

SE 201.3 – Systems Analysis and Design

Group Assignment

Project Proposal *Hostel Management System*

Submission Date: 20/06/2023

Group Members:

Group No 04		
Name	ID	Degree Program
KNN Tennakoon	25324	MIS
WSU Botheju	25064	MIS
DGYR Senevirathne	25005	MIS
HMT Deeptha	25492	MIS
KTN Dharmadasa	24920	SE
JGH Amalkith	26213	CS
SS Gamage	26027	CS
KGM Sisira	25312	CS

Contribution to the project

Topic:

- Hostel management system

Project report:

Student ID	Task/ Contribution
<ul style="list-style-type: none">• JGH Amalkith - 26213• KNN Tennakoon - 25324	<ol style="list-style-type: none">1. Introduction to the system.2. Feasibility analysis<ul style="list-style-type: none">- Technical Feasibility- Operational Feasibility- Schedule Feasibility- Economic Feasibility
<ul style="list-style-type: none">• DGYR Senevirathne - 25005• SS Gamage - 26027	<ol style="list-style-type: none">1. Followed methodology.<ul style="list-style-type: none">- Agile methodology2. Testing plane<ul style="list-style-type: none">- Security testing- Implementation & Software testing- Steps in using for the software testing.- Special category of the testing
<ul style="list-style-type: none">• KTN Dharmadasa - 24920• WSU Botheju - 25064	<ol style="list-style-type: none">1. Requirement collection methods2. Requirements<ul style="list-style-type: none">- Business- User- Functional- Non-functional
<ul style="list-style-type: none">• HMT Deeptha - 25492• KGM Sisira - 25312	<ol style="list-style-type: none">1. UI – design<ul style="list-style-type: none">- User interface (Figma)- Admin interface (Figma)2. Graphical illustrations<ul style="list-style-type: none">- DFD diagrams- User-case diagrams- Class diagrams- ER diagrams.

Table of Contents

1.	Introduction to the system.....	6
1.1)	Introduction of our project	6
1.2)	Reason for the selection of our project.....	6
1.3)	Purpose of the project.....	6
1.4)	Target Users.....	6
2.	Feasibility analysis.....	6
2.1)	Technical Feasibility	6
2.2)	Operational Feasibility	7
2.3)	Schedule Feasibility	7
2.4)	Economic Feasibility.....	8
2.5)	Development Costs	9
2.6)	Operational Costs	9
2.7)	Tangible Benefits.....	9
2.8)	Intangible Benefits	9
3.	Followed methodology.....	10
3.1)	Agile methodology.....	10
4.	Requirement collection methods.....	12
4.1)	Requirements.....	13
4.1.1)	Business Requirements.....	13
4.1.2)	User.....	14
4.1.3)	Functional Requirements.....	15
4.1.4)	non-functional Requirements	16
5.	Graphical illustrations of the system.....	17
5.1)	User-case diagram	17
5.2)	ER-Diagram	18
5.3)	Class- Diagram	18
5.4)	DFD Diagram.....	19
6.	UI designs	20
6.1)	User/Student UI (Figma).....	20
6.2)	Admin UI (Figma).....	22
7.	Testing plan.....	24
7.1)	Security Testing of the Project	24
7.1.1)	White Box Testing	24

7.1.2) Black Box Testing.....	24
7.1.3) Alpha Testing	25
7.1.4) Beta Testing	25
7.1.5) Unit Testing	25
7.1.6) Integration Testing	25
7.1.7) Validation Testing	25
7.1.8) System Testing.....	25
7.2) Implementation and Software Specification Testing.....	26
7.2.1) Detailed Design of Implementation.....	26
7.2.2) Technical Design	26
7.2.3) Test Specifications and Planning	26
7.2.4) Programming and Testing.....	26
7.2.5) User Training	26
7.2.6) Acceptance Test	26
7.2.7) Review Phase.....	27
7.2.8) Development Recap.....	27
7.2.9) Post-Implementation Review	27
7.3) The steps in using for the software testing.	27
7.4) Tests in the special category	30
7.4.1) Peak Load Test.....	30
7.4.2) Storage Testing	30
7.4.3) Performance Time Testing	30
7.4.4) Recovery Testing	30
7.4.5) Procedure Testing	30
7.4.6) Human Factors Testing	30

List Of Figures

Figure 3. 1:Agile methodology	10
Figure 3. 2: Agile Development.....	11
Figure 3. 3: Hostel Management System	12
Figure 3. 4: Use Case Diagram of HMS	13
Figure 5. 1: User-case diagram	17
Figure 5. 2: ER Diagram	18
Figure 5. 3: Class Diagram	18
Figure 5. 4: DFD Diagram	19
Figure 7. 1: Principle of Security Testing	24
Figure 7. 2: System Testing.....	26
Figure 7. 3: Test Results Scripts.....	27
Figure 7. 4: Software Testing Types	29

List Of Tables

Table 2. 1: Workable Schedule.....	8
Table 2. 2: Cost Analysis	9

1. Introduction to the system

Hostel management system

1.1) Introduction of our project

We planned to make a system for private boys' hostel outside near NSBM green university. So, the hostel management system is an essential tool for the efficient management and administration of hostels. Having identified this requirement, we intend to create a system for this.

1.2) Reason for the selection of our project

There are no hostel facilities for boys at NSBM Green University. Therefore, we realized that the hostel facilities for boys near the university are provided by other outsiders and there is no proper operating system for them. Therefore, to create an operating system, we took the M&M Boy's hostel, owned by a doctor, where a friend of ours is staying.

1.3) Purpose of the project

As the existing manual system is less efficient, the solution is to make the existing system an automated system and increase efficiency. Thus, to keep the information of the current boys in the dormitory in an orderly and safe manner, as well as to facilitate the data of the payment activities properly and those payment activities. Making an operational plan for the days of students' stay and the times of their departure and return on those days as well as special notices to be given to the students by the owner of the hostel.

1.4) Target Users

- NSBM Green University Male students
- Outside Boys Hostel Owners
- Parents
- Hostel Staff

2. Feasibility analysis

2.1) Technical Feasibility

The owner of the m and m boys' hostel has requested a hostel management system from us. He asked us for an effective and secure way to easily provide personal details of students in hostels, departure and return time, hostel stay time, payment methods, and related issues and special announcements.

At present, they maintain data and documents like personal information of students in the hostel through manuals and files. So, there is a critical need for a web-based system where the entire process is digitalized because the existing manual system is ineffective, and the applicant data is not protected thoroughly. So, the proposed system will be created utilizing the Agile development method because

most of its requirements have been obtained and there are only 40 days to implement and hand over the system.

Our technical team can create a user-friendly, multilingual graphical interface that can be used for the convenience of the applicants to provide these services through our proposed system. As a result, Photoshop designs and Figma prototype technologies are used to create the user interface.

The system's front end would be created using well-known programming languages like JavaScript, CSS, HTML, and Bootstrap; however, the system's back end would consist of databases that would need to be maintained. As a result, MySQL databases are used to store data in the databases where the developers will code the data using the PHP programming language. Thus, implementing a suggested hostel system can be concluded as technically feasible.

2.2) Operational Feasibility

Currently, files and books are used to store information in the hostel or for the operational process. Physical storage has to be separated from the hostel office to store those things. As a result, much time is spent receiving urgent information. Sometimes it is difficult to find the necessary information or information may be lost. Because materials like paper can degrade over time. Therefore, some problems may arise.

In that regard, our proposed system has provided answers to all those inquiries. First, we identified the client requirements and resources needed to implement this project and we analyzed the existing systems and processes in related organizations. So, our hostel management system is an online system, the data entry process can be conducted by the client so easily, and the students can also fill out their personal information with file attachments if necessary. This will reduce data entry errors that may occur during manual data entry by the hostel staff. Some students may have missed some information while giving some information to the students, and such problems have been solved through this system.

This makes it possible to do many manual tasks in the hostel easily and efficiently. Therefore, any information in the hostel is provided in a very short time whenever required, operational possibilities are achieved by our web-based online system.

2.3) Schedule Feasibility

The M&M Boy's Hostel currently running the manual hostel system is upgraded to an online web-based system at the owner's request. Therefore, a feasibility analysis regarding the time schedules must be conducted where the probability of a workload given to a technical team being feasible can be identified. With the owner providing 40 days, a workable schedule is provided below mentioning the tasks that should be completed corresponding to their time slots.

