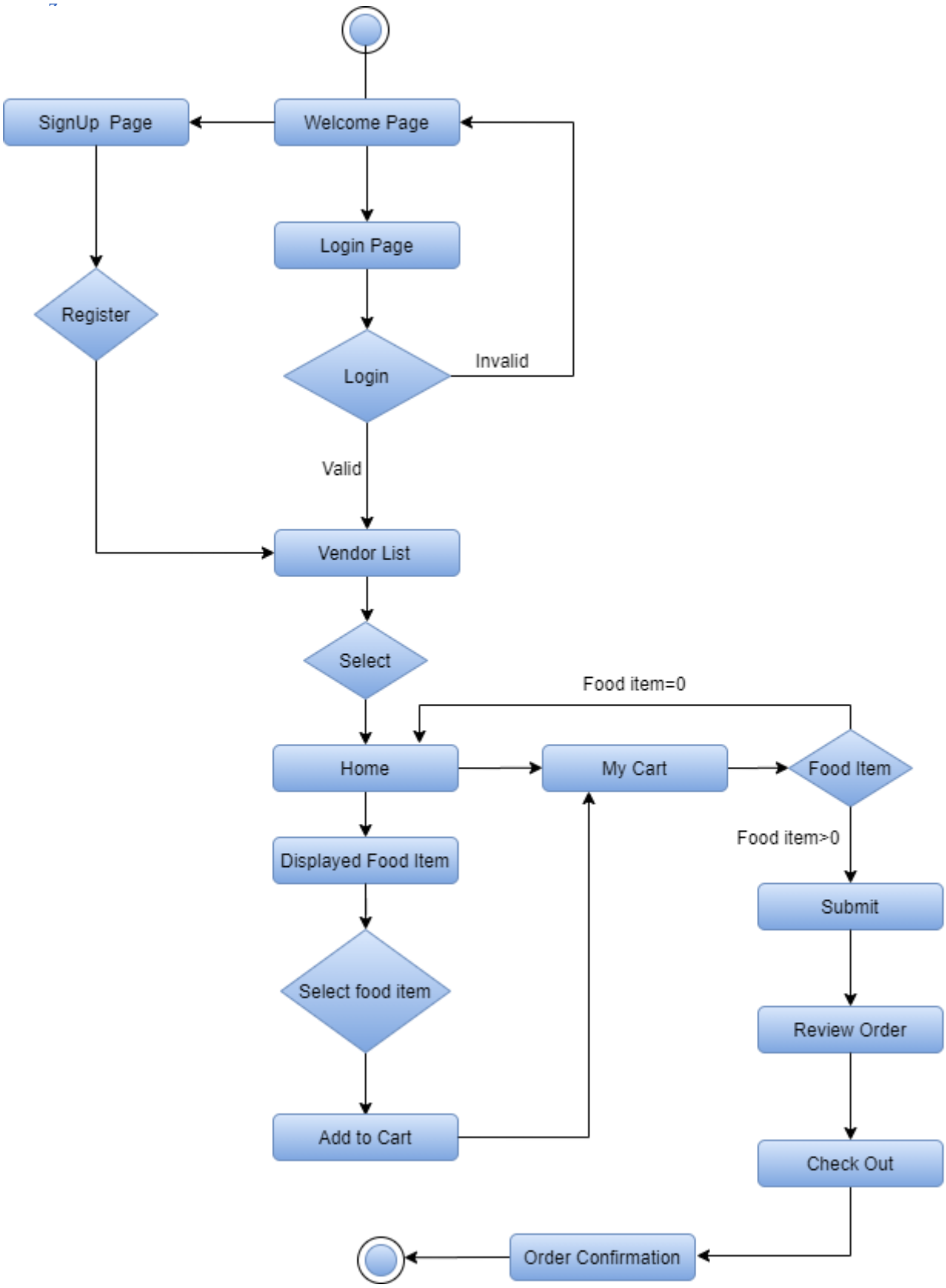


Fig. 3.5: Staff Activity Diagram

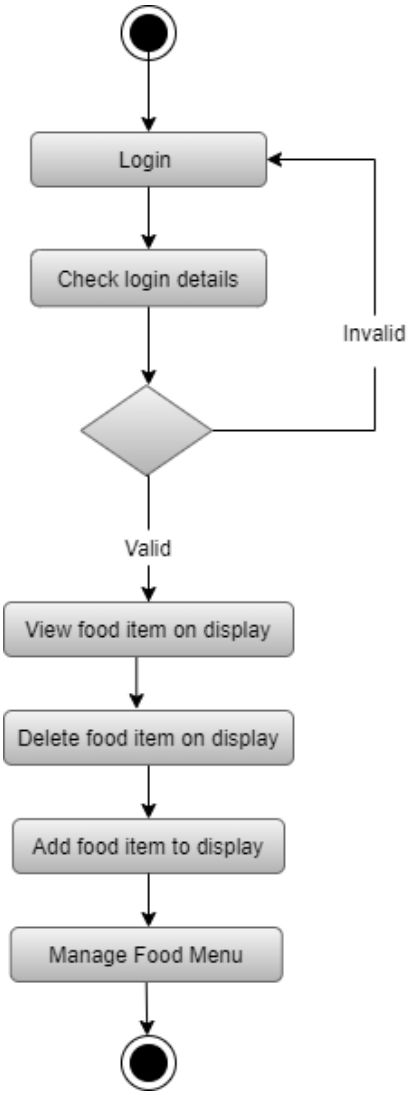


Fig. 3.6: Vendor Activity Diagram

3.2.4 Class diagram

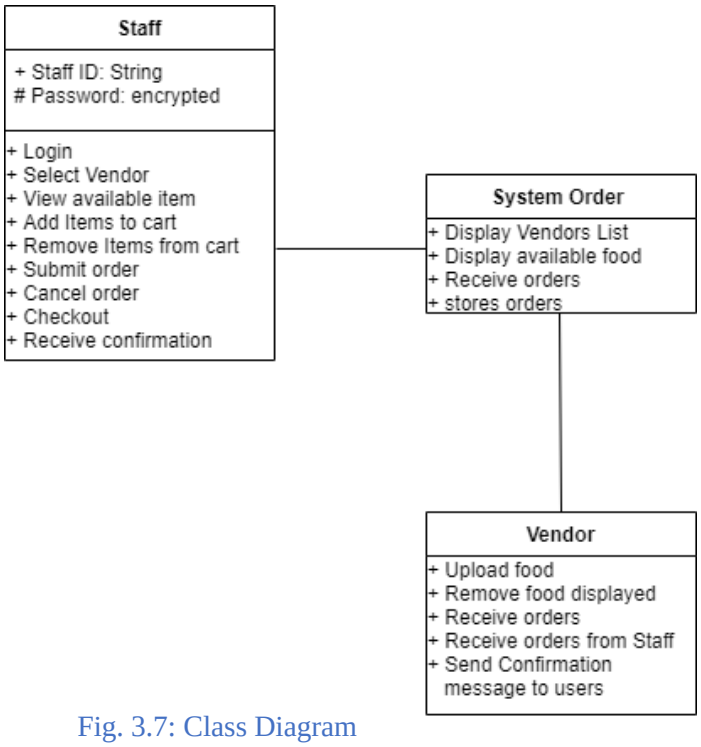


Fig. 3.7: Class Diagram

3.3 System Implementation

The implementation of this system begins by converting the logical structure of the system into physical architecture through coding and development of the system. The system was implemented using the following:

i Technology implemented: Flutter

ii Language: Dart programming language

iii Database: Firebase

iv User interface: Flutter, android emulator

v Software: Android studio

vi Operating system**:** Windows 8 or higher version.

3.3.1 Database implementation

The database of the system was developed using cloud fire-store. The Firebase Realtime Database is a cloud-hosted NoSQL database that stores and syncs data between users in real-time (i.e. a real-time document store). The database of the developed system contains details of the users (staff and vendors) during registration. The database also contain food that are available for view by staffs and also food added to cart or order made by staff.

3.3.2 Front-end implementation

The front-end implementation began by designing how the system user interface will look (rough sketch of the system). Later the sketched user interface was designed using Adobe XD, a software for user interface design. The designed UI was then implemented using flutter framework with the integration of dart programming language while the output user interface was viewed with and android studio emulator.

3.3.3 Back-end Implementation

The backend was implemented using firebase emulator. The firebase emulator authentication package was used to handle the authentication workflow of the various users. The database was modelled using cloud firestore. Cloud firestore adopts the document object model, i.e., it stores data using style JSON syntax.

3.4 System Testing Approach

The developed system was tested by using the unit testing approach, where the behaviour of a single functions, method and class are tested. In flutter, the test package offers the foundation for building unit tests, whereas the flutter test package adds tools for testing packages. Test dependency was added to the pubspec, a test class was created and embedded in a test file, the created test is used to run each unit of the system. Widget test was further used to tests a single widget. A widget test's objective is to ensure that the widget's UI looks and operates as intended. Testing a widget includes several classes and necessitates the use of a test environment that provides the necessary widget lifecycle context. Finally, integration test was used to test the complete system so as to verify that all the widgets and services being tested work together as expected.

CHAPTER FOUR

IMPLEMENTATION AND RESULT

4.1 Database Implementation

This database earlier discussed was implemented with firebase, below shows the offline cloud firestore images which were used in the implementation of the system.

