USE CASE DESCRIPTION

Login

1.	**Introduction:** This usecase documents the steps that must follow in order to log into the HPMS.
2.	**Actors:** Receptionist, Administrator, Doctor, Nurse
3.	**Precondition:** The user must have a valid login ID and password.
4.	**Postcondition:** If this use case is successful, the actor is logged into the system. If not, the system remains unchanged.
5.	**Flow of events:** **5.1. Basic Flow:** Starts when the actor wishes into login into the HPMS. 1. The system requests that the actor specify the function he/she would like to perform (either Login, Change Password). 2. Once the actor provides the requested information, one of the following flows is executed: • If the actor selects "Login", the **Login** flow is executed. • If the actor selects "Change Password", the **Change Password** flow is executed. **Basic Flow 1: Login** 1. The system requests that the actor enter his/her login ID and password information. 2. The actor enters his/her login ID and password. 3. The actor enters into the system. **Basic Flow 2: Change Password** 1. The system requests that the actor enter login ID, old password, new password and confirm the new password information. 2. The actor enters login ID, old password, new password and confirms the new password information. 3. The system validates the entered new password and password is confirmed. **5.2. Alternative Flow:** **Alternative Flow 1: Invalid Login ID/Password** If in the Login flow, the actor enters an invalid login ID and/or password empty, the system displays an error message. The actor returns to the beginning of the basic flow. **Alternative Flow 2: Invalid Entry** If in the Change Password flow, the actor enters an invalid login ID, old password, new password or the new password does not match with the confirm password, the system displays an error message. The actor returns to the beginning of the basic flow. **Alternative Flow 3: User Exits** This allows the user to exit during the use case. The use case ends.
6.	**Special requirements:** None
7.	**Associated use cases:** None

Patient Registration

1.	**Introduction:** This usecase documents the steps that the Receptionist must follow in order to register patient.
2.	**Actors:** Receptionist
3.	**Precondition:** The Receptionist must be logged onto the system before the use case begins.
4.	**Postcondition:** If the registration is successful, a patient ID must be allocated and the database is updated, else the system remains unchanged.
5.	**Flow of events:** **5.1. Basic Flow: New Patient Registration** 1. The patient details is entered manully. 2. Patient is registered by allocating a patient ID to new patient. 3. Billing details is generated and amount paid in cash. 4. The new patient details is saved into the database. **5.2. Alternative Flow:** **Alternative Flow 1: User Exits** This allow the user to exit at any time during the use case. The use case ends.
6.	**Special requirements:** None
7.	**Associated use cases:** Login**,** Generate Bills

Allocate Doctor

1.	**Introduction:** This usecase documents the steps that the Receptionist must follow in order to allocate doctor to new or old patients.
2.	**Actors:** Receptionist
3.	**Precondition:** The Receptionist must be logged onto the system before the use case begins. Patient must be registered with hospital.
4.	**Postcondition:** If the use case is successful, doctor visting hour is allocated to patient and the database is updated, else the system remains unchanged.
5.	**Flow of events:** **5.1. Basic Flow: Allocate Doctor** 1. Select the specified doctor. 2. Empty slot with date for doctor is obtained. 3. Doctor is allocated to the visiting patient and information is saved into the database. **5.2. Alternative Flow:** **Alternative Flow 1: Doctor Not Available** If the required doctor is not found on particular date inform patient, "doctor not available" on that particular date and exit. **Alternative Flow 2: Slot Not Available** If the required doctor is found then check for empty slot on particular date. If the empty slot is not found on particular date, inform patient, "slot not available" on that particular date and exit. **Alternative Flow 3: User Exits** This allow the user to exit at any time during the use case. The use case ends.
6.	**Special requirements:** None
7.	**Associated use cases:** Login, Patient Registration

Allocate Nurse

1.	**Introduction:** This usecase module describe the flow of events for allocating nurses to the required patients.
2.	**Actors:** Receptionist
3.	**Precondition:** The Receptionist must be logged onto the system before the use case begins. Patient must be pre-allocated doctor.
4.	**Postcondition:** If the use case is successful, a nurse is assigned to the patient and the database is updated, else the system remains unchanged.
5.	**Flow of events:** **5.1 Basic Flow: Allocate Nurse** 1. Select from the available nurses in particular department. 2. A nurse is allocated to the admitted patient and the information is saved. **5.2. Alternative Flow: User Exits** This allow the user to exit at any time during the use case. The use case ends.
6.	**Special requirements:** None
7.	**Associated use cases:** Login, Allocate Doctor

