

**Software Development Documentation  
of**

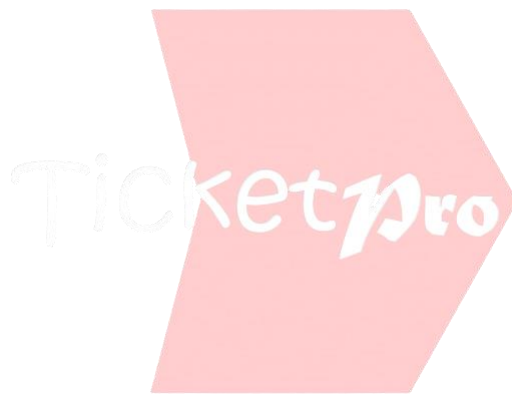
**TicketPro**

**GROUP 5 CS 130**

**Computer Science & Software Engineering**

**Team  
Members**

Nkandu Nonde	22176682
Silas Ndhlovu B	22103729
Sunny Hambalo	22176587
Mapalo Daka	22179349
Taonga Mkokweza	21169789
Munali Mbeha	22110420
Chiyesu Mashimbwa	22103660
Gerald Mumba	22107288
Kuunda Simuchimba	22175822



# Table of Contents

<b>1</b>	<b>INTRODUCTION .....</b>	<b>4</b>
	BACKGROUND OF THE STUDY .....	4
	PROBLEM STATEMENT .....	6
	OBJECTIVES.....	6
	PUPPOSE, SCOPE & APPLICABILITY .....	6
<b>2</b>	<b>RESEARCH METHODOLOGY.....</b>	<b>11</b>
	INTRODUCTION .....	11
	METHODOLOGY.....	11
	INFORMATION GATHERING & ANALYSIS .....	16
	REQUIREMENTS SPECIFICATION .....	16
	SYSTEM ANALYSIS.....	18
	UML DESIGN .....	22
<b>3</b>	<b>IMPLEMENTATION .....</b>	<b>26</b>
	INTRODUCTION .....	26
	SYSTEM IMPLEMENTATION .....	26
	RESULTS.....	29



# 1 Introduction

Technology is causing significant changes in the commercialization of products. It offers a distribution channel that enables customers to book air tickets rapidly, conveniently and with substantial price savings. It tears down the barriers to the availability of competitive ticket price information, intensifies market transparency and gives more power to the e-customer. The Ticket Reservation System is a software application designed to facilitate the efficient and seamless booking and management of tickets for various events, travel services, or any other ticket-based services. This system allows customers to browse available options, select preferred tickets, make payments, and receive confirmations for their reservations.

Step into ticket pro where we seamlessly combine convenience and efficiency to elevate your event experience! Whether you're anticipating an enthralling concert, immersing yourself in the vibrancy of a sporting event, moving from one place to another or indulging in the allure of a theatre performance. Our platform is designed to streamline the ticketing process and ensure that you can easily reserve your place among the audience.

Ticketpro

## 1.1 BACKGROUND OF THE STUDY

The technology acceptance model (TAM) was developed by Davis (1989) to explain acceptance of information technology for different tasks. He showed that the intention to use a system is determined by what an individual believes about that system; the two most important beliefs being perception of the usefulness of the new technology and the perceived ease or difficulty of that use. McKechnie et al. (2006) point out that TAM has been successfully applied to a range of computer-based technology.

A ticket reservation system has advantages, for both users and event organizers. One of its benefits is the convenience it offers. Users can easily reserve tickets from the comfort of their homes or while on the move eliminating the need to physically visit ticket outlets or endure long queues. With a ticket reservation system, users can save time that would have otherwise been spent traveling or waiting in line. The process of finding and securing tickets is made efficient and hassle-free allowing users to complete their bookings swiftly. This system provides access to a range of events, including concerts,

sports matches, theatre shows, movies and more. Ensuring user security is a priority for ticket reservation systems with measures in place to safeguard personal and financial information. Additionally, this system often offers marketing and promotional features to help event organizers expand their reach and boost ticket sales.

A ticket reservation system has the ability to address hurdles such, as ensuring tickets are available by handling scalability providing a good user experience and offering personalization options. A system for buying tickets is also capable of overcoming technical difficulties such as performance and responsiveness, security, integration with payment gateways, system reliability and availability, and data management.

Our motivation comes from the conviction that everyone should have access to high-quality booking services. Our mechanism for purchasing tickets aims to make these events accessible to everyone, regardless of location, background, or circumstance. We work to promote diversity, making sure that everyone has an equal opportunity to take in their favourite shows and remarkable experiences.

## **1.2 PROBLEM STATEMENT**

The current ticket ordering systems struggle to give event guests a seamless and user-friendly experience. During the busiest times for ticket sales, users have trouble locating available tickets, navigating through complicated interfaces and experiencing poor response times. Frustration, poorer customer happiness, and missed opportunities to attend desired activities are the results of these problems.

Our ticket ordering system will integrate a number of significant features and improvements to address these issues. To start, the system will use cutting-edge algorithms and real-time inventory management to guarantee accurate information on ticket availability. Through a simple and dynamic seating chart, users will be able to quickly identify available tickets and choose their preferred seats.

Secondly, we will concentrate on streamlining the ticket booking process, enhancing navigation, and improving the system's user interface.

Thirdly, the system will safeguard user information, stop illegal access, and guarantee secure and trustworthy transactions.

