**KWAME NKRUMAH UNIVERSITY OF SCIENCE AND TECHNOLOGY**

**FACULTY OF PHYSICAL AND COMPUTATIONAL SCIENCE**

**DEPARTMENT OF COMPUTER SCIENCE FINAL YEAR PROJECT**

****

**PROJECT TITLE:**

SMART BANK QUEUE MANAGEMENT SYSTEM

**SUBMITTED BY:**EMMANUEL RICHMOND ADOTEY

KWADJO ADJEI DUAH

**SUPERVISED BY:**

DR. FRIMPONG TWUM

JULY 2023

**DEDICATION**

We would like to dedicate this work first to the Lord for making it possible for us to reach this far in our academic journey and also to our parents for their support to see us through our education.

i

**DECLARATION BY STUDENT**

We declare, that we undertook this project known as “**SMART BANK QUEUE MANAGEMENT SYSTEM”** on **KNUST** campus, herein submitted under supervision.

Signed: Date:

……………………………….. ………………………………………..

KWADJO ADJEI DUAH

(9403319)

Signed: Date:

……………………………… ………………………………………..

Emmanuel Richmond Adotey

(9392019)

ii

# DECLARATION BY SUPERVISOR

I declare that I have personally supervised these students in undertaking the study report herein and I confirm that these students have my permission to present it for assessment.

Signed: Date:

……………………………. ………………………………..

DR. FRIMPONG TWUM

iii

**ACKNOWLEDGEMENT**

Our immense gratitude goes to our supervisor, Dr. Frimpong Twum for his guidance and insights as he took us through Software development life cycle as a course which has guided us to build a good system. We would also like to say a big thank you to our mates who gave suggestions for this project and helped in system testing.

iv

**ABSTRACT**

With the increase in the demand for banking services, most banks in Ghana tend to find it difficult attending to the services of their numerous customers in an organized fashion at their banking premises .This mostly brings about congestion especially with the long queue formation causing some customers to wait outside the banking halls and others who anticipate a long period of waiting before their turns to vacate the banking premises. Services like customer enquiries can be automated as a way to reduce the traffic at the banking halls and also with the help of queuing models, systems can be developed to help customers join bank queues virtually and have a live update of their waiting time and also position in the queue in with the aid of devices such as mobile phones and laptops provided they have internet connection.

The main purpose of this project is to develop a web application system that makes use of a suitable queuing model and an AI model to streamline the banking services offered by the tellers(A.K.A front desk wokers) and also a chatbot to handle the frequent customer enquiries to reduce the number of people who visit the banking premise for their service, there is also a report generation feature and system analytics feature.

V

**TABLE OF CONTENTS**

DEDICATION………………………………………………………………………………………….i

DECLARATION BY STUDENT………………………………………………………………….ii

DECLARATION BY SUPERVISOR…………………………………………………………….iii

ACKNOWLEDGEMENT………………………………………………………………………….iv

ABSTRACT…………………………………………………………………………………………….v

TABLE OF CONTENTS………………………………………………………………………….vi

LIST OF FIGURES…………………………………………………………………………………ix

LIST OF TABLES……………………………………………………………………………………x

CHAPTER ONE…………………………………………………………………………………….1

INTRODUCTION………………………………………………………………………………..1

1. Problem Statement…………………………………………………………….......

vi

**LIST OF FIGURES**

Figure 0.1-

Figure 0.2-

**LIST OF TABLES**

Table 1-

Table 2-

# CHAPTER ONE

# INTRODUCTION

**1.0 Overview**

Over the years, the style of banking has changed. Much of the transactions occur online but physical branches are still important for providing personalized service for complex products and services such as large amounts of deposits or withdrawal , foreign exchange services and others.

Time is a crucial factor and hence customers expect a time efficient and smooth experience when using banking systems. However, many branches still rely on inefficient methods of managing customer queues such as manual calling and allowing customers to join physical queues to wait for their turn for service. These methods tend to bring about poor service quality and low customer satisfaction especially when customers aren’t able to have a good estimate of their waiting time in the queue when a lot of people are present.

As the situation stands majority if not all banks in Ghana haven’t evolved to meet such customer expectation. I believe banking firms have to prioritize the waiting experience of customers as the increase in banking firms makes it easy to switch if one feels the nature of services offered doesn’t suit his or her preference.

However, achieving instant service in the conventional way is costly as it needs too many staff members. So another issue is; how can you boost efficiency and service quality without raising the cost excessively?

This is where smart bank queue management web application system comes into play which creates a virtual queue where customers can join with the aid of a mobile phone ,tablet or a personal computer connected to the internet.

This is a system with a great advantage in that customers can see in real time their position in the queue and also their estimated waiting time.

This system makes use of multi-server queuing model with first in first out queue discipline (how customers are served)concept to structure the bank queue by efficiently assigning incoming customers to the available tellers with the help of random forest predictive artificial intelligence model, in addition to this is a chatbot that can answer common questions, provide information, and offer guidance to customers via text .The system also has an analytics display and a report generation feature downloadable in a pdf format.

**Multi-Server Queuing Model**

A multi-server queuing model is a type of queuing model that involves more than one server that can provide service to the customers who arrive and join a queue. The servers can be identical or different in terms of their service rates, capacities, or types. The customers can also be homogeneous or heterogeneous in terms of their arrival rates, service demands, or preferences. A multi-server queuing model can be used to analyze and optimize the performance of various systems that involve multiple service providers, such as banks, restaurants, airports, hospitals, call centers, and so on.

Some of the characteristics and parameters of a multi-server queuing model are:

* The number of servers and their availability.
* The arrival process of the customers (usually assumed to be a Poisson process with rate λ)
* The service time distribution of the servers (usually assumed to be an exponential distribution with rate μ)
* The queue discipline or the rule that determines which customer is served next (usually assumed to be first-come-first-served or FCFS)
* The queue capacity or the maximum number of customers that can wait in the queue (usually assumed to be infinite or limited by a certain number)
* The system capacity or the maximum number of customers that can be in the system (usually assumed to be infinite or equal to the queue capacity plus the number of servers)

Some of the performance measures of a multi-server queuing model are:

* The average number of customers in the queue (Lq) and in the system (L)
* The average waiting time in the queue (Wq) and in the system (W)
* The probability that the queue is empty (P0) or full (Pn)
* The probability that a customer has to wait in the queue (Pw) or is served immediately (1 - Pw)
* The server utilization or the fraction of time that each server is busy (ρ)

There are different methods and formulas to calculate these measures depending on the specific assumptions and parameters of the model. One of the most common and simple multi-server queuing models is the M/M/c model, which assumes that:

* The arrival process is a Poisson process with rate λ where poisson distribution refers to random distribution which is independent of any previous event.
* The service time distribution is an exponential distribution with rate μ
* There are c identical servers with infinite capacity
* The queue discipline is First Come First Serve
* The queue capacity is finite(30)

**Random Forest Predictive A.I model**

Predictive AI models are artificial intelligence models that use data mining, machine learning, and statistical techniques to analyze historical and current data and make predictions about future events or outcomes. A random forest predictive AI model is a type of machine learning model that uses an ensemble of decision trees to make predictions for classification or regression problems. A decision tree is a simple model that splits the data into branches based on certain rules or criteria, and assigns a class label or a numerical value to each leaf node. A random forest combines multiple decision trees, each trained on a different subset of the data or features, and aggregates their predictions using a voting or averaging scheme.

Some of the advantages of random forest predictive AI models are:

* They can handle large and complex datasets with high dimensionality and heterogeneity.
* They can provide estimates of feature importance and variable selection.

This would be used to assign customers to the tellers.

**1.2 Problem Statement**

Majority of banks in Ghana in the likes of Access Bank Ayeduase branch and fidelity bank knust branch experience inefficiencies in queue management mostly during peak periods resulting in congestion and frustration of customers especially when they have no precise estimate of how long they would have to wait for their turn to be served. Long wait times and confusion during peak hours have led to customers waiting outside the banking hall, causing disorganization and instances of line-cutting. Furthermore , some customers choose to leave and return later due to extended wait times, leading to missed revenue opportunities for the bank such as when the bank isn’t aware of the customers that leave the queue without being attended to.

The existing systems fails to meet the above stated problems and hence the need for this online queue management system to create a virtual queue to streamline operations and enhance the overall efficiency of the banking services.

**Aim of the project**

The aim of this project is to develop a bank queue management system that allows customers to join the bank queue virtually for various front desk banking services and incorporate an automated chatbot to handle customer inquiries .The system will utilize a multi- server queue model to efficiently manage the activities of four tellers and minimize customer waiting periods. Additionally, random forest predictive AI model will be implemented to determine the best queue for incoming customers which would improve operational efficiency.

**Specific Objectives**

1. To build a database to manage the various data involved in the operations of the queuing system.

2. To build an automated chatbot to handle the user enquiries in order to avoid users from visiting the bank for assistance that can be easily handled.

3. To make use of Predictive AI model to help the system determine the best line of queue for incoming customers.

4.To make use of multi- server queue model with First In First Out queue discipline to handle the system of multiple servers(Four tellers in our case) to improve service time, reduce waiting times for customers , also making them aware of their estimated waiting time and optimizing staffing level.

5. To develop a web client for both mobile and desktop devices to enable customers join the queue at their comfort with the help of these devices.

6. To provide a space for user feedback about their experience.

7.To provide an analytics feature for the system .

8.To provide a report generation feature which can be downloaded

9. To provide a functionality to enable the bank know the customers who left the queue without been attended to.

10. To test the system to ensure its functionality and reliability before its implementation.

**Project Justification**

The development of an online queue management system for banks is crucial for several reasons.

Firstly, it will enhance the customer experience by reducing waiting times, allowing customers to know their queue position, increasing transparency, and providing better communication between the customers and the bank's staff. By having a system that allows customers to join a virtual queue and receive real-time updates on their queue status, customers will no longer have to wait in long queues, leading to reduced frustration and an overall better banking experience.

Secondly, an online queue management system for banks will also increase the bank's efficiency. By automating the queue management process, the bank can streamline its operations, optimize staff allocation, and improve customer flow. This system will also allow the bank to collect data on customer behavior, preferences, and feedback, leading to more informed decision-making processes and a better understanding of their customer base.

Thirdly the banks can know of their lost sales for the day due to the system providing them with information on the customers who left the queue after joining without being attended to.

**Project Motivation**

The banking sector is a crucial aspect of the global economy and is essential to people's daily lives. However, with the increasing population and technological advancements, banks have experienced a significant increase in customer traffic, leading to long queues and wait times. This causes conflicts to emerge during busy hours within the banks as many people are in a haste to do activities in their daily lives.

However this activities of bank tellers can be made semi-automated with the mindset of efficiently handling activities that can be replicated or done by a machine directly.

**Project Scope**

-The scope of this project covers the front desk bank activities of the customer care persons and tellers which are deposit, withdrawal, foreign exchange and bill payment .Customer enquiries on the other hand is handled by the chatbot except critical instances whereby customers are made to call customer care.

However the system has restrictions such as:

-The system wouldn’t handle bulk cash transaction since it is assigned a special service and hence customers would have to visit the branch for that.

-Technology Restrictions: The infrastructure and underlying technology both play a significant role in how well the queue management system works. The queue procedure may be disrupted and the client experience may be impacted if the system encounters technical problems or downtime.

-Internet connectivity: A reliable internet connection is essential for virtual queue management systems. Customers may have trouble entering the virtual queue or getting real-time updates in locations with spotty or slow internet connectivity.

-User Acceptance: Virtual queuing systems might not be something that all customers are familiar with or find comfortable. Some clients, particularly older people, might prefer traditional physical lines or could find it difficult to use the web client or communicate with the chatbot.

- Chatbot Limitations**:** Although chatbots can handle routine inquiries, complex or personalized customer queries may require human intervention. Chatbots may not always provide satisfactory responses, leading to customer frustration.

**Project limitations**

The following are the exisiting limitations :

1.Time : there wasn’t adequate time to consider and test many of the machine learning models which exist which are meant for prediction

2.Tools : Rust is a relatively new language with most of its third party packages being unstable and hence was delayed by the trial of packages to suit the system being developed.

3.Limited knowledge in server side events and web socket operation caused a week delay to figure it out and to finally determine that server side events was the best fit for our system use case.

4.Data expense: The data expense was such a headache with MTN offers and the campus wifi was so unstable that some results could not be processed.

**Beneficiaries of the Project**

This project is made generally to fit for implementation at any Ghanaian banks in the likes of fidelity bank knust branch and access bank adum/ayeduase branch . This is due to the system and style of banking in most ghanaian banks which provided the format for designing the smart bank queue system.

**Academic and practical relevance of the project**

1.Knowledge Advancement: The project involves the application of a number of technologies and approaches, including database management, multi-server queue models, predictive AI models, and automated chatbots. The project advances knowledge in the areas of computer science, artificial intelligence, data management, and queue theory by fusing these components.

2.Research Contributions: The research includes completing a thorough analysis of the advantages and disadvantages of the current bank queuing systems as well as the difficulties that banks encounter during peak times. This study adds to the body of knowledge in queuing theory and aids in the discovery of workable solutions to frequent queuing issues in the banking industry.

3. Innovative Approaches: The proposed use of multi-server queue models and predictive AI models represents novel methods to enhance consumer queuing experiences. Successful application of these models may offer perceptions and approaches that can be expanded upon and examined further in scholarly research.

Practical Relevance:

1.Customer Experience: The creation of a virtual queue management system, an automated chatbot, and waiting time alerts can greatly enhance the general customer experience. Customers' frustration levels are decreased and their happiness is increased since they can join queues remotely, receive prompt answers to their questions, and better understand how long they will be waiting.

