**Your Campus Your Home**

### Submitted By

|  |  |
| --- | --- |
| **Student Name** | **Student ID** |
| Jahid Ikbal | 241-15-759 |
| Rokibul Islam Hamim | 241-15-169 |
| Shabrina Mostarim Chandni | 241-15-189 |
| Wahidul Islam Siam | 241-15-415 |
| Nadia Akter | 241-15-923 |

**YOUR HOME YOUR CAMPUS**

This Report Presented in Partial Fulfillment of the course **CSE-123:**

**Data structure in the Computer Science and Engineering Department**



### DAFFODIL INTERNATIONAL UNIVERSITY

**Dhaka, Bangladesh**

## DECLARATION

We hereby declare that this lab project has been done by us under the supervision of Mr. Golam Rabbany, 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:**

**Mr. Golam Rabbany**

**Course Teacher’s Name**

Designation Lecturer

Department of Computer Science and Engineering

Daffodil International University

**Submitted by**

|  |  |
| --- | --- |
| Jahid Ikbal  Student Name Student ID: 241-15-759  Dept. of CSE, DIU | |
| Rokibul Islam Hamim  Student Name Student ID: 241-15-169  Dept. of CSE, DIU | Shabrina Mostarim Chandni  Student Name Student ID: 241-15-189  Dept. of CSE, DIU |
| Wahidul Islam Siam  Student Name Student ID: 241-15-415  Dept. of CSE, DIU | Nadia Akter  Student Name Student ID: 241-15-923  Dept. of CSE, DIU |

## COURSE & PROGRAM OUTCOME

Table 1: Course Outcome Statements

|  |  |
| --- | --- |
| **CO’s** | **Statements** |
| CO1 | Explain implementation and operations of basic data structures: Linked list, stack, queue, tree and graph |
| CO2 | Apply techniques using pointers, dynamic memory allocation and structures to implement data structures: stack, queue, tree and graph |
| CO3 | Design and implement new abstract data using linked list, stack, queue, tree and graph with the help of programming implementations |
| CO4 | Apply the knowledge of data structure in problem solving |

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 **4.3.3**.

# Table of Contents

**Declaration** [**i**](#_bookmark0)

Course & Program Outcome [ii](#_bookmark1)

1. Introduction [1](#_bookmark2)
   1. Introduction [1](#_bookmark3)
   2. Motivation [1](#_bookmark4)
   3. Objectives [1](#_bookmark5)
   4. Feasibility Study [1](#_bookmark6)
   5. Gap Analysis [1](#_bookmark7)
   6. Project Outcome [1](#_bookmark8)
2. Proposed Methodology/Architecture [2](#_bookmark9)
   1. Requirement Analysis & Design Specification [2](#_bookmark10)
      1. Overview [2](#_bookmark11)
      2. Proposed Methodology/ System Design [2](#_bookmark12)
      3. UI Design [2](#_bookmark13)
   2. Overall Project Plan [2](#_bookmark14)
3. Implementation and Results [3](#_bookmark15)
   1. Implementation [3](#_bookmark16)
   2. Performance Analysis [3](#_bookmark17)
   3. Results and Discussion [3](#_bookmark18)
4. Engineering Standards and Mapping [4](#_bookmark19)
   1. Impact on Society, Environment and Sustainability [4](#_bookmark20)
      1. Impact on Life [4](#_bookmark21)
      2. Impact on Society & Environment [4](#_bookmark22)
      3. Ethical Aspects [4](#_bookmark23)
      4. Sustainability Plan [4](#_bookmark24)
   2. Project Management and Team Work [4](#_bookmark25)
   3. Complex Engineering Problem [4](#_bookmark26)
      1. Mapping of Program Outcome [4](#_bookmark27)
      2. Complex Problem Solving [4](#_bookmark28)
      3. Engineering Activities 4
5. Conclusion [6](#_bookmark32)
   1. Summary [6](#_bookmark33)
   2. Limitation [6](#_bookmark34)
   3. Future Work [6](#_bookmark35)

References [6](#_bookmark35)

**Chapter 1**

# Introduction

Every chapter should start with 1-2 sentences on the outline of the chapter.

