School of Computing and Mathematical Sciences

CO7201 Individual Project

Final Report Template
AN AUTOMATED SYSTEM FOR LOCAL GPS

[SHREEYA DINESH DESAI]

[sdd13@student.le.ac.uk]

[239038636]

Project Supervisor: [DR. YAKUN JU]
Principal Marker: [DR. STANLEY FUNG]

Word Count: []
[27/02/2025]

DECLARATION

All sentences or passages quoted in this report, or computer code of any form whatsoever used and/or submitted at any stages, which are taken from other people's work have been specifically acknowledged by clear citation of the source, specifying author, work, date and page(s). Any part of my own written work, or software coding, which is substantially based upon other people's work, is duly accompanied by clear citation of the source, specifying author, work, date and page(s). I understand that failure to do this amounts to plagiarism and will be considered grounds for failure in this module and the degree examination as a whole.

Name: [Shreeya Dinesh Desai]
Date: [27/04/2025]

Table of Contents

1. Introduction

Currently in the United Kingdom, the National Health Service (NHS) healthcare staff is facing countless problem out of which two primary problems are lack of staff and insufficient funds. According to the (Medical staffing in the NHS, 2025) report, a large percentage of General Practitioners / Doctors are resigning from their job due to lack of work life balance and degrading personal health. Therefore, most of workload of the senior staffs are now handed over to beginner level or early age Practitioners which lack experiences.

In the year 2024, approximately 42.19% of the staff experienced workload-stress. While 30.24% of the staff felt burnt out as an outcome of their work. Additionally, the healthcare workers felt unappreciated and unhappy with their salaries (Medical staffing in the NHS, 2025).

Due to these challenges, there was an impact on the services provided to the patients. For instance, “Unfortunately, we don’t have any appointments available for today” is the most common phrase used by the receptionists due to the shortage of availability of GP (MacConnachie, 2024).The General Practitioner and NHS staff are facing numerous challenges such as shortage of staff, limited availability of appointments, adolescents not opting to study in the healthcare industry, lack of skilled Doctors. Due to these issues, patients have been suffering from adequate treatment and care. (Khan, 2023).

1.1 Aim.

Inorder to cater these problems, this project mainly aims to develop An Automated System for Local GP’s, which would be helpful for both the healthcare staff and Patients. The website would help the staff to set their availability smoothly and cancel their scheduled appointment with an ease. Also allowing Patients to manage their appointments, Pay for their Prescriptions, all these with a user-friendly interface of the applications for the Admin, Staff and Patients.

1.2 Objective.

The objective for this project would be to design a secure and user-friendly web application considering different demography of users, providing short appointment booking forms to patients with read through articles for minor injuries. Additionally, patients should be able to pay for prescriptions from anywhere in the UK.

On the other hand, the General Practitioner will have a simplified dashboard which would help the doctors to set their availability, view the booked appointment, cancel their availability, view the patient previous medical history and prescribe the medicine.

Lastly, the entire application would be deployed on the Microsoft Azure cloud considering the CAP (Consistency, Availability and Partition Tolerance) theorem.

1.3 Requirements.

1.3.1 Essential Requirements.

Registration

The patient will be able to register to the webapp using secure login credentials.

Registration of Doctors and Nurses will be performed by Admin.

Availability

The Doctor and Nurses will be allowed to set their availability two months prior.

Book appointment

According to the patient's requirement, they can book the appointment with the available Doctor.

Provide prescription

The Doctor will be able to access medical history of patients and provide a digital prescription on the web app.

Admin Dashboard

The Admin Dashboard will allow administrators to add, remove healthcare staff and will also allow administrators to book appointments for aged patients, view the list of patients and the staff.

Staff Dashboard

The Dashboard will help staff to set their availability, view booked appointments, provide prescriptions, view patients medical history and send prescription to pharmacy.

Patient Dashboard

The Dashboard will show the Doctor Availability, book appointment, view prescriptions, upload the prior medical history, previous booked appointments records.

1.3.2 Recommended Requirements.

Deployment on the cloud

Articles for minor injuries & awareness

View prescription

Buy and Pay prescriptions

For the prescribed medicine the patient can buy and pay for the prescription either online or offline depending upon the mode of delivery.