Table 2. 1: Workable Schedule

System Task	Time Span
1. Project Management and Planning Phase 1.1. Planning 1.1.1. Identification of Background and Scope 1.1.2. Feasibility Report 1.1.3. Gantt Chart 1.1.4. Project Work Plan	-02 Days -03 Days -01 Day -01 Day
2. Project Management and Planning Phase 2.1. Analysis 2.1.1. System Proposal 2.2. Requirements 2.2.1. System Requirement Specification (SRS)	-05 Days -05 Days
3. Project Management and Planning Phase 3.1. Front-End Developing 3.2. Back-End Developing	-03 Days -04 Days
4. Implementation Phase 4.1. Final Checking and Post System 4.1.1. Executing and Control 4.1.1.1. Testing and Debugging 4.1.1.2. Final Touches 4.1.1.3. Final Testing 4.1.2. Close up and Hand Over 4.1.2.1. Knowledge Training 4.1.2.2. Handing Over	-03 Days -01 Days -02 Days -04 Days -01 Days

The above table denotes the time allocations, so the project can be finished within 35 days. For that reason, this project is feasible in terms of scheduling.

2.4) Economic Feasibility

Our technical team consists of 08 members, and they dedicated their time to the success of this project. Considering this new system, the cost has been analyzed as follows,

Table 2. 2: Cost Analysis

Description of the Cost	Cost (Rs.)
Front-End Developer	20,000
Back-End Developer	40,000
Other Implementations	40,000
Total Cost	100,000

So, this assessment typically involves a cost-benefit analysis of this project to help determine the viability, cost, and benefits associated with a project before financial resources are allocated. Thus, we can say the above cost analysis is acceptable for our client and we can also get an acceptable profit, so we can get this proposed project is economically feasible.

2.5) Development Costs

Wages of the technical team.

- The consultant fees and the software fees.
- Cost for conducting a development training program for the hostel staff.
- For Third-Party Integrations (payment gateways, accounting software)
- Maintenance and Support (This can include bug fixes, updates, security patches, and customer support.)

2.6) Operational Costs

- Costs for the upgraded hardware and software components.
- Costs for Data Backup and Security
- Training and User Support
- Cost for Technical Support and Marketing

2.7) Tangible Benefits

- The labor cost of the department reduces.
- Reduction in the usage and expenses for paper materials.

2.8) Intangible Benefits

- This streamlines accounting processes, reduces the chances of financial errors, and provides better visibility into the hostel's financial performance.
- This increased efficiency can save time and reduce errors, leading to smoother operations and improved staff productivity.
- The system can facilitate better communication between staff members, students, and the owner.
- Automation and digitization of hostel operations through a management system can significantly reduce human errors and inconsistencies.

3. Followed methodology.

The development methodology that you followed to develop the system and how you applied the methodology (agile methodology is recommended)

Our group decided to design software for the hostel. The hostel system can have various features and capabilities. That depends on what we design for the software. Using this system we can provide several tasks, basically student registration, monthly payment, announcement, student attendance is some of functional requirements We hope to cover designing for this system.

In here we discuss the development methodology that followed to develop the system and how can we have applied to our system. so basically, we used agile methodology to implement our system.

So first what is agile methodology: The Agile method divides a project into many parts and uses these phases to manage the project. Continuous improvement at every stage and ongoing collaboration with stakeholders are required. Once the work begins the teams' cycle through a process of Planning, Executing, and Evaluating.

3.1) Agile methodology



Figure 3. 1:Agile methodology

- A group of programming – certain methodologies that focus on streamlining the SDLC.
- In agile methodology includes face to face communication. So that a good mutual relationship is built between the customer and the developer. Therefore, the entire requirements and inputs can gather from customers. It was very useful to us since we can ask from hostel management what are there need, what are the development requirements that they required like wise.

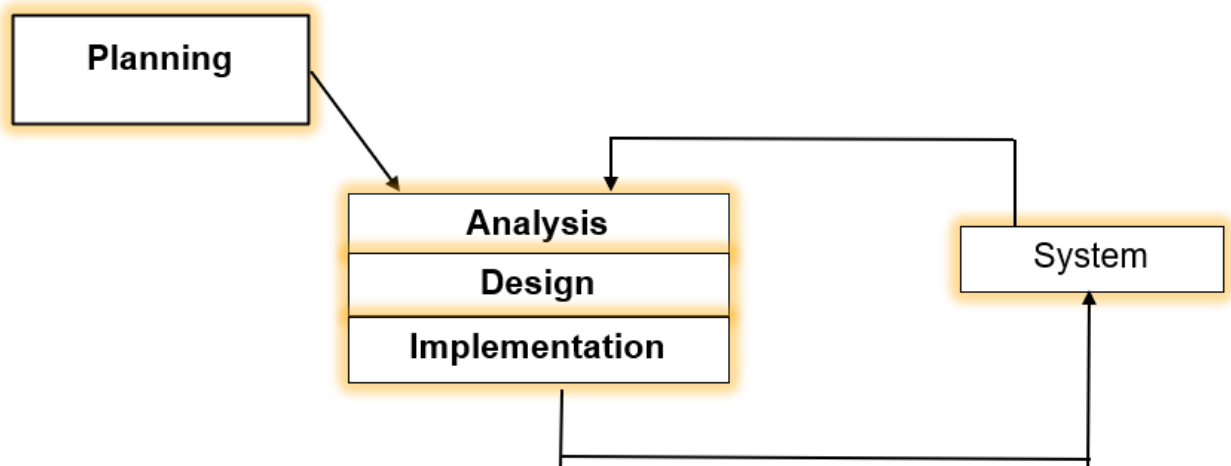


Figure 3. 2: Agile Development

All process is end of within 3 to 4 weeks. Every term was allocated to a special location. Every day we have scrum meetings. (When we stopped yesterday, what we have to do and collaborate. Sprint is one circle of our work. When we use agile methodology, we can quickly adapt with the system. because if we have to change some requirement, we can change it into the next sprint.

For the development of the hostel management software, we recommend following the Agile methodology. Agile is a flexible and iterative approach that allows for incremental development, collaboration, and adaptability to changing requirements. Here's an overview of how we can apply the Agile methodology to develop the system: Hostel management system is a software application that helps in managing room reservation, check-in/check-out, billing, inventory management and much more in a hostel. Developing such a system requires a well-defined development methodology to ensure that the system is developed efficiently and effectively. An agile methodology is recommended for developing a hostel management system due to its flexibility, iterative nature, and ability to adapt to changing needs.

1. **Define User Stories:** Begin by identifying and composing user stories to represent the functional requirements. For example, user stories could be "As a student, I want to register for accommodation" or "As a staff member, I want to mark student attendance."
2. **Prioritize and create a product backlog:** Arrange the user stories according to importance and system value. Make a list of all the tasks necessary to implement the user stories, or a product backlog.
3. **Sprint planning:** Divide the development process into brief sprints, which are iterations that run 2-4 weeks. Choose a number of high-priority user stories from the product backlog that the team will concentrate on during the sprint during the sprint planning meeting.
4. **Sprint Execution:** During each sprint, the development team works on implementing the selected user stories. Encourage collaboration and frequent communication among team members. At the end of

each day, hold a short daily stand-up meeting to discuss progress, challenges, and plan for the next day's work.

5. Continuous integration and testing: Put in place a continuous integration mechanism to make sure that the code alterations made by various team members merge without a hitch. Create automated tests to confirm the system's functionality and identify any regressions.
6. Sprint Review and Retrospective: At the end of each sprint, hold a sprint review meeting to demonstrate the completed functionality to stakeholders and gather feedback. Conduct a sprint retrospective meeting to reflect on the sprint process, identify areas for improvement, and adjust the development approach as necessary.
7. Iterative Development: Repeat the sprint cycle, incorporating feedback and new requirements. Continuously refine and update the product backlog based on changing priorities and emerging needs.

Throughout the development process, maintain close collaboration and regular communication with stakeholders, including hostel administrators, staff, and students. This will ensure that the software meets their needs and expectations.

Overall, the Agile methodology provided a flexible and efficient approach to developing a hostel management system. It perfectly accommodated continuous collaboration between our development team and stakeholders, which helped us deliver a system that met requirements and was of high quality.

4. Requirement collection methods

The process of acquiring and recording stakeholder needs and expectations for a software system is known as requirement collection. The requirements gathered provide the basis for the design, development, and testing of the software. Here are various ways to gather requirements using a hostel management system:



Figure 3. 3: Hostel Management System

1) Interviews: Hold one-on-one or small-group discussions with hostel managers, students, and other interested parties to learn more about their requirements, preferences, and expectations for the hostel management system. You might enquire about the difficulties they now have, the features they would like to see in the system, and how they would like to communicate with it.

