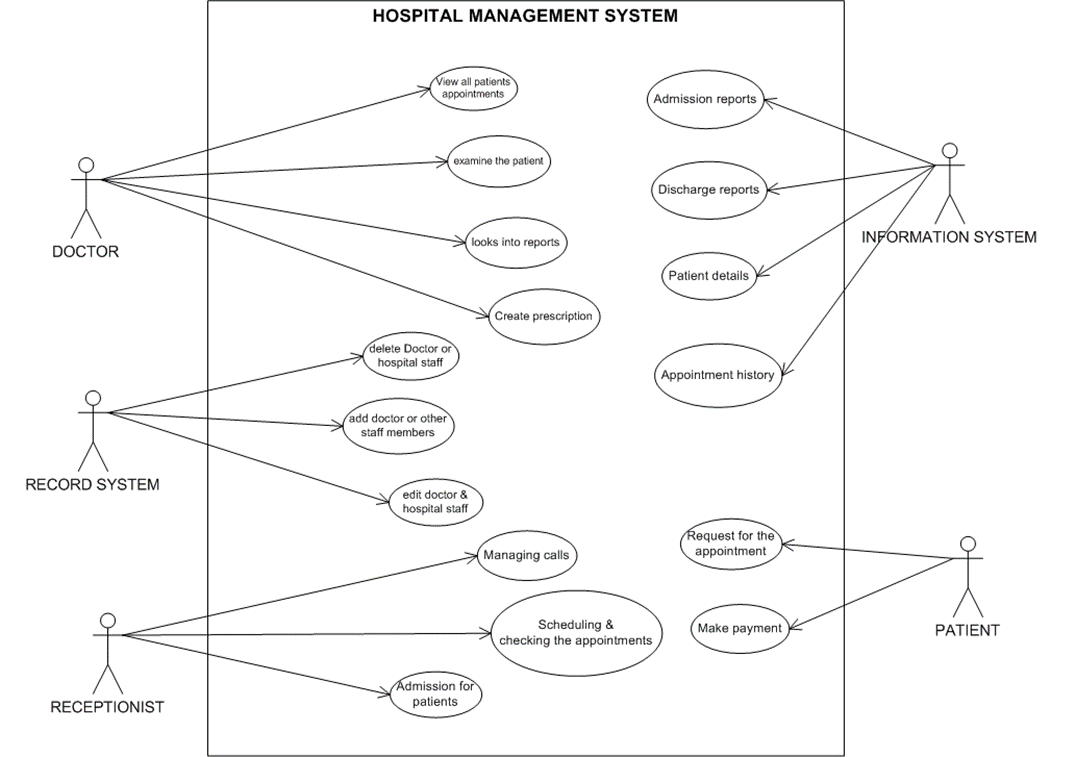
**CAPSTONE PROJECT PART-2/2**

6. Document 6- Please prepare a use case diagram, activity diagram and a use

case specification document.