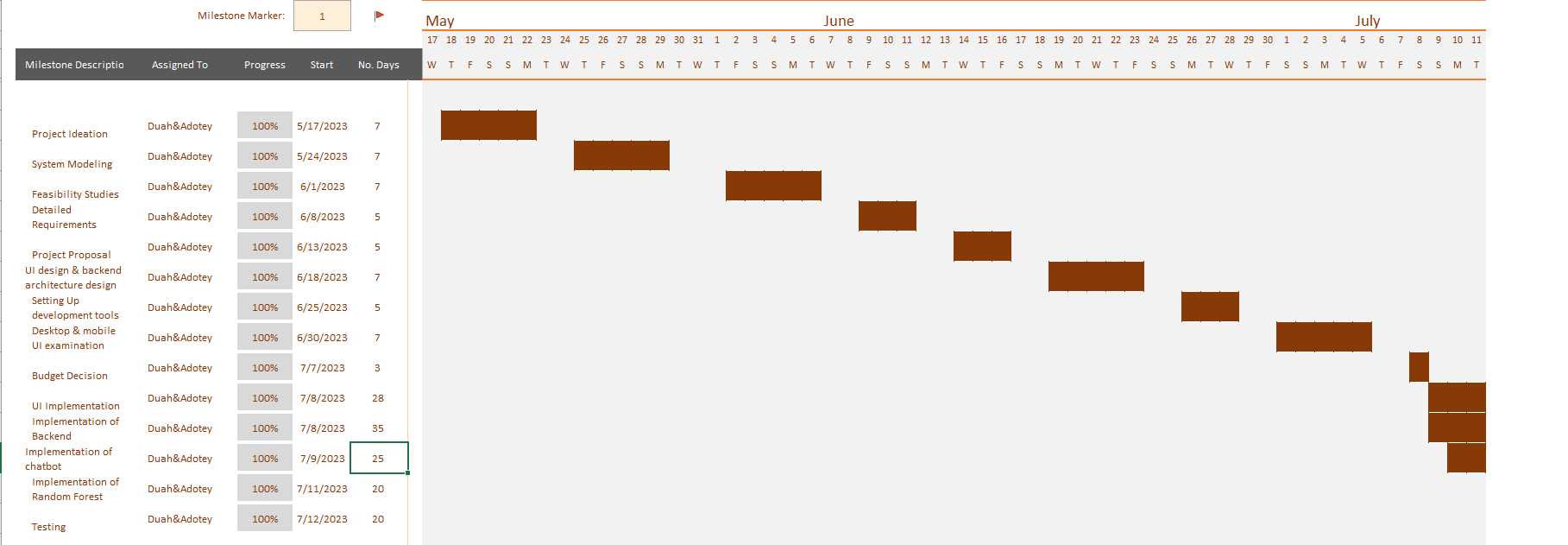
2. Operational Efficiency: By combining a multi-server queue model with FIFO discipline, it is possible to improve resource allocation and shorten client wait times. The system's capacity to choose the ideal queue for incoming clients further boosts operational effectiveness and guarantees top-notch service.

3. Cost savings: The bank may be able to cut labor expenses and better utilize resources by giving consumers a virtual queueing alternative and minimizing the need for in-person help through the chatbot.

4. Data-Driven Decision Making: The bank is able to get insightful data on client behavior, preferences, and feedback thanks to the project's deployment of a database to manage queue data. This information can help data-driven decision-making techniques make better service planning and overall operations decisions.

5. Competitive Advantage: By successfully implementing a productive and user-friendly queuing system, the bank may be able to gain a competitive edge in the market. More consumers can be attracted and kept through improving customer experiences and streamlining operations.

**Project activity planning and schedules**

****

*Figure 0.1- Gantt Chart of project planning and schedule*

**Structure of report**

**Abstract**

* A brief summary of the project's objectives, methodologies, and key findings.

Table of Contents

1. Introduction
   * Background and Problem Statement
   * Aim and Objectives
   * Scope of the Project
   * Project Justification and Motivation
2. Literature Review
   * Overview of Queue Management Systems
   * Customer Experience and Queuing Behavior Studies
   * Existing Queue Management Technologies
   * Automated Chatbots in Customer Service
   * Predictive AI Models for Queue Optimization
   * Case Studies and Implementation Examples
   * Customer Feedback and User Experience Studies
   * Critical Review and Gaps in Existing Literature
3. Methodology
   * Research Design and Approach
   * Data Collection Methods
   * Technology Stack and Tools
   * System Architecture
   * Implementation Details
   * Testing and Validation Procedures
4. System Design and Architecture
   * Overview of the Bank Queue Management System
   * Functional Requirements
   * Data Flow Diagrams
   * Use Case Diagrams
   * Class Diagrams
   * Sequence Diagrams
5. System Implementation
   * Database Design and Management
   * Chatbot Development and Integration
   * Predictive AI Model Implementation
   * Web Client Development
   * Multi-Server Queue Model Implementation
6. Results and Discussion
   * Evaluation of the Bank Queue Management System
   * User Feedback and Satisfaction
   * System Performance and Efficiency
   * Comparison with Existing Systems (if applicable)
7. Conclusion
   * Summary of Achievements
   * Project Outcomes and Contributions
   * Lessons Learned
   * Future Recommendations
8. References
   * Citations of relevant literature and resources used in the project.
9. Appendices
   * User Manuals and Guides
   * Screenshots or User Interface Mockups
   * Data Samples (if applicable)
   * Any other relevant supplementary information

**Deliverables**

1. A comprehensive review on existing bank queuing systems with chatbot.

2. Chatbot to address frequently asked questions and basic routine requests users make.

3. A user interface for the system that allows users to interact with the system seamlessly.

4. A database to handle the data involved in the operation of the system.

5. A documentation and user manual.

6.A report generation in pdf format of the system analysis.

7. A Web client responsive on both mobile and desktop.

8.A system analytics display page.

**Definition and explanation of terms**

Your proposed system:

Advantages:

* Your proposed system uses a multi-server queue model and predictive AI to optimize the customer flow and waiting time in the bank.
* Your proposed system uses a chatbot to interact with customers and provide them with information, guidance, and feedback on their service needs.
* Your proposed system can adapt to changing customer behavior and preferences, as well as different service scenarios and requirements.

Disadvantages:

* Your proposed system may be difficult to implement and maintain, as it requires advanced technology and expertise in AI and chatbots.
* Your proposed system may face resistance from customers or employees who are not familiar or comfortable with using AI or chatbots.
* Your proposed system may pose ethical or legal challenges, such as data privacy, security, transparency, accountability, or bias in AI or chatbots.

**CHAPTER TWO**

**REVIEW OF SIMILAR SYSTEMS**

The existing bank queue systems are of different types but they have some features in common, such as:

1. Self-service kiosks: These are devices that allow customers to check-in, select the service they need, and receive a ticket with a queue number. Some kiosks can also serve as cash-deposit devices to carry out standard transactions like balance inquiries, fund transfers or check deposit.

2. Digital signage: These are screens that show the queue status, such as the current number being served, the anticipated waiting time, and the service counter number. They can also show promotional messages, news, or entertainment content to keep customers engaged while they wait.

3. The interface for service providers: This is the device that allows service providers to communicate with the bank's line-up system. They can use a tablet, computer, or wireless device to monitor their performance, call the following client, divert clients to other services, and view client information.

4. Central processing unit: This is the core of the bank queue system that manages the queue logic, assigns customers to service providers, collects data and analytics, and integrates with other systems such as CRM or ERP.

Existing systems in this field are mainly customized and closed source hence their exact details aren’t available but just a some outline. Below are providers of such systems and their features.

1. Qmatic: Qmatic is a market pioneer for banking and other industries when it comes to queue management systems. They provide a complete solution that addresses every stage of the client journey, from booking to post-serving. Additionally, they offer sophisticated analytics and perceptions to assist banks in improving their operations and client happiness.

2. Verint is an organization that offers retail choreography services to banks and other businesses. They provide a virtual queuing system that does away with actual lines and lets clients wait wherever they like. Along with managing appointment, click-and-collect, and event customers, they also offer a comprehensive suite of solutions for walk-in customers.

3. Wavetec: Wavetec is a multinational corporation that offers queue management technologies to the banking industry and other industries. They provide a modular solution that may be altered to fit the demands and preferences of the bank. Additionally, they have a smartphone app that clients can use to schedule appointments, enter virtual lines, and access online banking services.

Qmatic:

Advantages:

1. Qmatic offers a comprehensive solution that covers the entire customer journey from pre-arrival to post-serving and integrates with various channels and platforms.

2. Qmatic has advanced analytics and insights that help banks optimize their operations, customer satisfaction and loyalty.

Disadvantages:

1. Qmatic is more expensive than other solutions, as it requires hardware, software, installation, training, and support.

2. Qmatic may not be able to leverage the full potential of AI and chatbot as it mainly relies on traditional methods of queue management.

Wavetec:

Advantages:

1. Wavetec offers a modular solution that can be customized according to the bank’s needs and preferences, with features such as self-service kiosks, digital signage, mobile app, central processing unit, and service providers’ interface.

2. Wavetec has a mobile app that enables customers to book appointments, join virtual queues, and access digital banking services, enhancing customer convenience and engagement.

Disadvantages:

1. Wavetec may not be able to handle high volumes of customers or peak hours efficiently, as it uses a single queue for multiple servers.

2. Wavetec may not be able to provide personalized or proactive service to customers, as it does not use AI or chatbots in its solution.

3. Wavetec may not be able to collect or analyze data effectively, as it does not have a robust analytics or reporting system.

Verint:

Advantages:

1.Verint offers a virtual queuing system that eliminates physical queues and allows customers to wait anywhere they want, using their mobile devices or online platforms[3](https://www.ijser.org/researchpaper/ANALYSIS-OF-MULTIPLE-QUEUE-MULTIPLE-SERVER-QUEUING-SYSTEM-A-CASE-STUDY-OF-FIRST-BANK-NIG-PLC-AFIKPO-BRANCH.pdf).

2. Verint has a full suite of solutions to manage walk-in customers alongside appointment, click and collect, and event customers, creating a seamless channel experience.

Disadvantages:

1. Verint may not be suitable for customers who prefer physical queues or face-to-face interactions, as it relies heavily on digital channels and self-service options

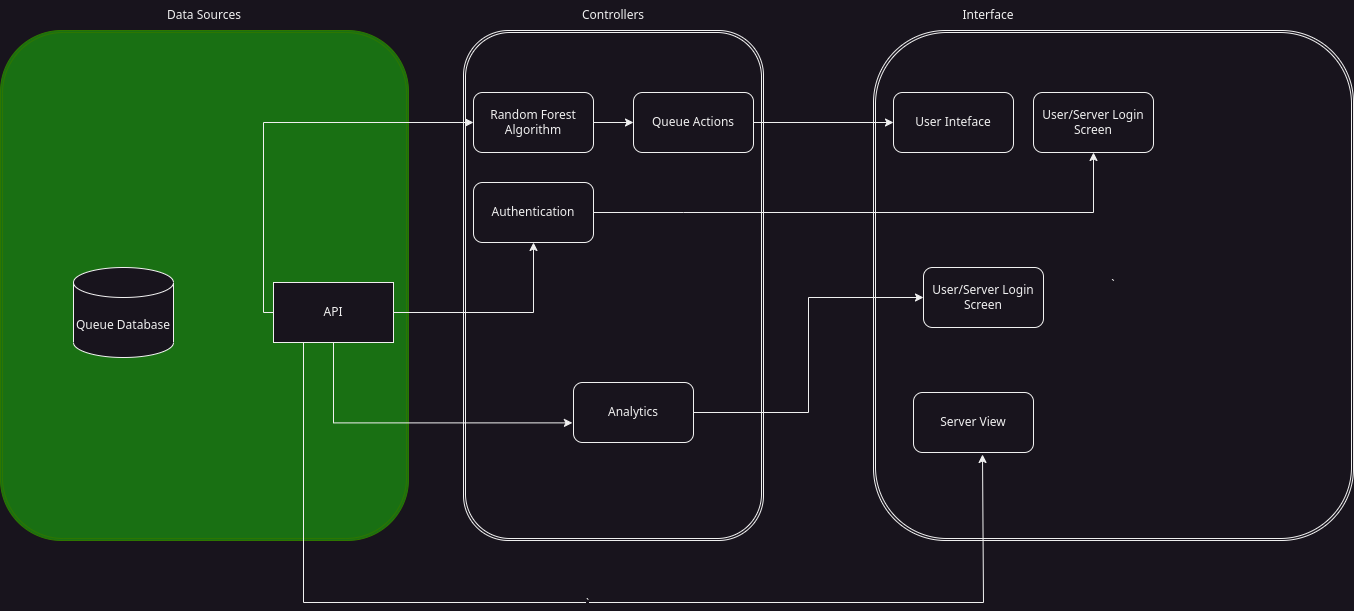
2. Verint may not be compatible with some existing systems or processes in the bank, as it requires a high level of integration and customization.

**The Proposed System**

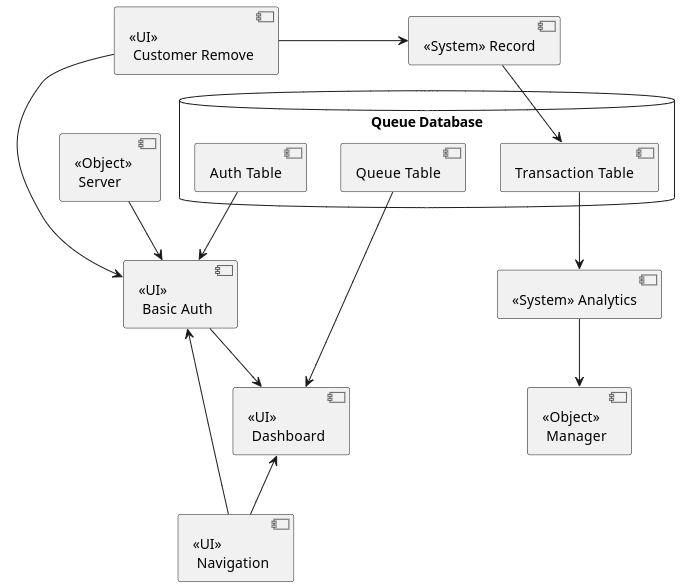
The proposed system makes use of multi-server queue model with first in first out queue discipline and predictive AI to optimize the customer flow and waiting time in the bank.

The proposed system uses a chatbot to interact with customers and provide them with information, guidance, and feedback on their service needs.