2) Create a poll to solicit opinions from many different stakeholders. Questions concerning the present hostel management system, issues they are having, and characteristics they would want to see in a new system can all be included in the survey.

3) Workshops: Arrange a meeting of hostel managers, students, and other interested parties to debate and decide on requirements. In addition to encouraging collaboration, this approach can be helpful in detecting requirements that clash.

4) Observations: Look for any usability problems or potential areas for improvement while seeing the present hostel management system in use. To determine the students' needs and requirements, you can also watch how they behave and interact with the current system.

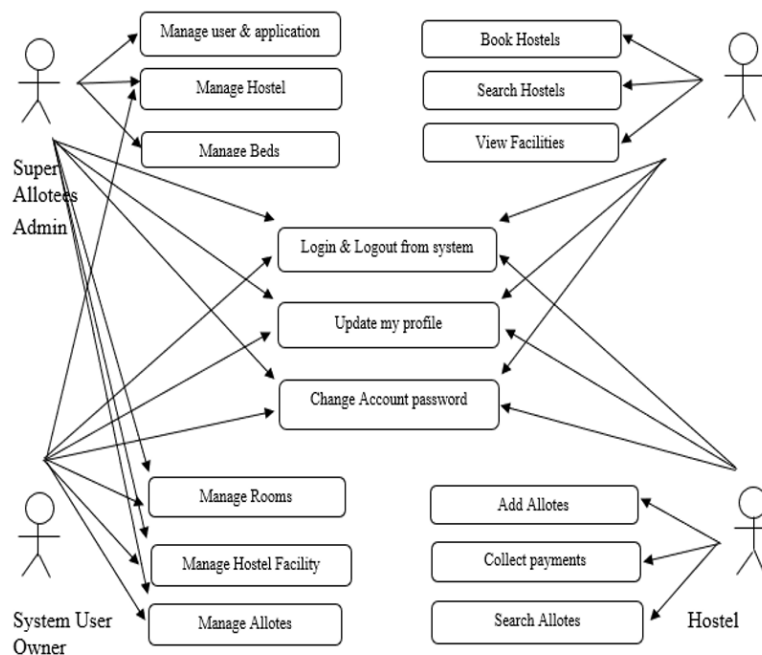


Figure 3. 4: Use Case Diagram of HMS

5) Create a functional prototype of the hostel management system to get input from stakeholders. This technique can be helpful in locating needs that are unclear or poorly specified. You can make a high-fidelity prototype to test particular features or a low-fidelity prototype to get input on the overall concept.

6) Analysis of Existing Documents: To determine the needs, examine the documentation that is already in existence, including staff reports, student feedback, and hostel policies. Using this technique, you can comprehend the current system and spot its flaws.

Only a handful of the techniques for gathering requirements for a hostel management system are included here. The project scope, the stakeholders engaged, and the available resources are some examples of variables that affect the approach chosen.

Google form: <https://forms.gle/d1a54TiQSx1qJS7j8>

4.1) Requirements

4.1.1) Business Requirements

Business needs for a hostel management system could include:

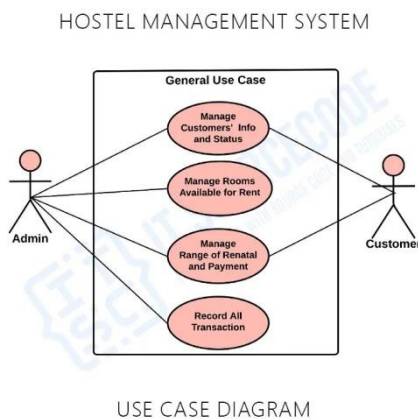
1. Reservation Management: The system ought to enable visitors or students to book hostel rooms online. Additionally, it needs to make it possible for hostel managers to manage and keep an eye on the availability of rooms, assign rooms, and handle payments.

2. Hostel administrators should be able to manage requests for room maintenance, room transfers, and room assignments via the system. It ought to offer a mechanism to monitor cleaning schedules, vacancies, and room occupancy.
3. Check-in and Check-out: The system should allow visitors or students to enter and exit the hostel. Administrators should be able to check guests' identities and collect money for extra services.
4. Billing and Payment: The system should give hostel managers a mechanism to create bills, invoices, and receipts for payments made by visitors or students. It should also enable the processing of payments and the monitoring of unpaid amounts.
5. Inventory management: The system should give hostel managers a mechanism to keep track of their stock of items including food and drink, linens, and amenities. It should also generate purchase orders for restocking and send notifications when inventory levels are low.
6. Reporting and Analytics: To assist administrators in tracking important performance indicators including occupancy rates, revenue, and expenses, the system should offer reporting and analytics functionalities. It needs to offer information about the preferences and conduct of the visitors or students.
7. Access Control and Security: The system needs to have access control and security capabilities to guarantee that only authorized users can access it. To guard against data loss, it must also offer backup and recovery techniques.

These are but a few of the operational specifications for a hostel management system. The precise specifications will be determined by the hostel's size and scope, as well as by the demands and tastes of the many parties involved.

4.1.2) User

A hostel management system may have the following user requirements:



- 1) User-friendly Interface: Even for non-technical users, the system should have an interface that is simple to use and navigate.
- 2) Students or visitors should be able to reserve hostel rooms online thanks to the system. The reservation procedure should be straightforward and simple to use, with obvious prompts and instructions.
- 3) Student or visitor online reservations for hostel rooms should be supported by the system. With straightforward steps and prompts, the reservation procedure ought to be simple and simple to utilize.
- 4) Room Selection: The system should enable visitors or students to browse the various rooms and choose the one that best suits their requirements. Additionally, it must include comprehensive details on each room, including the amenities and features offered.
- 5) Check-in and Check-out: The system should enable quick and simple check-in and check-out of the hostel for students or visitors. Wait times should be kept to a minimum and the process streamlined.
- 6) Multiple payment methods, including credit card, debit card, and mobile payments, should be supported by the system. Additionally, a safe and secure payment processing mechanism ought to be offered.

- 7) **Mobile Access:** The system ought to offer a mobile-friendly user interface so that visitors or students can access it using their mobile devices. This can be a responsive website or a mobile app.
- 8) **Personalization:** The system should let users alter many aspects of their experience, such as the language they like to use, the preferences they have for their rooms, or the extra services they would like to add.
- 9) **Feedback and Complaints:** The system ought to give visitors and students a way to express their opinions and lodge grievances during their stay. The system should also provide prompt resolution of any problems that could emerge.
- 10) **Security and privacy:** To protect the personal information of students or visitors, the system should include strong security and privacy measures. Additionally, it must adhere to data privacy laws like the CCPA and GDPR.

*Just a few of the user specifications for a hostel management system are listed here. The needs and preferences of the involved stakeholders will determine the specific requirements.

4.1.3) Functional Requirements

The following functional specifications for a hostel management system could apply:

- 1) **Reservation Management:** The system should enable visitors or students to book hostel rooms online. Additionally, it needs to make it possible for hostel managers to manage and keep an eye on room availability, assign rooms, and handle payments.
- 2) **Hostel managers** should be able to manage requests for room maintenance, room transfers, and room assignments via the system. Additionally, it must give users a method to monitor cleaning schedules, vacancies, and room occupancy.
- 3) **Check-in and Check-out:** The system needs to make it possible for visitors or students to enter and exit the hostel. Administrators ought to be able to confirm visitors' identities and collect money for extra services.
- 4) **Billing and Payment:** The system should give hostel managers a mechanism to create bills, invoices, and receipts for payments made by visitors or students. It should also enable the processing of payments and the monitoring of unpaid amounts.
- 5) **Inventory management:** The system should give hostel managers a mechanism to keep track of their stock of items including food and drink, linens, and amenities. It should also generate purchase orders for restocking and send notifications when inventory levels are low.
- 6) **Reporting and analytics elements** should be included in the system to assist administrators in keeping track of important performance metrics including occupancy rates, revenue, and expenditures. Additionally, it needs to shed light on the tastes and conduct of the visitors or students.
- 7) **Communication and Notifications:** The system must give administrators a method to get in touch with students or visitors, for example, by providing alerts, notifications, or reminders about future events, activities, or changes.
- 8) **Security and Access Control:** To guarantee that only authorized workers may access the system, the system should have security features like access control. To prevent data loss, it should also offer backup and recovery techniques.
- 9) **Integration with other systems:** For a seamless user experience, the system should be able to connect to other systems like payment gateways, messaging services, or social media.

*The functional specifications for a hostel management system don't stop at those. The particular criteria will vary depending on the hostel's size and scope, as well as the wants and preferences of the many stakeholders.