Ans. **Use case diagram**



b. **Activity diagram**

Patient visit hospital

Make payment

Prepare bill

Get dis-charge

Perform Operation

Assign operation theatre

Assign Doctor

Allocate Room

Check Room Availability

Suggest treatment

Assign Doctor

need to admit

yes

NO

No

Give Treatment

need to shift

No

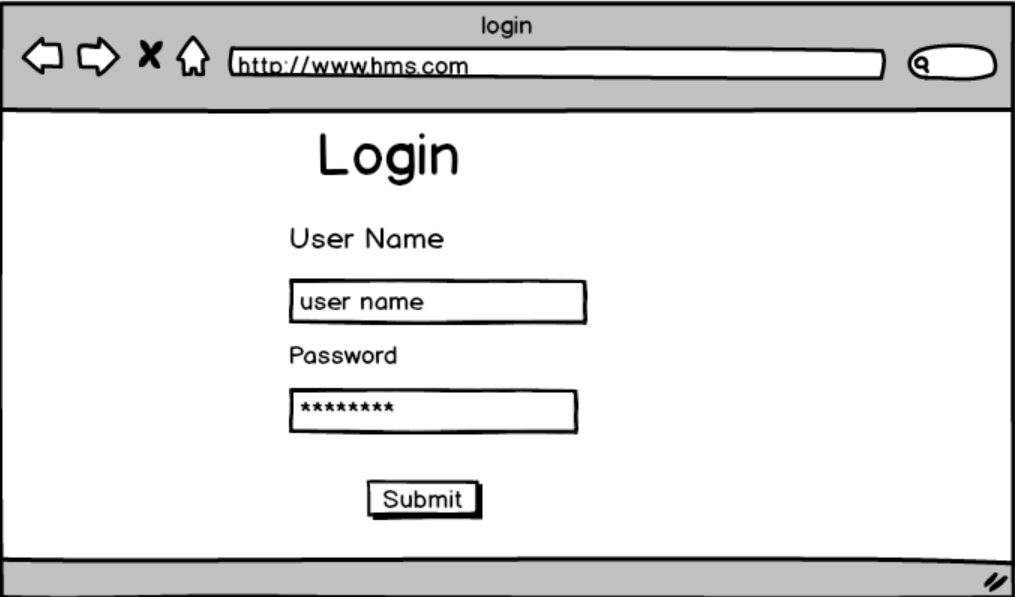
Yes

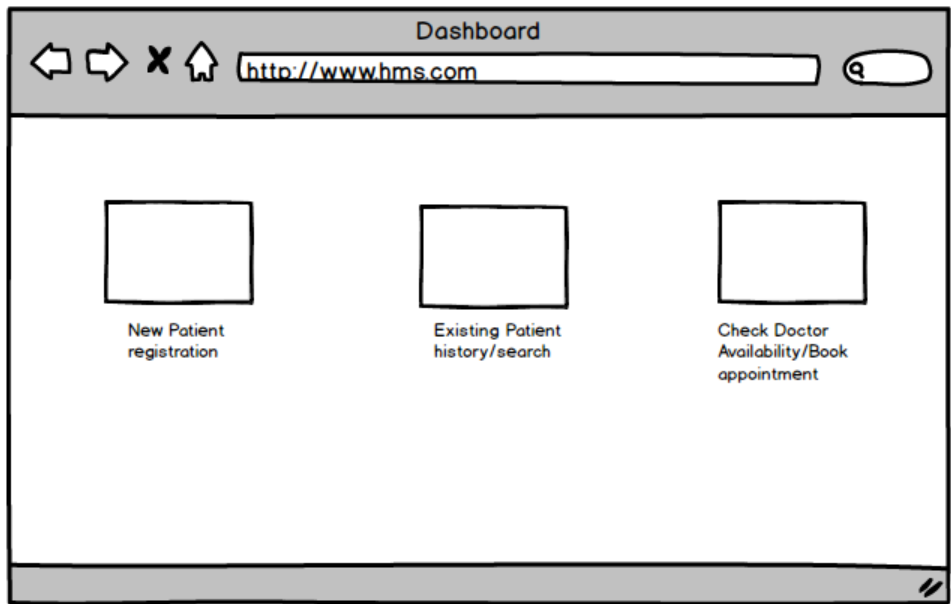
**C.** **Use case specification document**

