**FACULTY HOUSING MANAGEMENT SYSTEM**

PROJECT MANAGEMENT PLAN

Issued by: Group-07

Version *1.0.1*

*Date:06/10/2024*

## Project Management Plan Approval

The undersigned acknowledge they have reviewed the**Project Management Plan** and agree with the approach it presents. Changes to this **Project Management Plan** will be coordinated with and approved by the undersigned or their designated representatives.

|  |  |  |
| --- | --- | --- |
| Signature: |  |  |
| Print Name: | DANIYAL KHALIL |  |
|  |  |  |
| Role: | PROJECT MANAGER |  |

|  |  |  |
| --- | --- | --- |
| Signature: |  |  |
| Print Name: | M. SAQIB TAHIQ |  |
|  |  |  |
| Role: | DEVELOPER |  |

|  |  |  |
| --- | --- | --- |
| Signature: |  |  |
| Print Name: | TALHA TARIQ |  |
|  |  |  |
| Role: | DEVELOPER |  |

# VERSION HISTORY

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Version #** | **Implemented By** | **Revision Date** | **Approved By** | **Approval Date** | **Reason** |
| 1.0.1 | *Daniyal khalil* | *06/10/2024* | *Dr. Onaiza* | *07 /10/2024* | *NIL* |
|  |  |  |  |  |  |

***PREFACE***

This project management system is designed to streamline the process of housing allocation and maintenance requests for faculty members at a university. The system's goal is to create an organized and efficient workflow that manages housing applications, seniority-based allotment, and ongoing maintenance needs with precision and transparency.

The University maintains a list of faculty housing units, constantly updated to reflect their allocated or unallocated status. The allocation process begins when a house becomes available or unallocated. Faculty members are invited to apply, and these applications are sorted based on their seniority. The system ensures that the highest-ranked applicant is offered the house, after which they must confirm their interest for the house to be officially allocated.

Beyond allocation, the system ensures that faculty members' maintenance requests are managed effectively. Maintenance requests for an allocated house are categorized, saved, and tracked to ensure timely fulfillment. Unfulfilled requests are automatically flagged and sent to the maintenance department on a daily basis until resolved, allowing for continuous follow-up on pending issues.

This system will enhance the transparency and accountability of housing and maintenance management. By automating key steps in the process, it reduces administrative burden and improves the overall experience for faculty members. With its ability to handle seniority-based allocations and provide ongoing maintenance support, the project management system aligns with the University's mission to offer seamless service and comfortable living conditions to its faculty members.

# TABLE OF CONTENTS

1. [PROJECT OVERVIEW 5](#_bookmark0)
   1. [PROJECT SUMMARY 5](#_bookmark1)
   2. PURPOSE OF A SYSTEM 5
   3. SCOPE 5
   4. OBJECTIVES OF A SYSTEM 6
   5. ASSUMPTION AND CONSTRAINTS 6
   6. PROJECT DELIVERABLES 7
   7. SCHEDULE SUMMARY 7
2. [REFERENCES 7](#_bookmark2)
3. [DEFINITIONS 8](#_bookmark4)
4. PROJECT CONTENTS 8
   1. [PROCESS MODEL 8](#_bookmark5)
   2. [METHODS TOOLS AND TECHNIQUES 9](#_bookmark6)
   3. [PRODUCT ACCEPTANCE TEST 11](#_bookmark7)
5. SUPPORTING PROCESS PLANS 24
   1. [RISK MANAGEMENT 24](#_bookmark7)
6. **PROJECT PLANNING 29**

6.1Gantt chart 29

# LIST OF FIGURE

# 1.Gantt chart

# LIST OF TABLES

1. Functional testing
2. Non-functional testing
3. Team Roles and Responsibilities
4. Entity Roles and Responsibilities

## PROJECT OVERVIEW

### PROJECT SUMMARY

A University allocates houses to faculty members based on seniority. A list of

houses are maintained by the estate office, which also maintains status of the

houses (allocated/unallocated). When a house is unallocated or becomes vacant,

the estate office invites applications according to grade. These applications are

sorted by seniority, and letters are sent to the allotted. A faculty member must

confirm his/her interest in the house in order for it to be allocated. All

maintenance requests for an allocated house are saved category wise. It is also

tracked whether the request was fulfilled. Unfulfilled requests are sent every day

to the maintenance department, till they indicate that the issue has been taken

care.

### PURPOSE OF A SYSTEM :

The purpose of the housing allocation system is to ensure fair and efficient allocation of houses to faculty members based on seniority. It streamlines application processing, tracks house status, and manages maintenance requests, ensuring timely communication and responsiveness to faculty needs. Ultimately, the system aims to enhance user satisfaction and optimize resource management.

### PROJECT SCOPE :

### CONTEXT

### It is a standalone system…

### MAIN FUNCTIONALITY

### Following are some of the functionalities that system must contain:

 **Housing Allocation Based on Seniority**  
Automates the allocation of housing to faculty members based on their seniority.

 **Application Management**  
Allows faculty members to submit housing applications and tracks their status.

 **House Status Tracking**  
Maintains a real-time record of the status of each house (allocated, un allocated, vacant).

 **Confirmation Process**  
Sends notifications to faculty members for house offers and tracks confirmation responses.

 **Maintenance Request Management**  
Enables submission of maintenance requests, categorized by type, and tracks resolution status.

 **Reporting and Analytic**  
Provides reports on house allocations, pending applications, and maintenance requests.

 **User Role Management**  
Allows different access levels for estate office personnel, maintenance staff, and faculty members.

