SECURE HOSPITAL SYSTEM

Kumar Swamy Mettela
School of Computing and Augmented Intelligence
Arizona State University
Tempe, United States of America
kmettela@asu.edu

***Abstract*— The aim of the project is to design the skeleton of a secure hospital system using all the required security features to protect the data from malicious attempts from various communities. The main purpose is that the system that it is available for all the internal and external users interacting with the system. This project is more cost-effective and a favorable stand for all the clinical experiences that happen within the secure streamlining methods that are implemented to make the system protective in all terms considering the hazardous attempts to make the system corrupt. The various operations of the different people interacting with the system are either to create, update, modify or delete the records present in the system functionality guideline.**

I. Overview

In the modern era, all the communities or the sectors that are dealing with huge amounts of data are moving towards digitalization of all the fields of data to make the system more robust and make it more appropriate for easy practice in the medical field. The major change that is going on in the field of healthcare is the change in the procedure of entering all the fields of data online. The secure Hospital Management system was a tool for all the organizations out in the market to change the data from paper to online data. After making all the data paperless there was a huge amount of success in the patient care and the management of the hospital staff have a more flexible method to focus on the tasks. The most important feature of the development of the hospital management system maintains the data of the users interacting with the system without any privacy issues and most importantly allocating a certain functionality of the different users based on the work they are inclined in completing. In order to protect the privacy of the data for the different users interacting with the system, there are many security features while implementing the functionality of the hospital management system. The website that is developed for the interaction of the different categories of users is made simple and understandable to any person using the site.

The primary importance of the project is to develop a software application that is web-based to securely connect the users to the platform for various operations. The main implementation is having all the security features that are named by the guideline of the project development to protect the malicious attempts to steal the data or to use the application to cause any damage to the system. There are various users that are interacting with the administration of the hospital to get all the required information. Data manipulation techniques are used to manipulate the information that is stored in the system schema to restore or use the data for a particular task. The main functionality of the system is generating the bills and the transaction for the patients who are the backbone of the system to keep it working in a safe environment. The important feature that is used in the development of the project is to record all the transactions of all the people interacting with the system by implementing the Hyperledger fabric to the integration to make a note of all the movements of data. The session management is implemented in the system to protect the data in the system from any unwanted attempts to the login history which is used as a token for the session to control the flow of the transaction. The scope of all the variable attributes used in the transaction is implemented based on the respective functionality of the scope of the attribute. Most of the security features are implemented based on the core functionality of the users interacting with the system and to protect the system from dangerous attacks.

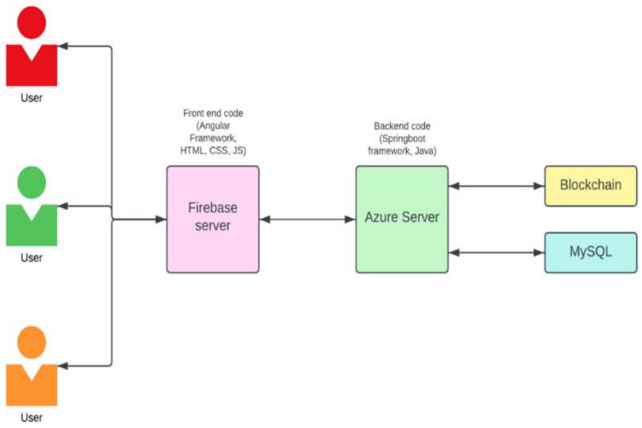
User
User
User
Front end code (Angular Framework, HTML, CSS, JS)
Firebase server
Backend code (Springboot framework, Java)
Azure Server
Blockchain
MySQL

Fig. 1. System Overview

The configuration of the system to de developed is defined by the security and functional features given in the design document by the professor. The system consists of both the internal and the external users interacting with the system to get the desired information from the application or to perform a desired task in the system to implement the task as per the requirement. The six users of the system are patients (external users), hospital staff, doctors, admin, lab staff, and insurance staff (internal users). There are approximately nine security features that are implemented in the system to protect the data from malicious attempts to hack the web application.

II. System design and implementation

The hospital management system is spread to a wide variety of platforms and in a range of industries that provides service to the clients by protecting their data with a lot of security feature. All the people interacting with the system are connected by using web browsers and using an internet connection to interact with the system. All the customer confidentiality is maintained by using the data encryption techniques to protect the data from the malicious attempts of hacking and destroying the data. An HTTPS protocol is used for the data encryption of the public domain website. Another layer of data protection is used by implanting the two-factor authentication to prevent any wrong attempts to enter the system.

