

**Govt. Graduate College (W) Satellite Town, Gujranwala**

**Final Documentation**

**ResideMe**

**STATEMENT OF SUBMISSION**

This is to certify that the following students have successfully completed the final project named: ResideMe at The Govt. Graduate College (w) Satellite Town, Gujranwala, and to fulfill the partial requirement of the degree of BSIT**.**

|  |  |  |
| --- | --- | --- |
| **Sr. No.** | **Student’s Roll No** | **Student’s Name** |
| **1** | **042302** | **Aqsa Shehzadi** |
| **2** | **042308** | **Ayesha Muhammad Yaqoob** |
| **3** | **042338** | **Hafiza Attika Imran** |

**Internal Supervisor Head of Department**

**Ms. Nadia Irum Qazi Ms. Sadaf Sikander**

**Associate Professor Lecturer**

**Govt. Graduate College (W), Govt. Graduate College (W),**

**Satellite town, Gujranwala Satellite town, Gujranwala**

**Acknowledgment**

This project is the direct result of teamwork. We sincerely thank the instructors and students who have shared their suggestions with us over the last few weeks. Many thanks go to Sadaf Sikanderour Project coordinator and supervisor, whose management helped us to get this project produced in a timely and efficient manner. And finally, we thank to our parents and all the other people who weren’t directly involved in this project but who have stood by us the whole way, we appreciate your encouragement and support. We are also thankful to our friends and siblings whose silent support led us to complete our project.

**Write to us**

We welcome your response to this project. If there is anything you want to mention about the improvement of this project, please let us know:

|  |  |
| --- | --- |
| Aqsa Shehzadi | [aqshaqsh1234@gmail.com](mailto:aqshaqsh1234@gmail.com) |
| Ayesha Muhammad Yaqoob | [aisha.yaqoob095@gmail.com](mailto:aisha.yaqoob095@gmail.com) |
| Hafiza Attika Imran | [iamabdulrehmanimran2004@gmail.com](mailto:iamabdulrehmanimran2004@gmail.com) |
|  |  |

**For Our Beloved Parents, Teachers and Friends**

“We dedicated this project to Allah Almighty, our Creator, the most Merciful, the most Beneficent, and who is the supreme part of the Whole universe. He has been the source of strength throughout this whole program. We are working on a dissertation dedicated to our family, teachers, and friends. My parent's words of encouragement and push for tenacity ring in my ears, making me feel a special sense of gratitude, and our friends, who never left our side and is very special”

**Table of Contents**

[Chapter 1 1](#_Toc171624168)

[**1. Introduction** 2](#_Toc171624169)

[**1.1 Project Feasibility Report** 3](#_Toc171624170)

[**1.1.1 Technical Feasibility** 3](#_Toc171624171)

[**1.1.2. Operational Feasibility** 4](#_Toc171624172)

[**1.1.3. Economic Feasibility** 4](#_Toc171624173)

[**1.1.4. Schedule Feasibility** 4](#_Toc171624174)

[**1.1.5. Specification Feasibility** 4](#_Toc171624175)

[**1.1.6. Information Feasibility** 5](#_Toc171624176)

[**1.1.7. Motivational Feasibility** 5](#_Toc171624177)

[**1.1.8. Legal & Ethical Feasibility** 5](#_Toc171624178)

[Chapter 2 6](#_Toc171624179)

[**2.1 Project Scope** 7](#_Toc171624180)

[Chapter 3 5](#_Toc171624181)

[3. Project Costing: 6](#_Toc171624182)

[**3.1 Project Cost Estimation By Function Point Analysis:** 7](#_Toc171624183)

[4. Project Planning 5](#_Toc171624184)

[**4.1 Project Scope** 5](#_Toc171624185)

[**4.2 CMP Critical Path** 5](#_Toc171624186)

[**4.2.1 Estimate Activity Completion** 5](#_Toc171624187)

[**Determine the Sequence of the Activities** 6](#_Toc171624188)

[**4.2 Estimate Activity Completion Time** 6](#_Toc171624189)

[**4.3 Network Diagram** 7](#_Toc171624190)

[**Microsoft Word 2013** 10](#_Toc171624191)

[**Gantt Project** 10](#_Toc171624192)

[5. Requirement Engineering 17](#_Toc171624193)

[5.1 Introduction 17](#_Toc171624194)

[**5.2 Existing System** 17](#_Toc171624195)

[**5.3 Scope of the system** 18](#_Toc171624196)

[5.10 Requirements Trace Ability Matrix 21](#_Toc171624197)

[**5.11 High Level Case Diagram** 22](#_Toc171624198)

[6. Introduction 22](#_Toc171624199)

[**6.1 Use Case Description** 22](#_Toc171624200)

[UC\_1 Login 22](#_Toc171624201)

[**Table 6.2** 23](#_Toc171624202)

[**Table 6.3** 23](#_Toc171624203)

[**Table 6.4** 24](#_Toc171624204)

[**Table 6.5** 24](#_Toc171624205)

[**Table 6.6** 24](#_Toc171624206)

[**Table 6.7** 25](#_Toc171624207)

[**6.2. Use Case Diagram (refined and updated)** 26](#_Toc171624208)

[7. Sequence Diagram 30](#_Toc171624209)

[**7.1 Login** 30](#_Toc171624210)

[**7.2 User Profile** 31](#_Toc171624211)

[8. Collaboration Diagram 35](#_Toc171624212)