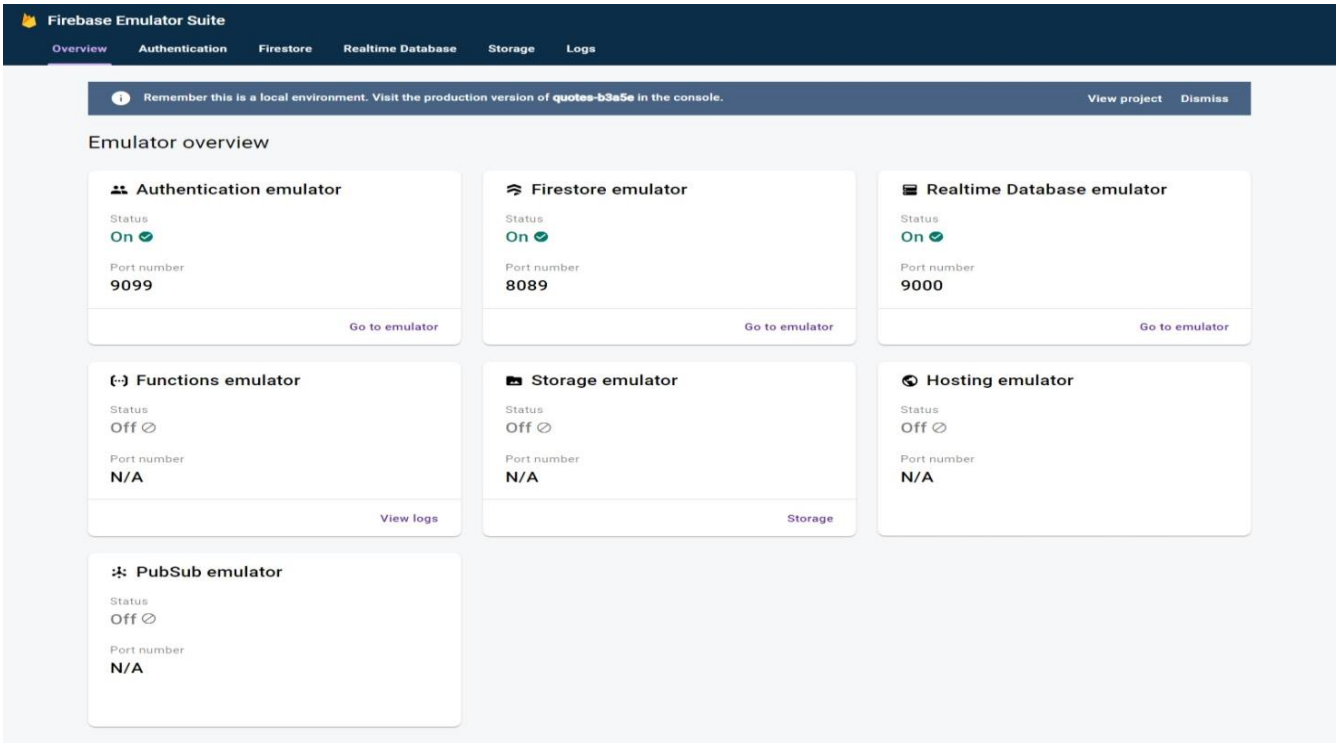


Fig. 4.1: Overview of Firebase Database

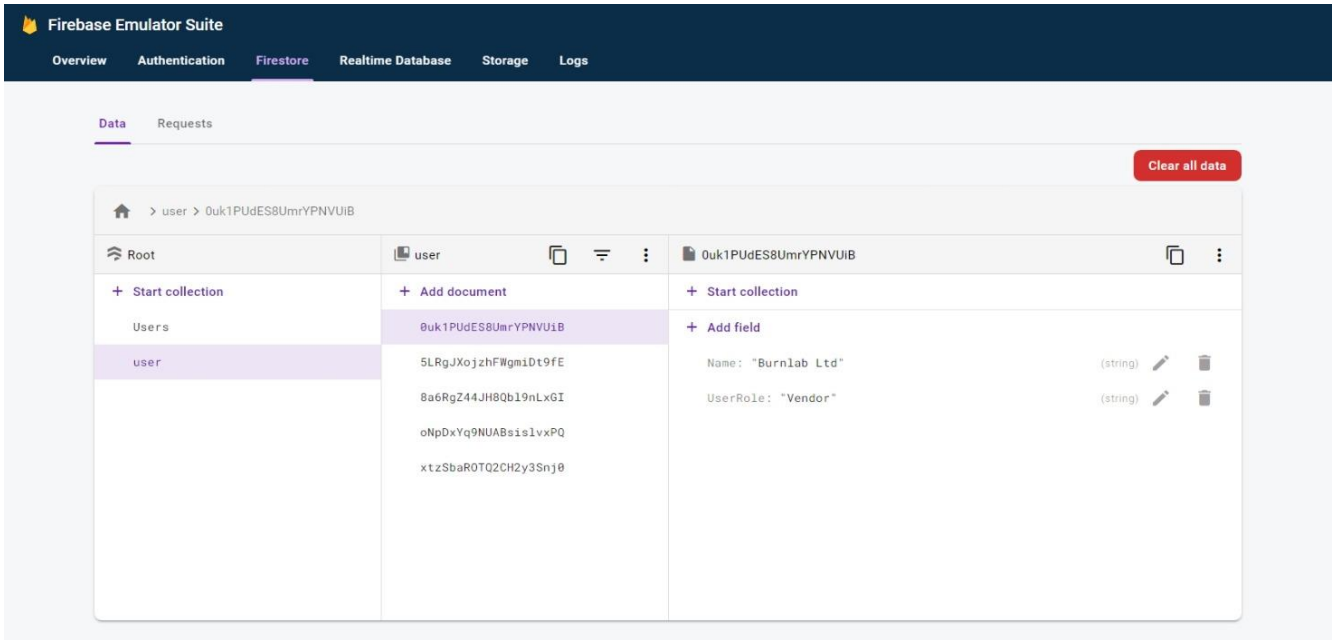


Fig. 4.2: Firebase Database

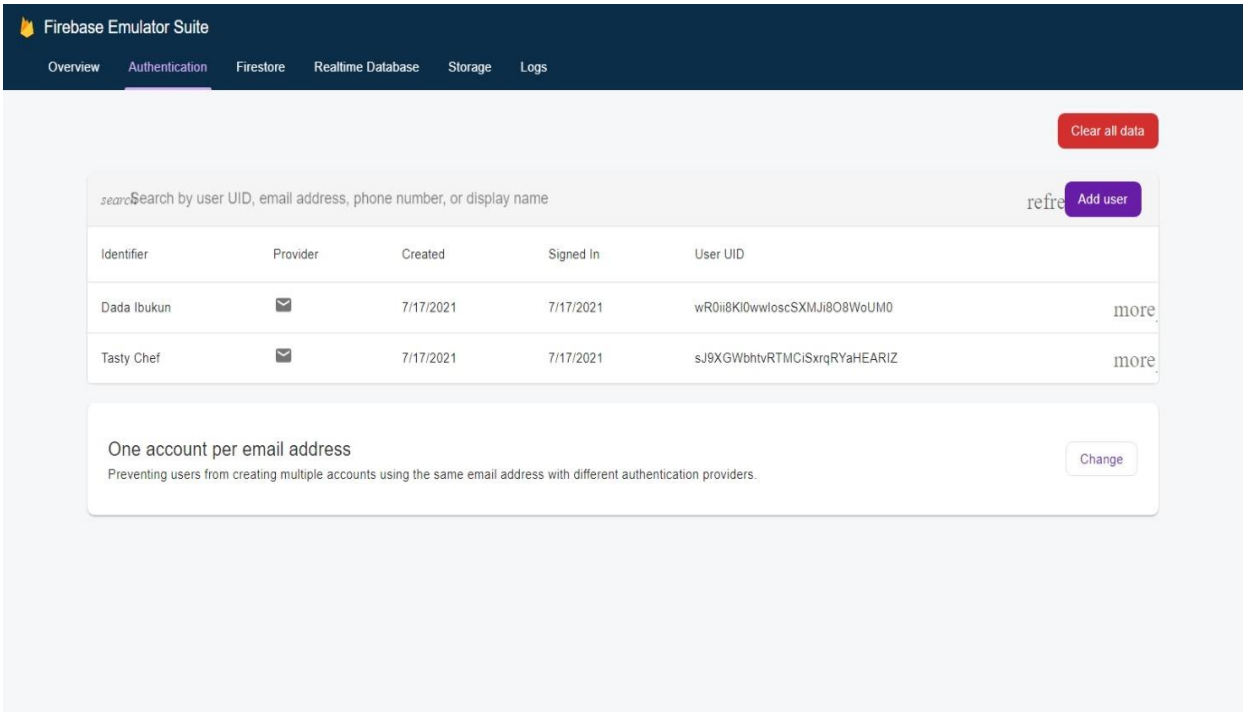


Fig. 4.3: Firebase Authentication

4.2 Result of Implementation of Front-End Interface

The application shows an image of the splash screen and directly shows the login page for staff, below the login button, there is a register button and a Vendor login button. After the staff logs in with his or her details, it takes them directly to the vendors list page where they select vendor of their choice which direct them to the homepage of the selected vendor. On the homepage, there are list of food available for users to add to cart, added item to cart are then submitted with details of the user location for delivery before check out.

