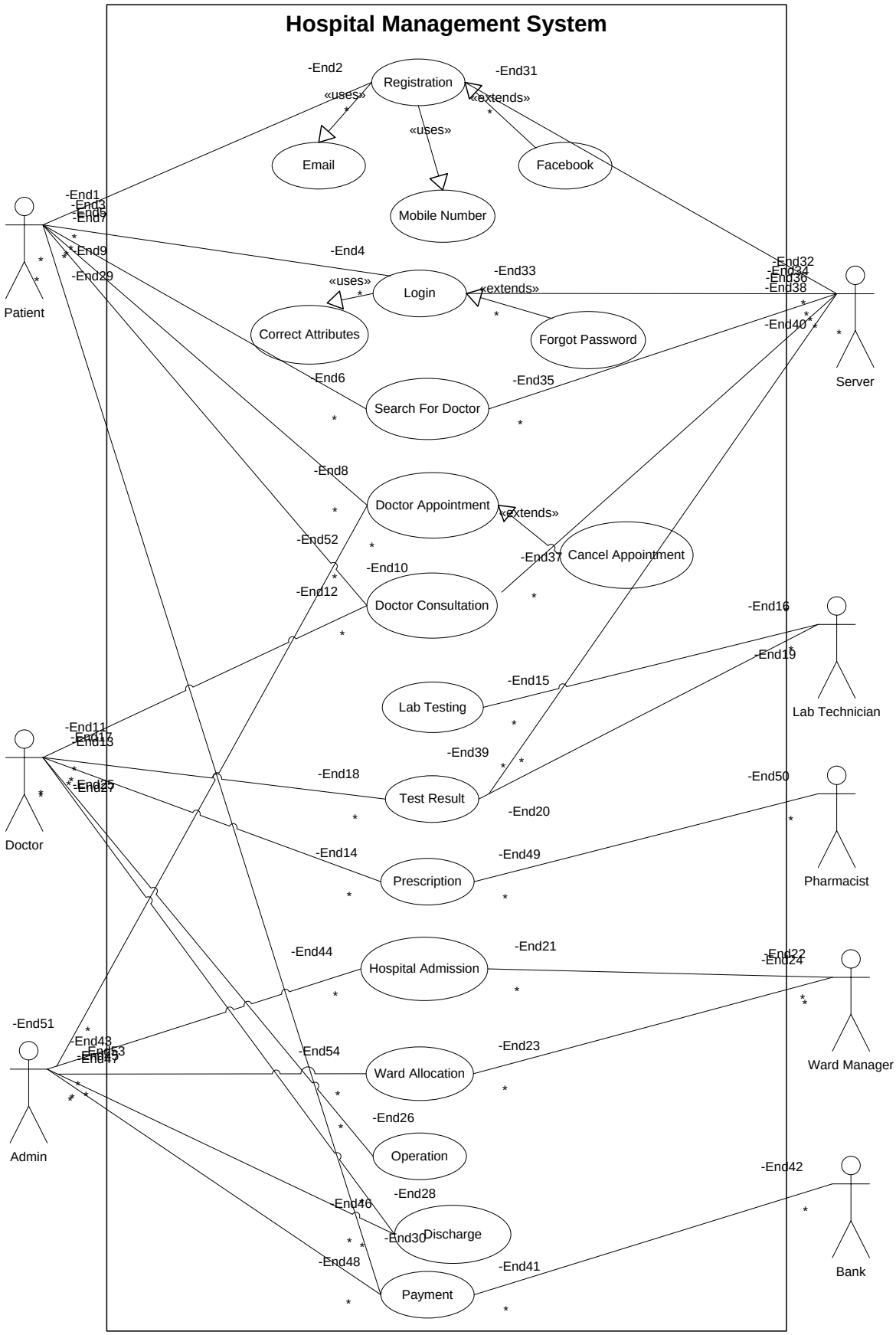
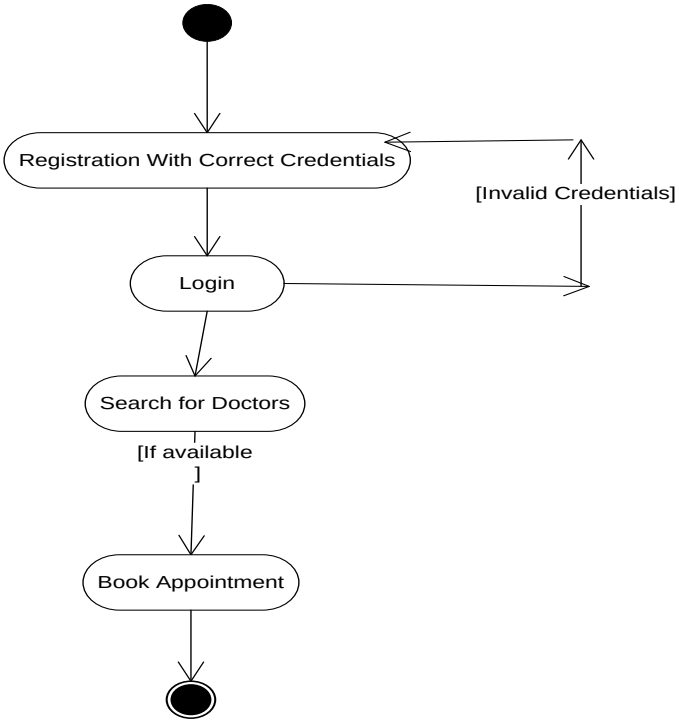
Hospital Management System