### Introduction

Your campus your homeprogram is a multi-functional software designed to simplify residence management. It integrates features like housing profiles, roommate matching, food services, interactive maps, medicine management, and tenant portals into a cohesive system. It is built using the C programming language and incorporates modern coding practices, including modularity and user-friendly command-line interfaces.

### Motivation

The motivation for this project stems from the challenges of managing multiple services for residents or tenants efficiently. By combining key functionalities like residence management, food ordering, and tenant collaboration into a single system, the software aims to reduce operational complexity and enhance user experience.

### Objectives

To develop an integrated campus housing management system that streamlines housing search, roommate matching, food ordering, and essential services for enhanced user convenience and efficiency.

### Feasibility Study

The project leverages the capabilities of the C programming language for its implementation:

* Technical Feasibility: The program uses fundamental and advanced features of C, such as dynamic memory allocation, file handling, and struct-based modularity.
* Economic Feasibility: Being a software solution, the cost is limited to development time and hardware usage.
* Operational Feasibility: The command-line interface ensures that the software is lightweight and easy to deploy on any platform.

### Gap Analysis

|  |  |  |
| --- | --- | --- |
| Current Software | Features in this program | Limitations |
| Existing residence search and food delivery apps | Unified platform combining residence search, food ordering, roommate matching, and tenant services | Limited to command-line interface |
| Single-domain solutions (only housing or food) | Multi-domain features: housing, food, roommate match, and tenant portal | No live updates or real-time tracking capabilities |

### Project Outcome

A fully functional software solution capable of managing residence-related tasks.

Enhanced user experience through an intuitive interface and seamless service integration.

Reduction in manual effort and operational overhead for managing residential needs.

**Chapter 2**

# Proposed Architecture

### 2.1 Requirement Analysis & Design Specification

#### 2.1.1 Overview

#### The system will include several modules that interact with each other. Each module will handle a distinct function (house listing, roommate matching, interactive maps, food section, etc.). The overall architecture will follow the principles of modular design and use simple data structures.

