UNIVERSITY OF NAIROBI

SCHOOL OF COMPUTING AND INFORMATICS

UNIQUE PATIENT IDENTIFICATION AND MANAGEMENT USING QUICK RESPONSE CODE

BHUNDIA NAVIK JAYANT

P15/33494/2015

SUPERVISOR: DR. ELISHA ODIRA ABADE

A project progress report submitted as partial requirement for the award of Degree in Computer Science of the University of Nairobi, June 2019.

DECLARATION

I hereby declare that this project work is a record of an original work done by me. I affirm that this project has not been presented for any other University award.

NAME: Bhundia Navik Jayant.

REGISTRATION NUMBER: P15/33494/2015.

SIGNATURE: _____

DATE: _____

This project has been submitted as partial fulfilment of the requirements of Bachelor of Science in Computer Science of the University of Nairobi and has been done with the guidance of my supervisor.

NAME: Dr Elisha Odira Abade.

SIGNATURE: _____

DATE: _____

DEDICATION

To my parents for giving me the opportunity to be here, to Doctors who work tirelessly to save lives and to all the patients who wait in long queues to get treatment.

ACKNOWLEDGEMENT

I am grateful to God for the good health and strength he grants me to learn, for providing priceless people who without their effort and collaboration, this project would not have been a success.

I personally would like to thank the following people for their contribution to my inspiration, knowledge and other help in working through this project.

1. My parent to make me believe that I could do it.
2. My supervisor Dr Abade for his support and guidance even where I thought I could not make it.
3. Felix Ongati, who supported me and contributed a lot to the successful progress of this project.
4. Urvi Davadra, a professional medic who helped me with all necessary information I required for the project.

ABSTRACT

The information a physician needs to optimize treatment of a patient is most of the times unavailable, yet health informatics industry, through Electronic Medical Records (EMRs) and other automated clinical systems accumulates large volumes of data that is just specific to a given health service provider. Most of the health service providers have their own EMR hence neither of them has fully updated patient record that can assist the physician in decision making.

This project combines knowledge from Distributed databases and Networking, to come up with a web-based application, that holds a fully updated patient record, despite the fact, the health service provider visited by the patient, as long as it's using the web-based application. This will provide a clear and useful insight to health care worker hence helping them to make better and accurate decisions.

Table of Contents

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1: INTRODUCTION

1.1 Background

Kenya's population as at 2017 was estimated to be around 49,167,383 people. Out of which 20,751,094 were people under 15 years, 27,096,636 were people between 15 and 64 year while 1.319,653 were above 64 years. According to the United Nations estimates, there is an increase in the population which currently stands at 51,470,286 people (Worldometers.info, 2019). Increase in population has led to greater implications for human resources for health since large population strains resources leading to ill health. Therefore, it requires vibrant health care system with adequate, skilled and well distributed human resources to deal with diseases and other ailments (Health Sector Human Resource Strategy 2014-2018, 2014).

Despite this facts, Kenya's progress in improving the overall health status of its population has yield mixed results. The country continues to face a significant threat from major communicable diseases and rapidly growing non-communicable diseases. According to a research conducted by the Kenya Ministry of Health, with the support of United States Agency For International Development (USAID), it showed that there is a steady increase in the use of outpatient care in Kenya. There was a 35 percent increase from 2007 to 2013, in the average number of visits to an outpatient health provider (EXPLORING 10 YEARS OF HEALTH SERVICE COST AND USE IN KENYA, 2015).

Many hospitals in Kenya, such as The Aga Khan University Hospital, Karen Hospital, Gertrude Children's Hospital, Neema Uhai Hospital and Meridian Medical Center have already invested in software's for Electronic Health Records (EHR). The Electronic health record is an electronic version of the patient's medical history that is maintained by the provider and includes administrative clinical data of a particular person every time they get treatment. These systems make it very easy to retrieve the patient's information, as well as, assures better accuracy in treatment of the patient since the physician can view the medical history of the patient. The records that are produced by this system are just limited to that specific hospital. This make it difficult when a patient visits a different hospital since a new file will be generated for them which will not contain their medical history (Universal Health 2030, 2019).

However, there are many health service providers in Kenya who still use the paper-based method for keeping record of the patients. In this case, when a patient visits an outpatient service, they are

supposed to pay a small consultation fee. Once they pay that fee, they are given a Hospital visit card which they have to keep safe so that they can use it for future use. A unique file for the patient is created which is also known as the patient's medical history file (Mwangi, 2013).

The patient medical history file contains all the details of the previous medication the patient has undergone. It contains following information about the patient: date of visit, vitals, allergies, diagnosis, lab test reports, conclusions and prescribed medicine. Of every visit the patients make, their details are stored in the same file. These files are stored in one room where they are arranged on the shelves either alphabetically or as per the unique number allocated to the patient in ascending order (Mwangi, 2013).

These patient files are made up of paper hence they are prone to wear and tear, for example, file getting torn. This can be awful since it holds crucial information about the patient which is very useful for the medics so that they can administer the correct medication. In cases where the patient file is lost, the reception just creates a new file for the patient. This is not acceptable since all the previous record for the patient would be no longer available and the medics will have to administer medicine based on some assumptions. This can lead to serious health complications on the patient or in some cases also death (Mwangi, 2013).

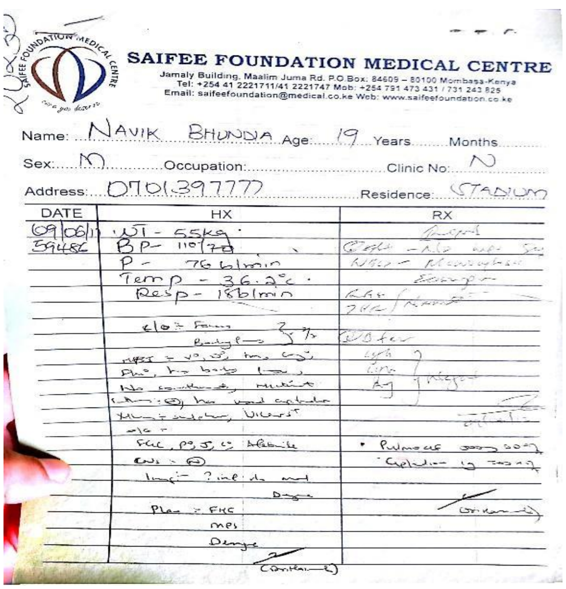
The portability of the patient's file is very minimal. If the patient visits a different hospitals outpatient service, then a new file will be created for them. This makes the doctor to be unaware of the patient medical history, despite the fact, that file is available at another hospital. This makes the patient to stick to the same hospital and incase his medical records are required in another hospital that it get very challenging since the file cannot leave the premises unless otherwise.

It is very hard for the receptionists and nurses to manage the bulkiness of the files. These files occupy a lot of space. It becomes troublesome when they are many patients visiting the outpatient at the same time (Mwangi, 2013). In such cases, there are incidence of patient misidentification that have been reported. In Kenyatta National Hospital, a wrong patient's brain was operated because of misidentification of the patient. This is very bad because it might create other complication for the patient which he/she never had before (DAILY NATION, 2019).

Figure 1 below shows a file room in hospital which is used to store patients files that contain their medical records

Figure 1: Outpatient file room

Figure 2 below shows a patient's medical record card. It is used to keep the track of patient's medical history and is filed in the patient's medical file in that hospital.



SAIFEE FOUNDATION MEDICAL CENTRE

Jamaly Building, Maalim Juma Rd. P.O.Box: 84609 – 80100 Mombasa-Kenya
Tel: +254 41 2221711/41 2221747 Mob: +254 791 473 431 / 731 243 825
Email: saifeefoundation@medical.co.ke Web: www.saifeefoundation.co.ke

Name: NAVIK BHUNDIA Age: 19 Years Months

Sex: M Occupation: Clinic No: N

Address: 0701397777 Residence: STADIUM

DATE	HX	RX
09/06/17	WT - 55kg	
	BP - 110/70	
	P - 76 b/min	
	Temp - 36.2°c	
	Resp - 18b/min	

Figure 2: Patient Medical History Card

1.2 Problem Definition

Lack of a method of identifying patients across multiple health systems for different health providers. This prevents the doctors from accessing the updated medical history of the patients leading to repeated and wrong diagnosis, hence prescribing wrong medication.

In some hospitals, patients medical records are manually handled. Their bulkiness increases time of their retrieval leading to long queues. Moreover, the files wear and tear leading to generation of new patients files with no track of their previous medical history.

1.3 Goal and Objectives

The goal of this project is to come up with a single, consistent patient medial history that is accessible across different health providers. This will be achieved by developing a web-based application that will be able to store patients medical case history uniquely using the unique patient identification number that will be assigned to them and the medical history will be retrieved at any health provider the patient visits.

The research and system objectives for this system prototype are as follows:

1.3.1 Research Objectives

The research objectives for understanding and building this system are to investigate:

1. The process of handling a new patient in outpatient.
2. The process of looking for previous records for existing patient's information.
3. How the patient's record is taken at different levels of diagnosis.
4. How the information is stored for future use.

1.3.2 System Objectives

The objectives of this system prototype are as follows:

1. To generate Quick Response code with a unique identification for each patient.
2. To use the Quick Response code with unique identification generated to access fully updated patient record across multiple hospitals.
3. To evaluate the performance of the system.

1.4 Project Justification

Most of the problems faced by both the medical staff and the patients are well taken care of in Unique Patient Identification and Management Using quick response code web application system prototype.

This system prototype will solve most of the problems in the following ways:

Unique Identification of patients:-It will generate a Quick Response code for the patients using their Birth certificate number, National Identification Number and Passport number (if available). This will assure that each patient is uniquely identified.

Ease retrieving patients' records:-It will reduce the time wasted in searching for patient files which contains the history in huge file rooms. The patient will just be required to scan the Quick Response code and all the medical history will be available in his/her profile. Furthermore, the patient can go to any hospital and their record will be available unlike the paper-based file system.

Secured Records:-It will assure that only the doctor will be able to view the patient's medical history and no other medical staff will have access to it. This will be done by having different access level for different departments. The login credentials will also be recorded so we can know which doctor treated the patient.

Reduce confusion:-It will reduce confusion that might arise when laboratory staff will be trying to conduct different tests on patients' blood sample. This will be achieved by sticking a Quick Response code to the test-tube in which the blood will be collected. So that when they enter the results into the system, they are just required to scan the Quick Response and the profile will open hence they are just required to feed in the results.

Well maintained patient history:-It will assure that the patients' history is well maintained since all the departments will feed the data collected direct into the system. This will reduce the risk of losing important patient data if it were still to be in files.

Accurate data presentation: - It will be able to draw charts on the cases of diseases encountered for weekly, monthly and yearly period. This will help a lot in identifying the trend of diseases across different regions from the analysis given by the charts.

1.5 Project Scope

The scope of this project will be to cover outpatient services of private hospitals in Nairobi county to use this system in patient identification and interoperability of their medical health records among the different hospitals in Nairobi.

CHAPTER 2: LITERATURE REVIEW

2.1 History of Health Medical Records

The first forms of the health medical records were in the form of narratives which were written by the ancient Greeks. This was done to document successful cures on diseases and also to share the observations about the symptoms and their outcomes (Palgon, 2019).

In the 1920s, as the healthcare advanced, the physicians realized that the best ways of improving diagnosis and treatment of illnesses was to carefully document the observations and actions while treating the patients. Therefore, there was standardization of medical records and growth of complete record-keeping from 1920 until 1960s. These records were paper based (Palgon, 2019).

In 1960s, the development of computers opened up ways to maintain records electronically, but it was really expensive to purchase and maintain the mainframes. In 1970s, the computers became smaller, software were designed to support clinical functions of laboratory, patient registration and billing (Palgon, 2019).

From the 1990s till now there have been many advances in the storage of patient's information in the databases that can be used across all the departments of the healthcare organization. This was achieved by the introduction of Master Patient Index (MPI). Despite the advancements in the health systems there is still a gap in aggregating and harmonizing the information from various systems to produce data that can be easily analyzed (Palgon, 2019).

2.2 Health Medical Records in Kenya

In Kenya there are two main ways in which the patient's medical records are kept, these are, Electronic Medical Record (EMR) and Paper-based filing system.

2.2.1 Electronic Medical Record

Most of the private hospitals in Kenya have started investing on Electronic medical record systems. These systems capture all details of the patient's medical history . Therefore, when a patient visits a hospital using Electronic Medical Record system, they can just walk into the consultation room, pay for the tests, head to the laboratory for examination, go back to the doctor's office for diagnosis and get the prescription from the pharmacy without having to carry any prescription notes or patients file since all the communication is done online (Universal Health 2030, 2019).

These systems speed up the process of attending to patients and curing them since it does not require manual searching and retrieval of patient medical file. It also helps the physician in making

accurate prognosis since they have full access of the patient's medical history (Universal Health 2030, 2019).

2.2.2 Paper-based Patient Medical Record

Despite the fact that many hospitals are moving to Electronic Medical Record, there still exists many hospitals that store the patient's medical record in a paper-based filing system. When the patients visited the hospitals, they are required to pay consultation fees. Once they pay the consultation fees, they are given a hospital visit card which hold the patients number and names. The patient is required to go to the hospital with that card whenever they fill ill (Mwangi, 2013).

A paper file is kept for each patient who visits the hospital in which the details of the patients, their vitals, allergies, diagnosis and treatments information is kept. Due to high population and bulkiness of these files, it becomes hard to have proper storage and retrieval of these files. When a patient goes to outpatient facility. They first go to the reception to show their hospital visit card. They have to stand in long queues waiting for their chance. Once they show the reception their card. The patients file is retrieved from the file room and proceeds to the vitals room with the file. At the vitals room the nurse takes the patients vital information and records it in the file (Mwangi, 2013). The patient the proceeds to the doctor where he/she is diagnosed by the doctor. The doctor asks the patient relevant questions to collect the symptoms. He/she then records these symptoms in the patient's file. If the symptoms are clear enough for the doctor to conclude on what the patient is suffering from, then the doctor writes his/her conclusion on the file and prescribes appropriate medicine for the patient. In the case where further diagnosis is required by the doctor to conclude, the doctor writes the lab test to be carried out in the patient's file and the patient is sent to the laboratory department.

At the laboratory depending on the type of the test to be carried out i.e. blood test, stool test or urine test, the appropriate sample is taken from the patient and test is carried out. Once the lab report is out the lab technician writes in the file the conclusion of the test. The patient goes to the doctor where now depending on the conclusion of the test he/she administers the medicines to the patient.

All this information about the patient is stored in the file. This information is very crucial for accurate treatment of the patients. The information gives the medics a better understanding of the

patient's medical history, the allergies and other crucial information which help the doctor to know what exactly the patient will be suffering from.

Advancement in the ICT sector in Kenya has led to improvements in the health sector. The District Health Information System and Electronic Medical Records have improved the manner in which the medical records are now being stored. Many Hospital in Kenya are now investing so much money in it.

Despite these advancements in the recent years, many hospitals are still using files to keeps the patients records and these records are updated into the digital system either by scanning them or by keeping a clerk who feeds in this information.

2.3 Quick Response Code

Quick Response code is a two-dimensional square barcode which can store encoded data. They were created in 1994 by Denso Wave, a Japanese subsidiary in Toyota Group. The use of this technology is now free (Unitag, 2011).

They are very useful since they can store (and digitally present) much more data, including URL links, geo co-ordinates and texts (Lyne, 2009). Quick Response code are quite unique since for every difference in the data that is used to generate it and unique Quick Response code is created. Therefore, Quick Response codes will be generated for patients using their Birth Certificate number, National Identification number and Passport Number. This Quick Response code will be used to identify the patient uniquely in which ever hospital they go as long as the hospital uses the same system.

2.4 Similar Existing System

There are some systems that already exists in the process of digitizing the patient's medical records, as well as, collection of this data and analyzing it to create patient-based statistical data

2.4.1: KenyaEMR

The KenyaEMR is a tailored distribution of OpenMRS which meets the requirements laid out in the Kenya Ministry of Health document. This system was originally developed by I-TECH and is currently being supported by Palladium Group through KHMIS Project (Wiki.openmrs.org, 2016). It contains modules that are potentially reusable by other distributions, and other general- purpose community modules. It also contains API which allows add-on modules to add new content such as forms and reports. The system uses the new OpenMRS UI Framework module to create an entirely new point-of -care oriented user interface (Wiki.openmrs.org, 2016).

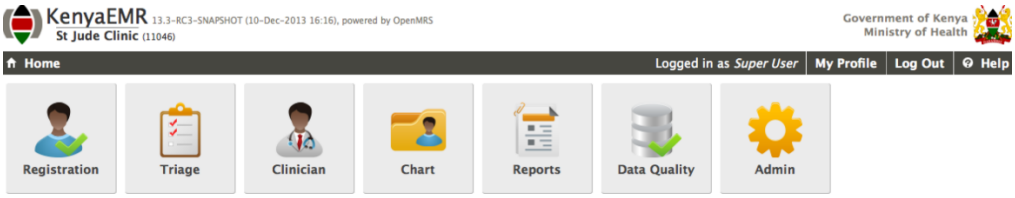


Figure 3: KenyaEMR system homepage.