[**8.1 Login** 35](#_Toc171624213)

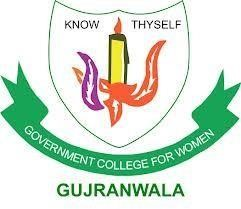
[**8.2 User Profile** 36](#_Toc171624214)

[9.1 Domain Model 41](#_Toc171624215)

[10.1 Design Class Diagram 42](#_Toc171624216)

[11.1 Data Model 43](#_Toc171624217)

[12. Project Interfaces 45](#_Toc171624218)



# Chapter 1

**Project Feasibility Report**

## **1. Introduction**

**Problem Statement:**

Student hostels and residences face numerous challenges, including administrative inefficiencies, resource management issues, financial problems, communication breakdowns, and inventory management difficulties. These problems lead to high error rates in manual processes, suboptimal room allocation, delayed fee collection, ineffective updates, and increased operational costs. Additionally, the lack of comprehensive reporting and data analytics hampers decision-making and long-term planning, culminating in student dissatisfaction due to unresolved complaints and inconsistent services. Handling human labor and their pay involves challenges in accurate record-keeping, payroll calculations, compliance, timely payments, and data security, which can be addressed with automation and robust management systems.

**Solution:**

The ResideMe aims to streamline hostel administration and offers a comprehensive solution to these problems, providing the following benefits:

**1.** **Administrative Efficiency**:

* **Automated Processes**: Automates student registration, room allocation, fee collection, and attendance tracking, reducing manual errors and saving time.
* **Centralized Database**: Maintains all student and administrative data in a centralized, easily accessible database.

**2.** **Effective Resource Management**:

* **Optimized Room Allocation**: Uses algorithms to efficiently allocate rooms and beds based on availability, preferences, and criteria, ensuring optimal utilization of resources.
* **Maintenance Tracking**: Tracks maintenance requests and schedules, ensuring timely repairs and upkeep of facilities.

**3.** **Financial Transparency and Efficiency**:

* **Automated Fee Collection**: Facilitates timely and accurate fee collection with automated reminders and notifications, reducing delays and discrepancies.
* **Financial Reports**: Generates clear and accurate financial reports for better transparency and accountability.

**4.** **Improved Communication**:

* **Centralized Communication Platform**: Provides a centralized platform for sending notifications and alerts to students, effective and timely communication.
* **Web Access**: Ensures easy access to information and updates through web interfaces.

**5.** **Inventory Management**:

* **Real-Time Inventory Tracking**: Tracks inventory levels in real-time, preventing shortages and overstocking.
* **Automated Reordering**: Sets thresholds and triggers for automatic reordering of supplies, ensuring continuous availability.

**6.** **Data Insights and Reporting**:

* **Comprehensive Reporting**: Offers detailed reports on occupancy, financials, maintenance, and more, providing valuable insights for informed decision-making.
* **Data Analytics**: Analyzes historical data and trends to help with long-term planning and improvement efforts.

**7.** **Enhanced Student Experience**:

* **Efficient Complaint Resolution**: Provides a system for logging, tracking, and resolving student complaints quickly and efficiently.
* **Consistent Services**: Ensures consistent and high-quality services and facilities through streamlined management processes.

## **1.1 Project Feasibility Report**

The feasibility report for the ResideMe highlights its efficiency and effectiveness in modern hostel administration. Technically, it utilizes scalable web and mobile technologies, ensuring compatibility with existing infrastructures. Economically, the initial investment is offset by long-term savings from reduced administrative costs and errors. Operationally, the RESIDEME automates tasks such as admissions, fee management, and maintenance, improving communication and productivity. Security features ensure data integrity and privacy, addressing stakeholder concerns. Overall, the ResideMe is a viable and beneficial solution for contemporary hostel management.

* Technical
* Operational
* Economic
* Schedule
* Specification
* Information
* Motivational
* Legal and Ethical

## **1.1.1 Technical Feasibility**

The technical feasibility of the ResideMe is solid, utilizing modern technologies for a scalable, secure, and user-friendly platform:

1. **Technology Stack:** Laravel for the backend, MySQL for data management, and Next.js for the frontend, ensures a scalable, secure, and efficient solution for managing student accommodations.
2. **Modular Design**: The system can be designed with a modular architecture, making it easier to develop, test, and maintain individual components (e.g., user management, room allocation, fee collection).
3. **Network Requirements**: Reliable internet connectivity to ensure seamless access to the system for all users.
4. **Automated Testing**: Tools and frameworks available for automated testing to ensure system reliability and performance.
5. **Security Measures**: Implementing robust security measures such as encryption, authentication, and access control to protect sensitive data.
6. **User Interface:** Features a responsive, intuitive design for accessibility across devices.
7. **Integration Capabilities:** Seamlessly integrates with existing systems and third-party tools via APIs.
8. **Maintenance and Support:** Provides clear documentation, automated testing, CI/CD pipelines, and ongoing maintenance.

These factors confirm the technical feasibility of implementing the RESIDEME effectively.

### **1.1.2. Operational Feasibility**

The operational feasibility of the ResideMe is high, offering significant improvements in hostel management. It automates key processes like admissions, room allocations, fee management, and maintenance requests, reducing manual workload and errors. The system's user-friendly interface ensures easy adoption by students, staff, and administrators, with minimal training required. Real-time data access enhances decision-making and communication, while integrated messaging and notifications streamline communication. The RESIDEME optimizes resource management and is scalable to accommodate future growth. It also ensures data security and regulatory compliance, addressing privacy concerns. Cost-effective and supported by robust maintenance structures, the ResideMe enhances overall efficiency and operational effectiveness.