 **Data Security and Compliance**  
Ensures secure storage of housing and maintenance data while adhering to privacy regulations.

 **Notification and Alerts System**  
Sends automated notifications about housing offers, application updates, and maintenance statuses.

### OBJECTIVE OF A SYSTEM :

### The objective of the system is to makes the process faster, reduces manual work and to provide 24 hour services to the user. It helps the estate office handle applications based on seniority, send notifications, and confirm interest in available houses.

### ASSUMPTIONS & CONSTRAINTS :

### Assumption:

* Faculty members have access to necessary technology.
* Existing data on housing and seniority is accurate
* Users will receive adequate training for the system.

### Constraints:

* The project must stay within the set budget.
* The project must be completed by the agreed deadline.
* Data privacy and security regulations must be followed during the entire process.
* System should be implemented using **JAVA** and **OOPS** principals only.
* Maximum 3 members are allowed throughout the project.

### PROJECT DELIVERABLES :

 **System Design Document**: Outlining the architecture, functionalities, and technical specifications of the system.

 **User Interface Prototypes**: Visual designs of the user interface for faculty members and estate office staff.

 **Implementation Plan**: A detailed plan for deploying the system, including timelines and resource allocation.

 **Testing Documentation**: Test plans, test cases, and results to ensure the system meets requirements and functions correctly.

 **User Manuals**: Comprehensive guides for faculty and staff on how to navigate and utilize the system effectively.

 **Maintenance Procedures**: Documentation on how to handle and escalate maintenance requests.

 **Training Materials**: Resources such as presentations and tutorials to train users on the new system.

 **Feedback Collection Tools**: Surveys or forms to gather user feedback post-implementation for ongoing improvements.

### SCHEDULE SUMMARY :

Project at a whole will take almost 16 weeks starting from 02-10-2024 to 21-1-2025.For discussion 7 days are allocated which will cover introduction to software development, projects and allocations. For **Planning** 7 days are allocated, covering project plan development and review**.** For **Analysis** 21 days which include use cases, SRS development, and refinement For **Design** 28 days are dedicated to design principles, interface, and detailed design with refinement. For **Development** 28 days are allocated covering coding and integration activities, including database management, front-end development activities, and classes development. For **Testing and presentations** 21 days are allocated covering system testing and final presentation of the system.

## REFERENCES :

The following table summarizes the documents referenced in this document.

|  |  |  |
| --- | --- | --- |
| **Document Name and Version** | **Description** | **Location** |
| IEEE 16326-2009 | *IEEE Standard for Software Project Management Plans* | *https://www.iso.org/obp/ui/fr/#iso:std:iso-iec-ieee:16326* |
| HPG-24 | University Housing Policies and Guidelines. | *www.qau.edu.pk* |
| CS\_04 | Case study 4th | *-NIL-* |

1. **DEFINITIONS :**

The following table provides definitions for terms relevant to this document.

|  |  |
| --- | --- |
| **Term** | **Definition** |
| **PMP** | Project Management Plan |
| **FHM** | Faculty Housing Management System |
| **Estate Office** | The university department managing housing and maintenance. |
| **OOPS** | Object oriented programming principal |
| **Maintenance Department** | The team responsible for resolving maintenance issues |
| **UAT** | User acceptance test |
| **PAT** | Project acceptance test |

**4 PROCESS CONTENT :**

### 4.1 PROCESS MODEL:

Incremental Model for University House Allocation System

Reasons for Choosing:

1. **Ideal for Complex Projects:** The University Housing Allocation System is quite intricate, with multiple components such as faculty registration, house management, and maintenance tracking. By using the incremental model, we can break down these components into smaller, more manageable pieces. This allows us to develop, test, and validate each part individually, significantly reducing the risk of major issues at the end of the project.
2. **Gradual Delivery of Features:** The incremental approach is perfect when we need certain system features to be delivered and tested early. For example, we could implement the house management feature first, followed by the maintenance request feature. This step-by-step process allows us to get early feedback from users, ensuring that key functionalities are in place and working well before moving on to the next phase.
3. **Flexibility to Incorporate Feedback:** Since the faculty and the estate office will be the primary users of the system, their feedback is essential. The incremental model allows us to make adjustments based on user input after each module is delivered. This flexibility means we can adapt the system to meet evolving user needs, such as changes in house allocation policies or the addition of new maintenance features, ensuring the system remains relevant and user-friendly.
4. **Reduced Development Risks:** Breaking down a complex project like the house allocation system into smaller steps significantly lowers the chances of major mistakes occurring at the end of the development process. By developing and testing each component individually, we can identify and fix issues early on, preventing them from affecting the overall system. This approach not only enhances the quality of the final product but also fosters a smoother development experience
5. **Focus on Key Features First:** The incremental approach allows the team to prioritize and implement the most critical features first, such as the seniority-based housing allocation. Once these essential functionalities are established, we can gradually introduce secondary features like automated maintenance notifications. This way, we ensure that the most important parts of the system are up and running as soon as possible, providing immediate value to users.
6. **Clear Progress Tracking:** Since each module is delivered in stages, it becomes easier for the team to monitor the project's progress and adjust timelines as necessary. For example, while developing the Faculty Registration module, we can still work on the Notification System and Maintenance Tracking modules simultaneously. This overlap ensures that delays in one area won’t necessarily hinder progress in others, allowing us to keep the project moving forward smoothly.