2.4.2: IQ-CARE

It is a robust electronic medical records package and has been deployed in over 100 locations in Kenya, Uganda, Nigeria and Zimbabwe. It is flexible and scalable with features to create multiple departments and forms. It also sets up facility and patient home page reports and queries. The data

of the patients is entered into the system by the clinicians through a browser front-end (CodePlex Archive, 2018).

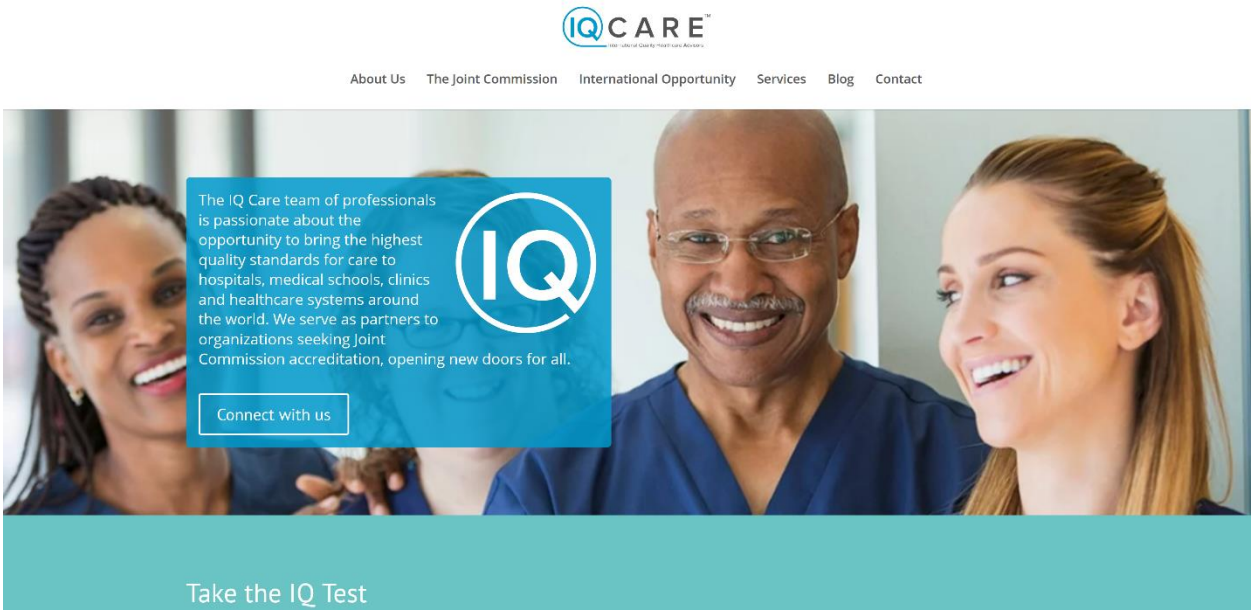


Figure 4: IQ-CARE system homepage.

2.4.3: District Health Information System 2

The District Health Information System 2 is a tool for collection, validation, analysis and presentation of aggregate and patient-based statistical data that is used to integrate health information management activities. It is a generic tool with an open meta-data model and a flexible user interface that allows the user to design the contents of a specific information system without the need for programming (DHIS2 User guide, 2018).

This system was developed by the Health Information System Programme (HISP). Some of the key features in the DHIS 2 are:

1. It provides comprehensive data management solution based on data warehousing principles and a modular structure which can easily be customized to different requirements of a management information system, supporting analysis at different levels of organizational hierarchy.
2. It provides data entry tools which can either be in form of standard lists or tables or can be customized to replicate paper forms.
3. Provides different kinds of tools for data validation and improvement of data quality.

This system works in the following way in the process of data: collection, quality, processing, analysis and usage (DHIS2 User guide, 2018).

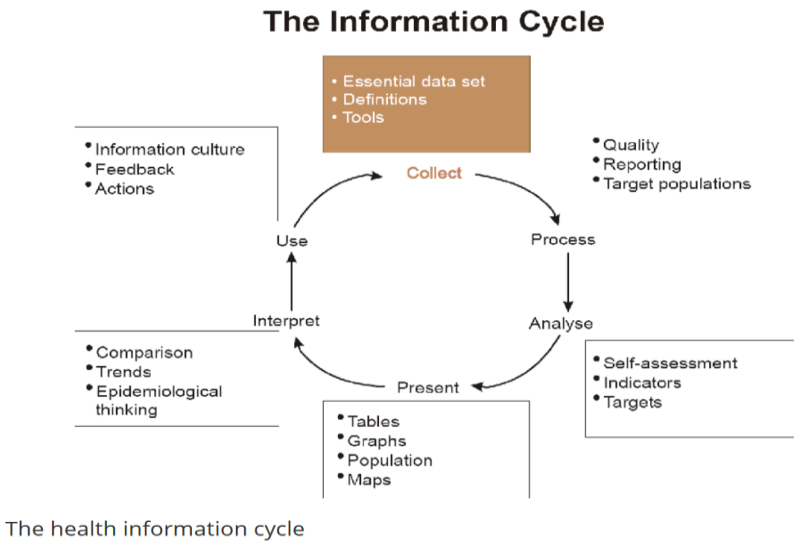


Figure 5: Information Life Cycle DHIS 2

2.4.4: Hospitali plus

This is a hospital management system which helps the hospitals to manage its resources well. It also captures the patient's information and keeps the history. The software was created so as to minimize the challenges that the clinicians undergo when they are digging through boxes of medical records, often losing valuable time and energy, trying to assign new patients to new charts and asking patients to shuffle their medical records through the queues and sometimes losing track of who had what data in their clinics (Magnum, 2017).

2.4.5: Electronic Patient Record Management System

This is an Electronic Patient Record Management System that has a centralized Database which contains the in-patient records. It is implemented using a combination of PHP and MySQL database. It contains patient's personal info, departmental lies-in, physician, treatments and lab results.

2.4.6: Patient Record Information System (PaRIS)

This is a Patient Record Information System for primary health care centers in Indonesia. It is a cost effective and sustainable system that enhances the work of a physician in order to provide better and applicable public health care service.

2.5 Gap that exists with the available systems.

The existing systems in the above section still lack a proper way of uniquely identifying the patients and collecting the data in a proper way. Therefore, by the use of Quick Response code which will be created using their birth certificate number, ID number and passport number if non-citizen to identify the patients uniquely. It will be very easy to directly input the patient's record into their profile in the system as they are being diagnosed and treated at various departments in the out-patient, they include Reception, Triage room, Doctor, Laboratory, Radiology, Pharmacy.

It will also reduce confusion on the lab technicians since the can print the Quick Response from the system and stick to the vacutainer that contain the specimen hence while recording the test results, they just scan the Quick Response and directly record into the patient's profile. Therefore, this system will be beneficial for both the medical staff and for the patients as well since they will receive very accurate treatment. Furthermore, the medical records of the patient will be available at any hospital as long as they are using the same system.

CHAPTER 3 : METHODOLOGY

3.1 System Development Methodology

3.1.1: Iterative Prototyping Methodology

The main methodology used to develop this system will be iterative prototyping method. This is a software development methodology in which a prototype of the system is made and is it improved from time to time based on the reviews obtained on the prototype from the stakeholders and final users of the system.

The idea of using the prototyping model is to first come up with a representation of the module in the first stages so that the modules requirements and specification are clearly understood by the developers which assures that the system will meet its goals.

Each development cycle provides an opportunity for feedback, refining the requirements and viewing the progress made in the system development. This process iterates until a final required system is obtained. The image below shows how the iterative prototyping methodology works.

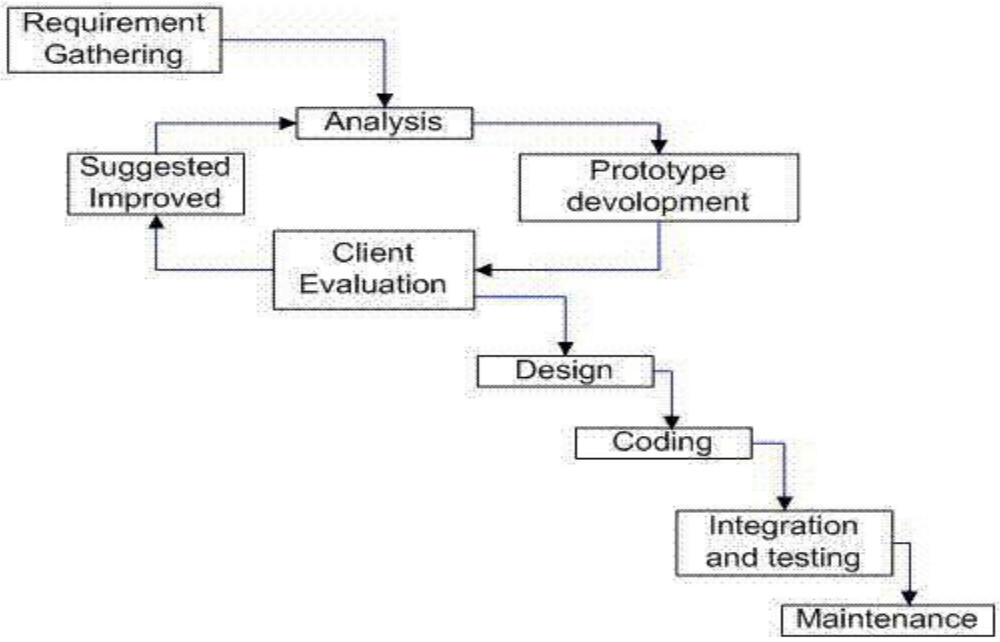


Figure 6: Iterative Prototyping Methodology

The methodology was implemented in the following manner:

1. **Requirement gathering and analysis**: In this phase, The requirements for the proposed project were collected and analyzed. The requirements were then documented after conducting repeated analysis until well elaborated specifications were obtained and the users were satisfied with the prototype. The following were the methods used to gather information that was required:
    a. Unstructured interviews were conducted with the aim of knowing how patients are identified, the process they undergo and how the data is recorded at different levels which include, Reception, Triage, Doctor, Laboratory and Pharmacy.
    b. Online document was reviewed, and they were mostly related to how the patients cards, lab test request form, and other materials that are used while recording the patient's medical data.
2. **System Design: In this stage**, the overall system architecture was defined. It involved studying of specifications gathered in the requirement gathering and analysis to come up with the design for the overall system. The design technique used was Unified Modelling Language and it included:
    a. Use case diagram
    b. Flowcharts
    c. Conceptual design
3. **Implementation**: It involved coming up the system code of the several units required in the system. It included general interface design and user interface design according to the type of users and functionality.
4. **Integration and Testing**: In this stage, units which were developed at the implementation stage were integrated into the overall system. The various tests carried out were:
    a. Unit testing
    b. Module testing
    c. Validation testing
    d. System testing

5. **Deployment and Maintenance**: Once the testing was conducted and all the errors were corrected, the system was deployed. If any concerns arise after system deployment, then will be handled under maintenance of the system through upgrades of the system.

3.1.2: Justification for Iterative Prototyping Methodology

The main aim of using the iterative prototyping method is that:

1. It provides robust user feedback.
2. The developers can easily identify problems in the earlier stages of development.
3. It measurably increases the usability of the system since the users of the system will be actively involved during the development of the system.

3.2 Research Methodology

This describes the research design that will be used while doing the research on the patients' identification and record management. Most of the research will conducted at the outpatient sector of hospitals. The key objectives of carrying out the research will be to:

1. Identify how the hospitals identify the different patients who visit the outpatients.
2. The process the patient undergoes when visiting outpatient.
3. How they medical data is recorded at each visit.

The research will be conducted by :

1. Interviewing the medical staff at different levels and some of the patients as well.
2. Observing the medical staff while they are conducting their duties.
3. Researching on relevant websites to get further information.

3.3 Resources Used

Some of the resource that will be used are:

1. ReactJS: to build the front-end of the system
2. NodeJS: to create communication between front-end and back-end.
3. Express: to generate the API routes to allow communication from back-end to database.
4. MySQL database: for storing the records.
5. Ubuntu/windows operating system.
6. Apache server: for hosting the website.

3.4 System Analysis

It refers to the process of decomposing the proposed system into components so as to come up with the requirements that are necessary for the components to work well together to achieve the overall goal of the system (Anon, 2015).

3.4.1 Data Gathering and Collection Methods

Several approaches were used to gather data that was necessary for the development of the proposed system. There were two main data collection methods used to obtain data for this project they include, Unstructured interviews and Online document reviews.

3.4.1.1 Unstructured Interviews

Unstructured interview refers to interviews in which the questions to be asked are not set prior to the interview. The interviewer can have some basic structure but not fully prearranged questions to ask. These interviews were conducted with Dr Hetal Gohil who holds Bachelors in Medicine and Surgery, currently working at Kenyatta National Hospital and Nairobi Hospital. The other professional consulted was Urvi Davadra who is a Medical Laboratory Scientist and works at Pandya Memorial hospital. The main goal while interviewing them was to know how the patients are identified, the process they undergo and how the data is recorded at different levels which include, Reception, Triage, Doctor, Laboratory and Pharmacy.

3.4.1.2 Online Document Reviews

It refers to the process of collecting data through reviewing already existing documents. These documents were mostly related to how the patients cards, lab test request form, and other materials that are used while recording the patient's medical data.

3.4.2 Feasibility Analysis

Feasibility study is one of the crucial things since it's a study conducted to evaluate the proposed systems potential to succeed. It gives the measure of how beneficial the system can be to the organization. The feasibilities carried out for this system include:

3.4.2.1 Operational Feasibility

It refers to the measure of how well the proposed solution will help the organization. It helps in taking advantage of the opportunities and fulfilling the requirements so that the users are satisfied while using the system. The proposed system is therefore designed in such a way for the user to use the system easily based on their roles.

3.4.2.2 Technical Feasibility

It refers to the measure availability of the technology resources and level of expertise that would be necessary to build and run the project. Most of the software used is open source and readily available in the market include React JS, Node JS, express and MySQL database. The users of the system also have adequate knowledge that is required to interact with the system.

3.4.2.3 Economic Feasibility

It refers to the measure of money aspects related to the project by performing a price profit analysis and addressing each tangible and intangible advantage of the system. The software used to develop the system are open source and readily available hence making it cost effective. Hardware required will be a computer which is available. The expenses that will be involved will be cost of developer, internet connection and hosting of the service. These costs are reasonable compared to the service they will provide hence making it economically feasible.

Item to be used	**Estimated Costing (Ksh)**
Laptop	65000
Zuku Internet connection	6000
Apache Server	
Transport fees	7000
Printing	3000
Total	**81,000**

Table 1: Budget costs for the whole project

3.4.2.4 Schedule Feasibility

It refers to the measure of the time that will be required to fully develop and successfully implement the system. The time allocated to this project is 8 months which is reasonable to develop and implement most of the functionalities of the proposed system.

Task			
Description	**Start Date**	**End Date**	**Duration(days)**
Research and Problem definition	10/12/2018	21/12/2018	11
Creating Proposal	22/12/2018	26/12/2018	4
Refinement of proposal	1/01/2019	6/01/2019	6
System Analysis & Requirement elicitation	10/01/2019	30/01/2019	20
Data Gathering	31/01/2019	09/02/2019	10
System Design	10/02/2019	12/03/2019	30
System Design Refinement	13/03/2019	17/03/2019	5
Coding front-end & User interface	18/03/2019	13/04/2019	27
Coding database	14/04/2019	08/05/2019	25
Testing and refining Front-end	09/05/2019	18/05/2019	10
Testing Database	19/05/2019	28/05/2019	10
Finalization of Project report	28/04/2019	1/06/2019	161

Table 2: Timelines for the project.