1.3.3 Optional Requirements.

Responsive Web Application.

One to one chat

Due shortage of Doctor/Nurse, if in case there's a follow-up required for a specific patient, or a patient requires immediate attention the chat feature can be leveraged.

Video Consultation

1.4 Tools and Technology used.

The below mentioned details are the tool and technologies used in this project.

Component	Name	Summary
Database	SQL and NoSQL	Storing and managing the data either in structured or unstructured format.
Backend	Python, Flask	Develop business logic.
Frontend	React JS, HTML, CSS, JS	Design a dynamic and responsive web application.
API	REST	Used to communicate with the database.
Authentication	JWT	Used for Secure authentication.
Cloud Deployment	AWS / Azure	Hosting the application, considering availability and consistency from CAP principles.
Version control	Git, GitLab	Version controls the project.
IDE	Visual Studio Code	Tool for editing code.
Testing	Manual testing, User Feedback(frontend), Test Cases, Postman (API)	Tests the robustness and security of the application.
Designing	Figma, Sketch (paper & pen), Draw.io	Designed a low-fidelity design, wireframes, ER Diagram, Use case diagram and Class diagram.
Documentation	MS Word	
Operating System	Windows	

1.5 Structure Overview.

Introduction: Provides a brief idea about the project including the aim, objective, requirements and tool and technologies used.
Background Research: This section tells us about the research that has been conducted during these three months of the dissertation. Moreover, this section is divided into two parts Literature review and existing applications.
System architecture and Design: This section is divided into two parts: Backend architecture which would have the designs of ER- diagrams, Class diagram and Use-case diagram. Frontend

architecture: Display the High-Fidelity designs prototypes and the wireframes plus the core functionalities of the project.
Development process: This section will give us an idea about the entire development process from the scratch to the integration.
Testing: The entire system has been tested manually considering different test case scenarios.
Deployment: This section will speak about the detailed deployment process.
Results: Learning outcomes, conclusion and Future scope of this project would be discussed in this project.

2. Background Research

This dissertation project involved ample number of background research with reference to problems faced by the General Practitioner and Patients, how did the WebApp or Mobile app help the patients to monitor their health, how will the cloud technologies be helpful in the healthcare sector, how has been the security handled with regards to the confidential data. Research has been conducted with the help of google scholar, IEEE, articles, PubMed, BBC news (for the most updated news about General Practitioner), Oxford academic and Springer.

2.1 Literature Review

Security has been considered as a major part of any WebApp development, primarily for the healthcare sector where the confidentiality of the data needs to be maintained. Inorder to maintain the confidentiality of the data there are various approaches of developing a secure application or software. According to (Shuaibu & Ruqayyat Ahmad Ibrahim, 2017) the three renowned approaches that is Microsoft Security Development Life Cycle (SDLC), Software Security Touchpoint (SST) and Comprehensive Lightweight Application Security Process (CLASP). Since their approaches are different, but the crucial details would always be the same (Refer Appendix 10.1-a).

Moreover, various types of attacks take place, where one can detect the cross-site scripting (XSS) with the help of security testing methodology approach. Inorder to boost the security with minimal vulnerabilities, threat modelling approach has been used. The authors (Shuaibu & Ruqayyat Ahmad Ibrahim, 2017) have proposed their own methodology that was segregated into two parts: first was the selection and designing the framework for the development, developing and testing a prototype and the second phase was the evaluation. Also, the design of the webApp model has been provided which has helped to determine the steps and the main motive behind those steps (Refer Appendix 10.1-c).

In the previous decades, the manual health records had various cons such as inconsistent data, missing required files, limited storage and misdiagnosed cases. Thanks to the Electronic Healthcare Record system (EHRC) where one could get the patients record fingertip. This technology has helped the doctors to improve the results of any disease accuracy as well as the communication with the patient. But the storage has always been a concern.