.4.2 **METHODS TOOLS AND TECHNIQUES :**

**Techniques**

1. **Designing with Objects in Mind (OOD):** The system’s architecture is built around the principles of object-oriented design (OOD). We treat essential entities, such as FacultyMember, House, and MaintenanceRequest, as distinct classes. Each class has its own specific methods, like assigning houses or submitting maintenance requests, as well as attributes, such as faculty grades or housing availability. This structured approach helps keep the system organized and makes it easier to manage different functionalities.
2. **Data Security through Encapsulation:** To protect sensitive information, like housing allocation statuses and personal details of faculty members, we’ll use encapsulation. This means we’ll restrict access to this data through secure methods, such as getters and setters. By doing so, we not only safeguard privacy but also ensure that we comply with security regulations, providing peace of mind for all users involved.
3. **Gradual Advancement:** The system will be developed one feature at a time, focusing on individual modules like house management and maintenance tracking. Each module will be completed and thoroughly tested on its own before being integrated into the complete system. This step-by-step approach ensures that everything functions smoothly and reduces the chances of errors when the components are connected.
4. **Polymorphism for Adaptability:** Polymorphism will allow the system to manage different types of housing—such as faculty houses, administrative houses, and more—through a single interface. This flexibility makes it easier to accommodate future enhancements or the addition of new housing categories, ensuring that the system can evolve alongside the university's needs.
5. **Code Reviews to Ensure Correctness:** We will conduct regular code reviews to ensure that everything is functioning optimally and is free from errors. These reviews will focus on critical modules, particularly the faculty seniority-based housing allocation, to confirm their accuracy and efficiency. This collaborative effort helps catch issues early and fosters a culture of continuous improvement within the team.
6. **Unit Testing via Automation:** We will implement automated unit tests to verify that each component of the system works as expected. This includes key modules like House Allocation and Maintenance Request Management. Automated testing allows us to swiftly identify any issues and ensures that updates or new features don’t disrupt existing functionalities.
7. **Safety and Risk Control:** Early in the development process, we will identify and address potential risks such as performance issues and data breaches. To safeguard personal information, we will implement robust security measures, including data encryption and access controls. This proactive approach helps ensure that our system remains secure and reliable throughout its lifecycle.
8. **Tests and User Input:** Faculty members and representatives from the university estate office will actively participate in User Acceptance Testing (UAT) to confirm that the system meets their needs and expectations. This phase will specifically focus on crucial features like seniority-based housing distribution and maintenance tracking, allowing users to provide valuable feedback that can enhance the overall functionality of the system.
9. **Notifications for Automated Maintenance:** To ensure that maintenance issues are addressed promptly, the system will include tools for managing maintenance requests and sending automated daily reminders for any outstanding tasks. This proactive approach helps keep everything running smoothly and ensures that faculty members can rely on timely responses to their maintenance needs.

**Tools**

1.  **Java:** The system will be built using Java because it supports the key principles of Object-Oriented Programming (OOP), which are essential for creating a robust and scalable housing allocation system.
2.  **IntelliJ IDEA:** As our primary integrated development environment (IDE), IntelliJ IDEA will provide us with powerful tools for coding, testing, and debugging Java applications, making the development process more efficient and streamlined.
3.  **MySQL:** We’ll use MySQL to manage and store all relevant information about housing assignments, faculty members, and maintenance requests. This relational database will help ensure that data retrieval and management are efficient and effective.
4.  **JUnit:** To ensure that each part of our system functions correctly before integration, we’ll implement JUnit for automated testing. This will help us catch any issues early on and maintain high-quality code throughout the development process
5.  **Lucidchart:** We’ll use Lucidchart to create visual diagrams that clarify our system architecture and design. This will help us map out class structures and data flows, making it easier for everyone to understand how the different components interact.
6.  **GitHub:** GitHub will serve as our version control system, allowing team members to collaborate effectively. It will help us track changes to the codebase and manage contributions, ensuring that everyone is on the same page.
7.  **LibreOffice:** We’ll use LibreOffice to prepare our project documentation, including system requirements, design specifications, and user manuals. This will help us maintain clear and organized records of our work throughout the project.
8.  **Google Drive:** Google Drive will facilitate easy file sharing and document access among team members. This platform will enhance communication and ensure that everyone has access to crucial information whenever they need it.
9.  **Daily Stand-Up Meetings:** We’ll hold daily stand-up meetings to discuss any issues, track progress, and keep the team aligned on project goals and timelines. These brief check-ins will help maintain focus and encourage collaboration.

**Methods**

### 1. Initial Development Stages

We’ll break the project into distinct stages, which will help us allocate resources efficiently and keep track of our progress in an organized manner. This structured approach will ensure that each phase is well-defined and manageable.

### 2. The Agile Methods

Our team will adopt an Agile methodology, focusing on iterative development. This means we’ll work in small, collaborative teams that adapt and evolve requirements and solutions as we go along. This flexibility will allow us to respond to changes quickly and effectively.

### 3. Cooperation Instruments

To facilitate real-time collaboration, we’ll utilize tools like Google Drive and GitHub. These platforms will ensure that every team member has access to the latest project files and updates, making it easier for us to work together seamlessly.

### 4. Frequent Evaluations of Progress

During our team meetings, we’ll regularly discuss our progress to ensure it aligns with our project goals and timeline. This ongoing evaluation will allow us to make any necessary adjustments along the way, keeping us on track.

### 5. Procedures for Documentation

We’ll maintain thorough documentation of all our decisions, designs, and developments throughout the project. This will create a clear reference for both our team and other stakeholders, ensuring everyone is informed and aligned.