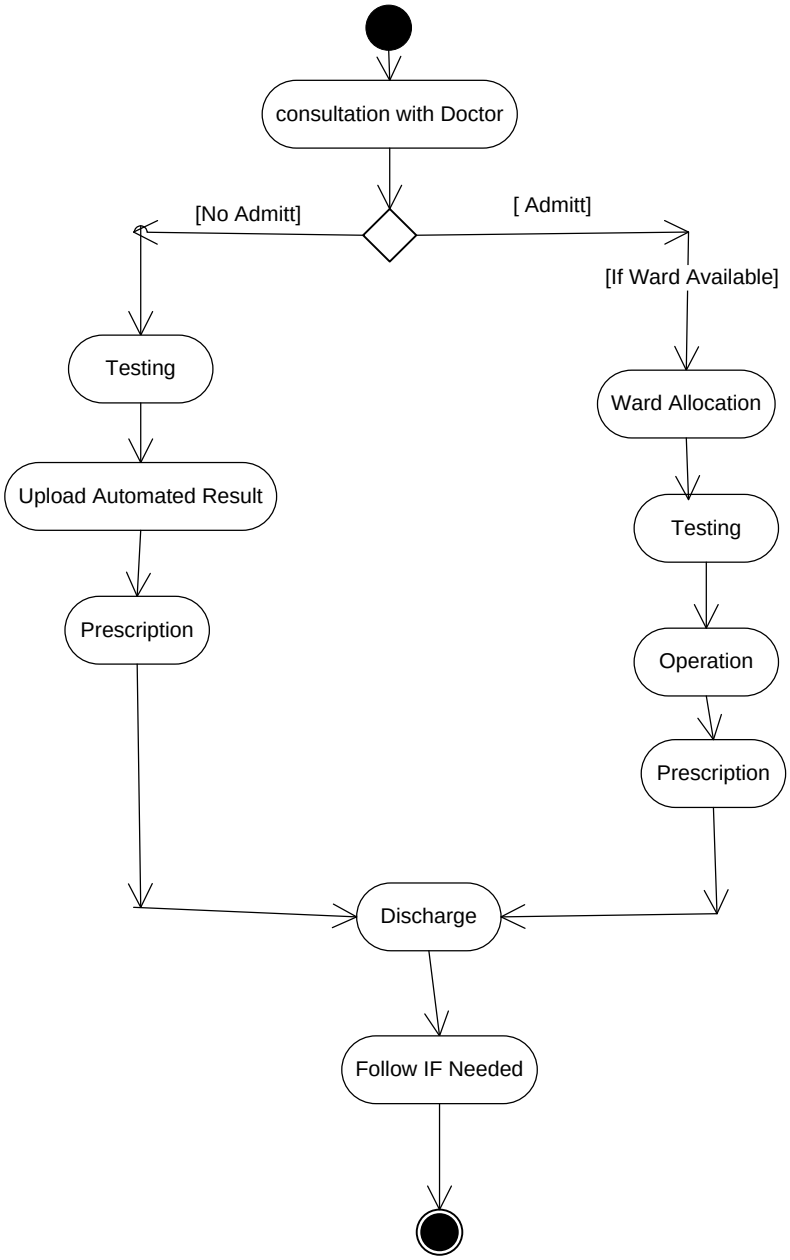
Document- 6: Use Case Diagram, Activity Diagram And Use Case Specification

