Suite Dreams

Version 1.1

11/30/24

Kyren Stephenson, Jonathan Fasano, Alexes Pagan, John Doyle, Arman Hemmati Alamdari, Parsa Jafaripour

# 

# Contents

## 1. Introduction

1.1 Project Purpose [I:PP]..................................................................... 4

1.2 Project Scope [I:PS]....................................................................... 4

1.3 Product Overview [I:PO]................................................................. 4

1.4 Intended Audience [I:IA]................................................................. 5

1.5 UI Sketch [I:US].............................................................................. 6

## 2. Stakeholders

2.1 Students [ST:STU]........................................................................... 7

2.2 Administrators [ST:AD].................................................................. 7

2.2.1 Resident Life [ST:RL]........................................................ 7

2.2.2 Student Access Services [ST:SAS]................................... 7

2.2.3 Campus Public Safety [ST:CPS]........................................ 8

2.3 Developers [ST:D]........................................................................... 8

## 3. System Features

### 3.1 Functional Requirements [SF:FR]................................................... 9

3.1.1 Registration and Login [SF:RL]......................................... 9

3.1.2 Dorm Selection [SF:DS]..................................................... 9

3.1.3 Roommate Matching [SF:RM]........................................... 9

3.1.4 Cancel Room Request [SF:CRR]....................................... 10

3.1.5 Submit Application [SF:SA].............................................. 10

3.1.6 Room Status [SA:RS]........................................................ 10

3.1.7 Accessibility Features [SF:AF].......................................... 10

### 3.2 Non-functional Requirements

3.2.1 Performance [NFR:P]......................................................... 11

3.2.1.1 Login and Registration [P:LR].............................. 11

3.2.1.2 Dorm and Roommate Selection [P:DRS].............. 11

3.2.1.3 Database Indexing [P:DI]...................................... 11

3.2.2 Languages [NFR:L]............................................................ 11

3.2.2.1 Backend [L:B]....................................................... 12

3.2.2.2 Frontend [L:F]....................................................... 12

3.2.2.3 Frontend Enhancements........................................ 12

3.2.3 Data Management [NFR:DM]............................................ 12

3.2.4 Matching Algorithm Performance [NFR:MA]................... 13

3.2.5 Reliability [NFR:R]............................................................ 14

3.2.5.1 Error Handling [R:EH].......................................... 14

3.2.5.2 User Activity Logging [R:UAL]........................... 15

3.2.5.3 Data Backup and Recovery [R:DBR]................... 15

3.2.5.4 Testing [R:T]......................................................... 15

3.2.6 Usability [NFR:U].............................................................. 15

## 

## 1. Introduction

1.1 Project Purpose [I:PP]

The dorm selection process is infamous for its tedious, frustrating, and inconvenient process. Many students across the country live on-campus at their universities, and they engage in the difficult process of dorm and roommate selection. Suite Dreams is a program developed to make that process not only easier but enjoyable. This application will make the dorm selection process transparent, letting the user see the dorm on the application prior to selecting it and algorithmically selecting compatible roommates.

1.2 Project Scope [I:PS]

The project is a web-based application that allows students to login using their university accounts. Students will view available dorm options and submit their ranked room preferences. The project also includes features for both individual and group dorming. Students have the ability to complete a roommate questionnaire pairing them with a compatible roommate, and allows users to waitlist or cancel their room selections. Students will be able to see their room assignment if they have been matched or none if they are in the waitlist waiting to be matched. An algorithm is used to properly allocate students together in dorms. The system will incorporate viewing dorm photos along with allowing students to select accessibility accommodations.