**4.3 Product Acceptance Plan (PAP) :**

### 4.3.1 Objective Standards for Product Acceptance

For the product to be approved, it must meet the following standards:

* Meets Requirements: The system should fulfill all the functional and non-functional requirements outlined in the specifications document.
* No Critical Errors: There should be no major bugs or issues found during testing that could affect the system’s performance.
* Successful User Testing: User Acceptance Testing (UAT) must be completed successfully, without any significant concerns raised by the users.
* Adherence to Security Standards: The system needs to comply with security protocols to ensure the protection of user data.
* Comprehensive Documentation: Complete user guides and technical documents must be provided and thoroughly reviewed to assist users and maintainers.
* Performance Criteria: The software should meet the established performance benchmarks for response times and data retrieval rates, ensuring a smooth user experience.

**Acceptance Standards**

To gain acceptance, the system must meet the following requirements:

* **House Management**: The system should provide an accurate view of available houses and their statuses, whether they are assigned or still unallocated.
* **Faculty Registration**: Faculty members should have the ability to apply for housing and easily track the progress of their applications throughout the process.
* **Maintenance Requests**: Users must be able to submit maintenance requests and follow up on their status to ensure timely resolution of issues.
* **Notification System**: The system needs to send out timely notifications for maintenance requests and updates regarding application statuses, keeping users informed at all times.

 **Data Security**: It’s essential that user data is encrypted and stored securely to protect sensitive information.

 **User-Friendly Interface**: The system should be intuitive and easy to navigate, ensuring that all users can utilize it without any difficulty.

 **Performance and Scalability**: The system must be capable of handling multiple users at the same time without any delays or performance issues.

 **Compatibility**: The system should work seamlessly across various devices and browsers, allowing users to access it from wherever they are.

**Acceptance Testing**

1 **Functional Testing**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | |  | |  | |  | |
| **Feature** |  | | **Test Description** |  | **Expected Outcome** |  | **Status** |
| House Management |  | | Test to see if houses are displayed correctly. |  | Accurate display of houses and  their status. |  | In Progress |
| Faculty Registration |  | | Test the application process for faculty members. |  | Faculty can apply for houses and check their status. |  | Not Started |
| Maintenance Requests |  | | Test submitting and tracking maintenance requests. |  | Users can submit and track maintenance requests. |  | Not Started |
| Notification System |  | | Test the alert system for updates. |  | Users receive timely notifications about important updates. |  | Not Started |

2 **Non-Functional Testing**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Area** | | **Description** | | **Expected Outcome** | |  | **Status** |
| Security |  | Check if data is stored and sent securely. |  | Data is encrypted and access is controlled. |  | | Not Started |
|  |  |  |  |  |  | |  |
| Performance |  | Test system responsiveness under heavy use. |  | System stays responsive during busy times. |  | | Not Started |
| Compatibility |  | Test on different devices and browsers. |  | Works smoothly across various platforms. |  | | Not Started |
| Scalability |  | Ensure the system can handle more users as needed. |  | System works well without slowing down. |  | | Not Started |

**4.3.2 Product Acceptance Procedures**

**1 Testing**

* **Unit Testing**: This step involves checking each component of the system to ensure it works as intended.
* **Integration Testing**: Here, we verify that all components work together as a cohesive unit.
* **System Testing**: At this stage, we assess the entire system to confirm that it meets the specified requirements.
* **User Acceptance Testing (UAT)**: This is the final round of testing where real users interact with the system to ensure it meets their needs and expectations.

**2 Demonstrations**

* **Live Demonstrations**: We will host live demonstrations for stakeholders to showcase key features of the system, such as how to track maintenance requests and manage housing applications effectively.

**3 Evaluation**

* **Data Analysis**: We will analyze the user interaction data gathered during testing to pinpoint any issues or slowdowns that may arise.
* **Performance Analysis**: This involves assessing the system’s capacity to handle a high number of users simultaneously, ensuring it performs well under pressure.

### 4 Examination

* Code Reviews: Our team will conduct peer reviews of the code to ensure it is secure and follows best practices. This collaborative effort helps catch any potential issues before the system goes live.
* Documentation Review: We will carefully examine all project documentation, including user manuals and technical guides, to ensure they are comprehensive and easy to understand.
* Compliance Check: A thorough review will be conducted to ensure the system adheres to data security standards and legal requirements, safeguarding user information.

### 4.3.3 Acceptance Procedure

#### Preparation

* Stakeholder Consultation: We will discuss the acceptance criteria with professors, the estate office, and IT support to align expectations and clarify any concerns.
* Training Material and Documentation Readiness: We'll ensure that all training materials and necessary documentation are complete and ready for distribution to users.

#### Execution of Testing

* Functional and Non-Functional Testing: We will carry out both functional and non-functional testing to assess the system’s performance and usability.
* Findings Documentation: All results from the testing process will be documented in the specified tables for transparency and accountability.
* Issue Listing and Prioritization: We will compile a list of identified issues, prioritizing and categorizing them to facilitate efficient resolution.

**4.3.4 Resolving Conflicts**

* **Issue Monitoring and Assignment**: We'll keep a close eye on any problems that arise, assigning them to the appropriate development team members for resolution. It's crucial that we address all significant bugs before final acceptance to ensure a smooth rollout.
* **Retesting**: Once issues are fixed, we will retest those components to confirm that they are functioning properly and that no new issues have emerged.

**4.3.5 Review of Stakeholders**

* **Feature Demonstration**: During a dedicated review meeting, we will showcase the key features of the system to our stakeholders, highlighting how the system meets their needs.
* **Gathering Feedback**: We’ll actively seek input on performance and usability from the stakeholders, valuing their insights and perspectives.
* **Adjustments Based on Feedback**: We will make any necessary adjustments to the system based on the feedback received, ensuring it aligns with user expectations and requirements.

