Jomo Kenyatta University of Agriculture and Technology

BSc. Computer Science

Design and Implementation of Computer Applications

# Flight Booking and Ticketing Application Proposal

## Group members

* Karugu Anthony Muturi – SCT211-0016/2022
* Njoroge Kimberly Wangari – SCT211-0060/2022
* Teddy Muli – SCT211-0023/2022
* Britney Knight – SCT211-0046/2022
* Collins Omollo – SCT211-0021/2022
* Eunice Nyaboke – SCT211-0544/2022

## Introduction.

Welcome to the proposal for our Flight Ticket Booking Application. In today's fast-paced world, efficient travel solutions are needed. Our application aims to revolutionize the flight booking experience by providing a user-friendly platform with advanced and useful features. This proposal outlines the functionality, technical requirements, and schedule for the development of our application. The target audience of the application is travellerswho want to take flights.

## Functional Requirements

1. Search Functionality: Users should be able to search for flights using the destination filter. The application will display available flights based on the destination entered by the user.

2. Seat Selection: Users should have the ability to choose seats from a list of available seats. Unavailable seats should also be displayed to prevent double booking.

3. Cost Calculation: The application must calculate the final cost of the booking. It should take into account factors such as seat class, child discounts, and any additional services selected by the user.

4. Booking Management: Users should be able to cancel a booking if needed. The application should provide a simple process for cancellation and update the booking status accordingly.

5. User Information Storage: The application should store user information securely. This includes personal details required for booking, such as name, contact information, and passport number.

6. Booking Retrieval: Users should be able to retrieve their bookings by providing their passport number. The application should display the relevant booking details associated with the provided passport number.

## Nonfunctional Requirements:

1. Performance: The application should be responsive and provide quick search results and booking confirmations.

2. Security: User data must be stored securely to protect personal information. This includes encryption of sensitive data and implementing secure authentication methods.

3. Usability: The application should have an intuitive user interface that is easy to navigate. Users should be able to complete booking processes with minimal effort.

4. Reliability: The application should be reliable and available whenever users need to make bookings or access their information.

5. Scalability: The system should be able to accommodate changes in future according to the needs of the airline and its target clients.

## Hardware and Software Requirements:

Hardware:

- Minimum Processor: Intel Core i3 or equivalent

- Minimum RAM: 4GB

- Hard Disk Space: 900MB

- Display: 1280x768 resolution or higher

Software:

- Operating System: Windows 7 or later

- Visual Studio IDE (Integrated Development Environment)

- Microsoft SQL Server for database management

- .NET Framework

## Schedule of the Project:

Week 1: Proposal submission. Define project scope, objectives, and requirements.

* Application design phase including database, flowcharts, and sequence diagrams.

Week 2: Implementation phase. Development the application and its features.

Week 3: Testing and debugging phase. Perform comprehensive testing to ensure functionality and resolve any issues.

Final Week: Presentation preparation. Prepare a demonstration of the application's features and functionalities for stakeholders.

Throughout the project, regular meetings will be held to track progress, address any challenges, and ensure deadlines are met.

## Stages

Absolutely! Here's a condensed version of the stages of development, completed within four weeks:

Week 1: Proposal and Planning

- Submit proposal with project objectives and requirements.

- Define functional and non-functional requirements.

- Plan project scope, schedule, and resources.

Week 2: Design and Setup

- Design user interface wireframes and database schema.

- Set up development environment and database.

- Finalize architecture and technology stack.

Week 3: Development and Testing

- Develop frontend and backend components.

- Implement flight search, seat selection, and booking management.

- Conduct unit and integration testing to ensure functionality.

Week 4: Refinement and Presentation

- Address any bugs or issues identified during testing.

- Prepare presentation materials and demo.

- Rehearse presentation and gather feedback.

- Finalize documentation and deliver the Flight Ticket Booking Application.

The schedule will ensure the efficient completion of the project.

## Feasibility Study

1. Technical Feasibility:

- Technology Stack: Visual Basic.NET offers robust support for building desktop applications, making it suitable for developing the Flight Ticket Booking Application.

- Database Management: Microsoft SQL Server provides reliable and scalable database management capabilities, ensuring efficient storage and retrieval of user and flight data.

- Integration Capabilities: Third-party APIs can be integrated for accessing and processing payments for example from banks or Mpesa and more mobile wallets, enhancing the functionality of the application.

2. Operational Feasibility:

- User-Friendly Interface: The application will feature an intuitive user interface, simplifying the flight booking process for users of all levels of technical proficiency.

- Efficient Booking Management: With features such as seat selection, cost calculation, and booking cancellation, the application streamlines booking management tasks, improving operational efficiency and reducing hold ups at physical locations.

- Scalability: The application can be easily scaled to accommodate increased user demand and additional features as needed, ensuring long-term operational viability.

3. Economic Feasibility:

- Cost-Benefit Analysis: The development and maintenance costs of the Flight Ticket Booking Application are expected to be offset by the potential revenue generated from booking fees and commissions.

