Department of Computer Science & Engineering

INTELLIGENT MEDICAL DATA RECORDING & MANAGEMENT SYSTEM

Authors:

Motasim Foad [14101054]

Israt Sahira Rafa[14101234]

Sheikh Redwan Ahmed Navid [13101089]

Supervisor:

Dr. Md. Ashraful Alam
Assistant Professor
Department of Computer Science & Engineering

December, 2018

Declaration

We, hereby declare that this thesis is based on results we have found ourselves. Materials of work from researchers conducted by others are mentioned in references.

Signature of Supervisor

Dr. Md. Ashraful Alam
Assistant Professor
Department of Computer Science & Engineering
Brac University

Signature of Authors

Motasim Foad [14101054]

Israt Sahira Rafa[14101234]

Sheikh Redwan Ahmed Navid [13101089]

0.1 Abstract

Our country, Bangladesh is fast approaching for digitization but we cannot think of it without digitizing our health-care system as it plays a vital role in development. In context of Bangladesh, management of medical records is still handled in physical documents and it poses a huge problem as these files tends to damage, obliterate, deteriorate and could be stolen or mishandled. So, the purpose of Intelligent Medical Data Recording and Management System (IMDRMS) is to elevate the health-care transactions from an inefficient, error-prone paper-based system to a more reliable, real-time paperless system and allows physicians and hospital authorities to function in a smoother, safer, and more secure manner to retrieve and update the information of any patient with a click of a button which is designed with all the latest web technologies and API's. Basically this project aims to take information of an individual patient by taking input from doctors, pharmacists and hospitals but what made our project intelligent is storing all the data of a patient with one particular patient id whether the patient is registered or not and by that time if that patient get registered all the data that has been stored with that id will automatically transfer information's into that registered patient. So, a patient does not need to accumulate his or her data again. Bangladesh is planning to improve health-care sector but the process is steadily progressing. To boost up that progress IMDRMS can play a prominent role in medical sector and could be one stop point for patient, doctor, dispensaries and diagnostic center.

Keywords: Medical data, EMR, Intelligent system, Digital prescription, Query language

0.2 ACKNOWLEDGEMENT

First of all, we would like to show our utmost gratitude towards the Almighty Allah for giving us strength and keeping us in a good health to finish our thesis paper. Secondly, we would thank and show our immense respect to our honorable Supervisor Dr. Md. Ashraful Alam sir, for his enormous contributions, incomparable guidance and tireless support in conducting the research work and preparing this report and obviously to underpin us. His constant involvement and support encouraged us towards the completion of this thesis work. We are really beholden and humbled to have him as our supervisor. We are also thankful towards our parents as well as our beloved friends for their moral support and advice. They helped us a lot with their valuable suggestions and direct or indirect participation which helped us maintaining a better work-flow and achieving our goal. Lastly, we would like to thank specially our very own BRAC University for allocating necessary equipment's with high definition computers, labs for every thesis student and also giving us the opportunity to work under such intellectual professors to conduct this research.

Contents

List of Figures

0.3 CHAPTER 1

0.3.1 INTRODUCTION

This chapter gives a general review of the purpose of this research work and an overview of the entire research. An EMR (Electronic Medical Record) is the systematized collection of patient's health information that electronically-stored in a digital format. Throughout the year EMR has become a very adaptable and most important technology in health-care system in developing countries despite having lack of IT skills for its design strategy, critical, technical and social features of the system that intended to support skeptical users. In addition, reports on medical result errors and erroneous handwritten documents made it more demanding. So our thesis paper conducts the basic idea of EMR system and overall overview of the API's and servers. We used HTML, CSS, JSx, Bootstraps, MDL, ReactJS for front end and to create the backend API we used Apollo client to build UI that will fetch data with GraphQl by the framework GraphCool which turns our backend engine to BAAS.

Our project is designed to re-present real time data that accurately captures the state of the patient at all times. It allows for an entire patient history to be viewed without the need to track down the patient's previous medical record volume and assists in ensuring data is likely to accurate, appropriate and legible. It reduces the chances of data replication as there is only one modifiable file, which means the file is constantly up to date when viewed at a later date and eliminates the issue of lost forms or paperwork. Due to all the information being in a single file, it makes it much more effective when extracting medical data for the examination of possible trends and long term changes in the patient. Moreover, a record of health related issues of an individual can be created, gathered, managed and authorized by an authorized person (could be clinicians or staff) within a particular organization. It has the potential to provide substantial benefits to physicians, clinic practices, and health care organizations. These systems can facilitate workflow and improve the quality of patient care and patient safety.

Despite of those benefits, the usage of EMR sometimes become barrier due to high capital cost and insufficient return on investment for small practices and safety net providers, failure to redesign clinical process and workflow to incorporate the technology systems, lack of skilled resources for implementation and support, concern regarding negative unintended consequences of technology etc. But with the help of modern technologies the platform will transfer into a serverless platform with a simple query system to access all data of a particular patient through individual login id and password that can be accessed by the authority of the organization to make it more users friendly.

0.3.2 1.1Motivation

In this modern era, a health-care system cannot be thought without the modern advancement and implementation of advance API and servers. Though most of the organizations still rely on paper based records for their day to day record keeping and health related activities. But with this traditional method it becomes more challenging at the end as it takes a lot of physical space to keep and store paper based records of registered patients. Another difficulty a patient face is to physically carry out all these paper document of their medical records from one hospital to another. Since all these are paper based so it has high risk to be destroyed in natural calamities, fire outbreak or earthquake and in this kind of situation looking for individual patient records while having many registered patient may also take a longer time compared to doing electronically. For all these occurrences there arises a need to develop a simplified IMDRMS to replace the paper based system. On the top, most of the EMR designed in SAAS (Software as a Service) which is mainly focused on giving software update for end users as on cloud it is always running on the latest version but IMDRMS is replaced by BAAS API because it gives developers fast solution in a stable backend and it also have parse attribute to integrate with computer languages and to cover up all the services that is needed by an application to avoid obsolescing of this important innovation of human.

1.2 Objective

The main objectives of this research are the following:

- To overview patient's prescription to retrieve the current health situation.
- Reduction in health report errors.
- Automated prescription forwarding and tracking.
- Doctor's information and past history will be added up for more clarification.
- Time saving and one stop point for Patient, Doctors, Dispensaries and Diagnostic EMR.

1.3 Thesis Overview

Chapter 1 introduced the motivation and objective of this research.

Chapter 2 consists of the literature review where we have denoted the background study in various API's, servers and query languages used in this thesis.

Chapter 3 is the section where we have included architecture of our proposed model of model with block diagrams and workflow of the proposed system.

In chapter 4, we have described about our implementation of the idea and validation of results to strengthen our claim and the outcomes.

And lastly, in chapter 5 contains the conclusion, future opportunities and planning of this research.

0.4 CHAPTER 2

0.4.1 LITERATURE REVIEW

In this study and related research, the project serving with some software to make the project more efficient and user friendly and it takes aspiration from EMR system. EMR (Electronically Medical Record) that is shared through network-connected, enterprise-wide information systems or other information networks and exchanges. In 2011 study in diabetes care, published in the *New England Journal of Medicine,* found evidence that practices with EMR provided better quality care. According to survey that is conducted by Net Health Team, a majority patient approximately 61.9% patient prefers to work with organization that has implemented EMR system. Bangladesh healthcare tech startup also clustered together to bring telemedicine to facilitate remote service delivery, and communication between patients and physicians (example, Jeeon), Appointment Scheduling to help people schedule appointments with physician (example, Doctor Ola), Preventative healthcare system to educate and helping users keep tabs on various parameters of health with the help of a mobile platform. (example, Rx71, CMED), Emergency response to Ensure delivery of emergency services with a pool of dedicated first responders and donors (e.g Criticalink), Pharmacy Delivery to allow users to order required pharmaceutical products online and have them delivered to their doorstep. (e.g Bhalothakun, Pharma71), Mental and female Health to Provide real time, on demand services such as consultation, advice from experts on socio, psychological, health and legal issues (Example, Maya Apa), Maternity Health to provide consultation, support through helpful content to expecting mothers. (example, Aponjon by DNET), Comprehensive Health Service: A bundle of service for a monthly subscription fee including health content, over the phone consultation, appointment booking, and insurance.(Example, Tonic, by Telenor Health) but our thesis paper tried to bring all the users in a single platform to communicate with each other.

0.4.2 2.1 Building the Conceptual Design for IMDRMS

In this section each of the construct detailed has been strategically selected based on its use in prior research for this project. According to this thesis work, the transition from paper records to electronic records is based in terms of technology-related knowledge, modern web API's, query languages along with behaviors relating to healthcare systems; the following attributes have been selected for this project:

0.4.3 2.1.1 REACT API

React is a JavaScript library for building UI. It is a JavaScript library used for building reusable UI sections. According to React official documentation, the definition of React is a library for building compassable UI's. It engages the making of reusable UI parts, which show data those movements after some time. Heaps of people use React as the V (View) in MVC. React abstracts away the DOM from us, offering a less intricate programming model and better execution.

React can in like manner render on the server using Node, and it can control neighborhood applications using React Native. React realizes one-way open data stream, which lessens the standard and is less requesting to reason about than traditional data definitive.

JSX – JSX is JavaScript dialect structure increase. It isn't vital to use JSX in React advancement, yet it is recommended.

Components – React is about sections. We need to consider everything a section. This will empower us to keep up the code when wearing down greater scale wanders.

Unidirectional data Flow and Flux – React executes one-way data stream which makes it easy to reason about user application. Change is an illustration that helps keeping user data unidirectional.

License– React is approved under the Facebook Inc. Documentation is approved under CC BY 4.0.

Utilization of React

Presently, the primary inquiry emerges before us is the reason one should utilize ReactJS. There are such huge numbers of open-source stages for influencing the front-end to web application improvement less demanding, as Angular. With the front-end world changing regularly, it's difficult to commit time to taking in another structure – particularly when that system could eventually turn into a deadlock. Along these lines, in case we are searching for the following best thing however we are feeling somewhat lost in the structure wilderness, so we chose React as the UI for our website.

ReactJs is only less difficult to get a handle on immediately. The segment based approach, all around characterized lifecycle, and utilization of out and out JavaScript make React exceptionally easy to learn, construct an expert web and versatile applications, and bolster it. React utilizes a unique sentence structure called JSX which enables us to blend HTML with JavaScript. Anybody with an essential past information in programming can without much of a stretch comprehend React while Angular and Ember are alluded to as 'Area particular Language', suggesting that it is hard to learn them. For Reactus simply require fundamental information of CSS and HTML. React can be utilized to make portable applications (React Native). What's more, React is a diehard enthusiast of reusability, which means broad code reusability is upheld. So in the meantime we can make IOS, Android and Web application. React utilizes one-way information authoritative and an application design called Flux controls the stream of information to segments through one control point – the dispatcher. It's less demanding to investigate independent parts of expansive ReactJS applications. React does not offer any idea of an implicit compartment for reliance. ReactJS applications are super simple to test. React perspectives can be dealt with as elements of the state, so we can control with state we go to the ReactJS view and investigate the yield and activated activities, occasions, capacities, and so forth. It is genuinely little and let us pick different apparatuses us need to work with.JSX, JavaScript as a "templating" dialect. React simplifies working with DOM. JavaScript is driving the prominence outlines on GitHub for a considerable length of time

Utilization of ReactJs:

ReactJs is used by many top guns' like Facebook, Instagram, Netflix, Apple, Yahoo, New York Times, WhatsApp, Dropbox and so on. All these websites are ever changing and needs the UI to adjust and reform accordingly.