**4.3.6 Final Acceptance**

* **Stakeholder Approval**: Once testing is complete and the system meets all acceptance criteria, we will obtain final approval from stakeholders. This step is crucial for ensuring that everyone is on board and confident in the system's readiness for deployment.

### 1. Outcomes

The acceptance procedure will produce the following key items:

* Test Reports: These are detailed reports that outline the results of both functional and non-functional tests. They provide insights into how the system performed and any areas that may need improvement.
* Issue Logs: This document will track any problems encountered during testing, along with their resolutions. It ensures transparency and helps the team learn from any challenges faced during the process.
* Final Sign-Off Document: Once testing is successfully completed, this document will be signed by all stakeholders. It serves as a formal acknowledgment that the system meets the required standards and is ready for deployment

**4.3.7 Project Organization**

1 **General Overview**

This section highlights the project team and their valuable contributions to developing the House Allocation System. Each member brings unique skills and perspectives that have played a crucial role in shaping the system. Their collaborative efforts ensure that the project runs smoothly, meets objectives, and ultimately serves the needs of the university and its faculty effectively.

.

2 **Team Roles and Responsibilities**

|  |  |
| --- | --- |
| **Role** | **Responsibilities** |
| **Project Manager** | - Oversees the project, ensuring timelines and budgets are met. - Coordinates communication among team members and stakeholders. - Monitors progress and adjusts plans as needed. |
| **Developers** | - Responsible for coding and implementing system features according to specifications. - Collaborate with other developers to integrate modules effectively. - Conduct unit tests to ensure code quality. |
| **Quality Assurance** | - Conducts comprehensive testing to ensure the system meets quality standards. - Identifies bugs and issues, ensuring they are resolved before the system is delivered. - Performs regression testing after fixes to confirm functionality. |
| **Stakeholders** | - Provide input and feedback throughout the project lifecycle. - Help define requirements based on university needs. - Review progress and outcomes, ensuring the system meets expectations. |

**4.3.8 Comprehensive Role Explanations**

**Project Supervisor**  
The project manager plays a vital role in overseeing the entire process from start to finish. Their responsibilities include:

* **Project Planning:** Setting the budget and timeline for the project to ensure it stays on track.
* **Team Coordination:** Facilitating effective communication and collaboration among team members to promote a cohesive working environment.
* **Progress Monitoring:** Keeping a close eye on the project’s progress and making necessary adjustments to ensure objectives are met.

**Developers**  
The developers are the backbone of the project, responsible for:

* **Coding:** Writing the code that forms the foundation of the House Allocation System.
* **Module Testing:** Conducting thorough tests on each component to ensure they function as intended before integrating them into the larger system.
* **Collaboration:** Working closely with fellow developers to ensure that every piece of the system fits together seamlessly, contributing to a cohesive and functional end product.

.

**4.3.9 Assurance of Quality**

The Quality Assurance team plays a vital role in ensuring that the House Allocation System meets high standards before it goes live. Their responsibilities include:

* **System Testing:** Conducting thorough tests to identify any defects or issues within the system, ensuring everything works smoothly.
* **Quality Checks:** Verifying that the system meets all predetermined quality requirements before it is deployed, ensuring it’s reliable and effective.
* **Final Reviews:** Performing one last round of testing to confirm that the system is ready for delivery to stakeholders, making sure it’s polished and meets expectations.

**Participants**

The stakeholders are essential to the project’s success by:

* **Providing Requirements:** Helping to define the system’s goals based on their needs and experiences, ensuring that it serves its intended purpose.
* **Offering Feedback:** Giving regular input throughout the development process to ensure the system aligns with academic objectives and meets user expectations.
* **Final Approval:** Reviewing the completed system to ensure it is ready for deployment and granting the necessary approvals for launch.

### External Interfaces

### 4.3.10 External Interface Overview

This section outlines the connections between the University House Allocation System and other key organizations that play a crucial role in its operation. By fostering clear communication and establishing well-defined roles among the various entities involved, we aim to ensure the successful deployment and smooth functioning of the system. This collaborative approach not only enhances efficiency but also ensures that everyone is on the same page, contributing to a seamless user experience for faculty and staff alike

#### External Entities and Their Roles

|  |  |
| --- | --- |
| Entity | Role and Responsibilities |
|  |  |
| University Estate Office | - Information Provider: Supplies data on available houses, including details like size, location, and current occupancy status (allocated/unallocated). - Policy Maker: Establishes allocation criteria based on faculty seniority and ensures that these policies are adhered to throughout the allocation process. - Application Processing: Reviews applications from faculty members and makes decisions on house allocations based on established guidelines. |
| Maintenance Department | - Request Management: Receives and processes maintenance requests submitted by faculty through the House Allocation System. Ensures timely responses and resolutions. - Status Tracking: Updates the system with the status of maintenance requests (e.g., pending, in progress, completed) to keep faculty informed and improve transparency in maintenance operations. |
|  |  |
| IT Support Services | - Technical Support: Provides ongoing technical assistance to ensure that the House Allocation System operates smoothly and securely throughout its lifecycle. - System Maintenance: Monitors system performance and addresses technical issues promptly to prevent downtime and ensure user satisfaction. - Data Security: Implements security measures to protect sensitive data, ensuring compliance with data protection regulations. |

#### 

#### Role Descriptions

##### **University Estate Office**

The University Estate Office is a key player in the allocation process, responsible for:

 **Information Provider:** This office will supply detailed information about the available houses, including their locations, the number of rooms, amenities, and current occupancy status. By integrating this information directly into the system, faculty will be empowered to make well-informed decisions when applying for residences.