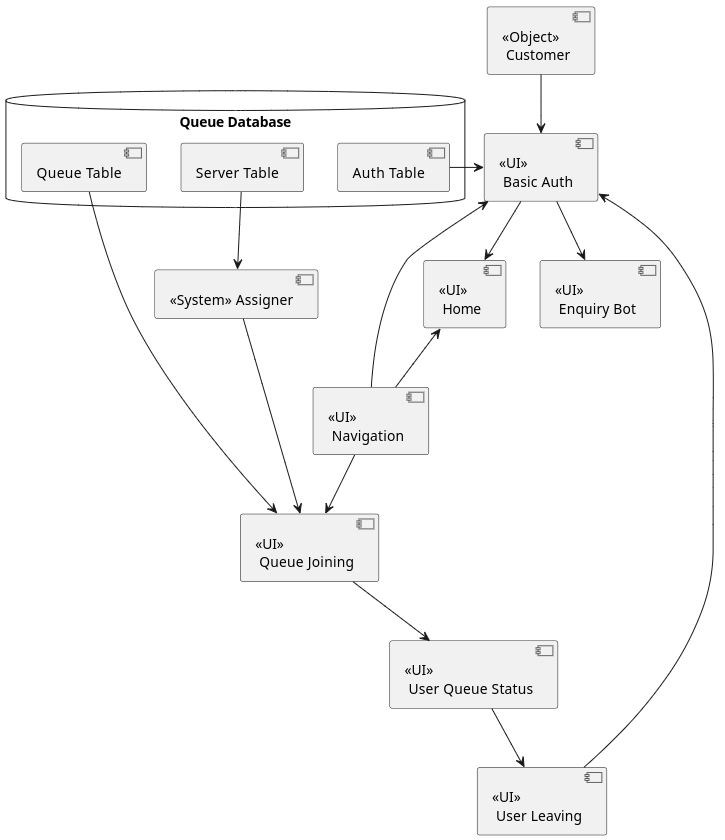
**Architecture of the proposed system**

****

*Figure 0.2 – System Architecture*

**Components Designs and Component description** *Figure 0.3 – Components design for Server*

* **Queue Table:** Contains information on the bank queue details such as position of the customer, estimated waiting time ,teller assigned and others.
* **Dashboard:** The dashboard displays the Tellers workspace where each teller can view the customers that have been assigned to then and perform various operations with that table such as complete transactions on a button click or cancel a transaction .
* **Basic Auth:** This refers to the authentication done by the system to verify customers before they are given access by requiring the user account number and also user password.
* **Analytics:** Displays the system’s data analysis with graphical display aided by tools such as pie chart and line graph. The results from the queuing model relating to the entire systems operation is displayed on this page .
* **Server:**System responsible for accepting requests and giving responses in the form of resources by the system’s client.
* **Transaction Table:** Contains details of the transactions that have occurred be it the failed ones and also the completed transactions

****

*Figure 0.4 – Components design for Customer view*

* **Queue table**: This is the database table that stores the information of the clients who join the queue . It contains fields such as client ID, name, phone number, teller ID, estimated waiting time, and queue status.
* **Server table**: This is the database table that stores the information of the tellers who serve the clients. It contains fields such as teller ID, name, phone number, availability, and assigned clients.
* **Auth table**: This is the database table that stores the authentication information of the clients and tellers who use the system. It contains fields such as user ID, password, role, and session ID.
* **System assigner**: This is the component that assigns a teller to a client when they join the queue. It also updates the estimated waiting time and queue status of each client in the queue table.
* **UI home**: This is the user interface component that displays the home page of the system. It allows the users log in or sign up using their credentials.
* **UI bot**: This is the user interface component that displays a chatbot that interacts with the users. It provides information and guidance on common customer questions.
* **UI navigation**: This is the user interface component that displays a navigation bar that allows the users to access different features of the system. It also shows the user’s name and role.
* **UI queue joining**: This is the user interface component that allows the clients to join the queue virtually. It asks for their name and phone number and shows them their assigned teller and estimated waiting time.
* **UI user queue status**: This is the user interface component that shows the clients their current queue status. It displays their position in the queue, their tellers and availability, and their estimated waiting time. It also allows them to leave the queue if they want to.
* **UI user leaving**: This is the user interface component that confirms the clients’ decision to leave the queue. It asks them for feedback and thanks them for using the system.
* **UI teller dashboard**: This is the user interface component that shows the tellers their assigned clients and their queue status. It allows them to mark a transaction as completed or cancelled and update their availability.

**Proposed System/Software Features**

**1.**  **A web client for both mobile and desktop:** A desktop and web client to enable customers join the queue at their comfort with the help of these devices of which would be available to the user.

**2. Join & leave bank queue :** The proposed system allows the user an interface to join the bank queue after filling the required information on a button click and also to leave the queue when he or she wants to.

**3. Estimated Customer waiting time & Teller assigned:** The proposed system provides the customer with information concerning the teller he or she has been assigned to and a real time details on the customers estimated waiting time.

**4.Chatbot:** The proposed system offers the customer a chatbot to handle the various frequent customer enquiries to deter them from going to the banking premises to seek help which could easily be handled by the bot.

**5. Predictive AI model:**  The proposed system makes use of random forest predictive artificial intelligence model to determine the best line of queue(the best teller to be assigned) for the incoming customer.

**6. Multi-server queue model :** The proposed system makes use of a multi-server queue model with first come first serve queue discipline to manage the queue and provide the system with details such as the average waiting time, average arrival rate, average service rate of the system and certain probabilities for the system analytics.

**7**. **An analytics display feature :** The proposed system makes use of graphical representation of the system analysis mainly pertaining to the teller activities to be used by the manager to make certain decisions.

**8**.**Report generation:**The proposed system provides a pdf downloadable report of the queue management operation details according to specified periods such as daily,weekly ,monthly and yearly which can be downloaded on a button click.

**Development tools and environment**

**1.** Visual Studio Code: VS Code is a desktop source code editor that runs on Windows, macOS, and Linux. It is compact but effective. It contains support for JavaScript, TypeScript, and NodeJS built in, as well as a robust ecosystem of extensions for additional languages and runtimes (including C++, C#, Java, Python, PHP, Go, and.NET). (Documentation for Visual Studio Code, n.d.)

**2. ReactJS**: React is a free and open-source front-end JavaScript library for creating user interfaces based on UI components. It is commonly known as React.js or ReactJS. It is kept up-to-date by Meta (previously Facebook) and a group of independent programmers and businesses (React (JavaScript Library) - Wikipedia, 2013). With frameworks like Next.js, React can be the foundation for single-page, mobile, or server-rendered applications. Making React apps typically 23 necessitates the use of extra libraries for routing and specific client-side functionality because React is only concerned with state management and presenting that information to the DOM.

**3. Rust:** Rust is a multi-paradigm, general-purpose programming language that emphasizes performance, type safety, and concurrency. [It enforces memory safety without requiring the use of a garbage collector or reference counting present in other memory-safe languages](https://www.bing.com/search?form=NTPCHB&q=Bing+AI&showconv=1). Rust is supported by a rich ecosystem of tools, libraries, and frameworks, such as Cargo, rustdoc , rustfmt , and many more. [Rust is widely used for both system programming and web development, especially for performance-critical applications](https://www.bing.com/search?form=NTPCHB&q=Bing+AI&showconv=1) and hence was selected in the development of this system.

**4. Actix web:** Actix Web is a web framework for Rust that is based on the actor model. It is powerful and extremely fast. It supports HTTP/1.x and HTTP/2, streaming and pipelining, WebSocket, content compression and decompression, SSL, middleware, and more. It also integrates with the AWS HTTP client. Actix Web has a type-safe and flexible API that allows you to write concise and expressive code. It also has excellent documentation and is one of the fastest web frameworks available according to TechEmpower Framework Benchmark. We considered this framework due to the extensive use of mathematical functions for the queue management which require accuracy and speed when in use by a lot of clients who would be viewing frequent real time updates related to the queue.

**5.MySQL:** MySQL is a cross-platform relational database management system that is open source. MySQL is based on the Structured Query Language, which is the popular language for accessing and managing the records in the database. MySQL is supported by Oracle Corporation. It is widely used for both small and large applications, especially for web development. MySQL is licensed under the GNU General Public License (GPL), which is considered free by most distributions (MySQL - Wikipedia, 2021).

**6.Random forest predictive AI model:** In the course of developing the system, we selected this Machine Learning Algorithm based on accuracy and runtime speed to be able to set the best server that would be able to serve the user

* The models used were all based on prediction and used the time taken by each server to complete the transaction
* The best ML models under review based on these criteria were
  + KMean
    - A popular and fast algorithm, K-Means groups data points by similarities and so is often used for the clustering model. It can quickly render things like personalized retail offers to individuals within a huge group, such as a million or more customers with a similar liking of lined red wool coats
  + Random Forest
    - This algorithm is derived from a combination of decision trees, none of which are related, and can use both classification and regression to classify vast amounts of data
  + Gradient Boosting
    - This algorithm uses several combined decision trees which are related. It builds out one tree at a time, thus enabling the next tree to correct flaws in the previous tree. It is normally used in rankings
* The test was done using the scikit-learn python package and using the iris dataset provided by the package
* The script used is defined below

Test Script

{

from sklearn import datasets  
from sklearn.model\_selection import cross\_val\_score  
from sklearn.linear\_model import LinearRegression  
from sklearn.cluster import KMeans  
from sklearn.ensemble import RandomForestClassifier, GradientBoostingClassifier  
from time import process\_time  
  
# Load the iris dataset  
iris = datasets.load\_iris()  
X = iris.data  
y = iris.target  
  
# Define the models  
models = [  
 ('RFC', RandomForestClassifier()),  
 ('GBC', GradientBoostingClassifier()),  
 ('KM', KMeans(n\_init=10))  
]  
  
# Compare models  
for name, model in models:  
 start = process\_time()  
 scores = cross\_val\_score(model, X, y, cv=5, scoring='accuracy')  
 print(f'{name}(Accuracy): {scores.mean():.2f} (+/- {scores.std() \* 2:.2f})')  
 end = process\_time()  
 print(f'{name}(Speed): {end - start}s')

}

Test Results

| Model Name | Speed(Seconds) | Accuracy |
| --- | --- | --- |
| KMeans | 0.5907096660000004 | 1% |
| Random Forest | 0.5669434889999998 | 97% |
| Gradient Boosting | 1.0802188810000004 | 96% |

According to the test the random forest classifier was the fastest and most accurate

**7. OpenAI API for Chatbot** : The OpenAI API is a powerful platform that allows you to access and use a variety of artificial intelligence models for different tasks and domains. You can generate text, code, images, or speech with the OpenAI API, as well as fine-tune your own custom models for your specific needs. The OpenAI API is easy to use and integrate with any language or framework, and it offers a range of models with different capabilities and price points.This was used for the chatbot with the provided high accuracy due to the time constraint which would have been required to train a different model to suit the bank enquiries.

**8. React Chatbot Kit:** React Chatbot Kit is a library that helps you create chatbots with React. It provides an easy way to get started with building chatbots that can handle different types of messages, such as text, images, buttons, and options. React Chatbot Kit also offers a flexible and customizable API that allows you to define your own logic and actions for your chatbot.

**9.React Charts:** This tool was used for the visual display of the analytics pane for use cases such as pie chart, line graph etc.

**Benefits of implementation of the proposed system**

The benefits of the proposed system are outlines below:

**1.**The online bank queue system enables the bank to know the customers that left the queue without been served with the provision of a leave queue button provided on the customers end in case they want to leave the queue due to various reasons which the bank can enquire from the customers to improve their services etc.The Tellers screen also provides a cancel functionality incase the expected customer isn’t present for some period of time after joining the queue and hence can be recorded as customers who left the queue without being served.

**2.** The provision of the chat bot reduces the traffic at the banking premises by handling frequently occurring or asked questions by customers virtually. Some routine questions such as how to open an account, what documents would be required and others can be handled easily by the bot and hence customers don’t have to drive all the way to the banking premises to make such enquiries .

**3.**The proposed system enables the customers to view their estimated waiting time to be served and also make them aware of their position in the queue not leaving out the teller they have been assigned to. This information is vital to each customer to keep them waiting in the queue rather then they not knowing and having to wonder how long they would have to wait before been attended to. The system gives a real time update if there has been changes to the waiting time, queue position and teller assigned.

**4.** The proposed system enables the bank view the queue information of the entire system by the aid of the queuing model .It provides information such as the system’s average service rate(of the tellers combined),average waiting time in the system(in the bank queue), probability of the system been idle or busy and other details which wouldn’t have been available in the absence of this proposed system because the existing systems don’t implement this feature to get access to such details.

**5.** The proposed system provides the bank managers with the details of each teller’s service rate which is known as “Turn Over time” in banking terms. The system tracks each tellers service rate per customer served and keeps that record which can be used by the AI model to determine the best line of queue for incoming customers. Managers can use this information to decide on how to handle tellers who aren’t consistent in their turnover time.

**6.** The proposed system provides a downloadable pdf format of the system analytics or details of the banking operation per specified period of time which can be used for reporting to management.

**Methodology**

**3.0 Overview**

The project methodology shows the different approaches and instruments that were used in the development of the proposed solution. We used the Agile development method and the incremental process as a software model given our ability to manage change. In addition, development tools such as reactJS for the frontend , rust for the backend making use of its actix web framework and react charts for the system analytics display not leaving out the chatbot with OpenAI’s API and react chat toolkit.

**Requirement Specification**

Software Requirement Specification (SRS) is a formal report, which acts as a representation of software that enables the customers to review whether it (SRS) is according to their requirements. Also, it comprises user requirements for a system as well as detailed specifications of the system requirements. (Software Engineering | Software Requirement Specifications – Javatpoint , n.d.)

**Stakeholders of the system**

The stake holders of the Smart Bank Queue System are:

* **Customers**: They are the primary users of the system who want to have a convenient and satisfying banking experience. They can join the queue online or in-person, receive notifications and updates on their queue status, and choose their preferred service and staff.
* **Bank staff**: They are the service providers who interact with the customers and handle their requests. They can use the system to check-in and check-out customers, view their assigned customers and their profiles, and update their availability and skills.
* **Bank managers**: They are the supervisors who oversee the branch performance and customer satisfaction. They can use the system to monitor the queue activity, staff productivity, customer feedback, and service quality. They can also use the system to generate reports and insights to improve the branch efficiency and effectiveness.
* **Bank executives**: They are the decision-makers who set the strategic goals and objectives for the bank. They can use the system to evaluate the branch performance, customer loyalty, and market trends. They can also use the system to compare different branches and regions and identify best practices and areas for improvement.
* **System developers**: They are the technical experts who design, develop, test, and maintain the system. They can use the system to implement new features, fix bugs, enhance security, and ensure compatibility with other systems.
* **System vendors**: They are the external partners who provide the system software, hardware, and support. They can use the system to deliver the system solution, install and configure the system, train and assist the users, and update and upgrade the system.

