ASHESI UNIVERSITY

HEALTH DATA EXCHANGE INFORMATION SYSTEM FOR REGISTERED HEALTH CARE PROVIDERS IN GHANA

APPLIED PROJECT

B.Sc. Computer Science

Benjamin Kusi

2019

ASHESI UNIVERSITY

HEALTH DATA EXCHANGE INFORMATION SYSTEM FOR

REGISTERED HEALTH CARE PROVIDERS IN GHANA

APPLIED PROJECT

Applied project submitted to the Department of Computer Science, Ashesi

University in partial fulfilment of the requirements for the award of Bachelor of Science degree in Computer Science.

Benjamin Kusi

April 2019

Declaration

I hereby declare that this capstone is the result of my own original work and that no part of it has been presented for another degree in this university or elsewhere.

Candidate's Signature:. . . . . .

Candidate's Name:. . . . . .

Date: . . . . . .

I hereby declare that preparation and presentation of this capstone were supervised in accordance with the guidelines on supervision of capstone laid down by Ashesi University.

Supervisor's Signature: . . . . . .

Supervisor's Name:. . . . . ..

Date:. . . . . .

Acknowledgements

I want to express my special thanks to my supervisor Francis Gatsi whose encouragement and academic advice helped me undertake this project on the topic Health Data Exchange Information System For Registered Health Care Providers In Ghana. Mr. Gatsi's effort and dedication to my project helped me to do comprehensive research, and in the process, I learnt new things and I am thankful to him.

Secondly, I would also like to thank Dr. Nathan Amanquah, Head of Department of Engineering who helped me with my project's architectural design.

Abstract

The consequences of a patient's health record not being available at the time of treatment can be very cataclysmic. Accidental human deaths have occurred by the interference of certain medications with some undisclosed health conditions of patients due to the absence of their medical history not being available at the time of treatment.

The purpose of this capstone project is to build and test the effectiveness of web-based inter-operable software which enables the global transfer of the required health information of patients across different hospitals in a secured manner; some of which may be using different health information systems (HIS) running on different storage facilities. Literature review research showed that currently, a system that provides the global exchange of health information across various health care providers.

In this capstone project report, a simple architectural design was used to implement the proposed interoperable health information exchange system. Most health care providers use different health information systems and as such, the results obtained from testing the developed software showed that with a single application programming interface(API), health care providers could request for a patient's health record from another hospital and with different APIs, a patients health record can be sent to the appropriate health care provider.

Contents

1 Chapter 1

1.1 Introduction

The improper administration of medical treatment to patients can be very catastrophic and deadly. People lose their lives because their past medical records are not made available at the time of treatment in a hospital. These inconveniences are very common in most African countries, especially in Ghana. “A recent Johns Hopkins study claims more than 250,000 people in the U.S. die every year from medical errors. Other reports claim the numbers to be as high as 440,000.”[1]. Primarily, the cause of these inconveniences is that technology is underutilized in building software which healthcare providers could use to exchange the needed medical records of patients particularly at the time of medical treatment.

In Ghana, most health care providers use health information systems to manage the records of their patients. Some hospitals use health information systems by a single provider. However, there is no guaranteed mechanism to provide a platform for these health care providers to globally exchange a patient’s medical records in situations where a patient whose health records are with one hospital needs medical attention at another hospital. A research interview carried out revealed that some of the causes of inconveniences such as death that patients suffer are due to the unavailability of their medical history at the time of treatment.

- To begin with, laboratory tests are carried out in situations where a patient’s

medical history is not made available at the time of treatment. Quite often, the lab report does not review some critical conditions that a patient might have. As such, the treatment that the patient receives might interfere with some health conditions that the patient may have and hence, lose his or her life.

- Also, sometimes through questions, medical practitioners can obtain some medical conditions such as allergies that a patient might have. However, most patients are often not knowledgeable of the present state of their medical conditions. As such, when they receive medical treatment, it interferes with their system and hence, they lose their lives as well.