Components

Some components of ReactJs are plain JavaScript, little API, Props and State, Virtual DOM, Unidirectional Data Flow, Renders DOM, hub and local (iOS, Android), Component based approach gives a chance to make convoluted UI‘s creating littler parts. It is anything but difficult to ”outline” to UI. React effectively updates and render only the correct parts when our information changes. Declarative perspectives make our code more unsurprising and less demanding to investigate. React gives a lightweight virtual DOM, Powerful perspectives without formats, unidirectional information stream and explicit transformation. A React application comprises of Reusable segments. These parts influence code to reuse, testing, and partition of concerns simple. React has state, it handles mapping from contribution to state changes, and it renders parts. In this sense, it does everything that a MVC does.

React Advantages

Utilizations in virtual DOM which is a JavaScript question. This will upgrade applications execution, since JavaScript virtual DOM is faster than the general DOM. Can be used on client and server side and furthermore with various frameworks. Segment and data outlines upgrade clarity, which keeps up greater applications.

React Limitations

Covers simply the view layer of the application, therefore in any case us need to pick distinctive advances to get a whole tooling set for development. After developing front-end the system needs a service and server to conduct the whole procedure so in the back-end there will be an admin and as per service we use Graphcool and GraphQl Baas as a server. Graphcool takes care of storing data and provides a GraphQL endpoint for the proposed application, then it needs to define a schema on Graphcool and it will automatically generate functions to query and mutate data. Lastly, it will store data, integrate with third party authentication provider and serverless functions such as Baas (Backend as a service). On the other hand, GraphQl is a query language that has been used in our application rather using REST API for its spontaneous and quick response from backend where in Rest API it needs to send multiple requests to query the post, comments, and author details.

0.4.4 2.1.2 GraphQl

GraphQl is a data query language which can execute queries and in the meantime provide a system to build a client application. It was developed internally by Facebook in 2012 before being publicly released in 2015 for describing the capabilities and requirements of data models for client-server applications. After that bunch of different companies and startups for instance github, pin interest, credit karma, yelp, mix cloud, startups.co, intuit and many more adopt it. In a simple language, it is designed to build client applications by providing an intuitive and flexible syntax and system for narrating their data requirements and interactions. GraphQl is not a programming language of an arbitrary computation rather a language where application servers take their capabilities and map them to uniform language that graphQl can encodes. GraphQl is based on its design principles consists of hierarchical, product-centric, strong-typing, introspective and client-specified queries which provides a unified interface friendly to product development and a powerful platform for tool-building. First thing to notice is that graphQl queries mirror their response which makes it easy to predict what type of query it will return from a data and that makes it very easy to learn and use. Another important aspect of graphQl is its hierarchical nature where it gives structure of these applications hierarchically for clients to describe data representation. It also defines an application specific type system to ensure every query is synthetically correct and valid before execution. Moreover, a graphQl publishes the capabilities that its clients are allowed to consume and returns exactly what was asked for not more than that. Lastly, GraphQl is introspective as it serves as a powerful platform for building common tools and client software libraries and server libraries like C#, .NET, Clojure, Elixir, Erlang, Go, Groovy, Java, JavaScript, PHP, Python, Scala and Ruby. So GraphQl is a string that is sent to a server to be interpreted and fulfilled, which then returns JSON back to the client fulfilling the above mentioned principles.

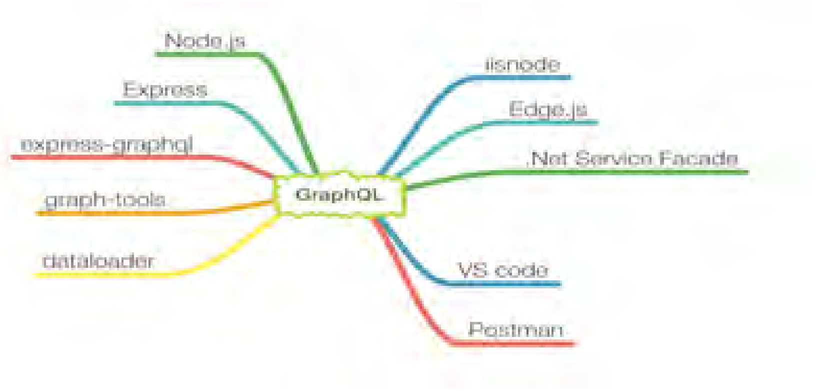


Figure 1: 2. 1 GraphQl variations in services

How GraphQl works:
A GraphQl has three basic steps to follows. These are:

- Query
- Schemas and resolvers
- Execution

Query
A query cannot be done without a type system so it needs a query type to execute a GraphQl. For instance, in our project, if we write a query of doctor or patient or pharmacy or hospital with its given id and email address then simply it will return back the query as it was asked with id and email information. It is notable that the response is almost the same as that of the query.

Schema & Resolvers
The graphQl has actually two core parts one is schema and another one is resolvers which determines how GrpahQl works. The schema is a model of the data that can be fetched through the Graphcool server. It defines what queries

clients are allowed to make, what types of data can be fetched from the server, and what the relationships between these types are.

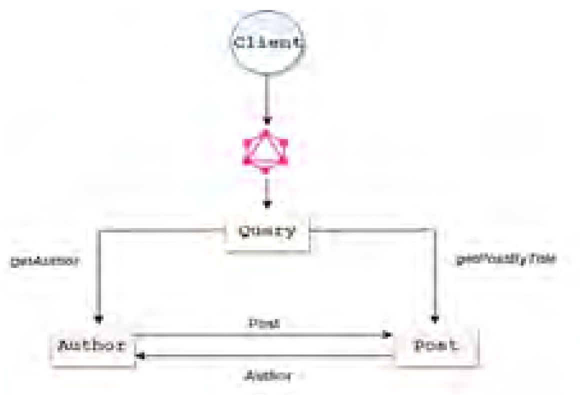


Figure 2: 2. 2 A simple GraphQL schema with three types: Author, Post and Query.

Here in this schema it states three points including author, post and query where query is a kind of entry point to get into the schema where every query needs to starts with one of its field suppose *getAuthor* or *getPost*. So, here in the project, doctor can easily find patient information or prescription that has been updated by doctor. It is likely to REST endpoints but more powerful.

A schema has set of types which completely describe the possible data that can be query on that service so when that query comes in it validates and execute against that schema. Since we already know GraphQl doesn't have any specific language syntax so it uses "GraphQl schema language" to allow us to talk about schemas in a language-agnostic way. The most basic components of a GraphQL schema are object types, which just represent a kind of object you can fetch from your service, and what fields it has like in the query section as mentioned doctor, patient, pharmacy or hospital could be object where id and mail could the field on it which means id and mail address is the only field that can appear whenever user is called in the GraphQl. Another GraphQl object type is arguments; it can be either required or optional and for optional it can define a default value. GraphQl also have enumeration or enum type which validate that any arguments of this type are one of the allowed values and communicate through the type system that a field will always be one of a finite set of values such as if setenum episode in the query that means one of mail address are expected. Every GraphQl may have query but not mutation as mutation is special kind of type. So the schema tells the server what queries clients are allowed to make, and how different types are related, but there is one critical piece of information that it doesn't contain where the data for each type comes from and that's what resolve functions are for.

Another important function of GraphQl procedure is resolve function which specify who the fields in the schema are connected to different backend. It can contain arbitrary code which means a GraphQl can conduct with any kind of backend or even different servers. For instance, the Author type could be stored in a SQL database, while posts are stored in MongoDB (a free and open-source cross-platform document-oriented database program), where each field is backed by a function called the resolver which is provided by the GraphQl server developer. However if a field produces an object value then the query will contain another selection of fields which apply to that object until scalar values are reached. A resolver function receives three arguments:

- Object type: The previous object, which for a field on the root Query type is often not used.
- Argument type: The arguments provided to the field in the GraphQL query.
- Context: A value which is provided to every resolver and holds important contextual information like the currently logged in user, or access to a database.

Yet there are few resolvers who fulfill those arguments such as asynchronous resolver which can use context to access the database to load the data for a user by the given id in the query and return a promise. GraphQl is a type system that determines what to do next like before generating user field it already knew that it will return a user. Since calling user from previous field so it accumulates the object argument. After executing name field it automatically resolve its id and mail field. Moreover, in the human query there list of mail address which also needs to load to execute the whole query so it will wait to load all those list of mail address and return as a enum values.

Execution

Execution is the last step where it couple schema and resolve function together to execute the query and get the desired result. Before executing it follows three steps to respond to the query. Firstly, the server pass the string to Ast (Abstrct syntax Tree) to check the syntax error and if found the procedure will stop and return syntax error to the client. Secondly, it checks for whether the query is correct or not through some questions which is automatically resolved by GraphQl and this stage is called validation stage. Lastly, after the validation it will start executing the query and collectively these produce a structure that mirrors the original query which can then be sent (typically as JSON) to the client which requested it.

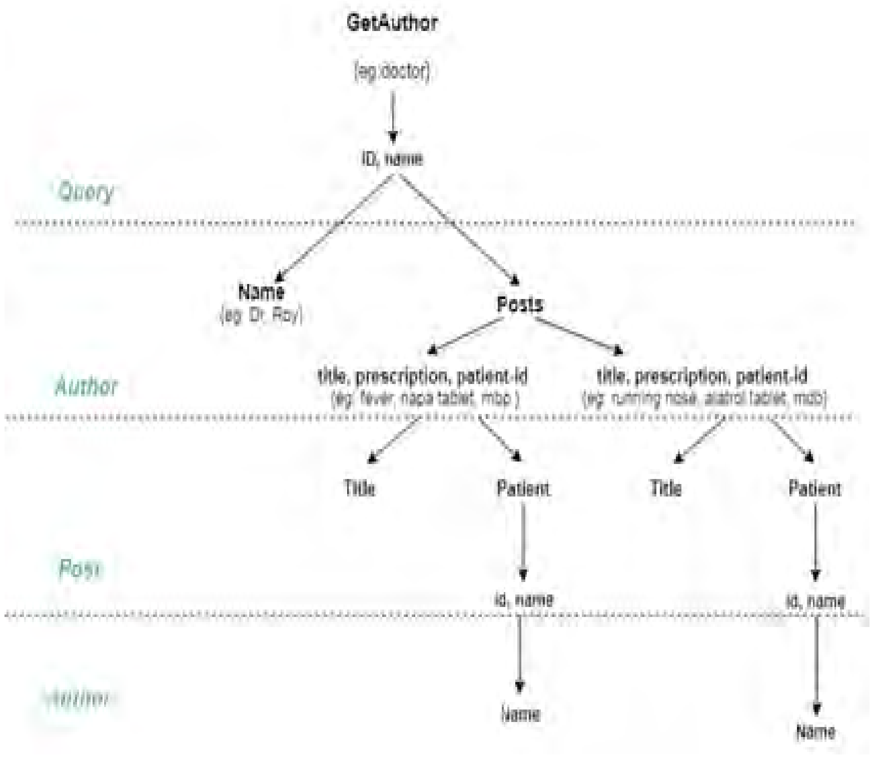

GetAuthor
(eg:doctor)
ID, name
Query
Name
(eg: Dr. Roy)
Posts
Author
title, prescription, patient-id
(eg: fever, napa tablet, mbp)
title, prescription, patient-id
(eg: running nose, alatrol tablet, mdb)
Title
Patient
Title
Patient
Post
id, name
id, name
Name
Name

Figure 3: 2. 3 Execution starts at the top to resolve functions in executed level concurrently

Benefits of GraphQl

There are lots of points that people start using GraphQl over so many protocols such as REST, XML-RPC, and SOAP. Both XML-RPC (mid 90's) and SOAP (2001-2007) are network related protocol which returns in XML and communicate through HTTP but REST (2000-) is bit different as it is a technique for implementing Web services using XML documents plus standard HTTP, using its well-known operations (PUT, GET, POST, DELETE). The service-specific API is created by defining URLs and XML documents that model the data structures and requests/responses required. But after GraphQl being invented REST usability got down and the main reason behind is it needs multiple requests to complete a query where a GraphQl can complete a query with a single request. So basically rest is a collection of endpoints where each end point requires a resource and when it comes to multiple resources it needs to perform multiple round trips to get the data.