The general overview of the system is that whenever a user is trying to interact with the system through the web HTTPS protocol the user is forwarded to the registering page where the individual is allowed to register to the system to use further applications of the system. Once the request is sent from the user interface to the API called function the respective trigger of data information is delivered to the user on the derided functionality. The frontend is developed on the Angular framework which is the eye of the website for the users interacting with the system which is hosted in the Firebase. Once the triggers are raised by the frontend of the application the respective API is called in the backend to deliver the functionality of the task requested for in the operation, the backend application consists of Spring Boot Framework which is operated by Java in the Azure server. The data that is displayed for the task of operation is stored in MySQL database schema and is retrieved from this table which consists of all the data interacting and working for the hospital management system. All the transactions that are taking place in the system are recorded by the blockchain technique to prompt the tasks whenever asked in the form of API calls to the system.

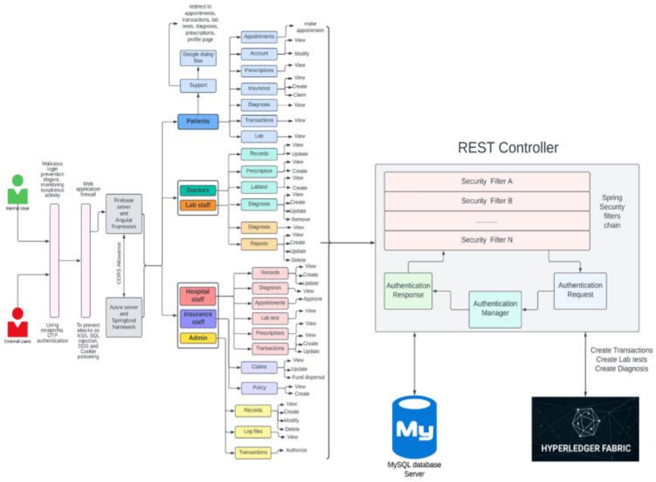

Fig. 2. System Architecture.

Functionalities of the system for any internal or external user. The user attempting to use or login the web application is protected by two layers of protection from anyone hacking the application by using OTP authentication and the firewall implementation, once this step is over the user tries to perform or fetch a certain amount of information from the application, these actions are triggered with the help of API calls that are used to perform all the CRUD operation the tasks by redirecting the API call to the respective user information and these transactions are controlled by a REST controller to filter all the data and send only the data that is asked for the certain API call from the backend applications which contains the request and response from the authentication manager to approve the data to prompt for in the task assigned, all these transactions of data that is flowing is stored in the Hyperledger Fabric to make all the data masked and display whenever asked. This is the general overview of the system that is making things simpler for the hospital management to take a major step in the development of patient care. The languages and the technologies used for the implementation of the web application are listed in the tabular format.

TABLE I. TECHNOLOGIES

Frontend Technologies	Angular 13, HTML 5, CSS 3, BOOTSTRAP, JavaScript
Backend Technologies	Spring Boot, Java, Java Services, REST API
Database	MySQL
IDE	Visual Studio, Intelij, PHP MyAdmin, GIT
Cloud Services	Azure Cloud Services, Firebase Cloud Service
Testing Tools	Postman, Google Developer Tools
Additional Technologies	Hyperledger Fabric Blockchain, Google Dialog Flow, Google SMPT, Server, Google reCAPTCHA Service

The major modules of the system are the patients whose responsibility is to request an appointment from a specific or a specialized doctor, they can read, view their reports and diagnosis, request lab tests, and pay their transactions. The admin staff can have all the logs of the people interacting with the system, they can also approve or deny the request of the patient and maintain all the data of records in the database. The insurance is responsible for approving the insurance claim or denying one. The lab staff is responsible for developing, modifying, and deleting the lab reports of the patient and they also can verify and see the diagnosis of the patient. The doctor's functionality is to maintain the patient records and update the current diagnosis, recommend the lab test, or prescribe the right medication to the patient. The responsibility of the hospital staff is to approve the request of the patients and maintain a record of the patient data on their diagnosis and are responsible for the establishment of the transaction in the patient cycle. All these are developed by a use case diagram to better understand the functionalities of the users interacting with the system.

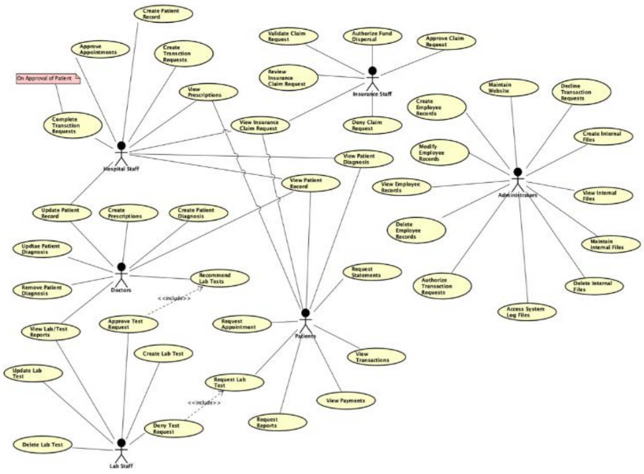

Fig. 3. System Usecase Diagram.