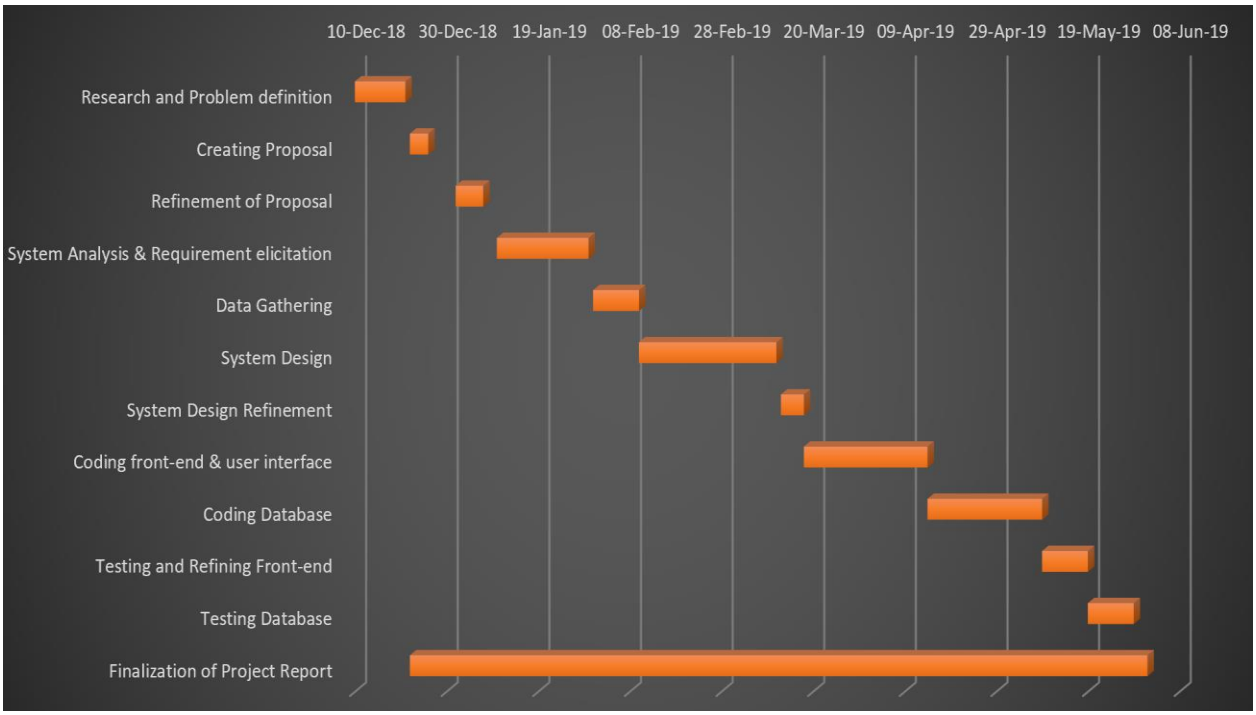


Figure 7: Gantt chart of the project development.

3.4.3 Requirement Elicitation

It describes the functional and non-functional requirements that the proposed system will accomplish. The elicitation for this project is broken down into functional and non-functional requirements.

3.4.3.1 Functional Requirements

It refers to the functions that will be performed by the system. The system should be able to:

1. Register and Login hospital staff based on their occupation.
2. Register patient uniquely and generate quick response code for them.
3. Scan the patients quick response code and prompt for consent.
4. Add patients medical data at respective departments.
5. Show patients medical history to the doctor.
6. Shares patients fully updated patients medical history to another hospital at patients consent.

3.4.3.2 Non-functional Requirements

It refers to the desirable features the proposed system should have. They include:

1. **Security**: The system should only permit access to the system based on role and should only show medical history based on patients consent.
2. **Usability**: The system should have a good user interface which is simple and easy to use by the users.
3. **Efficiency**: The system should make it easy and faster retrieval and storage of patient medical records.
4. **Scalability:** The system should be able to include as many hospitals as possible.
5. **Reliable**: The system should be available and accessible at any time.

3.4.4 System Analysis Model

3.4.4.1 Use Case Diagrams

It refers to diagrams that are used to analyze the systems high-level requirements and these requirements are represented using different use cases. The list below shows the relevant user requirements that were used to generate the use case diagram.

1. System Admin
    a. Register hospital staff.
    b. Approve edits required by hospital staff.
    c. Keep record of hospital staff.
    d. Keep record of patients
2. Receptionist
    a. Login into system.
    b. Register patient.
    c. Print patient quick response code.
    d. Regenerate patient ID.
    e. Start patient case.

3. Triage Nurse
    a. Login into system.
    b. View the patient waiting list.
    c. Enter patient vitals.
4. Doctor
    a. Login into system.
    b. View patients waiting list.
    c. View patient medical history.
    d. Add patient allergies, complaints and prognosis.
    e. Send a laboratory test request form.
    f. View laboratory test results.
    g. Add patient diagnosis and prescription.
5. Laboratory Technician
    a. Login.
    b. View patients waiting list.
    c. View test to be conducted.
    d. Print quick response code.
    e. Enter laboratory test results.
6. Pharmacy
    a. Login
    b. View patients waiting list.
    c. View prescription.
    d. Print prescription.

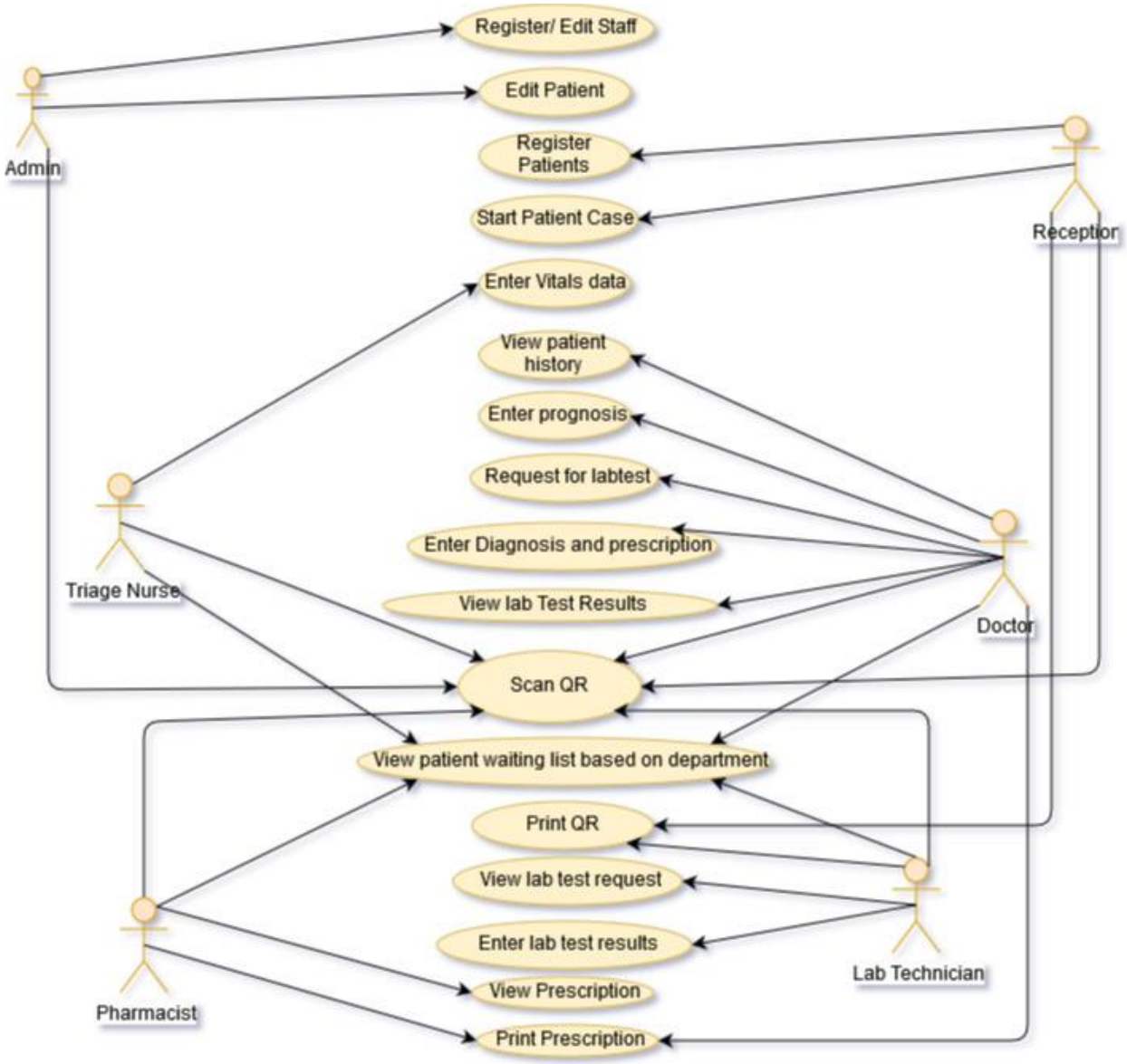


Figure 8: Use Case diagram for the system.

3.4.4.2 Data Flow Diagrams

It represents the flow of data through the system and the processes that act on the system.

DFD level 0

The Data Flow Diagram below shows how the different external entities will interact with the system.

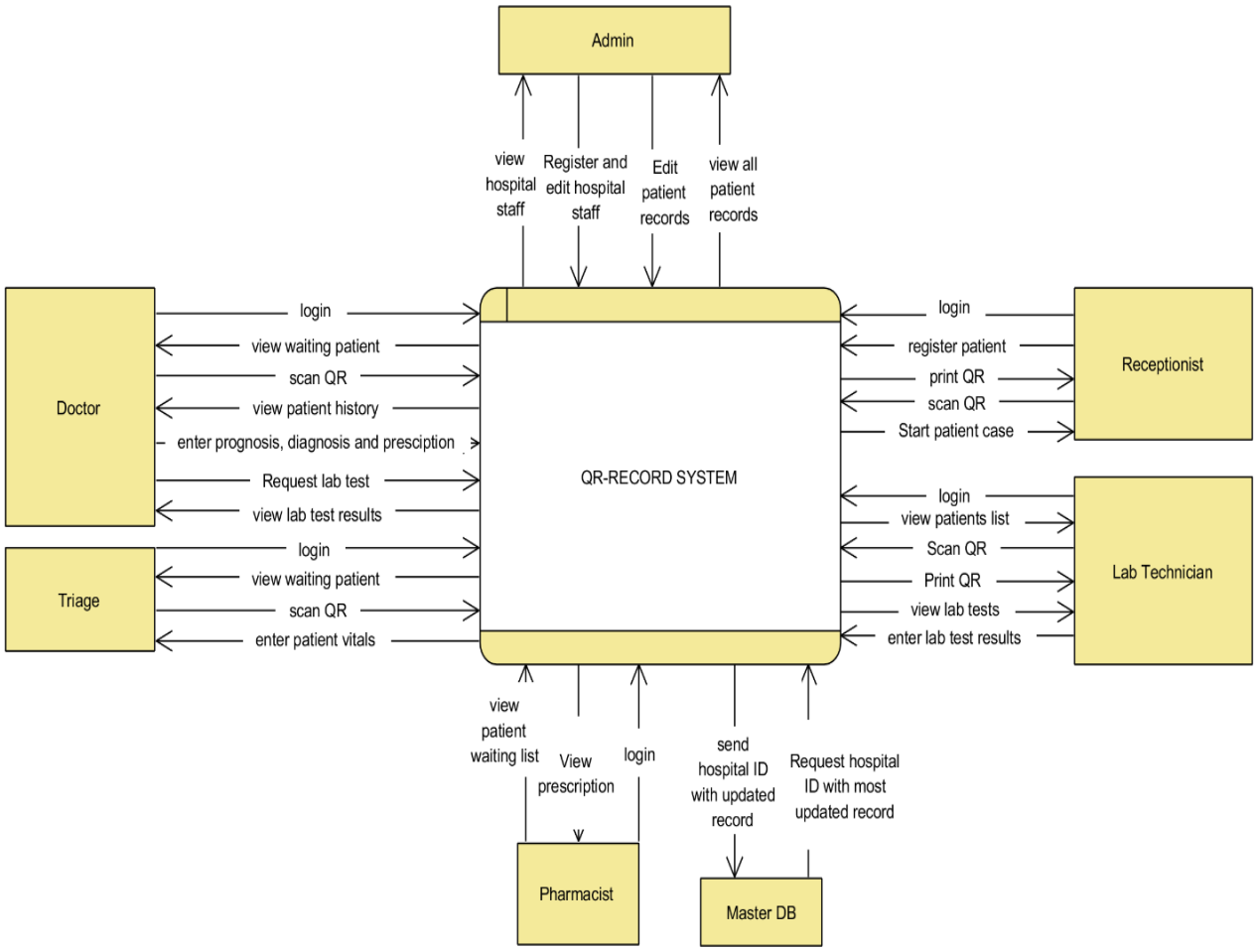


Figure 9: Data flow diagram level 0

DFD level 1

The Data Flow Diagram below shows how multiple processes and sub-processes will interact with each other.

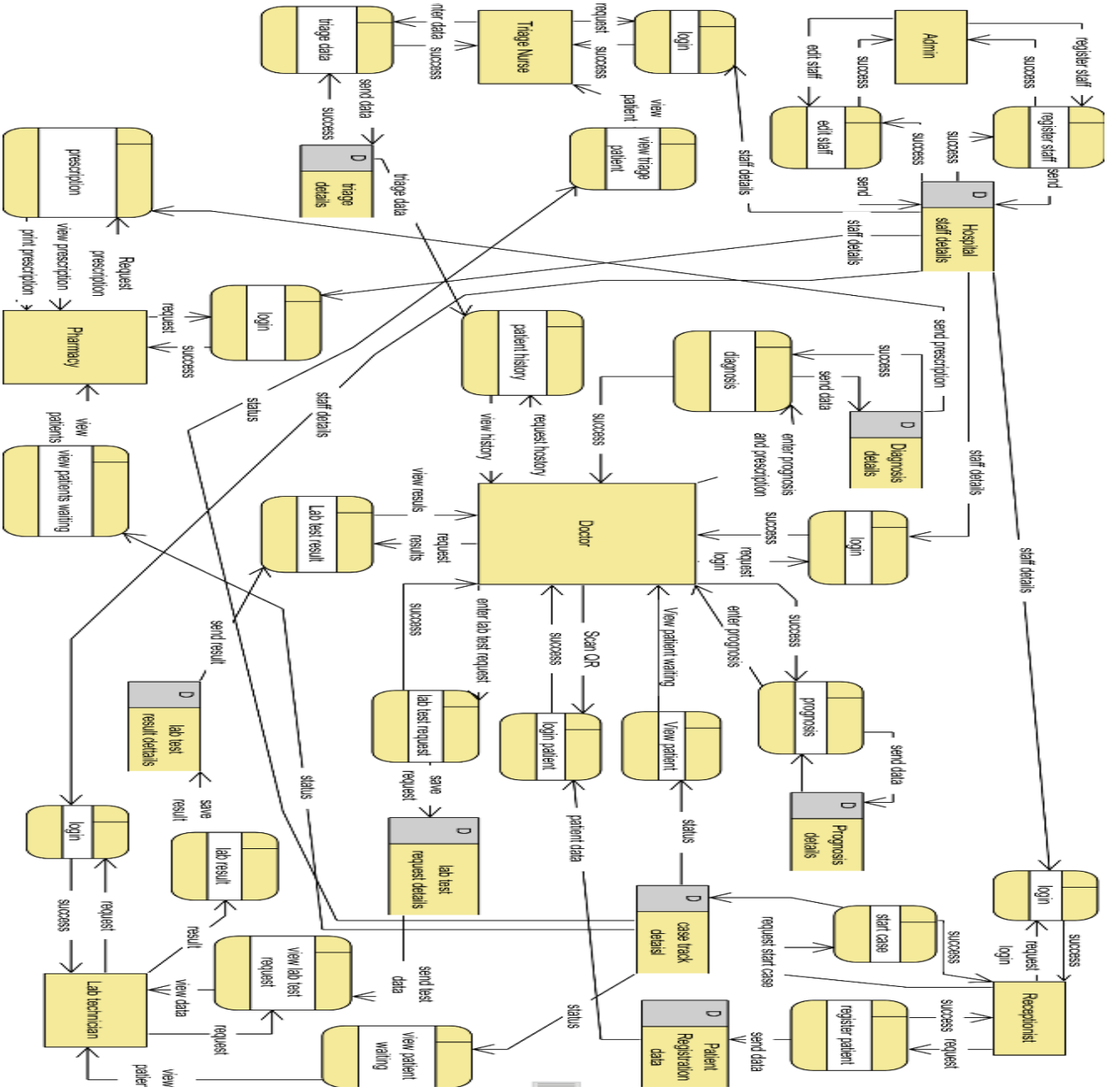


Figure 10: Data Flow Diagram level 1

3.5 System Design

It refers to the process of defining the system architecture, components and interfaces for a system to satisfy specified requirements. A good design is very crucial since it will determine how the system will be implemented.