According to the World Health Organization, a "Health technology is the application of organized information and abilities within the frame of devices, medicines, vaccines, procedures, and systems developed to unravel health problems and improve the quality of lives."[2]. Building a system that could allow hospitals to exchange patients information would have been easier if all hospitals were connected on one platform and with a binding agreement, to transfer health information among themselves. In an article titled "Software engineering for connected health", the authors mention that "connected health enables the delivery of process-driven collaborative health control and healthcare practice by individuals, healthcare professionals, patients and carers through the use of technology."[3]

The problem even becomes more complicated because of the differences in

API requirements concerning the health information systems used by various health care providers.

Subsequently, there is a need to build an interoperable system that can permit the transfer of required health records of patients among health care providers with a high level of security.

1.1.1 Problem Background Findings

In Africa specifically in Ghana, health centers have no medium through which they could exchange data concerning a patient's medical history to help facilitate the administration of appropriate medical treatment. Identifying this problem in the year 2018 shows since Ghana gained her independence, hospitals have not been able to have such a facility to enhance their activities.

Most hospitals already have health information systems for managing the records of their patients' medical history. However, there is no centralized medium of exchanging this information globally across hospitals and clinics.

This project, therefore, seeks to design and build a centralized system that would enable the global exchange of data (the medical records of patients) between health care providers to aid in the determination and administration of appropriate medical treatment. Some of the merits of designing and implementing this centralized health information exchange system for its respective stakeholders are as follows:

- It does not require a central database to manage the medical records of patients.
- It provides a high level of security for how a patient's record should be accessed.
- It ensures that data is transferred in real time upon the respective patient's consent. However, in cases of emergency when the said patient is unconscious, based on agreed terms and conditions of the use of the software, the doctor can request for the patient's data upon request for the said patient's medical history from a particular hospital.

1.1.2 Stakeholders

- Registered healthcare practitioners who work in a designated hospital and clinics.
- Patients

1.1.3 System Developer Questions

In anticipation of building an efficient and reliable technology to solve the identified problem stated above, this project seeks to answer some questions necessary to facilitate and enhance the design and building of this technology. Below are some of the design (requirements) identified:

- How could a patient's medical records be kept persistently and securely?

- How could the appropriate data (medical records of a patient) be transmitted securely between hospitals?
- What would be the most appropriate data transmission format concerning how the data would be transmitted from one end and how the other end will receive it?
- If the health data is requested in person by a patient, what would be the appropriate format in which the patient would receive the data?

2 Chapter 2

2.1 Requirement Analysis Gathering

Hospitals keep health records, and these records are not shared due to both technical limitations and security concerns. Hence a sufficient medical history is not used as a basis for administering any medical treatment to most patients.

To help understand the scope of the identified problem above and to provide a suitable solution space; using a questionnaire and with research participant's consent, a research interview was conducted at two hospitals with one being a public hospital and the other a private hospital. The targeted participants were doctors, and the total number of doctors who participated in the research study were six with two doctors from the private hospital and the others from the public hospital.

2.2 Data Collection Method

2.2.1 Onsite Interview

With this type of research data collection method, respondents were interviewed at their various hospitals. The reason for choosing this type of data collection method include its ability to capture rich information to provide comprehensive insights into the data collected.

2.3 Research Method

2.3.1 Qualitative Research Method:

"Qualitative Research is primarily exploratory research. It is used to gain an understanding of underlying reasons, opinions, and motivations."[4]. Due to the type of research questions administered, this research analysis method was used to identify common themes, differences and similarities between participants' responses to provide a comprehensive data interpretation that incorporates various key extracted features of the responses.

2.4 Analysis

In analyzing participants' responses, some of the findings drawn from the qualitative research analysis carried out are:

- There is an urgent need for a system that solves the identified problem.
- Some patients lose their lives because some critical medical conditions that they may have are unknown to the medical practitioner at the time of treatment even though a lab test may have been carried out.
- Often, much time is wasted on getting a patient's medical records before treatment is commenced.

