**Flight Reservation Management System**

### Submitted By

|  |  |
| --- | --- |
| **Student Name** | **Student ID** |
| Md Salauddin Sarkar Pappo | 241-15-753 |
| Md. Arman Hossen Ripon | 241-15-883 |
| S M Ajmain Bin Barkat | 241-15-620 |

**LAB PROJECT REPORT**

This Report Presented in Partial Fulfillment of the course **CSE124: Data Structure Lab, Computer Science and Engineering Department**

### DAFFODIL INTERNATIONAL UNIVERSITY

**Dhaka, Bangladesh**

**December 8, 2024**

## **DECLARATION**

We hereby declare that this lab project has been done by us under the supervision of **Md. Rasel Sarker**, **Lecturer**, Department of Computer Science and Engineering, Daffodil International University. We also declare that neither this project nor any part of this project has been submitted elsewhere as lab projects.

**Submitted To:**

**Md. Rasel Sarker**

Lecturer

Department of Computer Science and Engineering  
Daffodil International University

**Submitted by**

|  |  |
| --- | --- |
| Md. Arman Hossen Ripon Student ID:241-15-883  Dept. of CSE, DIU | |
| S M Ajmain Bin Barkat Student ID: 241-15-620  Dept. of CSE, DIU | Md Salauddin Sarkar Pappo  Student ID: 241-15-753  Dept. of CSE, DIU |

# 

## **COURSE & PROGRAM OUTCOME**

The following course have course outcomes as following:

Table 1: Course Outcome Statements

|  |  |
| --- | --- |
| **CO’s** | **Statements** |
| CO1 | **Define** and **Relate** classes, objects, members of the class, and relationships among  them needed for solving specific problems |
| CO2 | **Formulate** knowledge of object-oriented programming and C in problem solving |
| CO3 | **Analyze** Unified Modeling Language (UML) models to **Present** a specific problem |
| CO4 | **Develop** solutions for real-world complex problems **applying** OOP concepts while  evaluating their effectiveness based on project standards. |

Table 2: Mapping of CO, PO, Blooms, KP and CEP

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **CO** | **PO** | **Blooms** | **KP** | **CEP** |
| CO1 | PO1 | C1, C2 | KP3 | EP1, EP3 |
| CO2 | PO2 | C2 | KP3 | EP1, EP3 |
| CO3 | PO3 | C4, A1 | KP3 | EP1, EP2 |
| CO4 | PO3 | C3, C6, A3,  P3 | KP4 | EP1, EP3 |

# The mapping justification of this table is provided in section 4.3.1, 4.3.2 and 4.3.

Course & Program Outcome [ii](file:///E:\Downloads\Telegram%20Desktop\5_6210647448751183336.docx#_bookmark1)