4.1.4) non-functional Requirements

The following could be non-functional needs for a hostel management system:

- 1) **Performance:** The system must be able to process many users and transactions without experiencing sluggishness or crashing. Additionally, it must offer quick responses to customer requests.
- 2) **Reliability:** To meet the needs of students or visitors, the system must be dependable and accessible around-the-clock. In the event that a system malfunctions or data is lost, it should also include backup and recovery procedures.
- 3) **Security:** To safeguard visitors' or students' personal information, the system should have strong security features. Regulations governing data protection, such as the GDPR or CCPA, should also be followed.
- 4) **Scalability:** As the hostel gets bigger, the system should be able to manage more users and transactions. It must also be able to connect to other programs or services when necessary.
- 5) **Usability:** The program should be simple to use and understandable to non-technical users as well. In order to help customers through the process, it should also offer clear directions and feedback.
- 6) **Accessibility:** Users with impairments, such as those involving vision or hearing, should be able to utilize the system. Furthermore, it must adhere to WCAG guidelines for accessibility.
- 7) The system should work with a variety of browsers, operating systems, and gadgets. Additionally, it must be able to connect to other programs or services as necessary.
- 8) **Maintainability:** The system needs to be simple to upgrade and maintain, with good documentation and code standards. Tools for testing and debugging should also be provided.
- 9) **Performance:** The system must be able to manage many users and transactions without stuttering or going offline. Additionally, it must respond to user requests quickly.

*The non-functional needs for a hostel management system don't end here. The particular criteria will depend on the wants and preferences of the participating parties.

5. Graphical illustrations of the system

Hostel management system

5.1) User-case diagram

URL: https://drive.google.com/file/d/1bVjznOSMGR0Als0-uJ2ZbgpzgKshFd6h/view?usp=drive_link

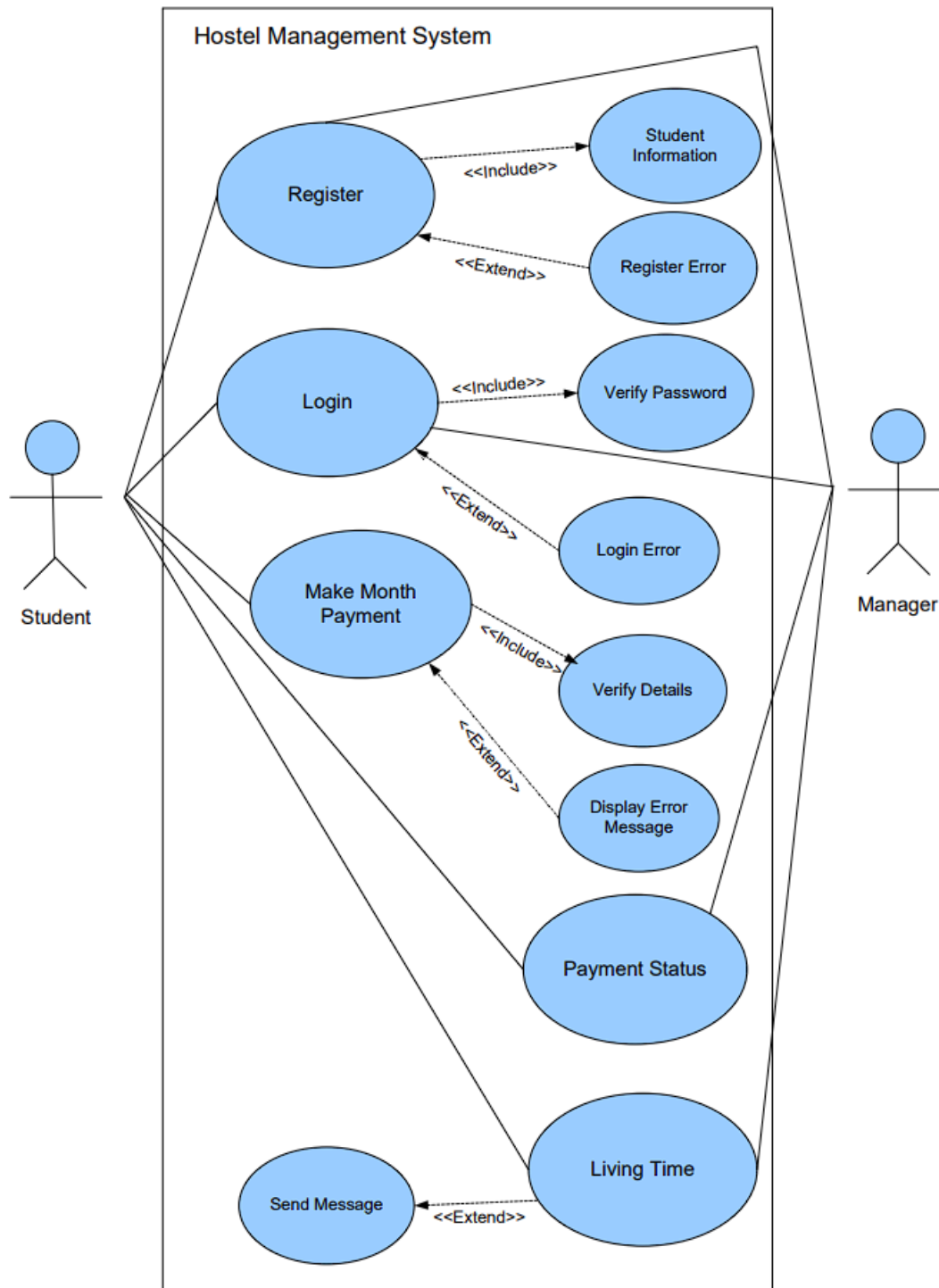


Figure 5. 1: User-case diagram

5.2) ER-Diagram

URL: https://drive.google.com/file/d/1kALufx-8hlxcqNqxEuN35wZRudt41LU8/view?usp=drive_link