### **1.1.3. Economic Feasibility**

The economic feasibility of the ResideMe is favorable, offering substantial cost savings and financial benefits over time. Initial development and implementation costs are offset by reduced administrative expenses, as the system automates labor-intensive processes like admissions, fee management, and maintenance tracking. Improved resource utilization and minimized errors lead to further savings. Additionally, the ResideMe reduces the need for physical storage and paper-based record-keeping, cutting operational costs. Enhanced efficiency and productivity contribute to long-term financial gains, making the ResideMe a cost-effective investment for modern hostel management.

### **1.1.4. Schedule Feasibility**

The schedule feasibility of the ResideMe is achievable through careful planning and resource management. A detailed project timeline is essential, outlining clear milestones for each development phase—from initial requirements gathering and system design to coding, testing, and deployment. Effective resource allocation, including human and technological resources, ensures that development stays on track. Identifying dependencies and potential risks early allows for proactive mitigation strategies to minimize disruptions. Adopting an iterative development approach facilitates ongoing adjustments based on feedback and ensures alignment with project goals. Adequate time allocated for testing and validation ensures that the ResideMe meets quality standards before deployment. A well-planned deployment strategy, including user training and support, further contributes to the schedule feasibility by anticipating and addressing implementation challenges efficiently.

### **1.1.5. Specification Feasibility**

The specification feasibility of the ResideMe is robust, driven by clear and comprehensive requirements that align with modern hostel management needs. Key specifications include automated processes for admissions, room allocation, fee management, and maintenance requests, all managed through a user-friendly interface accessible via web and mobile platforms. The system integrates secure authentication, role-based access control, and data encryption to safeguard student information and ensure compliance with data protection regulations. Real-time data synchronization across devices enhances operational efficiency, while features like automated notifications and reporting capabilities facilitate effective communication and decision-making. Scalability is ensured through flexible architecture and support for future expansions and integrations with existing systems. Overall, the ResideMe specifications meet contemporary standards for usability, security, scalability, and functionality, making it a viable solution for enhancing hostel management operations.

### **1.1.6. Information Feasibility**

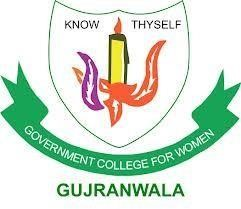
The information feasibility of the ResideMe underscores its capability to effectively handle and utilize data for optimal hostel operations. This includes leveraging modern database management systems and robust data processing capabilities to ensure seamless storage, retrieval, and analysis of information related to student admissions, room allocations, fee management, and maintenance requests. Real-time data synchronization across web and mobile platforms facilitates timely updates and access for administrators, students, and staff, enhancing operational efficiency. Security measures such as encryption and access controls safeguard sensitive information, ensuring compliance with data protection regulations. The system's ability to provide accurate and accessible information supports informed decision-making and improves overall transparency in hostel management processes. In essence, the ResideMe's information feasibility rests on its capacity to efficiently manage and utilize data to enhance organizational performance and user satisfaction.

### **1.1.7. Motivational Feasibility**

The motivational feasibility of the ResideMe is evident through its potential to inspire positive change and enhance user motivation within hostel environments. By streamlining administrative tasks and improving access to essential services such as room allocations, fee management, and maintenance requests, the RESIDEME reduces bureaucratic hurdles and empowers both students and staff. Students benefit from simplified processes that enhance their overall living experience, promoting a sense of convenience and satisfaction. For staff and administrators, the system enables more efficient management practices, freeing up time for more meaningful interactions and strategic initiatives. Improved communication channels and transparency foster a collaborative and supportive atmosphere, motivating stakeholders to actively engage in the hostel community. Ultimately, the ResideMe's motivational feasibility lies in its ability to create a conducive environment where productivity, satisfaction, and mutual respect thrive, benefiting all involved parties.

### **1.1.8. Legal & Ethical Feasibility**

The legal and ethical feasibility of the ResideMe ensures compliance with applicable laws and ethical standards while upholding user rights and privacy. It incorporates robust data protection measures, such as encryption and secure authentication, to safeguard sensitive information like student records and financial data. Compliance with regulations such as GDPR or HIPAA, depending on jurisdiction, ensures that data handling meets legal requirements. Ethically, the ResideMe respects user privacy through transparent data practices and ensures equitable access to services for all students. Clear policies and procedures regarding data access and usage uphold ethical principles, promoting trust among stakeholders. Overall, the ResideMe's legal and ethical feasibility ensures responsible and lawful operations that prioritize user confidentiality, fairness, and compliance with regulatory standards.



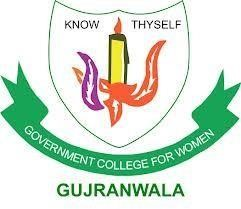
# Chapter 2

**Project scope**

## **2.1 Project Scope**

The project scope of the ResideMe encompasses a comprehensive solution aimed at enhancing the efficiency and effectiveness of hostel administration. It includes functionalities such as automated student admissions, room allocation based on predefined criteria, management of hostel fees and payments, and tracking of student activities and attendance. The ResideMe will feature a user-friendly interface accessible via web platforms, facilitating real-time data access and updates for students, hostel staff, and administrators. Additional features include a maintenance request system to streamline facility management and ensure prompt resolution of issues. The scope also involves integrating security measures like data encryption and role-based access control to protect sensitive information. Scalability is considered to accommodate varying hostel sizes and future expansions. Ultimately, the ResideMe aims to optimize resource utilization, improve communication, and provide a seamless experience for all users involved in hostel operations.