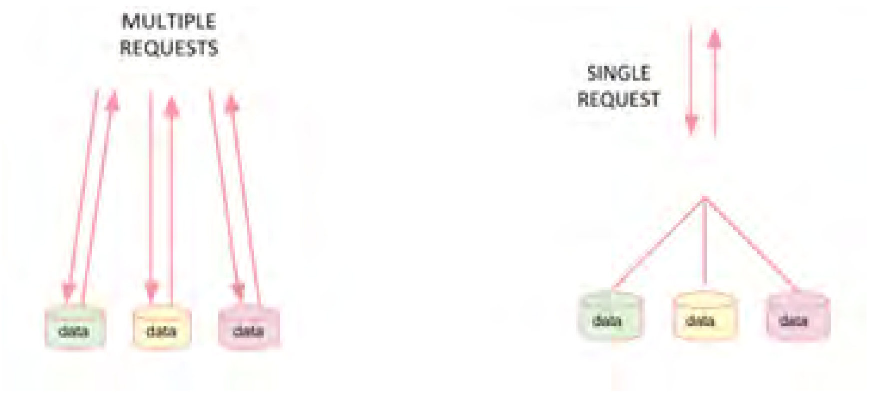


Figure 4: 2. 4 Differences between REST and GraphQl

Another big problem with REST APIs is versioning as multiple versioning means multiple end points which leads to new endpoints or more complexity. But in GraphQL it has an interesting take on that as avoided Versioning and adds new fields without removing the old ones, because we have a graph and we can flexibly grow the graph by adding more nodes. In a REST API, there is no client request language. Clients have no control over what they will get from server. Whereas GraphQL eliminates the need for the server to hardcode the shape or size of the data, and decouples clients from servers which means we can maintain and improve clients separately from servers through client request language. A common drawback of rest API is that it doesn't allow for rapid iteration on the frontend so if any changes made to the user interface there is a high risk that there's more data than before as well as backend needs to adjust with this still through this process it will slow down the ability and kills productivity but with the help of GraphQl the problem is solved for its flexible nature as in GraphQl clients specify exactly what data it requires. Moreover, GraphQl uses strong schema type system that an API is written down in schema definition language which serves as a contract between the client and the server that defines a client's accessibility of data. Once the schema is being defined the frontend and backend engineer do not need to communicate with each other as they aware of the structure over the network.

In GraphQL, you can traverse from the entry point to related data, following relationships defined in the schema, in a single request. In REST, you have to call multiple endpoints to fetch related resources.

0.5 2.1.3 A* Algorithm for searching attributes

One of the most important aspect is searching such as searching prescriptions, searching medical reports, searching user information through user id or phone number or NID and for that our system used A* (aster) search algorithm to acquire the searching procedure.

PSUDOCODE:

```
1. Initialize the open list
2. Initialize the closed list
put the starting node on the open
list (you can leave its **f** at zero)
```

```
3. while the open list is not empty
a) find the node with the least **f** on
the open list, call it "q"
b) pop q off the open list
c) generate q's 8 successors and set their
parents to q
d) for each successor
i) if successor is the goal, stop search
successor.**g** = q.**g** + distance between
Successor and q
successor.**h** = distance from goal to
Successor (This can be done using many
Ways, we will discuss three heuristics-
Manhattan, Diagonal and Euclidean
Heuristics)
successor.**f** = successor.**g** + successor.**h**
ii) if a node with the same position as
successor is in the OPEN list which has a
lower**f** than successor, skip this successor
iii) if a node with the same position as
successor is in the CLOSED list which has
a lower **f** than successor, skip this successor
otherwise, add the node to the open list
end (for loop)
e) push q on the closed list
end (while loop)
```

0.6 2.1.4 BAAS

It is counter-intuitive to think but serverless actually needs a server where it both has server hardware and server process that running in the application. There is a single difference between normal server based applications and serverless such as server based is looking after server hardware, its operating system efficiency along with configuration whereas serverless only requires the computing or coding which takes as input, perform logic and return the requested output with much more concurrency as the developer needed without worrying about container orchestration, OS (Operating System) configuration, or the underlying servers. In around 2012, it was first heard in regard to continuous integration and source control system that is being hosted as a server instead of using companies own server. But it starts gaining popularity the year after Amazon API Gateway introduced AWS Lambda back in 2014. Before serverless there were few more services such as IaaS, SaaS, PaaS which sometime makes confusion but these all three represents each single identity.

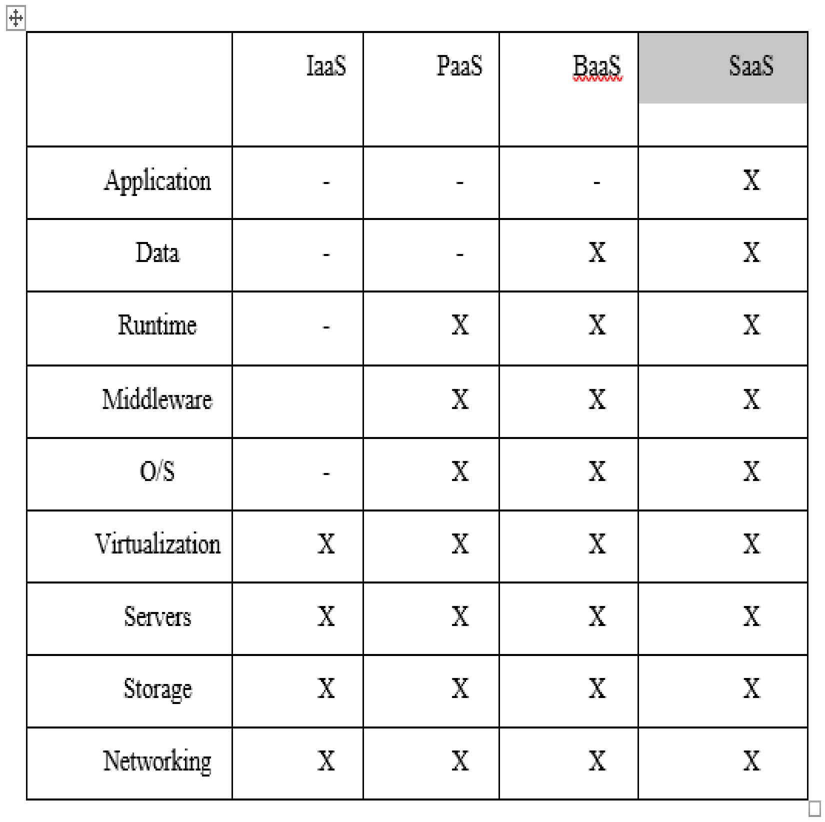
- IaaS: IaaS stands for Infrastructure-as-a-service and one of the main categories of cloud computing. Providers are responsible for providing server, networking, virtualization, hard drive, storage and users for application, runtime, and middleware. Amazon Web Services (AWS), Cisco Metapod, Microsoft Azure, Google Compute Engine (GCE).

- SaaS: Software-as-a-Service uses web to deliver the application through third party vendors that is accessible from client side and it represents the most widely cloud market in cloud computing. The main advantage of SaaS is it doesn't require installing and running application to individual computers since everything is managed by vendors.Google Apps, Salesforce, Workday, Concur, Citrix GoToMeeting, and Cisco WebEx.

- PaaS: Platform-as-a-Service allows developer to develop and customize their application only to focus on writing code without worrying about underlying infrastructure. With this technology, a third party provider can manage virtualization, servers, storage, networking and PaaS software itself. Apprenda is private cloud PaaS for .net and java along with Baidu open platform, Tencent development platform, and Sina development platform.

After these cloud services serverless emerge and falls into two categories Baas (Backend as a service) and Faas (Function as a service) while Baas is a serverless backend here developer only focus on deploying code and also in Faas developers don't need to bother themselves into multithreading or load balancing rather penetrate them into coding.

Table 2.5A brief differences between several cloud services



	IaaS	PaaS	BaaS	SaaS
Application	-	-	-	X
Data	-	-	X	X
Runtime	-	X	X	X
Middleware		X	X	X
O/S	-	X	X	X
Virtualization	X	X	X	X
Servers	X	X	X	X
Storage	X	X	X	X
Networking	X	X	X	X

BaaS is a powerful and cloud based Backend-as-a-Service that provides web and mobile app developers with a way to connect their applications to backend cloud storage and processing while also providing common features such as flexible data storage, user management, push notifications, geolocation, client authentication, social networking integration, performance monitoring and other features according to the mobile users demand from the apps via RESTful API. It helps developers to accelerate software development to simplify API creation and by creating BaaS it can be linked to the application directly to avoid writing handful amount of codes.

Push notification: it is a custom badge, auto incremented and HTML, video link and youtubeurl based service that send notification to single or multi users. For instance, a facebook user will get notified in respect to channels it was subscribed from the user.

Social service: This service is used for all the social media sites such as Facebook, LinkedIn and Twitter to fetch, invite, post, update, refer social friends.

E-mail service: It is used to create and manage templates and configure host e-mail account and send e-mails to one or multiple recipients.

Database file storage: Storage service provides an efficient way to manage JSON documents in NoSQL database on the cloud so it can easily be stored, updated searched. If a Json data is stored in cloud database then it will be registered with a specific id for further use with that specific id.

But with the help of Baas a backend is supposed to be developed within a short period of time as Baas by itself was containing cloud storage, push notification, social media integration and its own infrastructure as well. All these were listed in their own Baas API so they could be added to the software. It also made it possible to prevent developers from dealing with physical app server, database, client-server library, writing of admin control panel, API design and hosting. But Baas has broadened its version than it used to be. With this new product, AWS is enlarging the BaaS and Serverless brand, which changed the previous specification.It actually behaves as a bridge that connecting the backend to the frontend of an application and it consist of five main platforms and these are given below:

Back4App: Back4App is one of the main platforms of Baas service. It is an open source hub for building, managing and hosting API's for web, mobile, IoT apps with 80% faster and keeps full control in the backend.

Parse server: It as a mobile backend as a service initially developed by Facebook but then originally developed by Parse, Inc. in 2013 and shut down in 2017. It was used to send parse server a password reset and e-mail verification.

Firebase:A mobile and web app development platform of Baas that provides developers with a plethora of tools and services to help them develop high-quality apps, grow their user base, and earn more profit. It was developed by Firebase, Inc. and acquired by Google in 2014 and now it has become a multifunctional platform for mobile and web platform.

Appcelerator: It helps companies solve for this new mobile reality: delivering native cross-platform apps at the speed of web, mobilizing any data source and driving success with real-time analytics from an open, cloud- based platform.

Kinvey: Kinvey is the first Backend as a Service that makes it super easy for developers and enterprises to setup and operate a cloud backend for their mobile, tablet and web apps. The progression of cloud computing makes it possible to integrate all the capabilities by a single click through Baas.