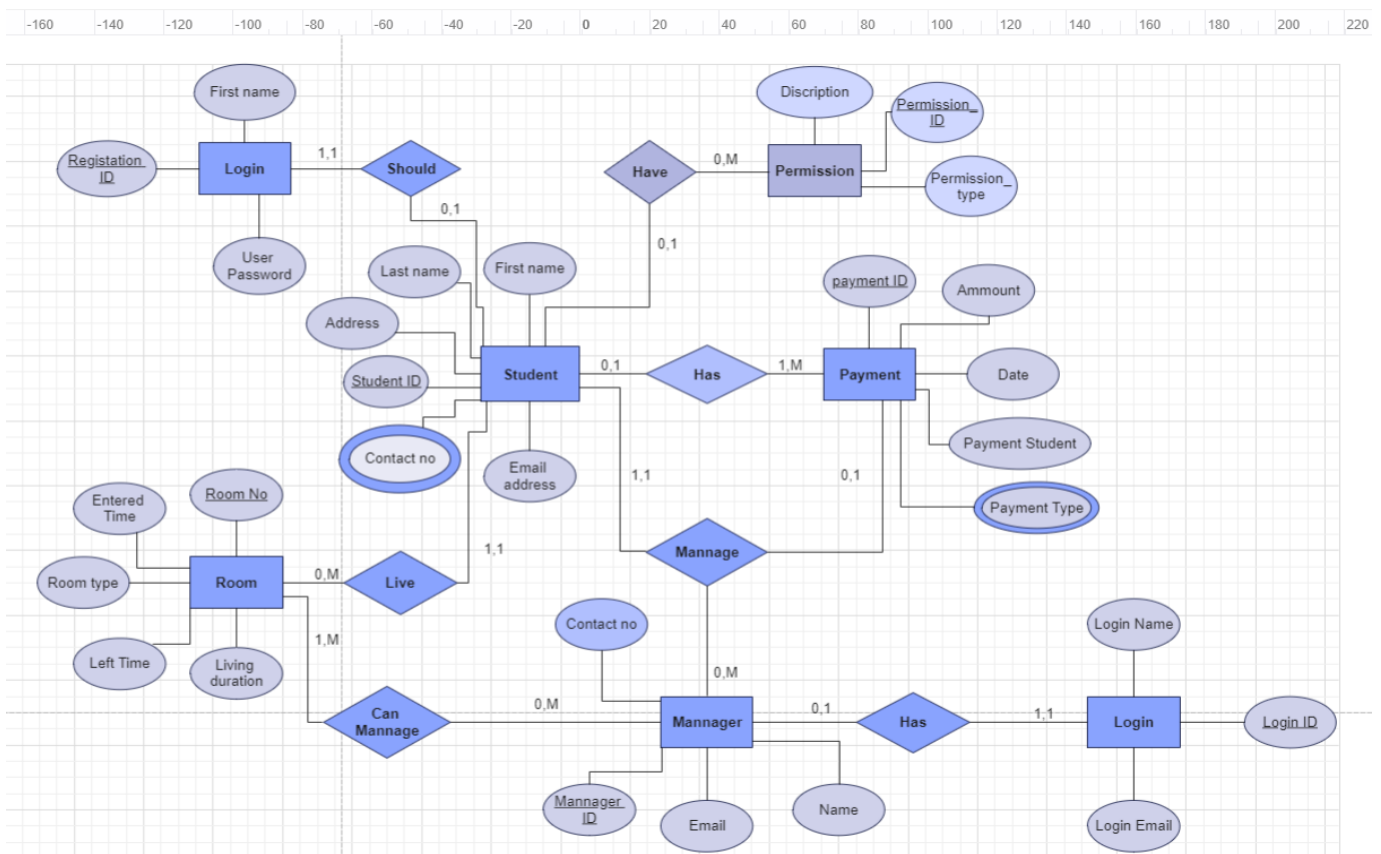


Figure 5. 2: ER Diagram

5.3) Class- Diagram

URL: https://drive.google.com/file/d/1TvNAsmhV-QITRPyKo24fVDwBolwvr1kj/view?usp=drive_link

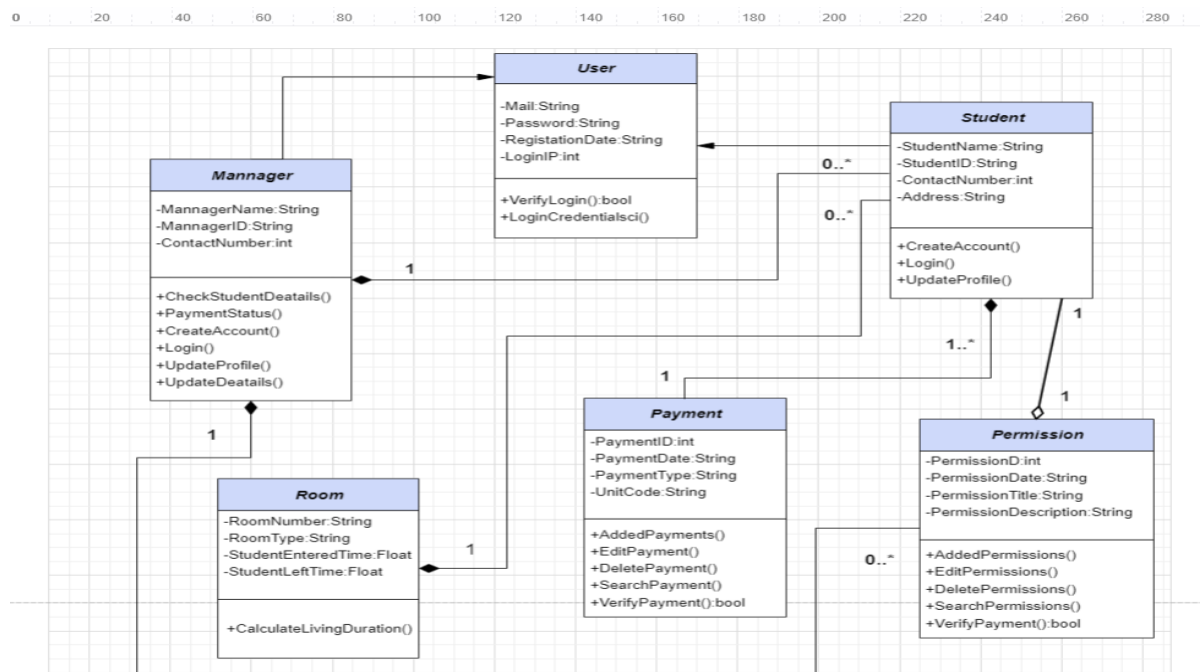


Figure 5. 3: Class Diagram

5.4) DFD Diagram

URL: https://drive.google.com/file/d/1ccORe-U_NAYGlwgFf1HTP3jUzMu-VpAn/view?usp=drive_link