.



# Chapter 3

**Project Costing**

# 3. Project Costing:

Function-oriented software matrices use a measure of the functionality delivered by the application as a normalization value. Since “Functionality can not be measured directly, it must be derived indirectly using other direct measures”. Function-oriented metrics were first proposed by Albrecht, who suggested a measure called the function point. Function Point are derived using an empirical relationship based on countable (direct) measure of software’s information domain and assessment of software complexity.

We have chosen Function Point Analysis for our Project Cost Estimation.

Number of external inputs: 6

Number of external outputs: 4

Number of external inquiries: 4

Number of internal logic files: 6

Number of external interface files: 7

**External Inputs:**

* Internet connectivity
* User registration
* Login
* Room booking
* Complaints registration
* Fee submission

**External Outputs:**

* Room details report
* Hostel overview report
* Facility availability report
* Student registration confirmation

**External Inquiries:**

* Room availability inquiry
* Student information inquiry
* Facility details inquiry
* Student registration status inquiry

**Internal Logic Files:**

* Room data file
* Hostel data file
* Student data file
* Registration data file
* Error messages file
* Resource management file

**External Interface Files:**

* User Authentication Interface
* Room Management Interface
* Hostel Information Interface
* Student Management Interface
* Student Registration Interface
* Remote Access Interface
* Resource Management Interface

## **3.1 Project Cost Estimation By Function Point Analysis:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Type of Component** |  | | | |
|  | **Low** | **Average** | **High** | **Total** |
| External Inputs | 3\*6=18 | 4\*6=24 | 6\*6=36 | 78 |
| External Outputs | 4\*4=16 | 5\*4=20 | 7\*4=28 | 64 |
| External Inquiries | 3\*4=12 | 4\*4=16 | 6\*4=24 | 52 |
| Internal Logical Files | 7\*6=42 | 10\*6=60 | 15\*6=90 | 192 |
| External Interface Files | 5\*7=35 | 7\*7=49 | 10\*7=70 | 154 |
| **Total Number of Unadjusted Function Points** | | | | **540** |

**Table 3.1**

**Calculation of (Fi):**

|  |  |  |
| --- | --- | --- |
| **Sr.#** | **Questions** | **Scale** |
| 1 | Does the system require heavy configuration? | 3 |
| 2 | Is communication required? | 4 |
| 3 | Are there distributed processing functions? | 2 |
| 4 | Is performance critical? | 5 |
| 5 | Will the system run in an existing, heavily utilized operational environment? | 4 |
| 6 | Does the system require online data entry? | 4 |
| 7 | Does the online data entry require the input transaction to be built over operations? | 3 |
| 8 | Are the files updated online? | 4 |
| 9 | Are the inputs, output, files, or inquiries complex? | 3 |
| 10 | Is the internal processing complex? | 3 |
| 11 | Is the code designed to be reusable? | 4 |
| 12 | Are conversion/installation included in the design? | 4 |
| 13 | Is the system designed for multiple installations in different organizations? | 3 |
| 14 | Is the application designed to facilitate change? | 4 |
| **Value Adjusted Factors (Fi)** | | **50** |

**Table 3.2**

**Calculation of Function Point (FP):**

FP estimated = Count Total \* (0.65 + 0.01 \* (Fi))

= 540 \* (0.65 + 0.01 \* 50)

= 540 \* 1.15

= 621

**Final Function Point Calculation is:**

Final Adjusted FP =UFC\*VAF

=540 \* 621

= 335340

**Cost per Function Point:**

Labor Rate = Rs. 3600

Cost per FP = Labor Rate / Productivity parameter

= 3600 / 10

= 360

**Total Project Cost:**

Total Project Cost = FP estimated \* Cost Per FP

= 621\* 360

=223560

**Total Estimated Effort:**

Total Estimated Effort = FP est. / productivity parameter

= 621 / 10

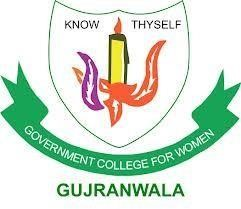
= 62.1

**Duration of Project**

Duration of Project = Effort months / No. of persons

**=** 18/3

= 6 months



**Chapter 4**

**Project Planning**

# 4. Project Planning

## **4.1 Project Scope**

The project scope of the ResideMe encompasses a comprehensive solution aimed at enhancing the efficiency and effectiveness of hostel administration. It includes functionalities such as automated student admissions, room allocation based on predefined criteria, management of hostel fees and payments, and tracking of student activities and attendance. The ResideMe will feature a user-friendly interface accessible via web platforms, facilitating real-time data access and updates for students, hostel staff, and administrators. Additional features include a maintenance request system to streamline facility management and ensure prompt resolution of issues. The scope also involves integrating security measures like data encryption and role-based access control to protect sensitive information. Scalability is considered to accommodate varying hostel sizes and future expansions. Ultimately, the ResideMe aims to optimize resource utilization, improve communication, and provide a seamless experience for all users involved in hostel operations.