3.5.1 Conceptual Architecture

The concept of this system is to come up with unique identification for patient, This will be done by registering patient depending on whether they are Kenyan citizen. If they are Kenyan citizen, then they will be further classified based on their age. The patients above 18 year will be registered using National ID number and National Hospital Insurance Fund Number, whereas, if the patient is below 18 years then they will be registered using Birth Notification number and Birth Certificate Number. Finally, if the patient is not a citizen then they will be registered using passport number. A quick response code will be generated for this unique patient identification and will be used to store the patients' medical history. If a patient visits hospital A the reception will start a case for the patient. The hospital A local database will communicate to the master database using application programming interface to check which hospital holds the patients latest record. The master database will send hospital B ID to hospital A since in this case we assume hospital B ID holding the patients updated medical record. Then hospital A will then communicate with hospital B with consent from the patient, which is a One Time Generated password that will be sent to the patient via SMS, on the record it requires so as to have an updated record of the patient. Hospital B will send Hospital A the record which it has requested for and then Hospital A will start the patients service through triage, doctor, lab technician and pharmacy then finally updates the master that it holds the updated record for the patient. The figure below shows how the local database of hospital will communicate with the master and then to the hospital holding the updated record.

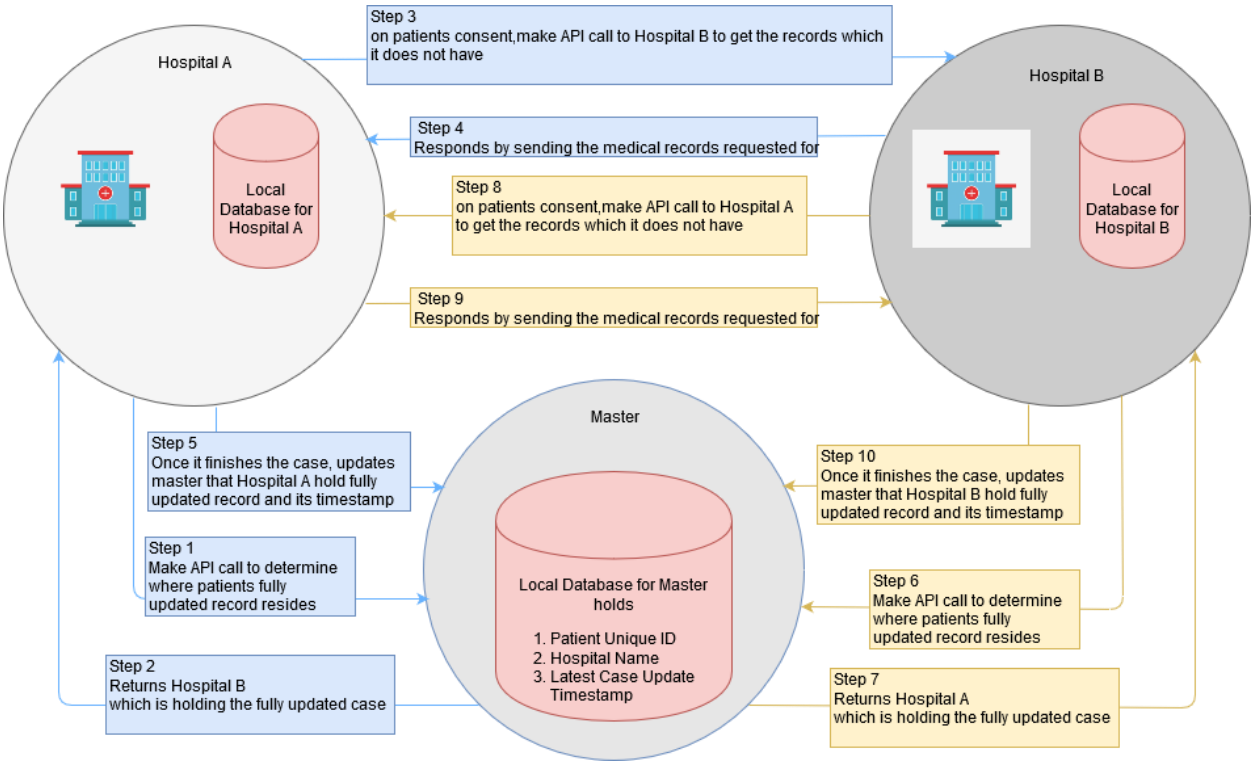


Figure 11: Conceptual Architecture for the system.

3.5.2 System Flowchart

3.5.2.1 Hospital Interaction Flowchart

The diagrams below represent the whole process when a patient visits an outpatient service.

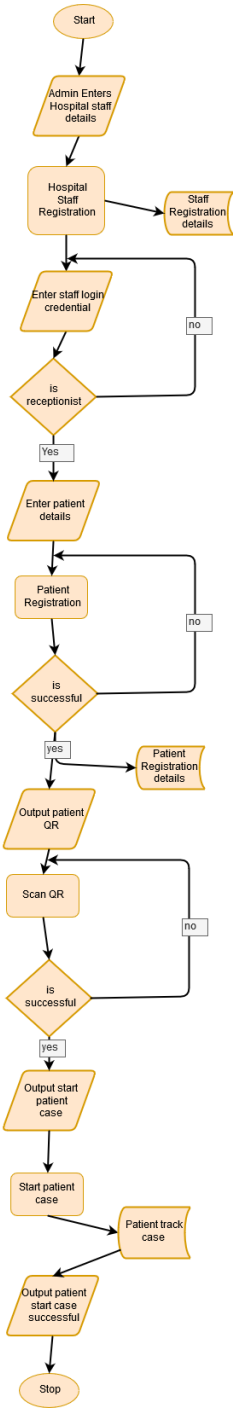


Figure 12: Reception interaction flowchart

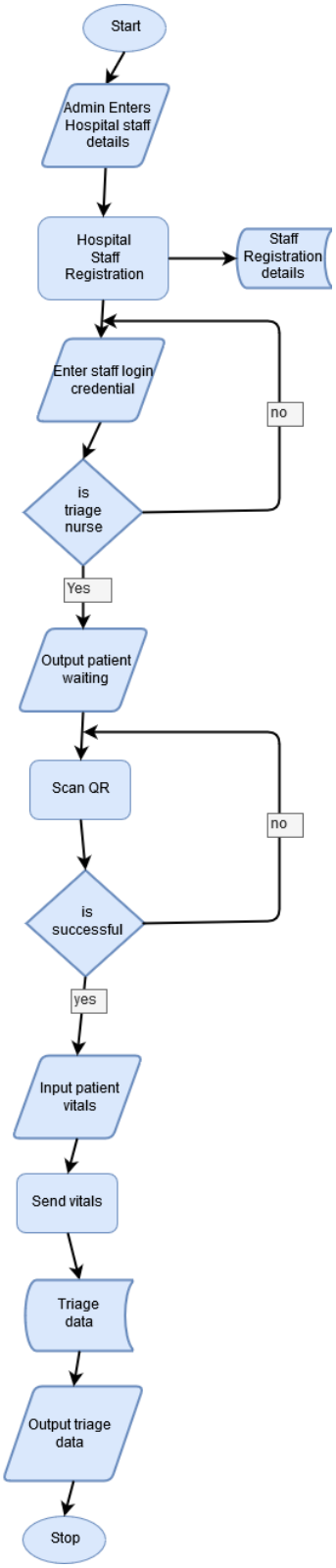


Figure 13: Triage interaction flowchart

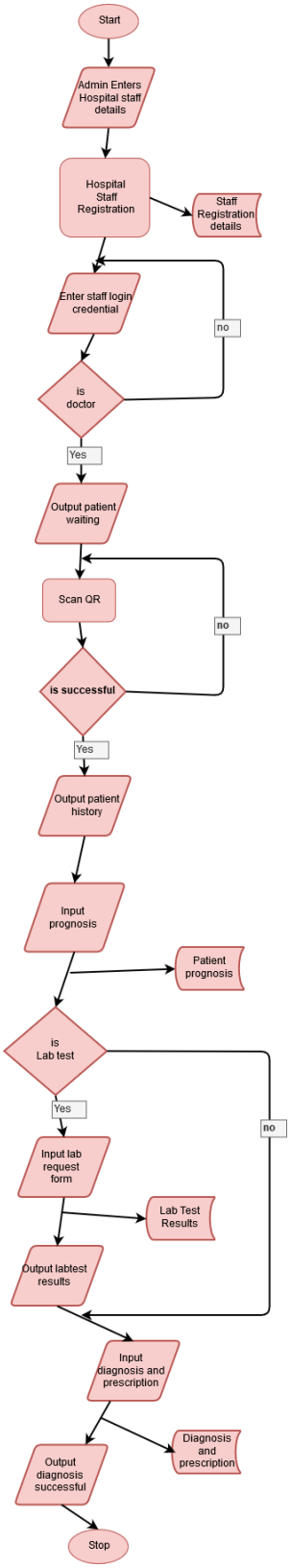


Figure 14: Doctor interaction flowchart

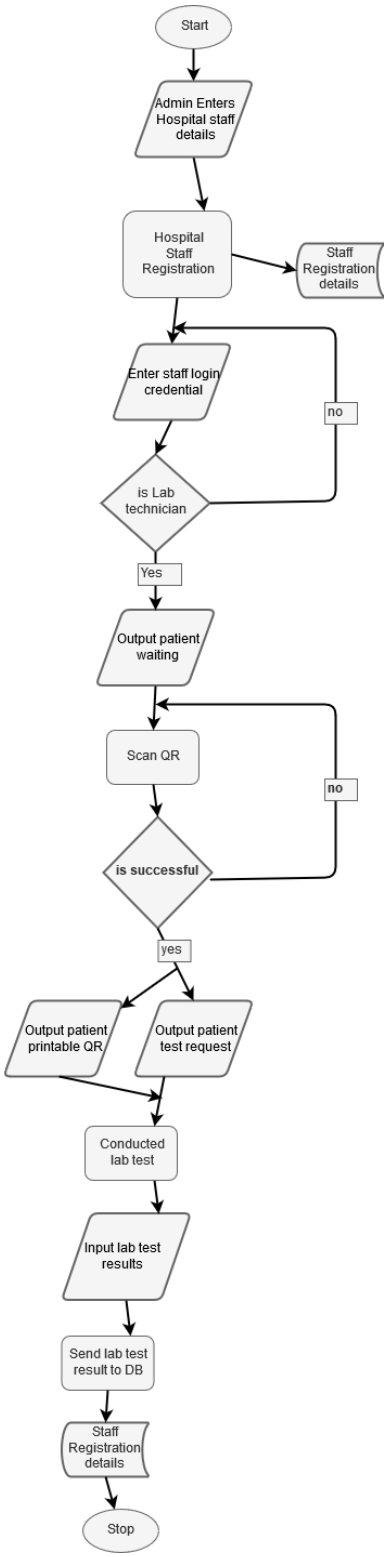


Figure 15: Lab technician interaction flowchart

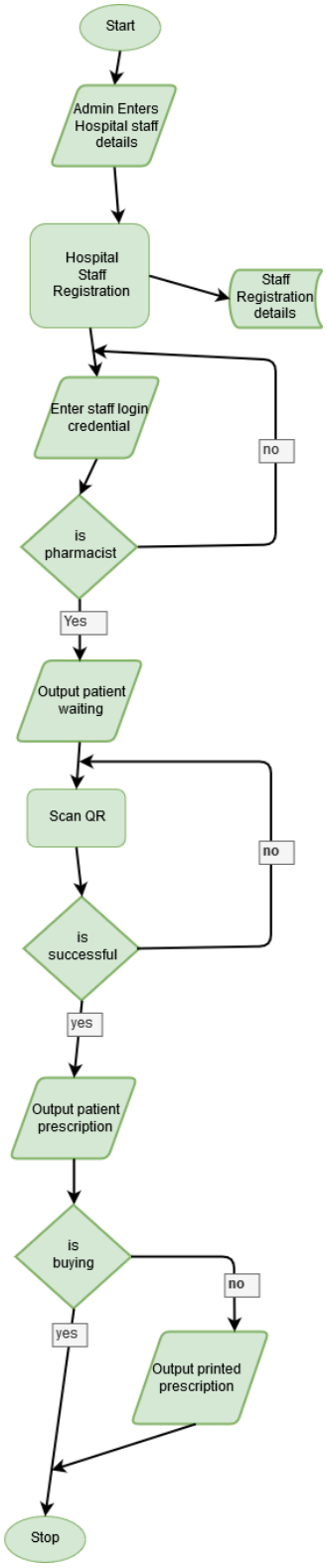


Figure 16: Pharmacy interaction flowchart

3.5.2.2 Data Interchange Flowchart

The figure below shows how the data will move when a patient visits another hospital.

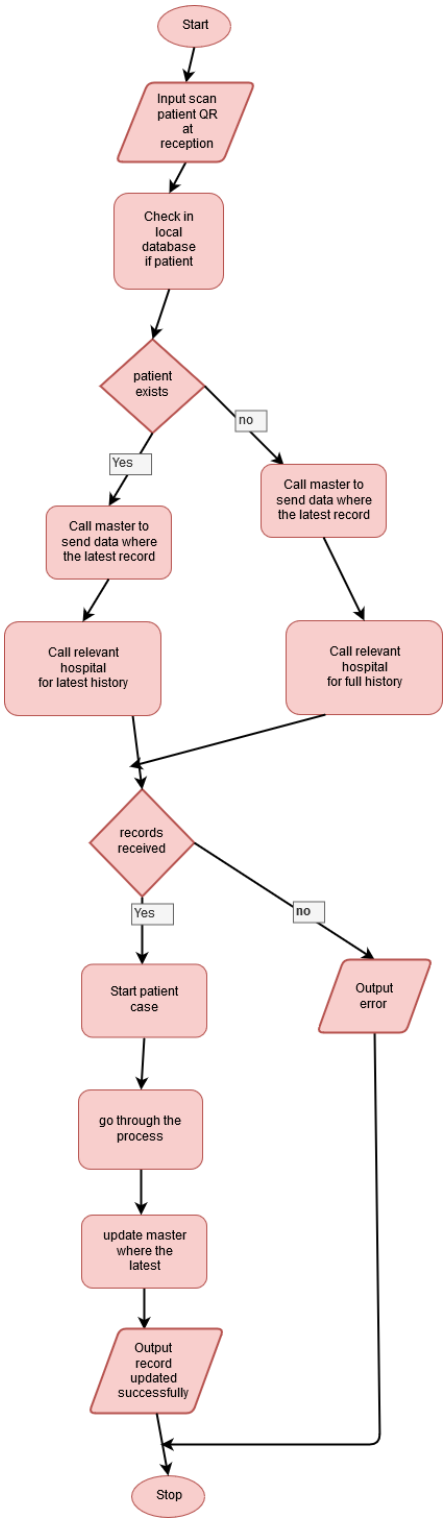


Figure 17: Data Interchange Flowchart.

3.5.2.3 Patient Unique Identification Number Generation Flowchart

The figure below shows how the patient unique identification number will be generated and finally be converted into a quick response code.

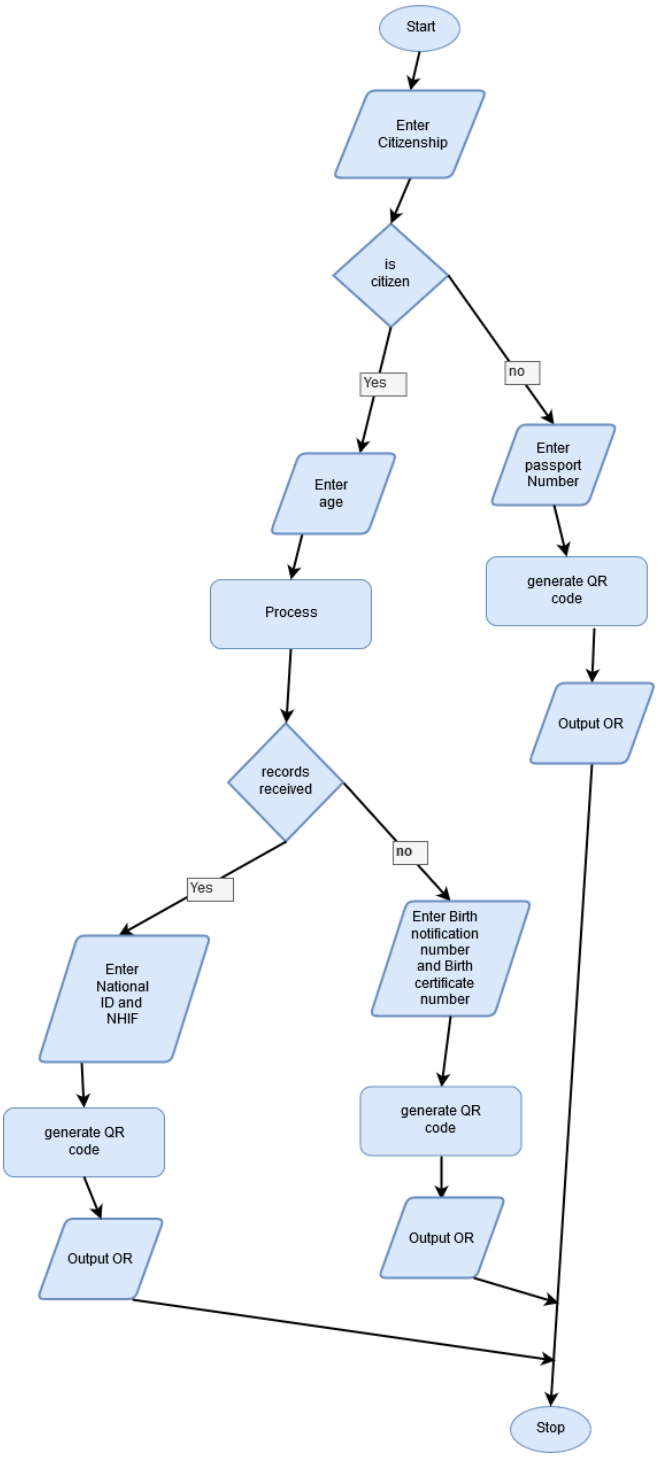


Figure 18: Unique Patient Identification Number Generation Flowchart.