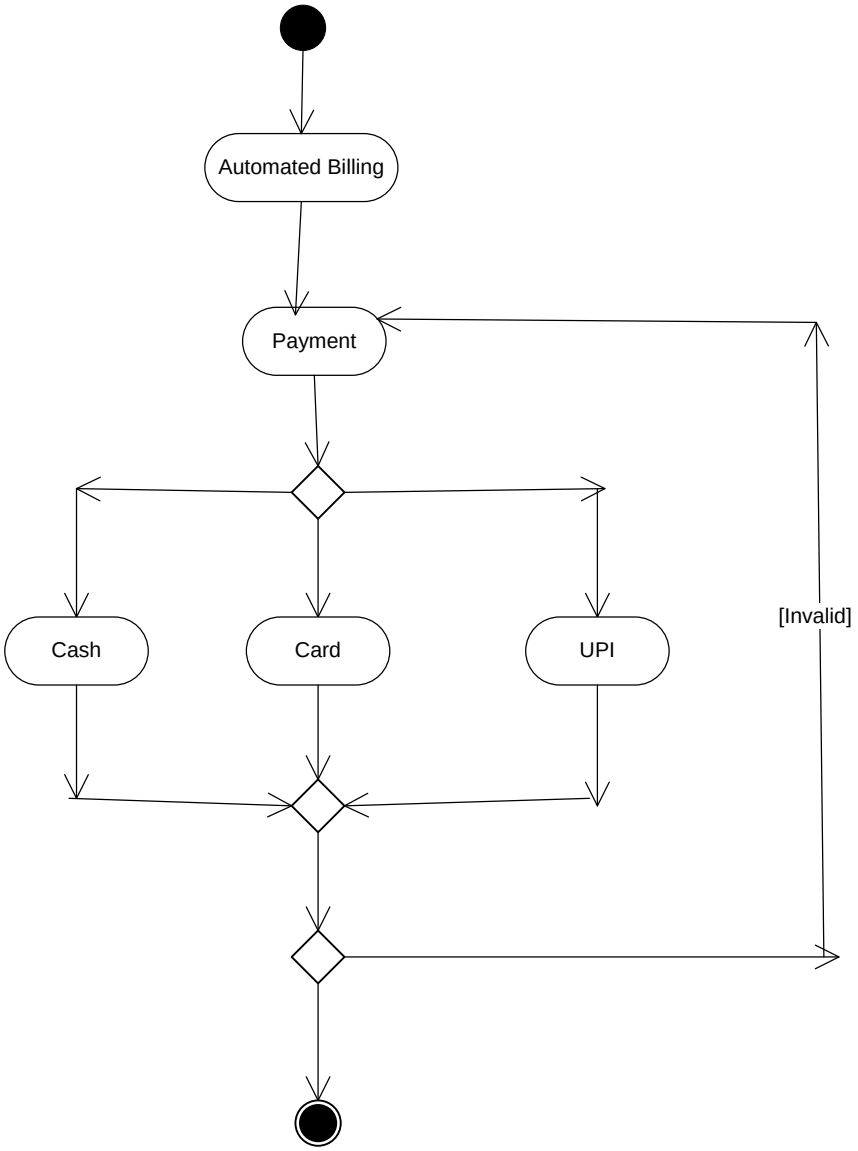
Use Case Diagram:



Activity Diagram:


Registration With Correct Credentials
[Invalid Credentials]
Login
Search for Doctors
[If available
]
Book Appointment


consultation with Doctor
[No Admitt]
[Admitt]
[If Ward Available]
Testing
Upload Automated Result
Prescription
Ward Allocation
Testing
Operation
Prescription
Discharge
Follow IF Needed



Use Case Specifications:

1. User Registration

Use Case ID: UC001

Description: Registers a new patient by collecting personal information and generating a unique patient ID.

Primary Actor: Patient, Doctor, Admin, Lab Tech, Ward Manager, Pharmacist

Secondary Actor: Server

Preconditions: The primary actors must have access to the HMS.

Postconditions: A unique patient ID is generated and stored.

Basic Flow:

1. The patient provides personal details to the receptionist.
2. The receptionist inputs the information into the HMS.
3. The system generates a unique patient ID.
4. The receptionist provides the ID to the patient.

Alternate Flow: If the system is unavailable, the receptionist collects details manually and updates them later.

Exceptional Flow:

Invalid input data triggers an error message, prompting the receptionist to re-enter details.

If the patient or doctor is already registered, notify the user and suggest logging in instead.

Inputs and Outputs:

- **Input:** Patient name, contact details, address, medical history.
- **Output:** Unique patient ID.

Assumptions:

- Patient details provided are accurate.
- The system has sufficient storage capacity.

Constraints:

- All mandatory fields must be filled.
- Only registered staff can register patients.

Dependencies: A functional database for storing patient data

2. User Login

Use Case ID: UC002

Description: Enables authorized users to access the HMS platform using accurate credentials.

Primary Actor: Patient, Doctor, Admin, Lab Tech, Ward Manager, Pharmacist

Secondary Actor: Server

Preconditions: User must be registered in the system.

Postconditions: The user is authenticated and granted access to the system.

Basic Flow:

1. The user enters their username and password.
2. The system validates the credentials.
3. The user is logged in.

Alternate Flow: If credentials are invalid, an error message is displayed, and the user retries.

Exceptional Flow:

- Invalid Credentials: Display an error message after 3 incorrect attempts and temporarily lock the account.
- Account Not Found: Prompt the user to register if the credentials do not match any existing account.
- System Timeout: Notify the user if the authentication server fails and suggest trying again later.

Inputs and Outputs:

- **Input:** Username, password.
- **Output:** Access to the system.

Assumptions:

- All users have valid credentials.
- The login server is functional.

Constraints: Passwords must meet the hospital's security standards.

Dependencies: User registration module must be active.

3. <u>Doctor Appointment Scheduling</u>

Use Case ID: UC003

Doctor Appointment Scheduling

Description: Schedules an appointment for a patient with a doctor.

Primary Actor: Patient, Receptionist

Secondary Actor: Doctor, Admin, Server

Preconditions: The patient is registered in the system.