1.3 Product Overview [I:PO]

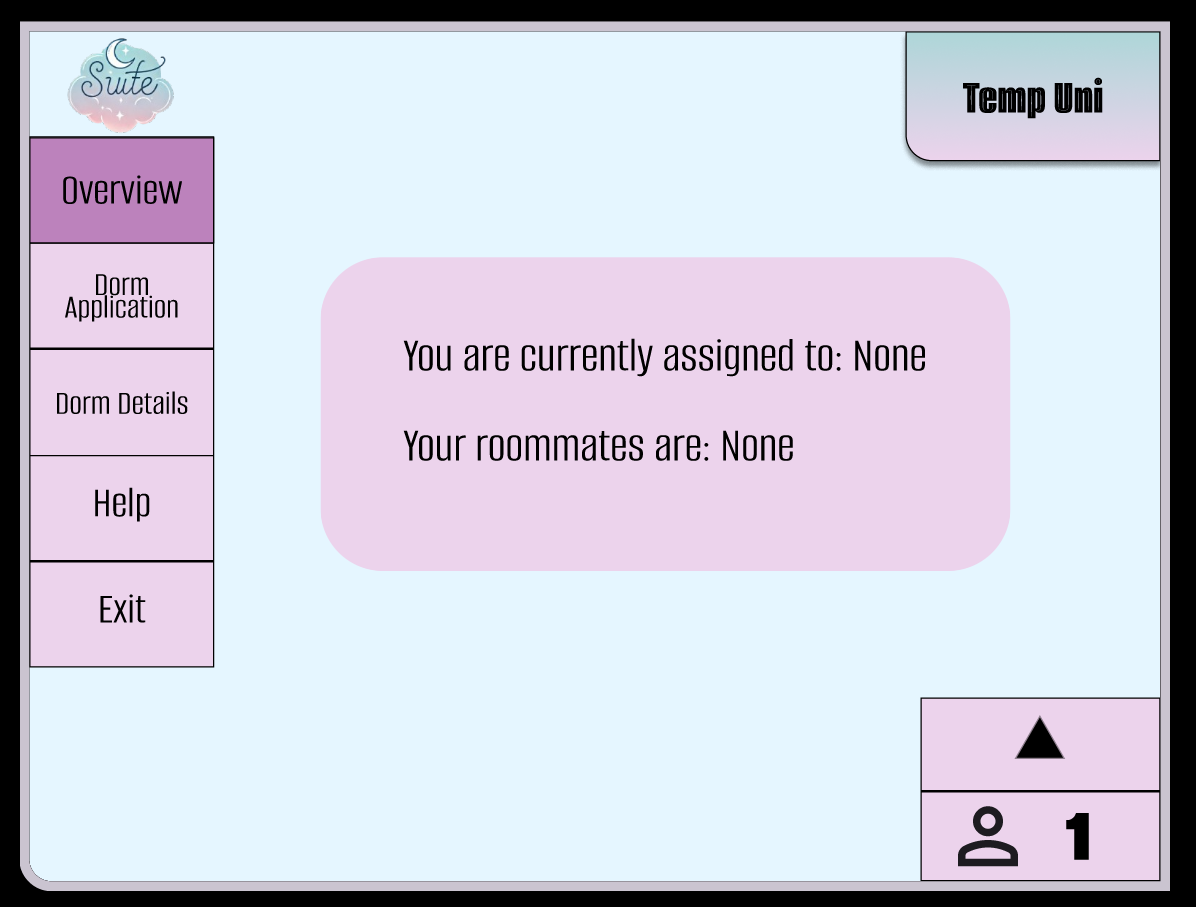
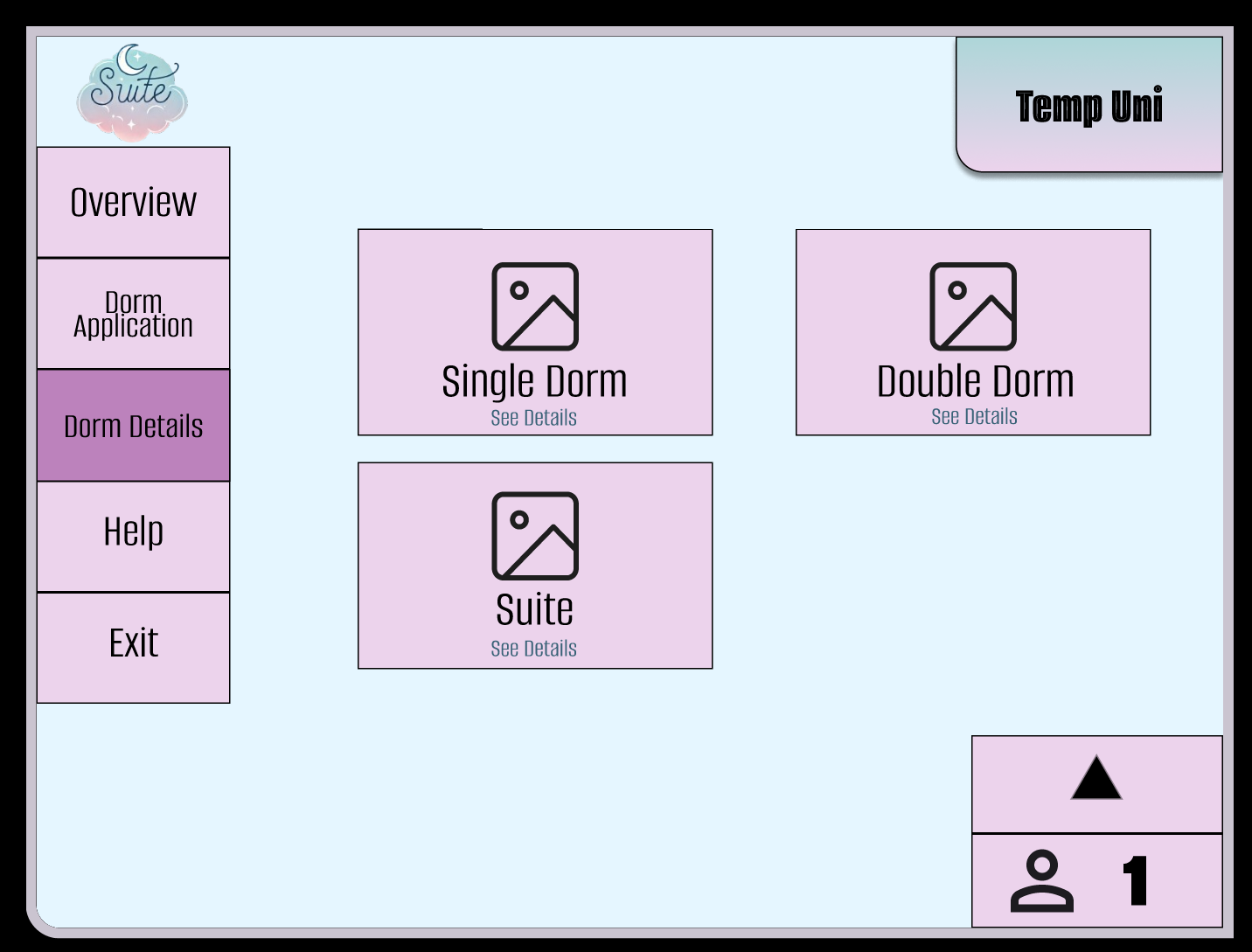
* User Account Login [PO:UAL]: Students log in using their university credentials to access the dorm selection system.
* Dorm Availability Display [PO:DAD]: Students can browse and view a list of all dorms and specific details about the dorm including photos, capacity and room types. Until a university supplies images, placeholder images are used.
* Ranked Room Preferences [PO:RRP]: Students can submit a ranked list of room preferences based on availability.
* Roommate Pairing Questionnaire [PO:RPQ]: A questionnaire that helps pair compatible roommates based on preferences and lifestyle habits.
* Group and Single Applications [PO:GSA]: Students can apply as individuals or form groups by entering other student IDs.
* Cancel Applications [PO:CA]: Students can cancel their dorm applications at any point via contacting the university, in which an Administrator will remove them, and then free up that space.
* Waitlist Management [PO:WM]: Option for students to be added to a waitlist if their preferred dorm is unavailable and all are full.
* Accessibility Accommodations [PO:AA]: Students can select specific accessibility needs for dorm placement, including wheelchair accessibility and service animals.
* Algorithmic Based Matching [PO:ABM]: The system uses an algorithm to allocate rooms based on preferences and availability, along with matching compatible roommates.
* Fast Room Assignment Option [PO:FRAO]: A "Skip" button allows students to bypass ranked preferences and get a quicker dorm assignment.