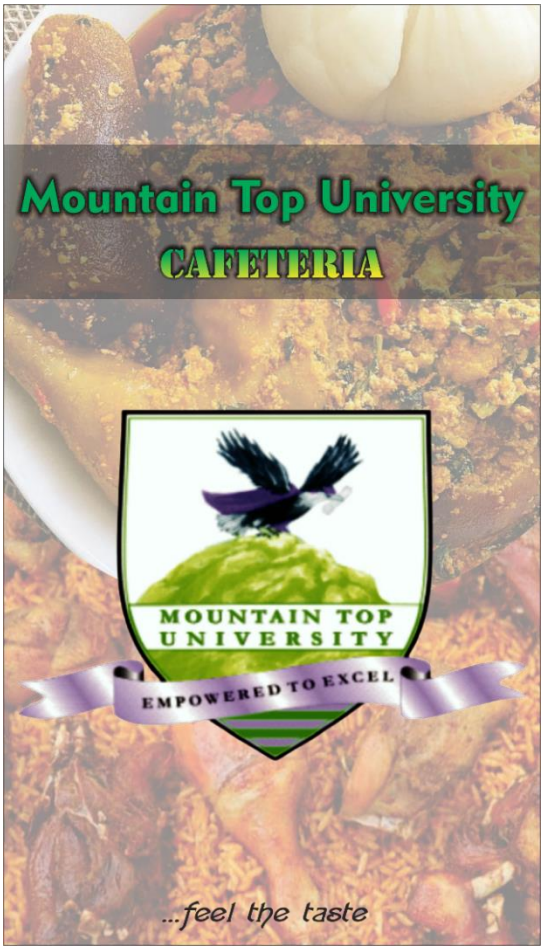


Fig. 4.4: Splash Screen

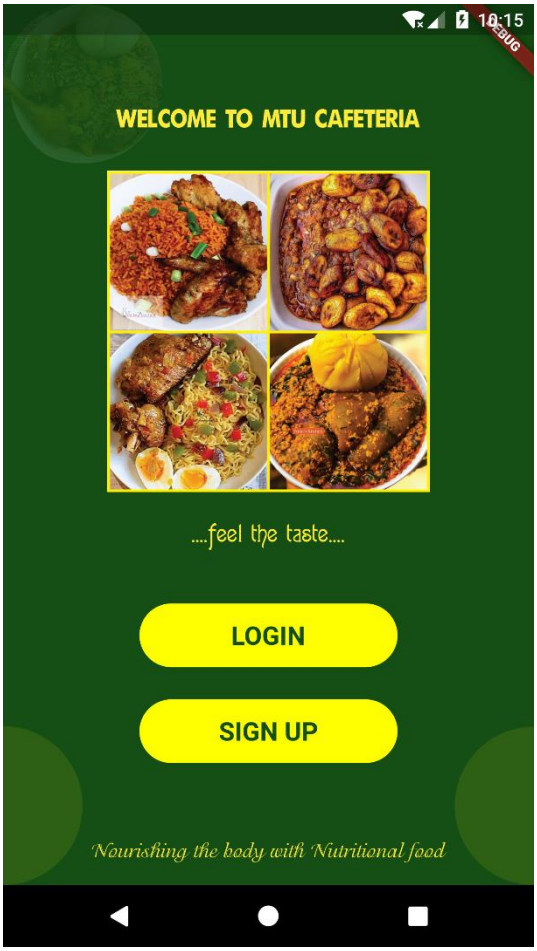


Fig. 4.5: Welcome Page

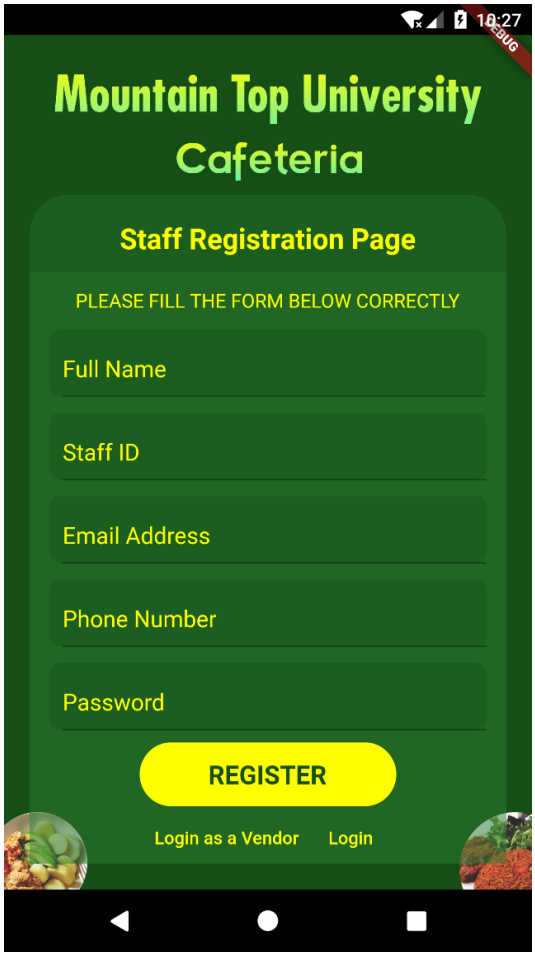


Fig. 4.6: Staff Registration Page

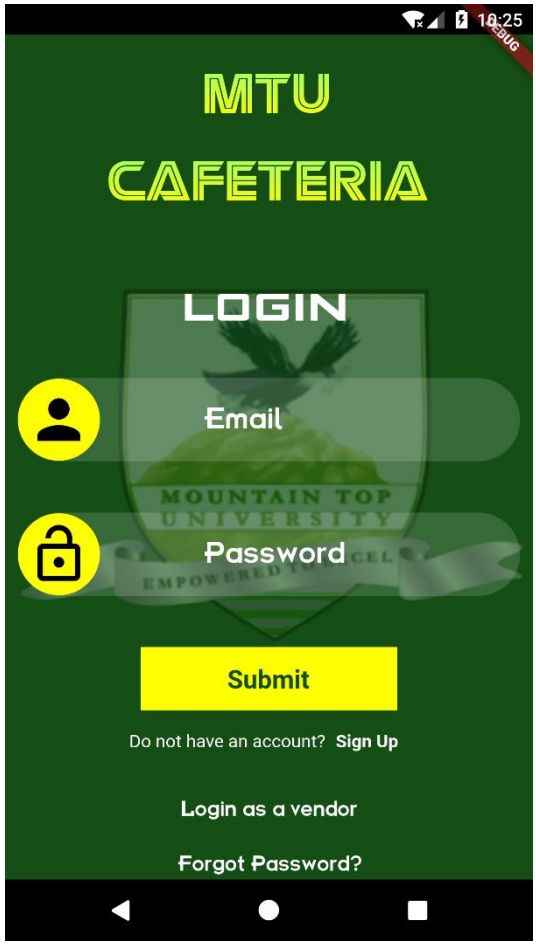


Fig. 4.7: Staff Login Page

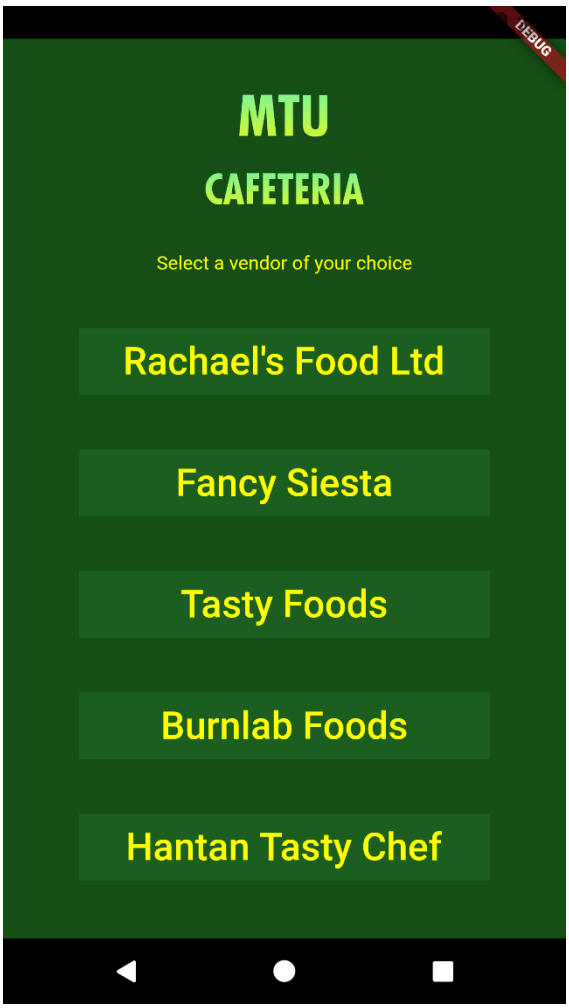


Fig. 4.8: Vendor's List Page

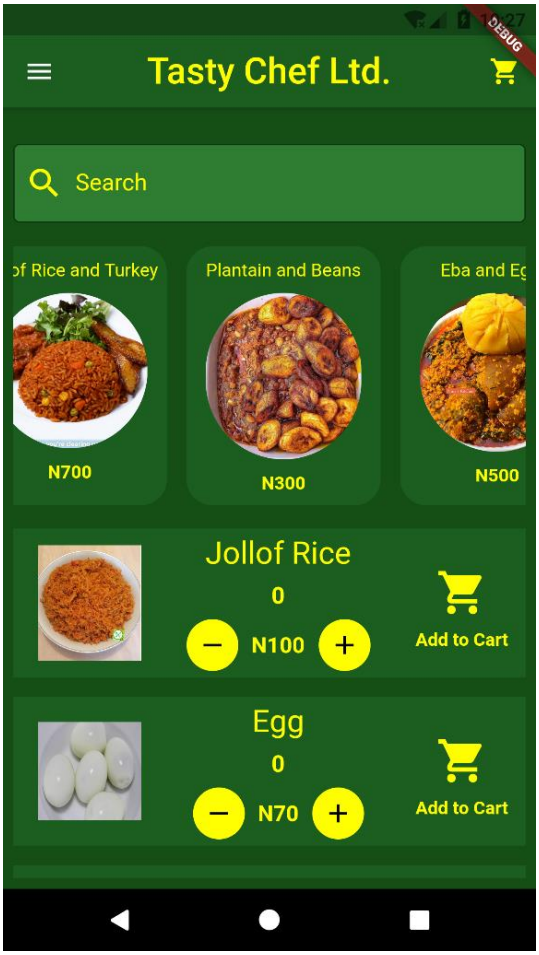


Fig. 4.9: Homepage

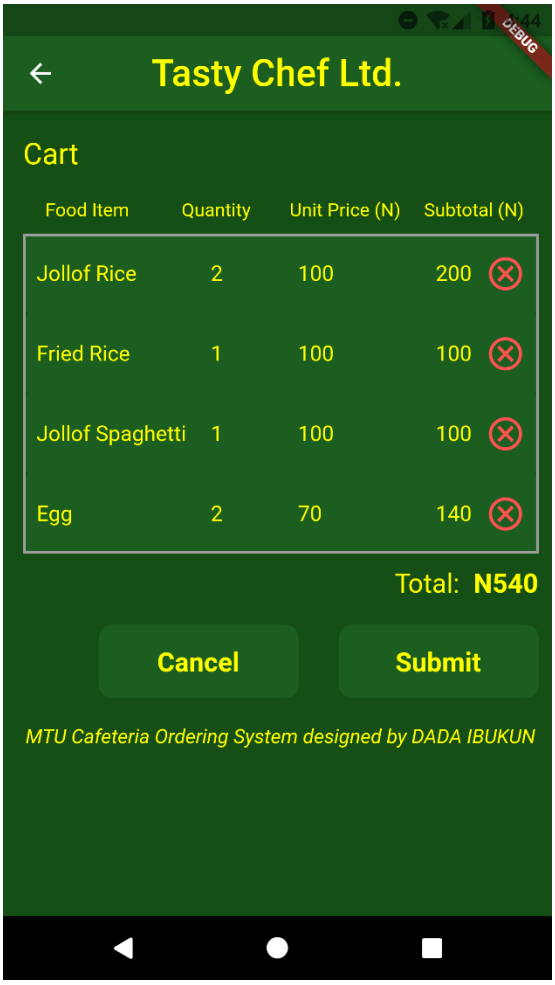


Fig. 4.10: Staff Cart

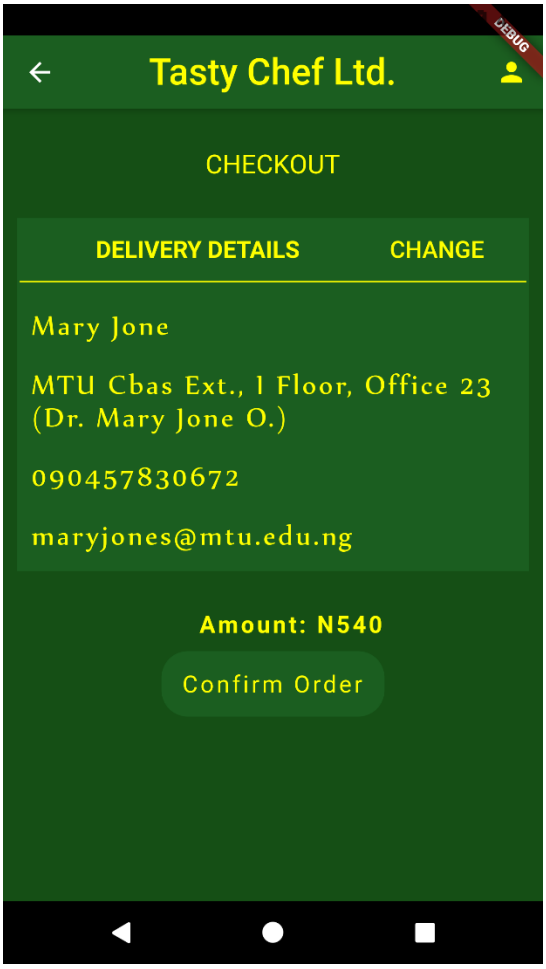


Fig. 4.11: Staff Checkout Page

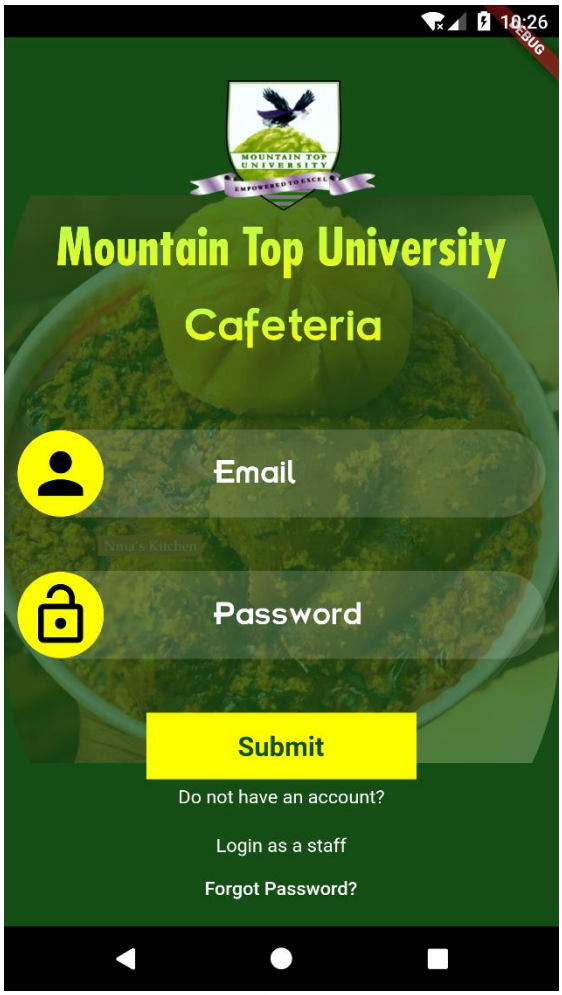


Fig. 4.12: Vendor Login Page

Fig. 4.13: Tasty Chef Homepage

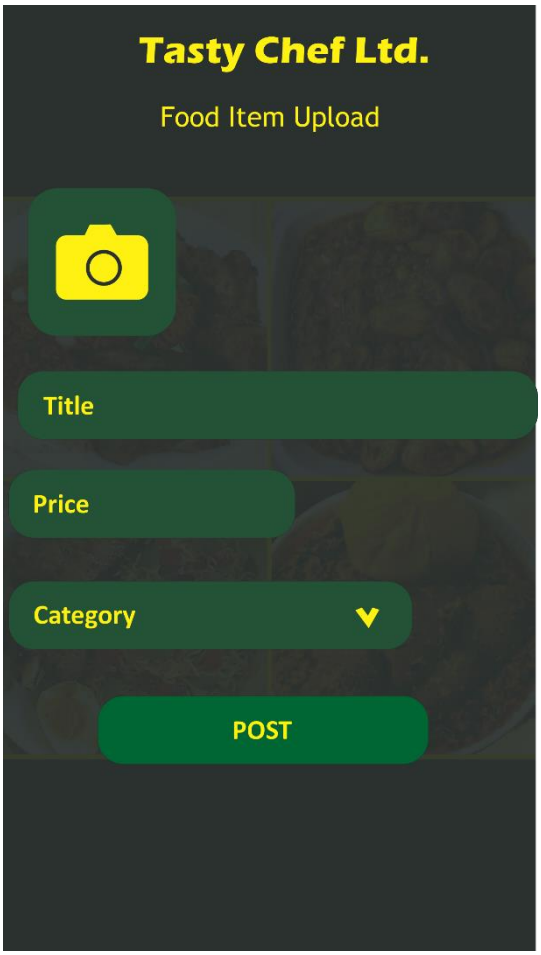


Fig. 4.14: Food Upload Screen

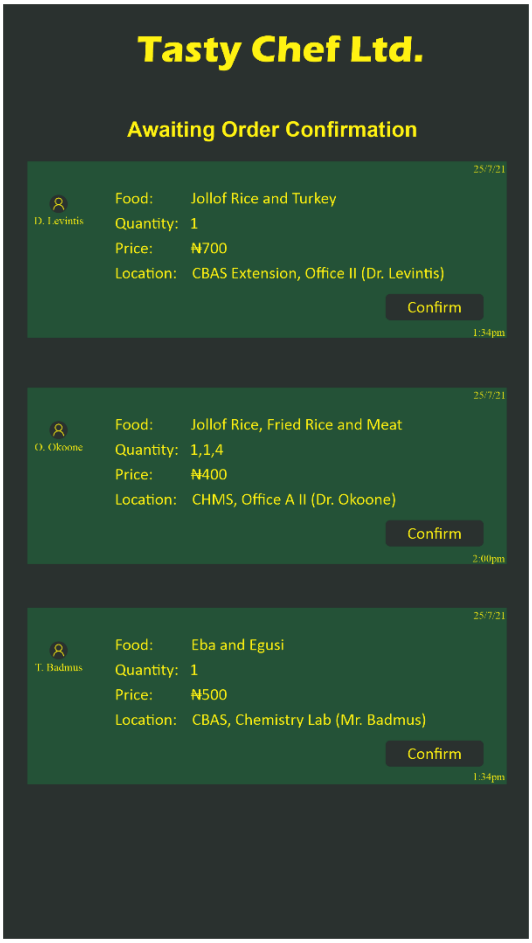


Fig. 4.15: Order Confirmation

CHAPTER FIVE

SUMMARY, CONLUSION AND RECOMMENDATION

5.1 Summary

At the end of this project work, I was able to design and develop software that can successfully handle online food ordering for Mountain Top University Cafeteria. The system will prevent staff from joining queue or help staff to save time by ordering their food on their phone,

5.2 Conclusion

In conclusion, the developed system allows staff of the institution to order for their meal and also get it delivered at a precise location stated by them. On the other hand, the vendor are able to upload available food menu and get responses from their client or customer. This system was implemented using certain tools such as firebase, flutter and an android emulator.