Postconditions: The doctor's schedule is updated with the new appointment.

Basic Flow:

1. The patient requests an appointment with a specific doctor.
2. The receptionist checks the doctor's availability.
3. The appointment is scheduled, and confirmation is provided.

Alternate Flow: If the doctor's schedule is unavailable, the receptionist suggests alternate dates or times.

Exceptional Flow:

- Overlapping Appointments: Prevent the booking and suggest alternative times.
- Doctor Unavailable: Notify the user and provide a list of available doctors or times.
- Invalid Date/Time Input: Validate user input and prompt correction.
- Payment Failure (if prepaid): Notify the user and request retrying the payment process**.**

Inputs and Outputs:

- Input: Patient ID, doctor's name, date/time.
- Output: Appointment confirmation.

Assumptions: The doctor's schedule is up-to-date.

Constraints: Appointments cannot overlap existing bookings**.**

Dependencies: Accurate scheduling data.

4. Doctor Consultation

Use Case ID: UC004

Description: Facilitates patient consultation with the doctor.

Primary Actor: Doctor, Patient

Secondary Actor: Admin, Server

Preconditions: The patient must have a scheduled appointment. Doctor must be available

Postconditions: The consultation details are recorded.

Basic Flow:

1. The patient meets the doctor at the scheduled time.
2. The doctor reviews the patient's history and conducts the consultation.
3. Prescriptions or tests are advised.

Alternate Flow:

If the patient is late, the doctor may reschedule or adjust the order of appointments.

If Doctor is not available, then patient can choose another doctor or reschedule.

Exceptional Flow:

- Patient Not Checked In: Notify the doctor if the patient has not checked in and reschedule if necessary.
- Incomplete Medical History: Alert the doctor if key patient history details are missing.
- System Lag/Failure: Enable manual entry of consultation details for later system update.

Inputs and Outputs:

- **Input:** Patient history, symptoms.
- **Output:** Consultation summary, prescription, or test requests.

Assumptions: The doctor has access to the patient's medical records.

Constraints: Consultation duration is fixed based on hospital policy.

Dependencies: Patient and doctor availability.

5. <u>Lab Testing and Results</u>

Use Case ID: UC005

Description: Manages lab tests prescribed by doctors and delivers results.

Primary Actor: Lab Technician, Patient

Secondary Actor: Doctor, Sever

Preconditions: The doctor must have prescribed the tests.

Postconditions: Test results are updated in the system**.**

Basic Flow:

1. The patient submits the test request to the lab.
2. The lab technician performs the test.
3. The results are entered into the system.

Alternate Flow: If testing equipment is unavailable, the patient is informed about a delay.

Exceptional Flow:

- Sample Collection Error: Alert the lab technician and reschedule sample collection.
- Results Not Available: Notify the patient with a revised timeline and reason.
- Report Download Issue: Provide alternative access methods or offer to send the report via email.

Inputs and Outputs:

- Input: Test prescription, patient ID, sample.
- Output: Test results, notifications.

Assumptions: The lab has the required resources for testing.

Constraints: Test results must be provided within a stipulated timeframe.

Dependencies: Functional laboratory equipment.

6. <u>Prescription Management</u>

Use Case ID: UC006

Description: Generates and stores prescriptions based on consultations.

Primary Actor: Doctor, Patient

Secondary Actor: Pharmacist, Server

Preconditions: The consultation must be completed.

Postconditions: The prescription is saved and provided to the patient.

Basic Flow:

1. The doctor prescribes medications.
2. The prescription is generated and stored in the system.
3. The patient receives a copy of the prescription.

Alternate Flow: In case of unavailability of prescribed medicines, alternatives may be suggested**.**

Exceptional Flow:

- Drug Unavailability: Suggest alternatives or alert the patient to delayed fulfilment.
- Invalid Prescription Format: Notify the issuing doctor to correct the format.
- Authorization Error: Restrict access to prescriptions and alert for security issues.

Inputs and Outputs:

- Input: Diagnosis details, patient details.
- Output: Prescription copy, notifications to pharmacy.

Assumptions: Prescriptions are based on accurate diagnosis.

Constraints: Only authorized doctors can issue prescriptions.

Dependencies: Updated patient medical history.

7. <u>Ward Management</u>

Use Case ID: UC007

Description: Allocates and manages wards for patients requiring admission.

Primary Actor: Ward Manager, Receptionist, Admin

Secondary Actor: Patient

Preconditions: Wards must be available.

Postconditions: The patient is assigned a ward**.**

Basic Flow:

1. The patient is advised for admission.
2. The ward manager checks availability.

3. The ward is allocated to the patient.

Alternate Flow: If no wards are available, the system schedules the patient for the next available slot.

Exceptional Flow:

- No Vacant Beds: Notify the admission desk and provide alternatives (e.g., other wards or facilities).
- Invalid Ward Assignment: Alert the staff and reassign the patient correctly.
- Maintenance Issues: Temporarily disable the affected ward in the system and notify the facilities team.