The main concept of taking the subject is to implement the security features to the web application in order to secure the application from hazardous attempts from external sources to override the system or to hack into the system to steal the data. Some of the major implementations of the security features to the system are:

- We use a self-signed public key certificate to make the system secure and validate the authorization of the user interacting with the system.
- One Time password is implemented to make the login history of the user more secure from spilling the important out of the web application into the hands of different users.

- The capacity of the system is enhanced so that multiple users can use the system without any problem.
- The software is made available for 24 hours period on all days to ensure it is reachable to anyone from any corner of the place to perform a task or request the required information.
- The feature of blocking the user from using the system if the user tries to login after several failed attempts and the limit for this function is set at three.
- Session management is implemented to keep track of all the data through the functionalities performed in the system.
- The web-based attacks and the attempt to login into the system are prevented by implementing the option of a firewall and reCAPTCHA.
- Data masking and hashing algorithms are used to encrypt the data that is stored in the database tables.
- The activity of the clients that are using the system is logged as an activity for further purposes.
- The transaction security is maintained with the help of the Hyperledger blockchain function to safeguard the data upon any function call.

III. Results

Considering all the peer evaluations on the project helped us understand where improvements must be made on the functionalities and the security features of the system. There are numerous suggestions on how to improve the web application to make it a better place for the safety of the user data and make it a better place for the interaction of the different users. These results are purely based on the peer perspective of the system design and the interactions that they have added to the base functionalities of the web application. We have tabulated all the results from the peer evaluation and classified them into functional and security vulnerabilities.

TABLE II. Peer Evaluation Results

Vulnerability	Count
Total Vulnerability Obtained From Peers	282
Vulnerability after Merging	61
Invalid Functional Vulnerability Reported	30
Invalid Security Vulnerability Reported	14
Valid Functional Vulnerability Reported	10
Valid Security Vulnerability Reported	7

All the valid vulnerabilities are developed and corrected in the later commits of the web application removing any malfunctioning of the system to the requirement given in the design manual.

IV. My contribution

I have enhanced the implementation of various functionalities of the system be it the functional and the security features. My contributions in the project have been a key insight into how to develop the application from the start by considering all the requirements and implementing various security features like the reCAPTCHA and one-time password features to make it a better place for an individual to use the web application.

- I contributed to designing the overall architecture, general use case, and activity diagrams of the application which is the most important key contribution to gathering all the functional and the security features of the system.
- Designed especially the use case, activity, and login sequence diagrams of the patient functionality in the web application. The use case tells the functional requirement and the various involvements of the patient in the system, the activity diagram tells the flow of the activity of the particular function the patient is corresponding to acting, the login sequence diagram gives the flow of various actions that are taking place during the implementation of the desired functionality.

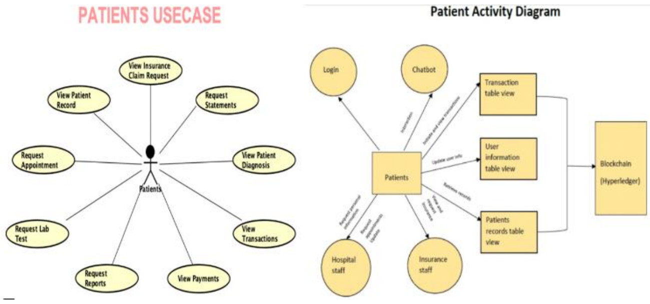

Fig. 4. Usecase and Activity Diagram of Patients.

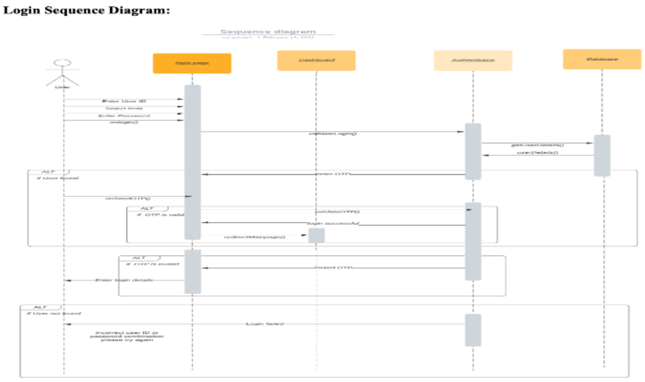

Fig. 5. Login Sequence Diagram.