3.5.3 Entity Relation Diagram

3.5.3.1 Local Database Entity Relationship Diagram.

The figure below represents local database entity relationship diagram.

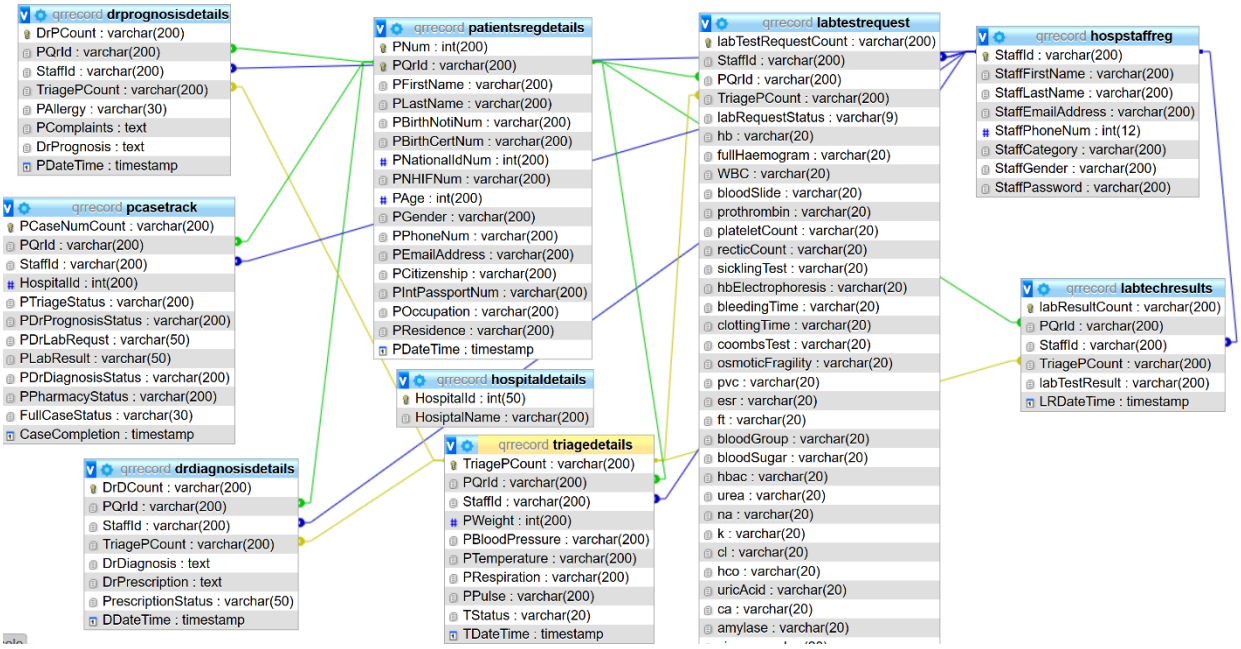


Figure 19: Local database Entity Relationship Diagram

3.5.3.2 Master Database Entity Relationship Diagram

The figure below shows master database entity relationship diagram.

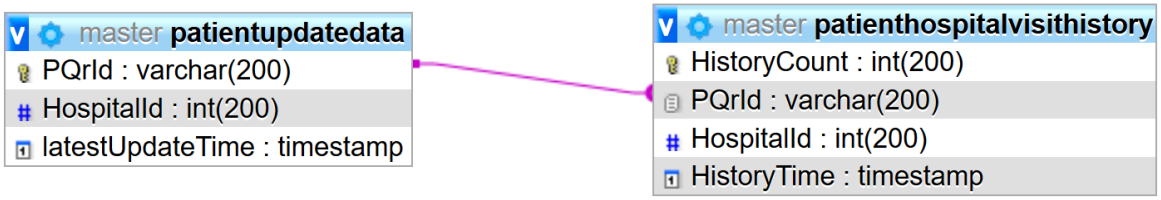


Figure 20: Master database Entity Relationship Diagram

3.5.4 User Interface Design

3.5.4.1 Designing Web-application User Interface using ReactJS

React makes it painless to create interactive user interfaces. One can design simple views for each state in the application and React will efficiently update and render only the right components when the data changes. Furthermore, it makes the code more predictable and easier to debug.

The entire web-application system user interface was build using React components and additional styling was given to each component using CSS and Material UI, which makes the user interface appealing and user friendly to the main users of the system.

CHAPTER 4: IMPLEMENTATION AND TESTING

4.1 Implementation.

This chapter describes how the system was developed. The hardware specifications, software and programming tools.

The QRRECORD system was developed using the following resources.

4.1.1 Hardware Specification

The laptop used was HP OMEN and its basic specifications are:

1. Processor Intel Core i7.
2. RAM 16 GB,
3. Storage 2 TB.

4.1.2 Software Specifications

The software's and different development environments used are as follows:

1. **Windows 10 Operating system**: This was the operating system used in my laptop.
2. **Visual Studio Code**: This is the development environment used to write both frontend and backend code.
3. **Africa's Talking SMS Gateway**: This was used to integrate SMS to the project.
4. **Virtual Box:** This was used to create two virtual machine running windows operating system, to demonstrate a different hospital and master.
5. **XAMPP V3.2.2:** It was used to provide MySQL services.
6. **Mozilla Firefox:** It was used to display the web application.
7. **MySQL database management**: It was used to create database for the system.
8. **GitHub for version control**: It was used to take backups for the system as it was being developed and keep track of the different versions.
9. **ReactJS**: It was used to create frontend for the system.
10. **NodeJS**: It was used to create backend for the system.
11. **ExpressJS**: It was used to create API routes for system so that ReactJS frontend could communicate with NodeJS backend which eventually inserts the data to the database.

4.2 Testing

Testing is one of the most crucial part in integration of a system. It is used for checking purpose so as determine that the application is free of errors, carries out all intended functionality and produced results which were expected from it.

4.2.1 Models/Unit Testing

The system was developed on a module by module basis, and once a mode was completed it underwent testing to ensure that it was necessary to ensure that as a unit, the module had no errors and that all the defects identified were corrected before integrating with the other component of the system.

Test case 1: Hospital Staff Registration

Input: Hospital staff details

Output: It registers the Hospital staff successfully if data in all fields were entered properly, else gives an error if one of the fields was not correct.

Test case 2: Hospital Staff Login

Input: Hospital Staff username and password.

Output: Login is successful if both the credentials presented are correct. If they don't match, then an error is returned.

Test case 3: Patients Registration Form

Input: Patient details

Output: It successfully registered the patient both at the local database and the master database and generated a printable unique patient identification card.

Test case 4: Regeneration of unique patient Id

Input: Birth Notification Number, Birth Certificate Number, National ID number and Passport Number.

Output: It successfully regenerates a printable unique patient ID for that specific patient.

Test case 5: Patient Start Case

Input: Scanning patient id

Output: The checks where the patient exists and the sends a SMS to the patient. Once the patient gives the password, case for the patient can be started.

Test case 6: Triage Form

Input: Triage details

Output: Triage details are submitted successfully if all the fields are filled properly, else an error is displayed showing field with error.

Test case 7: Prognosis Form

Input: Prognosis details.

Output: Prognosis details are submitted successfully if all the fields are filled properly, else an error is displayed showing field with error.

Test case 8: Lab Test Request Form

Input: Lab tests to be conducted.

Output: Lab Test is submitted successfully if all fields are filled properly, else an error is displayed showing field with error.

Test case 9: Lab Test Result Form

Input: Lab Result details.

Output: Lab Result is submitted successfully if all fields are filled properly, else an error is displayed showing field with error.

Test case 10: Diagnosis Form

Input: Diagnosis details.

Output: Diagnosis details are submitted successfully if all the fields are filled properly, else an error is displayed showing field with error.

Test case 11: Prescription

Input: Patient unique ID.

Output: It displays patients prescription and allows to print the patients prescription.

4.2.2: Integration Testing

This was focused on the design and how the various modules when integrated can generate the patient history accordingly and it can also retrieve data from the master and other hospital accordingly.

Test case 1: Patients History

Input: Scanning unique patient ID.

Output: It produces all patient history of the patient. It orders history case-wise and displays from the latest ongoing case to oldest case of the patient.

4.2.3 Validation Testing

Form validations have been used for all the forms in the system. The sample validations include:

1. The length of National ID number should not be more than 9 digits and it should not have any Alphanumeric.
2. The length of Phone number should not be more than 12 digits.
3. Email address validations.
4. Presence of the required field validations.

4.2.4 Usability Testing

The usability testing was done using peers and one professional. Most of the usability testing was carried out with peers and they tested the system to judge the following:

1. **Friendly Interface**: The system interface should be user friendly and it should allow the user to navigate back and forward. In case of any mistake, the user should be able to continue without any problem.

   The peers liked the interface and were able to carry out all actions without any hesitations.
2. **Ease of Use**: The system should be easy to use, and major content should be easily visible and accessible.

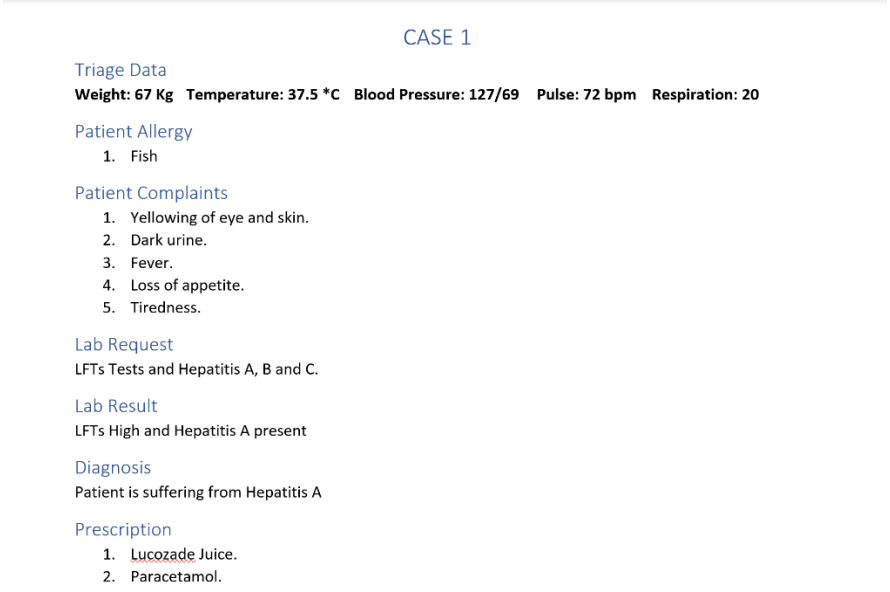
   The peers were satisfied with the usage of the system, since most of the content was easy to use and if they made any error then a help message guided them on the next step.

3. **Relevance to Use:** The system should be of relevance to the user. The professional was very satisfied with the content at each department in the system and approved that it was sufficient for the activities to be carried out successfully.

4.2.5 System Testing

The system testing entailed checking the operations of the integrated system to ensure that it was working as expected and that it met its previously defined functional requirements. Once the integration was complete and tested, the operation of the system as a single unit was then tested to ensure that it did what was expected of it.

This involved using the system by feeding the necessary data to ensure that it acted on the data as planned, The defects identified were dealt with appropriately to ensure that the system met its functional requirements. Some of the test data that was used are in the figures below:



CASE 1

Triage Data

Weight: 67 Kg Temperature: 37.5 *C Blood Pressure: 127/69 Pulse: 72 bpm Respiration: 20

Patient Allergy

1. Fish

Patient Complaints

1. Yellowing of eye and skin.
2. Dark urine.
3. Fever.
4. Loss of appetite.
5. Tiredness.

Lab Request

LFTs Tests and Hepatitis A, B and C.

Lab Result

LFTs High and Hepatitis A present

Diagnosis

Patient is suffering from Hepatitis A

Prescription

1. Lucozade Juice.
2. Paracetamol.

CASE 2

Triage Data

Weight: 58 Kg Temperature: 40 *C Blood Pressure: 140/92 Pulse: 79 bpm Respiration: 20

Patient Allergy

1. Sulphur

Patient Complaints

1. Fever.
2. Sore throat.
3. Swollen Tonsils

Lab Request

Full Haemogram

Lab Result

Neutrophils – High,
other white blood cells normal,
RBC – normal,
Platelets – normal.

Diagnosis

Patient is suffering from bacterial infection.

Prescription

1. Azinthromycin 500mg 1 tablet per day.
2. Cipladon 1000mg soluble 3 tables per day.

CHAPTER 5: CONCLUSION

5.1 Achievements

A web-based hospital system was developed that enabled generation of unique patient Id and use that Id generated to obtain the records of the patients from different health providers.

1. Unique Patient Id generation using either Birth notification number, Birth certificate number, National ID or Passport Number. This numbers are concatenated, and a hash is generated which is used now to create the QR code.
2. The Unique Patient Id is scanned, and it retrieves the medical records from the hospital holding the latest record when the patient gives the health worker the consent. This was achieved with use of two Virtual Machines to simulate Master and Hospital B. A network was created to make host pc with Hospital A be able to ping the virtual machines, virtual machines to be able to ping each other and virtual machines to be able to ping the host pc. It was done by use of host-only adapters in virtual box and statics IPs were given to host pc, virtual machine with Master and virtual machine with Hospital B. Finally, RESTful Web Services were used to make the systems in different computers communicate.
3. The accuracy was measured based on the consistency of the records that are received from other hospitals, that is, records that are transferred from one hospital to another, remains the same on arrival to the receiving hospital.

5.2 Limitations

The following are main constraints related to the project:

1. **Schedule:** Some system features could not be carried out since they were not part of the schedule. These features could be very beneficial to the overall project, they include, radiology tests and attaching lab result documents.
2. **Scope:** The scope of the system was limited and focused mainly on the outpatient medical records of patients.
3. **Budget:** The budget allocated was not inclusive of certain costs. Therefore, the demonstration of the master and other hospital was given using virtual machine and not with physical computers.

5.3 Recommendations

The following recommendations were suggested to the system:

1. Add updates to manage the patient records that needs editing.
2. Improve the consent to be via biometrics rather than being sent SMS.
3. Perform analytics on the patients' medical history and send SMS alert if a certain pattern of symptoms is found, which could lead to a possible disease.

5.4 Conclusion

The major objectives of the system were met which included:

1. Generation of Unique Patient ID, that will be used to track the patient.
2. Using the Unique Patient ID, the medical records of the patients can be obtained from the health provider holding the latest record.
3. The records that are received are same as the once which the hospital they were retrieved from.