The complete research questions administered can be found at the appendix section.

After a careful review of the results obtained from the analysis of participants' responses and a literature review research, the adaptive, agile software development (ASD) approach was used to develop the software. The reason for choosing agile (ASD) was because it provides for the development of software to be "mission-focused, feature-based, time-boxed, risk-driven, changes-tolerant and an iterative process for adjusting the changing requirements of the software."[5]. As such, this would help the development of the health information exchange system to be robust and to meet the changing needs of its stakeholders.

2.5 Requirement Specification

"Functional requirements are the services that a system should provide, how the system should react to particular inputs, and how the system should behave in a specific situation while non-functional requirements are the constraints on the services offered by the system which applies to the system as a whole."[6]. Based on the interview research literature review conducted to understand the identified problem space and design and robust framework that meets users requirements, the following functional and non-functional requirements were recognized:

2.6 Functional Requirements

A user should be able to:

- log into the system

- as an administrator, register new hospitals and clinics as users of the application
- send a patient's medical records to the requester
- request a patient's medical records from a hospital or clinic
- choose the format in which the data would be transferred
- select the level of permission a requester should have with respect to storing and viewing a patient's medical records

2.7 Non-Functional Requirements

- ***Security:*** the consequence of disclosing a patient's medical records to inappropriate recipients can be very devastating. This could lead to embarrassment, identity theft, loss of employment opportunities and among others. Especially concerning adolescents "confidentiality protection of their medical records is essential because it is consistent with the development of maturity."[7]. As such, the system should ensure that users can view only information that is within their domain. Also, the transfer of a patient's records should be carried out securely and includes the appropriate format the data has to take, standard file transmission protocols and among others. Also, users requesting a patient's records should be authenticated before access is granted.
- ***Availability:*** the system should be up and running at all times.

- ***Reliability:*** whenever users requests for the system to perform a task, the system should be able to do this at all times in the most effective and efficient way.

- ***Safety:*** the system should ensure that the user of the application is not harmed in any way possible.

- ***Cross-platform:*** the system should be able to run across all platforms (mobile phones, tablets and computers) in an efficient way that serves users to a higher degree of their satisfaction.

- ***Bug report:*** the system should be able to generate a timely report on its performance to enhance its maintainability to meet any current changing needs of its users.

Based on the various requirements engineering of the system outline in the above section, the following technologies were used to develop the system:

- Front-End Development – (JavaScript, HTML, Jquery, CSS, Bootstrap)

- Back-End Development – (Java, Mysql)

- Application Framework – (Spring Boot Web Application Framework)

3 Chapter 3

3.1 Architectural Design

3.1.1 Web Model View Controller Design (Web MVC) - Pattern

"The Web MVC design pattern helps to integrate a user interface with the application domain. It seeks to separate the domain logic from the presentation logic of the web application."[8]. It is best suited for web applications and allows for the said application to be structured and maintained easily. It has three components known as the model, view, and controller. According to [9], the author, Nishan Aryal describes these three components as follows:

3.1.2 Model

The model is located just above the storage location (e.g., Database). It contains the business logic and descriptions about the type of information stored in the storage location. The business logic component of the model is where all the functional requirements of the system are specified by the use of an appropriate programming language such as Java, Python, and among others.

3.1.3 Controller

The controller serves as the pathway for rendering information from the model to the view section of the architectural pattern. Based on the route definition specified in the business logic of the model section, the controller uses that specified

route as the pathway to channel the respective accompanying data to the view section.

3.1.4 View

The view section displays information in a more human-readable form. Usually, the technologies used here are mainly HTML, JavaScript and CSS among others. It also ensures that the information displayed is structured and can be easily located by users. Below is an image description of the model view controller.