Allocate Wards

1.	**Introduction:** This use case module describes the flow of events for allocating beds or wards to the required patients.
2.	**Actors:** Receptionist
3.	**Precondition:** The Receptionist must be logged onto the system before the use case begins. Patient must be pre-allocated doctor and nurse. Bed or ward type that the patient need must be known prior to allocation.
4.	**Postcondition:** If the use case is successful, a ward is allocated to the patient and the database is updated, else the system remains unchanged.
5.	**Flow of events:** **5.1. Basic Flow: Allocate Ward** 1. Specified ward type is selected. 2. The system display the list of available beds or wards. 3. Ward is allocated as mentioned by patient and information is saved into the database. **5.2. Alternative Flow:** **Alternative Flow 1: Ward Not Available** If the required bed or ward is not available, then the request for allocation is denied and use case ends. **Alternative Flow 2: User Exits** This allow the user to exit at any time during the use case. The use case ends.
6.	**Special requirements:** None
7.	**Associated use cases:** Login

Maintain Doctor Details

1.	**Introduction:** This usecase documents the steps that the Administrator must follow in order to maintain doctor details and add, update, delete and view doctor information.
2.	**Actors:** Administrator
3.	**Precondition:** The Administrator must be logged onto the system before this use case begins.
4.	**Postcondition:** If the use case is successful, then the doctor information is added, updated, deleted or viewed. Otherwise, the system state is unchanged.
5.	**Flow of events:** **5.1. Basic Flow:** This use case starts when the Administrator wishes to add/update/delete/view doctor information. 1. The system requests that the Administrator specify the function he/she would like to perform (either add, update, delete or view a doctor). 2. Once the Administrator provides the requested information, one of the subflows is executed. • If the Administrator selects "Add a Doctor", the **Add a Doctor** subflow is executed. • If the Administrator selects "Update a Doctor", the **Update a Doctor** subflow is executed. • If the Administrator selects "Delete a Doctor", the **Delete a Doctor** subflow is executed. • If the Administrator selects "View a Doctor", the **View a Doctor** subflow is executed. **Basic Flow 1: Add a Doctor** The system requests that the Administrator enter the doctor information. This includes: • Doctor ID • Doctor Name • Specialization • Sitting Location • Visiting Days • Visiting Time Once the Administrator provides the requested information, the doctor is added to the system. **Basic Flow 2: Update a Doctor** 1. The system requests that the Administrator enter the Doctor ID. 2. The Administrator enters the Doctor ID. 3. The system retrieves and display the doctor information. 4. The Administrator makes the desired changes to the doctor information. This includes any of the information specified in the **Add a Doctor** subflow. 5. Once the Administrator updates the necessary information, the system updates the doctor information with the updated information. **Basic Flow 3: Delete a Doctor** 1. The system requests that the Administrator specify the Doctor ID. 2. The Administrator enters the Doctor ID. The system retrieves and display the doctor information. 3. The system prompts the Administrator to confirm the deletion of the doctor record. 4. The Administrator verifies the deletion. 5. The system deletes the record.

	Basic Flow 4: View a Doctor 1. The system requests that the Administrator specify the Doctor ID. 2. The system retrieves and displays the doctor information. **5.2. Alternative Flow:** **Alternative Flow 1: Invaild Entry** If in the **Add a Doctor** or **Update a Doctor** flow, the actor enters invalid Doctor ID/Doctor Name/Specialization/Sitting Location/Visiting Days/Visiting Time or leaves the Doctor ID/ Doctor Name/Specialization/Sitting Location/Visiting Days/Visiting Time empty, the system displays an appropriate error message. The actor returns to the basic flow and may reenter the invalid entry. **Alternative Flow 2: Doctor Already Exists** If in the **Add a Doctor** flow, a doctor with a specified Doctor ID already exists, the system displays an error message. The administrator returns to the basic flow and may reenter the doctor. **Alternative Flow 3: Doctor Not Found** If in the **Update a Doctor** or **Delete a Doctor** or **View a Doctor** flow, the doctor information with the specified Doctor ID does not exists, the system displays an error message. The Administrator returns to the basic flow and may reenter the Doctor ID. **Alternative Flow 4: Update Cancelled** If in the **Update a Doctor** flow, the Administrator decides not to update the doctor, the update is cancelled and the **Basic Flow** is restarted at the beginning. **Alternative Flow 5: Delete Cancelled** If in the **Delete a Doctor** flow, the Administrator decides not to delete the doctor, the delete is cancelled and the **Basic Flow** is restarted at the beginning. **Alternative Flow 6: User Exits** This allows the user to exit at any time during the use case. The use case ends.
6.	**Special requirements:** None
7.	**Associated use cases:** Login