Inputs and Outputs:

- Input: Patient ID, ward request.
- Output: Ward allocation confirmation.

Assumptions: Ward availability data is up-to-date.

Constraints: Maximum occupancy limits per ward.

Dependencies: Real-time ward management system.

8. <u>Patient Management</u>

Use Case ID: UC008

Description: Tracks and updates patient information and treatment status.

Primary Actor: Patient, Admin, Hospital Staff

Secondary Actor: Server

Preconditions: The patient must be registered in the system. Admin and staff member must have access to platform.

Postconditions: Patient records are updated.

Basic Flow:

1. Staff access patient records.

2. Information is reviewed or updated.
3. Changes are saved in the system.

Alternate Flow: If patient data is outdated, a flag is raised for verification.

Exceptional Flow:

- Duplicate Records: Prevent duplicate entries and suggest merging records if applicable.
- Missing Patient Details: Highlight missing mandatory fields and restrict operations until completion.
- Unauthorized Access: Log and alert security for any unauthorized patient data access attempts.

Inputs and Outputs:

- Input: Patient data, updates.
- Output: Updated patient records.

Assumptions: Staff inputs data accurately.

Constraints: Access restricted to authorized personnel.

Dependencies: Functional database system.

9. Discharge Management

Use Case ID: UC009

Description: Manages the discharge process for patients. Generate unique discharge ID on advice of doctor.

Primary Actor: Doctor, Receptionist, Admin

Secondary Actor: Patient, Server

Preconditions: The doctor must approve discharge.

Postconditions: The patient receives a discharge ticket.

Basic Flow:

1. The doctor evaluates the patient's recovery status.
2. Discharge is approved and communicated to the receptionist.
3. A discharge ticket is generated and provided to the patient.

Alternate Flow: If the patient's condition worsens, discharge is postponed.

Exceptional Flow:

- Pending Bills: Alert the billing desk and notify the patient.
- Unresolved Medical Issues: Notify the doctor for clearance before proceeding.
- Discharge Document Issues: Flag incomplete or incorrect documentation and prompt resolution.

Inputs and Outputs:

- Input: Patient ID, doctor's discharge approval.
- Output: Discharge ticket, final bill summary.

Assumptions:

- Recovery assessments are accurate.
- Billing details are error-free.

Constraints: Discharge approvals must follow hospital protocols.

Dependencies: Medical assessment and billing modules.

10. <u>Payment Management</u>

Use Case ID: UC010

Description: Handles the financial transactions for services rendered. All payment history is managed and stored. Refund process also managed.

Primary Actor: Receptionist, Patient, Admin

Secondary Actor: Bank, Sever

Preconditions: The billing details must be generated. Discharge ID must already be generated.

Postconditions: The payment is recorded in the system.

Basic Flow:

1. The patient receives the final bill.
2. The receptionist verifies the bill and collects payment.
3. A receipt is generated and provided to the patient.

Alternate Flow: If payment cannot be completed immediately, a payment plan is set up. Cash option may be provided.

Exceptional Flow:

- Payment Gateway Failure: Notify the patient and offer alternative payment options.
- Duplicate Payments: Detect and automatically refund duplicates or notify the billing team.
- Invalid Payment Details: Alert the user to correct details or use a different payment method.

Inputs and Outputs:

- Input: Billing details, payment method.
- Output: Receipt, updated payment records.

Assumptions: Payment systems are operational.

Constraints: Only authorized staff can handle payments.

Dependencies: Functional billing system and payment gateway.

11. <u>Patient Surgery/Operation</u>

Use Case ID: UC011

Description: Manages the scheduling, preparation, and execution of surgical procedures.

Primary Actor: Surgeon, Patient

Secondary Actor: Nurse, Anesthesiologist, Ward Manager, Admin, Server

Preconditions: The patient has been assessed and approved for surgery.

Postconditions: The surgery is completed and recovery plan initiated.

Basic Flow:

1. The surgeon schedules the surgery after patient assessment.
2. Necessary preparations (e.g., anesthesia, equipment) are made.
3. Surgery is performed, and the patient is transferred to recovery.

Alternate Flow:

- If unforeseen complications arise, surgery is rescheduled or additional measures are taken.
- If surgeon is not available rescheduling must be done.

Exceptional Flow:

- Pre-surgery Requirements Not Met: Alert the surgical team if prerequisites (e.g., consent forms, tests) are incomplete.
- Emergency Situations: Notify the hospital's emergency response team immediately.
- Equipment Malfunction: Log and escalate to the maintenance team while ensuring alternative arrangements.

Inputs and Outputs:

- Input: Patient medical history, consent forms, surgical plan.
- Output: Surgery report, patient recovery plan.

Assumptions: Required surgical equipment and staff are available.

Constraints: Surgery can only proceed with patient consent.

Dependencies: Functional operating room and supporting medical staff.

12. <u>Patient Profile Management</u>

Use Case ID: UC012

Description: Allows patients to view and update their personal information.

Primary Actor: Patient

Secondary Actor: Doctor, Admin, Server

Preconditions: The patient must be registered and have login credentials**.**

Postconditions: The patient profile is updated in the system.