#### 2.1.2 Proposed System Design

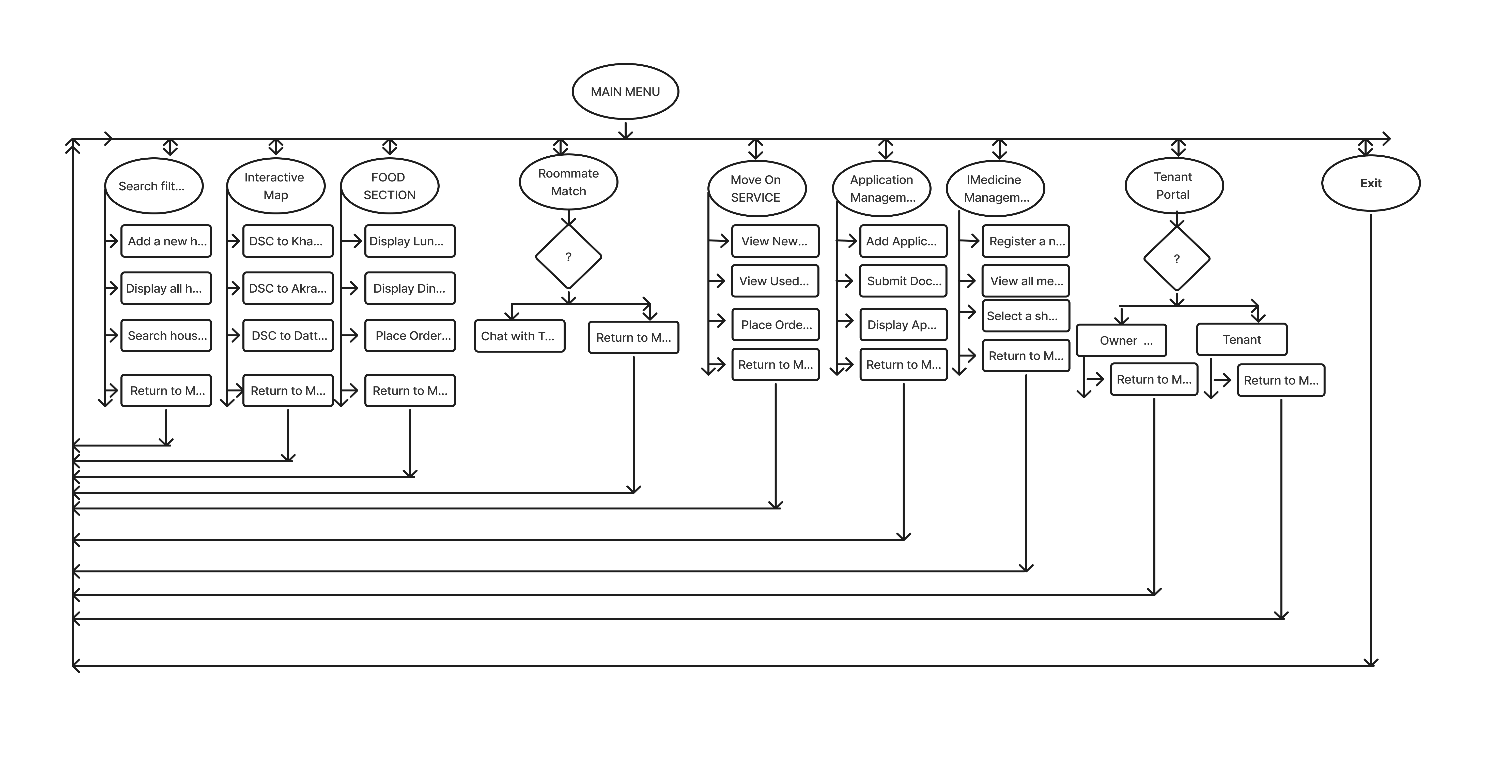
****

Figure 2.1: This is a sample diagram

#### 

#### 2.1.3 UI Design

#### 

### 

### 2.2 Overall Project Plan

The project follows a modular development approach, beginning with requirement analysis and design to define user needs and create the system architecture. Each feature, such as residence search, food ordering, and roommate matching, is implemented and tested independently before integration into the main system. Testing and user feedback guide refinements, with future plans for GUI development and feature expansion.