Maintain Nurse Details

1.	**Introduction:** This usecase documents the steps that the Administrator must follow in order to maintain nurse details and add, update, delete and view nurse information.
2.	**Actors:** Administrator
3.	**Precondition:** The Administrator must be logged onto the system before this use case begins.
4.	**Postcondition:** If the use case is successful, then the nurse information is added, updated, deleted or viewed. Otherwise, the system state is unchanged.
5.	**Flow of events:** **5.1. Basic Flow:** 1. This use case starts when the Administrator wishes to add/update/delete/view nurse information. 2. The system requests that the Administrator specify the function he/she would like to perform (either add, update, delete or view a nurse). 3. Once the Administrator provides the requested information, one of the subflows is executed. • If the Administrator selects “Add a Nurse”, the **Add a Nurse** subflow is executed. • If the Administrator selects “Update a Nurse”, the **Update a Nurse** subflow

is executed.
- If the Administrator selects “Delete a Nurse”, the **Delete a Nurse** subflow is executed.
- If the Administrator selects “View a Nurse”, the **View a Nurse** subflow is executed.

Basic Flow 1: Add a Nurse

The system requests that the Administrator enter the nurse information. This includes:

- Nurse ID
- Nurse Name
- Department
- Visiting Days
- Visiting Time

Once the Administrator provides the requested information, the nurse is added to the system.

Basic Flow 2: Update a Nurse

1. The system requests that the Administrator enter the nurse ID.
2. The Administrator enters the nurse ID.
3. The system retrieves and display the nurse information.
4. The Administrator makes the desired changes to the nurse information. This includes any of the information specified in the **Add a Nurse** subflow.
5. Once the Administrator updates the necessary information, the system updates the nurse information with the updated information.

Basic Flow 3: Delete a Nurse

1. The system requests that the Administrator specify the nurse ID.
2. The Administrator enters the nurse ID. The system retrieves and display the nurse information.
3. The system prompts the Administrator to confirm the deletion of the nurse record.
4. The Administrator verifies the deletion.
5. The system deletes the record.

Basic Flow 4: View a Nurse

1. The system requests that the Administrator specify the nurse ID.
2. The system retrieves and displays the nurse information.

5.2. Alternative Flow:

Alternative Flow 1: Invaild Entry

If in the **Add a Nurse** or **Update a Nurse** flow, the actor enters invalid Nurse ID/ Nurse Name/Specialization/Department/Visiting Days/Visiting Time or leaves the Nurse ID/ Nurse Name/Specialization/Department/Visiting Days/Visiting Time empty, the system displays an appropriate error message. The actor returns to the basic flow and may reenter the invalid entry.

Alternative Flow 2: Nurse Already Exists

If in the **Add a Nurse** flow, a nurse with a specified nurse ID already exists, the system displays an error message. The administrator returns to the basic flow and may reenter the nurse.

Alternative Flow 3: Nurse Not Found

If in the **Update a Nurse** or **Delete a Nurse** or **View a Nurse** flow, the nurse information with the specified nurse ID does not exists, the system displays an error message. The Administrator returns to the basic flow and may reenter the nurse ID.

Alternative Flow 4: Update Cancelled

If in the **Update a Nurse** flow, the Administrator decides not to update the nurse, the update is cancelled and the **Basic Flow** is restarted at the beginning.

Alternative Flow 5: Delete Cancelled

If in the **Delete a Nurse** flow, the Administrator decides not to delete the nurse, the delete is cancelled and the **Basic Flow** is restarted at the beginning.