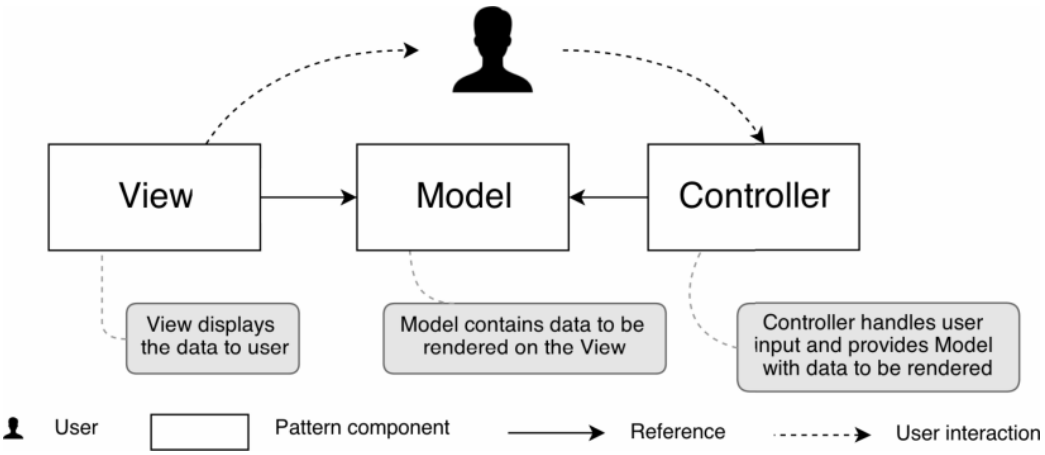

Figure 1: Web MVC Design Pattern
Source: [8]

3.1.5 Reasons for choosing the Web MVC Design Pattern

The following are the major reasons why the Web MVC design pattern was chosen:

3.1.6 Maintainability

It makes it easier to maintain the application regarding the identification of the various components of the application to specify any changing needs that may arise so that the system meets the current requirements of its intended users.

3.1.7 Testability

The MVC pattern provides for testing the various unit components of the application. Hence program bugs are easily detected and fixed for the smooth running of the application.

3.1.8 Relevance to this project

The MVC architectural pattern will help structure the various sections of the web application in a more maintainable way. Hence, all the components of the application will be easily integrated including other existing application components (health information systems) managed by the various stakeholders of this project.

4 Chapter 4

4.1 Implementation

4.1.1 The Architectural Design of the Developed Software

Below is an image of the system's architectural design:

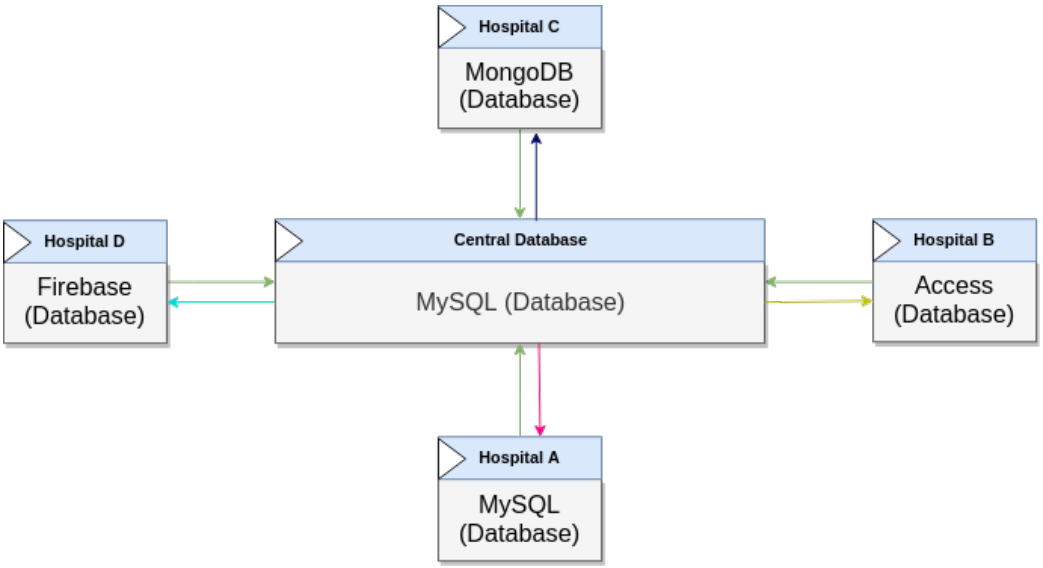

Figure 2: System's Architectural Design

The above figure shows that different hospitals whose health information systems are running on their private databases are allowed to interact (indicated by a green line) with the central database which is managed internally by the application. However, the part of the system that maintains the central database can figure out how to communicate with other health information systems (indicated by different line colors) that are being used by different hospitals. The central

database only manages the sign-up details of hospitals and clinics who are users of the software application.

Below is a description with screenshots of an example use case of the developed application given that a health care provider agrees to use the application:

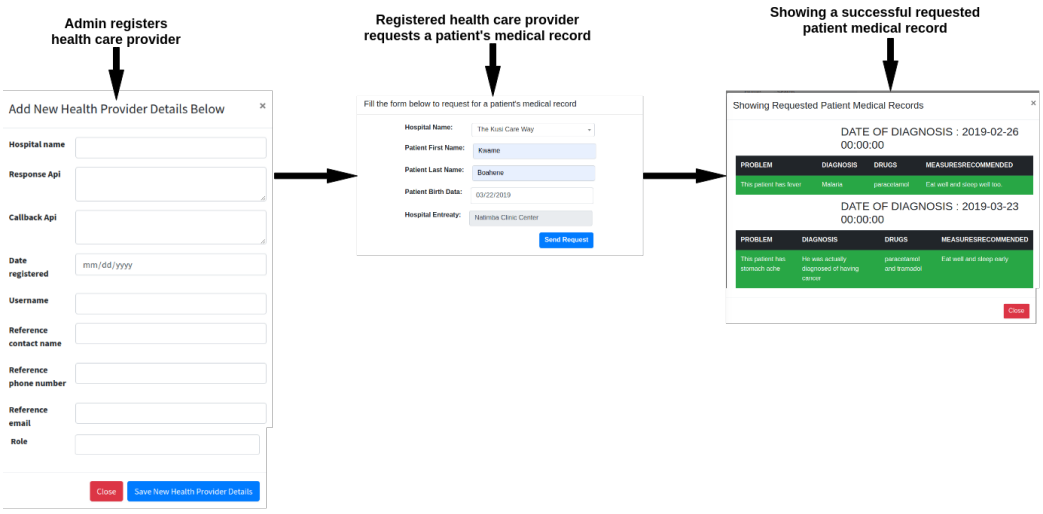

Figure 3: Journey Map Of System Use Case

5 Chapter 5

5.1 Testing and Results Analysis

To ensure that the developed application serves its users most efficiently and reliably, the following types of software tests which were published in an article titled "Types of Software Testing: Different Testing Types with Details".[10] Each type of software testing has its own features, advantages, and disadvantages

5.1.1 Functional and Non-functional Testing

- ***Unit test:*** this test involves testing each of the individual components as a single unit of the system to identify whether their unique requirements are met. The following core units of the system were tested:
  - Binding of API parameters to their respective method calls.
  - Identification and retrieval of requested patient health records from a specified health care provider.
  - Delivery of requested patient health record to the respective information requester.
  - Showing of notification for every user action.

  After this test was carried out, about 70% of the individual units of the system met their different requirements. But after identifying the various

defects, introducing new changes and fixing bugs as well, about a 30% increase was achieved concerning the unit components of the system meeting their requirements.