**Requirements Gathering Process**

* **Relevant stakeholders**: The qualified representatives for each relevant stakeholder group, depending on the system include:

1. Customers stakeholders

2. Decision-makers

3. Users

4. System administrators

5. Other impacted customer departments

6. Internal stakeholders

7. Executives

8. Marketing

9. Customer support

10. Other partners

**Goals and objectives**: The main goal of this system is to improve the customer experience and satisfaction by reducing their waiting time and providing them with personalized services. The detailed objectives of this system are:

* + To allow customers to join a virtual queue using their mobile devices or kiosks at the branch
  + To provide customers with real-time information on their queue status, estimated waiting time, and assigned staff
  + To enable customers to choose their transaction type
  + To allow customers to leave feedback on their service experience.
  + To enable bank staff to view their assigned customers
  + To enable bank managers to monitor the queue activity, staff productivity, customer feedback, and service quality
  + To allow bank managers to generate reports and insights on branch performance
  + To enable bank executives to evaluate branch performance, customer loyalty, and market trends
* **Elicit requirements from stakeholders**: The first of three overlapping, highly iterative requirements collecting subprocesses is underway now. In this process, various techniques are used to gather information from different stakeholders about their needs, expectations, preferences, constraints, assumptions, risks, etc. Some of these techniques are:
  + Interviews: One-on-one or group discussions with stakeholders to ask open-ended or closed-ended questions about their requirements
  + Surveys: Online or offline questionnaires with multiple-choice or rating-scale questions to collect quantitative or qualitative data from a large number of stakeholders
  + Focus groups: Moderated sessions with a selected group of stakeholders to explore their opinions, attitudes, perceptions, etc. about a topic or issue related to the system
  + Observation: Direct or indirect observation of stakeholders’ behaviors, actions, interactions, etc. in their natural or simulated environment
  + Prototyping: Creating mock-ups or models of the system or its components to demonstrate or test its functionality or usability with stakeholders
  + Brainstorming: Generating ideas or solutions for a problem or opportunity related to the system in a group setting
* **Document the requirements**: As soon as the requirements begin to emerge from the elicitation process, the documentation process starts in an agreed format. The requirements documentation:
  + Is available for review by all stakeholders
  + Can easily be navigated and understood by the team and stakeholders
  + Provides a facility for traceability to other documentation Some of the formats that can be used for documenting requirements are:
  + User stories: Short descriptions of what a user wants to do or achieve with the system in a specific format such as “As a <role>, I want <goal>, so that <benefit>”
  + Use cases: Detailed descriptions of how a user interacts with the system to accomplish a goal, including the preconditions, postconditions, main flow, and alternative flows
  + Functional requirements: Statements of what the system must do or provide to meet the user needs, such as inputs, outputs, processes, calculations, validations, etc.
  + Non-functional requirements: Statements of how the system must perform or behave to meet the quality standards, such as reliability, availability, security, usability, etc.
* **Confirm the requirements**: The requirements are reviewed with all stakeholders to ensure that they capture what was intended and all parties have a common understanding of each of them. All ambiguity found in the requirements is revised. The requirements are validated through prototyping and testing, which includes feasibility, usability, and product concept testing.
* **Prioritize the requirements**: The requirements are prioritized according to how they will affect the system’s goals and objectives. By assigning labels to features such as “must have,” “high want,” and “good to have,” they are ranked in order of importance. However, ranking each requirement within those categories is equally crucial. This is due to two factors:
  + Time to market: Sometimes plans change. When they do, you might have to cut back on features and specifications to stay on schedule for delivery. Your team shouldn’t start by implementing the most straightforward requirements just to discover that there isn’t enough time to finish all of your must-haves.
  + The evolution of requirements: You’re likely to identify new requirements as the implementation progresses. Some might be of the utmost importance and take precedence over already-existing requirements. You must understand where those new requirements fall in your hierarchy. If you don’t, less significant requirements will decide what gets deployed first, which could negatively affect the success of your product.

**Functional Requirements**

1**.** The system will enable a user / customer to join the bank queue with the aid of a mobile phone or personal computer provided with internet connection.

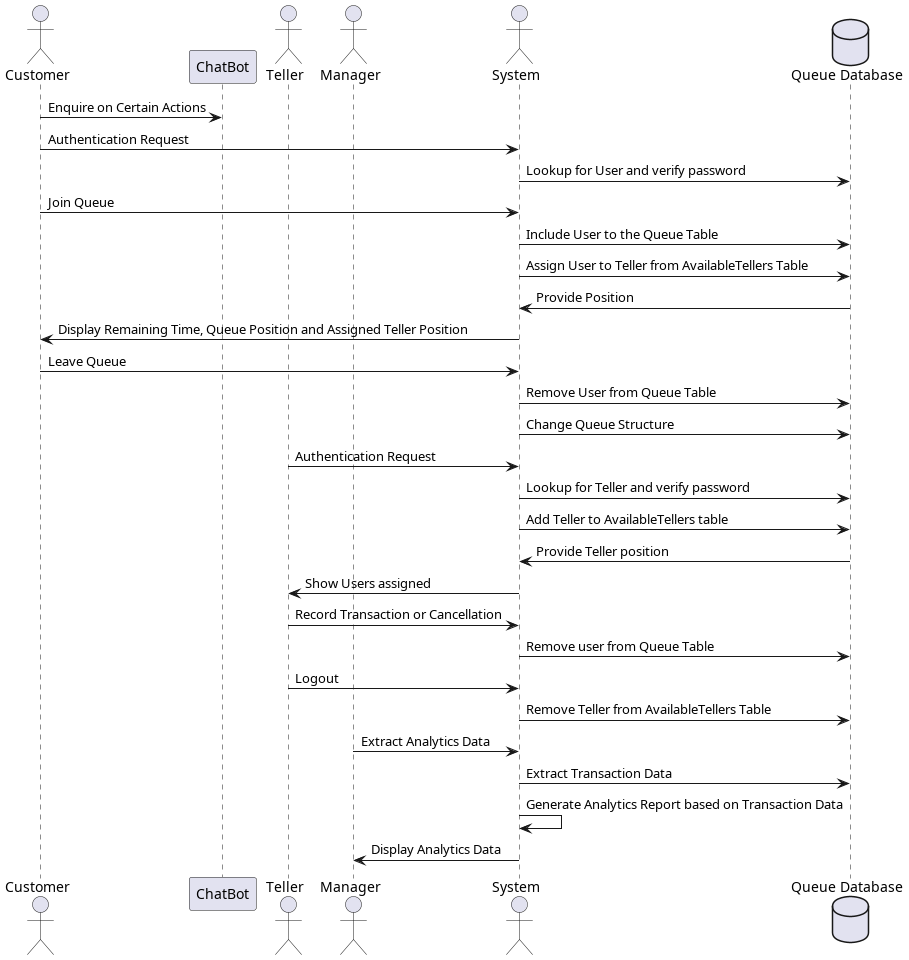
2. The system will be able to assign a customer to a teller based on the the queue status and teller service rate.

3. The system will be able to reassign the queue when a user leaves.

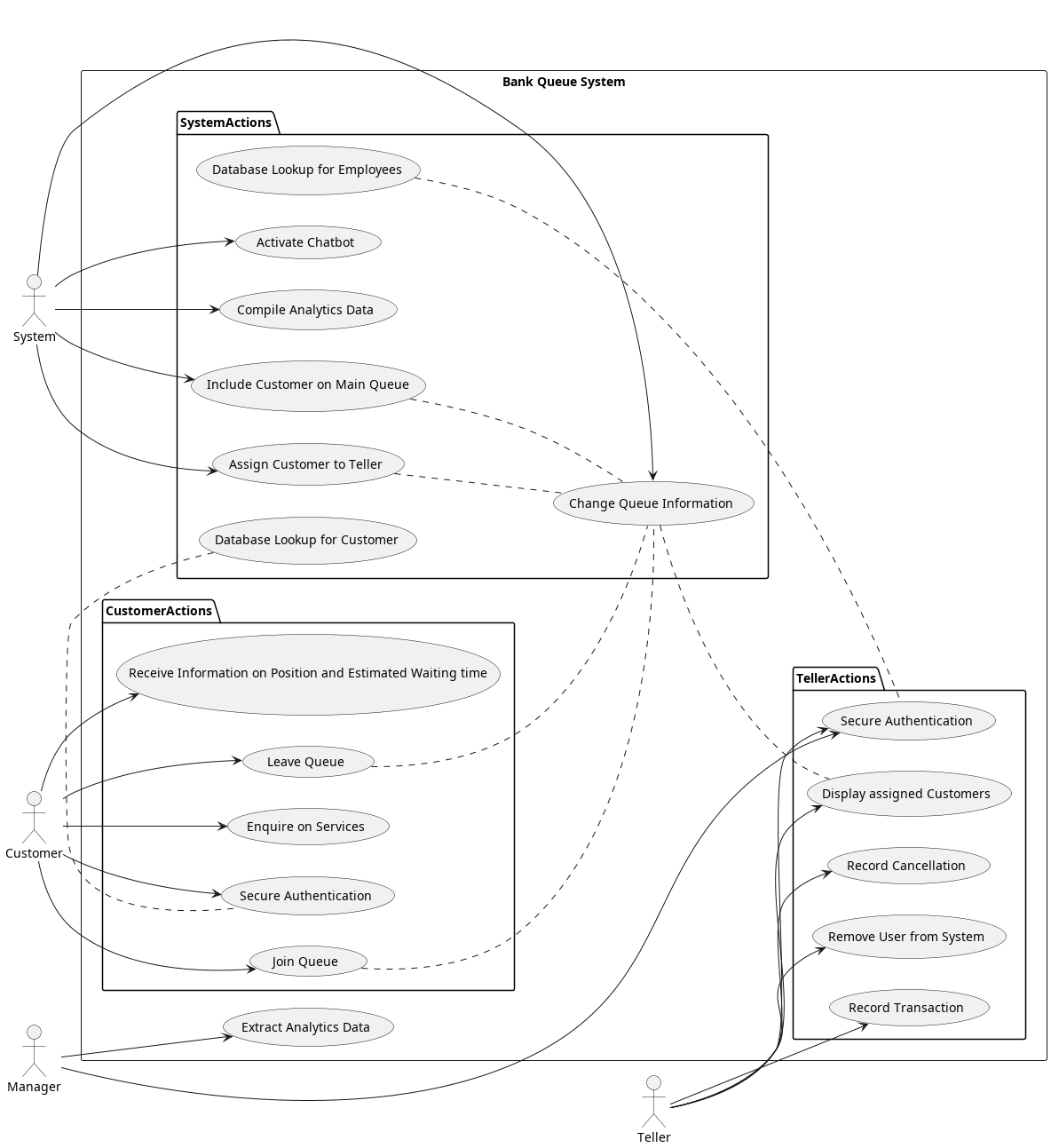
4. The system will enable tellers to record transactions both completed and cancelled transactions

5. The system will be able to generate analytical data with a pdf downloadable report.

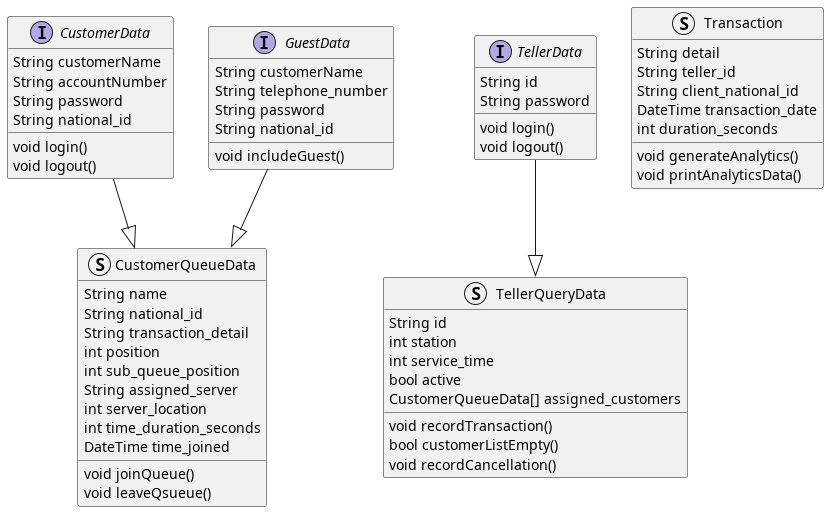
**UML Diagrams**

****

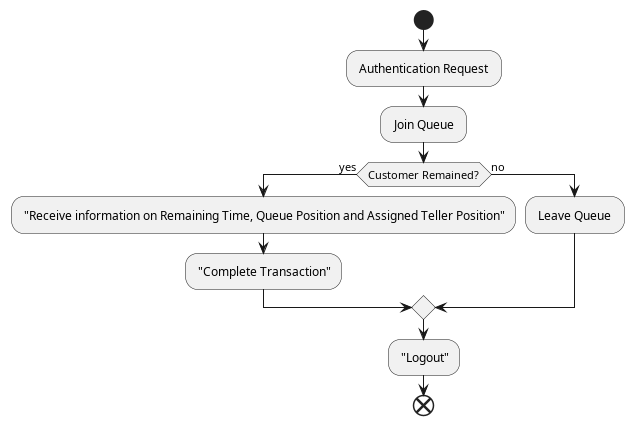
*Figure 0.5 – Sequence diagram for Smart bank queue management system*



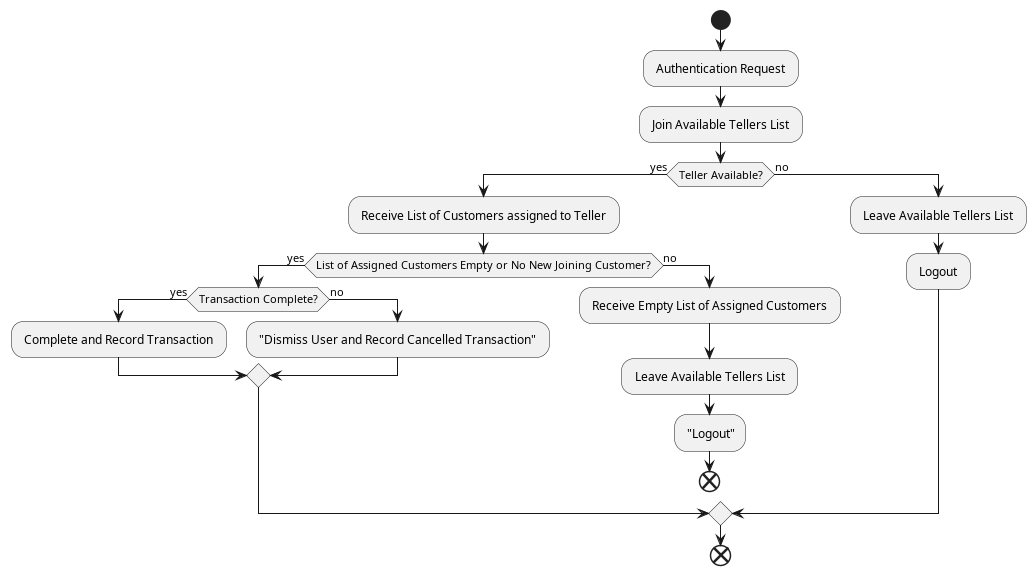
*Figure 0.6 – Use case diagram for smart bank queue system*

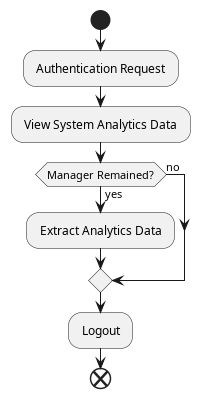


*Figure 0.7 – Class diagram for smart bank queue system*



*Figure 0.8- Activity diagram for customer*

*Figure 0.9-Activity diagram for Teller*



*Figure 1.0- Activity diagram for Manager*

**Non-functional requirements**

**1. Usability**: The system will be easy to use and understand by the customers and the staff. The system provides a user-friendly interface that is consistent, intuitive, and responsive.

**2**. **Performance:** The system will provide fast and reliable service to the customers and the staff. The system will process request with an average latency of less than 5 seconds. The system can handle at least 1000 concurrent users without degrading the performance.