- Projected a relational database schema for the system which consisted of all the data that the hospital management should contain and nearly 18 tables are created to support the system with the data masking and hashing of the data present in the database tables.

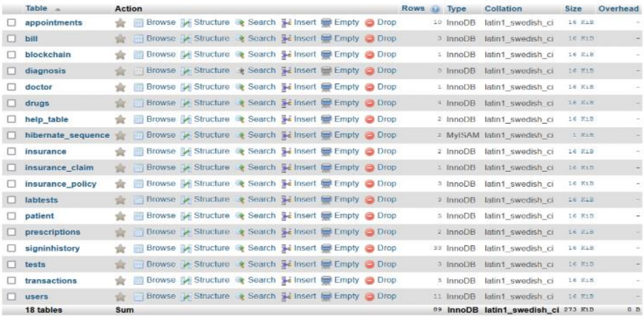

	Table	Action	Rows	Type	Collation	Size	Overhead
□	appointments	Browse Structure Search Insert Empty Drop	10	InnoDB	latin1_swedish_ci	16 KiB	-
□	bill	Browse Structure Search Insert Empty Drop	3	InnoDB	latin1_swedish_ci	16 KiB	-
□	blockchain	Browse Structure Search Insert Empty Drop	1	InnoDB	latin1_swedish_ci	16 KiB	-
□	diagnosis	Browse Structure Search Insert Empty Drop	0	InnoDB	latin1_swedish_ci	16 KiB	-
□	doctor	Browse Structure Search Insert Empty Drop	1	InnoDB	latin1_swedish_ci	16 KiB	-
□	drugs	Browse Structure Search Insert Empty Drop	4	InnoDB	latin1_swedish_ci	16 KiB	-
□	help_table	Browse Structure Search Insert Empty Drop	2	InnoDB	latin1_swedish_ci	16 KiB	-
□	hibernate_sequence	Browse Structure Search Insert Empty Drop	2	MyISAM	latin1_swedish_ci	1 KiB	-
□	insurance	Browse Structure Search Insert Empty Drop	2	InnoDB	latin1_swedish_ci	16 KiB	-
□	insurance_claim	Browse Structure Search Insert Empty Drop	1	InnoDB	latin1_swedish_ci	16 KiB	-
□	insurance_policy	Browse Structure Search Insert Empty Drop	3	InnoDB	latin1_swedish_ci	16 KiB	-
□	labtests	Browse Structure Search Insert Empty Drop	3	InnoDB	latin1_swedish_ci	16 KiB	-
□	patient	Browse Structure Search Insert Empty Drop	5	InnoDB	latin1_swedish_ci	16 KiB	-
□	prescriptions	Browse Structure Search Insert Empty Drop	2	InnoDB	latin1_swedish_ci	16 KiB	-
□	signinhistory	Browse Structure Search Insert Empty Drop	33	InnoDB	latin1_swedish_ci	16 KiB	-
□	tests	Browse Structure Search Insert Empty Drop	3	InnoDB	latin1_swedish_ci	16 KiB	-
□	transactions	Browse Structure Search Insert Empty Drop	3	InnoDB	latin1_swedish_ci	16 KiB	-
□	users	Browse Structure Search Insert Empty Drop	11	InnoDB	latin1_swedish_ci	16 KiB	-
	18 tables	Sum	89	InnoDB	latin1_swedish_ci	273 KiB	0 B

Fig. 6. Database Schema of Tables.

- The API integration for the patient is developed and tested using the postman tool to get the CURD operations running for the task the patient must implement. The API calls implemented are to view the lab test reports, view the medical insurance claim, view payments and transactions of the patient in the system, and request reports and the bill statements are the major API functional call implemented.
- I worked towards deploying the application to the Azure cloud platform to make it available to all the users across the globe and deployed the database schema to the MySQL server with freemysql3 as the host for the web application.
- I have designed the mitigation steps of the web application to reduce the risks that can lead to the collapse of the system, I have strategized the risk management process by having clear testing principles for the website and provided the user guide to all the users interacting with the system.
- I have implemented the most used security feature, which is the validation of the one-time password to the respective email of the user using the Google mail transfer protocol, and enabled the reCAPTCHA feature to prevent malicious login attempts from robots or online hacking methods.