5.3 Recommendation

It is known that for any meaningful computer-based information management to be integrated into any organization, proper training and orientation has to be given both to the staff and management. Proper training should be given to the data entry vendors on how to handle the upload item images into the system. The importance and benefits of the system, as well as how it will benefit them in their many fields of work, should be emphasized to the personnel. They should also be made aware of the costs of sustaining this new technology so that they can manage it with caution.

References

Adithya R., A. S. (2017). Online Food Ordering System. *International Journal of Computer Applications (0975 – 8887) Volume 180*, 1-3.

Bourgeois, D. B. (2014, February). *Information Systems for Business and Beyound*. Retrieved from Pressbooks: https://bus206.pressbooks.com/chapter/chapter-1/

Girish. (2021, March 20). *SDLC – Software Prototype Model.* Retrieved from Senior Sir: https://seniorsir.com/sdlc-software-prototype-model/

Grand Valley State University . (2015, December). *Online Food Order System for Restaurants* . Retrieved from Grand Valley State University : https://scholarworks.gvsu.edu/cgi/viewcontent.cgi?article=1222&context=cistechlib

Guru99. (2020). *SDLC: Phases & Models of Software Development Life Cycle*. Retrieved from Guru99: https://www.guru99.com/software-development-life-cycle-tutorial.html

Haridwaj, P. (2020, August). *What is an ordering system?* Retrieved from Quora: https://www.quora.com/What-is-an-ordering-system

Hyderabad, I. (2019, April 18). *What is Ordering System*. Retrieved from Nibble Matrix: http://www.nibblematrix.com/what-is-ordering-system/

Javatpoint. (2007). *Big Bang Model.* Retrieved from Javatpoint: https://www.javatpoint.com/software-engineering-big-bang-model

Jevtic, G. (2019, May 15). *What is SDLC? Phases of Software Development, Models, & Best Practices*. Retrieved from PhoenixNAP: https://phoenixnap.com/blog/software-development-life-cycle

Parvatikar, S. S. (2020). Online Food Ordering Management System. *International Research Journal of Engineering and Technology (IRJET) Volume: 07 Issue: 11*, 1-4.

Piccolo, G., & Pigni, F. (2018). Information systems for managers: with cases. *Prospect Press*, 28.

Pomelo Pay. (2020, September 4). *Online Ordering Systems: What Are The Advantages of Using One?* Retrieved from Pomelo Pay: https://www.pomelopay.com/blog/advantages-online-ordering-system

Techopedia. (2020). *Information System (IS).* Retrieved from Techopedia: https://www.techopedia.com/definition/24142/information-system-is

The Economic Times. (2021, August 5). *Definition of Rapid Application Development* . Retrieved from The Economic Times: https://economictimes.indiatimes.com/definition/rapid-application-development