**Security Concepts:** This system has implemented four security concept.

* **Confidentiality**: The term "confidentiality" refers to a set of standards and procedures, typically carried out in the context of confidentiality agreements, that guarantee that the information is only accessible by specific parties or locations. This system allows account holders to access the system with their valid details and allows them alone to view their information relating to the bank queue aside the bank operators (such as teller viewing account holder ID card).
* **Authentication**: To ensure that the data the user is claiming belongs to it, authentication must be performed. Before a user of this system gets access to the details in the system, he or she must go through mandatory authentication for verification.
* **Data Integrity**: Maintaining and ensuring that the data is true and consistent throughout its full life cycle is known as data integrity. There are no forged data in the system. Each data displayed in the system comes as a result of the activities the user performed.
* **Non-repudiation:** Nonrepudiation ensures that no party can deny that it sent or received a message via encryption and/or digital signatures or approved some information. It also cannot deny the authenticity of its signature on a document. The system will use digital signatures to ensure the non-repudiation of origin and emission for the transactions made by the customers and the staff. Digital signatures are cryptographic techniques that use public-key encryption and hashing algorithms to generate unique and verifiable signatures for electronic messages or documents.

**Project Methodology**

A set of ideas and practices known as a project management methodology can help you structure your projects to ensure their success. (Project Management Methodologies - Everything You Need to Know) Examples of the various Project Methodologies include:

* Waterfall methodology
* Scrum methodology
* Kanban methodology
* Lean methodology
* eXtreme Programming (XP) methodology
* Scrumban methodology
* Adaptive project framework (APF) methodology
* Critical chain project management
* Critical path methodology (Critical Path Analysis)
* New product introduction (NPI)
* Outcome mapping
* Package-enabled reengineering (PER)
* Six Sigma
* Project Management Institute’s Project Management Book of Knowledge (PMI’s PMBOK)
* PRojects IN Controlled Environments (PRINCE2) Methodology
* Rapid application development (RAD) methodology
* Agile methodology

The agile methodology was used in this project due to its unique features which make it different from the other methodologies. Agile values people, their relationships, and interactions through tools; offers collaboration throughout the development process; responds to change rather than following a set plan, and focuses on presenting functional software rather than documentation. Unlike Plan driven, Agile is well equipped for the complexity and variability of development projects. According to the agile approach, we develop in short sprints or iterations, each of which contains a defined duration and a list of tasks to be delivered, but in no specific order. During sprints, we work to deliver functional software or other tangible and verifiable results. The Agile method helped us to collaborate and focus on team strengths and efficiencies, as well as internal feedback from various stakeholders.

**Various Software Processes**

A software process is a collection of actions with a corresponding result that results in a software product. These tasks are mainly done by software engineers. All software processes share these four fundamental process steps. These processes include:

• Software specifications: The software's capabilities and restrictions on operation must be specified. This procedure involves a thorough description of the software system that will be created, together with its functional and non-functional requirements.

• Software development: It is necessary to create the software to satisfy the need. Designing, programming, documenting, testing, and bug fixing are all done during this phase.

• Software validation: To make sure the program fulfills the customer's requirements, it must be validated. In this process, software products are evaluated to make sure they satisfy both the needs of end users and business requirements.

• Software evolution: To adapt to changing customer needs, the software must also change. Software is first created, and then it is promptly updated for several reasons. (Software Processes - Javatpoint, n.d.) (Software Processes in Software Engineering - GeeksforGeeks, 2019)

**Chosen Model and Justification**

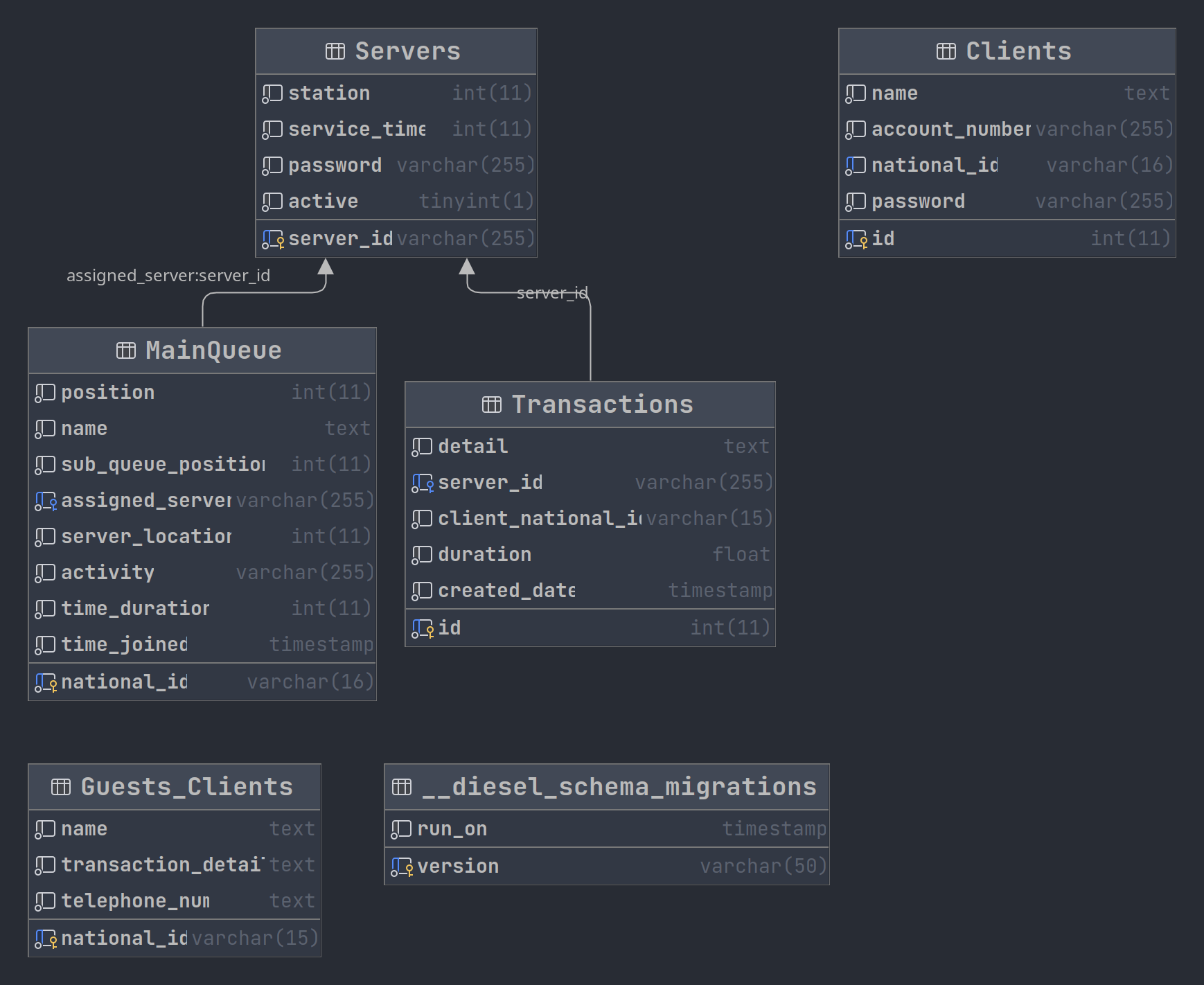
Depending on the type and functionality of the proposed system, different types of software development process models are used for different types of software. The requirements were determined after a careful examination of the existing systems. The approach used in the system development is the incremental process model. The incremental development model is based on the idea of developing the first implementation, exposing it to user feedback, and developing it in multiple versions until a suitable system is developed. Specification, development, and validation activities are nested rather than separated, with quick comments on all activities. The model is divided into three simultaneous activities:

• Specification

• Development

• Validation

**Logical Designs**



*Figure 1.1-DB SCHEMA*

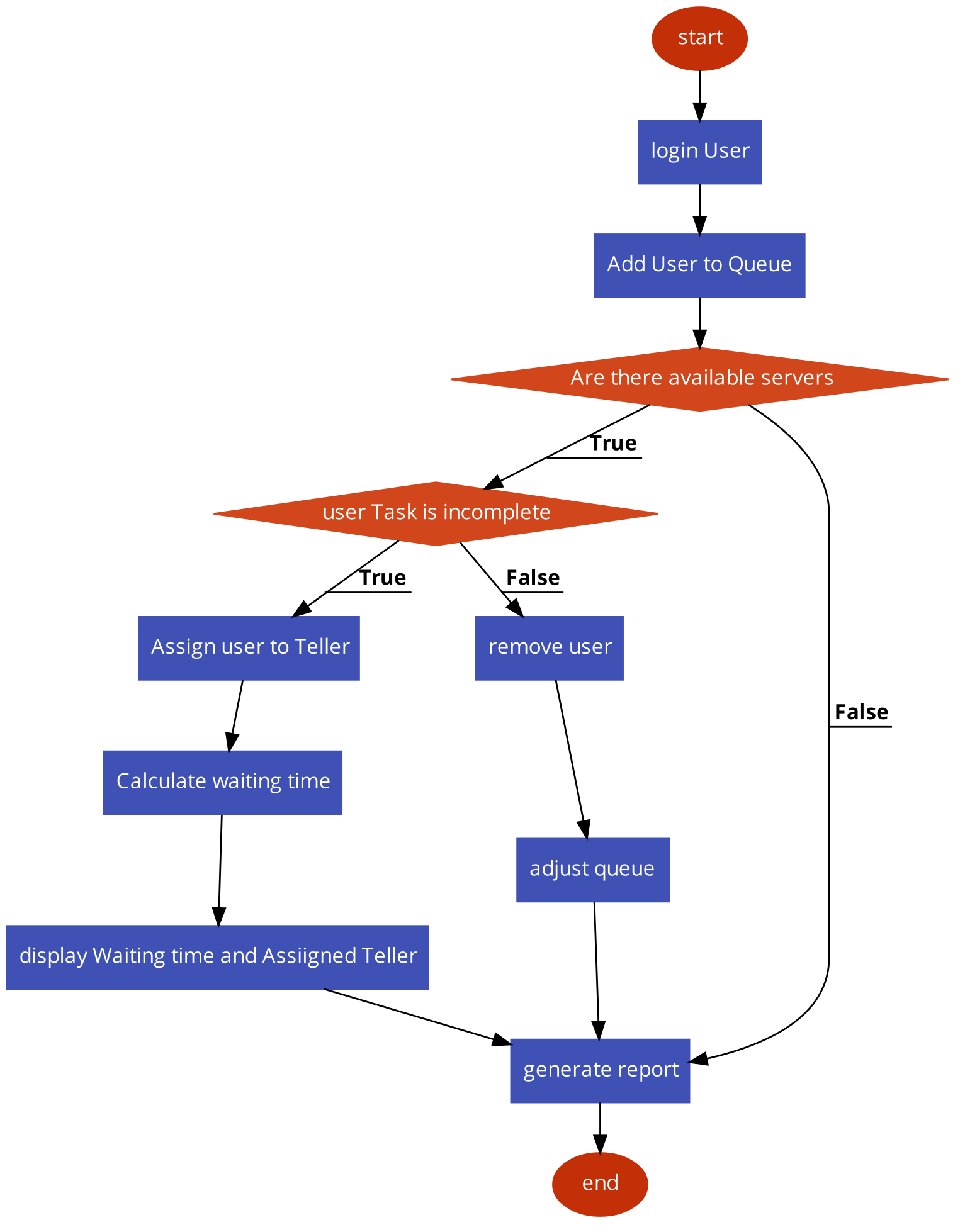
**Chapter 4**

**IMPLEMENTATION AND RESULTS**

**4.0 Overview**

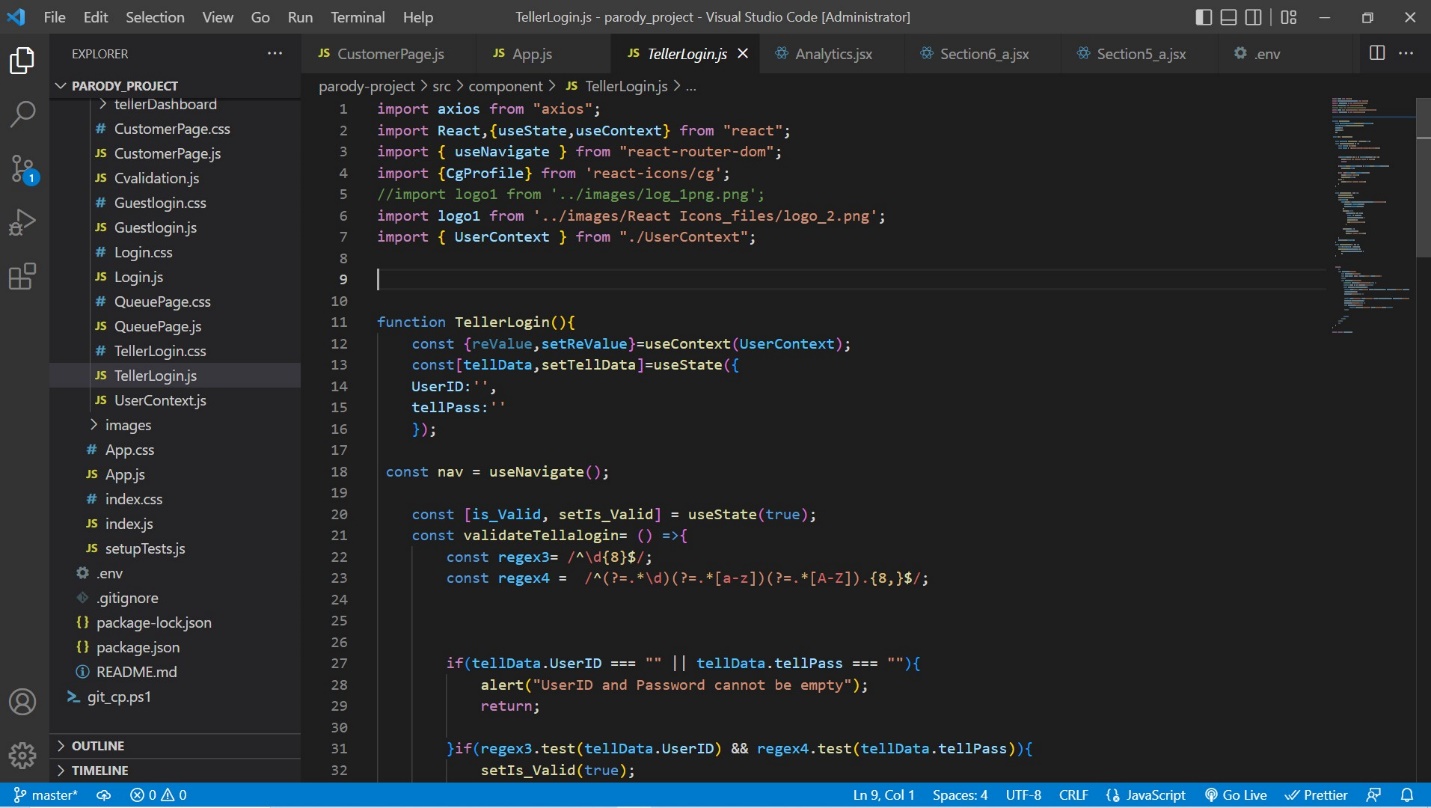
The system implementation and documentation is the phase of system development that constitutes proof of the making of the new system. Planning the implementation of a system begins in the detailed design stage. This phase of the project will involve the installation and testing of a thoroughly structured program to make sure that the result of the new system is the anticipated one. Most system failures are due to inadequate conduct of the implementation phase. This phase is very important because the success of the project depends on it.