Finally, the system will improve participation and raise the probability that users will show up at events that are in line with their interests.

The goal of the ticket booking system is to integrate these solutions in order to deliver a seamless, user-friendly, secure platform that streamlines the ticketing process, boosts customer happiness and makes sure that users can conveniently access and attend their selected events.

## 1.3 OBJECTIVES

The overall objective of this project is to:

1. Improve and optimize the event attendees' entire ticketing experience
2. Transfer business services
3. Improving the Security of Tickets
4. Enhance user encounter

## 1.4 PURPOSE, SCOPE AND APPLICABILITY

**PURPOSE:** The development of the ticket booking system aims to improve accessibility, convenience, and effectiveness in the ticketing process, offering a hassle-free experience for both customers and companies. Theoretical ideas from human-computer interaction, information systems, data analysis, recommendation systems, and software engineering can be used to improve the ticket booking system.

**SCOPE:** In order to make the process of selling and distributing tickets easier, ticket booking systems are frequently utilized across a variety of businesses, including transportation, entertainment, and events. A user-friendly interface is often part of a ticket booking system's approach, allowing users to browse alternatives, choose the tickets they want, and enter the information needed to make a reservation. Inventory management, which keeps track of a database of available tickets with information on seat locations, costs, and availability. After clients choose their tickets, the technology enables safe online payment processing, guaranteeing the confidentiality of critical information.

The following are typical presumptions made by ticketing systems: dependable inventory control, consistent payment processing and precise user input

Additionally, there are several restrictions on ticket booking systems, such as limited scalability, external dependencies, fraud, and scalper activity.

The goal of ticket booking systems is to make it easier to buy tickets for a variety of events, including sporting events, concerts, movies, flights, trains, and other forms of entertainment or transportation.

## **MAIN FUNCTIONS**

1. Purchasing in real time.
2. Store user data.
3. Ticket generation and management.
4. Payment via mobile money services and credit card services.

## **Functionality**

- **User Registration:** Users can create an account by providing their personal information, such as name, email address, and password.
- **Event Management:** Admin users can add, update, and delete events. Each event has details such as title, date, time, venue, and available seats.
- **Event Search:** Users can search for events based on various criteria, such as title, date, or venue.
- **Ticket Reservation:** Users can select an event and reserve a specific number of tickets, subject to availability.
- **Ticket Cancellation:** Users can cancel their reservations, freeing up the tickets for other users.
- **User Management:** Admin users can manage user accounts, including creating, updating, and deleting user profiles.
- **Reporting:** The system can generate reports on event attendance, ticket sales, and other relevant metrics.

### Functional (User requirements):

- The user should be able to view the list of tickets available and the respective details of each ticket.
- The user should be able to select the ticket type as per their choice.
- The user should be able to select the number of tickets to purchase.
- The user should have different methods for payment.
- The system should alert the user of an error in payment or lack of availability of desired ticket.
- The system should be operational 24/7.
- The system shall enable the user to enter the search text on the screen.
- The system shall enable the user to select multiple options on the screen to search.
- The system shall enable the user to navigate between the search results.
- The system shall notify the user when no matching product is found on the search.
- The system should work efficiently even with a large number of users and transactions taking place at once.
- The system shall allow the user to create a profile and set his credentials.
- The system shall authenticate user credentials to view the profile.
- The system should adequately inform the user of how their personal information will be used, stored and protected as well as any terms and conditions/policies that come with the services.
- The system shall send an order confirmation to the user through email.
- The user must be able to give feedback or issue complaints regarding the services offered by the system and receive assistance accordingly. (Suggestion list)

### Non-functional requirements:

- **Availability:** The system should be available to users at all time, with minimal planned maintenance windows. It should have backup and disaster recovery mechanisms in place to ensure continuous service availability.
- **Compliance:** The system should comply with relevant laws, regulations, and industry standards, such as data protection and privacy regulations. It should handle personal information appropriately and obtain necessary user consents when required.
- **Integration:** The system should have the ability to integrate with other relevant systems or platforms, such as payment gateways, customer relationship management (CRM) systems, or email notification services.  
Just to name but a few.



## **APPLICABILITY:**

### ***DIRECT APPLICATION***

1. Entertainment Events: Ticketing systems are widely utilized for concerts, plays, movies, sporting events, and other shows in the entertainment industry. The system allows users to immediately book tickets, select seats, and make payments.
2. Travel and Transportation: Systems for purchasing airline tickets give customers the ability to plan their travel, choose seats, and book flights. The same is true for railway ticket booking systems, which make it easy to reserve seats, check schedules, and purchase train tickets.
3. Hotels and lodging: A number of hotel reservation tools let customers book rooms, specify preferences, and control bookings. These systems frequently offer details on the costs, facilities, and room availability.
4. Tours and Attractions: Online ticket sales for tours, museums, theme parks, and other tourist attractions are made possible by ticketing systems. Users can check the schedule of events, buy tickets, and get confirmations or electronic tickets.
5. Conferences and Workshops: Professional events like conferences, workshops, seminars, or training sessions use ticket buying systems. Through the system, attendees can register, select sessions, and pay.