Baas architecture consists of four components such as IaaS (Infrastructure-as-a-service), PaaS (Platform-as-a-service), SDK (software Development Kit) and APIs (Application Programming Interface) here a developer can implement desired features by using APIs or SDKs.

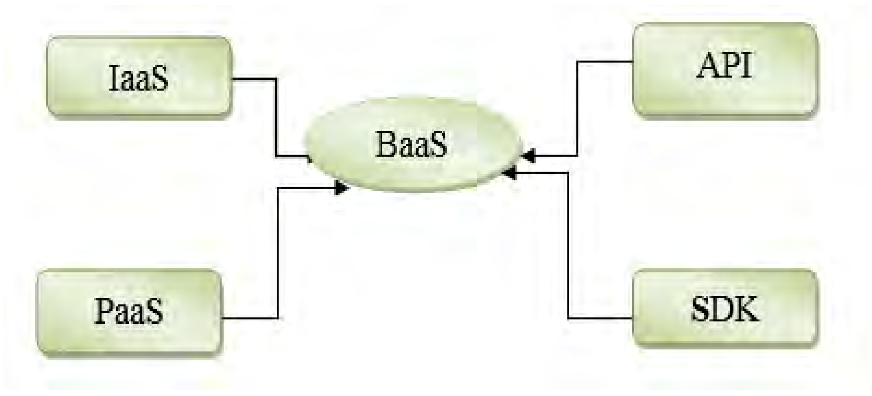


Figure 5: 2. 5 BaaS service with the architectural components

When a developer about to develop an application, they prefer four things such as cost effective, scalability, ease of development, application task and personal preference to complete developing the application. While building a large application with high tech features and resources it is supposed to be necessary to invest a lot of money into the development process but this doesn't seem a profitable aspect. That's what Baas do, helps to make an application in a very low cost since its architectural framework use cloud Baas which contains end-to–end infrastructure. To be a good competitor on human race companies really needs time consuming and cost effective service because if two of the company building same featured application or need to write same code repetitively it would eventually be a waste of time for both companies.In addition, a market can be full of apps that can be their competitors so it is highly required to develop it better, quicker and in high-grade. After considering these problems, companies really preferred to use Baas service which will let developers pay less attention to coding part and can concentrate on front end and design development and make the app more crowd pleasing.

BaaS is designed to remove backend development and deployment concerns for developers so that they no longer need to purchase servers (with IaaS), deploy backend environments (with PaaS), develop backend code (with BaaS) and can easily modify business logics (with SDKs and APIs), leading to quick implementations. Furthermore, it stops unnecessary stack development as that time developers main purpose would be to connecting to an API instead of spending hours in developing customized stack. Backend development can be 4 times faster depending on the type of application. It also allows large companies to change more quickly to market needs. Furthermore, Baas provides every backend solution that a company might not need a backend developer for small projects rather they could make their job done by frontend developers as they can build the entire software and for large projects backend developer can only focus on high value tasks instead of allocating development time on repetitive activities. So Baas is such cloud computing to speed up the application development process while developers working on a project and don't have that much time for customize backend for application Baas used as a magic wand to fetch real time output.

0.7 CHAPTER 3

0.7.1 PROPOSED SYSTEM

This chapter focuses on the architecture of proposed system where each segment has been described thoroughly and the total workflow of the system.

0.7.2 3.1 Architecture of Proposed System

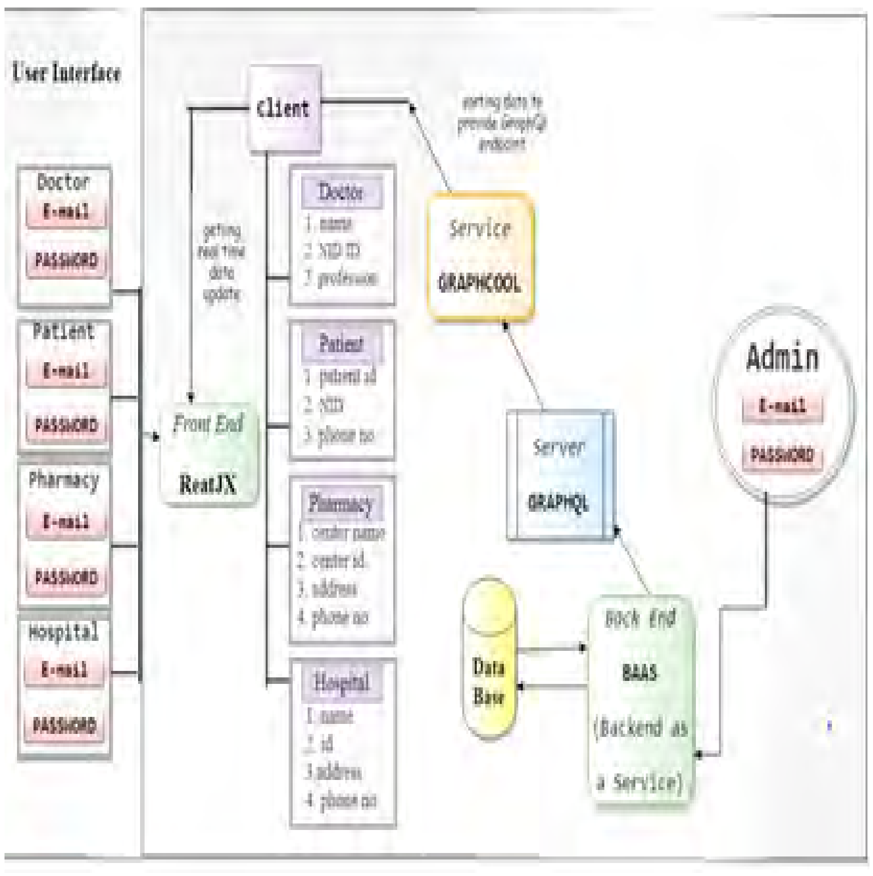


Figure 6: 3. 1 Architecture of proposed system

In our proposed system like other modern web applications it has two different distinctions front end and back end. In user interface, there are four types of users such as doctor, patient, pharmacy and hospital with Apollo client interface to connect towards backend. In user Interface we used ReactJs which worked perfectly in the dynamic interface that required for our IMDRMS platform. It can adapt to the data that entered into prescriptions and eventually adjust reports according to the patient given id. And in the backend we used GraphQl Baas through Graphcool service to get real time data update of a client. Here backend is working as a service to help the graphcool to adjust the user interface according to its need.

Another important aspect is to authenticate the user id for instance each time user login into the server with registered id the system will successfully take the user to their panel but if the user tries to log into the account with unauthorized account or id it will show a pop-message to put the correct information and this process authenticate the user by using JWT tokens. Later it matches all the tokens to identify whether the email address and password is correct or not and check server has that user or not.

0.7.3 3.2 Registration Process through UI of the proposed system

First of all, users like doctor, patient, pharmacy and hospital need to register them to get access to the system with their name, E-mail id, NID or birth certificate or passport no, phone no., so basically there will be two type user 1) registered and 2) non-registered user.

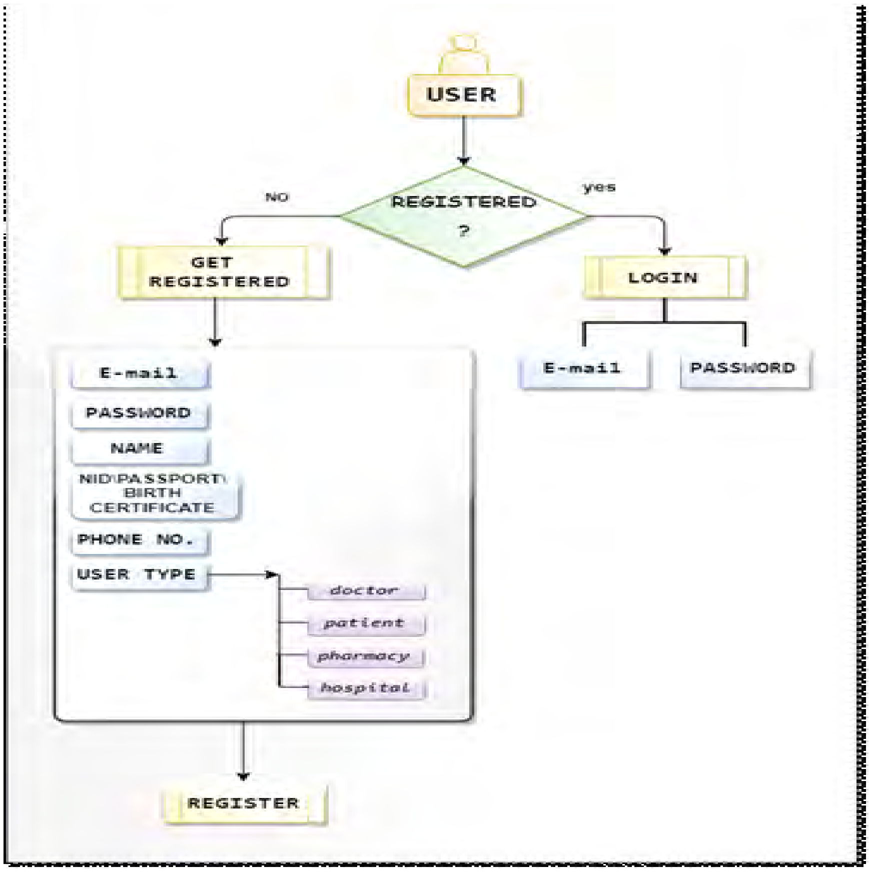


Figure 7: 3.2 Registration process of proposed system

0.7.4 3.4 Generating and storing medical information

This proposed system will allow doctors to find and store information instantly. Not only that but also it will help physicians and hospitals function in a smoother, safer, and more secure manner, allowing hospital personnel to retrieve and update the information of any patient with a single query so that doctors and authority can then concentrate more on the patient's issue than on the patient's records and authoritative duty.

0.7.5 3.4.1 Retrieval process of prescription and medical report of patient

This will describe how hospital authority, doctor and patient him/herself can retrieve medical information through a single graphQl query by Apollo interface from Amazon Dynamodb database. (AWS) Amazon DynamoDB is a fully managed proprietary NoSQL database service that supports key-value and document data structures and is offered by Amazon.com as part of the Amazon Web Services portfolio.