1. Introduction [1](file:///E:\Downloads\Telegram%20Desktop\5_6210647448751183336.docx#_bookmark2)
   1. Introduction [1](file:///E:\Downloads\Telegram%20Desktop\5_6210647448751183336.docx#_bookmark3)
   2. Motivation [1](file:///E:\Downloads\Telegram%20Desktop\5_6210647448751183336.docx#_bookmark4)
   3. Objectives [1](file:///E:\Downloads\Telegram%20Desktop\5_6210647448751183336.docx#_bookmark5)
   4. Feasibility Study [1](file:///E:\Downloads\Telegram%20Desktop\5_6210647448751183336.docx#_bookmark6)
   5. Gap Analysis [1](file:///E:\Downloads\Telegram%20Desktop\5_6210647448751183336.docx#_bookmark7)
   6. Project Outcome [1](file:///E:\Downloads\Telegram%20Desktop\5_6210647448751183336.docx#_bookmark8)
2. Proposed Methodology/Architecture [2](file:///E:\Downloads\Telegram%20Desktop\5_6210647448751183336.docx#_bookmark9)
   1. Requirement Analysis & Design Specification [2](file:///E:\Downloads\Telegram%20Desktop\5_6210647448751183336.docx#_bookmark10)
      1. Overview [2](file:///E:\Downloads\Telegram%20Desktop\5_6210647448751183336.docx#_bookmark11)
      2. UI Design [2](file:///E:\Downloads\Telegram%20Desktop\5_6210647448751183336.docx#_bookmark13)
   2. Overall Project Plan [2](file:///E:\Downloads\Telegram%20Desktop\5_6210647448751183336.docx#_bookmark14)
3. Implementation and Results [3](file:///E:\Downloads\Telegram%20Desktop\5_6210647448751183336.docx#_bookmark15)
   1. Implementation [3](file:///E:\Downloads\Telegram%20Desktop\5_6210647448751183336.docx#_bookmark16)
   2. Performance Analysis [3](file:///E:\Downloads\Telegram%20Desktop\5_6210647448751183336.docx#_bookmark17)
   3. Results and Discussion [3](file:///E:\Downloads\Telegram%20Desktop\5_6210647448751183336.docx#_bookmark18)
4. Engineering Standards and Mapping [4](file:///E:\Downloads\Telegram%20Desktop\5_6210647448751183336.docx#_bookmark19)
   1. Impact on Society, Environment and Sustainability [4](file:///E:\Downloads\Telegram%20Desktop\5_6210647448751183336.docx#_bookmark20)
      1. Impact on Life [4](file:///E:\Downloads\Telegram%20Desktop\5_6210647448751183336.docx#_bookmark21)
      2. Impact on Society & Environment [4](file:///E:\Downloads\Telegram%20Desktop\5_6210647448751183336.docx#_bookmark22)
      3. Ethical Aspects [4](file:///E:\Downloads\Telegram%20Desktop\5_6210647448751183336.docx#_bookmark23)
      4. Sustainability Plan [4](file:///E:\Downloads\Telegram%20Desktop\5_6210647448751183336.docx#_bookmark24)
   2. Project Management and Team Work [4](file:///E:\Downloads\Telegram%20Desktop\5_6210647448751183336.docx#_bookmark25)
   3. Complex Engineering Problem [4](file:///E:\Downloads\Telegram%20Desktop\5_6210647448751183336.docx#_bookmark26)
      1. Mapping of Program Outcome [4](file:///E:\Downloads\Telegram%20Desktop\5_6210647448751183336.docx#_bookmark27)
      2. Complex Problem Solving [4](file:///E:\Downloads\Telegram%20Desktop\5_6210647448751183336.docx#_bookmark28)
      3. Engineering Activities [5](file:///E:\Downloads\Telegram%20Desktop\5_6210647448751183336.docx#_bookmark30)
5. Conclusion [6](file:///E:\Downloads\Telegram%20Desktop\5_6210647448751183336.docx#_bookmark32)
   1. Summary [6](file:///E:\Downloads\Telegram%20Desktop\5_6210647448751183336.docx#_bookmark33)
   2. Limitation [6](file:///E:\Downloads\Telegram%20Desktop\5_6210647448751183336.docx#_bookmark34)
   3. Future Work [6](file:///E:\Downloads\Telegram%20Desktop\5_6210647448751183336.docx#_bookmark35)

**Chapter 1**

This chapter outlines the purpose, motivation, and objectives of the Flight Reservation Management System. It also discusses the feasibility study, gap analysis, and the potential outcomes of the project.

### Introduction

Manual flight reservation processes are prone to errors and inefficiencies. This project aims to automate processes like flight booking, user management, and flight data updates to enhance operational efficiency and user convenience.

### Motivation

The project was inspired by the need for a simple yet efficient tool to manage flight reservations. Current systems often cater to large-scale operations, leaving smaller setups or personal use cases underserved. This project provides a practical solution that improves accuracy and efficiency while offering a hands-on opportunity to develop programming skills. It is designed to benefit smaller travel agencies or individuals by addressing common reservation management challenges.