 **Policy Maker:** This group will be responsible for establishing the guidelines that govern house distribution, ensuring fairness and transparency in the process. The policies will prioritize faculty seniority, giving preference to higher-ranking faculty members when it comes to housing allocations.

 **Application Processing:** After faculty members submit their applications through the system, the estate office will review them, allocate houses based on the established criteria, and promptly notify faculty members about the outcomes of their applications.

##### **Maintenance Department**

The Maintenance Department ensures that the houses are well-maintained and responsive to issues reported by faculty:

 **Request Management:** Faculty members can easily submit maintenance requests through the system. The maintenance department will oversee these requests, ensuring they are allocated the necessary resources and prioritized based on urgency.

 **Status Tracking:** The maintenance division will regularly update the system with the current status of maintenance requests. This will allow faculty members to track their requests in real time, improving communication and enhancing their overall satisfaction with the maintenance services provided.

.

4.3.11 IT Support Services

The IT Support Services team is vital for maintaining the technical integrity of the House Allocation System:

* Technical Support: They will provide ongoing assistance to users facing any technical issues while using the system. This support is crucial not just during the initial rollout but also for any subsequent updates.
* System Maintenance: The IT team will regularly monitor the system's performance to ensure everything runs smoothly. This includes addressing bugs, implementing software updates, and performing server maintenance.
* Data Security: They will implement and continuously update robust security measures to protect sensitive user information, including faculty details and maintenance records, ensuring that all data remains secure.

**4.3.12 Internal Interfaces**

**1 Internal Interface Overview**

This section focuses on the internal structure of the project team, highlighting how various roles communicate and collaborate in developing the University House Allocation System.

**2 Collaboration Among Team Members**

**Function: Monitoring Progress & Project Alignment**

* **Regular Meetings:**
  + Weekly stand-up meetings will be held to discuss the status of each system module, any challenges encountered, and the next steps forward.
  + These meetings will involve all team members—developers, QA staff, and project managers—to ensure everyone is aligned on deliverables and timelines.
* **Communication Tools:**
  + Team members will utilize real-time communication platforms like Slack or Microsoft Teams to address issues as they arise, promoting quick resolutions and fostering collaboration.

* **Shared Calendar:**
  + A shared calendar will keep everyone informed about upcoming meetings, deadlines, and significant milestones, ensuring that all team members are on the same page.

**3 Task Management: Assigning Tasks and Tracking Progress**

* **Task Distribution:**
  + Responsibilities will be assigned to developers and QA testers based on their expertise and the project timeline, ensuring that everyone is focused on tasks that match their strengths.
  + To maintain accountability, project management software (like Jira or Trello) will be used to track every task.
* **Development Reports:**
  + Team members will regularly provide updates on their progress through project management tools and during meetings, helping to identify any obstacles early on.
  + Frequent check-ins will ensure that the project is progressing smoothly and according to schedule, keeping everyone aligned and informed.

### 4 Assisting Mechanisms

Support functions and quality control play a crucial role in the project’s success.

* Configuration Management:
  + Collaborating with the configuration management team will ensure that the codebase is properly version controlled. This makes it easy to track changes and roll back to previous versions if necessary.
  + Every change made will be documented to maintain an accurate record of the project's development, promoting transparency and accountability.
* Quality Control (QC):
  + The QA team will work closely with developers to establish testing schedules and standards, ensuring that each component is thoroughly tested before integration.
  + They will also create comprehensive test cases based on the system specifications to guarantee extensive coverage during testing phases. This collaboration will help identify and address potential issues early, leading to a more robust final product.

### 5 Knowledge Management & Communication Documentation

Role of Documentation:

* A well-structured documentation process will be implemented to ensure that all decisions, designs, and development progress are thoroughly recorded. This will create a clear and accessible history of the project.
* Each module of the system will come with detailed documentation that outlines its functionalities, organizational structure, and interdependencies, making it easier for team members to understand how everything fits together.

Documentation Types:

* Technical Documentation: This will include essential materials such as architecture diagrams, code structures, and database schemas. These documents will help the team grasp the system’s functionality and the relationships between different components.
* User Manuals: Comprehensive user manuals will be created to guide faculty members through the system. These manuals will cover important features, such as how to apply for housing and submit maintenance requests, ensuring that users can navigate the system with confidence and ease.

### 6 The Role of the Review Process: Feedback and Continuous Improvement

Code Reviews:

* Regular code reviews will be an integral part of our development process. These reviews will ensure that best practices are followed and help identify any potential security vulnerabilities in the code.
* Peer reviews will foster a culture of continuous learning and collaboration, encouraging team members to share knowledge and insights, which will ultimately strengthen the entire team.

Feedback Loop:

* After each testing phase, we will gather feedback from team members regarding the processes used and any challenges they encountered. This feedback is invaluable for understanding what worked well and what needs improvement.
* We will carefully analyze this feedback and use it to enhance the project in future iterations. By documenting the lessons learned, we can ensure that the entire team grows and improves with each phase of development

**4.3.13Authorities and Responsibilities**

### Summary of Authority and Obligations

This section outlines the responsibilities and roles of each team member involved in the University House Allocation System project, ensuring accountability and clarity in key activities.