Though the technology for storing the data was changed but the author (Nasaruddin & Izzatdin Abdul Aziz, 2018) thought of developing a web-based application where one could provide a feedback based on the service provided by the healthcare provider. The main objective behind this was to improve the facilities and services provided by the Staff. For example, an e-commerce website has various type of products where there's a section of feedback that has been provided by the customers which therefore help the other customer to

decide on that product. Similarly, if a patient provides feedback for that healthcare staff and facilities provided, this will help the other patients decision making. The authors (Nasaruddin & Izzatdin Abdul Aziz, 2018) have conducted research on websites and review systems, including Doctor2U E-commerce, Australian digital My Health Record, Rating, review, chat box, and feedback techniques, as well as gaps discovered in related work on the trust factor, which have given them a better understanding of the methodology used to obtain the feedback. When implementing their ideas, firstly the use case model for the patient (Refer Appendix, 10.1- e) doctor admin (Refer Appendix, 10.1- d) and staff admin (Refer Appendix, 10.1- b) were designed which helped to get an idea about the system. The Graphic User interface was in HTML and CSS format and the data was stored locally in the MySQL. Further, the testing of the application was done and the method used was Unit testing and integration testing.

2.2 Existing Applications.

This section will provide background research of the existing application.

3. System Architecture & Design

3.1 Backend Architecture

3.1.1 Microservice Architecture Overview.

This section would hold the microservice architecture overview. How it has been implemented / used in this project.

3.1.2 Service Description.

This section would provide an in-depth description about the major services what they do.

3.1.3 ER-Diagrams.

This section will display the Entity Relationship diagram of the project.

3.1.4 Class Diagram.

This section will display the Class diagram of the project.

3.1.5 Use- Case Diagram.

This section will display the Use case diagram of the project.

3.2 Frontend Architecture.

3.2.1 UI design (Wireframes and High-Fidelity Prototype).

This section will provide the Wireframes and High-Fidelity Prototype designed for this project

3.2.2 Core Functionalities.

This section will provide brief information about the functionalities.

3.3 Design challenges.

This section will provide the challenges faced during the design process.

4. Development Process.

4.1 Backend Development

4.1.1 Flask API development process.

The API development process would be explained here.

4.1.2 SQL and NO-SQL integration.

The SQL and NoSQL integration would be explained.

4.1.3 API endpoint development.

The API endpoints would be explained here.

4.2 Frontend Development.

4.2.1 Component based React design.

In this section, how the components have been implemented and used in the code will be explained.

4.2.2 Integration with the backend.

In this section, the integration of the frontend code with the backend would be explained.

4.3 Payment Implementation and Challenges Faced.

In this section, the implementation of the payment (Stripe payment) would be explained, and the challenges would be highlighted.

4.4 Challenges Faced during Integration.

In this section the challenges that were faced during integration would be explained.

5. Testing.

5.1 API Testing (Postman).

The testing of the API with the help of postman would be done and screenshot would be shown in the Appendix.

5.2 System Testing.

In this section, the testing of the entire system would be done where the test cases would be written.

5.3 Usability testing.

6. Deployment.

6.1 Architecture.

In this section, the architecture would be explained.

6.2 Azure Services Used.

In this section, the azure services used that would be explained.

6.3 Deployment Process.

In this section, the deployment process would be explained.

6.4 Deployment challenges faced.

In this section, the challenges faced during the deployment would be explained.

7. Results.

7.1 Achievements.

In this section, the achievement would be explained.

7.2 Lesson Learned.

In this section, the lesson learned by me would be explained.

7.3 Limitations

In this section, the limitation would be of the project would be explained.