### Objectives

1. Automate core flight reservation operations such as booking, flight management, and user account handling.

2. Minimize manual errors and ensure accurate, secure record-keeping.

3. Create a user-friendly system that is easy to implement and maintain.

### 1.4 Feasibility Study

Managing flight reservations manually is common but often leads to inefficiencies such as double bookings, mismanagement of flight schedules, and errors in record-keeping. While enterprise-level solutions exist, they are often too complex or expensive for smaller applications. This project explores creating a simple, accessible system that addresses these challenges effectively. By focusing on essential features like booking, seat availability, and flight updates, the system ensures practicality and affordability.

### Gap Analysis

Current flight reservation solutions often prioritize large-scale operations, leaving smaller entities or custom use cases underserved. This project fills this gap by developing a straightforward, efficient system that prioritizes simplicity, usability, and cost-effectiveness.

### Project Outcome

The project will deliver a functional Flight Reservation Management System capable of managing user accounts, booking tickets, and updating flight details. The system will ensure efficiency, accuracy, and an improved user experience.

**Chapter 2**

**Proposed Methodology/Architecture**

This chapter discusses the requirements, design specifications, system architecture, UI design, and overall project plan for the Flight Reservation Management System.

### Requirement Analysis & Design Specification

#### **Overview** The Flight Reservation Management System requires a structured approach to ensure functionality, efficiency, and user-friendliness. This section outlines the necessary components, including software requirements, user needs, and system functionalities.

* **Software Requirements**:

1. Programming Language: C
2. Compiler: GCC or any compatible C compiler
3. Operating System: Windows/Linux
4. Text editor or IDE: Code::Blocks, Visual Studio, or similar

* **User Needs**:

1. Users need to view available flights, book tickets, and manage bookings.
2. Admins need to manage flight details, including adding new flights and updating existing records.

* **System Functionalities**:

1. **Admin**: Login, add/update/view flights.
2. **User**: Signup, login, book tickets, update bookings, view available flights.
3. File handling for persistent storage of user data, flight details, and booking records.

#### **UI Design** The user interface is text-based, designed for simplicity and ease of use. It provides:

1. **Menu-driven navigation**: Clear options for both admins and users.
2. **Prompts and validations**: Ensures accurate input and guides the user through each operation.
3. **Console output**: Displays flight information, booking summaries, and error messages in an organized manner

**2.2** **Overall Project Plan**  
  
The project is divided into the following key phases:

1. **Requirement Gathering:**
2. Identify the functional requirements for user and admin roles.
3. Define the features for flight management, user account handling, and ticket booking.
4. **Design and Development:**

Build the core modules:

1. **Flight Management**: Manage flights using linked lists for dynamic data storage.
2. **User Management**: Enable user signup, login, and file-based data persistence.
3. **Booking System**: Allow ticket booking, update seat availability, and calculate costs.

Implement file handling for user, flight, and booking data.

1. **Testing and Debugging:**
   * + 1. Test all core functionalities such as booking, flight updates, and user authentication.
       2. Address edge cases like invalid inputs, overbooking, or file access errors.
2. **Final Deployment:**
3. Deliver a fully functional Flight Reservation Management System.
4. Ensure the system is stable, with clear instructions for compilation and execution.

**Chapter 3**

# **Implementation and Results**

This chapter presents the implementation details, analyzes the system's performance, and discusses the results achieved during the project.

### Implementation The Flight Reservation Management System was implemented using C programming. The system follows a modular design, with separate functions for user management, flight management, and ticket booking. The program leverages dynamic data structures like linked lists for flight records and a stack for tracking admin login attempts. Persistent data storage is achieved through file handling, ensuring all user, flight, and booking data remain accessible across program executions. The system provides a text-based interface that facilitates interactive operations for both administrators and users.