- Return on Investment: With an increasing demand for online booking services, the application has the potential to yield a favorable return on investment over time.

4. Schedule Feasibility:

- Four-Week Timeline: The proposed development schedule allows for a focused and efficient development process, ensuring the timely completion of the project.

- Phased Approach: Breaking down the project into distinct phases, including planning, design, development, testing, and presentation, helps manage time effectively and mitigate risks.

5. Market Feasibility:

- Growing Demand: The increasing trend towards online booking and the convenience of self-service platforms indicate a favorable market environment for the Flight Ticket Booking Application.

- Competitive Landscape: While there are existing flight booking platforms, the application's unique features and user-centric design offer a competitive advantage in the market.

## Design

### Interface Design

Bookings

Booking

Passport Number

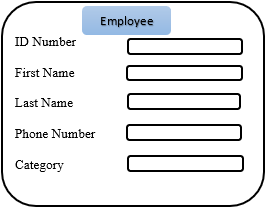
Flight Number

Seat Number

Cancel

Confirm

Employee



Flight Information

Flights

Flight Number

To

From

Price

Date

Number of seats

Classes

Class seats

Employee ID

Cancel

Confirm

Customer Section

Customer

Passport Number

First Name

Last Name

Phone Number

Date of Birth

### Database Design

ENTITY RELATIONSHIP DIAGRAM

Entities

1. Customers
2. Flights
3. Employees
4. Bookings
5. Customers

* Passport number – Primary Key
* First name
* Last Name
* Phone Number
* Date of birth

1. Flights

* Flight number – Primary Key
* To
* From
* Price
* Date
* No of seats
* Classes
* Class seats
* Employee ID – Foreign Key

1. Employees

* ID Number – Primary Key
* First Name
* Last Name
* Phone number
* Category

1. Bookings

* Passport Number – Foreign Key
* Flight Number – Foreign Key
* Seat No.

Employees

Customers

Flights

Bookings

### Program Design

1. Requirements Gathering

Functional Requirements: Include searching for flights, booking flights, managing bookings (cancel or change), user registration and login, payment processing, and user feedback.

Non-functional Requirements: Focus on performance (response times), scalability (to handle growth in users), reliability (uptime), and security (data protection and privacy).

2. System Analysis and Design

Use Case Diagrams: Identify actors (e.g., passengers, guest users, system administrator) and their interactions with the system through use cases.

Activity Diagrams: Illustrate the flow of activities for critical processes like booking a flight or managing a reservation.

3. Database Design

Entity-Relationship Diagram (ERD): Design an ERD to model entities such as Users, Flights, Bookings, Payments, and their relationships.

Normalization: Ensure the database design is normalized to eliminate redundancy and ensure data integrity.

Database Schema: Define tables, fields, data types, and constraints based on the ERD.

4. Application Architecture

Front-end: Design the user interface using Visual Basic.

5. User Interface Design

Wireframes: Create wireframes for key screens such as the homepage, search results, booking form, user dashboard, and admin panel.

Prototyping: Develop interactive prototypes to test and refine the user interface and experience.

6. Implementation

Front-end Development: Implement the front-end based on the designs using Visual Basic.

7. Testing

Unit Testing: Write and run unit tests for individual components and functions.

Integration Testing: Test the integration points between different parts of the system, including external integrations.

System Testing: Conduct system-wide testing to ensure the entire application works as expected.

User Acceptance Testing (UAT): Have end-users test the system to validate that it meets their needs and requirements.

8. Deployment and Maintenance

Deployment Strategy: Plan the deployment of the system to production, including any necessary infrastructure setup (e.g., servers, and databases).

Maintenance Plan: Establish processes for monitoring the system, fixing bugs, and implementing updates and new features based on user feedback.

9. Security and Compliance

Security Measures: Implement security measures like HTTPS, data encryption, secure authentication, and authorization to protect user data and transactions.

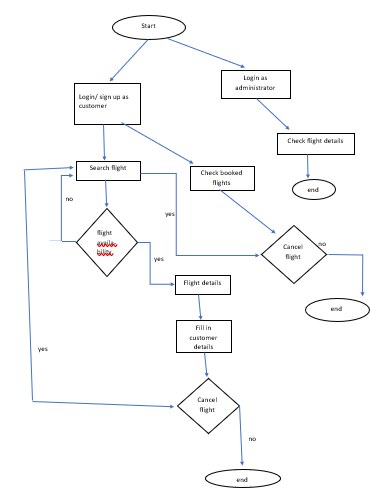
Compliance: Ensure compliance with relevant laws and regulations, such as GDPR for user data protection.

Documentation and Training

Documentation: Prepare comprehensive documentation for the system, including API documentation, user manuals, and system administration guides.

Training: Provide training for end-users and administrators on using the system effectively.

### Flowchart



### Sequence diagram

