CSE 545 Software Security

Course Project Requirements

Secure Hospital System

Spring 2022

1. Introduction

The aim of the course project is to develop a skeleton secure hospital system (SHS) with limited functional, performance, and security requirements for secure hospital management, user account management and secure transactions. You can make changes to the requirements only with prior written approval from the professor.

2. Requirements

Multiple users should be able to securely use this system from any place and at any time with the availability of Internet access and web browser.

2.1. User Categories

The users in this system are classified into the following six categories based on their roles:

2.1.1 Internal Users

Internal users can be classified to 4 groups:

1) **Hospital Staff:** Responsible for approving appointment requests, creating patient records, view the patient records, view diagnosis of the patients, view prescriptions, view lab/test reports, create transaction requests and complete the transaction requests on approval from patient.

2) **Doctors:** Responsible for viewing and updating patient records; create, update, and remove diagnosis information of patients; create prescription records (prescribe medicine based on diagnosis); recommend lab tests (the tests must be recommended under diagnosis) and view the lab test reports.

3) **Lab Staff:** Responsible for creating, updating, and deleting lab tests reports; view diagnosis; and approve or deny requests for tests received from patients upon verification of recommendation from

doctor. A lab staff can approve the test request only if the test is recommended by the doctor in the diagnosis or else, he/she must reject the request.

4) **Insurance Staff:** Responsible for viewing and reviewing the insurance claim request, validate the claim request, approve, or deny the claim request, authorize funds dispersal.

5) **Administrators:** Responsible for creating, modifying, viewing, and deleting employee records; authorize or decline transaction requests; create, view, maintain and delete all internal files; access system log files; and ensure smooth functioning of the hospital system.

2.1.2 External Users

6) **Patients:** Individuals can request an appointment with a particular doctor or a general appointment, view their records, view diagnosis, view medical prescriptions, request lab tests, view payments and transactions, and request reports and statements.

2.2. User Account Management

⇒ **Hospital Staff**

- Can approve appointment requests based on doctor availability.
- Create patient records.
- Update personal information of patient on request by patient.
- View patient diagnosis.
- View prescription.
- View lab test reports.
- Create transaction requests.
- Complete transactions upon approval.
- Create receipts and bills.

⇒ **Doctors**

- View and update patient records.
- Create, update, and remove patient diagnosis.
- Create prescriptions.
- Recommend lab tests.
- View lab test reports.

⇒ **Lab Staff**

- Create, update, and delete lab test reports.
- View patient diagnosis.
- Approve or deny lab test requests.

⇒ **Insurance Staff**

- View, review and validate claim requests received from patients.

- Create new insurance policies and insurance records.
- Approve or deny claim request.
- Authorize funds dispersal upon approval.

⇒ Administrators

- Create, view, modify, and delete employee records.
- Create, view, and maintain all internal files.
- Authorize (approve or deny) transaction requests.
- Can access system log files (System log files are only accessible by administrators).

⇒ Patients

- Requests appointment with a particular doctor or a general appointment.
- View their records.
- Update their personal information.
- View diagnosis.
- View prescriptions.
- View lab test reports.
- View his/her medical insurance.
- View payments and transactions.
- Request reports and statements.

2.3. Hospital Functions (Required)

The system should provide at least the following functions:

1) **Appointment and visits:** Must provide an interface to patients to book an appointment for a particular doctor or a general appointment. Once a patient submits an appointment request, the hospital staff can review the request and approve or deny based on the doctor's availability. Once a request is approved, the patient record can be created if not already exist.
2) **Diagnosis:** Once an appointment is approved and doctor is assigned, the doctor can ask the patients for the symptoms, health issues, etc. and accordingly provide the diagnosis, recommend tests if required and prescribe medicines. The doctor can also schedule the next appointment if the patient needs multiple visits for his/her diagnosis.
3) **Lab Tests:** A patient can request a lab test based on the recommendation of the doctor and once the lab staff validates the request and verifies if the test is recommended by the doctor in the diagnosis of the patient, then the lab staff performs the lab test.
4) **Insurance Claim:** A patient can request an insurance claim for the medical expenses incurred to him in the hospital. Once a claim is submitted to the insurance team, the insurance staff can verify and validate the insurance claim, and accordingly either approve or deny the request. If the request is approved, then the insurance staff can authorize the dispersal of funds to the patients' accounts and accordingly the patients can pay their bills.
5) **Help & Support Centre:** Must provide an interface where a patient can update his personal and contact information and request assistance if required.

6) **Chatbot:** The system should provide a chatbot which helps users with their general queries and helps them redirect to appropriate webpages. (e.g., update contact information, schedule appointment, view lab reports, request reports, etc.) **[use Machine Learning].**

2.4. Security Designs (Required)

1) Public Key Certificates: The secure hospital system must use a certificate for the web application (self-signed or authorized by a Certification authority).
2) One Time Password: The system must employ One Time Password (OTP) technique with virtual keyboard feature to validate highly sensitive records or transactions for at least two of the functions in Section 2.3.
3) The SHS should allow multiple users to login simultaneously.
4) The SHS must be available 24/7 for user access.
5) **Prevent malicious login controls. (can use ML)**
6) Session Management
7) Must employ necessary security features to defend against attacks on the SHS system (project will be tested by the students and TA. Preventing DoS or DDoS attacks are out of scope for this project. Students will be penalized to deploy such attacks).
8) Must employ data masking techniques and hashing algorithms to protect user sensitive fields in the database.
9) Must implement a sign-in history function in a manner that a log keeps history of date and time that a customer signs in into their account.
10) All the valid and approved diagnosis reports, approved insurance claim requests, and **approved transactions must be captured in Hyperledger blockchain platform.**

2.5. Performance Requirement (Required)

1) Response time of any event should not exceed the standard response time for hospital applications (3 to 5 seconds).
2) System should withstand user load of approximately 50 users per second and operate in 24/7 environment.

3. Technology & Tools

- Each group can choose one or more of the following languages: Java, Python, PHP.
- Each group can choose either Windows or Linux or Mac for OS.
- Each group can choose either of the two web servers: IIS, Apache.
- Each group can choose either of the following cloud platforms to host their application: AWS, Azure, GCP.

If you want to use a technology/tool different from those above, you need to discuss it with the professor or TA and obtain a written permission. Note that the tool/technology you request to us for your course project must be available to the public for free.

References

References for Work Breakdown Structure

1. WBS: https://www.visual-paradigm.com/guide/project-management/what-is-work-breakdown-structure
2. Gantt chart: https://www.gantt.com

References for Design Document

1. How to write a good software design doc:https://www.freecodecamp.org/news/how-to-write-a-good-software-design-document-66fcf019569c/
2. How to create software design documents:https://www.lucidchart.com/blog/how-to-create-software-design-documents
3. Why you need a Design Document:https://medium.com/@PangaraWorld/step-by-step-guide-on-how-to-build-a-design-document-template-85fba32a54ed
4. Data Flow Diagram: https://www.lucidchart.com/pages/data-flow-diagram
5. Class Diagrams: https://www.lucidchart.com/pages/uml-class-diagram
6. Sequence Diagram: https://www.lucidchart.com/pages/uml-sequence-diagram
7. Use case Diagram: https://www.lucidchart.com/pages/uml-use-case-diagram
8. Misuse case:https://www.microtool.de/en/knowledge-base/what-is-a-misuse-case/

Tools

For creating documents:

1. Google docs: https://docs.google.com/
2. MSOffice

For creating diagrams:

1. Google Drawings: https://docs.google.com/drawings/
2. Lucid Chart: https://www.lucidchart.com/pages/
3. Draw.io: https://www.draw.io