8. Conclusion.

8.1 Future Scope.

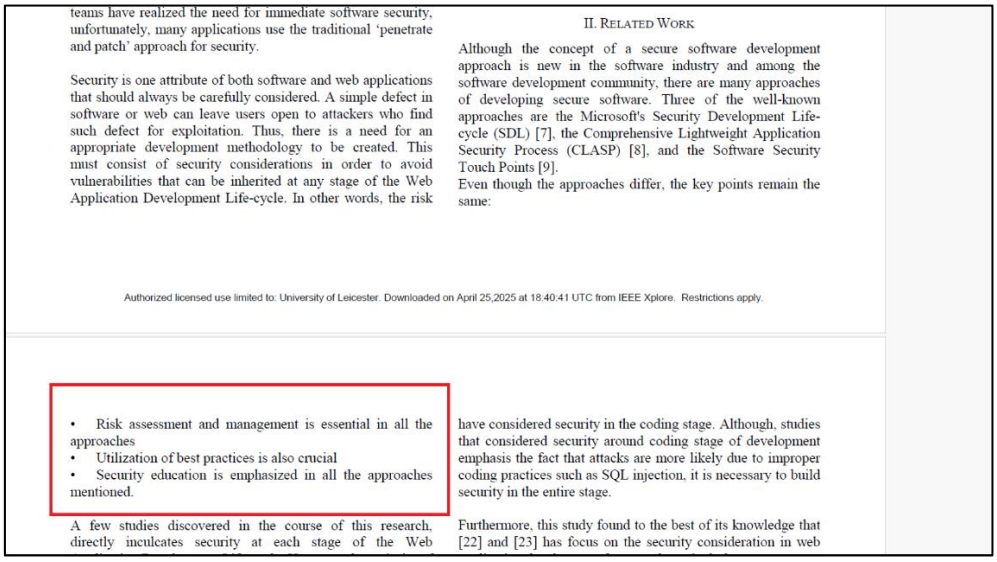
In this section, the future scope that would be explained.

9. References

1. Khan, Z. (2023). The Emerging Challenges and Strengths of the National Health Services: A Physician Perspective. *Cureus*. doi:10.7759/cureus.38617
2. MacConnachie, V. (2024, May 28). *Is it impossible to see a GP?* Retrieved from NHS Confederation: https://www.nhsconfed.org/articles/it-impossible-see-gp
3. *Medical staffing in the NHS.* (2025, March 28). Retrieved from BMA: https://www.bma.org.uk/advice-and-support/nhs-delivery-and-workforce/workforce/medical-staffing-in-the-nhs#:~:text=The%20high%20rate%20of%20doctors,their%20organisation%20values%20their%20work.
4. Nasaruddin, N. S., & Izzatdin Abdul Aziz. (2018). *Web-Based Electronic Healthcare Record System (Ehrs) Based on Feedback.* Langkawi, Malaysia: IEEE. doi: 10.1109/AINS.2018.8631427
5. Shuaibu, M. B., & Ruqayyat Ahmad Ibrahim. (2017). *Web application development model with security concern in the entire life-cycle.* Salmabad, Bahrain: IEEE. doi:10.1109/ICETAS.2017.8277849

10. Appendix.

10.1 Background Research



teams have realized the need for immediate software security, unfortunately, many applications use the traditional 'penetrate and patch' approach for security.

Security is one attribute of both software and web applications that should always be carefully considered. A simple defect in software or web can leave users open to attackers who find such defect for exploitation. Thus, there is a need for an appropriate development methodology to be created. This must consist of security considerations in order to avoid vulnerabilities that can be inherited at any stage of the Web Application Development Life-cycle. In other words, the risk

II. Related Work

Although the concept of a secure software development approach is new in the software industry and among the software development community, there are many approaches of developing secure software. Three of the well-known approaches are the Microsoft's Security Development Life-cycle (SDL) [7], the Comprehensive Lightweight Application Security Process (CLASP) [8], and the Software Security Touch Points [9].
Even though the approaches differ, the key points remain the same:

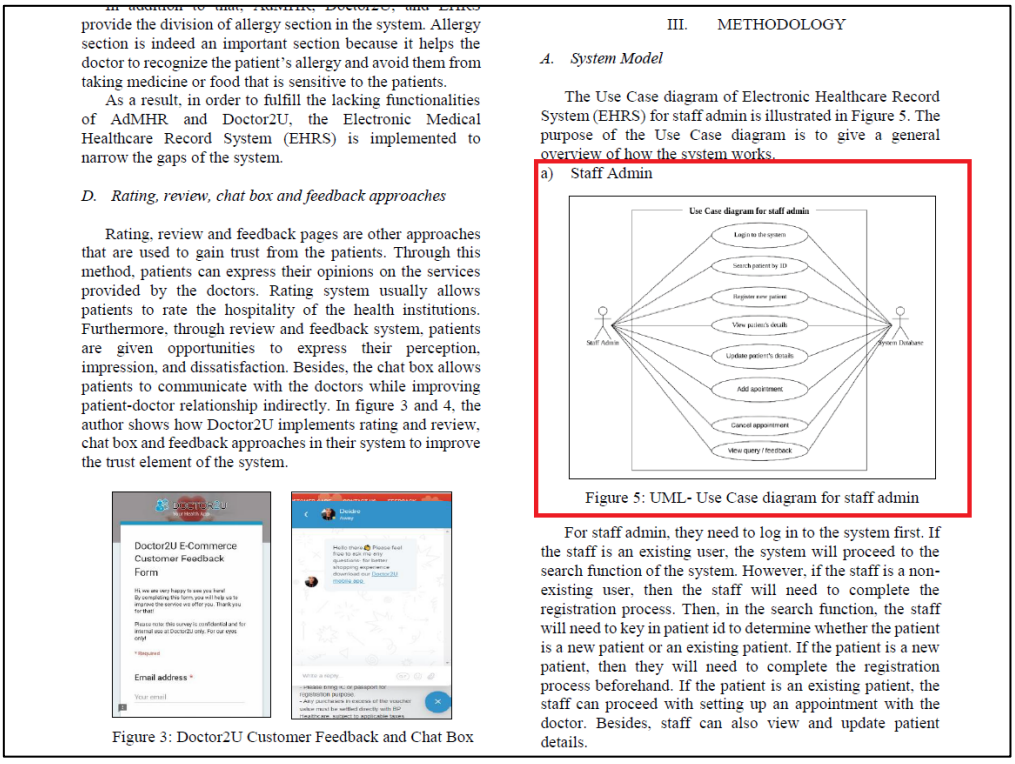
- Risk assessment and management is essential in all the approaches
- Utilization of best practices is also crucial
- Security education is emphasized in all the approaches mentioned.

A few studies discovered in the course of this research, directly inculcates security at each stage of the Web

have considered security in the coding stage. Although, studies that considered security around coding stage of development emphasis the fact that attacks are more likely due to improper coding practices such as SQL injection, it is necessary to build security in the entire stage.

Furthermore, this study found to the best of its knowledge that [22] and [23] has focus on the security consideration in web

Literature Review 10.1-a: Major points that remain the same for the approaches.



provide the division of allergy section in the system. Allergy section is indeed an important section because it helps the doctor to recognize the patient's allergy and avoid them from taking medicine or food that is sensitive to the patients.

As a result, in order to fulfill the lacking functionalities of AdMHR and Doctor2U, the Electronic Medical Healthcare Record System (EHRS) is implemented to narrow the gaps of the system.

D. Rating, review, chat box and feedback approaches

Rating, review and feedback pages are other approaches that are used to gain trust from the patients. Through this method, patients can express their opinions on the services provided by the doctors. Rating system usually allows patients to rate the hospitality of the health institutions. Furthermore, through review and feedback system, patients are given opportunities to express their perception, impression, and dissatisfaction. Besides, the chat box allows patients to communicate with the doctors while improving patient-doctor relationship indirectly. In figure 3 and 4, the author shows how Doctor2U implements rating and review, chat box and feedback approaches in their system to improve the trust element of the system.

Figure 3: Doctor2U Customer Feedback and Chat Box

III. METHODOLOGY

A. System Model

The Use Case diagram of Electronic Healthcare Record System (EHRS) for staff admin is illustrated in Figure 5. The purpose of the Use Case diagram is to give a general overview of how the system works.

a) Staff Admin

Figure 5: UML- Use Case diagram for staff admin

For staff admin, they need to log in to the system first. If the staff is an existing user, the system will proceed to the search function of the system. However, if the staff is a non-existing user, then the staff will need to complete the registration process. Then, in the search function, the staff will need to key in patient id to determine whether the patient is a new patient or an existing patient. If the patient is a new patient, then they will need to complete the registration process beforehand. If the patient is an existing patient, the staff can proceed with setting up an appointment with the doctor. Besides, staff can also view and update patient details.