#### **Key Features:**

1. **Admin Functions**:
   1. Add, view, and update flight details.
   2. Maintain flight availability and seat pricing.
2. **User Functions**:
   1. Signup and login for account creation and access.
   2. View available flights, book tickets, and update existing bookings.
   3. Calculate and display total ticket costs before finalizing bookings.

### Performance Analysis The system effectively handled core operations like flight booking, seat updates, and account management. Invalid inputs, such as selecting non-existent flights or overbooking, were managed with clear error messages, ensuring user-friendliness. Leveraging linked lists, the system provided fast operations like flight search and seat availability checks, while file handling ensured reliable data storage without performance issues. Stability was maintained across various test scenarios, including concurrent bookings and flight updates, with robust error handling preventing crashes during file I/O or invalid inputs.

### Results and Discussion The project successfully achieved its primary objective of developing an efficient and user-friendly Flight Reservation Management System. The system automates core functionalities, reducing manual errors and improving productivity for both users and administrators.

### Key Achievements:

### Simplified ticket booking and flight management operations.

### Improved accuracy in handling flight schedules and user bookings.

### Provided persistent data storage for user accounts, flights, and booking records.

**Limitations:**

* + 1. The text-based interface, while functional, could be improved with a graphical user interface (GUI) for enhanced usability.
    2. Passwords are stored in plain text, requiring improved security measures like encryption.

**Future Enhancements:**

* + 1. **Integration with Online Platforms:** Enable booking and management through web or mobile interfaces.
    2. **Advanced Features:** Include dynamic seat allocation, booking cancellation, and real-time flight availability updates.
    3. **Enhanced Security:** Use hashing algorithms to securely store passwords and protect user data.

**Chapter 4**

# **Engineering Standards and Mapping**

This chapter discusses the integration of engineering standards into the project and evaluates its societal, environmental, and sustainability impact. It also addresses project management, complex problem-solving, and engineering activities with appropriate mappings and justifications.

### Impact on Society, Environment and Sustainability

#### **Impact on Life:** The project enhances travel experiences by providing an efficient flight reservation system, improving convenience and accessibility for users. It reduces manual processes, ensuring accurate and timely bookings.

#### **Impact on Society & Environment:** The system promotes digital transformation by reducing reliance on manual methods, minimizing paperwork, and contributing to a greener environment. By streamlining operations for users and administrators, it improves societal efficiency and accessibility.

#### **Ethical Aspects:** Ethical considerations include data privacy and secure handling of sensitive user information such as account credentials and booking details. The project emphasizes equitable access to services and the integrity of financial transactions related to ticket bookings.

#### **Sustainability Plan:** The project adopts scalable and modular technology to ensure long-term usability and adaptability. Its minimal reliance on physical resources contributes to a low environmental footprint while fostering economic and social sustainability by improving the efficiency of travel operations.

### Project Management and Team Work

This section highlights the project’s resource management and decision-making process. The system was developed with a focus on cost-effective solutions, leveraging open-source tools to minimize expenses. Alternative approaches, such as enhanced user interfaces or database-driven backends, were considered but deferred due to budget constraints.

Collaborative efforts ensured effective resource allocation and timely completion of the project, with clearly defined roles and responsibilities. The project’s development plan incorporated iterative testing and feedback to ensure alignment with user needs and engineering standards.

### Complex Engineering Problem

#### **Mapping of Program Outcome**

This section outlines how the project's problem and solution align with the targeted Program Outcomes (POs), illustrating how the project addresses the specified objectives.