## **4.2 CMP Critical Path**

### **4.2.1 Estimate Activity Completion**

**Table 4.1**

|  |  |
| --- | --- |
| **Project Scope Definition** | **A** |
| **Requirement Gathering** | **B** |
| **Analysis** | **C** |
| **Design** | **D** |
| **Coding** | **E** |
| **Test and Debugging** | **F** |
| **Implementation** | **G** |

### **Determine the Sequence of the Activities**

Some activities are dependent on the completion of others. There are many activities that are dependent on each other.

The following activities are dependent on one another:

* Analysis Project Scope Definition, Requirement Gathering
* Design Analysis
* Coding Design

* Testing and Debugging Coding

* Implementation Testing and Debugging, Coding

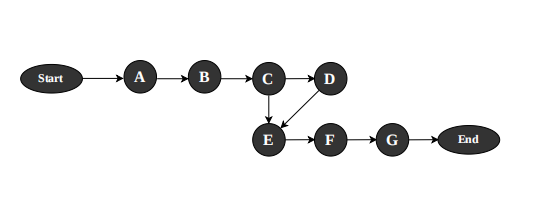
## **4.2 Estimate Activity Completion Time**

**Table 4.2**

|  |  |  |
| --- | --- | --- |
| **Activity ID** | **Predecessor** | **Duration (Days)** |
| A | None | 5 |
| B | A | 7 |
| C | B | 7 |
| D | C | 10 |
| E | D | 20 |
| F | E | 15 |
| G | F | 7 |

### **4.3 Network Diagram**

Once the activities and their sequence have been defined, the CPM diagram can be drawn.



**Figure 4.1**

**Path 1: A → B → C → D → E → F → G 5 + 7 + 7 +** 10 + 20 + 15 **+ 7 = 71**

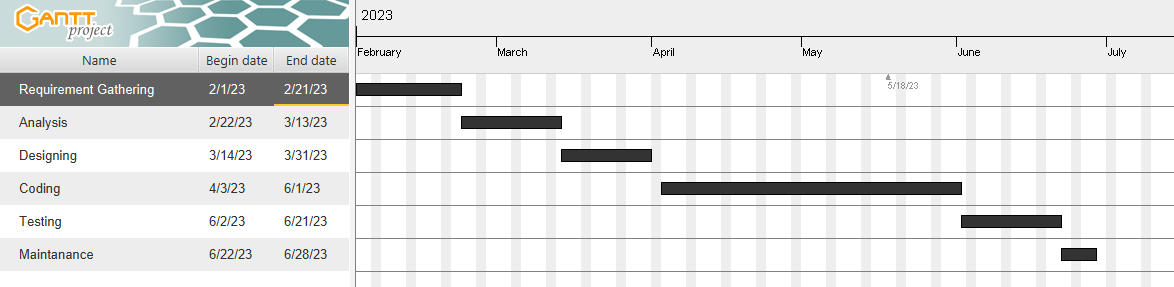
**Path 2: A → B → C → E → F → G 5 + 7 + 7 +** 20 + 15 **+ 7 = 61**

#### **4.4 Identify the Critical Path**

**Table 4.3**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Activity** | **Duration** | **ES** | **EF** | **LS** | **LF** | **TS** |
| A | 5 | 0 | 5 | 0 | 5 | 0 |
| B | 7 | 5 | 12 | 5 | 12 | 0 |
| C | 7 | 12 | 19 | 12 | 19 | 0 |
| D | 10 | 19 | 29 | 19 | 29 | 0 |
| E | 20 | 29 | 49 | 29 | 49 | 0 |
| F | 15 | 49 | 64 | 49 | 64 | 0 |
| G | 7 | 64 | 71 | 64 | 71 | 0 |

##### **4.5 Gantt Chart**



|  |  |  |  |
| --- | --- | --- | --- |
| Name | | Begin date | End date |
| Requirement Gathering | | 1/18/2024 | 2/10/24 |
| Analysis | | 2/11/2024 | 2/28/2024 |
| Designing | | 3/1/2024 | 3/20/2024 |
| Coding |  | 3/21/2024 | 6/21/2024 |
| Testing |  | 6/22/2024 | 7/22/2024 |
| Maintenance |  | 7/23/2024 | 8/23/2024 |

**4.6 Introduction to Team Members and Their Skill**

|  |  |
| --- | --- |
| **Student’s Roll No** | **Student’s Name** |
| **042308** | **Ayesha Muhammad Yaqoob** |
| **042338** | **Hafiza Attika Imran** |
| **042302** | **Aqsa Shehzadi** |

**Ayesha Yaqoob**

Ayesha Muhammad Yaqoob is leader of the group and backend programmer for this project. She is responsible for implementing and maintaining backend infrastructure, ensuring seamless functionality.

**Hafiza Attika Imran**

She has knowledge about Java and XML language.She is responsible for all Back-end programming of the project.She is responsible for the application being developed.She is also responsible for debugging.

**Aqsa Shehzadi**

She has knowledge about Java and XML language.She is responsible for all Back-end programming of the project.She is responsible for the application being developed.She is also responsible for debugging.