1.4 Intended Audience [I:IA]

University Students [IA:US] The main users of the application. They browse dorm options, submit their preferences, and complete roommate questionnaires prior to being assigned housing.

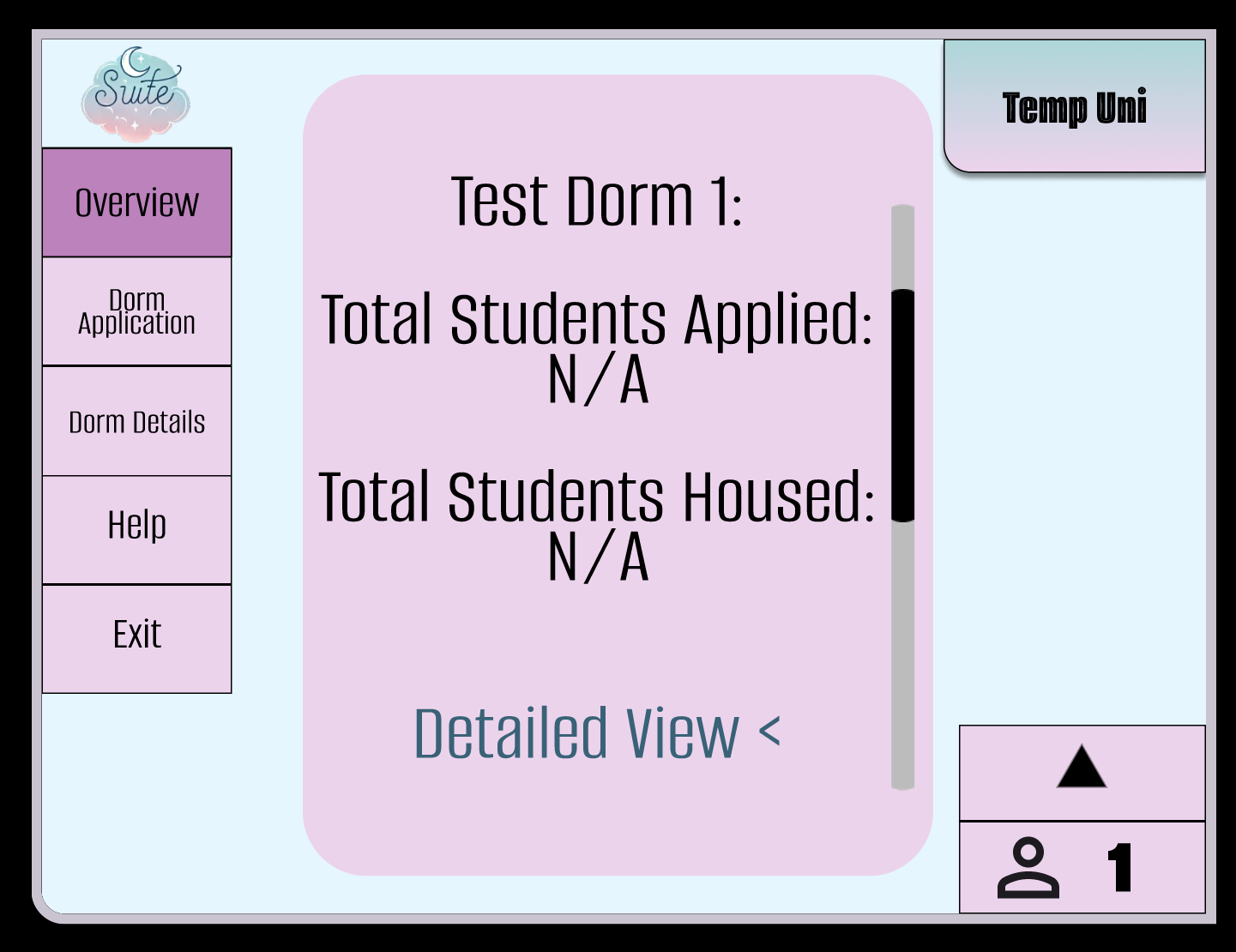
Housing Administration and University Officials [IA:HAUO] Administrators manage dorm availability, review applications and handle accessibility accommodations when necessary. Officials create policies and regulations that the application aligns with.