Basic Flow:

1. The patient logs into the HMS.
2. The patient views and updates their profile information.
3. Changes are saved and confirmed.

Alternate Flow: If invalid inputs are detected, an error message prompts.

Exceptional Flow:

- Data Update Failure: Log the issue and notify the user with an option to retry later.
- Unauthorized Edits: Restrict changes and log attempts for security review.
- Incomplete Profile: Highlight missing mandatory details and request updates.

Inputs and Outputs:

- **Input**: Updated personal information (e.g., address, contact details).
- **Output:** Confirmation of successful update**.**

Assumptions: Patients provide accurate and updated information.

Constraints: Changes to sensitive data (e.g., ID proof) require additional verification.

Dependencies: Functional patient database.

13. <u>Security Management</u>

Use Case ID: UC013

Description: Implements security measures to ensure the integrity and confidentiality of the system.

Primary Actor: IT Administrator

Secondary Actor: System Users (Staff, Patients)

Preconditions: The system must be operational.

Postconditions: Enhanced security protocols are applied.

Basic Flow:

1. The IT Administrator monitors security logs.
2. Periodic security updates and patches are applied.
3. User access logs are reviewed for anomalies.
4. Encryption protocols are verified.

Alternate Flow: In case of a detected breach, access to the system is temporarily restricted, and the issue is investigated.

Exceptional Flow:

- Data Update Failure: Log the issue and notify the user with an option to retry later.
- Unauthorized Edits: Restrict changes and log attempts for security review.
- Incomplete Profile: Highlight missing mandatory details and request updates**.**

Inputs and Outputs:

- Input: Security logs, user activity reports.
- Output: Updated security measures, breach reports.

Assumptions

- Users are aware of the importance of logging out after use.
- The system has mechanisms in place to detect inactive sessions.

Constraints

- The logout process must not exceed 2 seconds for a seamless user experience.
- Audit logs must comply with data protection regulations.
- The session termination process must clear all sensitive data without leaving traces.

Dependencies

- User authentication and session management module.
- Secure connection protocols (e.g., HTTPS).
- Auditing and logging module.

14. Logout

Use Case ID: UC014

Description: Enables authorized users to safely logout from the HMS platform to maintain privacy.

Primary Actor: Patient, Doctor, Admin, Lab Tech, Ward Manager, Pharmacist

Secondary Actor: Server

Preconditions: User must be login into the system.

Postconditions: The user is safely logout from the system.

Basic Flow

1. The user clicks on the "Logout" button.
2. The system displays a confirmation dialog asking if the user wants to log out.
3. The user confirms the logout action.
4. System Response:
    - Ends the user session.
    - Clears sensitive session data from memory.
    - Logs the logout event for auditing purposes.

- Redirects the user to the login screen.

Alternate Flow

1. The system detects a period of inactivity exceeding the configured timeout threshold.
2. The system displays a warning message to the user about the impending logout.
3. If the user does not respond within the warning period, the system:
    - Ends the user session.
    - Logs the auto-logout event for auditing purposes.
    - Redirects the user to the login screen.

Exceptional Flow

Logout Failure Due to Network Issues

The system attempts to end the session but encounters a network issue.

The system displays an error message indicating the logout process could not be completed.

The user is advised to retry or contact the system administrator if the issue persists.

Input

- Logout request initiated by the user or system.
- Confirmation of logout action.

Output

- User session termination.
- Audit log entry of the logout event.
- Redirect to the login screen.

Assumptions

- Users are aware of the importance of logging out after use.
- The system has mechanisms in place to detect inactive sessions.

Constraints

- The logout process must not exceed 2 seconds for a seamless user experience.
- Audit logs must comply with data protection regulations.
- The session termination process must clear all sensitive data without leaving traces.

Dependencies

- User authentication and session management module.
- Secure connection protocols (e.g., HTTPS).
- Auditing and logging module**.**

<u>Document: 7- Screens and Pages:</u>

XYZ Hospital
Life is Precious
English
Sign In
Seach
First Name
Last Name
Address
City
Bokaro
Phone No.
E-mail
Forgot Password ? Help
Register
Login
About Us
Careers
Privacy Policy
Terms and Policy
Contact Us 00123-99999
E-mail ID- xyzhosptals@gmailcom

XYZ Hospital
English
Life is Precious
Seach
Login:
User Name/Emai ID
Pohone Number
Password
OK
Forgot Password ?
Help
About Us
Careers
Privacy Policy
Terms and Policy
Contact Us 00123-99999
E-mail ID- xyzhosptals@gmailcom
Settings