## ***INDIRECT APPLICATION***

- 1. Analytics and Insights:** Systems for purchasing tickets provide useful information about customer preferences, purchasing habits, and demand trends. With the use of data analysis, marketing tactics, pricing, and resource allocation may all be improved.
- 2. Customer Relationship Management:** Contact details and customer profiles are frequently integrated into ticketing systems. By knowing the preferences of the customers, this data may be utilized to create focused marketing campaigns, provide them with tailored offers, and enhance customer support.
- 3. Revenue Management:** Through dynamic pricing and demand forecasting, ticketing systems assist organizations in optimizing pricing strategies, adjusting ticket availability, and maximizing revenue. This is done by analysing booking trends and historical data.
- 4. Inventories Management:** Ticket ordering software helps with the management of event, travel, or lodging inventories. Real-time data on seat assignments, room occupancy, and ticket availability aid in resource optimization and prevent overbooking.
- 5. Access Control and Security:** For access control at events, transportation gates, or tourist sites, ticketing systems can incorporate barcode or QR code scanning. This improves security, stops unauthorised access, and speeds up the admissions process.

The computing industry benefits from a ticket booking system by using technology to automate and streamline ticketing procedures. People also gain from the system's convenience, accessibility, efficiency, and improved user experience.

## CHAPTER 2: RESEARCH METHODOLOGY

### 2.1 INTRODUCTION

In this section we will discuss the various tools, methods and phases used to create the system from planning all the way to implementation. Any system to be created must follow a certain approach to be developed efficiently. It will also include the methods used to gather information.

### 2.2 METHODOLOGY

The System Development Life Cycle (SDLC) is a structured approach used to guide the development of software systems and applications. It consists of several phases, each with specific tasks and deliverables. Let's outline the SDLC methodology for a ticket reservation system:

#### 1. Planning Phase:

**Project scope:** Develop an online ticket reservation system for a theater that allows users to browse and reserve seats for different shows and events.

**Objectives:** Increase ticket sales, improve customer convenience, and streamline the reservation process.

**Requirements:** The system should support multiple events, allow users to select specific seats, handle payment transactions securely, and send confirmation emails after successful bookings.

**Feasibility Study:** Assess the technical, economic, legal, and operational aspects of the project to determine its viability.

**Project Plan:** Create a detailed plan with milestones, timelines, and resource allocation.

## 2. Analysis Phase:

**Gather Requirements:** Conduct interviews and workshops with theater staff, customers, and other stakeholders to collect detailed requirements.

**Use Cases:** Create use cases, such as "Reserve Ticket," "Browse Shows," and "Cancel Reservation," to illustrate the system's functionalities.

**Data Flow Diagram:** Develop a data flow diagram to show the flow of data and interactions between the user and the system.

**User Interface Design:** Prepare wireframes and mock-ups to visualize the user interface and seek feedback from stakeholders.

## 3. Design Phase:

**Architecture Design:** Define a scalable and secure architecture, involving front-end (user interface), back-end (reservation logic), and a database to store event details and reservations.

**Database Design:** Plan the database schema to manage events, seating arrangements, user information, and reservation records.

**Security Design:** Implement security measures, such as encryption for sensitive data and secure payment gateway integration.

**Integration:** If required, identify and plan for integrating the reservation system with other existing systems, such as a payment gateway or customer database.

#### 4. Development Phase:

**Front-end Development:** Implement the user interface using web technologies like HTML to allow users to browse events and select seats.

**Back-end Development:** Develop the reservation system's logic using appropriate programming languages (e.g. C++) to process bookings, check seat availability, and handle payment transactions.

**Database Implementation:** Set up the database and implement queries and procedures to store and retrieve event information and reservations.

**Payment Integration:** Integrate a secure payment gateway to facilitate online payments.

#### 5. Testing Phase:

**Unit Testing:** Test individual components, such as functions and modules, to ensure they work as expected.

**Integration Testing:** Verify that all the components work together seamlessly.

**System Testing:** Conduct end-to-end testing to validate the entire reservation system against the defined requirements.

**User Acceptance Testing (UAT):** Involve theater staff and selected users to test the system in a real-world environment and gather feedback.

