Linnéuniversitetet
Kalmar Växjö

2DV609 -

Project Course in Software Engineering

Design Document

Authors: Adam Rashdan, Kalid Diriye, Nitin Pasikanti, Ramin Jafari & Rashed Qazizada
Group: 4
Supervisor: Mirko D'Angelo & Mauro Caporuscio
Semester: Spring 2020

Topic

A Web application implementing a medical booking system. Using this application, users can virtually book an appointment for checkup.

- The name of the application is ‘Medical Appointment’

Introduction

This application provides a time-saving solution to the end-customers with opportunity to book an appointment to their relevant clinic.

The Team is considering a huge success of application as of its trivial, user-friendly layout. Furthermore, the team has decided to implement a Web Application using React web framework and JavaScript as the programming language.
The application also comes with a self-test for Covid-19 on first page of to attract user attention. After the user has completed the test, a message based on updated information is shown on an official link.

Purpose

This *Design Document* describes the web application as whole by first going through different parts of the system along with wireframes to support the implementation of the system.

Layout of the System

1. Homepage: This is the main page of the application. It shows the application name, logo, covid-19 test and other options for getting started.
2. Users: The user can make the following requests and receive responses from the web server:
    - Sign Up
    - Sign In
    - Book an appointment.
    - Cancel the booking/change the user details if required.

3. Admin (Secretary): The admin can validate the booking and assign the patients with a doctor.
4. Components: They are a collection of building blocks of the application.
    - Online test for Covid-19
    - Live Chat
    - Header/Navigation bar
    - Footer
    - Background page
5. Database: where data are stored and retrived

Design Requirements

The design requirements are the functional attributes that enable the team to convert ideas into design features. They are as follows:

1. The user interface for this application is via the web.
2. It should be fast and responsive to the requests.
3. It can save user data and can be retrieved to modify if necessary.
4. It should consist of a user registration.
5. It should consist of a log in option.
6. It should show a calendar with available slots for booking.

General priorities

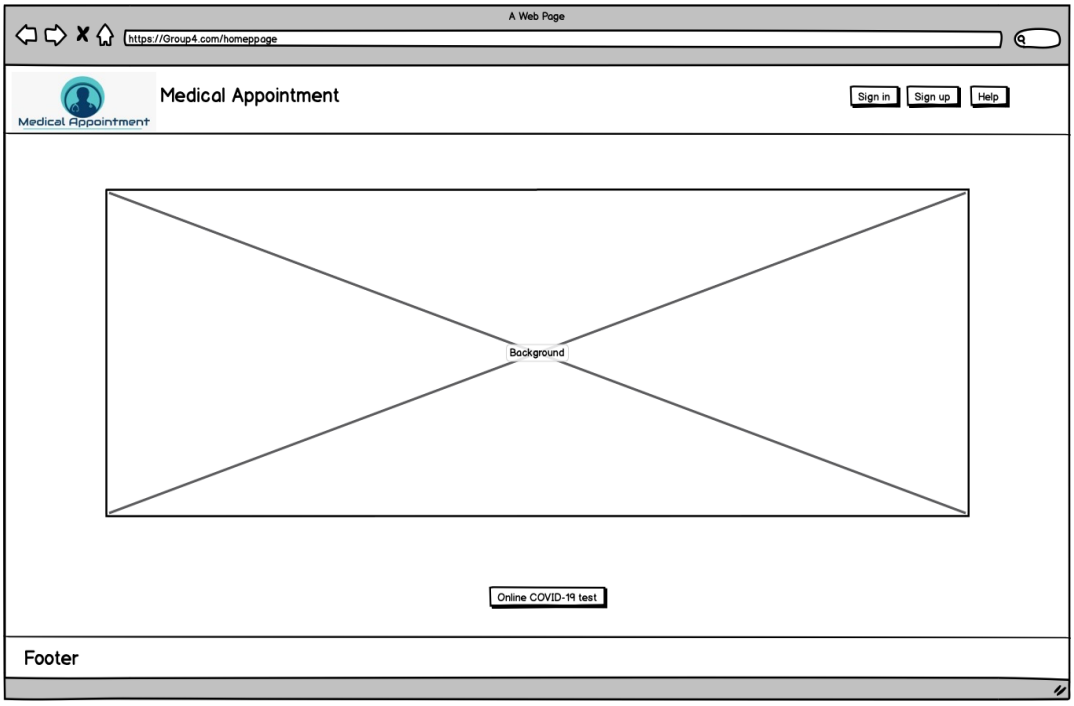
1. The system should be scalable.
2. The system must be easy to use.
3. The system should offer security for the user data against cyber attacks.
4. The make the process of booking easy, a color scheme will be implemented within the calendar where red stands for slots full, yellow for almost full and green indicates that none of the slots are booked.
5. The programming language used is Javascript and the framework is React.

Outline of the Design

This is the intentional creation of a plan or specification for the construction of an object or system or for the implementation of an activity or process.