Appendix A

References

Anon, (2015). ***System Analysis and Design*** *(Revision 8)*.
[online] Available at: https://www.athabascau.ca/syllabi/comp/comp361.php

CodePlex Archive. (2018).
CodePlex Archive. [online] Available at: https://archive.codeplex.com/?p=fgiqcare

DAILY NATION (2019). ***KNH Doctor open skull of wrong patient***. [online] Available at: https://www.nation.co.ke/news/KNH-doctors-perform-surgery-on-wrong-patient/1056-4325292-iyudmvz/index.html

DHIS2 User guide. (2018).
[eBook] Available at: https://docs.dhis2.org/master/en/user/dhis2_user_manual_en.pdf

EXPLORING 10 YEARS OF HEALTH SERVICE COST AND USE IN KENYA. (2015).
[eBook]Nairobi: Ministry Of Health, pp.1-2.
Available at https://www.healthpolicyproject.com/pubs/746_KHHEUSSummaryBrief.pdf

Health Sector Human Resource Strategy 2014-2018. (2014).
[eBook] Nairobi: Ministry of health, p.14.
Available at: http://www.health.go.ke/wp-content/uploads/2016/04/Kenya-HRH-Strategy-2014-2018.pdf

Lyne, M. (2009). ***What Is A QR Code And Why Do You Need One? - Search Engine Land***.
[online] Search Engine Land. Available at: https://searchengineland.com/what-is-a-qr-code-and-why-do-you-need-one-27588

Magnum. (2017). ***The Best Hospital Information Management System in Kenya***.
[online] Available at: http://magnumdigitalke.com/hospital-information-management-system-kenya/3

Mwangi, C. (2013). ***Computerization of the Kenyan health care record***.
[eBook] Caroline Mwangi.
Available at: https://www.theseus.fi/bitstream/handle/10024/56572/sherry%20thesis.pdf?sequence=1

Palgon, G. (2019). ***The History of Heath Information Management - From Then to Now | Liaison Technologies***. [online] Liaison Technologies.
Available at: https://www.liaison.com/blog/2017/05/02/history-heath-information-management-now/

SearchHealthIT. (2018). ***What is ICD-10 (International Classification of Diseases, Tenth Revision)? - Definition from WhatIs.com*.**
[online] Available at: https://searchhealthit.techtarget.com/definition/ICD-10

Unitag. (2011). ***What is a QR Code ?*.**
[online] Available at: https://www.unitag.io/qrcode/what-is-a-qrcode
Universal Health 2030. (2019). *KENYA: **Hospitals seek to ditch paper records for electronic data - Universal Health 2030***.
[online] Available at: http://universalhealth2030.org/2016/06/29/kenya-hospitals-seek-ditch-paper-records-electronic-data/
Wiki.openmrs.org. (2016). ***KenyaEMR Distribution - Documentation - OpenMRS Wiki*.**
[online] Available at: https://wiki.openmrs.org/display/docs/KenyaEMR+Distribution
Worldometers.info. (2019). ***Kenya Population (2019) - Worldometers***.
[online] Available at: http://www.worldometers.info/world-population/kenya-population/

Appendix B

User Manual

It contains all essential information for the user to make full use of the system. The manual will contain patient registration, starting case for patient, consent SMS, regeneration of unique patient ID, triage waiting list, triage details entry, prognosis details entry, lab request entry, viewing patients history, lab result entry, diagnosis entry, printing prescription and finally patient record track in master.

System Homepage

The figure below shows the default page they will be seen when the application is opened. It contains hospitals latest news feeds and overall information on how system works.

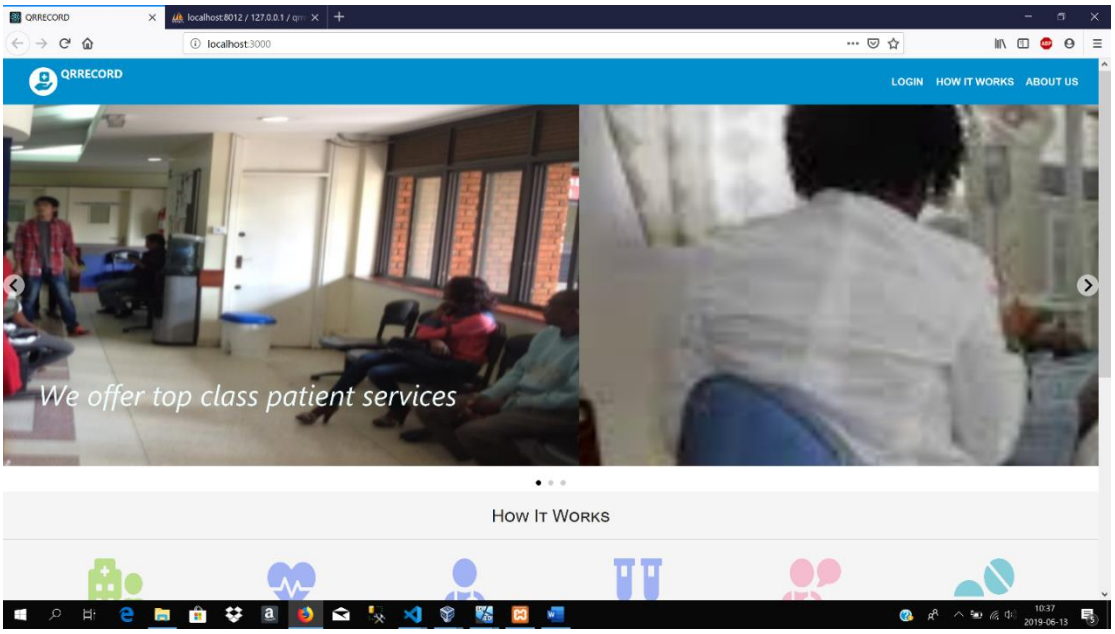


Figure 21: System Homepage

Patient Registration Page

Here all the required details for the patient will be filled by the health worker at reception and register button will be pressed to generate a unique patient ID for the patient, that will be printed and given to him or her.

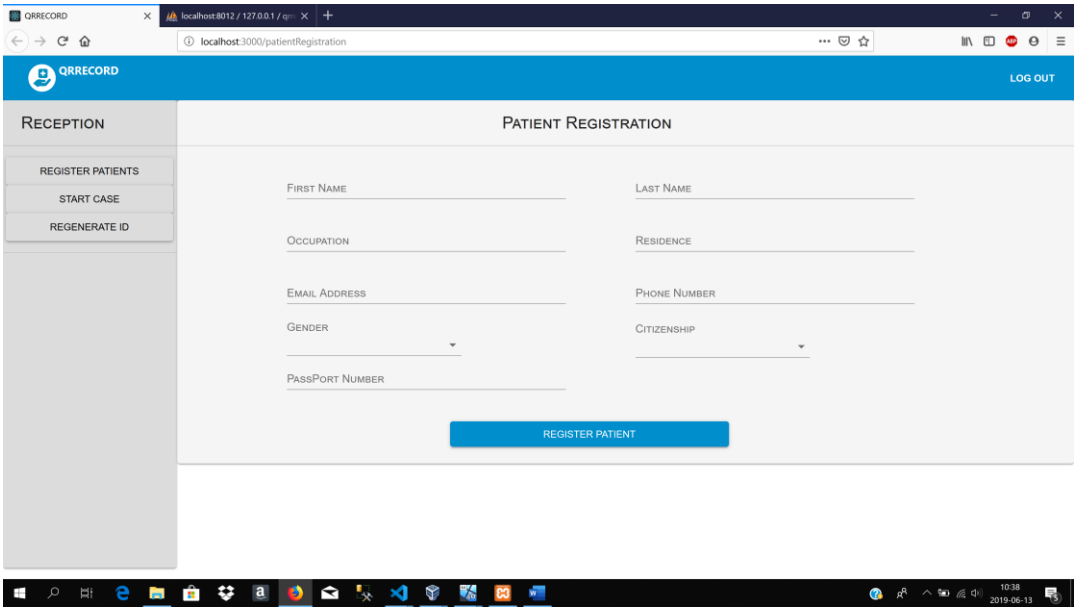


Figure 22: Patient registration

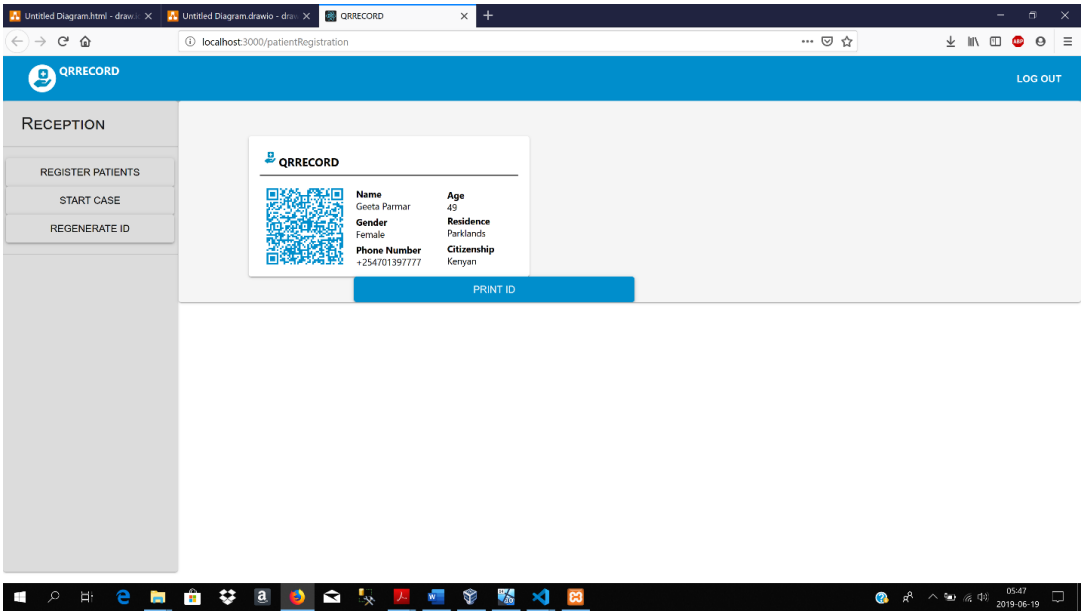


Figure 23: Patient unique ID

Patient Start Case Page

Once the patient has been successfully registered, the health worker at reception can start a case for the patient by scanning the patients unique ID. This will send a One Time Generated password to the patients mobile phone via SMS and the patient will be required to share the password with the reception if he/she agrees their medical records to be retrieved from a given hospital and wants to start case.

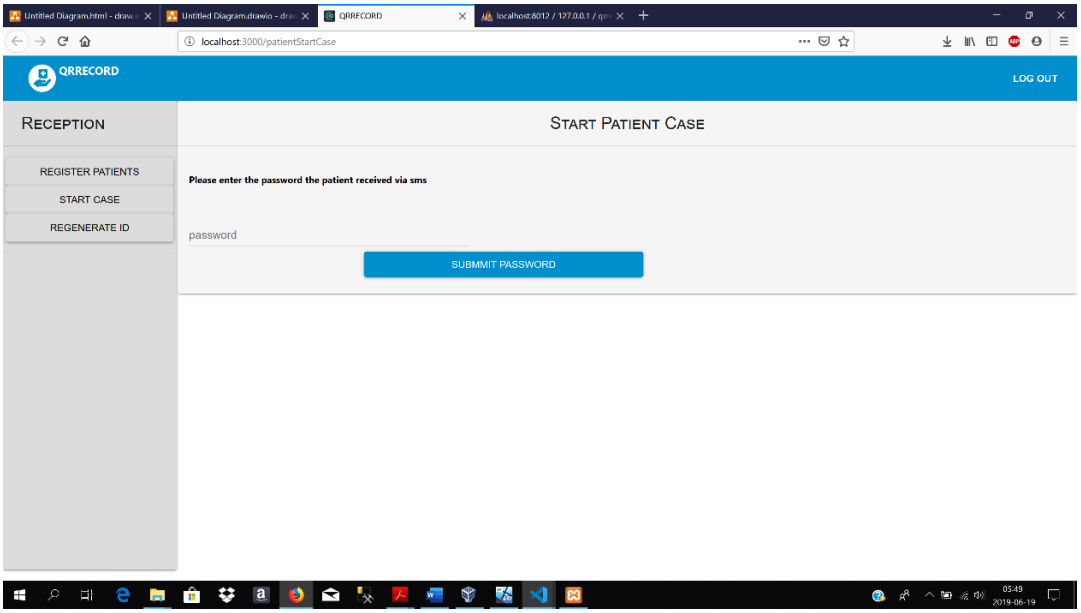


Figure 24: Consent required while starting case

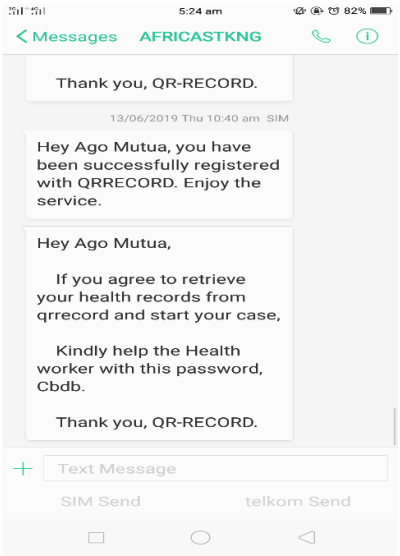


Figure 25: Password obtained via SMS

Regeneration of Patient Unique ID

In the case where a patient loses his/her unique patient ID, then the reception can regenerate the same ID again by entering the required information as shown in the image below and press the regenerate patient ID button. Finally, reception can print the ID for the patient.

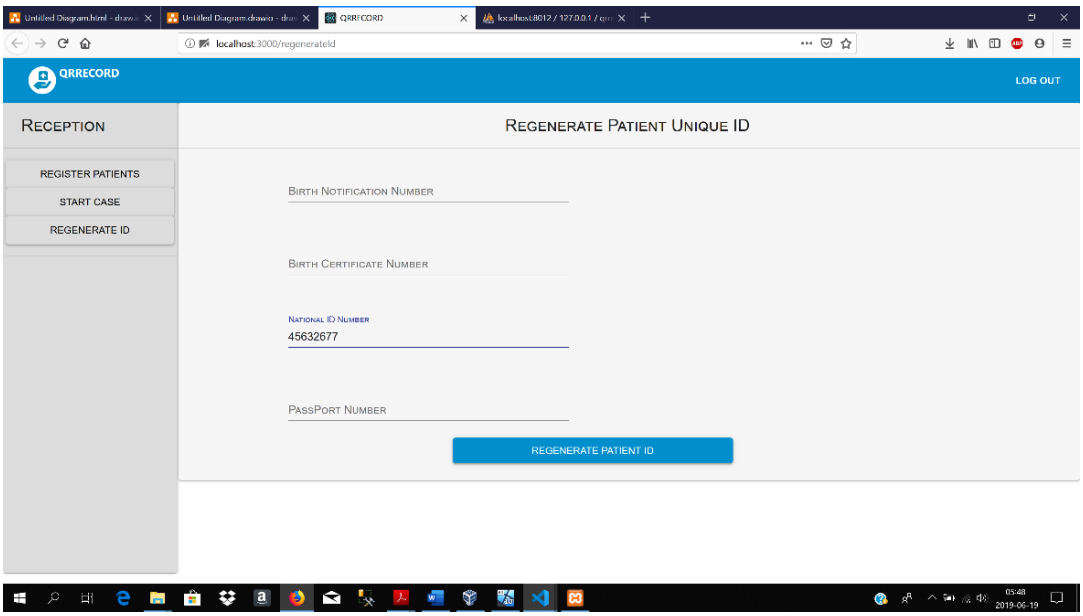


Figure 26: Regenerate patient unique ID

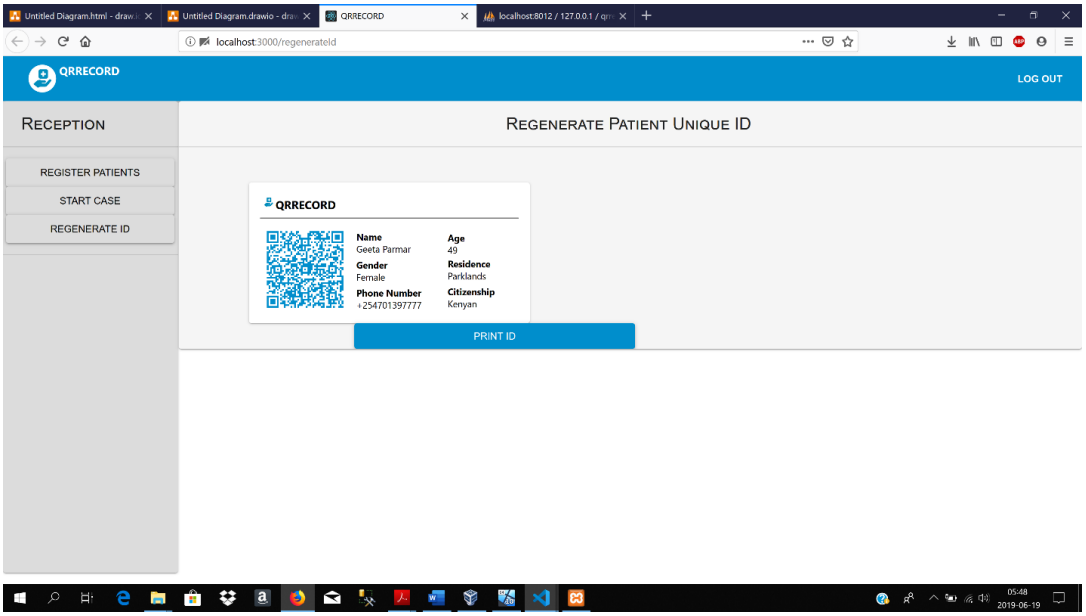


Figure 27: Regenerated ID ready to print

Triage Waiting List Page

Once the case has successfully started, the patient will appear in the triages waiting list as shown in the figure below. The triage nurse will scan the patients unique ID to enter the triage details by pressing the scan patient ID button.