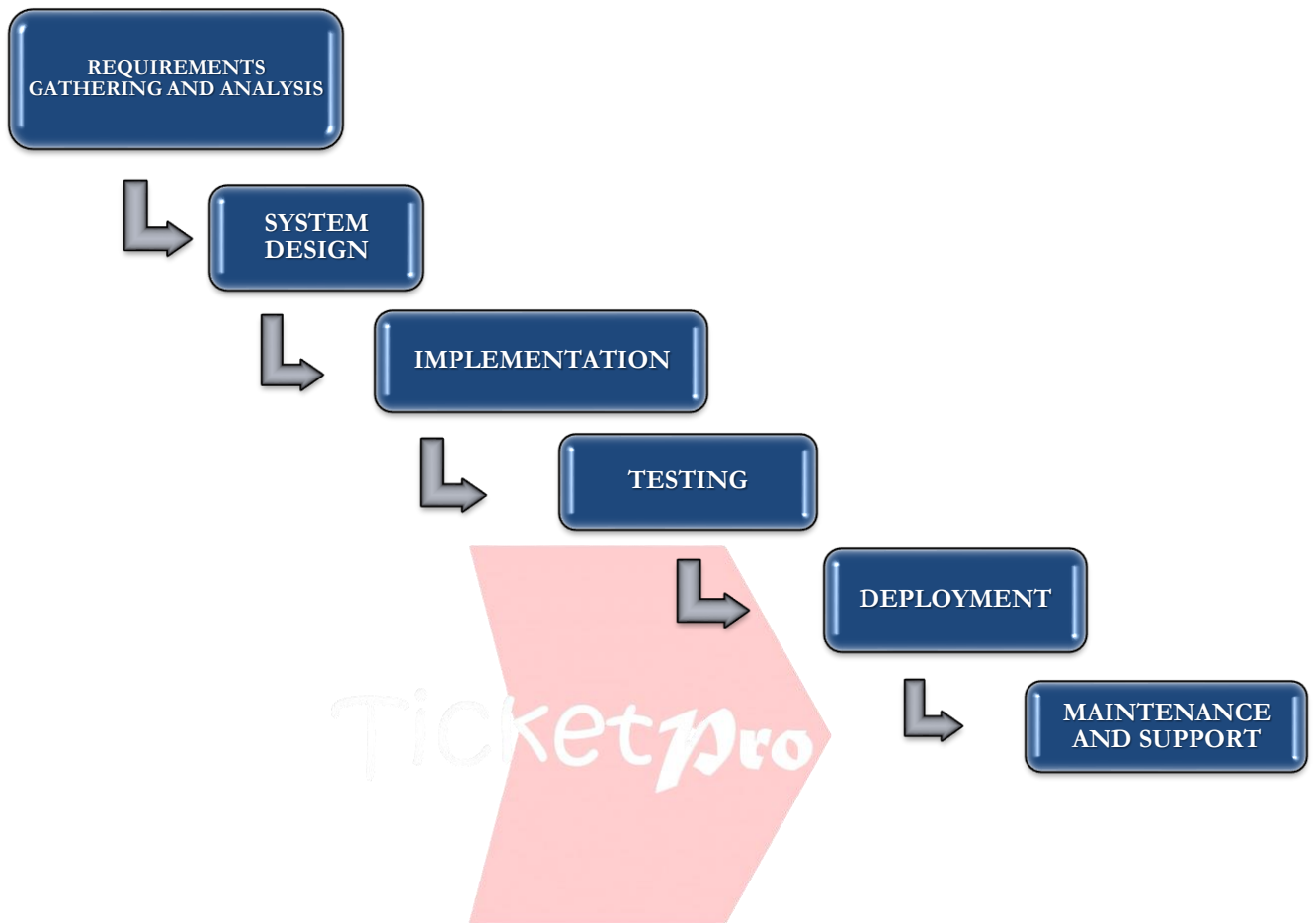
#### 6. Deployment Phase:

**Production Deployment:** Deploy the ticket reservation system on servers or cloud infrastructure for public use.

**Training:** Provide training to theatre staff for system operation and support.



# WATERFALL MODEL



## 2.3 INFORMATION GATHERING AND ANALYSIS

Methods of data collection.

**Observations:** Researchers observe and record the behavior of individuals or events in their natural settings. This method is particularly useful for studying behaviors and interactions.

**Case Studies:** In-depth analysis of a single individual, group, or entity to understand a particular phenomenon or situation in its real-life context.

**Social Media Data Mining:** Analyzing data from social media platforms to understand public sentiment, trends, and behaviors.

## 2.4 REQUIREMENTS AND SPECIFICATION

The requirements specification for a ticket reservation system outlines the functional and non-functional requirements that the system must fulfill to meet the needs of its users and stakeholders. Below is a sample of requirements for a ticket reservation system:

### 1. User Management:

- The system shall allow users to create accounts and log in with unique credentials.
- The system shall support different user roles, such as customers, administrators, and event organizers.
- Administrators shall have the ability to manage user accounts and access levels.

### 2. Event Management:

- The system shall allow event organizers to create and manage events, including event details (title, description, date, time, location, etc.).
- Event organizers shall be able to set the ticket types, availability, and pricing for each event.

### 3. Ticket Reservation:

- Customers shall be able to browse upcoming events and view event details and available ticket types.
- Customers shall be able to select seats or ticket quantities for the desired event.
- The system shall display the available seats and indicate already reserved seats.
- Customers shall be able to add selected tickets to their shopping cart.



#### **4. Cart and Checkout:**

- The system shall maintain a shopping cart for each customer, allowing them to add, remove, or modify ticket selections before checkout.
- Customers shall have the option to review the total cost before proceeding to payment.
- The system shall support secure online payment processing through various payment methods.
- Customers shall receive a confirmation email after successful payment.

#### **5. Seat Allocation:**

- The system shall automatically update seat availability upon successful reservation or release of tickets.
- Seat allocation shall follow a first-come-first-served basis, or customers may choose seats from a seating chart.

#### **6. Order Management:**

- Administrators shall be able to view and manage all reservations and bookings.
- Administrators shall have the authority to cancel or modify reservations, if necessary.

#### **7. Notification and Reminders:**

- The system shall send confirmation emails to customers upon successful reservation and payment.
- Customers shall receive reminders or notifications before the event date.

#### **8. Reporting and Analytics:**

- The system shall provide reports on ticket sales, event attendance, and revenue for event organizers and administrators.

#### **9. Security and Privacy:**

- The system shall implement robust security measures to protect user data and payment information.
- Personal customer information shall be kept confidential and comply with relevant data protection regulations.