Literature Review 10.1-b: Use Case Diagram for Staff Admin.

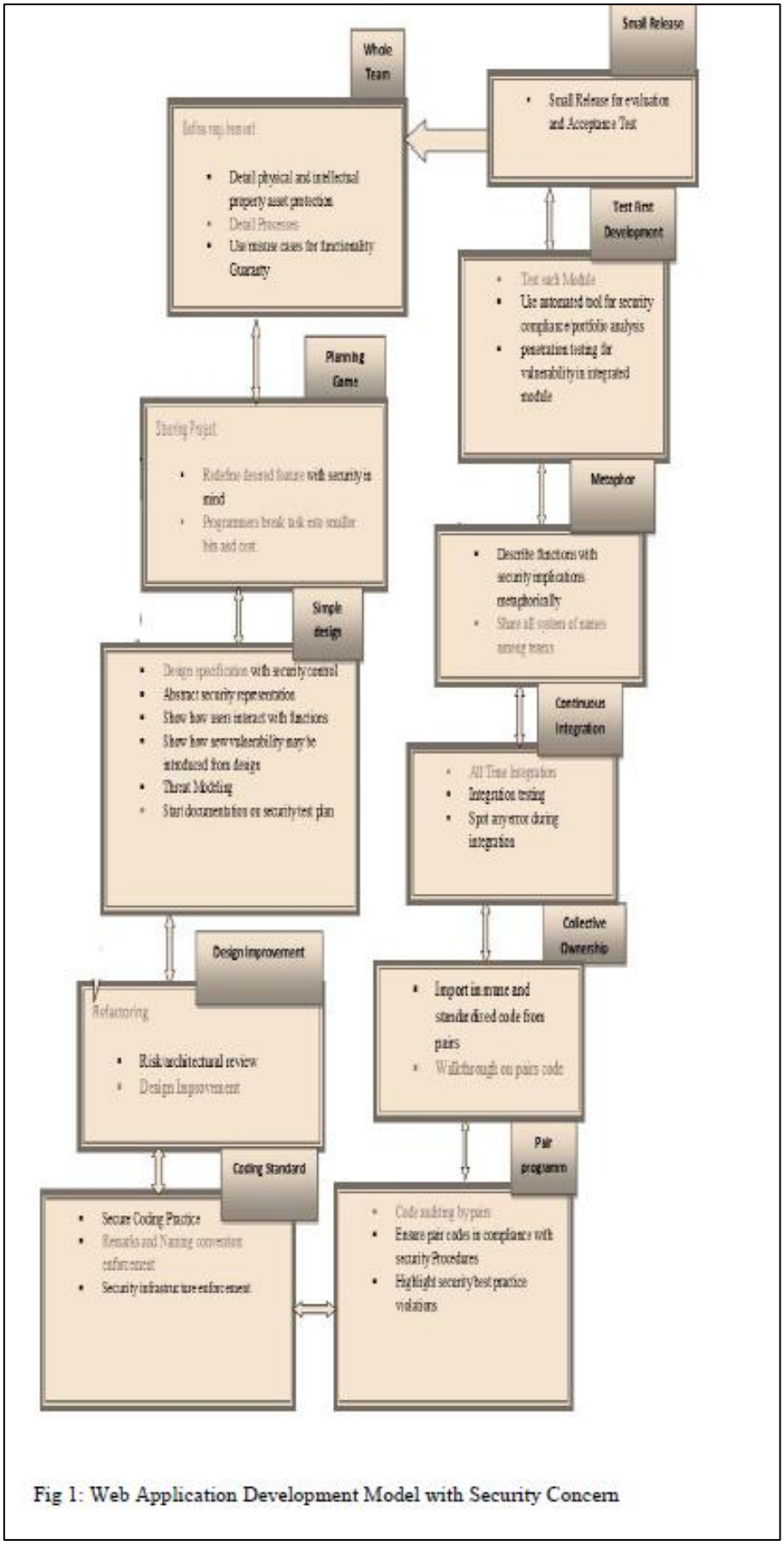


Fig 1: Web Application Development Model with Security Concern

Literature Review 10.1-c: Web application Development Model with security Concerns