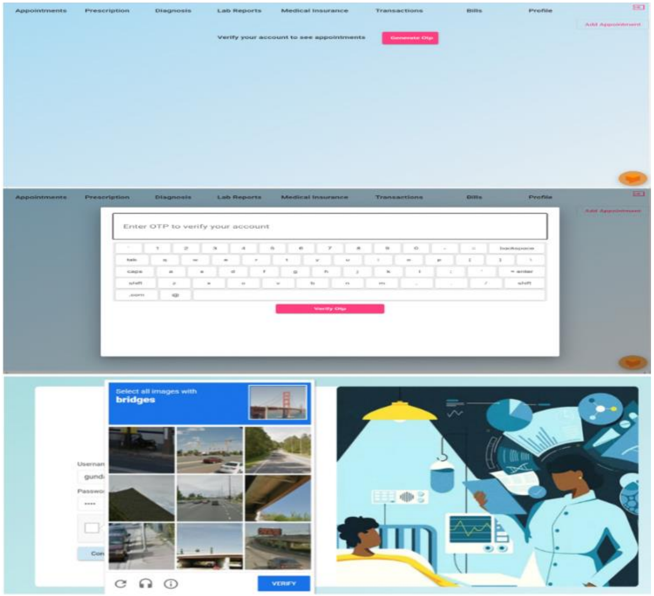
Verify your account to see appointments

Fig. 7. OTP and reCAPTCHA with Virtual Keyboard Validation.

- I have implemented the use chatbot application, where the users interact with the website when needed help for the clarification of the certain method, the chatbot application helps the users to give the required information on what is asked in the system.

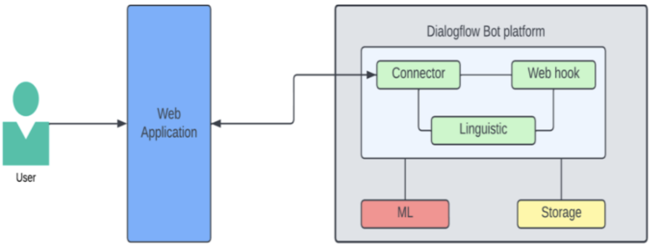

Fig. 8. Chatbot Integration.

- I have performed the functional testing of the web application to check the performance of the API calls over the functions of the users interacting with the system. The API calls include the CRUD application of the data to the existing files in the application.
- I have designed a virtual keyboard feature in the application to handle all the sensitive fields of data that are used while logging into the system or using any field that has data that cannot be shared, by using a virtual keyboard we can eliminate displaying the crucial data to the web offenders or the hackers.
- As Deputy Leader of the team, I oversaw the weekly progress of the team in completing the tasks assigned to each team member by following the agile methodology, I have documented the user guide and the project report for the secure hospital management system.

V. LESSONS LEARNED

The major takeaways from the project development and working in a team environment has a lot of challenges in combining tasks to get the desired results at the end of the deadline. I have understood that the design part is as equal to the code implementation. Taking the right decision to move on with the project requirements is a task that needs quick and correct decisions working in an agile environment. Adapting to the new circumstances every time there is a new feature adding to the requirement of the project made me learn the potential barriers in web development. Understanding the usage of the space in the database schema is curial to the memory consumption and the cost that leads to the development of the project. Coming to the technical aspect the new area that I have learned is the Hyperledger blockchain fabric that collects all the transaction details to make the system more secure and confident of not leaking any information outside the application. The requirement gathering was crucial in implementing the system architecture and the system design.

TABLE III. TEAM MEMBERS

Name	ASU ID
SANDHYA BALU (LEADER)	1223737762
KUMAR SWAMY METTELA (Dy. LEADER)	1223024075
SAI GANESH REDDY GUNDA	1222124462
RAJESH BADAM	1223151904
VIJAY KUMAR PENUBOLU	1223222299
AKHIL AMUDALA	1219419721
YASH DESHPANDE	1222186498
SUPRAJA VEDULLAPALLI	1224065037
SAUMYA AGHERA	1222349921

REFERENCES

[1] CSE 545 Course Project Requirements Document Spring 2022.

[2] Chakraborty, S., Aich, S. and Kim, H.C., 2019, February. A secure healthcare system design framework using blockchain technology. In 2019 21st International Conference on Advanced Communication Technology (ICACT) (pp. 260-264). IEEE.

[3] Pham, H.L., Tran, T.H. and Nakashima, Y., 2018, December. A secure remote healthcare system for hospital using blockchain smart contract. In 2018 IEEE globecom workshops (GC Wkshps) (pp. 1-6). IEEE.

[4] Hyperledger Blockchain reference provided by the professor: https://www.hyperledger.org/use/fabric.

[5] Spring Framework and API Integration https://spring.io/guides/gs/rest-service/