Room Booking System for Student Dormitories

Part 1: Exhaustive Functional & Non-Functional Requirements

For each requirement, **justify why it belongs to its category** and explain how it contributes to the overall project goals.

1. **Functional Requirements (at least 10-15)** – What the system must do, including key features, actions, and workflows.

- **Can register profile:**
  This is a functional requirement because it defines a specific action the system must allow. It contributes to the project goal by associating bookings with a specific profile, allowing for a personalized experience and increased convenience by consolidating all services in one location.
- **Can view available rooms:**
  This is a functional requirement because it defines a specific action the system must allow. It contributes to the project goal by providing students with up-to-date information on room availability—a critical functionality of this system.
- **Filter by type of room:**
  This is a functional requirement because it defines a specific action the system must allow. It contributes to the project goal by enhancing the user experience, allowing the user to narrow down options related to their search.
- **Filter by type of facilities:**
  This is a functional requirement because it defines a specific action the system must allow. It contributes to the project goal by improving the user experience, enabling the user to narrow down options based on the facilities available.
- **Book room:**
  This is a functional requirement because it defines a specific action the system must allow. It contributes to the project goal by providing the user with an interface to facilitate room reservations, offering several options such as preferred dates, roommates, and social preferences.
- **Submit repair requests:**
  This is a functional requirement because it defines a specific action the system must allow. It contributes to the project goal by providing a mechanism to report maintenance issues and initiate a maintenance workflow.
- **Track repair status:**
  This is a functional requirement because it defines a specific action the system must allow. It contributes to the project goal by providing the user with an interface to monitor the progress and status of their maintenance or repair requests.

- **Admins can assign rooms:**
  This is a functional requirement because it is an administrative action critical to the system’s functionality. It contributes to the project goal by enabling administrators to assign rooms efficiently.
- **Admins can approve bookings:**
  This is a functional requirement because it is an administrative action critical to the system’s functionality. It contributes to the project goal by enabling administrators to approve bookings and ensure that all reservations comply with established policies.
- **Admins can view occupancy reports:**
  This is a functional requirement because it is an administrative action critical to the system’s functionality. It contributes to the project goal by providing administrators with oversight through occupancy reports, facilitating effective resource management.
- **Receive reminders for booking deadlines:**
  This is a functional requirement because it is a user-experience action critical to the system’s functionality. It contributes to the project goal by providing users with timely reminders for booking deadlines.
- **Receive reminders for maintenance updates:**
  This is a functional requirement because it is a user-experience action critical to the system’s functionality. It contributes to the project goal by providing users with timely reminders for maintenance updates.

2. **Non-Functional Requirements (at least 10-15)** – How the system should perform, covering aspects like security, usability, scalability, performance, maintainability, compliance, and more.

- **Real-time room availability updates:**
  This is a non-functional requirement because it defines a performance characteristic of the system rather than a specific action. It contributes to the project goal by ensuring that room availability data is continuously updated, providing students with accurate and timely information. This minimizes booking conflicts.
- **Authenticate user details:**
  This is a non-functional requirement because it addresses the security aspect of the system. It contributes to the project goal by protecting sensitive user information and ensuring that only authorized users can access the system, maintaining data integrity and compliance with privacy standards.
- **Intuitive interface:**
  This is a non-functional requirement because it pertains to the usability of the system rather than a specific function. It contributes to the project goal by offering a user-friendly design that reduces the learning curve, improves overall user satisfaction, and streamlines the room booking process.

- **Accessibility for students with disabilities:**
  This is a non-functional requirement since it establishes a design and usability standard that promotes inclusivity. It supports the project goal by ensuring the system is usable by all students, regardless of their physical abilities, thus expanding the user base and meeting legal accessibility standards.
- **Support further scaling:**
  This is a non-functional requirement because it addresses the system's architecture and performance capacity rather than a specific action. It contributes to the project goal by ensuring that the system can accommodate increased user loads and data volume over time without compromising performance or reliability.
- **Fast booking:**
  This is a non-functional requirement because it specifies a system performance attribute, emphasizing the need for speed and responsiveness during the booking process. It contributes to the project goal by reducing wait times, enhancing the user experience, and ensuring that the booking process remains efficient even during peak usage periods.
- **Implement backups:**
  This requirement is classified as non-functional since it focuses on system reliability and data management rather than essential features. It supports the project objective by protecting data through consistent backups, allowing the system to swiftly recover from failures or data loss, and ensuring uninterrupted service for users.
- **Loading state indicators:**
  This is a non-functional requirement because it specifies how the system should communicate processing status rather than what it processes. It contributes to the project goal by showing a simple loading spinner whenever the system is processing a booking request or fetching room data, letting users know the system is working.
- **Confirmation messages:**
  This is a non-functional requirement because it specifies how the system provides feedback rather than what it does. It contributes to the project goal by displaying clear confirmation messages for all important actions, helping users know their actions were completed.
- **Consistent button placement:**
  This is a non-functional requirement because it defines interface layout standards rather than system functions. It contributes to the project goal by placing all action buttons such as submit, cancel, and back in consistent locations across different forms, making the interface predictable and easy to use.

Part 2: Tools & Technologies Selection

For each selected tool/technology, **justify your choice** in at least one paragraph, explaining:

- Why it is the best fit for your project.
- How it supports your project’s functional and non-functional requirements.

1. **Development Tools** (IDEs, programming languages, frameworks).

- **Programming Language - Python:** the Python programming language is a good choice for our project as it has an expansive library ecosystem that can provide core functionality for our application such as user authentication, booking management, and request handling. Python also has extensive support for integrating with various SQL databases. Python will support our non-functional requirements by offering solid performance, security, and scalability, ensuring our system will remain reliable and maintainable with time.
- **IDE - VS Code:** the code editor Visual Studio Code will be used because of its built-in Git integration, as well as add-ons for Python and Flask.
- **Web Framework - Flask:** the Flask Python web framework is lightweight and modular for our functional requirements, and seamlessly integrates with Python and SQL databases allowing our frontend business logic to communicate effectively with our data management backend. Flask is maintainable, scaleable, and highly performant, ensuring our system can handle new features and user load increases.

2. **Version Control Systems** (Git, GitHub, GitLab, etc.).

- For our project, GitHub would be the most ideal version control system to use. Given that our project has many different components, using GitHub will give us a singular platform for tracking the project and making changes collaboratively. Additionally, the group is more familiar with GitHub so using it would make the most sense and give us a slight advantage.

3. **Deployment & Hosting Technologies** (cloud platforms, relational databases, client-server models).

- **Database - SQL:** using SQL will allow us to explore further relational database management systems, such as PostgreSQL and MySQL, as the course progresses. This choice keeps our choice of RDMS open in the coming weeks. Furthermore, SQL directly allows for the functional requirements listed above such as providing real-time information on occupancy and room availability.