1. Use Cases

1.1 **Student Registration**

|  |
| --- |
| **Use Case Name:** Student Registration **ID:** UC-1 **Priority:** High |
| **Actor:** Student |
| **Description:** Allows students to register and access the system. |
| **Trigger:** A new student wants to create an account. |
| **Preconditions:**   1. The system is accessible online. |
| **Normal Flow:**   1. Students navigate to the registration page. 2. Fill in the registration form with name, email, and password. 3. Submits the form. 4. The system validates email and stores user data. 5. The system confirms registration. |
| **Postconditions:**   1. The student is registered and can log in. |

* 1. **Student Login**

|  |
| --- |
| **Use Case Name:** Student Login **ID:** UC-2 **Priority:** High |
| **Actor:** Student |
| **Description:** Allows students to log into the system. |
| **Trigger:** A registered student attempts to log in. |
| **Preconditions:**   1. The student must be registered. |
| **Normal Flow:**   1. Students navigate to the login page. 2. Enters login credentials. 3. System validates credentials. 4. If valid, access is granted; otherwise, an error message is displayed. |
| **Postconditions:**   1. The student gains access to their account. |

* 1. **Room Allocation**

|  |
| --- |
| **Use Case Name:** Room Allocation **ID:** UC-3 **Priority:** High |
| **Actor:** Student, Admin |
| **Description:** Assigns rooms to students automatically or manually |
| **Trigger:** A student completes registration. |
| **Preconditions:**   1. The student is registered, and rooms are available. |
| **Normal Flow:**   1. Admin accesses the room allocation module. 2. The system suggests an available room based on predefined rules. 3. Admin approves or modifies the room assignment. 4. System updates room allocation records. |
| **Postconditions:**   1. The student is assigned a room. |

**1.4 Rent Payment**

|  |
| --- |
| **Use Case Name:** Rent Payment **ID:** UC-4 **Priority:** High |
| **Actor:** Student |
| **Description:** Enables students to track and pay their rent. |
| **Trigger:** Student initiates a rent payment. |
| **Preconditions:**   1. Students are logged in and rent is due. |
| **Normal Flow:**   1. Student navigates to the payment section. 2. System displays rent is due. 3. Student selects payment methods and confirms payment. 4. The system updates payment records. |
| **Postconditions:**   1. Rent is marked as paid. |

**1.5 Maintenance/Complaint Request**

|  |
| --- |
| **Use Case Name:** Maintenance/Complaint Request **ID:** UC-5 **Priority:** High |
| **Actor:** Student, Admin |
| **Description:** Allows students to submit maintenance complaints and track resolution. |
| **Trigger:** Student experiences a maintenance issue. |
| **Preconditions:**   1. Students are logged in. |
| **Normal Flow:**   1. Student navigates to maintenance request section. 2. Submits details of the issue. 3. System logs request and notifies admin. 4. Admin assigns maintenance staff. 5. Staff resolves issue and updates status. |
| **Postconditions:**   1. Maintenance requests are addressed. |

* 1. **Leave/Exit Request Management**

|  |
| --- |
| **Use Case Name:** Leave/Exit Request Management **ID:** UC-6 **Priority:** Medium |
| **Actor:** Student, Admin |
| **Description:** Allows students to submit leave or exit requests. |
| **Trigger:** A student wants to leave the hostel temporarily or permanently. |
| **Preconditions:**   1. Students are logged in. |
| **Normal Flow:**   1. Student submits a leave request with dates and reason. 2. Admin reviews the request. 3. Admin approves or rejects the request. |
| **Postconditions:**   1. Leave requests are approved or denied. |

**1.7 Report Generation**

|  |
| --- |
| **Use Case Name:** Report Generation **ID:** UC-7 **Priority:** High |
| **Actor:** Admin |
| **Description:** Generates occupancy reports. |
| **Trigger:** Admin requests a report. |
| **Preconditions:**   1. The system has sufficient data. |
| **Normal Flow:**   1. Admin selects report type (occupancy, finance, maintenance). 2. The system fetches relevant data. 3. Report is generated and exported as PDF. |
| **Postconditions:**   1. Reports are generated and stored for reference. |

A diagram of a software project

AI-generated content may be incorrect.2.0 Level 0 DFD**:**

## 2.1 Process Model with Data Flow Diagram (DFD)

**Entities and Data Stores:**

1. **Student (User)**
   * The primary actor who interacts with the system to register, request rooms, pay rent, and submit maintenance requests.
   * Students can also log guest entries and receive room allocation details.
2. **Admin**
   * Responsible for approving student registrations, allocating rooms, processing payments, and handling maintenance requests.
   * Admin generates reports and ensures system operations run smoothly.
3. **Data Stores (System Databases)**
   * **D1: User Data Store –** Stores student registration details and login credentials.
   * **D2: Room Data Store –** Holds room allocation details and availability.
   * **D3: Payment Data Store –** Maintains records of rent payments.
   * **D4: Maintenance Data Store** – Stores details of maintenance requests and their resolution status.

**Processes and Their Functions:**

**1.0 User Registration**

* **Purpose:** Allows students to create an account for hostel management.
* **Input:** Student enters name, email, password to sign up**.**
* **Process:**
  1. The student accesses the registration page.
  2. The system validates user details (checks if email already exists).
  3. If validation is successful, the system stores the user’s details in D1: User Data Store.
* **Output:**
  1. If registration is successful, the student can log in.
  2. If registration fails (e.g., duplicate email), an error message is displayed.

**2.0 Room Allocation**

* **Purpose:** Assigns rooms to students and allows them to view room details.
* **Input**: Student requests a room.
* **Process:**
  1. Student requests room allocation.
  2. System queries D2: Room Data Store for available rooms.
  3. Admin approves room assignment.
  4. System updates D2 with the new allocation.
  5. Students receive room allocation confirmation**.**
* **Output:**
  1. Room is allocated successfully.
  2. Students can view room details.

**3.0 Record Payments**

* **Purpose:** Allows students to pay rent and track payment history.
* **Input:** Student makes payment.
* **Process:**
  1. Students initiate rent payment.
  2. The system fetches payment details from D3: Payment Data Store.
  3. Payment is processed and recorded in D3.
  4. Payment confirmation is sent to the student.
* **Output:**
  1. Rent is marked as paid.
  2. Receipt is generated for the student.

**4.0 Manage Maintenance**

* **Purpose:** Handles maintenance requests submitted by students**.**
* **Input:** Student submits a maintenance request.
* **Process:**
  1. Student submits a maintenance request.
  2. System logs the request in D4: Maintenance Data Store.
  3. Admin reviews and assigns maintenance staff.
  4. Staff update the system once the issue is resolved.
  5. Students receive maintenance completion notification.
* **Output:**
  1. The maintenance issue is resolved.
  2. Status updates are sent to the student.

**Data Flows in the System:**

1. **Student Registration:** Data is stored in D1 (User Data Store).
2. **Room Allocation:** System retrieves availability from D2 (Room Data Store) and assigns rooms.
3. **Rent Payment**: Payments are processed and stored in D3 (Payment Data Store).
4. **Maintenance Requests:** Student requests are logged, assigned, and updated in D4 (Maintenance Data Store).