**4.1 Mapping Logical Design onto Physical Platform**

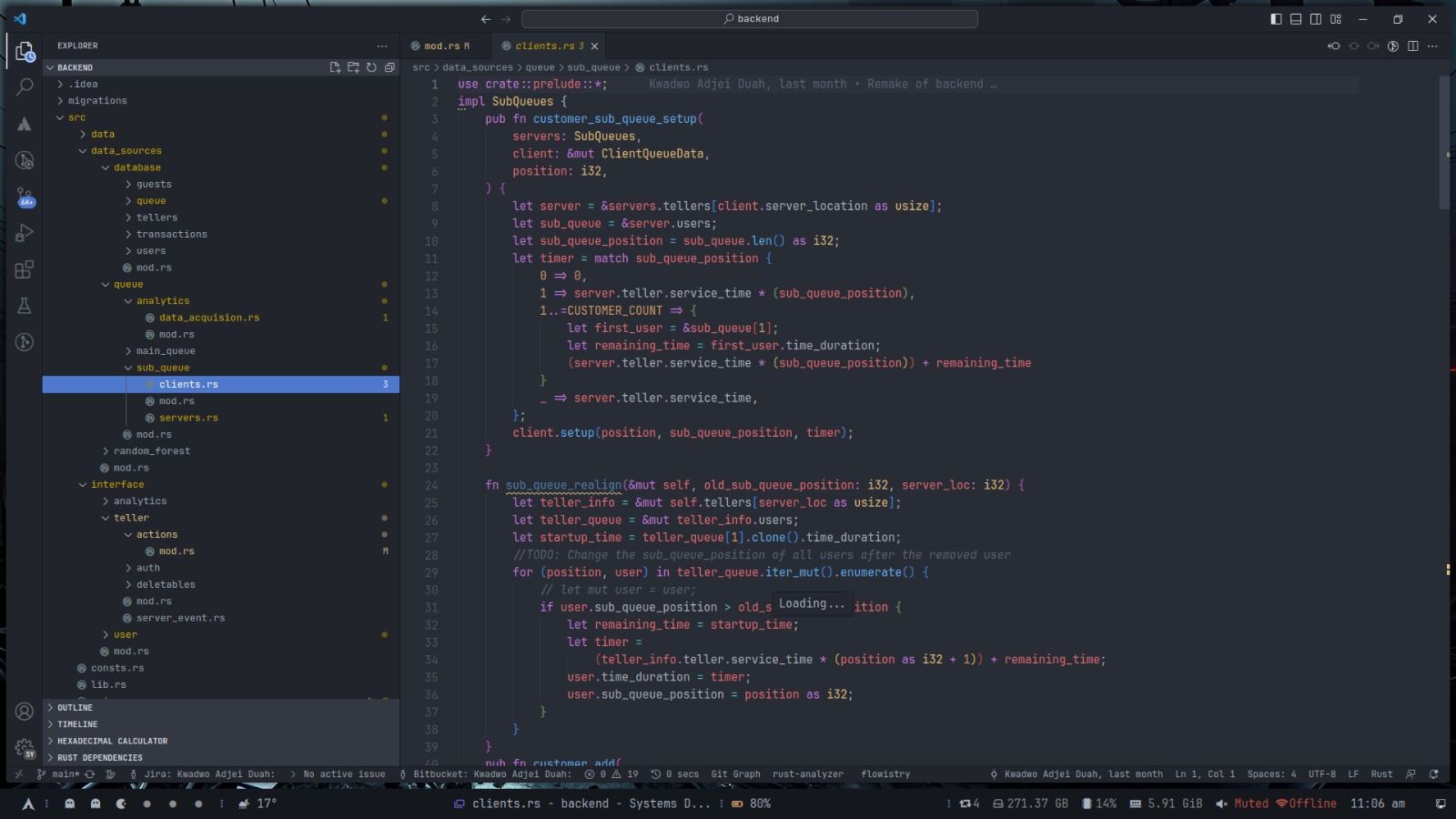
****

*Figure 1.1 –Flowchart of the system*

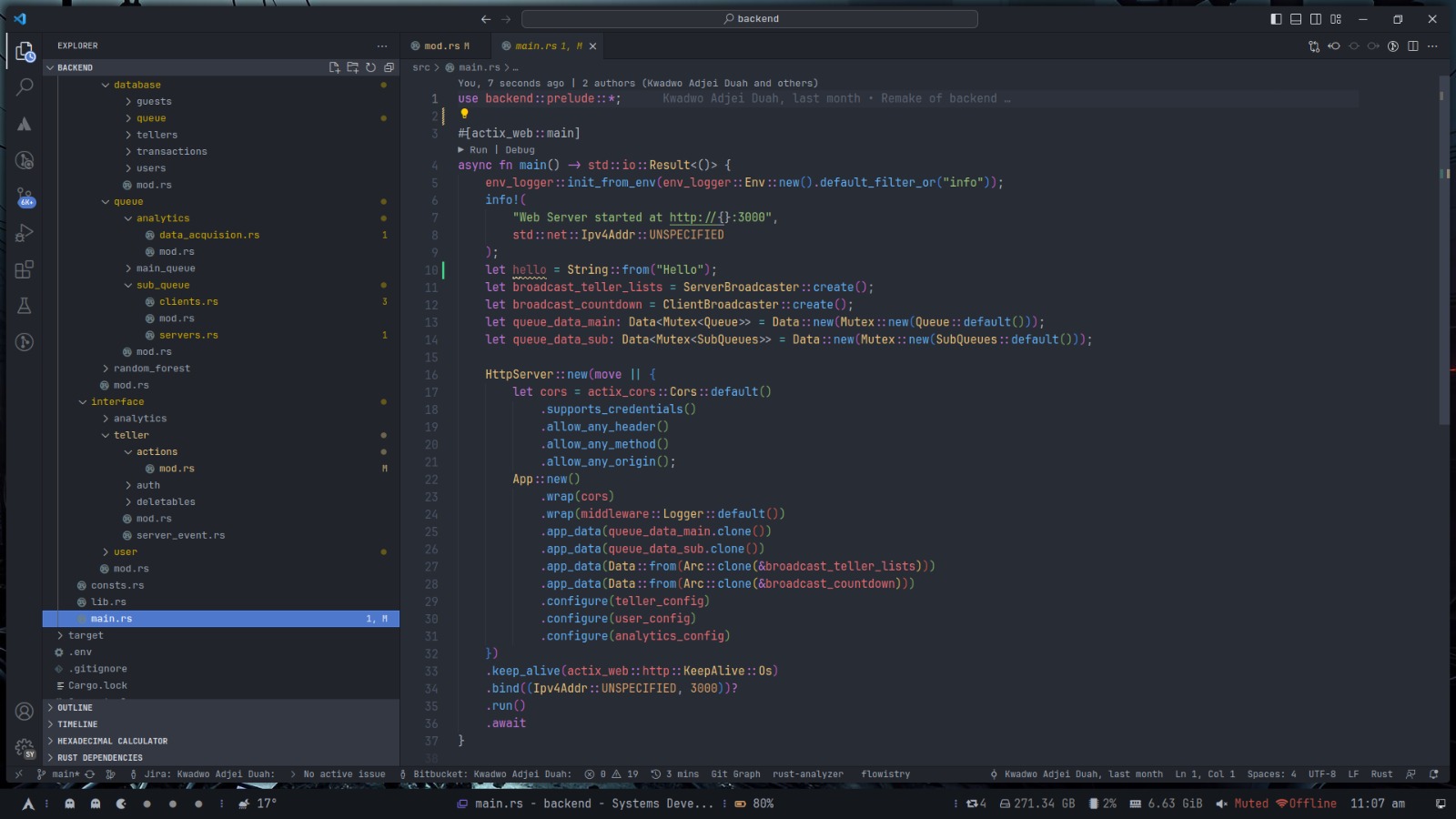
**Construction**

****

*Figure 1.2-Frontend code*



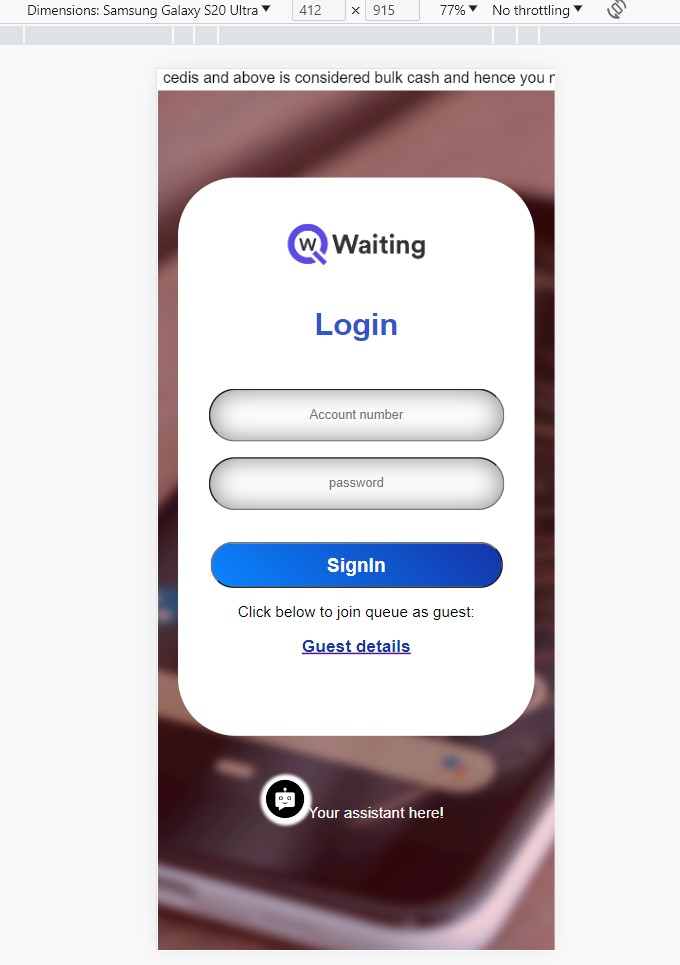
*Figure 1.3-Backend Code\_1*

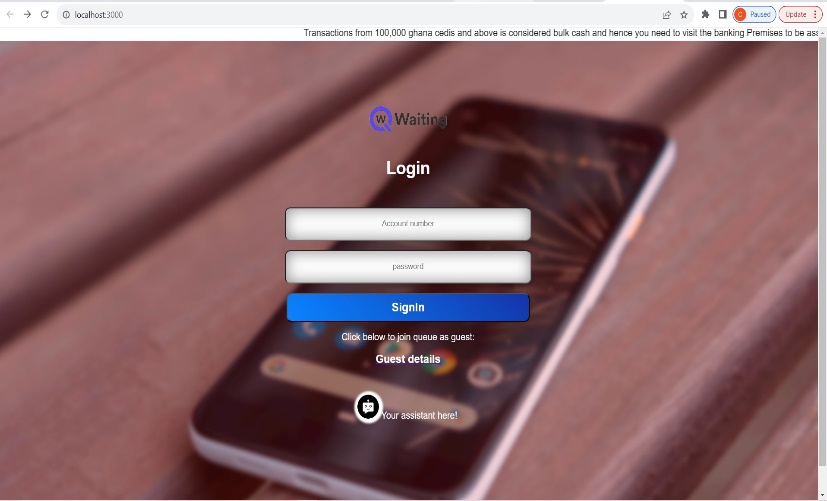


*Figure 1.4-Backend Code\_2*

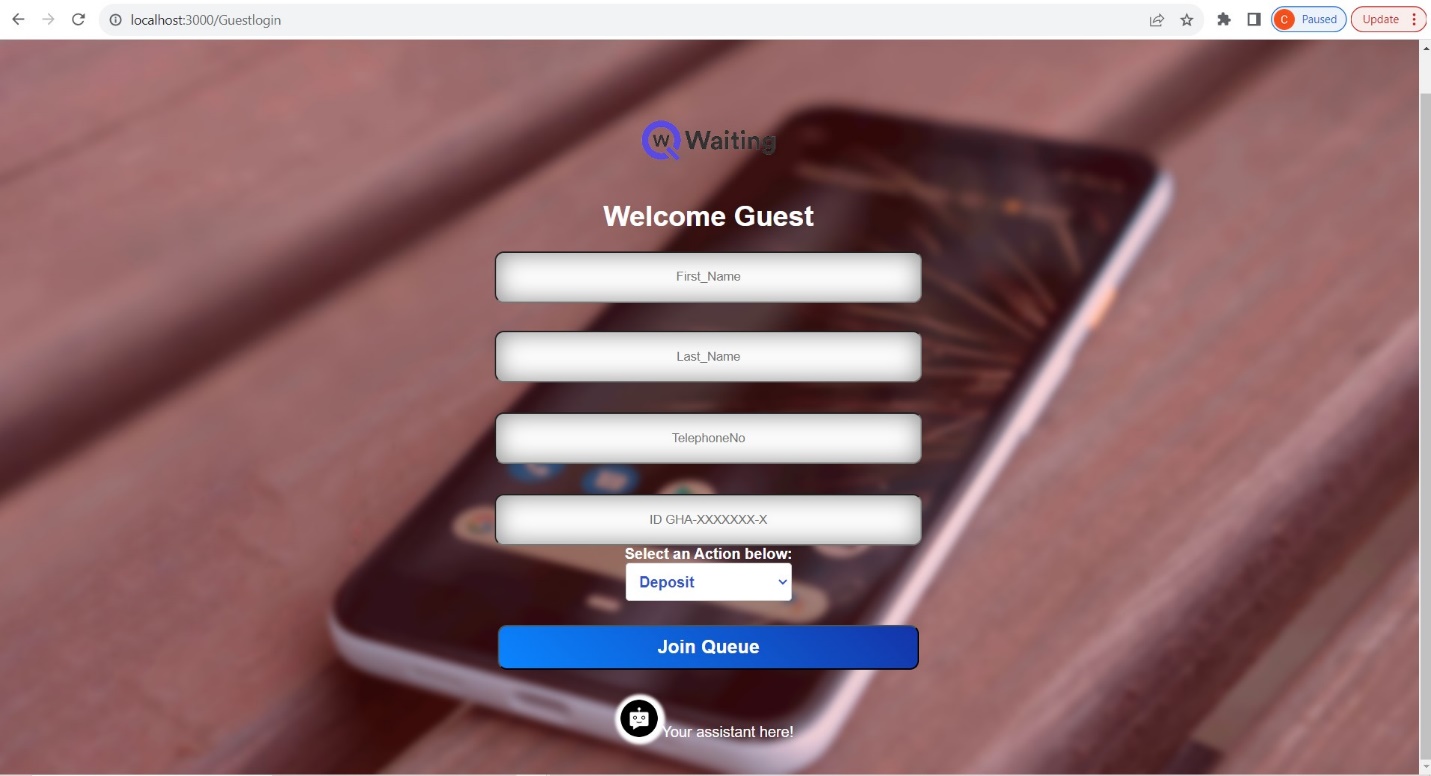
**Screenshots of the system**

Customer View (responsive view on mobile and desktop)