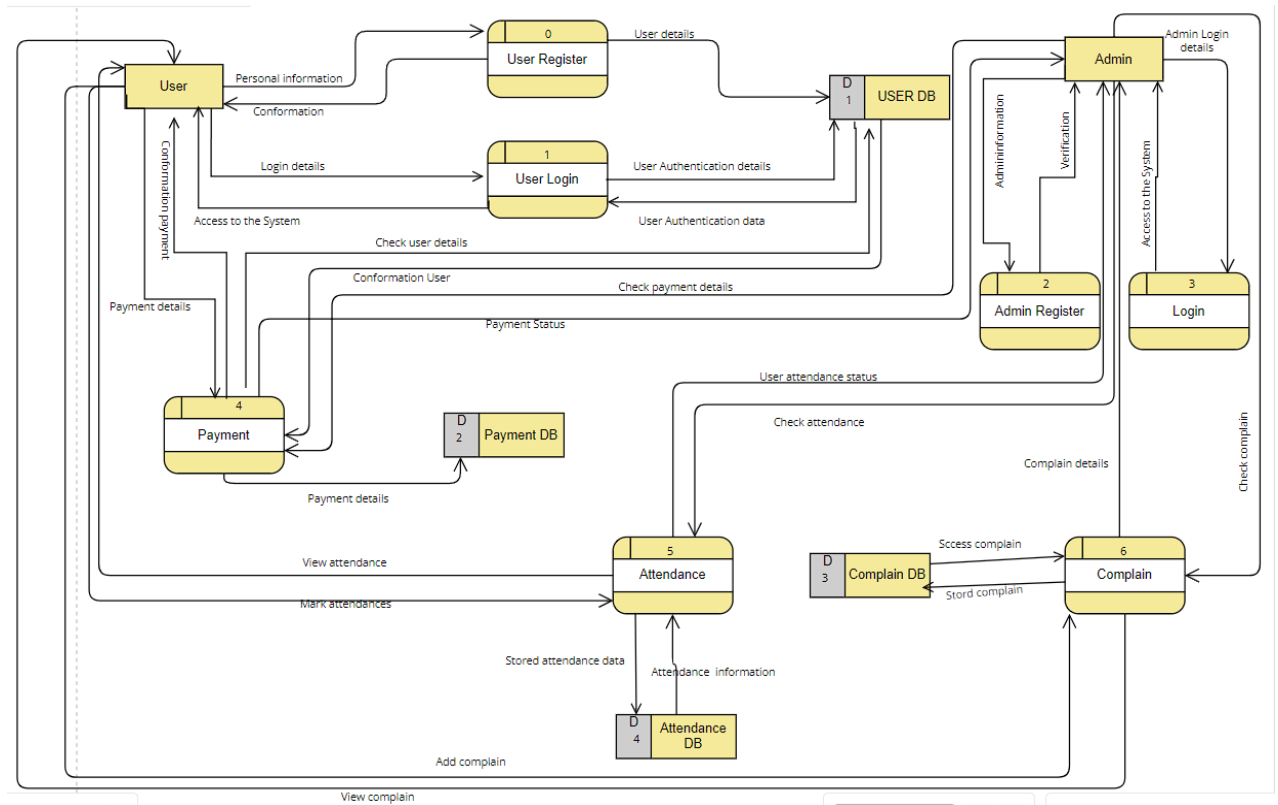


Figure 5. 4: DFD Diagram

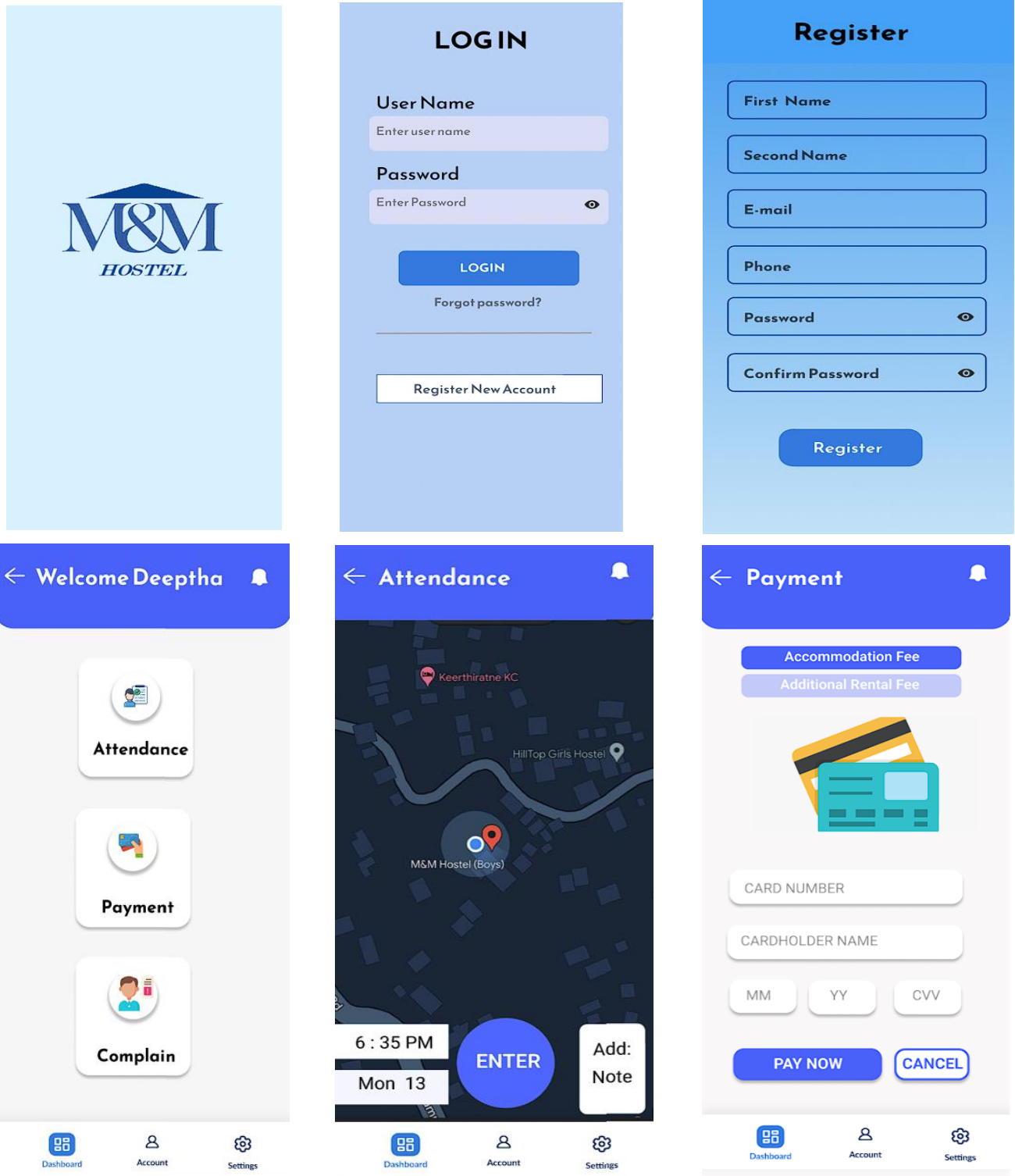
6. UI designs

Online hostel management system

6.1) User/Student UI (Figma)

URL: <https://www.figma.com/file/A1fMng5VyTEETb5laJznPB/Untitled?type=design&node-id=0-1>

(Please use this link to view full User interface & UI wireframe)



←

Complain

Your Name

Subject

Send

Dashboard

Account

Settings

←

Notification

Owner.DR. M.D Kariyawasam

Reminder: Hostel fees due this week.

16min ago

Owner.DR. M.D Kariyawasam

Alert: Hostel room inspections begin today.

6h ago

Owner.DR. M.D Kariyawasam

REMINDER: Hostel AC usage guidelines
Please ensure windows are closed and
switch off AC when leaving the room to
conserve energy.

8h ago

Owner.DR. M.D Kariyawasam

URGENT NOTICE: Hostel solar power
upgrade underway. Expect intermittent
power disruptions during the installation.
Thank you for your cooperation.

15h ago

Dashboard

Account

Settings

←

My Account

Deeptha HMT

Choose an image

Edit Profile

Full Name

Enter Full Name

Email

Enter Email Address

Contact Number

Enter your Contact Number

Enter parents contact Number

Student National id card Number

Enter National id card Num:

University Id card Number

Enter University id card Num:

Room Number

G 1

Room Num:

Cancel

Save

Dashboard

Account

Settings

←

Settings

i

Account Information

Password

Notification Settings

Support

Dashboard

Account

Settings

21 | Page

6.2) Admin UI (Figma)

URL: <https://www.figma.com/file/A1fMng5VyTEETb5laJznPB/Untitled?type=design&node-id=0-1>

(Please use this link to view full Admin interface & UI wireframe)

LOGIN

User Name
Enter user name

Password
Enter Password

LOGIN

Forgot password?

Register New Account

Register

First Name

Second Name

E-mail

Phone

Password

Confirm Password

Register

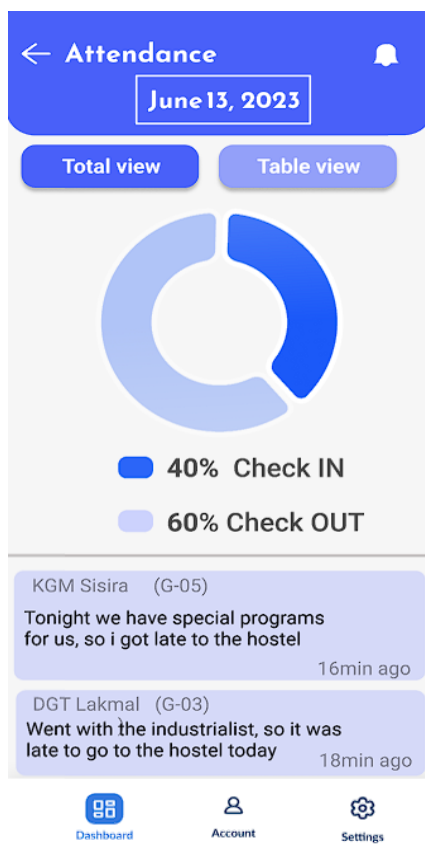
Welcome

Attendance

Hostel Fee

Announcement

Dashboard Account Settings



Hostel Fee

June - 2023

Paid List Unpaid List

Name	Room	Date
TY Gayashan	G-01	6/1/2023
WT Dilshan	G-02	6/1/2023
KGM Sisira	G-03	6/1/2023
GY Udara	G-04	6/1/2023

Dashboard Account Settings

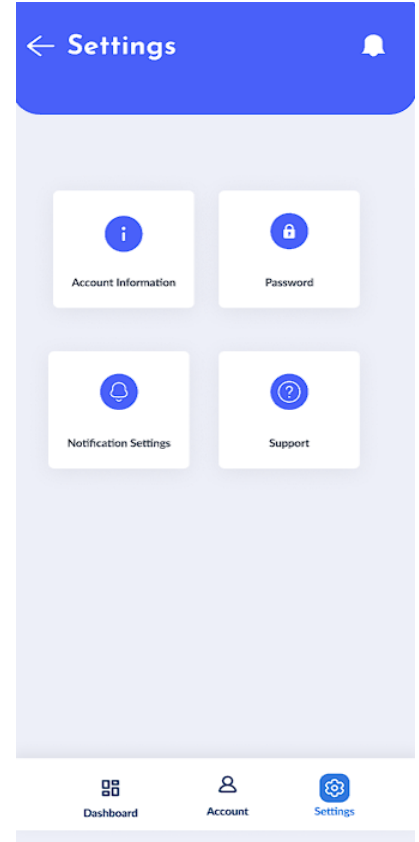
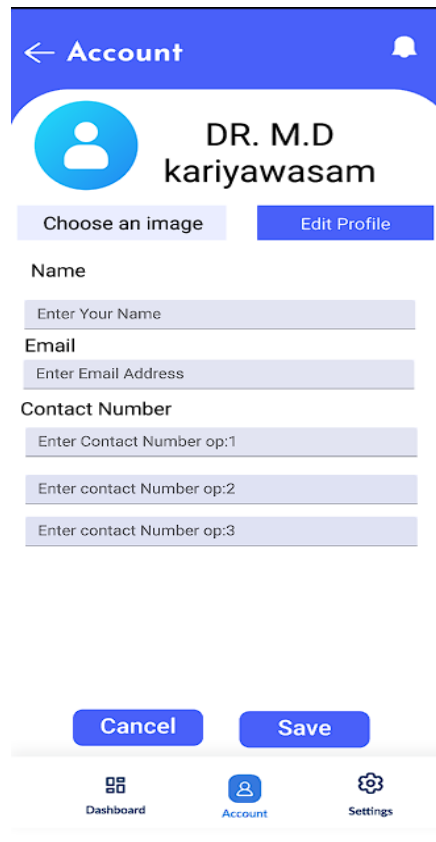
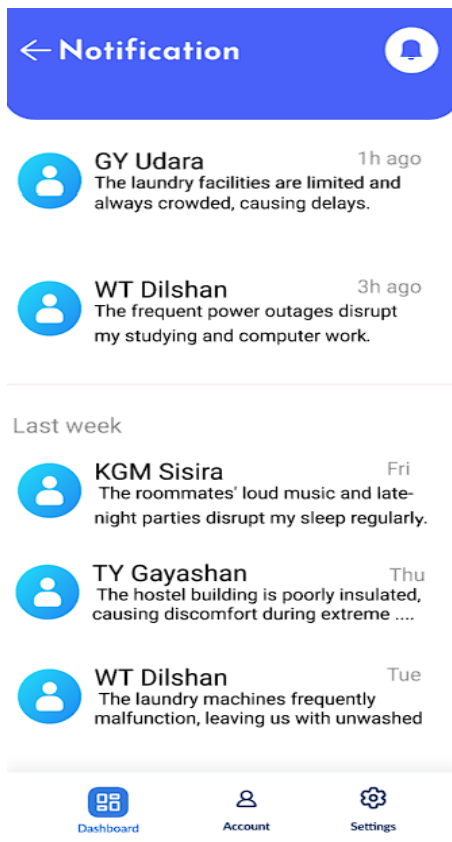
Announcement