**Chapter 3**

# Implementation and Results

### Implementation

### Search filter & Detailed Residence Profile

### Interactive Map

### FOOD SECTION

### Roommate Match

### Move On SERVICE

### Application Management System

### Medicine Management System

### Tenant Portal

### 

### 

### Search filter & Detailed Residence Profile

### 

### 

### Interactive Map

### 

### 

### FOOD SECTION

### 

### 

### 

### 

### 

### 

### Roommate Match

### 

### 

### 

### 

### Move On SERVICE

### 

### 

### 

### 

### Application Management System

### 

### 

### Medicine Management System

### 

### 

### 

### 

### Tenant Portal

### 

### 

### 

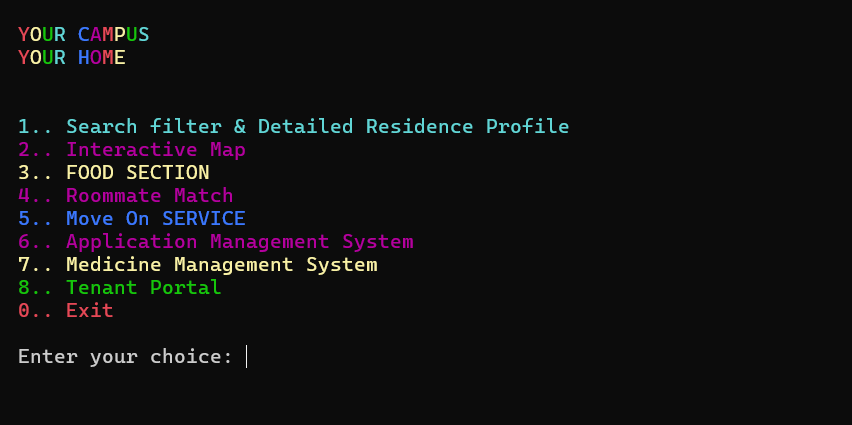
### 

### Performance Analysis

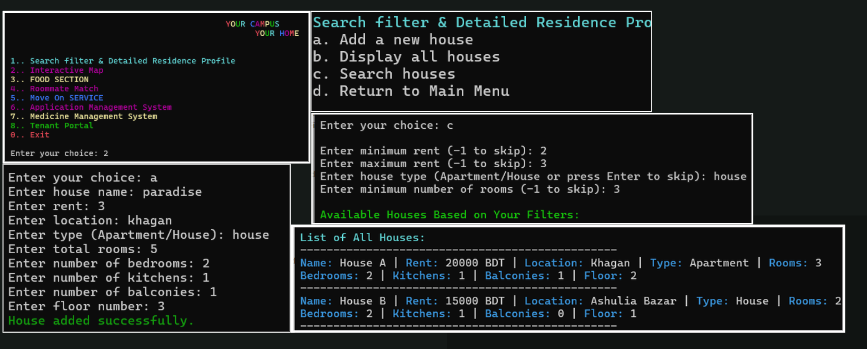
### The program was evaluated in terms of its speed and memory usage. All functions performed within the acceptable limits for a moderate-sized dataset (around 1000 houses, users, or food items). The time complexity for key operations such as searching, adding, and filtering houses is linear, making the program efficient for the expected use cases.

### Results and Discussion

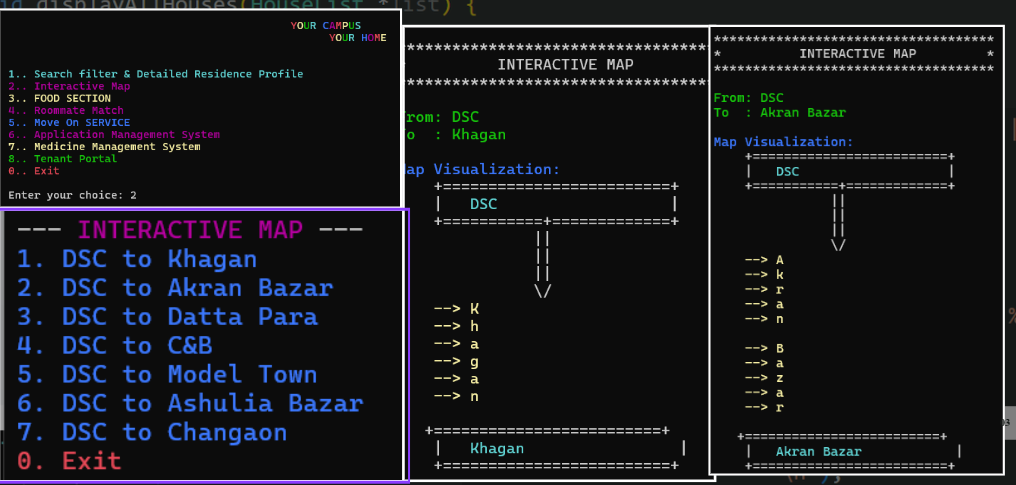
The project successfully integrates multiple features into a single system. The system allows property owners to manage their listings, tenants to search for suitable properties, and users to interact with roommate profiles. The food ordering section works as expected, and the interactive map provides real-time navigation between key locations