- ***Incremental integration testing:*** this involves the continuous testing of the performance of the application when new functionalities are added accordingly. This testing was critical to the application because different APIs were, and are continually being built and integrated into the system to meets its rising requirements.

- ***Component testing:*** this involves testing multiple functionalities of the system as a single unit. This test was carried out to identify if there were any defects after connecting the various features of the application. Below are the various components of the system that were tested:

  - Synchronization of different API calls for the retrieval and delivery of requested patient medical records.
  - Updating the number of successful and unsuccessful requests as and when a health care provider initiates the request.
  - Presentation of the appropriate user interface to respective users (health care providers)

  Initially, the results of this test revealed that not all the connected components of the system as a unit code were working together. However, after debugging the identified defects, the results of the new test showed that all

the combined unit functionalities of the system worked together as one unit just as expected.

- ***System testing:*** this involves testing the entire system as a whole that is based on the overall requirement specifications & requirements and covers all the combined parts of the system. Below are some of the core requirements uses as the basis for system testing:
  - Authentication of user credentials when logging in.
  - Retrieval and delivery of requested patient medical records to the respective health care provider.
  - Tracking of successful and unsuccessful request for a patient's medical records per each health care provider who initiated the request.

  After this test was carried out, on a scale of one to ten, the system met about 80% which corresponds to a rating of 8, of all the requirements specified. However, after identifying and debugging the bugs, the system now meets about 95% of all its requirements.

- ***User testing:*** this test involves giving the developed system to sample users to interact with, and provide feedback as to what is working or not working for them and any suggested improvements that can be done to satisfy user requirements. Below are some of the feedback obtained from sample users after this test was carried out:
  - there should be a notification for all major actions that a user takes

- there should be a settings menu to enable user personalize their profiles
- not all links to the home page of the application works
- the time it takes for a patient's medical records are received when requested is very good
- the choice of colour combinations used in the application are very contrasting and that's very good

After filtering and integrating the changes suggested as feedback from the users who tested the software, the application was sent back to them for the second round of testing. Interestingly, a new set of user requirements arose, and this made the testing process very iterative.

However, although the various test results identified above helped to determine whether the system and user requirements of the application were met, the following requirements were not met:

- Concerning when a user forgets their password; for a user to reset their password, a link to the password reset form has to be sent to the user's email address before access to the password reset page is granted for a specified amount of time.
- Users should be able to customize their profiles through a settings menu.

6 Chapter 6

6.1 Conclusion and Recommendations

This project sought to employ appropriate technologies to develop a web application that will help ensure that at any point in time, a patient's medical history is made available at the time of treatment. The developed application works just as expected. The software is designed to intelligently identify a patient's medical records stored in the database of another health provider. It accomplishes this by combining the first name, last name, and date of birth of a patient; which are then routed through an API responsible for fetching the required requested patient medical records. The software was developed by first, using appropriate research data collection methods to understand the needs of stakeholders to enable the development of a proper solution.

However, the developed application met about 95% of its functional and non-functional requirements which are as follows:

- Identification, retrieval and delivery of requested patient health records from a specified health care provider.
- Synchronization of different API calls for the retrieval and delivery of requested patient medical records.
- Tracking of successful and unsuccessful request for a patient's medical

records per each health care provider who initiated the request.

- Presentation of the appropriate user interface to respective users after a successful login authentication.

Although about 95% of the system's functional and non-functional requirements were met, the following are some of the identified limitations of the system:

- When the central database is down, there will be no communication among various APIs to send and retrieve patient medical records.
- When an API for sending or obtaining data is down, no substitute API is used. Hence, data cannot be transmitted nor received by the sending or requesting entity. However, this particular limitation is resolved by first checking if there is not of internet connection. If not, then a check is done with the API holder if any of its parameters have been changed.
- The Internet is required for the transmission of a patient's medical history.

6.1.1 Future Work

Based on all the feedback received from users and inherently from the developed system, through the various testing processes carried out, the following are future works that need to be completed to increase the robustness of the system and its satisfactory user rate:

- For all major user actions taken, there should be an appropriate feedback notification.