**4.7 Tools and Technology with Reasoning**

#### **Visual studio 2010**

Visual studio is a good Code Editor,that supports [syntax highlighting](http://en.wikipedia.org/wiki/Syntax_highlighting) and [code completion](http://en.wikipedia.org/wiki/Autocomplete) using [IntelliSense](http://en.wikipedia.org/wiki/IntelliSense) for [variables,](http://en.wikipedia.org/wiki/Variable_(programming)) [functions,](http://en.wikipedia.org/wiki/Subroutine) [methods,](http://en.wikipedia.org/wiki/Method_(computer_science)) [loops](http://en.wikipedia.org/wiki/Program_loops) and [LINQ](http://en.wikipedia.org/wiki/LINQ) queries The WPF designer, was introduced with Android Studio.

#### **JAVASCRIPT**

JavaScript, often abbreviated as JS, is a programming language and core technology of the Web, alongside HTML and CSS.

#### **Next.JS**

Next.JS is a React framework that enables several extra features, including server-side rendering and generating static websites. React is a JavaScript library that is traditionally used to build web applications rendered in the client's browser with JavaScript.

#### **HTML**

HTML is a markup language used by the browser to manipulate text, images, and other content, in order to display it in the required format.

#### **CSS**

CSS is used to define styles for your web pages, including the design, layout and variations in display for different devices and screen sizes.

#### **PHP**

PHP is an open-source, server-side programming language that can be used to create websites, applications, customer relationship management systems and more. It is a widely-used general-purpose language that can be embedded into HTML.

#### **Laravel**

Laravel is a free and Open-Source PHP web framework, created by Taylor Otwell and intented for the development of web applications following the model-view-controller architectural pattern and based on symphony.

### **Microsoft Word 2013**

Microsoft Office 2013 is used to prepare the documentation of the Project in a professional manner.

### **Gantt Project**

A Gantt chart is a project management tool that illustrates work completed over a period of time in relation to the time planned for the work. It typically includes two sections: the left side outlines a list of tasks, while the right side has a timeline with schedule bars that visualize work.

**MySQL DB**

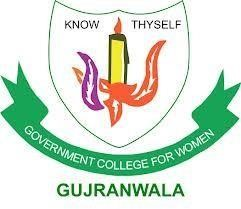
MySQL is a popular open-source relational database management system (RDBMS) widely used for backend data storage and management in web applications. It uses Structured Query Language (SQL) to manage and manipulate data and is renowned for its reliability, performance, and ease of use. MySQL supports a variety of platforms, including Linux, Windows, and macOS, and can be integrated seamlessly with programming languages like PHP and JavaScript (Node.js).

**Visual Paradigm Online**

Visual Paradigm Online (VP Online) Free Edition is a FREE online drawing software developed for the community. Draw free diagrams with an easy-to-use online drawing tool without limitations such as, number of diagrams, size of project file, etc.

**4.8 Risk List**

* Risk of unauthorized access to sensitive information.
* Slow performance with large datasets or peak usage times.
* Difficulty scaling the DB as user number grows.
* Risk of data corruption or loss during transactions.
* Insufficient training leading to misuse or errors.
* Lack of resources for ongoing maintenance and support.
* Inadequate testing leading to undiscovered bugs post-deployment.
* Challenges in migrating data from the manual system to the new digital system.

****

**Chapter 5**

**Requirement Engineering**

# 5. Requirement Engineering

# 5.1 Introduction

ResideMe is an advanced Hostel Management System designed to address the inefficiencies and challenges of traditional manual systems. This web application provides a comprehensive solution for managing hostel-related information, including rooms, students, facilities, and registrations. By leveraging modern technologies, ResideMe aims to deliver a user-friendly, error-free, secure, and efficient management system tailored to the unique needs of hostel operations.

**Systems Specifications**

The following are the clauses that must be included while describing the system specifications.

## **5.2 Existing System**

In many hostels, management tasks are handled manually using paper records and basic spreadsheets. This approach is time-consuming, prone to errors, and results in data inconsistencies. Access to information is limited, and there is no remote access capability, hindering real-time decision-making. Resource tracking and allocation are inefficient, and the security of physical records is a significant concern. Generating accurate reports is labor-intensive, affecting timely decision-making and overall efficiency.

**Business Organization Chart**

### **5.3 Scope of the system**

ResideMe aims to automate and streamline hostel management tasks with a user-friendly interface, robust security measures, and scalability to handle growth. It provides real-time accessibility, advanced reporting capabilities, seamless integration with existing systems, and ongoing support to ensure efficient and reliable hostel operations.

#### **5.4 Summary of Requirements**

1. **User-Friendly Interface**

Easy to use and accessible design.

1. **Automated Data Management**

Efficient data entry and validation.

1. **Robust Security**

User authentication and data protection.

1. **Scalability**

Ability to handle growth in users and data.

1. **High Performance**

Fast and reliable operation.

1. **Remote Access**

Manage the system from any location.

1. **Reporting and Analytics**

Generate accurate and customizable reports.

1. **Integration**

Compatible with PHP, Next.js, and MySQL.

1. **Support and Maintenance**

Ongoing technical support and system updates.

##### **5.5 Identifying External Entities**

1. **Students/Residents**

Register for accommodation and access facility information.

1. **Hostel Administrators**

Manage room assignments, student registrations, and facilities.

1. **Parents/Guardians**

View student accommodation details.

1. **University/Institution Management**

Oversee hostel operations and access reports.

1. **IT Support Staff**

Provide technical support and maintenance for the system.