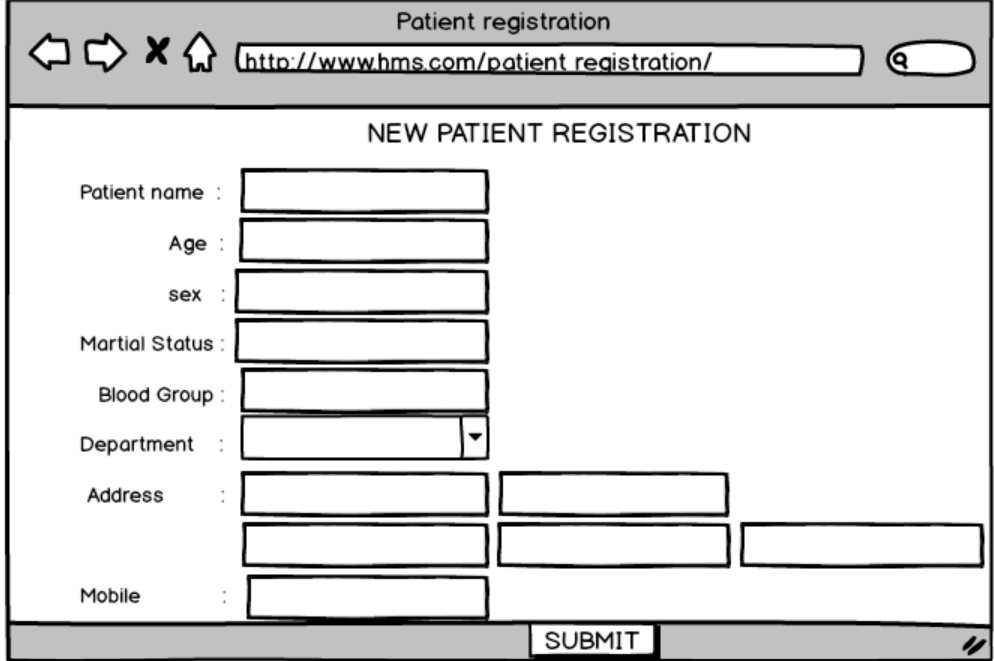
|  |  |
| --- | --- |
| **1. Use case Name** | Patient Admission |
| **2. Use case Description** | This use case describes the process of admitting a patient into the hospital. It covers the collection of patient details, room allocation, and updating the system with the patient’s admission status. The patient’s details are stored in the system and used for further treatment and billing purpose |
| **3. Actors Primary Actors Secondary actors** | **Primary Actors:**   * **Receptionist**: The staff member responsible for admitting the patient into the hospital.   **Secondary Actors:**   * **Patient:** The individual being admitted to the hospital. * **Administrator:** Manages hospital system settings, including room availability and access control. * **Doctor:** May be involved if the patient's admission requires assigning specific treatment or medical attention. * **Billing Department:** May be involved to ensure insurance and payment details are captured for the patient. * **Nurse:** Assists in transferring the patient to the designated room |
| **4. Basic Flow** | **Step 1**: The **Receptionist** logs into the system and selects the **"New Patient Admission"** option.  **Step 2**: The system prompts the receptionist to enter the patient’s **personal details** (e.g., name, date of birth, address, phone number, etc.).  **Step 3**: The system checks for any **existing patient records** using the **patient's name or ID**.  **Step 4**: If the patient is **new**, the receptionist proceeds to collect additional information, including medical history, allergies, insurance details, and emergency contact information.  **Step 5**: The system presents a list of **available rooms** based on patient needs (e.g., ICU, general ward).  **Step 6**: The **Receptionist** selects a room and assigns it to the patient.  **Step 7**: The system generates a **unique admission ID** for the patient.  **Step 8**: The system assigns the **doctor** and updates the room availability.  **Step 9**: The **Receptionist** confirms the admission and provides the patient with the room and admission details.  **Step 10**: The process ends, and the system logs the patient as **admitted**. |
| **5. ALTERNATE FLOW** | **Alternate Flow 1: No Available Rooms**   * **Step 1a**: If no rooms are available, the system will notify the receptionist that the patient cannot be admitted. * **Step 1b**: The receptionist may place the patient on a waiting list or suggest a different room type (e.g., private instead of shared). * **Step 1c**: Once an available room is found, the receptionist can continue the process from **Step 5**.   **Alternate Flow 2: Patient Already Registered**   * **Step 2a**: If the patient is already registered in the system, the system will automatically populate their **existing details**. * **Step 2b**: The receptionist confirms the details and proceeds with **Step 5**. |
| **6. Exceptional flows** | **6. Exceptional Flows:**  **Exception Flow 1: Missing Information**   * **Step 4a**: If any required fields (e.g., patient name, emergency contact, etc.) are missing or invalid, the system will prompt the receptionist to correct the information. * **Step 4b**: The receptionist will provide the missing information before proceeding to **Step 5**.   **Exception Flow 2: System Downtime**   * **Step 6a**: If the system encounters an error (e.g., room assignment failure), the receptionist will be notified of the issue. * **Step 6b**: The receptionist may choose to save the data temporarily or proceed manually and escalate the issue to the IT team for resolution. * **Step 6c**: The use case will resume once the issue is resolved. |
| **7. Pre- Conditions** | **. Pre-Conditions:**   * The patient must be **registered** in the hospital database (new or returning). * The receptionist is **logged into the hospital management system** with the required permissions. * **Room availability** information is up-to-date in the system. |
| **8. post-condition** | **Post-Conditions:**   * A **unique admission ID** is generated for the patient and stored in the system. * The patient's **room assignment** is updated, and the room is marked as occupied. * The **patient’s details** are stored and accessible for future reference (e.g., doctor visits, billing). * The **patient’s admission status** is recorded and available for further processes (e.g., treatment, billing). |
| **9. Assumptions** | * The system **integrates** with other hospital systems (e.g., medical records, billing). * The receptionist has been properly **trained** to enter patient data and use the system. * The hospital has **sufficient rooms** available for patient admission. * **Internet connectivity** is stable for system access. |
| **10. Constraints** | * **Room capacity** may be limited, preventing the hospital from admitting all patients. * **Data accuracy** depends on the receptionist entering correct and complete patient details. * **System downtime** could delay the admission process. |
| **11. Dependencies** | * The **Room Management System** is integrated with the hospital management system to track room availability. * The **Billing System** needs to be up-to-date with payment and insurance information to handle patient billing after admission. * **Medical Records System** needs to be synchronized with the patient admission process to update patient history. |
| **12. Inputs and Outputs** | * Patient’s personal information (name, date of birth, address, etc.). * Emergency contact and medical history. * Insurance details (if applicable). * Room type preferences.   **Outputs:**   * Patient’s unique **admission ID**. * Room and doctor assignments. * **Admission confirmation** with room number and patient details. * Updated **room availability** in the system. * Patient **admission report** for future processing. |
| **13. Business Rules** | * A **unique admission ID** must be generated for each new patient. * **Room assignment** should follow hospital policies (e.g., ICU for critical patients). * The **patient's medical record** should be linked to their admission record. * If the patient has **insurance**, verify details before finalizing admission. * **Room availability** should be updated in real-time to prevent double-booking. |
| **14.Miscellaneous Information** | Emergency Admission Process: For urgent admissions (e.g., accidents), the system may allow rapid room allocation and minimal data entry, with additional details to be filled in later.  Multi-language Support: The system may support multiple languages based on the hospital’s location to assist international or non-native patients. |

