## **EMR**

**Electronic Medical Record System - Ethiopia** 

#### Introduction

Medical Record System (MRS) allows to improve the management of patient health care, enhance healthcare service and allow for rapid communications between health care providers. One of the critical factors of Electronic Medical Recording (EMR) is interoperability the ability of different information systems and applications to communicate, exchange data accurately, effectively and consistently and to process the information that has been exchanged

## Functional Requirements - Patients

| Requirements   | Flag   |
|--|--------|
| System Must Record Patient Demographics                        | Must   |
| System Must retrieve Patient Demographics (Last name, DOB)     | Must   |
| System Must allow new patients to register                     | Must   |
| System Could allow patients to login with their credentials    | Could  |
| System Must allow patients to Update their Demographics        | Must   |
| System Must allow Patients to make an appointment              | Must   |
| System Must allow Patients to Manage an appointment            | Must   |
| System Must Scan and Store previous medical history            | Must   |
| System Should notify the patients about their appointment      | Should |
| System Must allow the patients to choose a particular Hospital | Must   |

<sup>\*\*\*</sup>Initial requirements are subject to change as per the client requirements.

## Functional Requirements - Doctors

| Requirements   | Flag   |
|--|--------|
| System must allow doctor to register   | Must   |
| System must allow doctor to view/manage appointments                                     | Must   |
| System must allow doctor to view patients records  | Must   |
| System must allow doctor to prescribe medication (Digitally)                             | Should |
| System should allow doctor to add lab test for particular patient                        | Must   |
| System must generate E- Prescription   | Must   |
| System Must allow Patients to Manage an appointment                                      | Must   |
| System should allow doctors to view and follow up reports provided by the laboratory     | Should |
| The system must be capable of storing unlimited number of Patient records and documents. | Must   |

# Non Functional Requirements - Doctors

| Requirements   | Flag   |
|--|--------|
| Accessibility - The system must support running in high availability mode  | Must   |
| Backup - Ability to perform full system recovery from scheduled backup files with RTO (recovery time objective) less than 4 hours. | Must   |
| Accessibility - The management interface must be accessible via web browser  | Could  |
| Performance - The system must be capable of performing full documents search in a reasonable amount of time (less than 5 seconds). | Must   |
| Fault tolerance – The system must be available and active all the time   | Must   |
| Compatibility – The system must run on all popular platforms   | Must   |
| Security – Confidential Patients and Hospital data must be stored and accessed securely  | Must   |
| Usability - The System must be intuitive and user friendly   | Must   |
| Extendibility – The system must allow adding new features and carry-forward of customisations for the next major releases.         | Should |

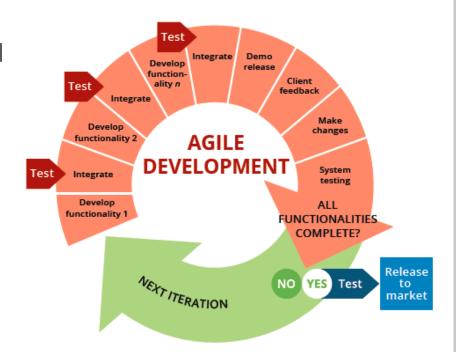
### Technical Requirements – Hardware

- Computer connected to the Internet (LAN for now)
- Document imaging, or scanners, Printers and other related devices
- Storage and Backup Devices
- Client Server Computer which maintains EMR data and Software Services.
- Computers for the Hospital Staff and Doctors.
- Network devices for connecting clients and server

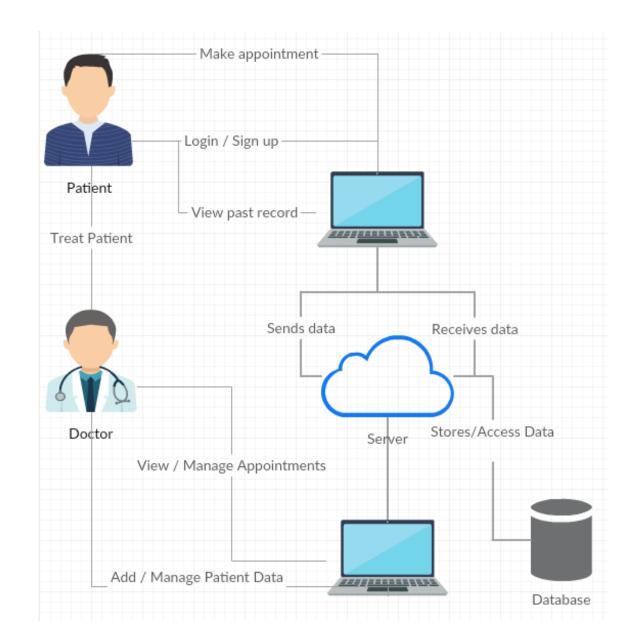
Note: The above requirement are subject to change based on the growing functionalities.