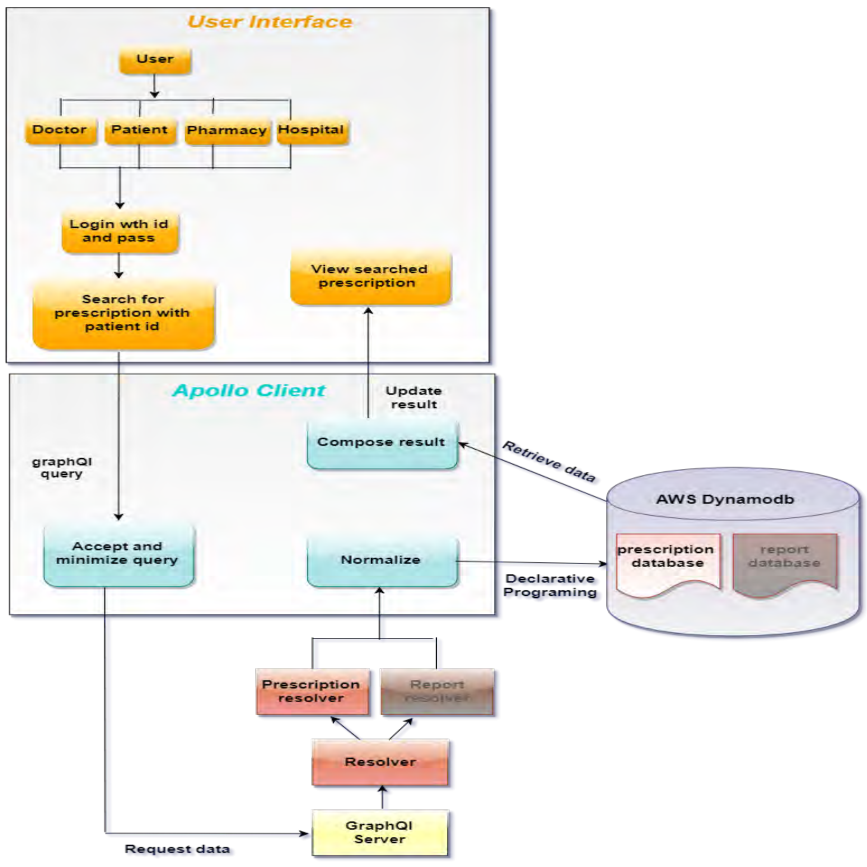


Figure 8: 3.3 Retrieving prescription data through graphQl server from Amazon Dynamodb

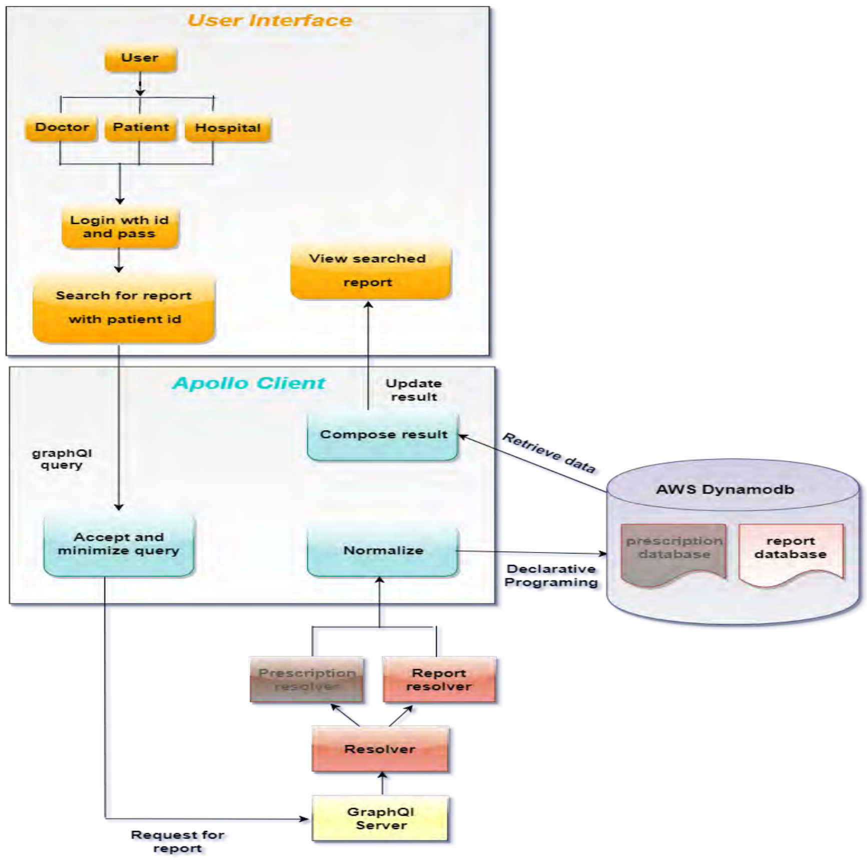


Figure 9: 3.4 Retrieving report data through graphQl server from Amazon

0.7.6 3.5 Modular Design

Modular design is a design which subdivides the system into different independent module for different flow path analysis to identify critical components in functionality and performance. It provides low coupling, high cohesion components, thus increase flexibility, modifiability, scalability and reusability. And for this we used use case diagram to analyze the flow paths and identify those critical components. Use case diagram of our proposed system is given below:

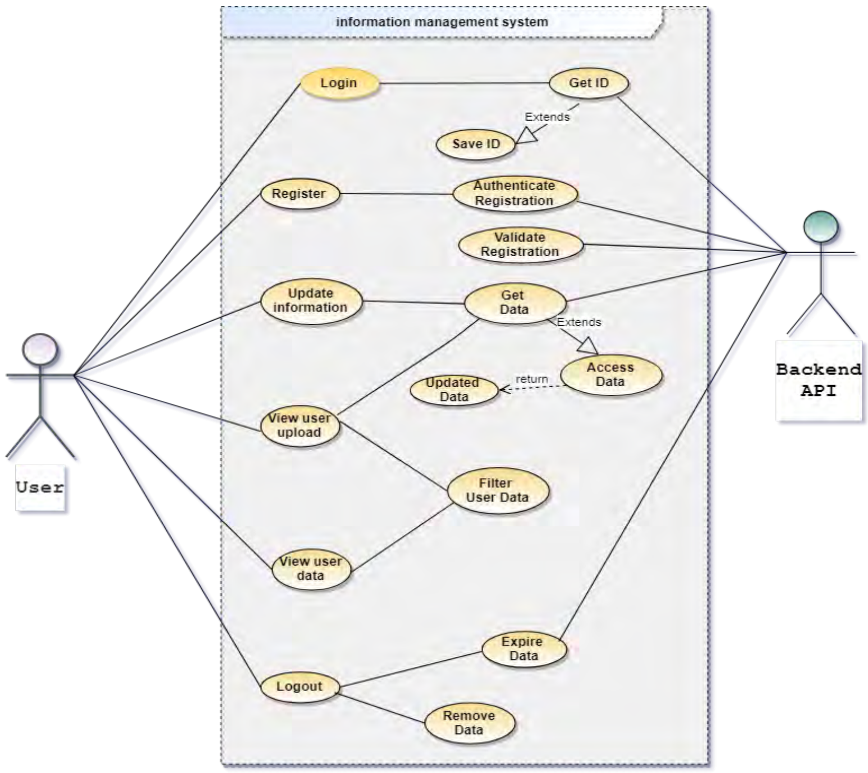
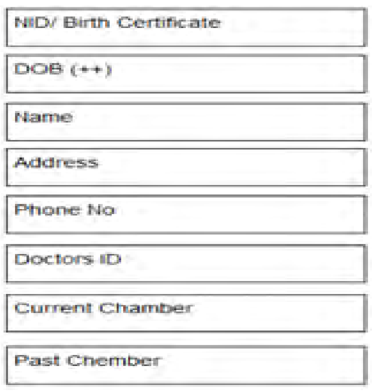


Figure 10: 3.5 use case diagram of information management of proposed system

0.8 3.6 Basic structure and user attributes of proposed system

Following the architecture of proposed system, it consists of many itration of planning, designing, testing, building and reviewing. After planning the system now need to move onto designing the system and to be very specific designing is meant for front-end of user interface that is going to be shown to the users such as how it will work, how they can provide data, update data, review data. Each of the user has individual and independent dicipline to do. First each user need to register with valid information such as:



NID/ Birth Certificate
DOB (++)
Name
Address
Phone No
Doctors ID
Current Chamber
Past Chember

Figure 11: 3.6 Doctor Registration bar

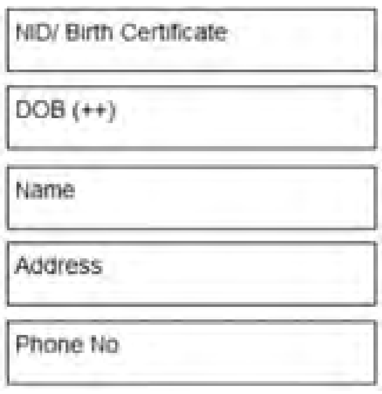


Figure 12: 3.7 Patient Registration bar

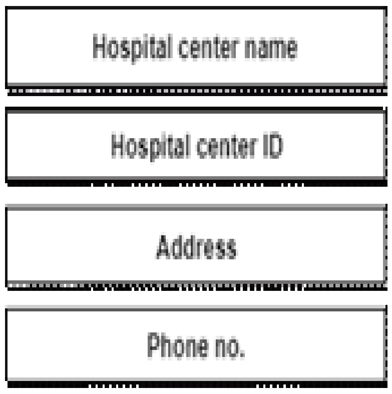


Figure 13: 3.8 Hospital registration bar

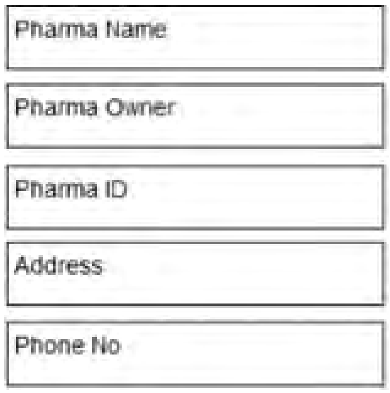


Figure 14: 3.9 Pharmacy registration bar

After completing each of the registration, the whole procedure of individual user's specification might look like the structure that given below:–

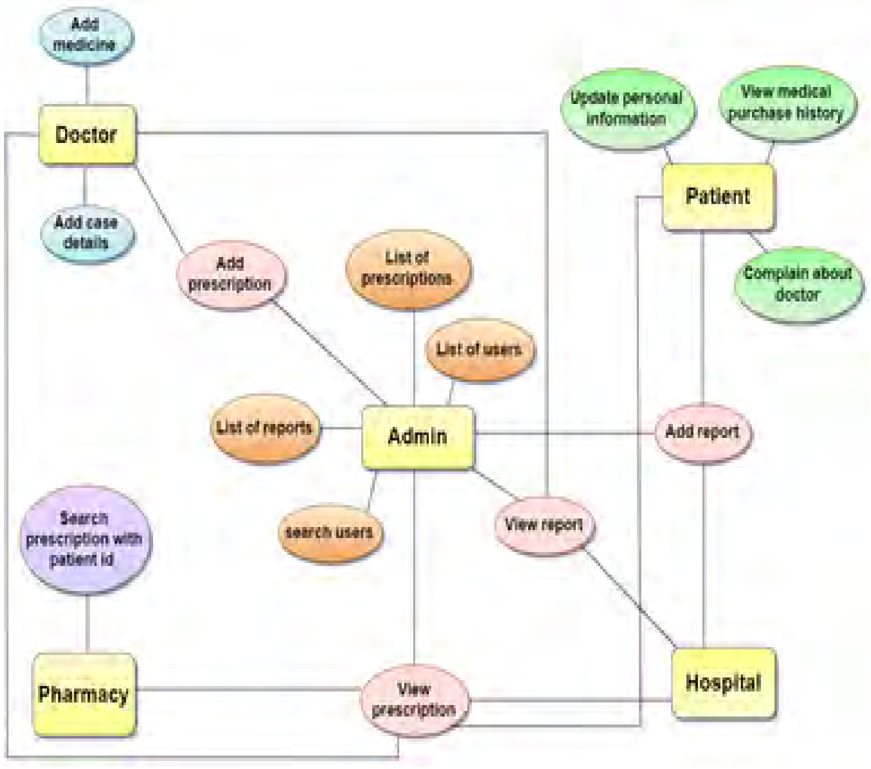


Figure 15: 10 E-R diagram of proposed system

0.9 CHAPTER 4

RESULT & IMPLEMENTATION

0.10 4.1 Implementation

This section provides an assessment on the implementation of the project which is based on the defined requirements mentioned before in the proposed system. The purpose of this section is to assure that every implementation has been done correctly and this web application makes sure to minimize the tedious paper work of storing medical information's. After the design phase is decided the development phase commences to give a proper view. To make everyone informed about any changes in the project we keep slack report that included what was completed yesterday, what is done today and a future notice for a future work plan. Furthermore, all git commits are pushed on to the GitHub repository whenever a feature has completed to get a good feedback to make the product and the development culture better. This project is aim to give doctor, patient, hospital and pharmacy all the medical history, medical reports, each user's information by one click. To start with, firstly they need to register with valid user information and the registration view look like this:

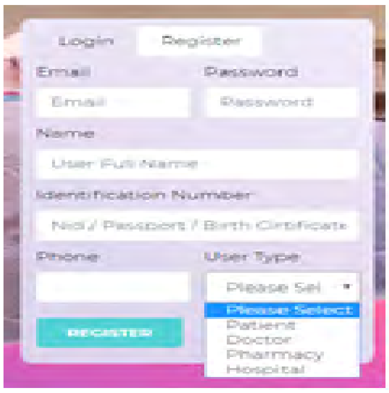


Figure 16: 4.1 Registration page with required registration fields

0.11 4.1.1Authentication

After a successful login, each and individual user will have independent panel where the user can go through their given segments. Authentication has been done through JSON Web Token in an action section to ensure every user id is unique and every request must go through those actions mention below:

- Users are required to have an account to be issued as an access account and a fresh account. Invalid accounts will show the response message error description on top of the application page. Moreover, username inputs remove capitalize letters and spaces when the request is sent.