#### **10. User Interface and Accessibility:**

- The user interface shall be intuitive, user-friendly, and accessible across different devices and

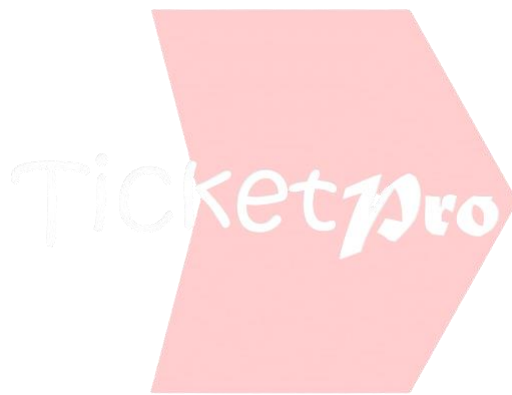
browsers.

### **11. Performance and Scalability:**

- The system shall be able to handle a large number of concurrent users and scale to accommodate increased demand during peak times.

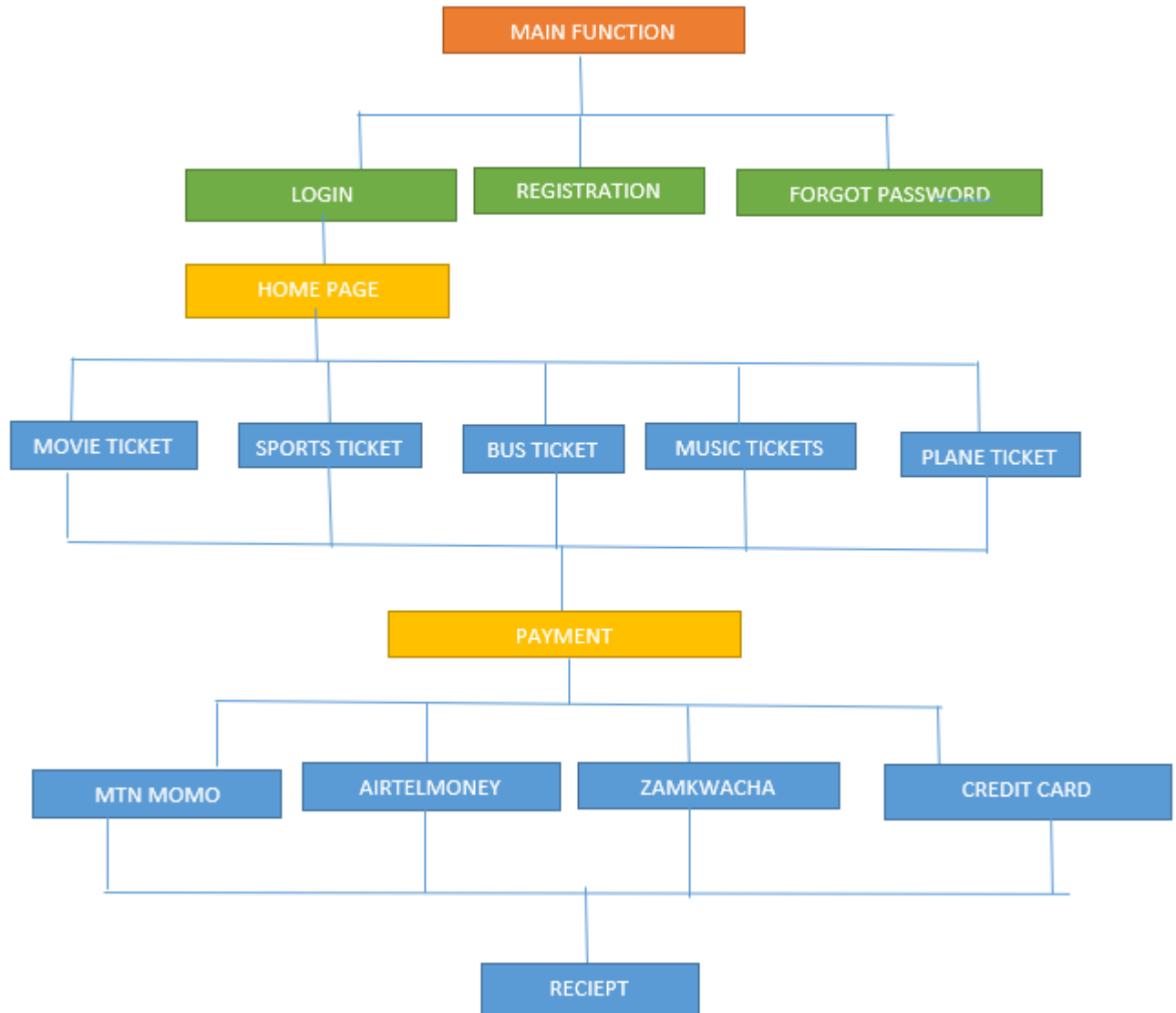
### **12. Integration:**

- The system may need to integrate with external services for payment processing and other functionalities.



# SYSTEM ANALYSIS

## STRUCTURED PROGRAMMING DESIGN



## **STRUCTURED PROGRAMMING APPROACH**

The approach that was used in developing the system was structured programming, as it had many advantages. Structured programming is a programming paradigm that is aimed at improving the programs clarity and development time by making extensive use of subroutines such as loops and decision structures.

The team mainly worked using modular programming when developing the system. The system was broken down into modules that were then divided amongst the programmers. The system consists mainly of functions that perform different specific task in the system. The functions used were as follows:

### **Main function**

This is the entry point of the system and calls three functions to initiate the use of the system. The programming language used strictly requires the main function to be called at the start of the execution. The functions called under the main include; Login function, Registration function and forgot password function.