	Alternative Flow 6: User Exits This allows the user to exit at any time during the use case. The use case ends.
6.	**Special requirements:** None
7.	**Associated use cases:** Login

Maintain Patient Details

1.	**Introduction:** This usecase documents the steps that the Administrator/Receptionist/Doctor/ Nurse must follow in order to maintain patient details and add, update, delete and view patient information.
2.	**Actors:** Receptionist, Administrator, Doctor, Nurse
3.	**Precondition:** The Administrator/Receptionist/Doctor/Nurse must be logged onto the system before this use case begins.
4.	**Postcondition:** If the use case is successful, then the patient information is added, updated, deleted or viewed. Otherwise, the system state is unchanged.
5.	**Flow of events:** **5.1. Basic Flow:** This use case starts when the Administrator wishes to add/update/delete/view patient information. 1. The system requests that the Administrator specify the function he/she would like to perform (either add, update, delete or view a patient). 2. Once the Administrator provides the requested information, one of the subflows is executed. • If the Administrator selects “Add a Patient”, the **Add a Patient** subflow is executed. • If the Administrator selects “Update a Patient”, the **Update a Patient** subflow is executed. • If the Administrator selects “Delete a Patient”, the **Delete a Patient** subflow is executed. • If the Administrator/Receptionist/Doctor/Nurses selects “View a Patient”, the **View a Patient** subflow is executed. **Basic Flow 1: Add a Patient** The system requests that the Administrator enter the patient information. This includes: • Patient ID • Patient Name • Relative Name • Phone No • Address • Ailments • Doctor Assigned • Visiting Date Once the Administrator provides the requested information, the patient is added to the system. **Basic Flow 2: Update a Patient** 1. The system requests that the Administrator enter the patient ID. 2. The Administrator enters the patient ID. 3. The system retrieves and display the patient information. 4. The Administrator makes the desired changes to the patient information. This includes any of the information specified in the **Add a Patient** subflow.

	5. Once the Administrator updates the necessary information, the system updates the patient information with the updated information. **Basic Flow 3: Delete a Patient** 1. The system requests that the Administrator specify the patient ID. 2. The Administrator enters the patient ID. The system retrieves and display the patient information. 3. The system prompts the Administrator to confirm the deletion of the patient record. 4. The Administrator verifies the deletion. 5. The system deletes the record. **Basic Flow 4: View a Patient** 1. The system requests that the Administrator/Receptionist/Doctor/Nurses specify the patient ID. 2. The system retrieves and displays the patient information. **5.2. Alternative Flow:** **Alternative Flow 1: Invaild Entry** If in the **Add a Patient** or **Update a Patient** flow, the actor enters invalid Patient ID/Patient Name/Relative Name/Phone No/Address/Ailments/Doctor Assigned/Visiting Date or leaves the Patient ID/Patient Name/Relative Name/Phone No/Address/Ailments/Doctor Assigned/ Visiting Date empty, the system displays an appropriate error message. The actor returns to the basic flow and may reenter the invalid entry. **Alternative Flow 2: Patient Already Exists** If in the **Add a Patient** flow, a patient with a specified patient ID already exists, the system displays an error message. The administrator returns to the basic flow and may reenter the patient. **Alternative Flow 3: Patient Not Found** If in the **Update a Patient** or **Delete a Patient** or **View a Patient** flow, the patient information with the specified patient ID does not exists, the system displays an error message. The Administrator returns to the basic flow and may reenter the patient ID. **Alternative Flow 4: Update Cancelled** If in the **Update a Patient** flow, the Administrator decides not to update the patient, the update is cancelled and the **Basic Flow** is restarted at the beginning. **Alternative Flow 5: Delete Cancelled** If in the **Delete a Patient** flow, the Administrator decides not to delete the patient, the delete is cancelled and the **Basic Flow** is restarted at the beginning. **Alternative Flow 6: User Exits** This allows the user to exit at any time during the use case. The use case ends.
6.	**Special requirements:** None
7.	**Associated use cases:** Login

Maintain Ward Details

1.	**Introduction:** This usecase documents the steps that the Administrator must follow in order to maintain ward details and add, update, delete and view ward information.
2.	**Actors:** Administrator, Nurses
3.	**Precondition:** The Administrator/Nurses must be logged onto the system before this use case begins.
4.	**Postcondition:** If the use case is successful, then the ward information is added, updated, deleted or viewed. Otherwise, the system state is unchanged.
5.	**Flow of Events:**

5.1. Basic Flow:

This use case starts when the Administrator wishes to add/update/delete/view ward information.