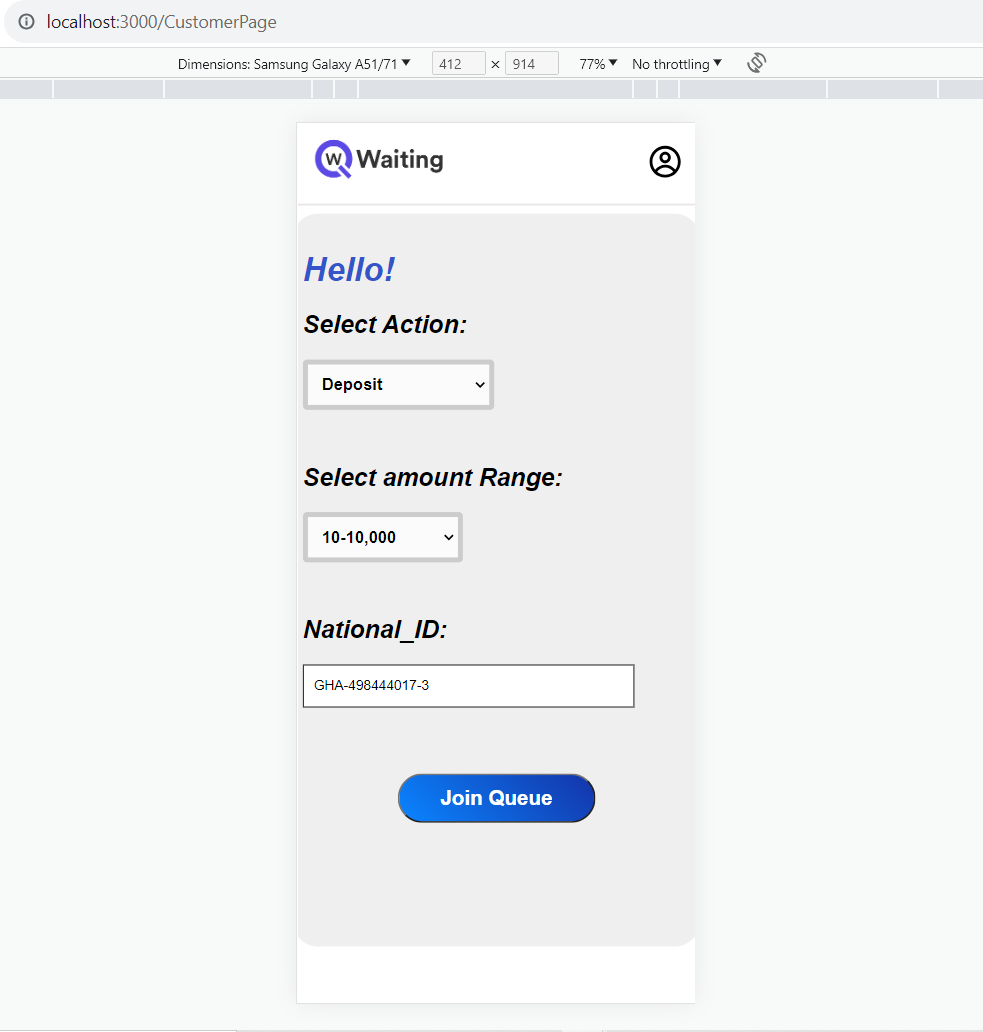




*Figure 1.5-Customer login interface*



*Figure 1.6- Guest(not an account owner) interface*

**

*Figure 1.7-Customer details page*

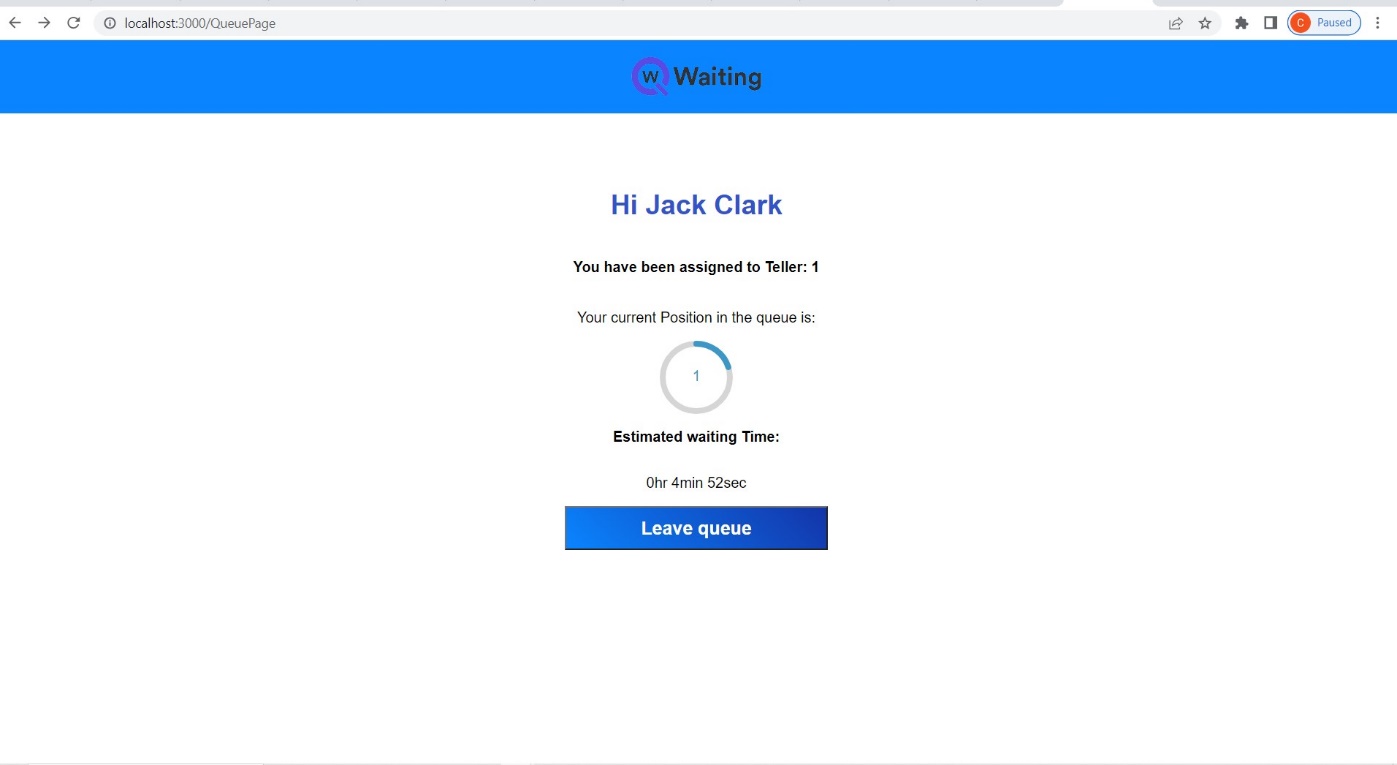


Figure 1.8-Queue Page (desktop View)

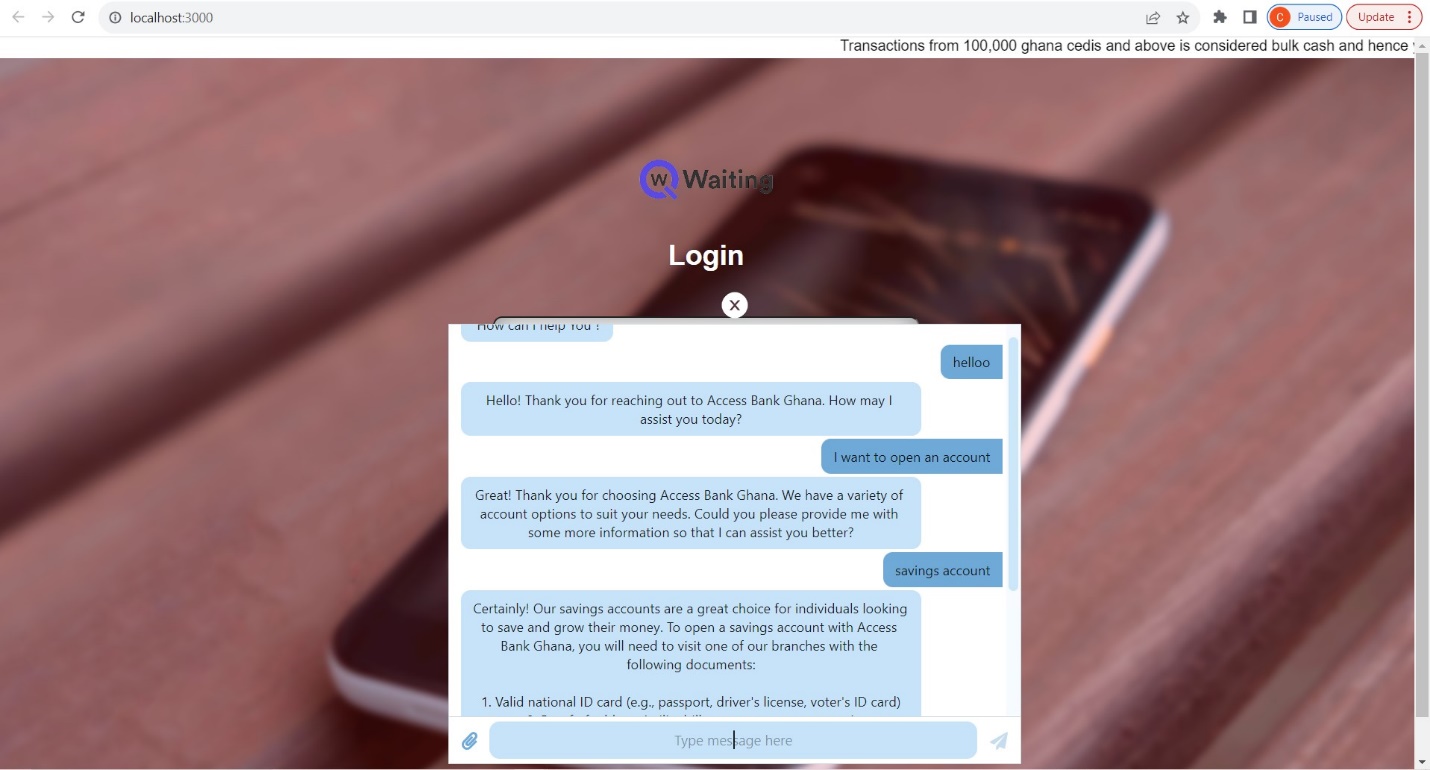
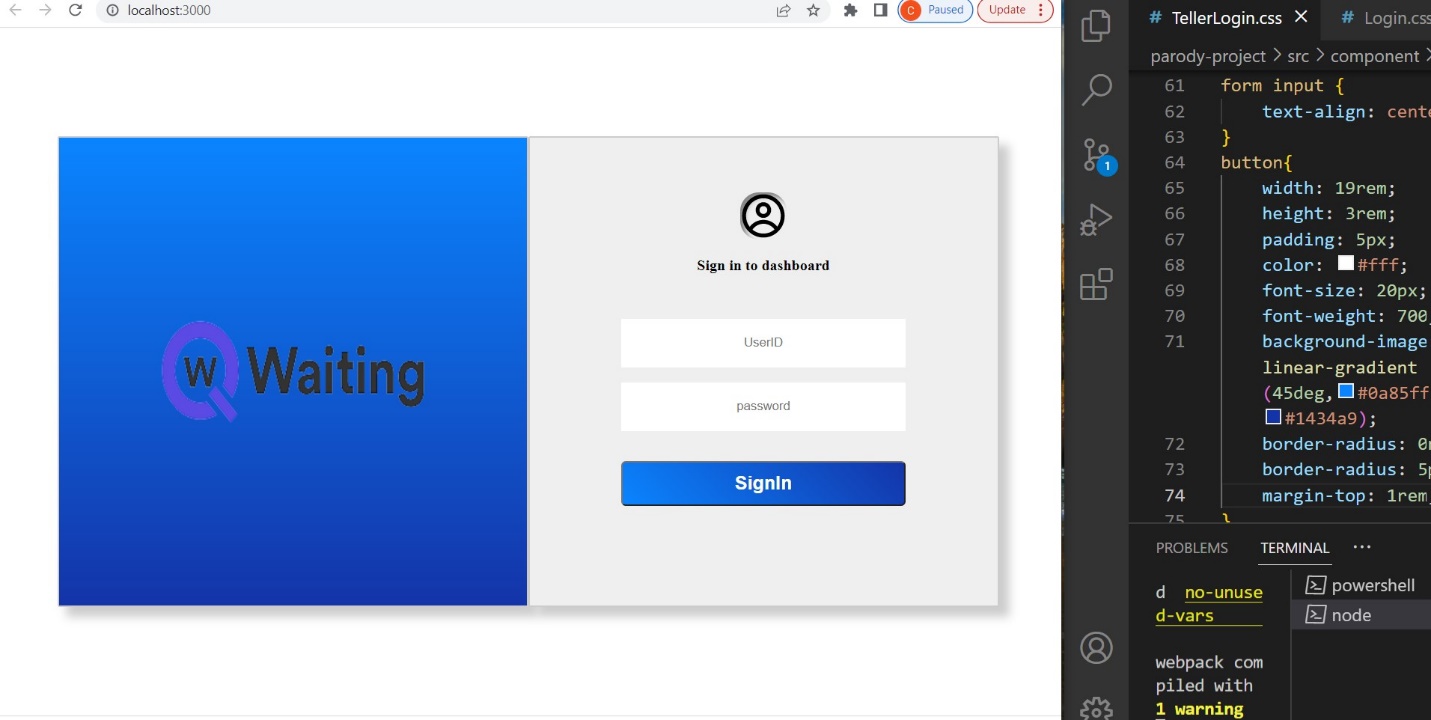
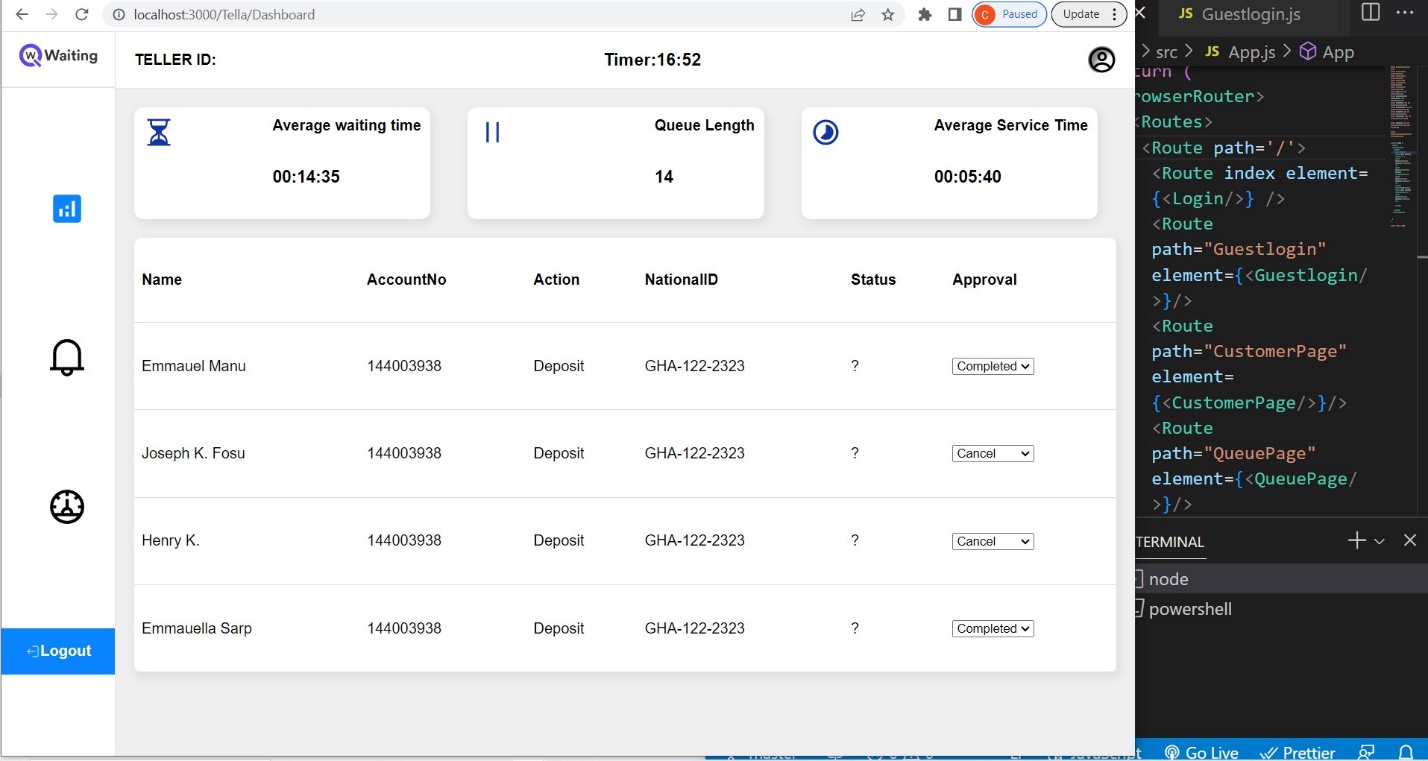