### 1 Tasks for Development

2 Function: Delivery and Ownership of Modules

* Assignment of Modules: Each developer will be assigned specific system modules, such as faculty registration, house management, and maintenance tracking. This targeted approach ensures that responsibilities are clear and manageable.
* For instance, a developer with expertise in backend technologies might take charge of the House Management Module, while another developer with experience in notifications could handle the Notification System. This division of labor allows team members to focus on their strengths, fostering efficiency and high-quality outcomes

 **Clearly Defined Deliverables:**  
Each assigned module will come with specific deliverables to ensure clarity and focus. For example:

* **House Management Module:** This module will include features that display available houses, their allocation and unallocation statuses, as well as relevant policies to guide users.
* **Maintenance Tracking Module:** This module will allow users to submit maintenance requests, receive updates on their progress, and facilitate communication with the maintenance department.

 **Due Dates:**  
To keep developers on track, regular progress checks will be implemented. These check-ins will help ensure that milestones are met and that any potential roadblocks are identified early, allowing for timely adjustments to the project timeline

**3 Testing Activities**

**Function: Validation and Quality Assurance**

* **Testing Accountability:**  
  The Quality Assurance (QA) Team will take charge of thoroughly testing each module to ensure it meets both the functional and non-functional requirements outlined in the project documentation. Each QA tester will focus on specific modules, such as the Faculty Registration Process and Maintenance Request Handling, to ensure detailed attention to quality.
* **Phases of Testing:**  
  The QA team will carry out several key testing phases, including:
  + **Unit Testing:** This involves examining individual components to confirm they function correctly on their own.
  + **Integration Testing:** This phase checks that various modules work together seamlessly, ensuring that the system operates as a cohesive whole.
  + **User Acceptance Testing (UAT):** In this final phase, feedback will be collected from actual users—such as faculty members and estate office staff—to verify the system’s usability and functionality. This user input is crucial for ensuring that the system meets their needs and expectations

**4 Quality Standards**

* **Quality Standards:**  
  Throughout all testing phases, the team will adhere to established quality standards. This includes performance benchmarks to ensure the system operates efficiently and security protocols to protect sensitive user data. By following these standards, we aim to deliver a reliable and secure House Allocation System.

**5 Making Decisions**

**Function: Project Management and Oversight**

* **The Project Manager's Authority:**  
  The Project Manager will hold the final authority on all significant decisions regarding the project's scope, resource allocation, and any necessary adjustments to the timeline. This responsibility ensures that any obstacles are addressed swiftly, keeping the project on track and aligned with its objectives.
* **The Decision-Making Process:**  
  To make informed decisions, the Project Manager will hold regular strategy meetings with team leads (developers and QA). These discussions will allow the team to provide valuable insights into the project’s status and any challenges they may be facing. To promote transparency, all decisions made during these meetings will be documented and shared with the entire team, ensuring everyone is informed and aligned.

**6 Escalation Procedures**

* **Escalation Procedures:**  
  In situations where the team encounters serious issues that they are unable to resolve, these problems will be escalated to the Project Manager for guidance and solutions. This process ensures that any bottlenecks are addressed promptly, preventing them from negatively impacting the overall project timeline.

**7 Expressing Accountabilities**

**Function: Guaranteeing Clarity and Responsibility**

* **Matrix of Responsibility:**  
  To clarify who is responsible for each task and deliverable, we will create a Responsibility Assignment Matrix (RAM). This matrix will be distributed to all team members, ensuring everyone understands their specific roles and responsibilities within the project.
* **Frequent Updates:**  
  To maintain accountability and alignment throughout the development process, the team will receive regular updates regarding any changes to the project's scope or responsibilities. These updates will help keep everyone on the same page and ensure that the team is working cohesively toward the project goals

Documentation of Updates:  
We will ensure that all updates are thoroughly documented and kept up to date. This will serve as a reliable reference for the team and make it easier to manage any changes in roles or responsibilities as the project progresses. Having clear documentation helps everyone stay informed and adapt smoothly to any adjustments needed along the way.

**4 .Risk Management Plan :**

**Team:**

This project is being managed by three students, using Java and Object-Oriented Programming (OOP) principles.

**1. Purpose**

This risk management plan's objective is to recognize and address any possible obstacles that might come up while the university's housing allocation system is being developed and run. This entails managing maintenance requests and supervising faculty housing assignments. We want to make sure that the system functions properly, avoiding interruptions and guaranteeing a positive experience for both developers and users, thus we try to address risks as early as possible. Our goal is to keep things as simple and problem-free as possible while developing and utilizing the system.

**2. Risk Identification**

**2.1. Potential Risks to the System**

* **Delays in House Allocation:** There could be delays in informing faculty members about house availability, slowing down the allocation process and creating frustration.
* **Sorting Errors:** Misclassification of faculty applications based on seniority or grade data could lead to inaccurate house allocations, affecting fairness and transparency.
* **Communication Breakdowns:** If the system provides inaccurate or delayed updates about house availability, it could lead to confusion and misunderstandings among faculty members.

**2.2. Potential Risks Related to Maintenance**

* **Unresolved Maintenance Requests:** Slow responses to maintenance requests could lead to frustration among faculty or even property damage if issues aren’t fixed in time.
* **Poor Request Tracking:** If the system doesn't effectively track maintenance requests, it could result in unnecessary delays in resolving problems, creating inefficiencies.
* **Resource Shortages:** A lack of maintenance personnel may lead to extended wait times, making it harder to address faculty needs quickly and efficiently

**2.3. Risks in Data Management**

* **Data Loss:** Without proper backups, essential data like faculty applications and house allocation statuses could be permanently lost, causing setbacks and confusion in the allocation process.
* **Data Breach:** Unauthorized access to sensitive faculty information could lead to privacy violations, putting the university’s data security and reputation at risk.