- Every user will have unique user id given by backend server.
- A registration form must have the following fields: E-mail, Password, username, identification number, phone number and selection of user type.

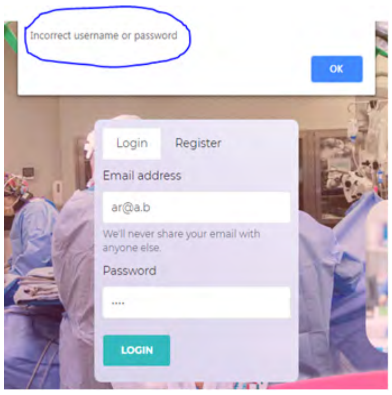

Incorrect username or password

Figure 17: 4.2 A pop-up message from server for having incorrect username and password

0.12 4.1.2Admin panel

In our thesis project admin has involved in some predefined tasks with several options and those options include having dashboard, search user, list of users, add report, list of reports, search prescription, list of prescription and create prescription.

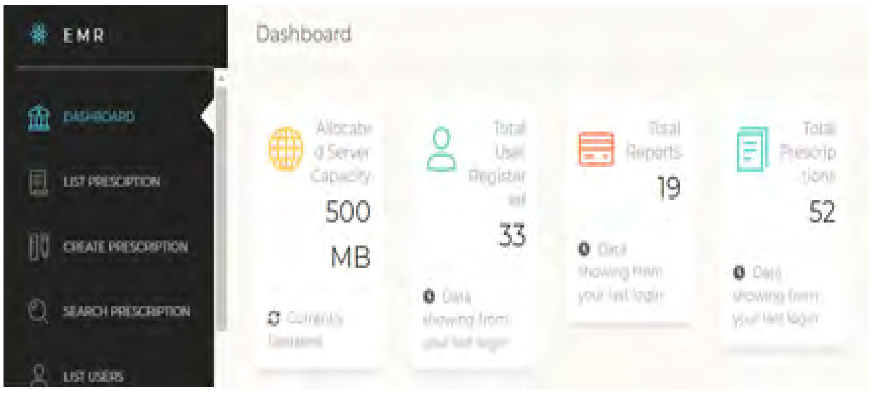


Figure 18: 4.3 admin dashboard showing server capacity, registered user, total reports and total number of prescriptions

Admin can see the list of users and search through the lists by user NID/phone/email id while also have the ability to view, update and delete certain user information.

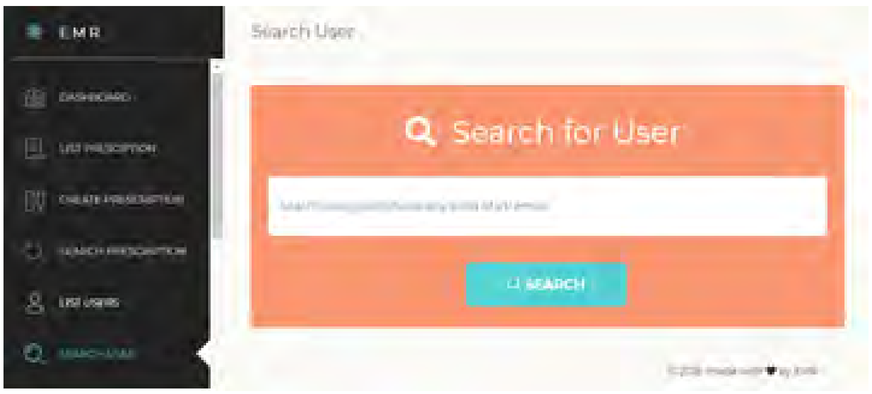


Figure 19: 4.4 searching user by user's NID or phone or email id

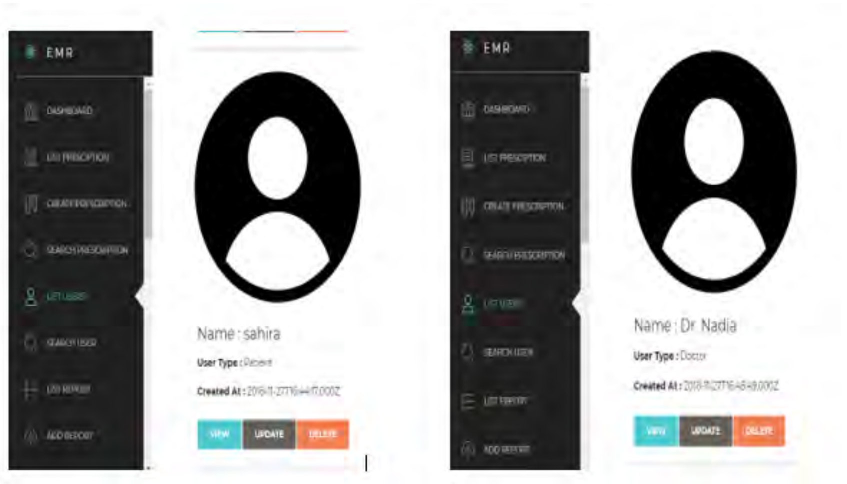


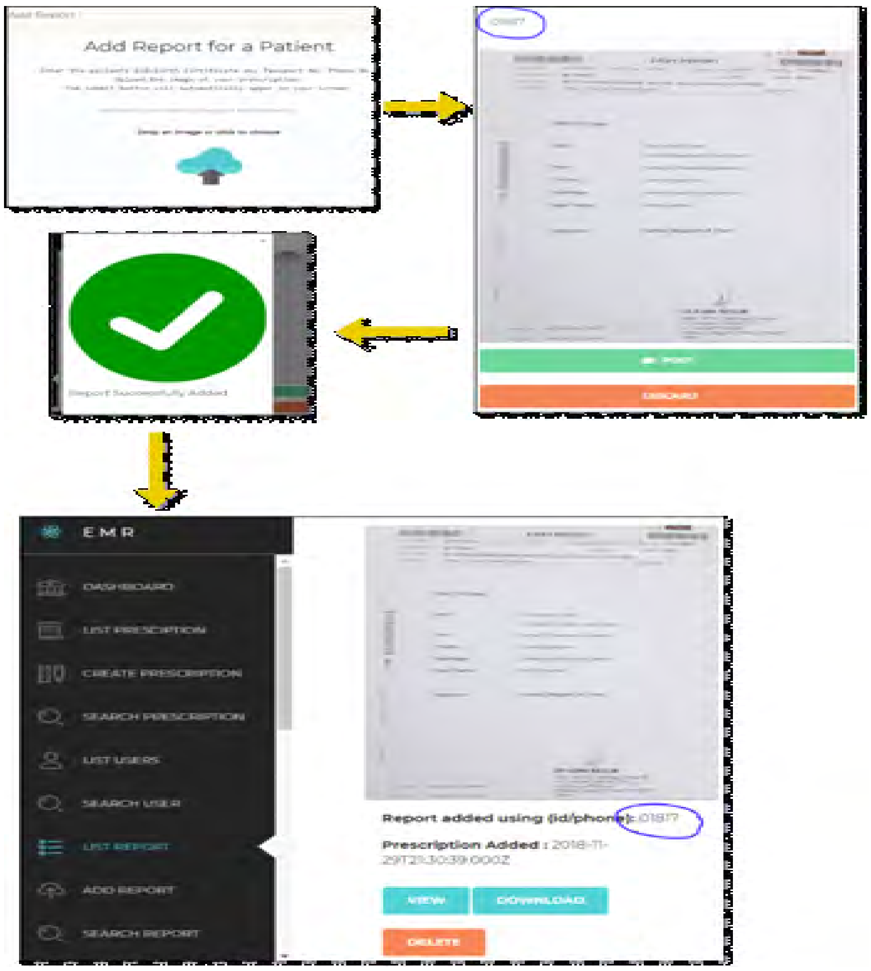
Figure 20: 4.5 Patient in the list of user and list of doctors in admin panel

4.1.2.2Prescription exploration

Similarly, admin can check the list of prescriptions, search for prescriptions and even admin has the liability to create prescription with verified user id. After adding the prescription successfully, it will be shown by doctor, patient, pharmacy, hospital authority and admin himself.

4.1.2.3Report exploration

Corresponding to adding a prescription admin can also add patient medical reports. After adding a medical report, a pop-up will be displayed saying report has been added successfully for the confirmation of that report and update the information in the report list of admin panel with view, download and delete option. Moreover, admin can search through the medical reports by giving the same registered patient id. A report adding process with a specific patient id for a specific patient and then displayed in the report list of admin panel is given below:



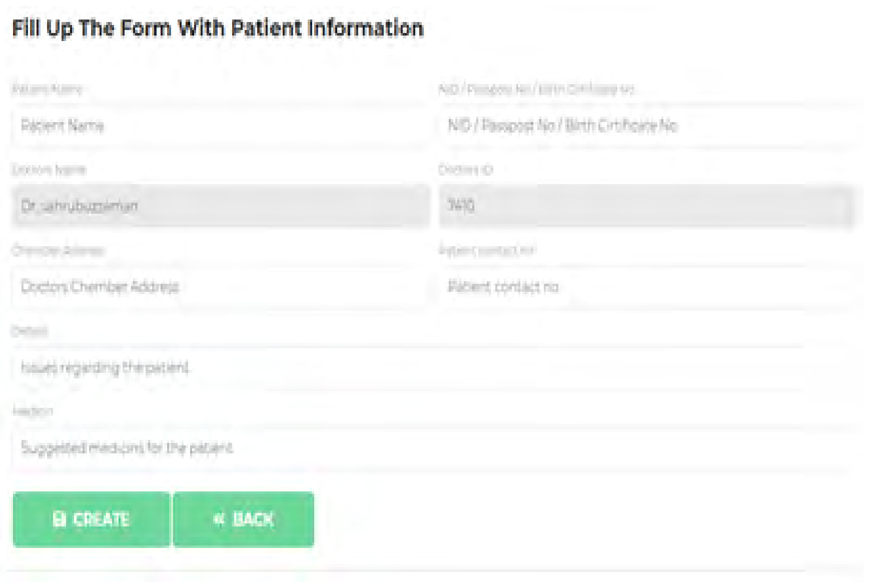
Report added using (id/phone): 01817
Prescription Added : 2018-11-29T21:30:39.000Z

Figure 21: 4.6 A medical report adding process with '01817' patient id.