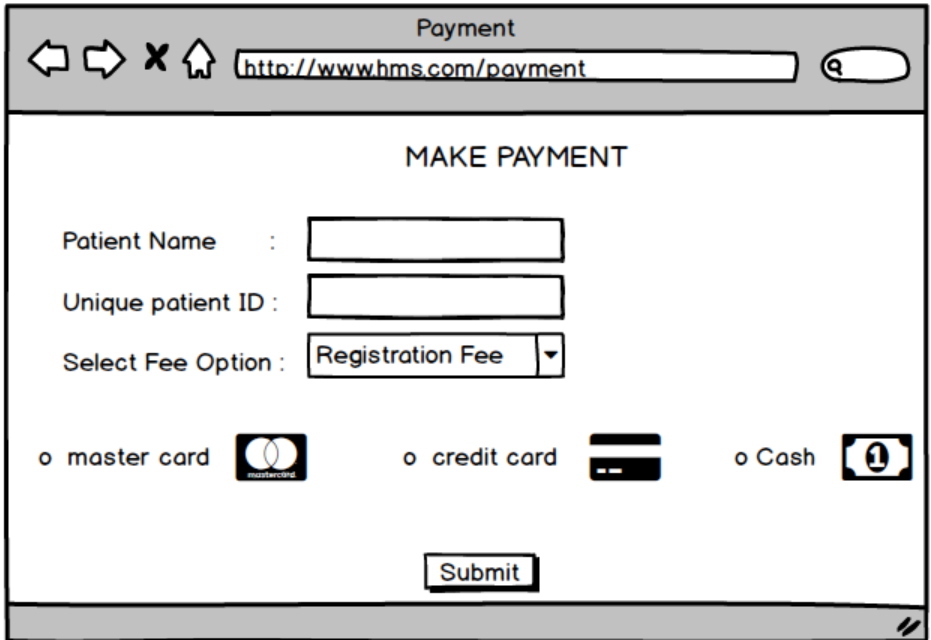
7 Ans .