1.5 UI Sketch [I:US]

Student User Interface: 

## 

Administrative User Interface:



## 2. Stakeholders [ST]

2.1 Students [ST:STU]

The students are our biggest stakeholders because they are directly interacting with all aspects of our software. They have the opportunity to view different photos of the rooms which gives them a better understanding of the size and layout of the space. They have the ability to submit a roommate-questionnaire that will aid in finding a suitable roommate/suitemate. They submit their final room selections through our application, and it is our goal to make this process as smooth as possible.

2.2 Administrators [ST:AD]

Administrators are anyone at the university who uses our software to assist with student record keeping. Our application places students into rooms based on their preferences and needs, while allowing administrators to add students to a room or remove students from a room. This assists campus administrators by providing a reliable way to locate residing students.

2.2.1 Residential Life [AD:RL]

Residential Life (Reslife) needs to know where students are on campus because they are the middle-ground between the students and other campus activities. They also share the room information with the resident assistants (RAs) in the building to help create a welcoming environment.

2.2.2 Student Access Services [AD:SAS]

Our software aids Student Access Services (SAS) in guaranteeing that any student requiring accommodations is most likely to receive it. Our program ensures that a wheelchair user has a higher chance of being placed into a single-room on the first floor of a building when possible. Similarly, students with a service animal are more likely to be placed in single-rooms or with an accommodating roommate.

2.2.3 Campus Public Safety [AD:CPS]

Campus Public Safety can view student residential information at any time. This provides extra security for the residents within the dorm building. Public Safety also uses this software to locate a student’s residence in the event of a medical emergency, to check-in, as well as proceed with disciplinary action.

2.3 Developers [ST:DE]

Developers are responsible for maintaining and enhancing the application based on the evolving needs of the university and its stakeholders. They ensure the system remains user-friendly, reliable, and secure while continuously improving its performance and scalability.

Developers work closely with administrators to integrate new features or modify existing ones, ensuring that student data, dormitory information, and roommate matching functions operate efficiently. The focus is on creating a smooth and intuitive experience for both students and administrative users, ensuring the platform meets the functional and non-functional requirements defined by the university.

## 3. System Features

3.1 Functional Requirements [SF:FR]

3.1.1 Registration and Login [FR:RL]

Students log in to the dorm application using their school account that is initialized once the student clicks on the link for the application. With their student account, the application accesses the student’s data such as their class standing. Admins can log in to the application, but unlike the student, they have access to more management based features. Admins have the ability to change a student’s room in the event the student wants to move.

3.1.2 Dorm Selection [FR:DS]

Selected students have early access in choosing their dorms. The application gives the first pick to those who have accessibility accommodations such as people with wheelchairs and any other mobility accommodations. The next round of priority goes to seniors and goes down one class standing in increments. If a freshman logs in to the application prematurely, the form notifies them that they cannot register for a room yet.

3.1.3 Roommate Matching [FR:RM]

To reduce the number of room changes during the school year, our program includes a roommate questionnaire. Admins can view the information the students input. The general format of the questionnaire starts with questions about preferences, then asks about room choices, and then asks if you would like to room with anyone specific before submitting the questionnaire. Some sample questions are:

* “Do you have any allergies?”
* “Are you an extrovert or an introvert?”
* “Do you like to study in the morning, afternoon, or night?”
* “Are you a clean or messy person?”