**2.4. Risks in Java Development and Object-Oriented Programming (OOP)**

* **Poor Class Design:** If Java classes are not designed thoughtfully, the system could become overly complex and difficult to maintain, leading to frequent issues or updates.
* **Misuse of OOP Principles:** Improper implementation of object-oriented principles like inheritance or encapsulation can make the system harder to scale, extend, or maintain over time.
* **Debugging and Error Handling Issues:** If Java code errors are not effectively handled, the system could crash or produce incorrect outputs, disrupting key functions like house allocation or maintenance tracking.

**2.5. Risks Associated with Human Resources**

* **Team Collaboration Issues:** If communication breaks down within the team, it could lead to misunderstandings, development delays, or inconsistent code that doesn't align with the project goals.
* **Lack of Java/OOP Experience:** Limited experience with Java or object-oriented programming could result in inefficient code, slowing down development and causing potential rework as team members learn on the job.
* **Team Member Unavailability:** If key team members are unavailable due to personal commitments or other responsibilities, it could slow down progress and strain the remaining team to meet deadlines.

**3. Risk Analysis**

**3.1. Uncertainties**

* **System Complexity:** Will the system effectively manage all the moving parts, such as tracking unresolved maintenance requests and allocating houses based on seniority without errors?
* **Team Capability:** Can the team successfully build a scalable and maintainable system using Java and object-oriented programming (OOP) principles, given the challenges?

**3.2. Risk Factors**

* **System Delays:** If the system is slow to communicate updates or faculty members are late in confirming their interest in available houses, it could cause significant delays in the allocation process.
* **Maintenance Errors:** Mishandling or misprioritizing maintenance requests could lead to prolonged wait times and frustration for faculty members, potentially worsening property conditions.

**3.3. Current Controls**

* **Automating House Allocation:** The goal is to replace the current manual process of sorting faculty applications with an automated system that speeds up allocation and ensures accuracy.
* **Streamlining Maintenance Requests:** Java will be used to automate and improve the current, slow maintenance request system, ensuring quicker processing and tracking of requests to avoid bottlenecks.

**4. Risk Control**

#### 4.1. System Control

* Automated Sorting and Notifications: To ensure fairness and efficiency in house allocation, the system will use Java to automatically prioritize senior faculty members and send timely notifications about house availability. This automation helps streamline the process and reduces the risk of human error or delays.
* Faculty Confirmation: To avoid unnecessary hold-ups, a confirmation step will be added where faculty members must express their interest in the offered house. This helps ensure that the allocation process continues smoothly, freeing up homes that are not in demand.

#### 4.2. Maintenance Request Control

* Categorizing Requests: Maintenance requests will be sorted into "routine" and "urgent" categories to ensure that critical issues are prioritized and addressed quickly. This prioritization helps prevent small issues from turning into bigger problems while ensuring that more urgent needs, like plumbing or electrical failures, are handled immediately.
* Automated Daily Reminders: To keep things on track, the system will automatically send daily reminders to the maintenance team about unresolved issues. This feature will ensure that no request is forgotten, helping to keep faculty living spaces in good condition.

#### 4.3. Data Management Control

* Data Backup: To protect the system from data loss, automated daily backups of all housing and maintenance data will be carried out. This backup ensures that important information is always secure, and in case of any errors, the system can be quickly restored without losing vital records.
* Access Control: To safeguard sensitive information, such as faculty details and house status, secure access controls will be implemented. Only authorized individuals will be able to view or modify this data, preventing unauthorized access and ensuring the privacy of personal and institutional information.

### 4.4. OOP Control and Java

* Proper Class Design: In developing the system, we’ll make sure Java classes are thoughtfully crafted, with each class having clear responsibilities. Key object-oriented principles like inheritance (for sharing common behaviors), encapsulation (to protect data), and polymorphism (for flexibility in handling different types of houses) will be applied carefully to keep the system efficient and easy to maintain.
* Error Handling: To keep the system running smoothly, we will implement robust error-handling methods. This ensures that even when things go wrong, the system can recover without crashing or producing inaccurate results, giving users a reliable experience.
* Consistent Testing: Regular testing will be performed to confirm that the system not only works well but also adheres to Object-Oriented Programming (OOP) standards. Testing will help us ensure the system is running efficiently, and any issues are resolved early.

### 4.5. Management of Human Resources

* Team Communication: We’ll set up regular team meetings to ensure that everyone stays on the same page, discuss ongoing progress, and resolve any challenges. Clear communication will help make sure tasks are well understood and carried out efficiently.
* Java/OOP Training: If needed, time will be allocated for team members to sharpen their Java skills and get more comfortable with OOP concepts. This will ensure everyone is equipped to handle the development tasks effectively.
* Task Distribution: Tasks will be assigned in a balanced way to make sure everyone is contributing equally. In case a team member is unavailable due to personal or professional reasons, backup plans will be in place to keep the project on track without disruptions.

**5. Risk Monitoring**

**5.1. Ongoing Monitoring**

* **System Performance Checks:** Regularly monitor how well the automated house allocation and maintenance tracking systems are functioning. This will help ensure that the system continues to run smoothly and that it's assigning houses based on seniority and handling maintenance requests as expected.
* **Response Time Reviews:** Keep an eye on how quickly houses are being allocated and how fast maintenance issues are being addressed. This will help us catch any delays early and make
* adjustments to improve system efficiency.

**5.2. Testing and Validation**

* **Frequent Testing of Java Classes:** We’ll consistently test key Java classes, especially those responsible for sorting house applications, confirming faculty interest, and sending out maintenance alerts. These tests will ensure that everything is running smoothly and that the system continues to meet the standards of Object-Oriented Programming (OOP).

1. **PROJECT PLANNING:**

**Gantt chart:**