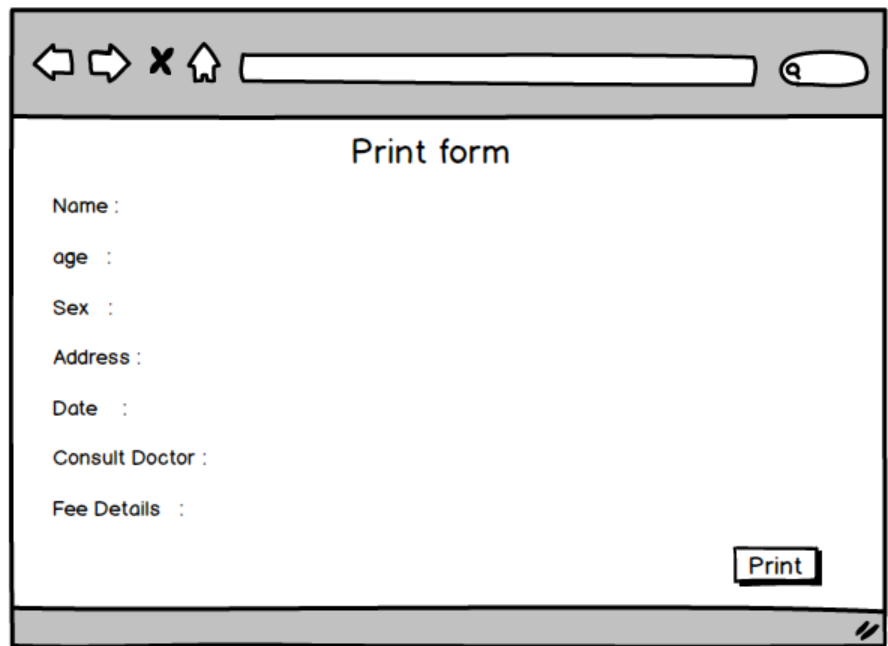


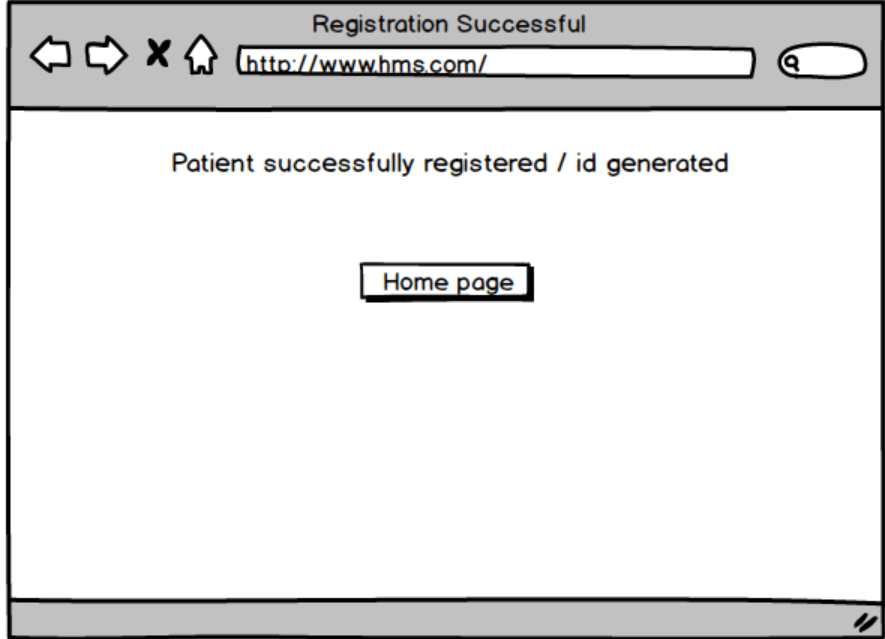












8Ans.