###### **5.6 Context Level Data Flow Diagram**

**5.7 Capture “shall” Statements**

#### **Table 5. 1**

|  |  |  |
| --- | --- | --- |
| **Para #** | **External Entity** | **Initial Requirement** |
| 1.0 | User | User shall Login to use the web application. |
| 1.0 | System | The system shall maintain the record of the user. |
| 1.0 | System | The system shall authenticate the user. |
| 1.0 | System | The system shall connect to Wi-Fi. |
| 1.0 | User | User shall be able to send the notifications to devices. |
| 1.0 | User | User shall be able to check recent notifications. |
| 1.0 | User | User shall be able to send notifications to the system from devices. |
| 1.0 | User | User shall be able to edit their profile. |
| 1.0 | System | The system shall maintain the user profile record. |
| 1.0 | System | The system shall allocate rooms to students. |
| 1.0 | User | User shall register and manage student profiles. |
| 1.0 | User | Use shall track attendance and disciplinary records. |
| 1.0 | User | User shall manage hostel facilities and schedules. |
| 1.0 | User | User shall access the system remotely. |
| 1.0 | System | The system shall generate reports for management review. |
| 1.0 | System | The system shall ensure data security through encryption and access controls. |
| 1.0 | System | The system shall provide backup and recovery features. |

**5.8 Allocate Requirements**

#### **Table 5. 2**

|  |  |  |
| --- | --- | --- |
| **Para #** | **Initial Requirements** | **Use Case Name** |
| 1.0 | A user shall login to the web application. | Login |
| 1.0 | The user shall be able to open and view their profile in the web application. | User profile |
| 1.0 | User shall view the current status of their room and facilities. | View current status |
| 1.0 | User shall be able to edit their profile information. | Edit profile |
| 1.0 | User shall register for a room using their student ID. | Room registrations |
| 1.0 | User shall receive notifications from the system. | Notifications |
| 1.0 | User shall be logout from the web application. | Logout |

**5.9 Prioritize Requirements**

#### **Table 5. 3**

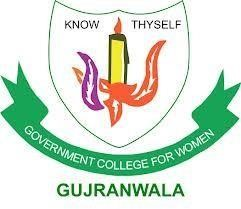
|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Para #** | **Initial Requirements** | | **Use Case**  **Name** | | **Use Case ID** | | **Rank** | |
| 1.0 | A user shall login to the web application. | | Login | | UC 01 | | High | |
| 1.0 | The user shall be able to open and view their profile in the web application. | | User profile | | UC 02 | | High | |
| 1.0 | User shall view the current status of their room and facilities. | | View current status | | UC 03 | | High | |
| 1.0 | User shall be able to edit their profile information. | | Edit profile | | UC 04 | | High | |
| 1.0 | | User shall register for a room using their ID. | | Room registration | | UC 05 | | High |
| 1.0 | | User shall receive notifications from the system. | | Notifications | | UC 06 | | Medium |
| 1.0 | | User shall be logout from the web application. | | Logout | | UC 07 | | High |

# 5.10 Requirements Trace Ability Matrix

#### **Table 5. 5**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Para #** | **Initial Requirements** | **Use Case**  **Name** | **Build** | **Category** |
| 1.0 | Admin shall be able to manage rooms. | Manage rooms | UC 01 | Admin |
| 1.0 | Admin shall be able to register students into the system. | Student registration | UC 02 | Admin |
| 1.0 | Admin shall be able to manage hostel facilities. | Manage facilities | UC 03 | Admin |
| 1.0 | Admin shall be able to manage hostel facilities. | Room booking | UC 04 | User |
| 1.0 | System shall handle payments and provide receipts | Payment management | UC 05 | User / Admin |
| 1.0 | User shall be able to provide feedback through the app. | Feedback system | UC 06 | User |

## **5.11 High Level Case Diagram**



**Chapter 6**

**Use Case Description**

# Introduction

We have completed analysis of the system. So we understand the current situation of the problem domain. Now we are ready to strive for a solution for the problem domain by using object-oriented approach. Following artifacts are discussed.

1. Use case description
2. Use case diagram refined
3. Domain Model
4. Sequence Diagram
5. Collaboration Diagram
6. Operation Contracts
7. Design Class Diagram
8. Data Model

## **6.1 Use Case Description**

Following are the use cases that we have used in our project.

**Table 6. 1**

|  |
| --- |
| UC\_1 Login |
| **Brief description:**  User will log in with email. |
| **Level:**  User |
| **Primary actor:**  User |
| **Precondition:**  • User must have an internet connection. |
| **Main success scenario:**   * Connect to the internet. * Users open the Application. * View the application interface. * Click on Login. * User enters his specific email. |
| **Alternate flows:**  1.1 Connection to the internet is disconnected due to some reason.  a. Then the AP mode is automatically set.  1.2 If user is disconnected from the server,  a. Then the user will try to reconnect. |
| **Postconditions:**  App save user with email. |

## **Table 6.2**

|  |
| --- |
| **UC\_2 : User Profile** |
| **Brief description:**  User saves Profile. |
| **Level:**  User |
| **Primary actor:**  User |
| **Precondition:**   * User must have internet connection. * User must have an login first. |
| **Main success scenario:**   * Connect to internet. * Users open the Application. * View the application interface. |
| **Alternate flows:**  1.1 Connection to the internet is disconnected due to some reason.  a. Then the AP mode is automatically set.  1.2 If User is disconnected from server,  a. Th the User will try to reconnect. |
| **Post conditions:**  User saves Profile Status. |