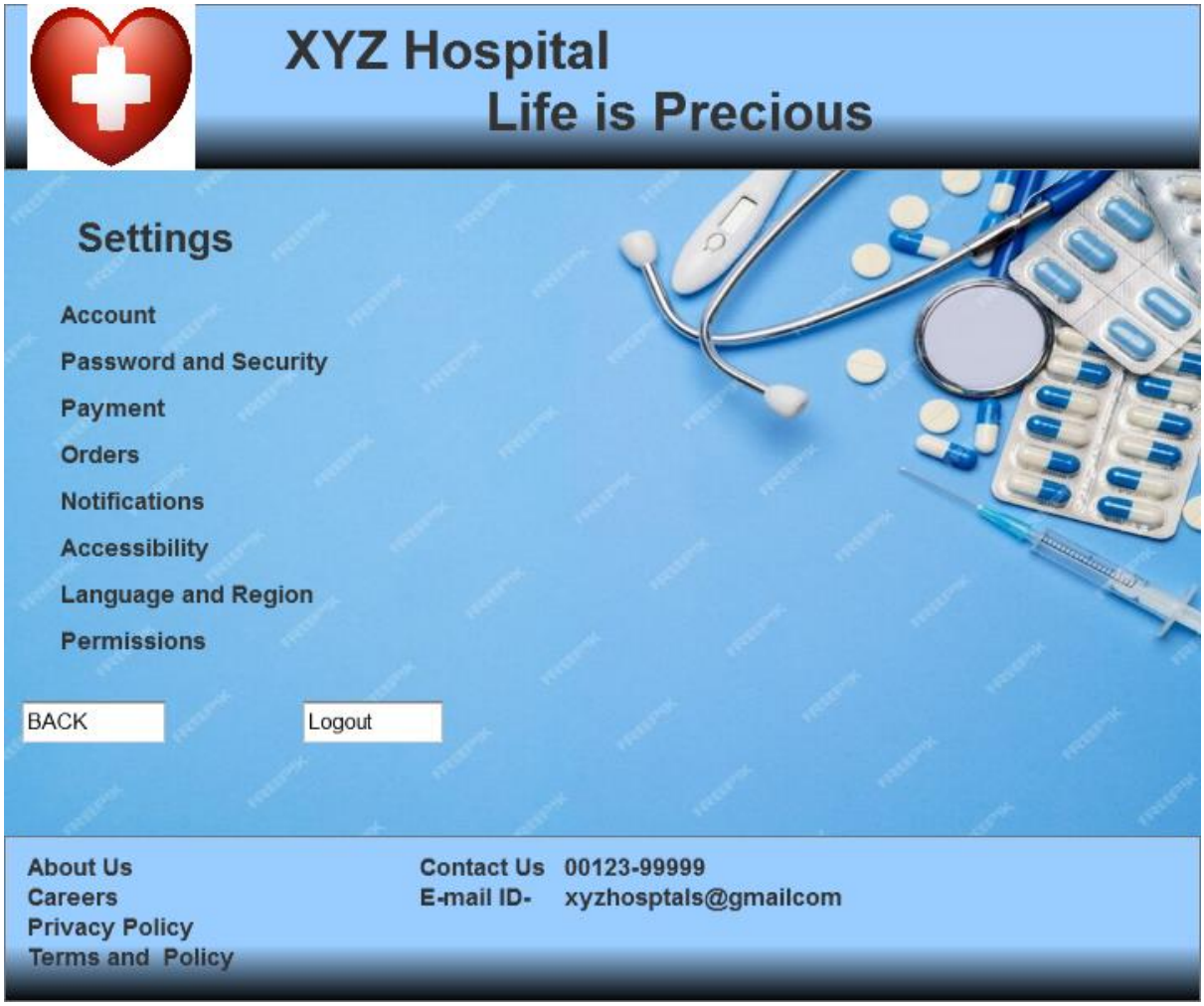
XYZ Hospital
English
Life is Precious
Welcome
Account
Department
OPD
Doctor
Dr. Hemant Singh
Availability
Yes/NO
Lab Result
Medicines
Help?
BACK
OK
About Us
Careers
Privacy Policy
Terms and Policy
Contact Us 00123-99999
E-mail ID- xyzhosptals@gmailcom
Setting

XYZ Hospital
English
Life is Precious
Account
Profile Pic
First Name
Last Name
Address
Phone No.
E-mail
Orders
Reset Password
Help?
BACK
Logout
About Us
Careers
Privacy Policy
Terms and Policy
Contact Us 00123-99999
E-mail ID- xyzhosptals@gmailcom
Settings

XYZ Hospital
English
Life is Precious
Medical Store
Medicines
Availability
Order
Cancel Order
BACK
Logout
About Us
Careers
Privacy Policy
Terms and Policy
Contact Us 00123-99999
E-mail ID- xyzhosptals@gmailcom
Settings

XYZ Hospital
English
Life is Precious
Cart
200.00
1 stripe
Safeximе
150.00
1 Bottle
Bio CoF D
Add More
Sub Total 350.00
Discount 00.00
Total 350.00
ORDER
BACK
About Us
Careers
Privacy Policy
Terms and Policy
Contact Us 00123-99999
E-mail ID- xyzhosptals@gmailcom
Settings

XYZ Hospital
English
Life is Precious
Payment Zone
Debit/Credit Card
UPI
COD
Confirm
Cancel Order
Logout
About Us
Careers
Privacy Policy
Terms and Policy
Contact Us 00123-99999
E-mail ID- xyzhosptals@gmailcom



Document :8- Tools- Visio and Axure

MS VISIO:

Business Analyst Experience with Microsoft Visio

As a Business Analyst (BA), proficiency in Microsoft Visio is critical for creating and managing visual representations of business processes, systems, and workflows. My experience with MS Visio includes the following key areas:

1. **Process Mapping and Flowcharts**

Created detailed process flow diagrams to document current and future-state business processes.