1. The system requests that the Administrator specify the function he/she would like to perform (either add, update, delete or view a ward).
2. Once the Administrator provides the requested information, one of the subflows is executed.
    - If the Administrator selects “Add a Ward”, the **Add a Ward** subflow is executed.
    - If the Administrator selects “Update a Ward”, the **Update a Ward** subflow is executed.
    - If the Administrator selects “Delete a Ward”, the **Delete a Ward** subflow is executed.
    - If the Administrator/Nurse selects “View a Ward”, the **View a Ward** subflow is executed.

Basic Flow 1: Add a Ward

The system requests that the Administrator enter the ward information. This includes:

- Ward No.
- Ward Name
- Depatment
- Location
- Incharge
- Status

Once the Administrator provides the requested information, the ward is added to the system.

Basic Flow 2: Update a Ward

1. The system requests that the Administrator enter the ward No.
2. The Administrator enters the ward No.
3. The system retrieves and display the ward information.
4. The Administrator makes the desired changes to the ward information. This includes any of the information specified in the **Add a Ward** subflow.
5. Once the Administrator updates the necessary information, the system updates the ward information with the updated information.

Basic Flow 3: Delete a Ward

1. The system requests that the Administrator specify the ward No.
2. The Administrator enters the ward No. The system retrieves and display the ward information.
3. The system prompts the Administrator to confirm the deletion of the ward record.
4. The Administrator verifies the deletion.
5. The system deletes the record.

Basic Flow 4: View a Ward

1. The system requests that the Administrator/Nurse specify the ward No.
2. The system retrieves and displays the ward information.

5.2. Alternative Flow:

Alternative Flow 1: Invaild Entry

If in the **Add a Ward** or **Update a Ward** flow, the actor enters invalid ward No/ ward Name/Department/Location/Incharge/Status or leaves the ward No/ ward Name/Department/Location/Incharge/Status empty, the system displays an appropriate error message. The actor returns to the basic flow and may reenter the invalid entry.

Alternative Flow 2: Ward Already Exists

If in the **Add a Ward** flow, a ward with a specified ward No already exists, the system displays an error message. The Administrator returns to the basic flow and may reenter the ward details.

	Alternative Flow 3: Ward Not Found If in the **Update a Ward** or **Delete a Ward** or **View a Ward** flow, the ward information with the specified ward No does not exists, the system displays an error message. The Administrator returns to the basic flow and may reenter the ward No. **Alternative Flow 4: Update Cancelled** If in the **Update a Ward** flow, the Administrator decides not to update the ward, the update is cancelled and the **Basic Flow** is restarted at the beginning. **Alternative Flow 5: Delete Cancelled** If in the **Delete a Ward** flow, the Administrator decides not to delete the ward, the delete is cancelled and the **Basic Flow** is restarted at the beginning. **Alternative Flow 6: User Exits** This allows the user to exit at any time during the use case. The use case ends.
6.	**Special requirements:** None
7.	**Associated use cases:** Login

Generate Bills

1.	**Introduction:** This usecase module describe the flow of events for generating bills.
2.	**Actors:** Receptionist, Administrator.
3.	**Precondition:** The Receptionist or Administrator must be logged onto the system before the use case begins. Patient must be registered.
4.	**Postcondition:** If the use case is successful, a bill is generated and the database is updated, else the system remains unchanged.
5.	**Flow of events:** **5.1 Basic Flow: Generate Bill** 1. A bill amount is calculated based on the kind of treatment. 2. A bill is generated with a unique bill ID. 2. Bill details along with patient ID is saved in the database. **5.2. Alternative Flow: User Exits** This allow the user to exit at any time during the use case. The use case ends.
6.	**Special requirements:** None
7.	**Associated use cases:** Login

Generate Reports

1.	**Introduction:** This usecase module describe the flow of events for generating reports.
2.	**Actors:** Receptionist
3.	**Precondition:** The user must be logged onto the system before the use case begins.
4.	**Postcondition:** If the use case is successful, the system displays the details based on selected operations. No changes are made to the database. Else, no report is displayed.
5.	**Flow of events:** **5.1 Basic Flow: Generate Reports** 1. The operator issues the command to generate the below reports. • Details of all registered patients. • Details of appointments. • Details of allocated beds/wards.

	2. The system displays a report including details of all the operations. **5.2. Alternative Flow** None
6.	**Special requirements:** None
7.	**Associated use cases:** Login