## Software development lifecycle

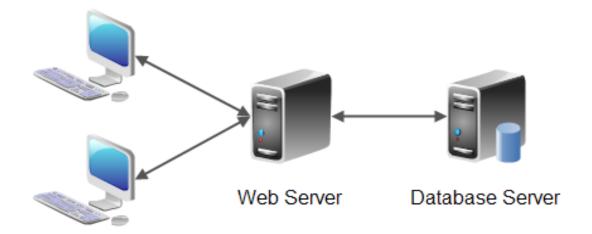
- This system will be developed based the agile development model
  - Quick Go Live
  - Fast Track product evolution
  - Creates Proof of concept in the early stages.
  - Allows to test different types of functionalities and user experiences for best results
  - Minimizes risks



## Rich picture System proposed



## System Architecture



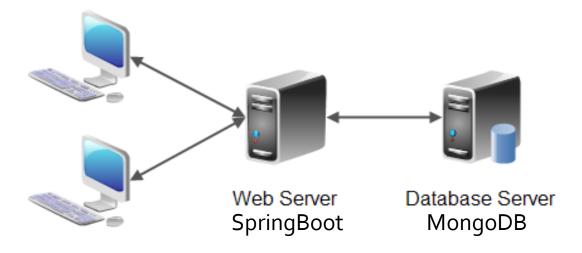
Web Browser

- Rich User Experience –Responsive design, user friendly across multiple platforms and various screen sizes.
- Ease of access Can work from anywhere within the hospital with LAN access.

# Three -Tier Architecture & Benefits

- Scalability—The application can balance huge loads to satisfy more Web requests without adding servers to the Application and Data tiers.
- Performance—The Presentation tier can cache requests, network utilization is minimized, and the load is reduced on the Application and Data tiers.
- Availability—If the server is down with sufficient caching, the Presentation tier can process Web requests.

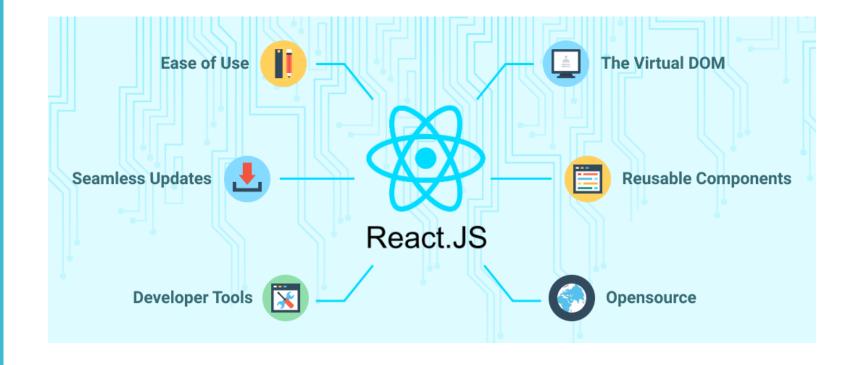
#### Technical Stack



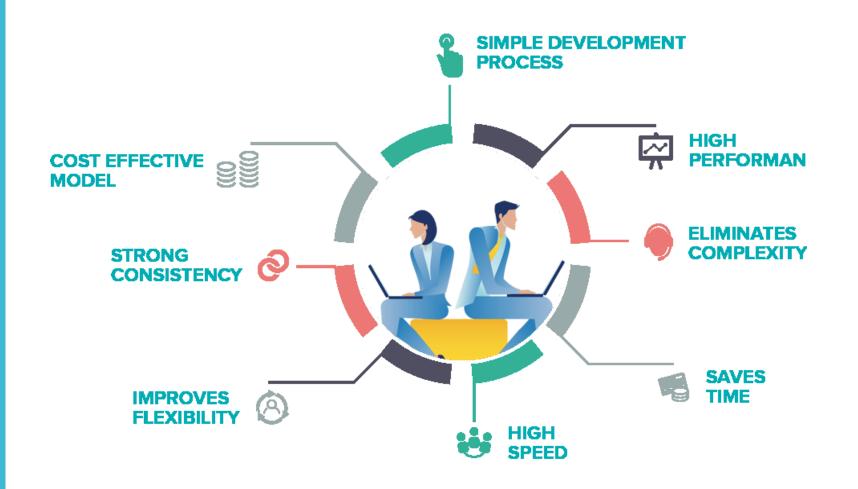
Web Browser ReactJs

- Web Browser / Front-end ReactJs, Bootstrap 4
- Web Server / API Spring Boot Rest web services
- Database server MongoDB