Mapped workflows to identify bottlenecks and inefficiencies in existing operations.

2. **Business Requirements Visualization**

Translated complex requirements into visual diagrams for better stakeholder understanding.

Utilized Visio to create wireframes or prototypes of software interfaces to support UI/UX discussions.

Employed use-case and sequence diagrams to illustrate functional requirements and user scenarios.

3. **Stakeholder Communication**

Delivered high-quality, presentation-ready visuals to facilitate discussions with stakeholders.

Customized diagrams using Visio templates and shapes to suit audience needs, from technical teams to executive leadership.

Annotated and shared Visio documents as part of project documentation and decision-making.

4. **Collaboration and Integration**

Used Visio's integration with Microsoft Office Suite (e.g., embedding Visio diagrams in Word or PowerPoint) for seamless reporting and presentations.

Leveraged Visio's co-authoring features for collaborative updates on shared diagrams.

Exported diagrams in various formats (PDF, PNG) for easy sharing with non-Visio users.

Achievements Using MS Visio

- Improved process clarity and stakeholder alignment by delivering accurate and visually appealing process maps.
- Reduced project turnaround time by creating reusable diagram templates for recurring processes.
- Supported successful system implementations by providing detailed technical diagrams to development teams.

MS AXURE:

My Experience Using Axure

I have worked with Axure to create interactive prototypes and wireframes, which helped in explaining and testing ideas for websites and applications. Here's how I've used it:

1. **Building Prototypes**

I used Axure to create clickable prototypes that showed how an application or website would work.

These prototypes helped in testing ideas before development started.

2. **Designing Wireframes**

I created detailed wireframes to show how screens should look and where different elements like buttons and menus should be placed.

These wireframes were useful for discussing designs with stakeholders and designers.

3. **Explaining Requirements**

Axure helped me turn requirements into visuals so that both technical and non-technical teams could easily understand them.

4. **Testing User Flows**

I used Axure to map out user journeys and test how easy it was for users to complete tasks.

The interactive prototypes made it simple to get feedback from stakeholders and end users.

5. **Collaboration**

I shared Axure files with team members for review and made updates based on their feedback.

The ability to publish prototypes online made it easy to collaborate with remote teams.

Advantages Of MS Axure

- **Interactivity:** Create interactive, clickable prototypes to simulate real-world behaviours.
- **Customization:** Allows for detailed annotations, making it easy to communicate requirements.
- **Iterative Design:** Facilitates rapid iteration based on stakeholder feedback.
- **Collaboration:** Supports team collaboration through shared libraries and cloud sharing.
- **Problem Detection:** Helps identify usability issues early in the development cycle.

Document 9- BA experience My experience as BA in following phases:

1. Requirement gathering:

● To gather requirements, we used MOSCOW technique.

● Client is not available for some period of time during this phase. So as a BA I need to source out point of contacts from his side and get the information ASAP.

● I validate the requirements using FURPS technique

● There are many requirements which are duplicated or repeated. We need to remove them immediately

● Prototyping is used to give more specific requirements.

2. Requirement Analysis:

● We need to draw UML diagrams to visually describe the requirements

● Activity diagrams also used to describe the process flow

● Communicate the diagrams to team. Some team members might not agree with them and might make changes. As a BA we need to consider the points and make modifications

● Prepare BRD and SRS.

3. Design:

● From the use case diagrams, we prepare test cases

● Communicate with client on design and solution documents

● Write negative test cases as well along with positive test cases.

● Do not miss a single test case. It might have huge impact on project development in later stages

● Prepare test data for testing

● Update RTM. This is just as we need to make sure that all the requirements are met

4. Development:

● Organized JAD sessions

● Clarifying queries of tech team during coding

● There might be some team members who doesn't agree with the concept or who doesn’t cooperate during JAD sessions. As a BA I handle the situation gently and had one on one discussion with them. Explained how their actions are going to affect the project. Setup healthy environment within the team.

● Referred diagrams to code the Unit

● Conducted regular meetings with technical team and client which is challenging. Some team members might not be available for the meeting. Recording the session and providing that to missed one and having one to one discussion later with that missed person is all I need to do

5. Testing:

● Prepare test cases from use cases

● Perform high level testing

● Test data is requested by BA from client

● Updated RTM

● Take signoff from client

● Prepare client for UAT

6. Deployment:

● Forwarded RTM to client which should be attached to project closure document

● Coordinates to complete and share end user manuals

● Plans and organizes training sessions

● Make sure all the candidates attend the meeting.