Week 2 Lab Task

Hospital Management App Development: Epics, User Stories, and Security Considerations

The Hospital Management App will include various core functionalities such as patient management, appointments, billing, and compliance. Each of these functionalities will be broken into epics and user stories, ensuring that security is embedded at each stage with "evil user stories" to address potential vulnerabilities.

Epic 1: Patient Management

Goal: To manage patient records, personal information, and medical histories securely and efficiently.

User Stories:

- US 1.1: Add New Patient

- As a hospital administrator, I want to add new patients with personal, medical, and insurance details, so that their information is stored securely and accessed by authorized personnel.

- Acceptance Criteria:

- Admins can input patient data (name, address, DOB, etc.).

- System encrypts sensitive information such as Social Security Numbers and medical history.

- Role-based access ensures only authorized users can access personal details.

- US 1.2: Update Patient Record

- As a doctor or nurse, I want to update a patient’s medical record after a consultation or treatment, so that the information remains current for future visits.

- Acceptance Criteria:

- System allows updates to medical records while maintaining audit trails.

- Only authorized staff can modify records, with timestamps of changes.

- US 1.3: View Patient History

- As a doctor, I want to access a patient’s medical history securely, so that I can provide informed care.

- Acceptance Criteria:

- Medical history is viewable to authorized users only.

- Encryption is applied to data both at rest and in transit.

Evil User Stories:

- US 1.4: Access Unauthorized Patient Data

- As a malicious user, I want to bypass authentication mechanisms and view private patient data, so that I can steal or misuse sensitive information.

- Mitigation:

- Implement multi-factor authentication (MFA) for accessing sensitive data.

- Enforce role-based access control (RBAC).

- Monitor failed login attempts and lock the account after multiple failed attempts.

Epic 2: Appointment Scheduling

Goal: To streamline the process of booking, modifying, and canceling appointments for patients and staff.

User Stories:

- US 2.1: Book an Appointment

- As a patient, I want to book an appointment with a doctor online, so that I can receive timely medical care.

- Acceptance Criteria:

- Patients can select date and time from available slots.

- System sends confirmation via email/SMS.

- Double-booking prevention through real-time calendar sync.

- US 2.2: Manage Doctor Availability

- As a doctor, I want to manage my available slots for appointments, so that I avoid scheduling conflicts.

- Acceptance Criteria:

- Doctors can modify their availability, and patients can only book within those time slots.

- Notifications sent when appointments are canceled or modified.

Evil User Stories:

- US 2.3: Overwhelm the Appointment System

- As a malicious user, I want to flood the system with fake appointment requests, so that legitimate patients cannot book appointments.

- Mitigation:

- Implement CAPTCHA or rate-limiting on appointment booking.

- Introduce validation (email/SMS) to confirm appointment bookings.

Epic 3: Medical Records Management

Goal: To securely store, update, and access patient medical records, lab results, and treatment history.

User Stories:

- US 3.1: Add Medical Records

- As a nurse, I want to add lab results and other medical documents to a patient's record, so that the doctor can review them.

- Acceptance Criteria:

- Medical documents can be uploaded and associated with patient records.

- Access control ensures only authorized personnel can upload or modify records.

- US 3.2: View Treatment History

- As a patient, I want to view my past treatment history securely, so that I am aware of my medical progress.

- Acceptance Criteria:

- Patients can view (but not modify) their treatment history.

- Sensitive data is encrypted, and multi-factor authentication is required for access.

Evil User Stories:

- US 3.3: Modify Medical Records Illegally

- As a malicious user, I want to alter patient medical records to create confusion or harm the patient.

- Mitigation:

- Implement strict role-based access control (RBAC) and audit trails for any modification to medical records.

- Notifications to admin on unusual activity or edits.

Epic 4: Billing and Invoicing

Goal: To manage billing, invoicing, and payments for patient treatments, including insurance claims.

User Stories:

- US 4.1: Generate Patient Invoice

- As an admin, I want to automatically generate an itemized invoice after patient treatment, so that I can manage hospital billing efficiently.