### **Registration function**

This function is called under the main function. It is the first function to be called when a new user uses the system. Its main task is to capture the details of the user, details such as first name, last name, phone number, preferred user name and secret password to be used in the system.

The first name, last name and phone number are mainly used for the generation of the ticket that the user purchases. The user name and password are used to enter the main page of the system where the user interacts with the system by making choices of what they want from the system.

The details captured in this function are written to a file that will be used as a storage for all the essential information required by the system to print out a ticket for the user and also login details.

### **Forgot password function**

This function helps the user to retrieve their account in case of a scenario where they forget their password. It uses the data stored in the file which stores the data that a user registers with. The function retrieves the most important data from the file (username and password).

After retrieving the data the function will request the user to enter the username they last remember

registering with. When the user enters a username that matches with one of the usernames in the file then the password will be displayed for them with a message stating that their account has been found. In case where the user enters a username that does not match any of the usernames in file, the system will display a message that states that the account with such a username has never been registered.

### **Login function**

This function leads to the home page of the system. For a user to log in, they must have registered themselves. In order to log in using the details that the user registered with, we use a file to retrieve the username and the password. If the user enters the username with a corresponding password, then access will be granted, else the system will not give access and request the user to try again.

When access is granted the login function calls the home page function.

### **Home page function**

The home page function is the introduction to what is contained in the system (what kind of tickets are available). The home page function calls five other functions depending on what kind of ticket the user wants, the appropriate function will be called. For example if the user wants to purchase a movie ticket then when they select the movie ticket option the home page will call the movie ticket function.

The main task of the home page function is to display the options the user can choose from and call the appropriate function for that option. The following are the functions that are called by the homepage:

### **Movie ticket function**

This function is called by the homepage function when the user chooses to purchase a movie ticket. It displays the movies that are available for purchase, from which the user can choose from.

### **Sports ticket function**

This function is called by the homepage function when the user chooses to purchase a sports ticket. It displays the available games for purchase, from which the user can choose from.

### **Bus ticket function**

This function is called when the user chooses the bus ticket from the options displayed by the homepage. It displays the available buses as well as the destinations, from which the user can choose from.

### **Music ticket function**

This function is called by the homepage function when the user chooses to a music ticket. It displays the available music concerts available, from which the user can choose from.

### **Plane ticket function**

This function is called by the homepage function when the user chooses to purchase a plane ticket. It displays the available plane tickets for purchase, from which the user can choose from.

All the five functions that are called in the homepage function call the payment function after the user chooses the specific ticket to purchase.

### **Payment function**

This function is called after the user chooses the specific ticket they are purchasing. It can be called by all the five functions that are called by the homepage function. In return the payment function calls four different functions depending on the type of payment the user prefers.

### **MTN MoMo function**

This function uses the MTN MoMo services to make payment for the ticket being purchased. It will request the user to enter the number they are making payment from and request for the user to key in their MTN MoMo PIN.

### **Airtel Money function**

This function uses the Airtel money services to make payment for the ticket being purchased. It will request the user to enter the number in which the payment is being made from and request the user to enter their Airtel money PIN.

### **ZamKwacha function**

This function uses the ZamKwacha services to make the payment for the ticket being purchased. It will request for the user to enter the number from which the payment will be made from and request for the PIN.

### **Credit card**

This function uses the customer's credit card details to make payment for the ticket being purchased. It will request for the bank account number and PIN to make the payment.

All the four function can call the receipt function for the final output of the system.

### **Receipt function**

This is the last function that is called in the system. This function uses all the choices the user has been choosing to generate the appropriate ticket that has been purchased.

It includes:

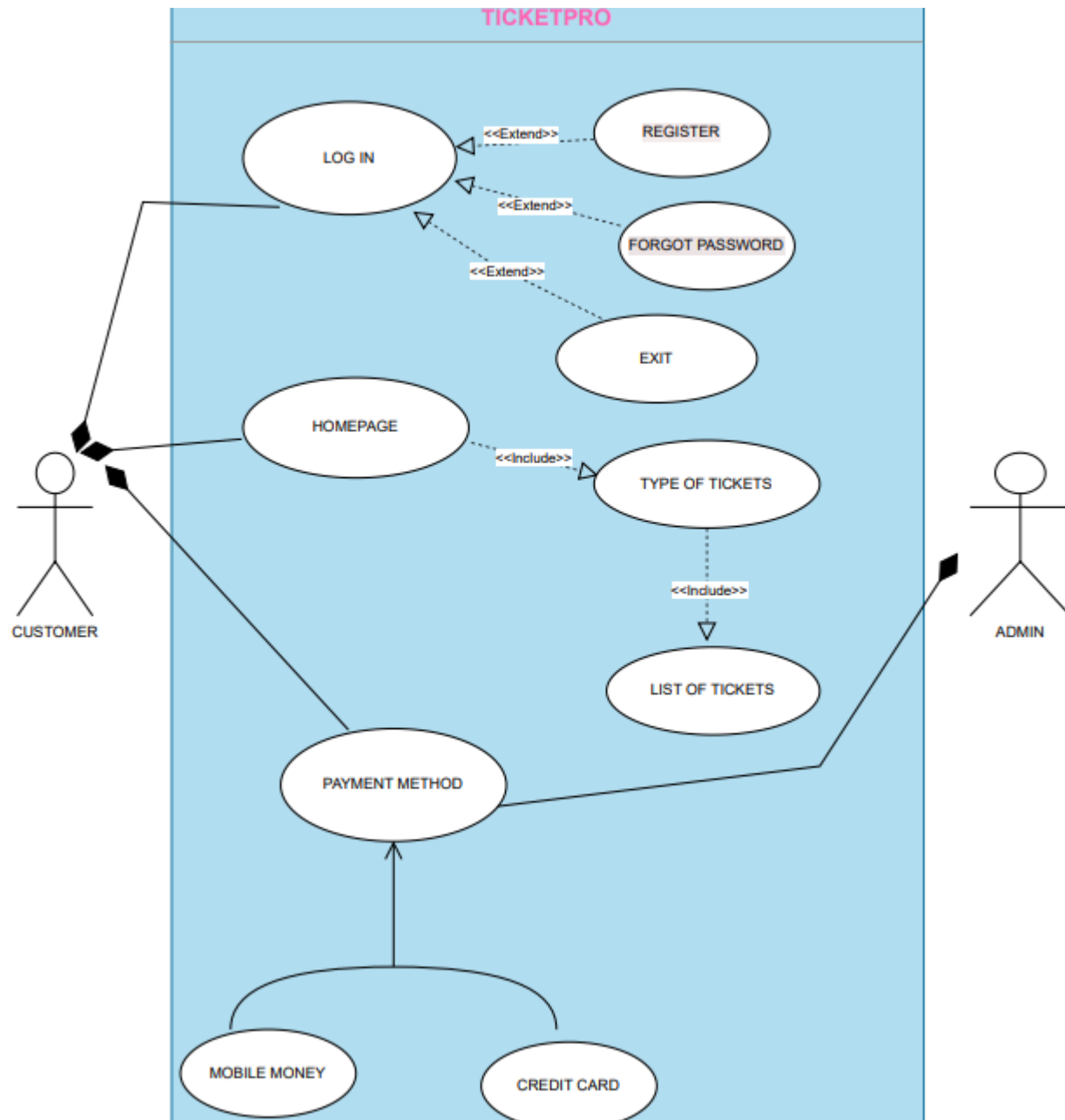
- Full names of the customer
- Phone number
- Ticket type
- Ticket name
- Price
- Method of payment
- Time and date the ticket was generated
- Status (whether paid or not)

At the end of the generation of the ticket the user can chose to log out or purchase another ticket.

## UML DESIGN

UML design is the shortest form of “Unified Modelling Language”. The purpose of this modelling language is to visualize the design of the system. For that we have chosen “Use Case Diagram.”

### *Use Case Diagram*



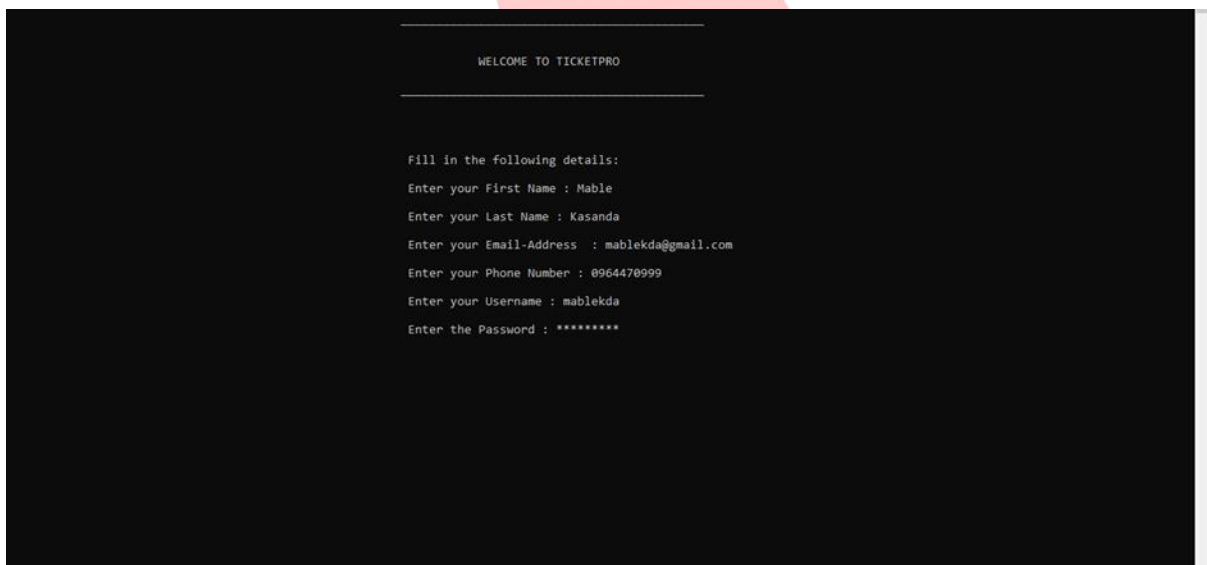


# PHYSICAL DESIGN

## *HOME PAGE*



## *REGISTRATION*



## LOGIN

```

Welcome to TicketPro

Please enter your Username and Password :

USERNAME: mablekda
PASSWORD: *****

ACCESS GRANTED!_

```

## TICKET AVAILABILITY

```

HOMEPAGE

What type of Ticket would you like to Book?

1. Movie Ticket
2. Sports Ticket
3. Bus Ticket
4. Music Concert Ticket
5. Plane Ticket

Enter your choice: 1