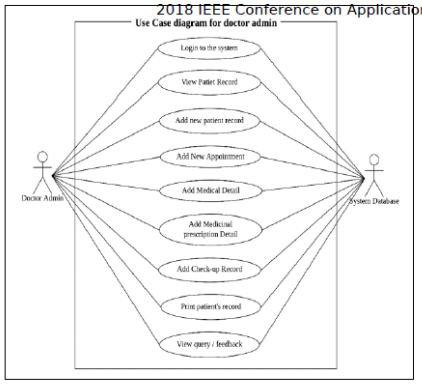


Figure 6: UML- Use Case diagram for doctor admin

The Use Case diagram of Electronic Healthcare Record System (EHRS) for doctor admin is illustrated in Figure 6. For doctor admin, they need log into the system first and register on a new account if they are new users. Doctor admin can have a full access to the patient's record. Thus, doctor

type of treatment provided by the health institution. Furthermore, the registered patient can also request for a documented medical record through the system. After getting treatment from the doctor, the patient can add query or give feedback, comments, and recommendations about the doctors, staffs and the environment of the health institution.

B. System Architecture

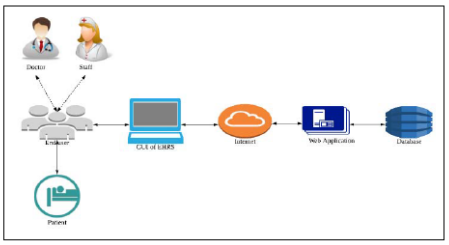


Figure 8: System Architecture of EHRS

Figure 8 shows the architecture design of EHRS where the flow of the system is briefly explained. The system architecture involves the end user, GUI of EHRS, internet,

Literature Review 10.1-d: Use Case Diagram for Doctor Admin.

after viewing the patient's record, the page will redirect to add a new patient record page. In this page, the doctor will input the details of the patient's health condition. Besides, doctor admin can also have the access to view and input Medical detail, Medicinal Prescription detail, and Check-up detail.

Upon completing adding a new record for the patient, the page will redirect to the homepage of the system. The doctor will have an option whether to view feedback or review page or not. If the doctor clicked to view feedback page, then the system will display the feedback page. However, if the doctor does not want to view feedback page, the system will log out and redirect to the homepage.

c) Patient

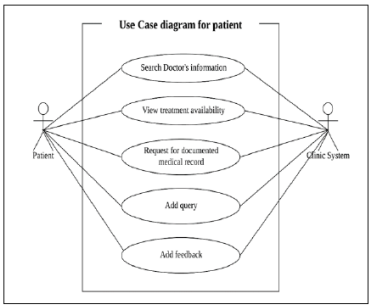


Figure 7: UML- Use Case diagram for patient

Figure 7 shows the Use Case diagram for the patient in EHRS. In this system, the patient can view information of the health institution. Before getting treatment from the doctor, the patient can view doctor's information and the

The end users are the patient, staff admin and doctor admin who will be using the system. Graphical User Interface (GUI) refers to the web browser of EHRS which is in HTML format. The system can only be used if the user has the access to internet connection. Furthermore, to enhance the design of the system, EHRS is supported by a few web applications such as CSS. The data of the system will be stored in a local database which is My SQL. EHRS is a platform where data can be transmitted across many levels of authority. Every medical detail of the patient will be updated in the system. Both patient and medical practitioners have the access to view the medical record.

IV. RESULTS AND DISCUSSIONS

A. Unit Testing

Unit testing is a method in which a module is being tested to figure out any issues occurred during the development of the project. In this project, syntax errors and logical errors are being measured. The table below shows the result of unit testing:

Table 2: The Result of Unit Testing

Unit Function	Flag	Trials	Bugs	Outcome
Login	• Username: must have "@" • Password: password must be matched • Button: print username	5	None	Username is printed
Register	• Username: must have "@" • Password: password must be matched	5	None	Username is printed

Literature Review 10.1-e: Use Case diagram for patient.