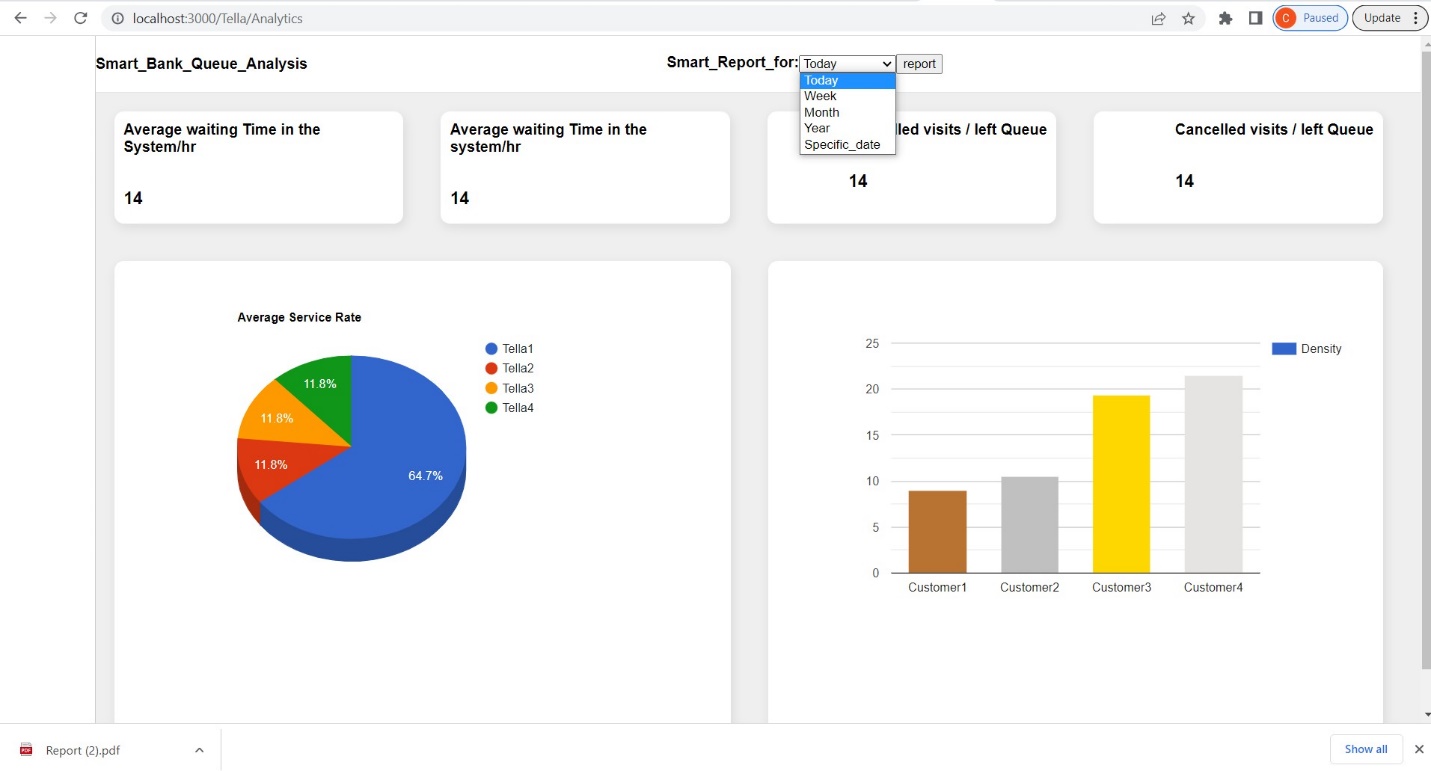
Figure 1.9-Chatbot view



*Figure2.0-Teller Login Page*

**

*Figure2.1-Teller Dashboard*

**

*Figure2.2-System Analytics Page& Report generation (downloadable pdf format)*

**Testing**

The tools used for testing the react app include:

• Jest is a JavaScript test runner that allows you to access the DOM through jsdom. While jsdom is only an approximation of how the browser works, it is often good enough for testing React components. Jest offers high iteration speed combined with powerful features like simulation modules and timers, giving you more control over code execution.

• React Testing Library is a set of tools that you can use to test React components without relying on their implementation details. This approach makes refactoring a breeze and also leads you to accessibility best practices. While there is no way to make a component "flat" without its children, you can do so by poking fun at a test runner like Jest.

The tools used for the rust backend server testing is:

**Cargo Test :** In-built rust testing tool.

**Results**

Following system implementation, system testing was carried out to evaluate the system's performance. System testing is a crucial step in the system development process. It functioned once the actual system or prototype has been constructed, or after development. The testing phase is very beneficial and significant since it might uncover hidden software bugs. User and system requirements are typically the foundation of testing criteria, which determines whether or not the system satisfies the requirements. System dependability is crucial for a system to function correctly for its end users, and testing can confirm this. Unit testing and module integration testing are the two components of system testing. A type of testing on each component in a big system is called unit testing. Unit testing is done on each module to ensure that it is fully functional before module integration. The procedure of combining all modules would be tested during the module integration testing. The ultimate system is complete after all modules can communicate with one another, and integration testing would test the entire system.

**Results for the Web App**

*Table-1 React\_application\_module testing*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Test Objective** | **Test Step** | **Expected Result** | **Result** |
| **1** | To ensure the react application launches successfully in the browser | Run the react application on the browser using the terminal | The application should run successfully without errors | Passed |
| **2** | To ensure the navigation buttons take a user to the required location | Toggle between each of the navigation tabs to switch from one page to the other | There should be a smooth transition between the pages in the application | Passed |
| **3** | To ensure the navigation buttons take a user to the required page | Refresh the page after each toggle | A refreshed page should display the same page updating details where necessary | Passed |

| No | Test Objective | Test Step | Expected Result | Result |
| --- | --- | --- | --- | --- |
| 1 | Start the server and connect to the MySQL Database Server | 1. Start the database server2. start the program in the command line and add the URL of the database as an argument | Program Started with showing the log of the process | Passed |
| 2 | Login Account holding customer | 1. Enter your account number and password2. Click on the Login button | User is moved to the Transaction screen | Passed |
| 3 | Joining Queue | 1. Provide the transaction detail2. Click on the Join Queue Button | User is moved to the Queue page | Passed |
| 4 | Leave Queue | Click on the Leave Queue Button | User is moved back to the Login Screen and the Queue Page of other users would be updated | Passed |
| 5 | Login Bank Employee | 1. Enter your employee ID and password2. Click on the Login button | Employee is moved back to the Dashboard Screen and sees the list of Customers assigned to him or her | Passed |
| 6 | Recording Successful Transactions | Upon Completing the transaction, click on the Complete button | Customer’s name is removed from the table and transaction data is stored in the database | Passed |
| 7 | Cancelling transaction | Click on the Cancel button | Customer’s name is removed from the table | Passed |
| 8 | Logout Teller | Upon an empty queue, click the logout button | Employee is sent back to the login button | Passed |

*Table2-backend\_server\_module Testing*

**Chapter 5**

**FINDINGS AND CONCLUSIONS**

**5.0 Overview**

The accomplishments made possible by this effort are listed below.

* To demonstrate how the smart bank queueing system works, interfaces for bank staff and customers which is responsive to both mobile devices and PC have been developed.
* Adequate privacy and ethical measures have been taken to ensure the protection of customer data and identity.
* A system design that optimizes the allocation of resources and reduces the waiting time and congestion of customers.
* An interface design that provides clear and user-friendly feedback and guidance for customers and staff.
* To help the bank monitor and evaluate the performance of the system, the system generates reports and statistics on various indicators such as queue length, service time, customer satisfaction, and so on.

**Findings**

At the end of our project study and implementation, below are some of the findings we got:

* Some customers are unaware of the benefits and features of the smart bank queueing system to the bank in operation.
* There are customers who appreciate the convenience and speed of the smart bank queueing system and want to see more improvements in the user interface
* Some customers are skeptical about the reliability and accuracy of the smart bank queueing system and prefer to use the traditional methods of banking services and others also who aren’t so much enlightened such as the aged prefer not to use the system and choose say they prefer someone helped them at the banking premises.
* The smart bank queueing system can be applied to other domains and scenarios, not just limited to banking services, such as retail, healthcare, education, and so on.

**Conclusions**

The computer system’s ability to efficiently and accurately analyze and optimize the queueing process of various bank services has shown that it is immensely beneficial to both the customers and the staff. Its value cannot be underestimated, especially in enhancing customer satisfaction and loyalty not leaving out the smooth waiting experience they attain.

The smart bank queueing system makes the banking experience more convenient and enjoyable for the customers, as they can save time and effort by using the system’s features and not bother so much about their physical position being overridden by another because their position in the queue is visible to them and orderly arranged. This system also reduces stress and frustration for the staff, as they can manage their workload and resources more effectively. This system is a lot of fun to use, as it provides clear and user-friendly guidance and information.

**Challenges/ limitations of the system**

1. The system can be used only with the aid of devices such as computer,tablet or mobile phone(smart phone) with internet connection to access the bank queue and hence customers without such devices fall short and have to go to the bank premises.

2.The illiterate would find difficulty using the smart bank queue system since it need some level of basic understanding of English and technological familiarity which are absent speaking of an illiterate.

3. Users would need to be at the banking premise to use the system rather than from far away in their various locations due to non-addition of GPS feature at the moment.

**Lesson learnt**

• Team Support: Software development is a huge task hence delegation and team support must be encouraged. Delegation of tasks appropriately and enforcing realistic deadlines are parameters any software engineer must practice.

• Communication: Communication does not only involve oral language. But also communicating with clear diagrams that all team members can understand.

• Error Handling Landscape: Throughout this project, we were heavily challenged with a lot of errors and bugs, seeing that we were not experienced in so much with the various models and server side events development. We have learnt by this project that a good software engineer must have a positive tolerance towards handling errors.

• Test the Parts Before the Whole: We learnt that in software development one must test the software in parts rather than testing the whole module. Doing so helps in identifying and fixing bugs quickly

**Recommendations for future works**

After thinking about the idea of the smart bank queue system and presenting it to our Supervisor Dr. Twum Frimpong and others the following suggestions are hereby made:

1.The inclusion of GPS feature to allow users join bank queue from their homes and not necessarily coming to the banking premises by checking to see if the distance to the bank is suitable to meet the estimated waiting time that would be assigned to the customer.

2.Advanced UI interfaces for Customers.

**Recommendations for project commercialization**

The following are the recommendations made after careful examination of the commercial scope:

1. Change of report generation format to suit the standard or format of the preferred organization making use of smart bank queue system

2.Purchasing of cloud services by the firms to run the system in case they don’t have the onsite expected system capacity to run the application and provide necessary backups.

**References**

*1. Project Management Methodologies - Everything You Need To Know. (n.d.). Project Management Methodologies - Everything You Need to Know. Retrieved September 29, 2022, from <https://www.teamwork.com/project-management-guide/project-management-methodologies/>*

*2. Software Processes in Software Engineering - GeeksforGeeks. (2019, September 4). GeeksforGeeks. Retrieved September 29, 2022, from* [*https://www.geeksforgeeks.org/software-processes-in-software-engineering/*](https://www.geeksforgeeks.org/software-processes-in-software-engineering/)

*3. Software Processes - javatpoint. (n.d.). www.javatpoint.com. Retrieved September 29, 2022, from* [*https://www.javatpoint.com/software-processes*](https://www.javatpoint.com/software-processes)

*4.*  *React (JavaScript library) - Wikipedia. (2013, May 29). React (JavaScript Library) - Wikipedia. Retrieved October 11, 2022, from* [*https://en.wikipedia.org/wiki/React\_(JavaScript\_library)#cite\_note-react-3*](https://en.wikipedia.org/wiki/React_(JavaScript_library)#cite_note-react-3)