To

Subject

Send

Dashboard Account Settings



7. Testing plan

7.1) Security Testing of the Project

Testing is vital for the success of any software. no system design is ever perfect. Testing is also carried out in two phases. The first phase is during the software engineering that is during the module creation. second phase is after the completion of software. This is system testing which verifies that the whole set of programs are hung together.

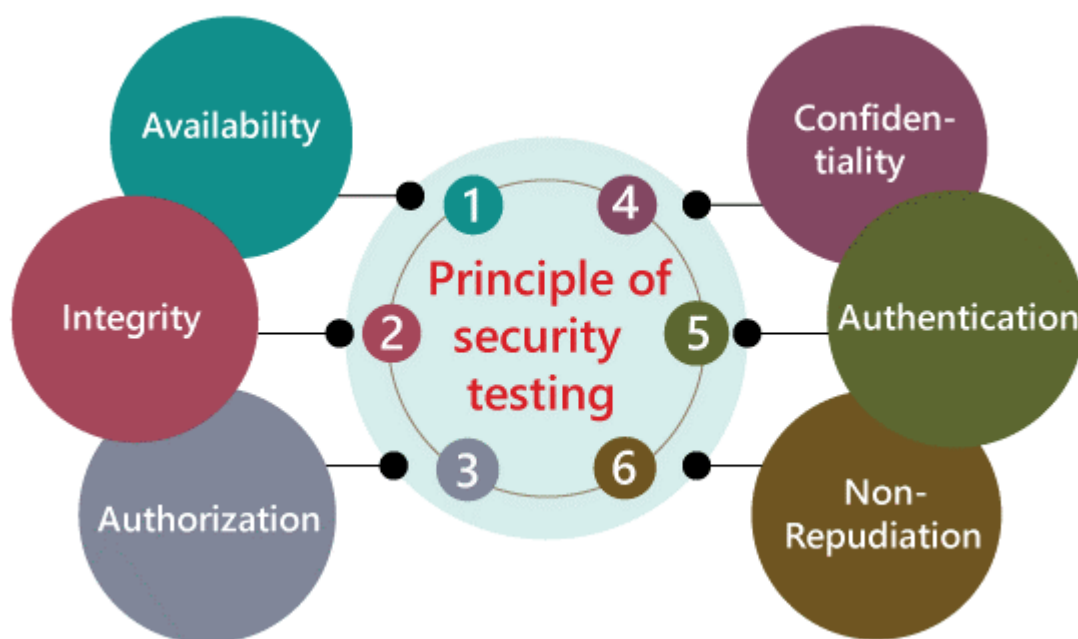


Figure 7. 1: Principle of Security Testing

7.1.1) White Box Testing

In this technique, the close examination of the logical parts through the software are tested by cases that exercise species sets of conditions or loops. all logical parts of the software checked once. errors that can be corrected using this technique are typographical errors, logical expressions which should be executed once may be getting executed more than once and error resulting by using wrong controls and loops. When the box testing tests all the independent part within a module a logical decision on their true and the false side are exercised, all loops and bounds within their operational bounds were exercised and internal data structure to ensure their validity were exercised once.

7.1.2) Black Box Testing

This method enables the software engineer to device sets of input techniques that fully exercise all functional requirements for a program. black box testing tests the input, the output and the external data. it checks whether the input data is correct and whether we are getting the desired.

7.1.3) Alpha Testing

Acceptance testing is also sometimes called alpha testing. Be spoke systems are developed for a single customer. The alpha testing proceeds until the system developer and the customer agree that the provided system is an acceptable implementation of the system requirements.

7.1.4) Beta Testing

On the other hand, when a system isto be marked as a software product, another process called beta testing is often conducted. During beta testing, a system is delivered among a number of potential users who agree to use it. The customers then report problems to the developers. This provides the product for real use and detects errors which may not have been anticipated by the system developers.

7.1.5) Unit Testing

Each module is considered independently. it focuses on each unit of software as implemented in the source code. it is white box testing.

7.1.6) Integration Testing

Integration testing aims at constructing the program structure while at the same constructing tests to uncover errors associated with interfacing the modules. modules are integrated by using the top-down approach.

7.1.7) Validation Testing

Validation testing was performed to ensure that all the functional and performance requirements are met.

7.1.8) System Testing

It is executing programs to check logical changes made in it with intention of finding errors. a system is tested for online response, volume of transaction, recovery from failure etc. System testing is done to ensure that the system satisfies all the user requirements.

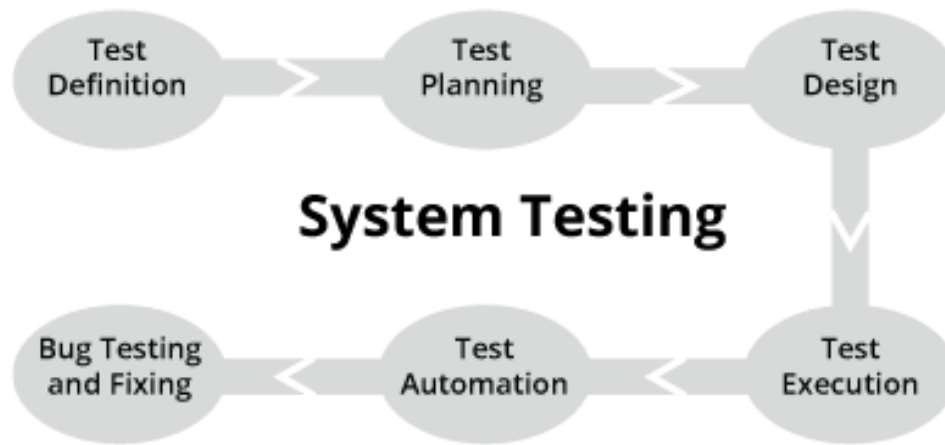


Figure 7. 2: System Testing

7.2) Implementation and Software Specification Testing

7.2.1) Detailed Design of Implementation

This phase of the systems development life cycle refines hardware and software specifications, establishes programming plans, trains users and implements extensive testing procedures, to evaluate design and operating specifications and/or provide the basis for further modification.

7.2.2) Technical Design

This activity builds upon specifications produced during new system design, adding detailed technical specifications and documentation.

7.2.3) Test Specifications and Planning

This activity prepares detailed test specifications for individual modules and programs, job streams, subsystems, and for the system as a whole.

7.2.4) Programming and Testing

This activity encompasses actual development, writing, and testing of program units or modules.

7.2.5) User Training

This activity encompasses writing user procedure manuals, preparation of user training materials, conducting training programs, and testing procedures.

7.2.6) Acceptance Test

A final procedural review to demonstrate a system and secure user approval before a system.

7.2.7) Review Phase

This phase evaluates the successes and failures during a systems development project, and measures the results of a new Computerized Tran system in terms of benefits and savings projected at the start of the project.

7.2.8) Development Recap

A review of a project immediately after completion to find successes and potential problems in future work.

7.2.9) Post-Implementation Review

A review. conducted after a new system has been in operation for some time, to evaluate actual system performance against original expectations and projections for cost-benefit improvements. Also identifies maintenance projects to enhance or improve the system.