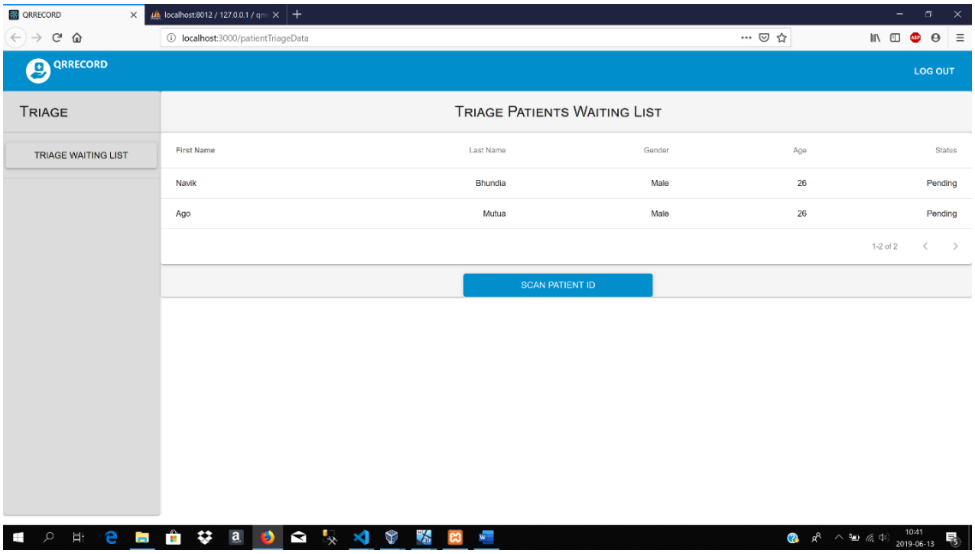


Figure 28: Triage waiting list

Triage Vitals Data

Once the patients unique ID is scanned, the triage nurse will enter five vitals of the patients and submit them by pressing the submit button.

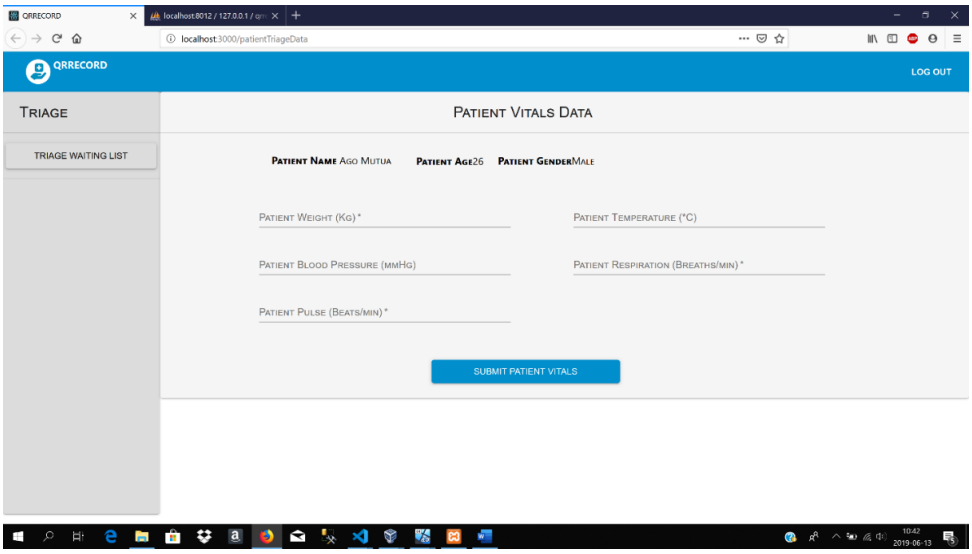


Figure 29: Triage vital data entry

Patient Medical History

Once the triage details have been submitted successfully, the doctor can view patients history by pressing the view patient history in the side bar. The history will be arranged, starting from the latest history to the oldest history.

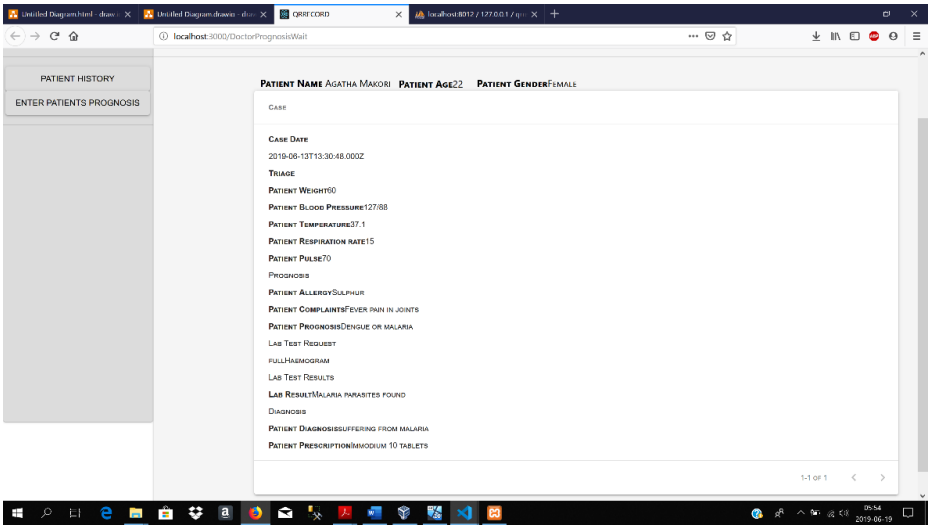


Figure 30:Patient medical history

Doctors Prognosis Details Page

Once the doctor has seen the history, he/she can now enter the required fields, as well as, request for a lab test if there is a need for lab test. The doctor finally submits the prognosis by pressing the submit prognosis button.

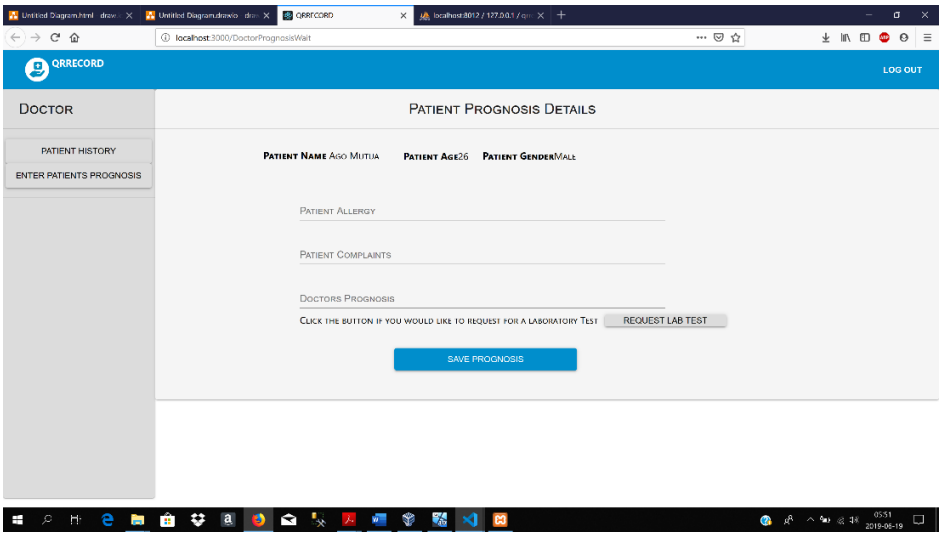


Figure 31: Prognosis details page

Lab Request Form

If there are any lab test to be conducted on the patient, the doctor enters the required tests in the form shown below and presses submit lab test button so that it can be viewed at laboratory.

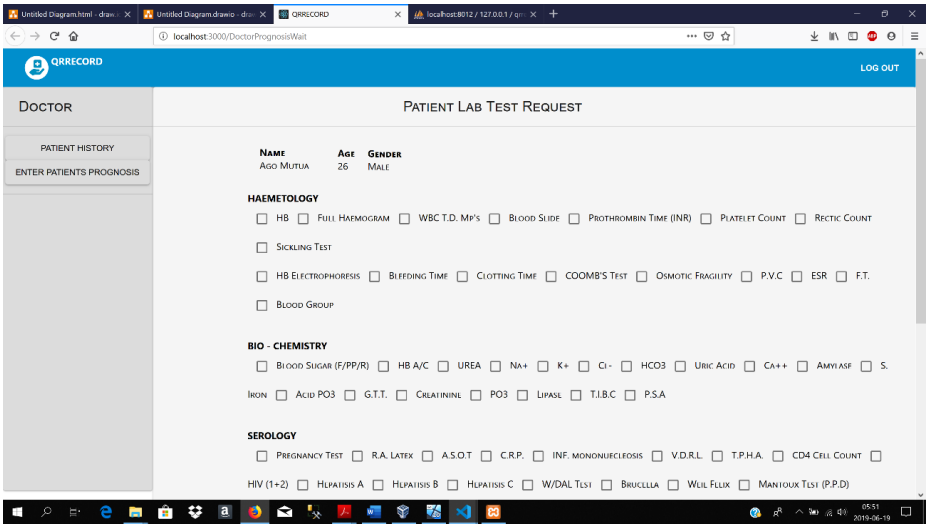


Figure 32: Lab request

Lab Results Form

Once a lab request has been made, the lab technician can scan the patients unique ID to see the lab test that are to be conducted and print QR code of that patient to stick to the vacutainer containing the specimen. Once tests are complete the lab technician scans the QR on the vacutainer and enters the required details about the tests and presses submit lab test result button.

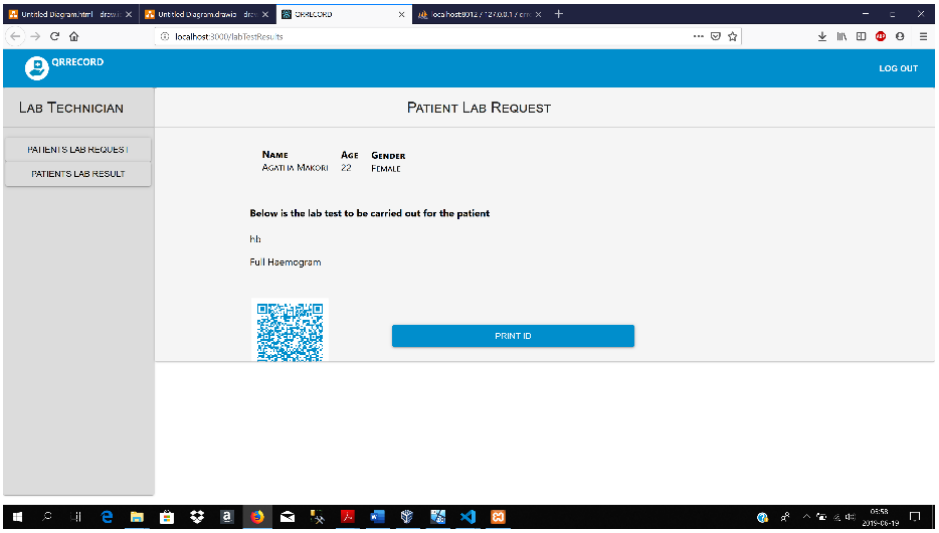


Figure 33: View lab test to be conducted

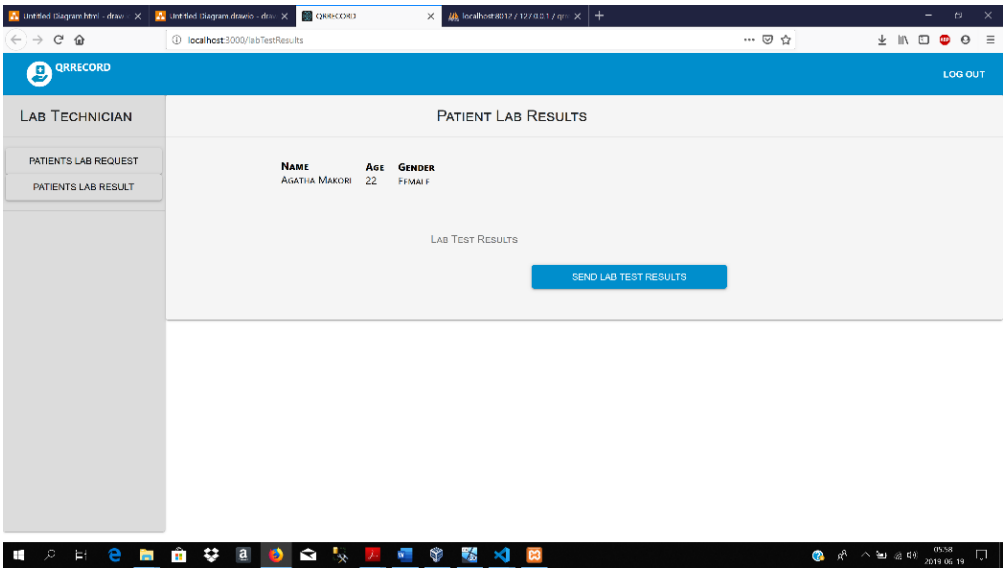


Figure 34: Lab results

Doctors Diagnosis Details

Once the lab test results are submitted by the lab technician, the doctor can scan the patients unique ID and view the patients history to see the lab test results. He/she can then enter the required diagnosis details and press submit diagnosis to submit the diagnosis data.

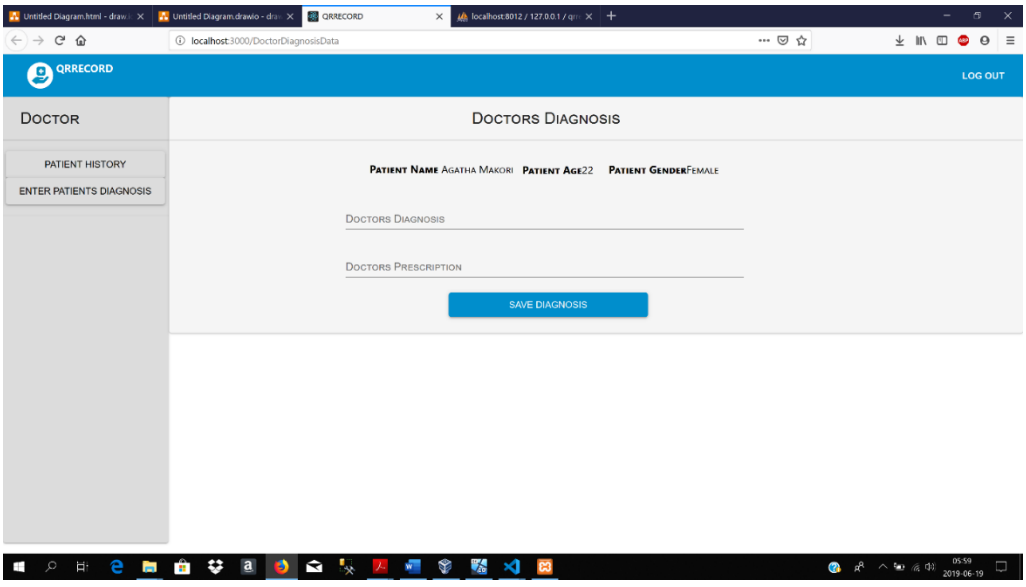


Figure 35: Doctor diagnosis details

Prescription

Once the diagnosis data has been submitted, the pharmacist can scan the patient unique ID to see the prescribed medicine and press the medicine cleared button to finish the case for the patient.

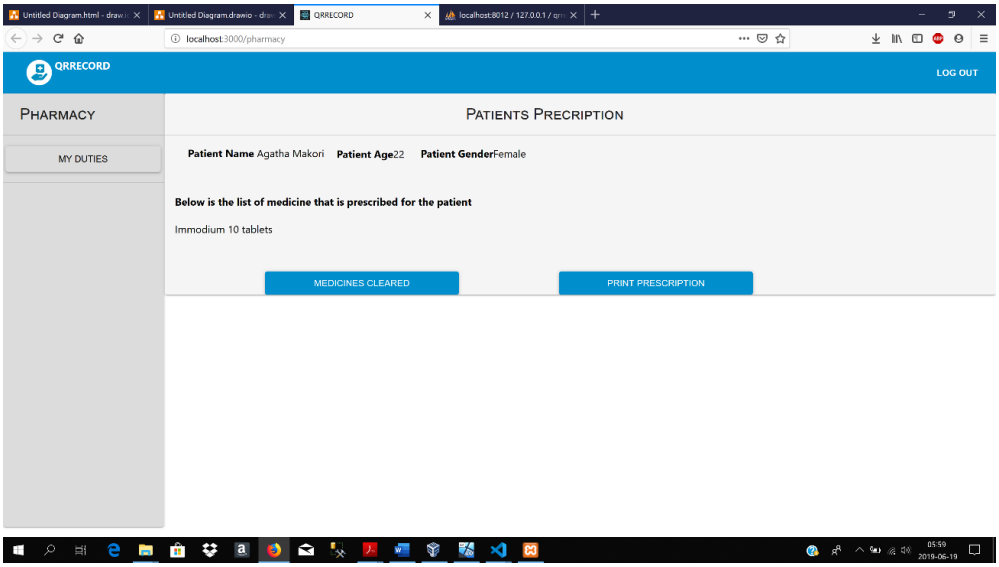


Figure 36: Patients prescription

Master Patient Record Track

Once the case has been successfully completed, the Master is updated with the Name of the hospital holding the patients latest medical record and the timestamp at which the update was made. It is shown in the figure below.