4.1.3Doctor panel

As mentioned before doctor has four features to cover which includes adding prescription, adding case details, reviewing reports and updating previous prescriptions. Whenever doctor gets a new patient he or she can add prescription for that patient mentioning their health issues and then prescribe medicine for that health problem on the given field. Furthermore, for a revisiting patient doctor doesn't need to prescribe new one rather doctor could update the previous prescription even he can go through previously added medical reports before prescribing anything to better know the health condition of the particular patient.

]



Fill Up The Form With Patient Information

Patient Name

Patient Name

NID / Passport No / Birth Certificate No

NID / Passpost No / Birth Cirtificate No

Doctors Name

Dr. sahrubuzzaman

Doctors ID

7410

Chamber Address

Doctors Chember Address

Patient contact no

Patient contact no

Details

Issues regarding the patient

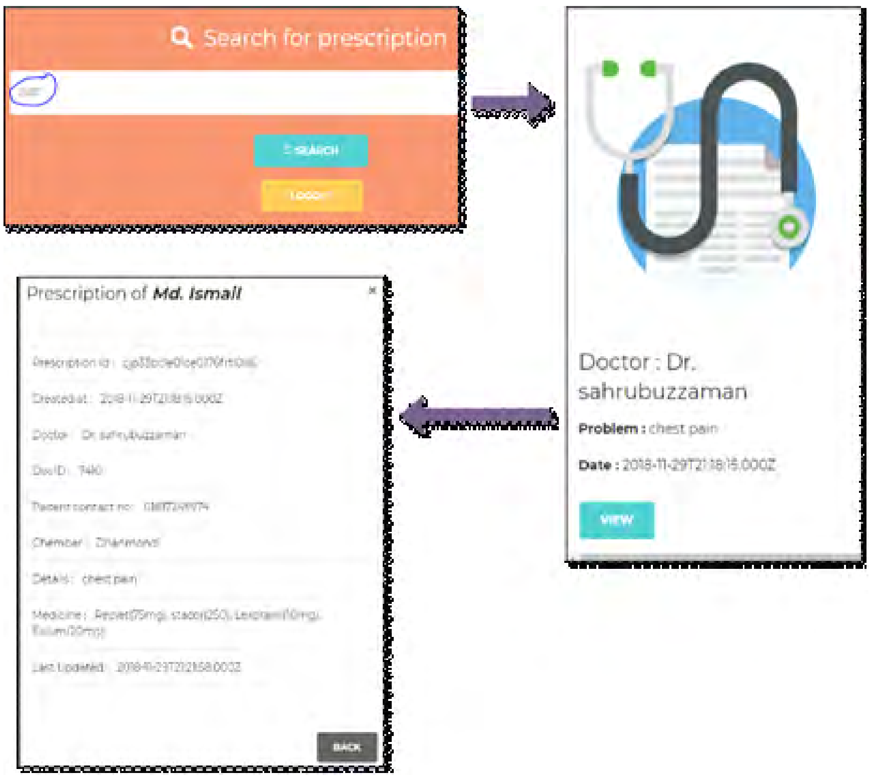
Medicin

Suggested medicins for the patient

Figure 22: 4.7 A prescription page from doctor panel

4.1.4Pharmacy panel

Pharmacists simply log into their account and will search for the prescription with a provided patient id. An updated prescription with all medicine information will send through server by that time immediately been checked by doctor. So, pharmacists will give those medicines which have been prescribed.

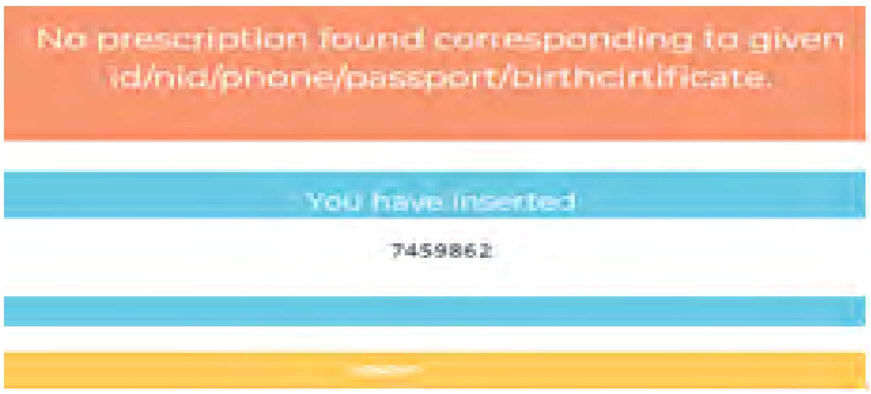


Doctor : Dr. sahrubuzzaman
Problem : chest pain
Date : 2018-11-29T21:18:15.000Z

Prescription of ***Md. Ismail***

Prescription id : cjp33bcle0lce0176rt10i6
Created at : 2018-11-29T21:18:15.000Z
Doctor : Dr. sahrubuzzaman
DocID : 7410
Patient contact no : 01817249974
Chamber : Dhanmondi
Details : chest pain
Medicine : Pezoet(75mg), stador(250), Lexotanil(10mg), Esium(20mg)
Last Updated : 2018-11-29T21:21:58.000Z

Figure 23: 4.8 Pharmacists search through patient prescription



No prescription found corresponding to given id/nid/phone/passport/birthcirtificate.

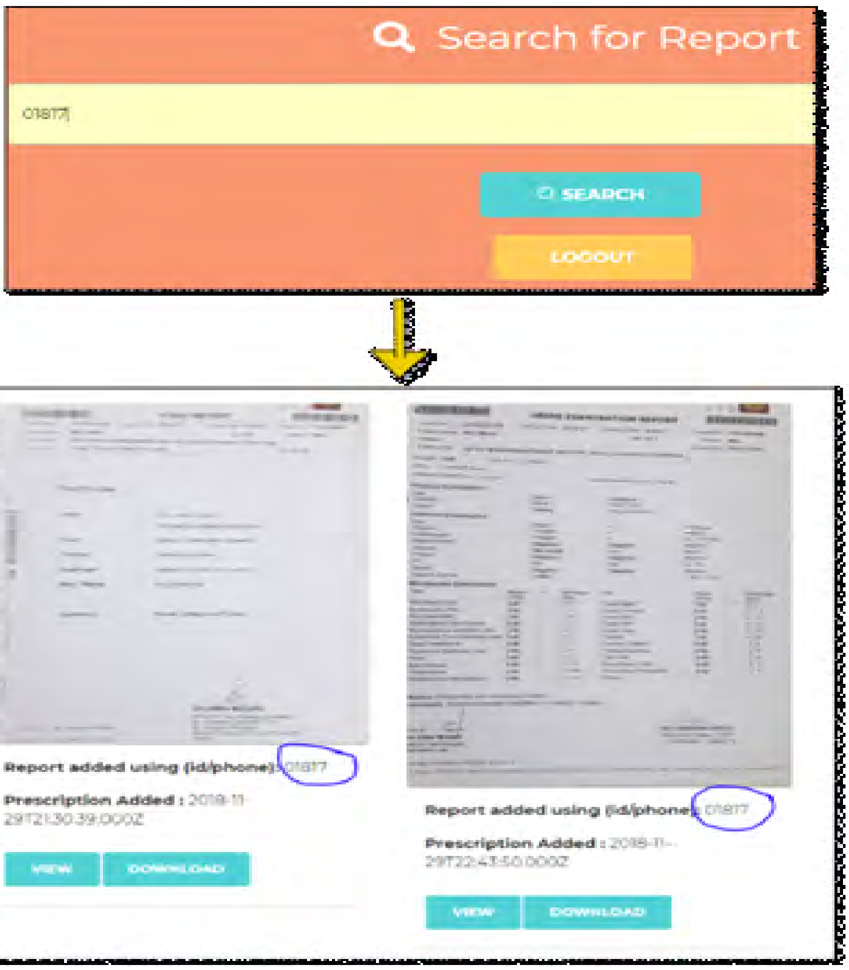
You have inserted

7459862

Figure 24: 4.9 preview page of searching prescription for unauthorized patient id

4.1.5 Hospital panel

Hospital considered as an authorized organization that can entail into adding report, reviewing existing medical reports added by patients and prescriptions added by doctors with various patient ids. Single patient can carry more than one medical report so our project assures that every medical report with same patient id will be all together in a single place to reduce hassle for that organization.



01817

Report added using (id/phone): 01817

Prescription Added : 2018-11-29T21:30:39.000Z

Report added using (id/phone): 01817

Prescription Added : 2018-11-29T22:43:50.000Z

Figure 25: 4.10 Hospital reviewing patient medical history

4.1.6 Patient panel

In the patient panel, patient has the accessibility to get all the prescribed information and medical reports from different doctors and different hospitals through same id and name. Before this implementation, every patient needs to borrow handful paper documents, registration file, previous reports for doctors, and previously prescribed medicines from different doctors from different organization so our thesis project made an easy solution of this hazardous work.

4.2 Result

Results are presented in a combination of narrative, summary and inferential statistics. This section describes the demonstration and validation of implementation.

4.2.1 Demonstration

In the process of condensing the demonstration, specific requirements of intended use regarding IMDRMS systems and data server surveys have been followed. This project has been demonstrated in local study and the response rate of this demonstration is high. In the context of Bangladesh most of the organizations still use the paper-based medical report and make pillars of medical information of patients which of them eventually been scanned and obliterated by the staff or authorities. As expected, a much higher frequency was found while show casing our IMDRMS platform to the organizations, doctors, patients, physicians. Most of the physician of the hospital found the procedure regarding information retrieval process much easier. Even patients are satisfied with storing information of doctor's prescriptions, medical reports etc. They are contented to add their previous reports in their own registered panel rather borrowing every medical report to doctors where doctor can easily go through the past reports before prescribing anything to the patient.

The comparison in IMDRMS use between doctors, patients, pharmacists and hospital is rising from the survey. In terms of user satisfaction and increasing number of user in admin dashboard can clarify that our project had simplified the work of paper-based hospital information system and that is why people shows keen interest to engage with this project.