7.3) The steps in using for the software testing. (The steps involved during Unit testing are as follows)

- I - Preparation of the test cases.
- II - Preparation of the possible test data with all the validation checks.
- III - Complete code review of the module.
- IV - Actual testing done manually.
- V - Modifications done for the errors found during testing.
- VI - Prepared the test result scripts.

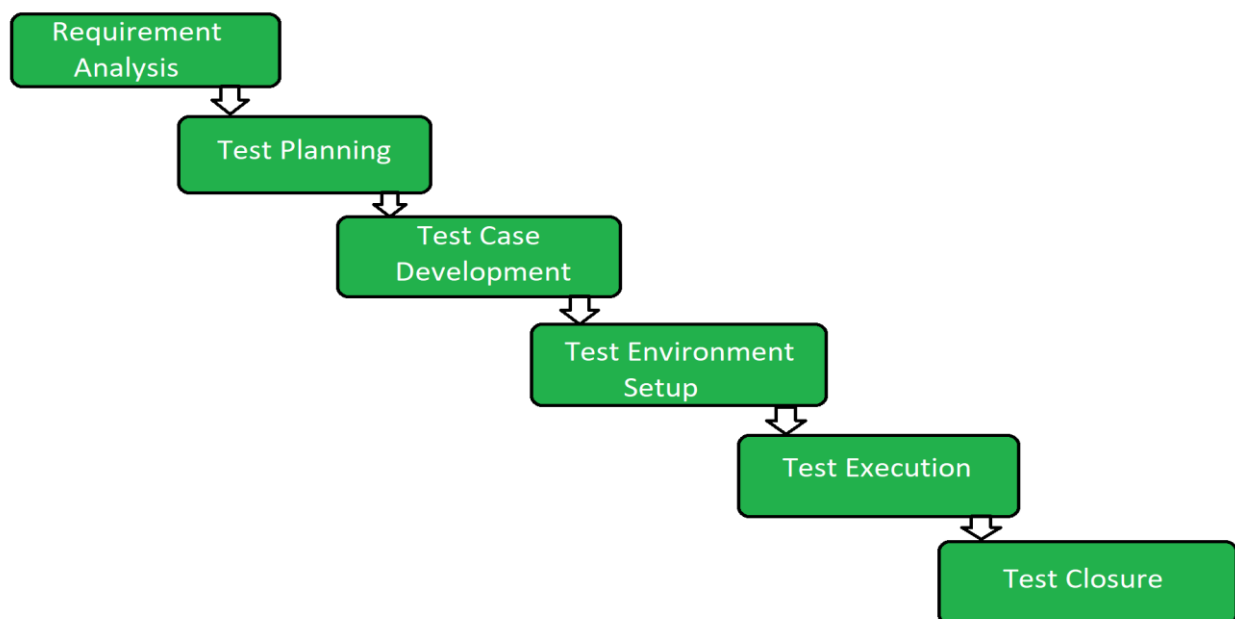


Figure 7. 3: Test Results Scripts

1. The unit testing done included the testing of the following items.

- I - Functionality of the entire module/forms.
- II - Validations for user input.
- III - Checking of the Coding standards to be maintained during coding.
- IV - Testing the module with all the possible test data.
- V - Testing of the functionality involving all types of calculations etc.
- VI - Commenting standard in the source files.

After completing the Unit testing of all the modules, the whole system is integrated with all its dependencies in that module. While System Integration, we integrated the modules one by one and tested the system at each step. This helped in the reduction of errors at the time of the system testing.

2. The steps involved during System testing are as follows.

- Integration of all the modules/forms in the system.
- Preparation of the test cases.
- Preparation of the possible test data with all the validation checks.
- Actual testing done manually.
- Recording of all the reproduced errors.
- Modifications done for the errors found during testing.
- Prepared the test result scripts after rectification of the errors.



3. The System Testing done included the testing of the following items.

I - Functionality of the entire system as a whole.

II - User Interface of the system.

III - Testing the dependent modules together with all the possible test data scripts.

IV - Verification and Validation testing.

V - Testing the reports with all its functionality.

After the completion of system testing, the next following phase was the Acceptance Testing. Clients at their end did this and accepted the system with appreciation. Thus, we reached the final phase of the project delivery.

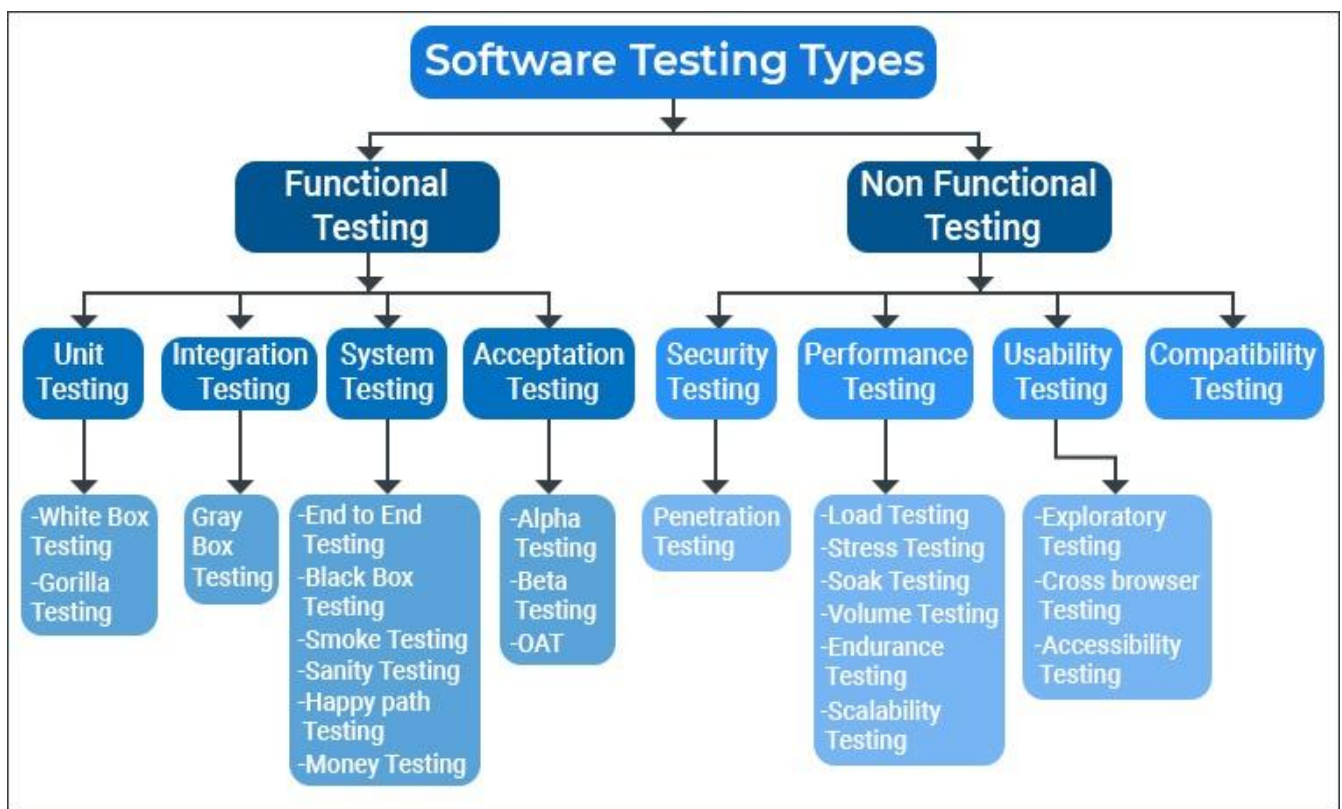


Figure 7. 4: Software Testing Types

7.4) Tests in the special category

There are six more tests in the special category. They are,

7.4.1) Peak Load Test

It determines whether the system will handle the volume of activities that occur when the system is at the peak of its processing demand. For example, test the system by activating all terminals at the same time.

7.4.2) Storage Testing

It determines the capacity of the system to store transaction data on a disk or in other files.

7.4.3) Performance Time Testing

It determines the length of time system used by the system to process transaction data. This test is conducted prior to implementation to determine how long it takes to get a response to an inquiry, make a backup copy of a file, or send a transmission and get a response.

7.4.4) Recovery Testing

This testing determines the ability of user to recover data or re-start system after failure. For example, load backup copy of data and resume processing without data or integrity loss.

7.4.5) Procedure Testing

It determines the clarity of documentation on operation and uses of system by having users do exactly what manuals request. For example, powering down system at the end of week or responding to paper-out light on printer.

7.4.6) Human Factors Testing

It determines how users will use the system when processing data or preparing reports.