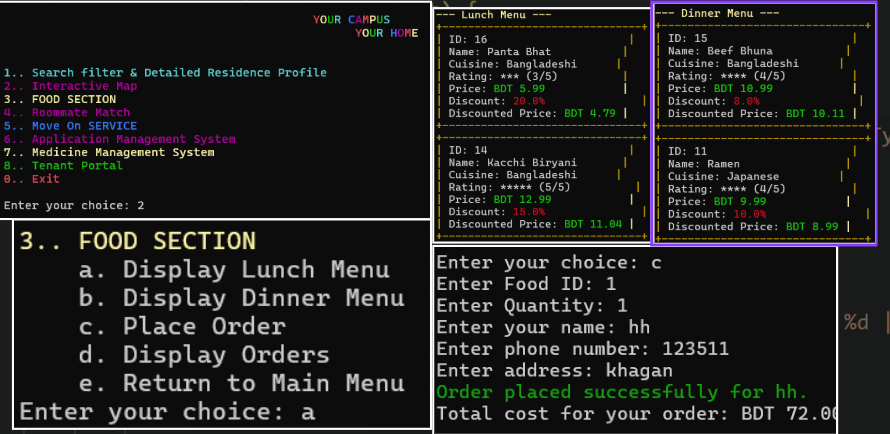
**Search filter & Detailed Residence Profile**