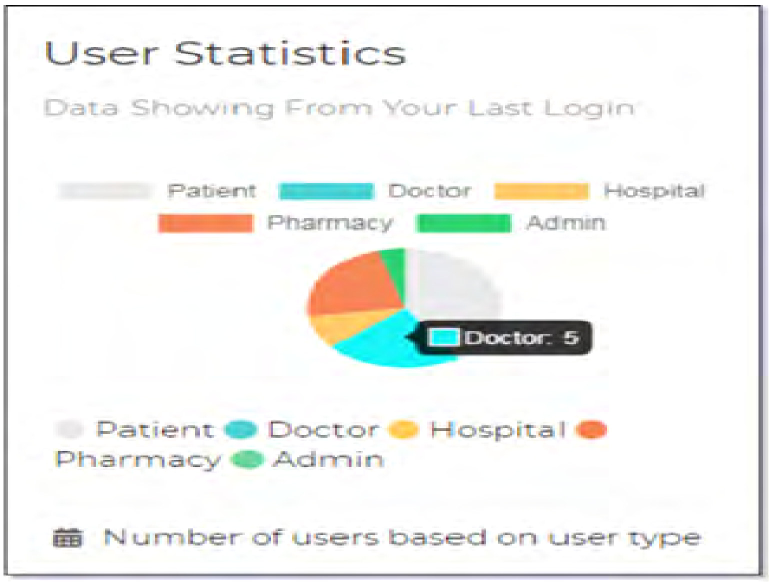
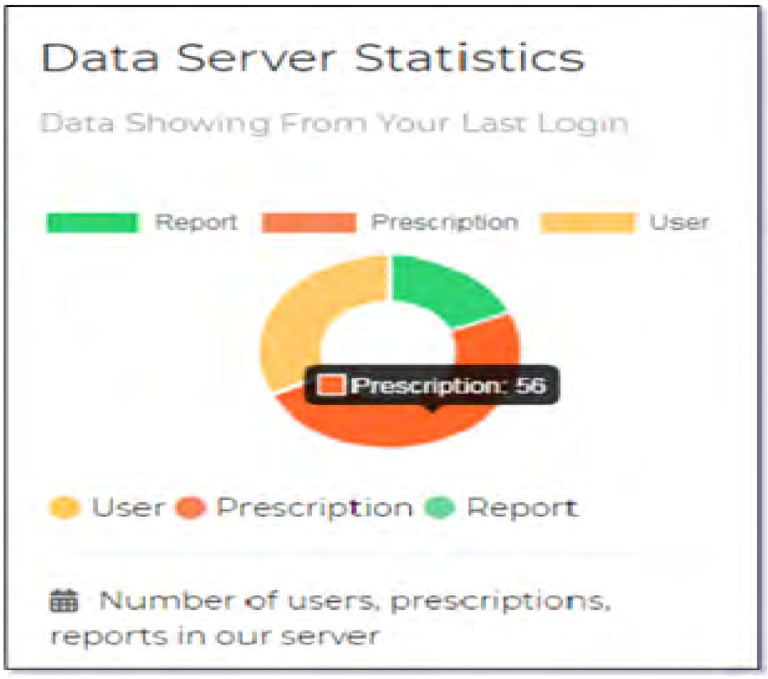


Figure 26: 4.11 Admin panel showing number of registered user statistics in real time

Not only that but also admin panel will display the total number of prescriptions and reports among total number of registered users to keep the track with data server. Each time user creates or updates data in their panel our IMDRMS automatically adjust the output through the backend server which will eventually be shown in admin dashboard. Each time the cursor moved it will display the real time data from admin last login for an instance here down in the figure we can see the total number of prescription is 56. The following data server statistics of total number of users, prescriptions and reports in the admin panel is given below:


Data Server Statistics

Data Showing From Your Last Login

Number of users, prescriptions, reports in our server

Figure 27: 4.12 Number of users, prescriptions, reports in IMDRMS server with real time value

4.2.2 Graphical representation of each disease and glance of leading disease calculated from prescriptions

Here we can see a graphical representation of diseases such as fever, typhoid, pneumonia, Malaria, diathermia, diabetes and cancer that always keep the organization informed and from the variation of the chart one can easily say our EMR platform can predict the trending disease.

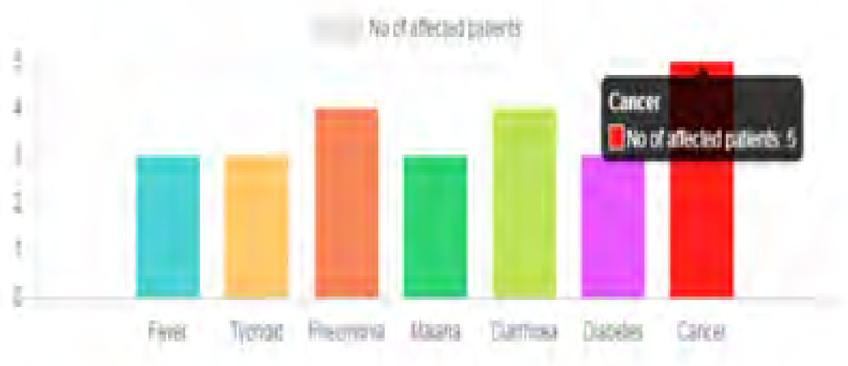


Figure 28: 4.13 Graphical representations of diseases and detecting leading disease

Right after saving each of the prescriptions it stores data into the database.

While compiling this chart, the system goes through every prescription that previously stored in the database and finds the field where the given diseases were actually mentioned. After finding the diseases it starts grouping the similar diseases in one queue and other similar diseases in different queues. Finally, the system counts all the listed queues like how many people or prescription been found with similar disease and compare with each other to display the final output in the graph. To clarify this, we can see in the given graph there are seven different diseases that have been mentioned with different flow rate and from them cancer bar seems with highest affected patient in comparison to other affected patient. Basically our IMDRMS can intelligently notify the outcome of expanding disease leaning among people. So by approving this system to every possible hospital, doctor, patient and pharmacy it can easily perceive of the disease that leading in our country to ensure most positive outcome.

0.13 Chapter 5

0.13.1 Conclusion

This section consists of a brief summary of our research work and the proposed IMDRMS system. IMDRMS has been proven fundamental tool for modernization in public health sector. It can drastically alter the nature of the patient-doctor, patient-physician, doctor-hospital, patient-hospital every possible relationship. Previously in the chapter1, we already get to know that traditional systems are more error prone, insecure, and unreliable. Today's huge volume of medical data can hindrance the quality of service and the ultimate solution for this problem would be adopting IMDRMS to overtake the traditional system. The greatest benefit of IMDRMS is its simplified accessibility and manageability. With the help of IMDRMS there would be no more getting reports from medical labs and adding them into the present patient file, no need to manage the paper pile of files and no need to utilize office space and manpower. Instead of that doctor can search any patient in seconds, can view his history in the order he wants, can view reports related to the patient and view his own diagnosis notes in just a click. Moreover prescription can be more accurate, patients coding is more systematic and hence results in clean data for the physician also reduce human errors encountered due to manual work. So, no need to depend on the data as now the patient's data can be accessed simultaneously by many persons at the same time to eliminate the dependency. One of the most important benefits of IMDRMS is its ability to link with different systems and aid in information sharing.

It is obvious that there is nothing unchallenged. We also faced some challenges during our thesis work. First we tried to go with leading edge technology but for having less information and documentation in react, graph and the other entire API's we used in this project, we skipped those for having such limitations.

Green earth is not just two words in fact it is an expression to see the color green and to take care of Mother Nature. One of the major ways we all can move in the green direction is to eliminate the use of paper as much as possible. A lot of paper is used in healthcare system and the reason behind is the amount of an unimaginable data and to manage this data we use unimaginable amount of paper which is eventually coming from trees. Hence it makes sense for us to focus on this segment to take the help of technology and in this regard IMDRMS has been a boon in health care system and has helped in reducing paper to a great extent. Now data are electronically managed which because of less paper utilization and printing of loads of data and adding up of piles of sheets one above another.

1. **Future work plan**

This paper aims to provide people a user friendly solution to store their valuable medical data into the IMDRMS. Nothing can be more important in today's world other than accessing your own system from anywhere in globe and that's what our IMDRMS system will provide. So, to keep the pace if this platform gets approved by organization it needs to push to the next level. Some of the possibilities:

- **Cross Platform app:** This platform already has user friendly user interface though it can be improved to cross platform app which will be made for ease of access.

- **Drug suggestion using Machine Learning and AI:** This project has the capability to detect the leading disease that currently affecting the patients by surveying from all the prescription so it can be implemented as a drug suggestion using Machine Learning and AI by analyzing case details.
- **Automated voice input device to make the process faster:** As the world is moving faster with the help of modern technology so we can add the feature of an automated voice input device to this system to make the process faster for doctors, patients, pharmacists eventually for the hospital systems.

Bibliography

[1] Pereira R; Duarte J; Salazar M; Santos M; Abelha A. (2012). Usability of an electronic health record. 2012 IEEE International Conference on Industrial Engineering and Engineering Management (pp. 35-39). Hong Kong, China: IEEE.

[2] Kuoni, S. (2012). Electronic Medical Records, Healthcare, and the Patient. (pp. 19-21, 30-33).

[3] J. Duarte, M. Salazar, C. Quintas, M. Santos, J. Neves, A. Abelha, and J. Machado. Data quality evaluation of electronic health records in the hospital admission process. 2010 ACIS-ICIS, (pp. 201-206).

[4] Izzatty N, Ismail B; Abdullah N. (2011). Developing electronic medical records (EMR) framework for Malaysia's public hospitals. 2011 IEEE Colloquium on Humanities, Science and Engineering (pp. 50-52). Penang, Malaysia: IEEE

[5] Chen W, Akay M. (2011). Developing EMR in Developing countries. IEEE Transactions on Information Technology in Biomedicine. (pp. 62 - 65) vol 15, no.1.

[6] G. Octo, MD. (2006. Report to the National Institutes of Health Division of Research Grants Computer Research Study Section on Computer Applications in Medical Communication and Information Retrieval Systems as Related to the Improvement of Patient Care and the Medical Record. vol 13, no 2, (pp. 127-135).

[7] Jin Y,Deyu T, Xianrong Z. (2011). Research on the distributed electronic medical records storage mode. 2011 IEEE International Symposium on IT in Medicine and Education (pp. 45-47). Cuangzhou, China: IEEE.

[8] K. Tung, Developing a frontend application using ReactJS and Redux, pp.15-17, 31-35.

[9] Tseng Y, Chung Y, Ping X. (2013). Web-Based Data-Querying Tool Based on Ontology-Driven Methodology and Flowchart-Based Model.

[10] Abdullah N, Ismail N. (2011). Developing Electronic Medical Records (EMR) framework for Malaysia's Public Hospital.

[11] Bonnie B, Jessica L, Jaime L, Qiana B, Robert J, Daniel A. (2009).Use of Electronic Medical Records for Health Outcomes Research (pp. 611-615).

[12] Zachary D, Lara K. (2015). Evaluating the epic electronic medical record system: A dichotomy in perspectives and solution recommendations.

[13] Lim E, Chennupati K, Surya G. (2009). Electronic Medical Records Management Systems: An Overview. vol 29, no 6.

[14] Murugabalaji K, Rod O, James L. (2013). Effect of an electronic medical record information system on emergency department performance. (pp. 201-204) vol 198, no 4.

[15] Joseph C, MD, Marjorie G. (2018) Developing an Electronic Medical Record System to Monitor Patients on Outpatient Parenteral Antibiotic Therapy at an Academic Medical Center. (pp. 556, 2018) vol 5, no 1.

[16] Jennifer B, Pharm D, Lonnye F. (2018).Clinical and financial effects of smart pump–electronic medical record interoperability at a hospital in a regional health system. vol 75, no 14.

[17] Octo G, MD. (March 2006.). Report to the National Institutes of Health Division of Research Grants Computer Research Study Section on Computer Applications in Medical Communication and Information Retrieval Systems as Related to the Improvement of Patient Care and the Medical Record. (pp. 127-135) vol 13, no 2.

[18] Vimla L, Kushniruk W, Seunami Y. (2000). Impact of a Computer-based Patient Record System on Data Collection, Knowledge Organization, and Reasoning. (pp. 569-580) vol 7, no 6.

[19] William M, Michael W. (2009). Implementing Electronic Medical Record Systems", vol 11, no 6.

[20] Piliouras T, Raymond P , Tian X, Zuo B, Yu S. (2013).Electronic health record systems: A current and future-oriented view. 2013 IEEE Long Island Systems, Applications and Technology Conference (LISAT). Farmingdale, NY, USA: IEEE.