- Concerning when a user forgets their password; for a user to reset their password, a link to the password reset form has to be sent to the the user's email address before access to the password reset page is granted for a specified amount of time.
- Users should be able to customize their profiles through a settings menu.

7 Appendix

7.1 Technology Definitions

7.1.1 Spring Boot

Spring Boot is a web application framework that makes it easy to create stand-alone, production-grade Spring based Applications that you can just run.[11] The Spring platform and third-party libraries allows developers to get started with minimum fuss. Below are sample features of the Spring platform.

- Create stand-alone Spring applications
- Embed Tomcat, Jetty or Undertow directly (no need to deploy WAR files)
- Provide opinionated 'starter' dependencies to simplify your build configuration
- Automatically configure Spring and 3rd party libraries whenever possible
- Provide production-ready features such as metrics, health checks and externalized configuration
- Absolutely no code generation and no requirement for XML configuration

7.1.2 jQuery

jQuery is a fast, small, and feature-rich JavaScript library. It makes things like HTML document traversal and manipulation, event handling, animation, and Ajax

much simpler with an easy-to-use API that works across a multitude of browsers. With a combination of versatility and extensibility, jQuery has changed the way that millions of people write JavaScript.[12]

7.1.3 Bootstrap

Bootstrap is an open source toolkit for developing with HTML, CSS, and JS. It allows users to quickly prototype their ideas or build their entire app with Sass variables & mix-ins, responsive grid system, extensive pre-built components, and powerful plugins built on jQuery.[13]

7.2 Project Management Gantt Chart

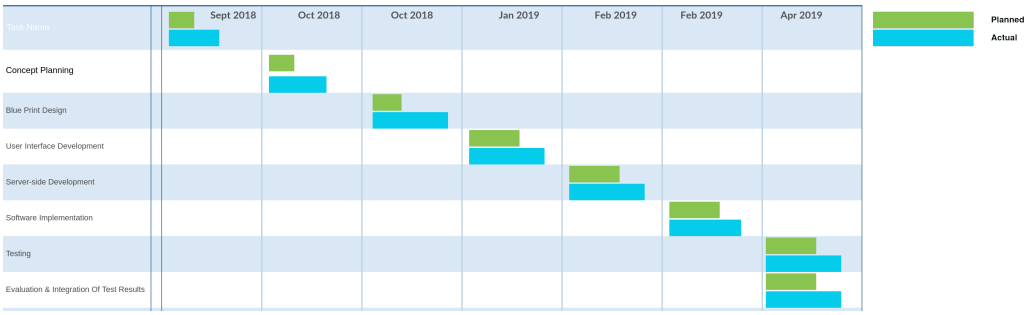

Figure 4: Project Management Gantt Chart

7.3 Research Interview Questions

Using Technology To Facilitate The Global Exchange Of Data (the medical records of a patient) Between Health Care Service Providers

Thank you for taking the time to fill this questionnaire. Please note your feedback would aid in the development of a centralized health information system that facilitate the global exchange of patients' medical records. The goal of building such a system is to ensure that at any point in time when a patient is receiving medical treatment, his or her medical history is made available to facilitate the treatment process just so that we could do away with needless human deaths.

* Required

1. **Which of the following best describes you? If not patient then please skip to question 8.** *
   Mark only one oval.
   - ◯ Patient
   - ◯ A registered doctor working in a registered hospital.
   - ◯ A registered doctor working in a registered clinic.
   - ◯ Other: _____

2. **What is your gender?** *
   Mark only one oval.
   - ◯ Male
   - ◯ Female
   - ◯ Prefer not to say
   - ◯ Other: _____

3. **What is your age range?** *
   Mark only one oval.
   - ◯ 10 - 15
   - ◯ 16 - 25
   - ◯ 26 - 40
   - ◯ 41 - 65
   - ◯ 66 - 80
   - ◯ 81 - 100
   - ◯ Other: _____