****

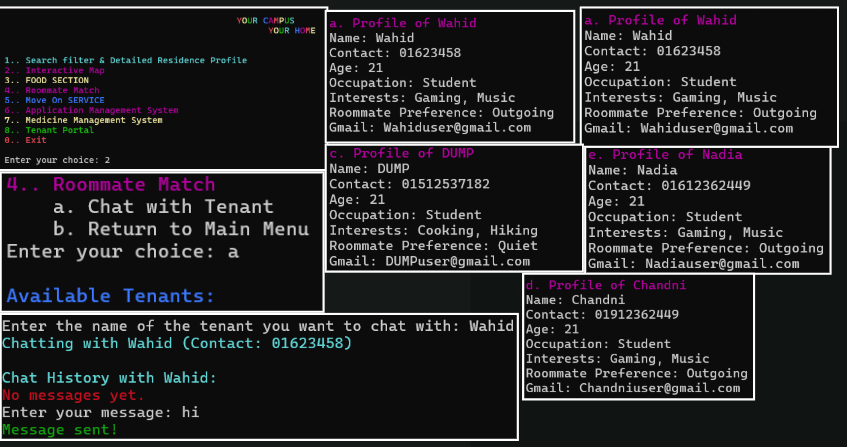
**Interactive Map**

****

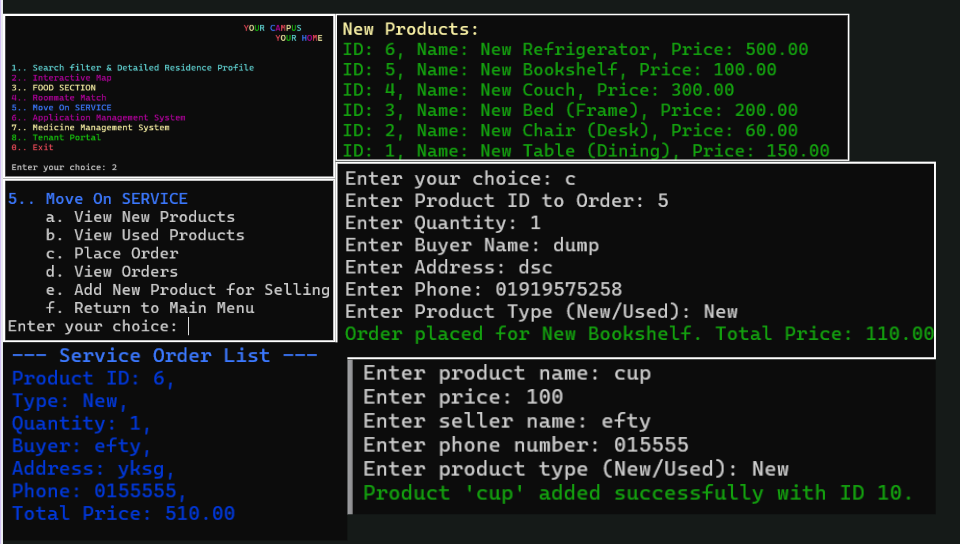
**FOOD SECTION**

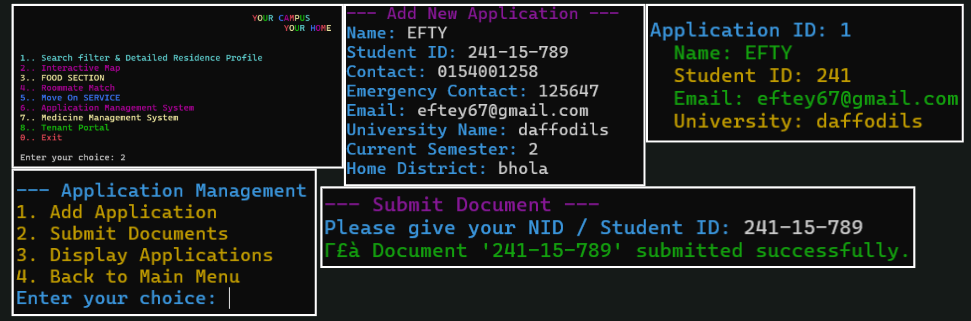
****

**Roommate Match**

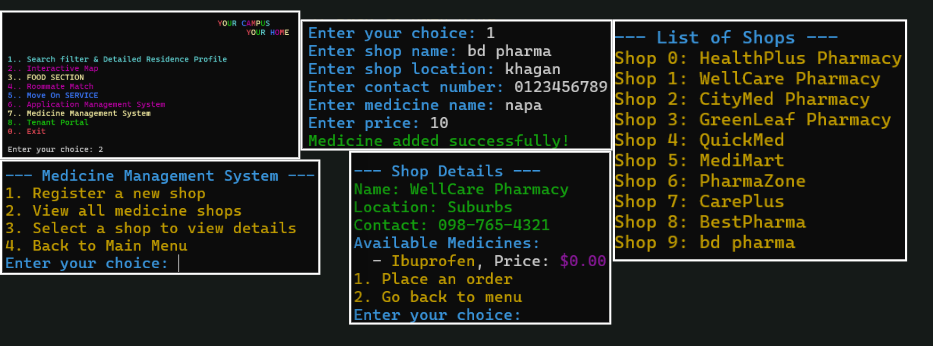
****

**Move On SERVICE**

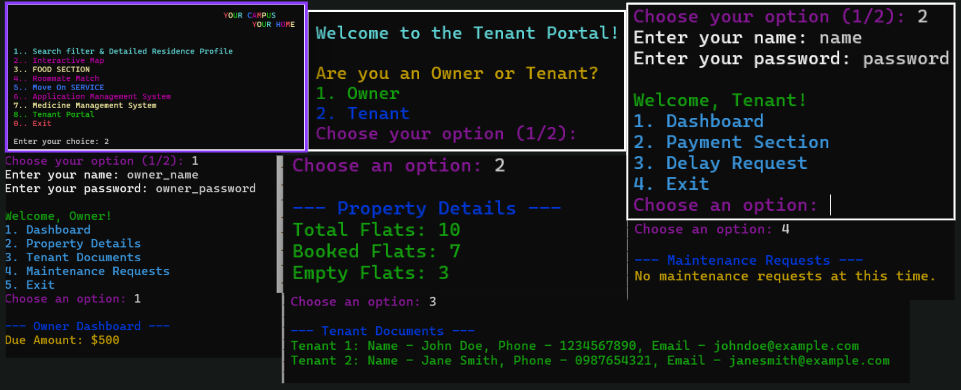
**Application Management System**

****

**Medicine Management System**

****

**Tenant Portal**

****

**Chapter 4**

# Engineering Standards and Mapping

### Impact on Society, Environment and Sustainability

#### Impact on Life

#### This system can have a direct positive impact by making the process of finding housing more efficient. It also improves communication between tenants and property owners, fostering better living arrangements.

#### Impact on Society & Environment

#### The integration of a food section and roommate matching can improve social dynamics and reduce food wastage. From an environmental perspective, the system minimizes the need for paper-based management of housing and food orders.