## Why ReactJS?

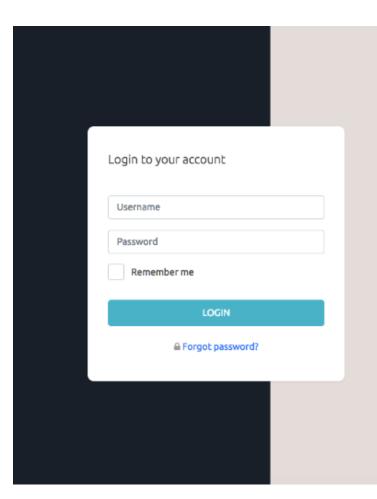


Why NoSQL?

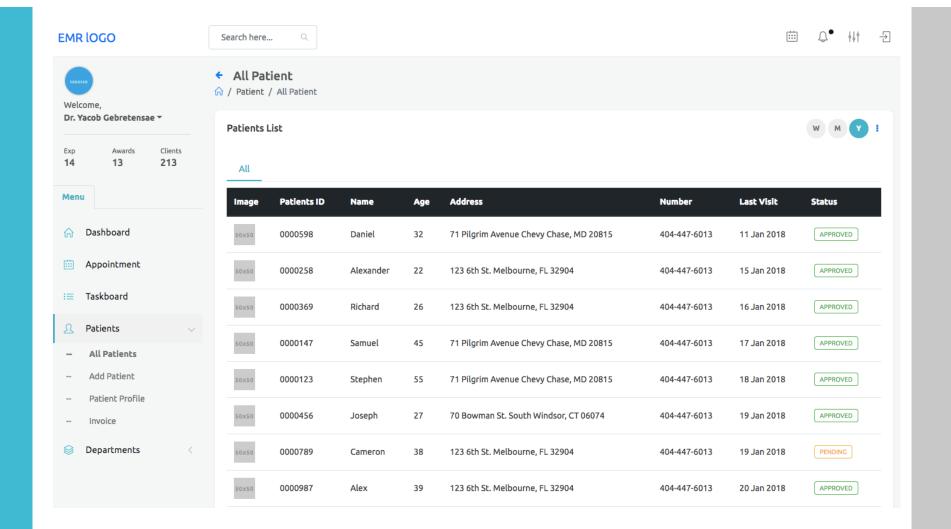


## System Overview

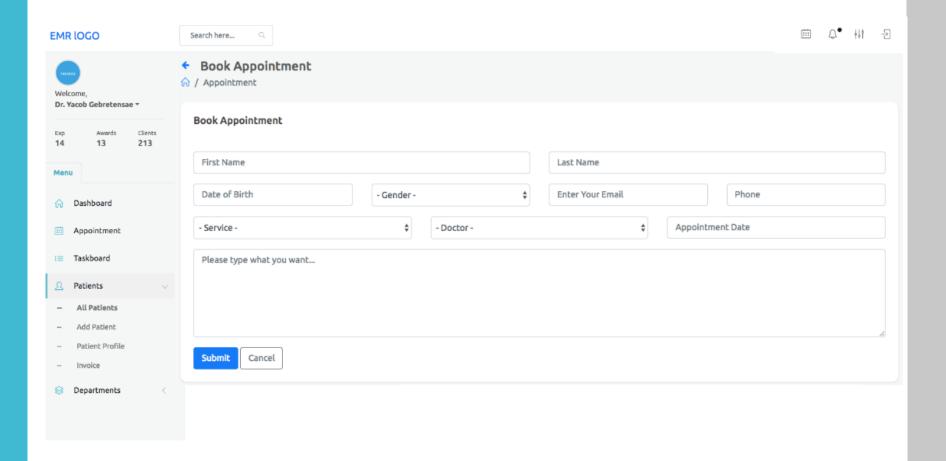
## Staff Login



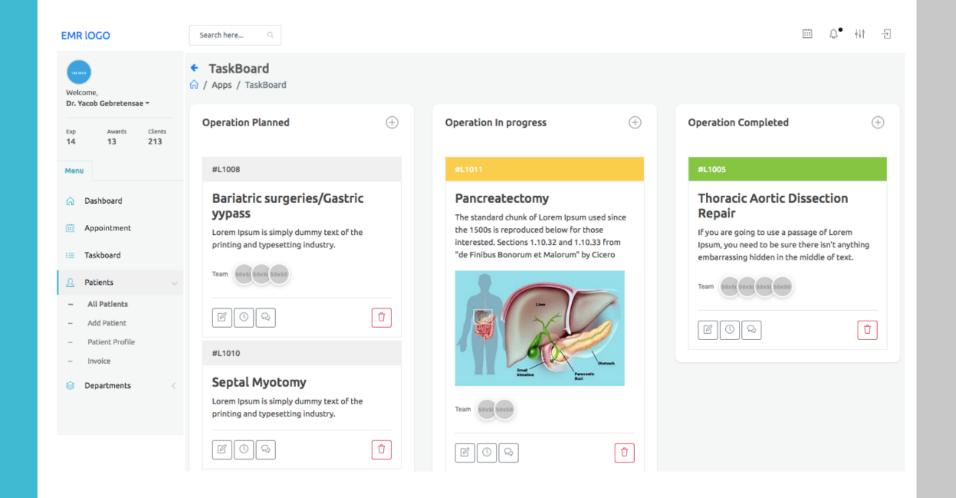
## **Patient Listing**



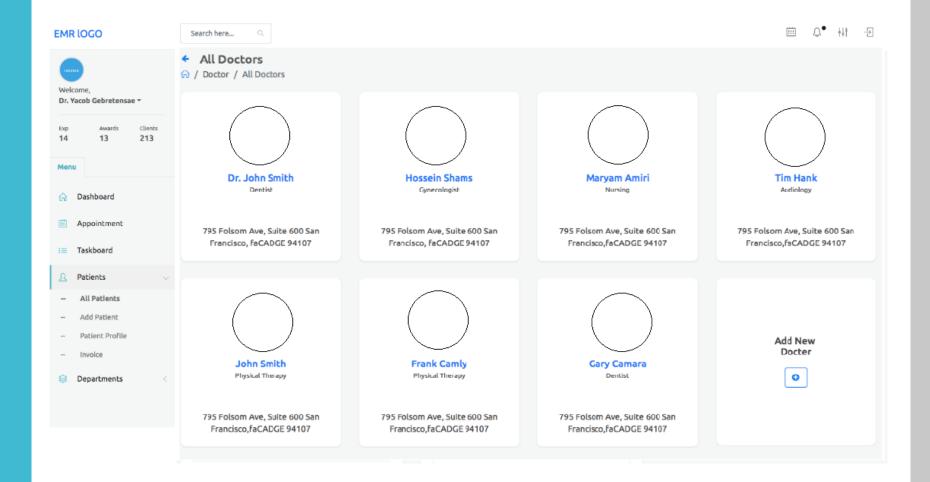
## Appointment Booking



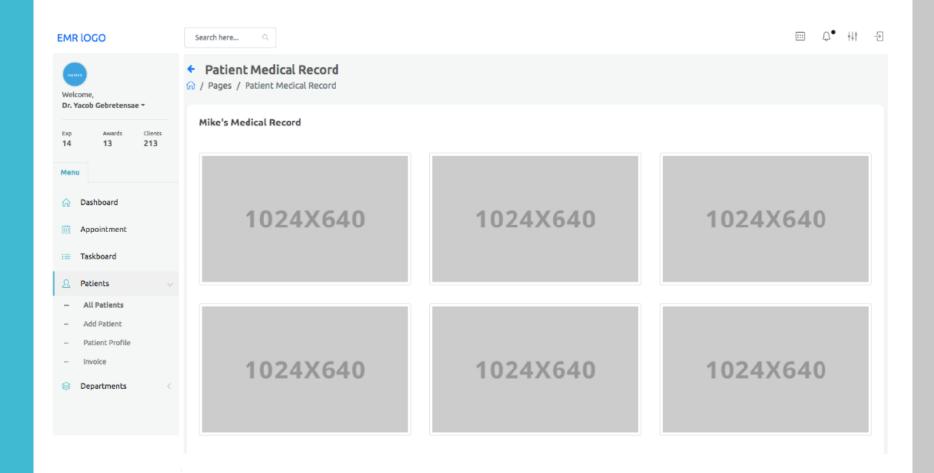
### Doctor Task Board



## **Doctor Listing**



## Patient Medical records



## More Importantly

## "Money is an illusion"

- Yacob Gebretensae

## We love Illusion



#### Credits

- Yacob Gebretensae, Data Scientist, MD, Healthcare researcher, London, UK.
- Niranchana Gowthaman M.Sc. Software Engineering, UK
- Raghavendra Prabhakar M.Sc. Applied Computing and Information Technology, UK