The roommate questionnaire groups students into one dorm that share similar answers. This promotes good beginnings between roommates.

3.1.4 Cancel Room Request [FR:CRR]

The ‘cancel room request’ selection is bolded. This makes the student aware of the cancellation form. Should this button be pressed, there is another button that confirms their cancellation. This request goes to an admin and they can cancel the student’s room assignment.

3.1.5 Submit Application [FR:SA]

When the student submits the dorming application, they have the opportunity to review their inputs. After verifying the details, they check a box confirming that their information is accurate before their application is forwarded into the assignment process.

3.1.6 Room Status [FR:RS]

Admins have access to the status of a room’s availability. Students can see when a dormitory building is completely full and not accepting new applicants as each dorm displays its capacity. A sample display for a dorm would be a capacity of 45/75, where 75 is how many students the dorm can hold and 45 is how many spaces are left. Students are also able to see an overview of their dorm assignment.

3.1.7 Accessibility Features [FR:AF]

A readable and clear font will be used. Under the ‘help’ tab, it is explained how each button works and ensures that the user understands the program.

3.2 Non-functional Requirements

3.2.1 Performance [NFR:P]

Performance is a critical aspect of the “Suite Dreams” application, especially during peak usage times when student demand surges. A responsive and efficient application enhances user satisfaction and ensures that all functionalities are accessible without delays.

3.2.1.1 Login and Registration [P:LR]

The system processes login and registration requests efficiently. This is achieved by optimizing database queries and using caching strategies.

3.2.1.2 Dorm and Roommate Selection [P:DRS]

The selection of available dorms is provided by the university of which the user is able to see prior to filling out the application. The application returns results promptly after a user submits their preferences. Implementing asynchronous processing for complex queries will allow the system to handle multiple requests simultaneously. Each dorm's capacity is recorded and displayed, and once capacity is met, the user can not be added to that dorm. If all dorms are full, students will no longer be able to be added to a dormitory.

3.2.1.3 Database Indexing [P:DI]

Use database indexes strategically on frequently queried columns to speed up data retrieval. This is crucial for improving the performance of search queries, especially in large datasets.

3.2.2 Languages [NFR:L]

The “Suite Dreams” application utilizes a combination of programming languages and frameworks tailored for web development, ensuring a robust and maintainable codebase.

3.2.2.1 Backend [L:B]

Python: The primary language for backend development, chosen for its simplicity and extensive ecosystem. Python’s readability promotes collaboration and maintenance. Python based frameworks are used to accelerate the development process and ensure security.

3.2.2.2 Frontend [L:F]

A JavaScript library for building user interfaces is used, chosen for its component-based architecture. This allows for the creation of reusable UI components, improving maintainability. Specifically uses Bootstrap, HTML, CSS and JavaScript for an intuitive and clean User Interface. Bootstrap gives the UI its core design and sets a solid foundation for the interface, while CSS and JavaScript ensure interactivity.

3.2.2.3 Frontend Enhancements [L:FE]

* State Management: A state management library is used to manage complex application states, improving predictability and maintainability. This ensures that the application remains responsive and consistent, even as it scales.
* Component Libraries: component libraries are used to accelerate UI development and ensure a consistent look and feel. These libraries provide pre-built, customizable components that are easily integrated into the application.

3.2.3 Data Management [NFR:DM]

The data management system in Suite Dreams efficiently handles all critical student information, including account details, dorm preferences, and responses to the roommate questionnaire. The system ensures scalable and secure storage, with the capability to support a growing user base.

Student information, such as ID, name, class standing, and dorm preferences, is securely stored within the system. Group applications are also managed, enabling students to link their applications with others for shared accommodations. Dormitory information, including layouts, room types, and availability, is dynamically updated in real-time based on student submissions, cancellations, and overall dorm capacity.