**Table 4.1: Justification of Program Outcomes**

|  |  |
| --- | --- |
| **PO’s** | **Justification** |
| PO1: Engineering Knowledge | The **Flight Reservation System** applies principles of software engineering, database management, and C programming to develop a robust platform for managing flight bookings and reservations, addressing real-world challenges in the airline industry. |
| PO2: Problem Analysis | |  | | --- | |  |  |  | | --- | | A comprehensive analysis was conducted to identify inefficiencies in the traditional flight booking process. This analysis guided the design and development of a solution that optimizes flight reservations, schedules, and customer data management to meet the needs of both customers and airlines. | |
| PO3: Modern Tool Usage | The project uses advanced C programming techniques, efficient data structures, and file management systems to implement the **Flight Reservation System**, ensuring high performance, smooth functionality, and reliability for users while managing flight reservations effectively. |

#### **Complex Problem Solving**

This section evaluates the challenges encountered in the **Flight Reservation System** project and aligns them with complex problem-solving categories. Each challenge is addressed with a detailed rationale, as shown in **Table 4.2**, which highlights the mappings and the reasoning behind solving each category effectively.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **EP1**  Dept of Knowledge | **EP2**  Range of Conflicting Requirements | **EP3**  Depth of Analysis | **EP4**  Familiarity of Issues | **EP5**  Extent of Applicable Codes | **EP6**  Extent  Of Stakeholder Involvement | **EP7**  Inter- dependence |
| The project applies advanced programming concepts in C, database management, and data structures (linked lists, arrays) to efficiently manage flight reservations and customer data.. | Balances the system's ease of use for customers with technical performance constraints, ensuring fast response times for search and booking processes without compromising data integrity. | Conducts a thorough analysis of flight reservation processes to identify inefficiencies in booking systems and streamline workflows, improving the overall experience. | Addresses common challenges in flight reservations such as data accuracy, booking conflicts, and schedule management, using proven solutions from similar systems. | Uses efficient C code and appropriate algorithms to optimize flight searches, booking management, and data storage, ensuring scalability and performance. | Actively involves users (customers and airline staff) to gather feedback for refining the system, ensuring that it meets user requirements and enhances the booking process. | Ensures smooth integration of all system modules (flight scheduling, reservations, user management) to provide a seamless and efficient experience for both customers and administrators |

**Table 4.2: Mapping with Complex Problem Solving**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **EA1**  Range of resources | **EA2**  Level of Interaction | **EA3**  Innovation | **EA4**  Consequences for society and  environment | **EA5**  Familiarity |
| The project uses programming languages, databases, and frameworks, ensuring a broad technological base. | |  | | --- | | Engages with users and stakeholders to identify requirements and refine the system. |  |  | | --- | |  | | |  | | --- | |  |  |  |  |  | | --- | --- | --- | | |  | | --- | | Introduces features like real-time flight tracking, booking management, and automated reservation updates. |  |  | | --- | |  | | | |  | | --- | | Reduces reliance on manual processes, streamlining flight reservation operations and improving efficiency for customers and airlines. |  |  | | --- | |  | | Addresses commonly encountered challenges in flight reservation systems, including booking conflicts, data accuracy, and efficient resource allocation. |

#### **Engineering Activities**

This section maps the scope of the **Flight Reservation System** project to key engineering activities, emphasizing the range of resources, innovation, and societal impact. **Table 4.3** presents these mappings along with rationales for their inclusion.

Table 4.3: Mapping with complex engineering activities.**Chapter 5**

# **Conclusion**

### Summary: The Flight Reservation System successfully addresses key challenges in flight booking and management by integrating features such as flight scheduling, reservation handling, and user data management. It enhances operational efficiency, reduces manual errors, and provides a reliable solution for managing flight reservations, tailored to meet the needs of both customers and airlines.

### Limitation: While the system performs effectively, it has limitations, such as its reliance on manual data entry and the absence of advanced features like real-time flight tracking or payment integration. The system also lacks multi-user support, which limits its scalability in larger operations.

### Future Work: Future enhancements could include adding a graphical user interface (GUI), implementing real-time flight tracking and notifications, integrating payment gateways, and supporting multi-user access for more efficient management. Additionally, improving system performance with better data handling and incorporating more complex booking algorithms could further optimize the user experience.