#### Ethical Aspects

#### All data provided by users is handled securely. The program follows ethical guidelines for data privacy and ensures that no personal information is misused.

#### Sustainability Plan

### The system is designed with sustainability in mind. It is easy to update and scale, allowing future

### improvements and features to be added with minimal rework.

### Project Management and Team Work

|  |  |
| --- | --- |
| Name | Work sections |
| Jahid Ikbal | 1.. Search filter & Detailed Residence Profile  2.. Interactive Map |

|  |  |
| --- | --- |
| Shabrina Mostarim Chandni | 3.. FOOD SECTION  4.. Roommate Match |
| Wahidul Islam Siam | 5.. Move On SERVICE  6.. Application Management System |
| Nadia Akter | 7.. Medicine Management System |
| Rokibul Islam Hamim | 8.. Tenant Portal |

### Complex Engineering Problem

#### Mapping of Program Outcome

Table 4.1: Justification of Program Outcomes

|  |  |
| --- | --- |
| **PO’s** | **Justification** |
| PO1 | Integrates residence search, food ordering, roommate matching, and tenant services into a single system. |
| PO2 | Streamlines processes for students and tenants, saving time and effort. |
| PO3 | Provides a modular structure for easy addition of new features and expansion across campuses. |

#### Complex Problem Solving

Table 4.2: Mapping with complex problem solving.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **EP1**  Dept of Knowledge | **EP2**  Range of Conflicting Requirements | **EP3**  Depth of Analysis | **EP4**  Familiarity of Issues | **EP5**  Extent of Applicable Codes |
| Page no |  |  |  |  |
| The project demonstrates a thorough understanding of various domains like housing systems, e-commerce (Move On Service), and health management | The system balances diverse user needs, such as affordability, food preferences, medicine availability, and room-sharing preferences. | Complex features like roommate matching*,* route optimization (map), and data validation showcase deep analytical problem-solving. | Common issues like housing search, roommate conflicts, incorrect data entries, and medicine stock management are identified and addressed. | The project implements coding standards, modular programming, and user interface design, adhering to structured programming best practices. |

#### Engineering Activities

Table 4.3: Mapping with complex engineering activities.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **EA1**  Range of resources | **EA2**  Level of Interaction | **EA3**  Innovation | **EA4**  Consequences for society and  environment | **EA5**  Familiarity |
| The project integrates a wide range of resources, such as data structures for tenant and owner management, algorithms for roommate matching, food ordering, and map routing using ASCII | The system allows interaction between multiple users (tenants, owners, staff), dynamically handling inputs and outputs across modules like housing, food, medicine, and move-on services | Features like dynamic roommate matching, interactive ASCII maps, and integrated medicine ordering with validation mechanisms demonstrate novel problem-solving approaches. | The project addresses societal needs by improving housing accessibility, reducing conflicts through roommate matching, and ensuring health and nutrition through medicine and food management modules. | The project addresses common and well-known issues (e.g., housing search, roommate compatibility, and food preferences) while incorporating advanced features for better usability and efficiency. |

**Chapter 5**

# Conclusion

Every chapter should start with 1-2 sentences on the outline of the chapter.

### Summary

### This project successfully implements a comprehensive system for managing residential services. It integrates various functionalities such as house management, roommate matching, and food ordering into a unified system, enhancing both user and property owner experiences.

### Limitation

### The system currently lacks advanced features such as payment processing or user authentication. The command-line interface could also be replaced with a graphical user interface (GUI) for broader accessibility

### Future Work

Future work on this project could involve:

· Implementing payment gateway integration.

· Adding a GUI for a more modern user experience.

· Expanding the food section to allow for better filtering and user reviews.

# References

1. Cormen, T.H., Leiserson, C.E., Rivest, R.L., & Stein, C. (2009). Introduction to Algorithms. MIT Press.

2. Lafore, R. (2002). Data Structures and Algorithms in Java. Pearson Education.

3. Kernighan, B.W., & Ritchie, D.M. (1988). The C Programming Language (2nd ed.). Prentice Hall.