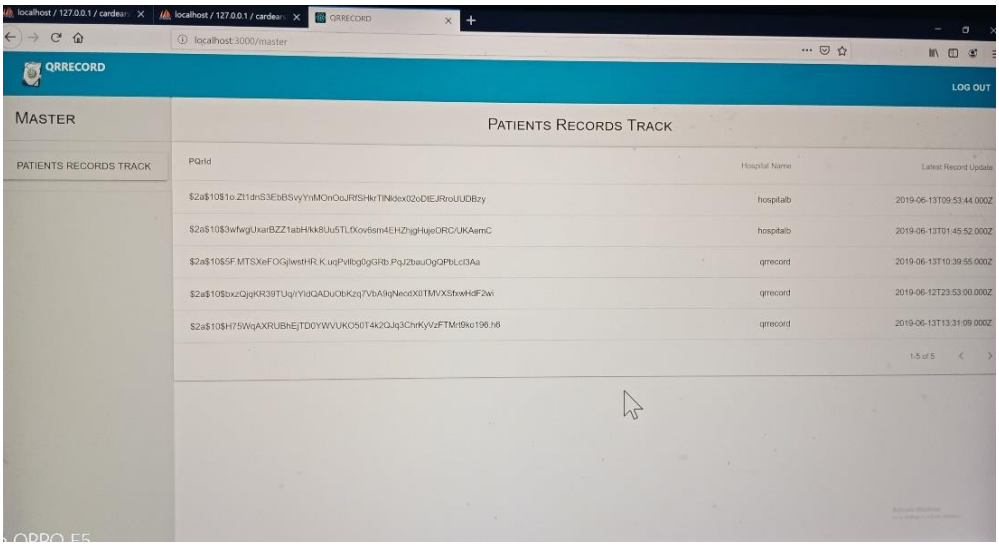


Figure 37: Patient record track at Master

Appendix C

Sample Code

Sample Code for Medical history retrieval

This contains code for Medical history exchange between one hospital and Master, and the hospital with the most updated history.

```
patientStartCase.getPatientById = function getPatientById(patientData, result){

  let patientUniqueId = patientData.patientUniqueId;
  console.log(patientUniqueId);
  console.log("at scanning", password2send);
  console.log(patientData.password);
  if(patientData.password === password2send){
  var sql = `SELECT * FROM patientsregdetails WHERE PQrId = '${patientUniqueId}'`;
  con.query(sql, function(err, res){
    if(err){
      console.log("error: ", err);
      result(err, null);
    }else{
      console.log("Patient ni mimi: ", res)

      if(res.length === 0){
      console.log("axios:",patientUniqueId);
      axios({
        method: 'post',
        url: 'http://192.168.1.12:3002/api/getPatientMaster',
        data: {
          "patientUniqueId": patientUniqueId,
        }
      }).then((response) => {
        console.log("cheking", response);
        var newdata = response;
        if(newdata.data[0].HospitalName === 'hospitalb'){

          axios({
            method: 'post',
            url: 'http://192.168.1.15:3001/api/getPatientFromHosp',
            data: {
```

```
                "patientUniqueId": patientUniqueId,
            }
        }).then((response1) => {

            console.log('we got frm 2nd hosp', response1);
            var data2hosp = response1;

            var sql = `INSERT INTO patientsregdetails
                        SET PQrId = '${data2hosp.data[0].PQrId}',
                    PFirstName = '${data2hosp.data[0].PFirstName}',
                    PlastName = '${data2hosp.data[0].PLastName}',
                PBirthNotiNum = '${data2hosp.data[0].PBirthNotiNum}',
                PBirthCertNum = '${data2hosp.data[0].PBirthCertNum}',
                PNationalIdNum = '${data2hosp.data[0].PNationalIdNum}',
                PNHIFNum = '${data2hosp.data[0].PNHIFNum}',
                    PAge = '${data2hosp.data[0].PAge}',
                    PGender = '${data2hosp.data[0].PGender}',
                    PPhoneNum = '${data2hosp.data[0].PPhoneNum}',
                PEmailAddress = '${data2hosp.data[0].PEmailAddress}',
                PCitizenship = '${data2hosp.data[0].PCitizenship}',
            PIntPassportNum = '${data2hosp.data[0].PIntPassportNum}',
                POccupation = '${data2hosp.data[0].POccupation}',
                PResidence = '${data2hosp.data[0].PResidence}',
                PDateTime = '${data2hosp.data[0].PDateTime}'`;
                con.query(sql,function(err,res){
                        if(err){
                            console.log("error: ", err);
                            result(err, null);
                        }else{
                        var sql1 = `INSERT INTO pcasetrack
                                SET
    PCaseNumCount = '${data2hosp.data[0].PCaseNumCount}',
                            PQrId = '${data2hosp.data[0].PQrId}',
                        StaffId = 123,
                        HospitalId = 0,
                PTriageStatus = '${data2hosp.data[0].PTriageStatus}',
        PDrPrognosisStatus = '${data2hosp.data[0].PDrPrognosisStatus}',
            PDrLabRequst = '${data2hosp.data[0].PDrLabRequst}',
                PLabResult = '${data2hosp.data[0].PLabResult}',
        PDrDiagnosisStatus = '${data2hosp.data[0].PDrDiagnosisStatus}',
```

```
                PPharmacyStatus = '${data2hosp.data[0].PPharmacyStatus}',
                FullCaseStatus = '${data2hosp.data[0].FullCaseStatus}',
                CaseCompletion = '${data2hosp.data[0].CaseCompletion}'`;

                        con.query(sql1,function(err,res){
                            if(err){
                                console.log("error: ", err);
                                result(err, null);
                                }else{
                    axios({
                     method: 'post',
                     url: 'http://192.168.1.15:3001/api/getPatientHistory',
                     data: {
                            "patientUniqueId": patientUniqueId,
                           }
                     }).then((response2) =>{

                        if(response2.length === 0){
var sql2 = `SELECT * FROM patientsregdetails WHERE PQrId = '${patientUniqueId}'`;
                            con.query(sql2,function(err,res){
                                if(err){
                                    console.log("error: ", err);
                                    result(err, null);
                                  }else{
                                    console.log("Patient: ", res)
                                    result(null,res);                                                    }
                                });
            }else{
                transPatientHist = response2;

                for(let i=0; i<transPatientHist.data.length; i++){
                  console.log("najaribiwa",transPatientHist.data.length);
                      var sql3 = `INSERT INTO triagedetails
                              SET
                 TriagePCount = '${transPatientHist.data[i].TriagePCount}',
                 PQrId = '${transPatientHist.data[i].PQrId}',
                 StaffId = '${transPatientHist.data[i].StaffId}',
                 PWeight = '${transPatientHist.data[i].PWeight}',
              PBloodPressure = '${transPatientHist.data[i].PBloodPressure}',
              PTemperature = '${transPatientHist.data[i].PTemperature}',
```

```
            PRespiration = '${transPatientHist.data[i].PRespiration}',
             PPulse = '${transPatientHist.data[i].PPulse}',
             TStatus = '${transPatientHist.data[i].TStatus}',
             TDateTime = '${transPatientHist.data[i].TDateTime}'
              `;
              con.query(sql3,function(err,res){
                  if(err){
                      console.log("error: ", err);
                      result(err, null);
                  }else{
                   var sql4 = `INSERT INTO drprognosisdetails
          SET
          DrPCount = '${transPatientHist.data[i].DrPCount}',
          PQrId = '${transPatientHist.data[i].PQrId}',
          StaffId = '${transPatientHist.data[i].StaffId}',
          TriagePCount = '${transPatientHist.data[i].TriagePCount}',
          PAllergy ='${transPatientHist.data[i].PAllergy}',
          PComplaints = '${transPatientHist.data[i].PComplaints}',
          DrPrognosis = '${transPatientHist.data[i].DrPrognosis}',
          PDateTime = '${transPatientHist.data[i].PDateTime}'
           `;
                                                    con.query(sql4,function(err,res){
      if(err){
        console.log("error: ", err);
        result(err, null);
       }else{
          console.log(data2hosp.data[0].PDrLabRequst)
           var sql5 = `INSERT INTO labtestrequest
                    SET
        labTestRequestCount = '${transPatientHist.data[i].labTestRequestCount}',
    StaffId = '${transPatientHist.data[i].StaffId}',
     PQrId = '${transPatientHist.data[i].PQrId}',
   TriagePCount = '${transPatientHist.data[i].TriagePCount}',
  labRequestStatus = '${transPatientHist.data[i].labRequestStatus}',
        hb = '${transPatientHist.data[i].hb}',
 fullHaemogram = '${transPatientHist.data[i].fullHaemogram}',
   WBC = '${transPatientHist.data[i].WBC}',
  bloodSlide = '${transPatientHist.data[i].bloodSlide}',
 prothrombin = '${transPatientHist.data[i].prothrombin}',
plateletCount = '${transPatientHist.data[i].plateletCount}',
```

```
 recticCount = '${transPatientHist.data[i].recticCount}',
  sicklingTest = '${transPatientHist.data[i].sicklingTest}',
hbElectrophoresis = '${transPatientHist.data[i].hbElectrophoresis}',
 bleedingTime = '${transPatientHist.data[i].bleedingTime}',
  clottingTime = '${transPatientHist.data[i].clottingTime}',
 coombsTest = '${transPatientHist.data[i].coombsTest}',
 osmoticFragility = '${transPatientHist.data[i].osmoticFragility}',
 pvc = '${transPatientHist.data[i].pvc}',
 esr = '${transPatientHist.data[i].esr }',
  ft = '${transPatientHist.data[i].ft}',
  bloodGroup = '${transPatientHist.data[i].bloodGroup}',
  bloodSugar = '${transPatientHist.data[i].bloodSugar}',
  hbac = '${transPatientHist.data[i].hbac}',
  urea = '${transPatientHist.data[i].urea}',
  na = '${transPatientHist.data[i].na}',
  k = '${transPatientHist.data[i].k}',
  cl = '${transPatientHist.data[i].cl}',
  hco = '${transPatientHist.data[i].hco}',
  uricAcid = '${transPatientHist.data[i].uricAcid}',
  ca = '${transPatientHist.data[i].ca}',
  amylase = '${transPatientHist.data[i].amylase}',
  siron = '${transPatientHist.data[i].siron}',
  acidpo = '${transPatientHist.data[i].acidpo}',
  gtt = '${transPatientHist.data[i].gtt}',
  creatinine = '${transPatientHist.data[i].creatinine}',
  po = '${transPatientHist.data[i].po}',
  lipase = '${transPatientHist.data[i].lipase}',
  tibc = '${transPatientHist.data[i].tibc}',
  psa = '${transPatientHist.data[i].psa}',
  pregnancyTest = '${transPatientHist.data[i].pregnancyTest}',
  raLatex = '${transPatientHist.data[i].raLatex}',
  asot = '${transPatientHist.data[i].asot}',
  crp = '${transPatientHist.data[i].crp}',
  mononeucleosis = '${transPatientHist.data[i].mononeucleosis}',
  vdrl = '${transPatientHist.data[i].vdrl}',
  tpha = '${transPatientHist.data[i].tpha}',
  cdCellcount = '${transPatientHist.data[i].cdCellcount}',
  hiv12 = '${transPatientHist.data[i].hiv12}',
  hepatisisA = '${transPatientHist.data[i].hepatisisA}',
  hepatisisB = '${transPatientHist.data[i].hepatisisB}',
```

```
 hepatisisC = '${transPatientHist.data[i].hepatisisC}',
 widalTest = '${transPatientHist.data[i].widalTest}',
 brucella = '${transPatientHist.data[i].brucella}',
 weilFelix = '${transPatientHist.data[i].weilFelix}',
 mantouxTest = '${transPatientHist.data[i].mantouxTest}',
 lipidProfile = '${transPatientHist.data[i].lipidProfile}',
 renalProfile = '${transPatientHist.data[i].renalProfile}',
 liverProfile = '${transPatientHist.data[i].liverProfile}',
 coagulationProfile = '${transPatientHist.data[i].coagulationProfile}',
 cardiacEnzymes = '${transPatientHist.data[i].cardiacEnzymes}',
 ckmb = '${transPatientHist.data[i].ckmb}',
 tropinin = '${transPatientHist.data[i].tropinin}',
            thyroidProfile ='${transPatientHist.data[i].thyroidProfile}',
tshonly = ' transPatientHist ${.data[i].tshonly}',
executiveCheck = '${transPatientHist.data[i].executiveCheck}',
urineRm = '${transPatientHist.data[i].urineRm}',
microAlbumin = '${transPatientHist.data[i].microAlbumin}',
bileSalt = '${transPatientHist.data[i].bileSalt}',
benceJohn = '${transPatientHist.data[i].benceJohn}',
stoolRm = '${transPatientHist.data[i].stoolRm}',
occultBlood = '${transPatientHist.data[i].occultBlood}',
rotaVirus = '${transPatientHist.data[i].rotaVirus}',
cryptosporidium = '${transPatientHist.data[i].cryptosporidium}',
sputumAfb = '${transPatientHist.data[i].sputumAfb}',
sputumTb = '${transPatientHist.data[i].sputumTb}',
hvsRm = '${transPatientHist.data[i].hvsRm}',
urethralSwab = '${transPatientHist.data[i].urethralSwab}',
csfRm = '${transPatientHist.data[i].csfRm}',
semenRm = '${transPatientHist.data[i].semenRm}',
kohPreparation = '${transPatientHist.data[i].kohPreparation}',
fungalCulture = '${transPatientHist.data[i].fungalCulture}',
bloodCulture = '${transPatientHist.data[i].bloodCulture}',
cs = '${transPatientHist.data[i].cs}',
lbDateTime = '${transPatientHist.data[i].lbDateTime}'
    `;                                                  con.query(sql5,function(err,res){
     if(err){
       console.log("error: ", err);
       result(err, null);
      }else{
   var sql6 = `INSERT INTO labtechresults
```

```
                    SET
          labResultCount = '${transPatientHist.data[i].labResultCount}',
          PQrId = '${transPatientHist.data[i].PQrId}',
          StaffId = '${transPatientHist.data[i].StaffId}',
          TriagePCount = '${transPatientHist.data[i].TriagePCount}',
          labTestResult = '${transPatientHist.data[i].labTestResult}',
          LRDateTime = '${transPatientHist.data[i].LRDateTime}'
              `;

         con.query(sql6,function(err,res){
           if(err){
               console.log("error: ", err);
               result(err, null);
           }else{
               var sql7 = `INSERT INTO drdiagnosisdetails
                                                SET
               DrDCount = '${transPatientHist.data[i].DrDCount}',
               PQrId = '${transPatientHist.data[i].PQrId}',
           StaffId = '${transPatientHist.data[i].StaffId}',
        TriagePCount = '${transPatientHist.data[i].TriagePCount}',
         DrDiagnosis = '${transPatientHist.data[i].DrDiagnosis}',
        DrPrescription = '${transPatientHist.data[i].DrPrescription}',
        PrescriptionStatus = '${transPatientHist.data[i].PrescriptionStatus}',
              DDateTime = '${transPatientHist.data[i].DDateTime}'
                 `;
            con.query(sql7,function(err,res){
          if(err){
              console.log("error: ", err);
              result(err, null);
          }else{
             console.log("history has been successfully inserted")
          }
       });
      }
    });
 }
 });
}
});
}
```

```
});
}
var sql8 = `SELECT * FROM patientsregdetails WHERE PQrId = '${patientUniqueId}'`;
        con.query(sql8,function(err,res){
            if(err){
                console.log("error: ", err);
                result(err, null);
            }else{
                console.log("Patient: ", res)
                result(null,res);
} }); } });
                                        }
                                    });

                                }
                            });
                    });
                }
            });
        }
```

Sample Code for SMS Integration

```
smsNotification.patientNotification = function patientNotification(patientDetails, result){

const credentials = {
   apiKey: "97d04247deb8ae757d49b2bdbf277bf81f3e1cf17c427229356afabe0716a59a",
   username: "QRRECORD"
};
const Africastalking = require("africastalking")(credentials);

sms = Africastalking.SMS;

const patientName = patientDetails.firstName + ' ' + patientDetails.lastName;

const number = patientDetails.phoneNumber;

const messageOptions = {
   to: [`${number}`],
```

```
   message: `Hey ${patientName}, you have been successfully registered with QRRECORD. Enjoy the service.`
};

sms.send(messageOptions)
 .then(Response => {
    console.log(Response.data);
   result(null, Response);
 })
 .catch( error =>{
    console.log(error);
 });
};
module.exports = smsNotification;
```