## **Table 6.3**

|  |
| --- |
| **UC\_3 Pair device with Mobile** |
| **Brief description:**  User pair the device with mobile application using an id. |
| **Level:**  User |
| **Primary actor:**  User |
| **Precondition:**   * User must have internet connection. * User must have an Assigned device for alert |
| **Main success scenario:**   * Connect to internet. * Users open the Application. * View the application interface. * Click on Floating Button. * Fill the form. * Save device Name and ID. |
| **Alternate flows:**  1.1 Connection to the internet is disconnected due to some reason.  a. Then the AP mode is automatically set. 1.2 If User is disconnected from server,  a. Then User will try to reconnect |
| **Post conditions:**  User saves the status of paired device in database. |

## **Table 6.4**

|  |
| --- |
| **UC\_4 Send Notification** |
| **Brief description:**  User can send notifications from the App to the Device. |
| **Level**: User |
| **Primary actor:** User |
| **Precondition:**   * User must have internet connection. * User must be log in to system * User must have a Assigned device for Notification |
| **Main success scenario:**   * Users open the Application. * Click on send Notification Card. * Select Device. * Click on send Notification Button |
| **Alternate flows:**  1.1 Connection to the internet is disconnected due to some reason.  a. Then the AP mode is automatically set.  1.2 If User is disconnected from server,  a. Then the User will try to reconnect. |
| **Post conditions:**  • Alert value will be saved in the database on Button click. |

## **Table 6.5**

|  |
| --- |
| **UC\_5 User Manual** |
| **Brief description:**  User will see the steps of the using the App with Device. |
| **Level:** User |
| **Primary actor:** User |
| **Precondition:**  • User must have an internet connection. |
| **Main success scenario:**   * Connect to the internet. * Users open the Application. * Click on the Setting. * Click on the User manual Button. |
| **Alternate flows:**  1.1 Connection to the internet is disconnected due to some reason.  a. Then the AP mode is automatically set.  1.2 If User is disconnected from the server.  a. Then the User will try to reconnect. |
| **Postconditions:**  • User will Read the Steps. |

## **Table 6.6**

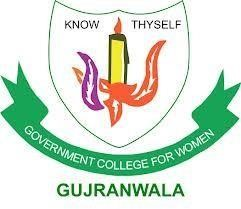
|  |
| --- |
| **UC\_6 Child Notification** |
| **Brief description:**  Receive Notification from the device. |
| **Level**: User |
| **Primary actor:** Device user |
| **Precondition:**   * User must have internet connection. * User must be log in to system. * User must have a Assigned device for Notification. |
| **Main success scenario:**   * Connect to internet. * Users open the Application. |
| **Alternate flows:**  1.1 Connection to the internet is disconnected due to some reason.  a. Then the AP mode is automatically set.  1.2 If User is disconnected from server,  a. Th the User will try to reconnect. |
| **Post conditions:**  User Receive notification from the Device. |

## **Table 6.7**

|  |
| --- |
| **UC\_7 Log out** |
| **Brief description:** Users can logout. |
| **Level**: User |
| **Primary actor:** User |
| **Precondition:**   * User must have internet connection. * User must be log in to system |
| **Main success scenario:**   * Users open the Application. * View the application interface. * Click on Logout |
| **Alternate flows:**  1.1 If Student is disconnected from server,  a. Then Student will try to reconnect. |
| **Post conditions:**  • System succeeded in logout the user. |

### **6.2. Use Case Diagram (refined and updated)**

**Figure 6.1**

****

**Chapter 7**

**Sequence Diagram**

# Sequence Diagram

## **7.1 Login**

**Figure 7.1**

### **7.2 User Profile**

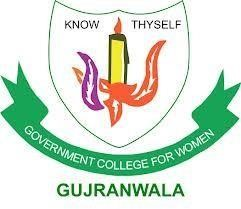
**Figure 7.2**

#### **7.3 Current Status**

**Figure 7.3**

##### **7.4 Logout**

**Figure 7.4**

****

**Chapter 8**

**Collaboration Diagram**

# Collaboration Diagram

## **8.1 Login**

**Figure 8.1**

## **8.2 User Profile**

**Figure 8.2**

#### **8.3 Current Status**

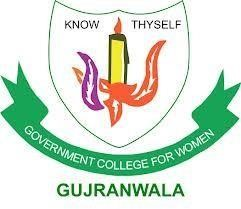
**Figure 8.3**

##### **8.4**

**Figure 8.4**

###### **8.5 Operation Contracts**

**Contract 1: Login**

****

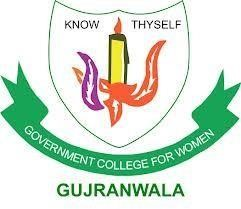
**Chapter 9**

**Domain Model**

# 9.1 Domain Model

**Figure 9.1**

# 

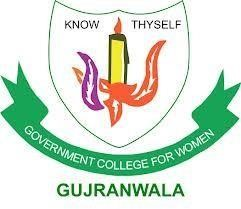


**Chapter 10**

**Class Diagram**

# 10.1 Design Class Diagram

**Figure 10.1**

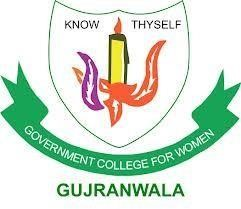
****

**Chapter 11**

**Data Model**

# 11.1 Data Model

**Figure 11.1**

****

**Chapter 12**

**Interfaces**

# 12. Project Interfaces