```

## PAYMENT

```

TICKETPRO

What Payment method would you like to proceed with the Transaction?

1. MTN Mobile Money
2. Airtel Money
3. Zam Kwacha
4. Credit Card

Enter choice: 1

```

## RECEIPT

```
TICKETPRO

*****
***** TICKETPRO RECEIPT *****
*****
**
** Name of Customer: Mable Kasanda
** Phone Number: 0964478999
** Email Address: mablekudaggmail.com
** Ticket Type: Movie Ticket
** Ticket Name: Extraction 2
** Price: K130
** Time: 12:28:14
** Date: 2023-08-08
**
** Payment Method: MTN Mobile Money
** STATUS: PAID
**
*****
*****

THANK YOU FOR USING TICKETPRO!

Press c to book another ticket
Press q to log out of your account
Enter your choice: _
```

## CHAPTER 3: IMPLEMENTATION

### 3.1 INTRODUCTION

Concocting and executing an all-encompassing ticket procuring system for movies, concerts, sports matches, planes, and buses is a labyrinthine endeavour that entails manifold constituents and phases. Below lies a panoramic survey of the enactment minutiae, along with the obligatory formation, constituents, and directives.

### 3.1 SYSTEM IMPLEMENATION

#### *Project Overview: Ticket Procuring System*

#### 1. System Design and Architecture:

- Select a stratified architecture, akin to a client-server blueprint.
- Employ Object-Oriented Programming (OOP) principles to fashion classes for movies, concerts, sports matches, planes, buses, users, and bookings.
- Enforce a database to stow information pertaining to events, users, and bookings.

## **2. System Components:**

- Backend conductor (utilizing C++).
- Database.
- Frontend client.
- User authentication and session supervision.
- Payment Gateway System.

## **3. System Setup for Development:**

- Operating System: Windows.
- Development Apparatus: C++ Compiler (e.g., g++, Visual C++) or your favoured IDE.
- Database Management System: Text File (.txt).
- GUI Framework: Qt (if employing a GUI).

## **4. Steps to Achieve the Project:**

### **4.1 Backend Development:**

- Enact classes for events (movies, concerts, sports matches, planes, buses), users, and bookings.
- Engineer APIs for CRUD operations on events, user management, and bookings.
- Execute business logic for inspecting seat availability, making bookings, and generating booking confirmations.

### **4.2 Database Development:**

- Formulate a relational database schema for events, users, and bookings.
- Enact stored procedures for intricate operations.

### **4.3 Frontend Development:**

- Design user-friendly forms for event selection, seat booking, and user registration.
- Execute user authentication and session supervision.
- Incorporate with backend APIs for data retrieval and manipulation.

### **4.4 Integration and Testing:**

- Probe the system constituents autonomously and subsequently converge them.
- Implement unit testing, integration testing, and end-to-end testing.

- Detect and rectify any glitches or predicaments.

## **5. Deployment Configuration:**

- Select a server hosting solution (e.g. web hosting provider).
- Install the requisite software on the server (database, backend server).
- Deploy the backend and database to the server.

## **6. Installation Procedure:**

- Establish the database by importing schema and initial data.
- Deploy the backend code to the server.
- Configure the frontend to connect to the backend API endpoints.

## **7. User Instruction:**

- Craft user documentation elucidating how to utilize the system.
- Conduct training sessions or proffer video tutorials if exigent.

## **8. Testing of the System:**

- Conduct exhaustive testing to ensure the system is operational and user-friendly.
- Test sundry scenarios, such as booking events, managing bookings, and user registration.

### **8.1. Test Case 1**

- ♣ Test Title: Login.
- ♣ Test Procedures: Type username & password.
- ♣ Test Data: Username & password must be correct. Otherwise system will show error “Incorrect username or password. Please try again”
- ♣ Expected Result: It will redirect to login page.

### **8.2. Test Case 2**

- ♣ Test Title: Registration
- ♣ Test Procedures: Enter Registration option.
- ♣ Test Data: Input customer’s information and click on the ENTER button.
- ♣ Expected Result: It will register new customer.

### 8.3. Test Case 3

- ♣ Test Title: Forgot Password
- ♣ Test Procedures: Enter the Forgot Password option.
- ♣ Test Data: Input User's last known password and click on the ENTER button.
- ♣ Expected Result: If username is valid, the system will allow access and display password.

### 9. Troubleshooting Directives:

- Compose a troubleshooting guide for commonplace issues users might encounter.
- Provide contact information for technical support.

### 10. Directives for Further Work:

- Contemplate appending features like payment integration, event recommendations, and reviews.
- Optimize performance for handling a voluminous number of users and events.
- Enforce security measures to safeguard user data and inhibit unauthorized access.

### Software Employed:

C++ for backend development. Text File (.txt) for the database.

## 3.4 RESULTS

```

TICKETPRO

*****
***** TICKETPRO RECEIPT *****
*****
**
** Name of Customer: Mable Kasanda
** Phone Number: 0954479999
** Email Address: mablekda@gmail.com
**
** Ticket type: Movie Ticket
** Ticket Name: Extraction 2
** Price: K130
** Time: 12:28:14
** Date: 2023-08-08
**
** Payment Method: MTN Mobile Money
** STATUS: PAID
**
*****
*****

THANK YOU FOR USING TICKETPRO!

Press c to book another ticket
Press q to log out of your account
Enter your choice: _

Alternative actions
```