- Acceptance Criteria:

- System generates invoices based on services provided.

- Invoices are stored and accessible for future reference.

- US 4.2: Pay Bills Online

- As a patient, I want to pay my hospital bills online securely, so that I can avoid visiting the hospital in person.

- Acceptance Criteria:

- Secure payment gateway integration.

- Patients can view and pay their bills through encrypted transactions.

Evil User Stories:

- US 4.3: Fraudulent Billing Manipulation

- As a malicious user, I want to modify billing data to overcharge or undercharge patients.

- Mitigation:

- Implement logging of all billing changes and restrict access to invoicing systems to authorized personnel only.

- Conduct regular audits to detect suspicious activities.

Epic 5: Doctor and Staff Management

Goal: To manage staff schedules, payroll, and availability.

User Stories:

- US 5.1: Manage Staff Shifts

- As an admin, I want to schedule shifts for doctors and nurses, so that there is sufficient staff coverage.

- Acceptance Criteria:

- Admin can create and modify staff schedules.

- Staff receives notifications about upcoming shifts.

- US 5.2: Track Staff Availability

- As a hospital admin, I want to track doctor availability and assign shifts efficiently, so that there are no scheduling conflicts.

- Acceptance Criteria:

- Doctors and nurses can update their availability, which automatically syncs with the scheduling system.

Evil User Stories:

- US 5.3: Create Fake Shifts

- As a malicious user, I want to create fake shifts for staff to confuse hospital operations.

- Mitigation:

- Role-based access control ensures only authorized personnel can modify schedules.

- Alert admin if unusual activity is detected in shift management.

Epic 6: Reporting and Analytics

Goal: To generate real-time reports on patient flow, hospital performance, and financial metrics.

User Stories:

- US 6.1: Generate Reports on Patient Flow

- As a hospital admin, I want to generate reports on patient admissions and discharges, so that I can analyze trends and improve hospital efficiency.

- Acceptance Criteria:

- Admin can filter reports by date range, doctor, or department.

- Reports can be exported as PDFs or Excel files.

- US 6.2: Financial Reports

- As a finance officer, I want to generate financial reports to monitor billing, payments, and profitability.

- Acceptance Criteria:

- Financial data is pulled in real-time from the billing system.

- Only authorized users can generate and view financial reports.

Evil User Stories:

- US 6.3: Manipulate Report Data

- As a malicious user, I want to modify reporting data to present false information to hospital management.

- Mitigation:

- Restrict access to report generation to authorized personnel.

- Implement audit trails that track who generates, views, or modifies reports.

Epic 7: Compliance and Security

Goal: Ensure the app complies with healthcare regulations (e.g., HIPAA) and implements best-in-class security protocols.

User Stories:

- US 7.1: Role-Based Access Control (RBAC)

- As an IT admin, I want to enforce role-based access for all users, so that sensitive data is only accessible by authorized personnel.

- Acceptance Criteria:

- The system supports role assignment, with access permissions based on roles.

- Access logs are created for every access or modification event.

- US 7.2: Data Encryption

- As a security officer, I want to ensure that all patient data is encrypted, both at rest and in transit, so that data breaches are minimized.

- Acceptance Criteria:

- AES-256 encryption for data at rest.

- Secure SSL/TLS encryption for data in transit.

Evil User Stories:

- US 7.3: Bypass Authentication

- As a malicious user, I want to bypass authentication and access the system, so that I can steal sensitive data.

- Mitigation:

- Implement MFA for all user logins.

- Monitor login attempts and lock accounts after multiple failed attempts.

- Use intrusion detection systems (IDS) to detect abnormal access patterns.

Conclusion:

By defining the epics and user stories for the Hospital Management App, with security-focused evil user stories, the development team can address functional needs while anticipating potential risks. Embedding security measures at every level helps mitigate vulnerabilities and ensures compliance with healthcare regulations, protecting both patient data and hospital operations.