4. **Do you have a private doctor that you always consult?** *
   Mark only one oval.
   - ◯ YES
   - ◯ NO
   - ◯ Other: _____

5. **Has the unavailability of your medical history in a hospital at the time of your treatment ever caused any inconveniences for you? Please explain the reason for your answer.** *

6. **How many different hospitals have you visited?** *

7. **Would you like your medical information to be exchanged between hospitals with your consent and with a high level of security access to your medical records at any point in time when receiving medical treatment? Please explain the reason for your answer.** *

8. **On a scale of 1 to 10, how easy is it to help a patient when their medical records are available at the time of treatment?** *
   Mark only one oval.

1	2	3	4	5	6	7	8	9	10
◯	◯	◯	◯	◯	◯	◯	◯	◯	◯

9. **On a scale of 1 to 10, how easy is it to help a patient when their medical records are unavailable at the time of treatment?** *
   Mark only one oval.

1	2	3	4	5	6	7	8	9	10
◯	◯	◯	◯	◯	◯	◯	◯	◯	◯

10. **How much time in hours, does it save when you know the medical history of a patient at the time of medical treatment?** *

11. **Which hospital or clinic do you currently work for?** *

12. **Which hospital or clinic have you worked for?** *

13. **How many years of work experience do you have?** *

14. **Do you have any HIS (Health Information System) for managing the records of your patients?** *

Mark only one oval.

- ◯ Yes
- ◯ No
- ◯ Other: _____

15. **If yes, what HIS are you using?** *

16. **What do you do when a patient visits your hospital, and his or her medical records are unknown to you at the point of treatment?** *

17. **Would you like your patient's information to be exchanged with your consent and with a high level of security access to your medical records? Please explain the reasons for your answer.** *

18. **Give examples about the type of patients you would most need their medical history before proceeding with any medical treatment.** *

References

[1] R. Sipherd. The third-leading cause of death in us most doctors don't want you to know about. [Online]. Available: http://aryalnishan.com.np/asp-net-mvc/asp-net-web-forms-vs-asp-net-mvc/attachment/asp-net-mvc/

[2] W. H. Organization. What is a health technology? [Online]. Available: https://www.who.int/health-technology-assessment/about/healthtechnology/en/

[3] N. Carroll, C. Kuziemsky, and I. Richardson, "Sotware engineering for connected health," *Association for Computing Machinery*, 2017.

[4] R. Methodology. (N/A) Research methodology. [Online]. Available: https://research-methodology.net/research-methods/survey-method/questionnaires-2/

[5] S. K. R. "Bharat Choudhary, *"An approach using agile method for software development"*. Institute of Electrical and Electronics Engineers, 2016.

[6] I. Sommerville", *"Software engineering"*. Boston: Pearson, 2011.

[7] J. S. Santelli, R. Klitzman, and R. Bayer, "New challenges for electronic health records," *The Journal of the American Medical Association 313(1):29-30 · January 2015with1,190 Reads*, 2015.

[8] A. Syromiatnikov and D. Weyns, “A journey through the land of model-view-design patterns,” *Institute of Electrical and Electronics Engineers*, 2014.

[9] N. Aryal. Asp.net mvc. [Online]. Available: http://aryalnishan.com.np/asp-net-mvc/asp-net-web-forms-vs-asp-net-mvc/attachment/asp-net-mvc/

[10] T. of Software Testing: Different Testing Types with Details. Sofware testing help. [Online]. Available: https://www.softwaretestinghelp.com/types-of-software-testing/

[11] Pivotal. Spring boot. [Online]. Available: https://spring.io/projects/spring-boot

[12] jQuery. What is jquery? [Online]. Available: https://jquery.com/

[13] Bootstrap. Getting started with bootstrap. [Online]. Available: https://getbootstrap.com/