Some commonly used tools for the concepts you mentioned: -

**1. Microsoft Visio:** Microsoft Visio is a popular tool for creating diagrams, flowcharts, and wireframes. It offers a wide range of templates and shapes to create visual representations of processes and systems. It is also commonly used for scenarios such as Process Mapping and Visual Collaboration. The latest version of Visio also has data visualization that allows users to create diagrams from Excel data and also embed Visio diagrams in Power Dashboards

**2. Balsamiq**: Balsamiq is a wireframing tool that allows you to create low-fidelity wireframes quickly. It has a simple and intuitive interface, making it easy to sketch out ideas and concepts3.

**3.Axure RP:** Axure RP is a powerful prototyping and wireframing tool that enables you to create interactive and dynamic prototypes. It offers advanced features for creating complex interactions and user flows.

9Ans. **.**

**1. Requirement Gathering:**

In the **Requirement Gathering** phase, my primary responsibility was to interact with stakeholders including hospital administrators, medical staff, and IT teams to collect detailed information about the hospital's processes and needs. I conducted **interviews**, **workshops**, and **surveys** to understand the requirements for patient management, billing, appointments, and medical records. I ensured that all functional and non-functional requirements were captured, documented, and clarified, helping the team understand the specific needs of various departments. By collaborating closely with end-users, I gathered both **high-level requirements** and **detailed functional specifications**, ensuring alignment with the hospital’s objectives.

**2. Requirement Analysis:**

During the **Requirement Analysis** phase, I worked on analyzing the gathered data to identify and document key features of the **Hospital Management System**. I created **use cases**, **user stories**, and **process flow diagrams** to ensure that the requirements were clearly understood and aligned with business objectives. I also worked on identifying **gaps** in the initial requirements and addressed any discrepancies between stakeholder expectations and technical feasibility. This phase also involved prioritizing requirements based on urgency, business value, and resource availability, helping stakeholders make informed decisions about the project’s scope.

**3. Design:**

In the **Design** phase, I collaborated with UX/UI designers and developers to translate the business requirements into system design specifications. I created **wireframes** and **prototypes** using tools like **Axure** and **Visio**, which helped visualize key system functionalities, including patient registration, appointment scheduling, and billing processes. I participated in design review sessions with stakeholders to ensure the designs were intuitive and met the end-user needs. I also supported the creation of **technical specifications** that outlined the architecture, database, and integration needs, ensuring that the system would be scalable and secure.

**4. Development:**

During the **Development** phase, I acted as a liaison between the business stakeholders and the development team. I ensured that the development team had a clear understanding of the requirements and functionality to be implemented. I regularly conducted **status meetings** to track progress, answer any questions, and help resolve any issues or changes that arose during development. I assisted in **refining user stories** and **handling change requests** by ensuring the scope remained aligned with the project goals while addressing any evolving needs of the hospital management

**5. Testing:**

In the **Testing** phase, I collaborated with the QA team to ensure that the **HMS** met the business requirements and was fully functional. I helped design **test cases**, **user acceptance tests (UAT)**, and **functional tests** based on the documented requirements. I coordinated with stakeholders to facilitate **user acceptance testing**, ensuring that the system performed as expected in real-world scenarios. I was also involved in **defect tracking** and **prioritization**, working with the development team to ensure issues were resolved before the system was released. My focus was on ensuring the system was both functional and user-friendly for the hospital staff.

**6. Deployment:**

In the **Deployment** phase, I worked closely with the technical team to ensure a smooth transition from development to production. I assisted in creating detailed **deployment plans**, outlining steps for server configurations, data migration, and system integration. I helped prepare **training materials** and conducted **training sessions** for hospital staff to ensure they could effectively use the system after deployment. I also supported the **post-deployment monitoring** process to address any issues or user concerns promptly, ensuring that the system performed seamlessly in the live environment.