Questionnaire responses are collected and stored securely, providing the necessary data for the matching algorithm to generate optimal roommate pairings. These responses are processed and organized to enable efficient data retrieval and analysis during the roommate matching process. The system prioritizes data privacy and security, with safeguards in place to protect sensitive information. Access is restricted to authorized users only, ensuring that students and administrators can view or modify data as required. To maintain data consistency, the system is designed to ensure reliable execution of all operations. Regular backups and contingency plans are in place to prevent data loss and ensure that information is consistently available when needed.

3.2.4 Matching Algorithm Performance [NFR:MA]

The roommate matching algorithm is a core feature of Suite Dream’s system. The model utilizes student questionnaire responses to predict roommate compatibility, ensuring that students are paired with others who share similar living preferences, such as sleep schedules, cleanliness, and social habits. The matching process works by analyzing questionnaire data to identify patterns and similarities among students, each having their own compatibility scores. By comparing these compatibility scores, the system efficiently determines which pairings are most likely to result in positive roommate experiences.

To ensure optimal performance, the system processes the data in real-time, providing quick and reliable matches. Additionally, the algorithm's performance is regularly evaluated using key metrics to ensure it continues to provide high-quality matches. The system adapts to the changing needs and preferences of the student population, making it a dynamic and integral part of the Suite Dreams solution.

3.2.5 Reliability [NFR:R]

Reliability is a cornerstone of the Suite Dreams application, encompassing system availability, data integrity, and consistent performance. A dependable system is crucial for maintaining user trust and ensuring the smooth operation of critical housing allocation processes. The application is designed to withstand high user loads, particularly during peak periods such as housing selection deadlines, while maintaining responsiveness and accuracy.

To achieve this level of reliability, the Suite Dreams application implements a multi-faceted approach. Error handling mechanisms are put in place to gracefully manage unexpected situations, providing users with clear, informative messages without exposing sensitive system information. Comprehensive logging systems track both errors and user activities, enabling quick issue resolution and providing insights for continuous improvement.

3.2.5.1 Error Handling [R:EH]

Graceful Degradation: The application is designed to handle errors gracefully, providing users with informative messages without exposing sensitive information. For instance, if a database error occurs, the application displays a user-friendly message indicating that the request could not be processed and suggests trying again later.

Form Validation: both client-side and server-side validation is implemented to catch errors before data submission.

3.2.5.2 User Activity Logging [R:UAL]

User actions (e.g., logins, dorm selections) are logged to monitor application usage patterns.

3.2.5.3 Data Backup and Recovery [R:DBR]

The database is backed up regularly (e.g., daily) using appropriate tools. Automated scripts are scheduled to ensure that backups are performed consistently.

3.2.5.4 Testing [R:T]

Comprehensive unit tests are written for both backend and frontend components using a testing framework and related libraries. Testing is conducted to ensure that different modules of the application work together seamlessly.

3.2.6 Usability [NFR:U]

Usability is essential for the success of Suite Dreams, as it is used by a diverse range of students with varying levels of technical proficiency. The system focuses on being intuitive, accessible, and easy to navigate. The User Interface (UI) is designed following user-centered principles, ensuring ease of navigation with a clean and minimalist layout. Important features, such as dorm selection, questionnaire completion, and submission, are easily accessible, making the process straightforward for users. Accessibility is prioritized to make the system usable for individuals with disabilities. This includes support for keyboard navigation, screen reader compatibility, and adjustable font sizes. Performance feedback is provided to users immediately after performing actions like submitting applications or editing preferences. Indicators such as loading spinners or progress bars show when the system is processing, while clear and specific error messages guide users in resolving issues. User testing is conducted throughout development with a wide range of students to identify and address usability issues. This includes formative testing during development and summative testing post-development. The testing process evaluates task completion rates, time to complete tasks, and overall user satisfaction to ensure the system meets the needs of its users.