Tutorialspoint. (2021). *SDLC - Agile Model.* Retrieved from Tutorialspoint: https://www.tutorialspoint.com/sdlc/sdlc_agile_model.htm

Tutorialspoint. (2021). *SDLC - Big Bang Model.* Retrieved from Tutorialspoint Simple Easy Learning: https://www.tutorialspoint.com/sdlc/sdlc_bigbang_model.htm

University of Houston. (2000, April 07). *HISTORY OF INFORMATION SYSTEMS* . Retrieved from University of Houston: https://uh.edu/~mrana/try.htm

Varsha, P. S. (2015). Implementing Customizable Online Food Ordering System Using Web Based Application. *International Journal of Innovative Science,*

Engineering & Technology, 1-8. Retrieved from IJISET: http://ijiset.com/vol2/v2s4/IJISET_V2_I4_112.pdf

Vidyagiri, B. K. (2020). Online Food Ordering Management System. *International Research Journal of Engineering and Technology (IRJET)*, 409.

Visual Paradigm. (2002). *What is Unified Modeling Language (UML)*. Retrieved from Visual Paaradigm: https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-uml/

Wikipedia. (2021, June 6). *Information system*. Retrieved from Wikipedia: https://en.wikipedia.org/wiki/Information_system

Zwass, V. (2011, December 27). *information retrieval system, information storage and retrieval system, information storage system*. Retrieved from Britannica: https://www.britannica.com/topic/information-system