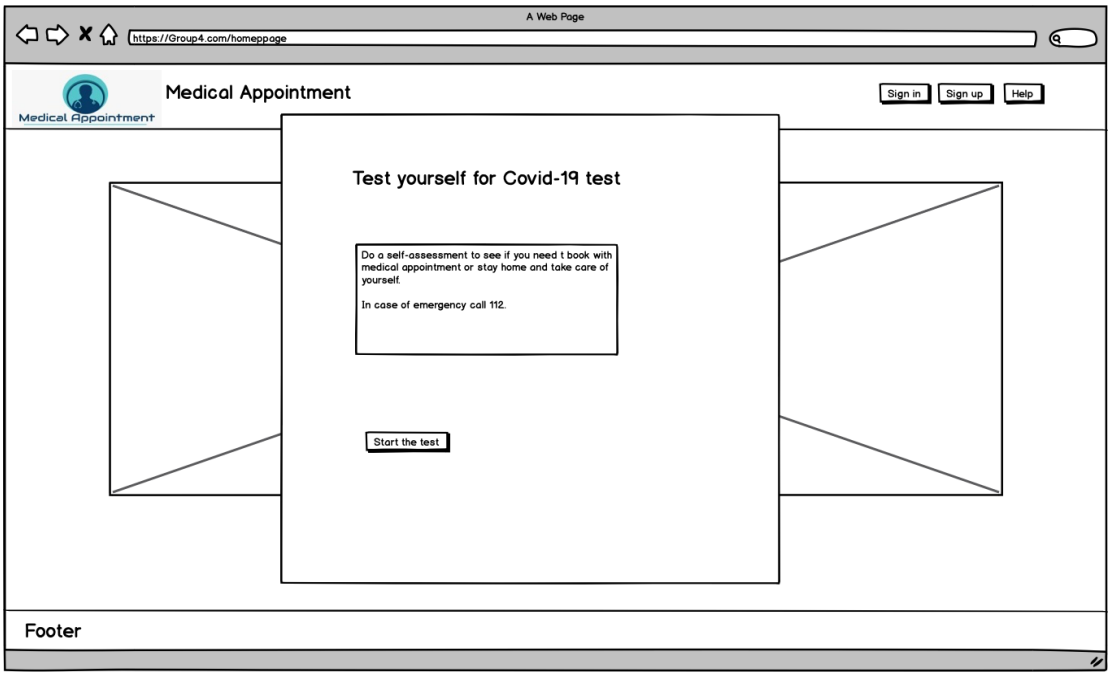
Below are wireframes structures implementing the different parts of the system (components) and a few of these have also been discussed in the design requirements.

1. The application comes with a landing page/ homepage.

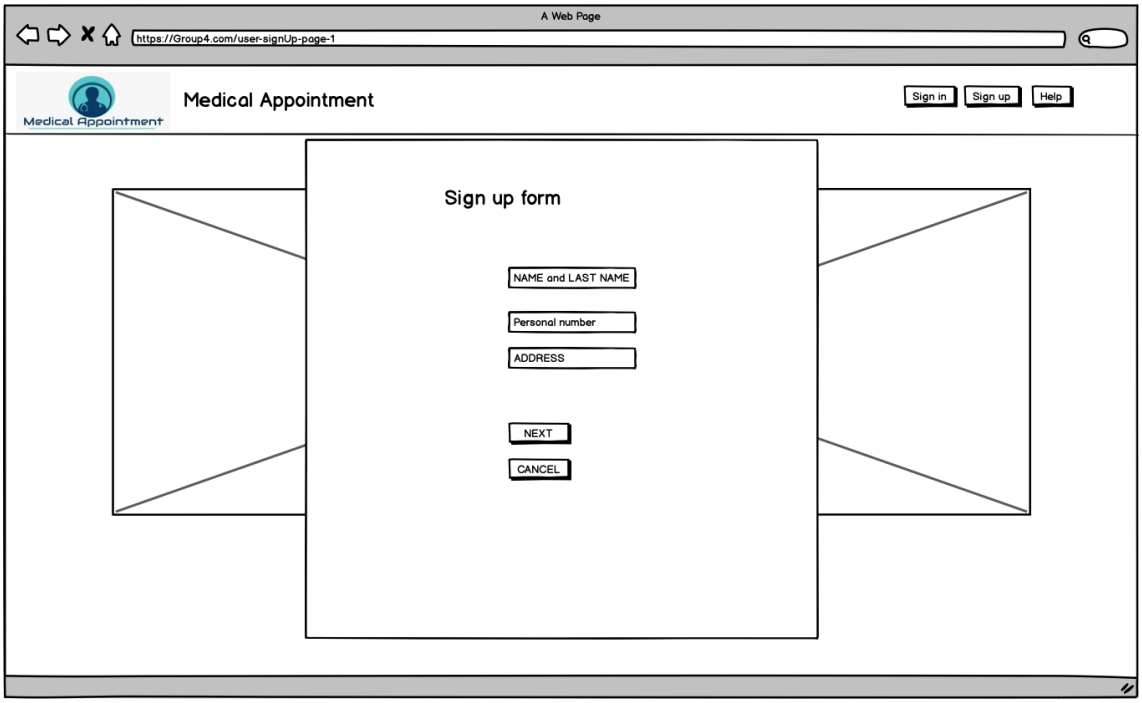


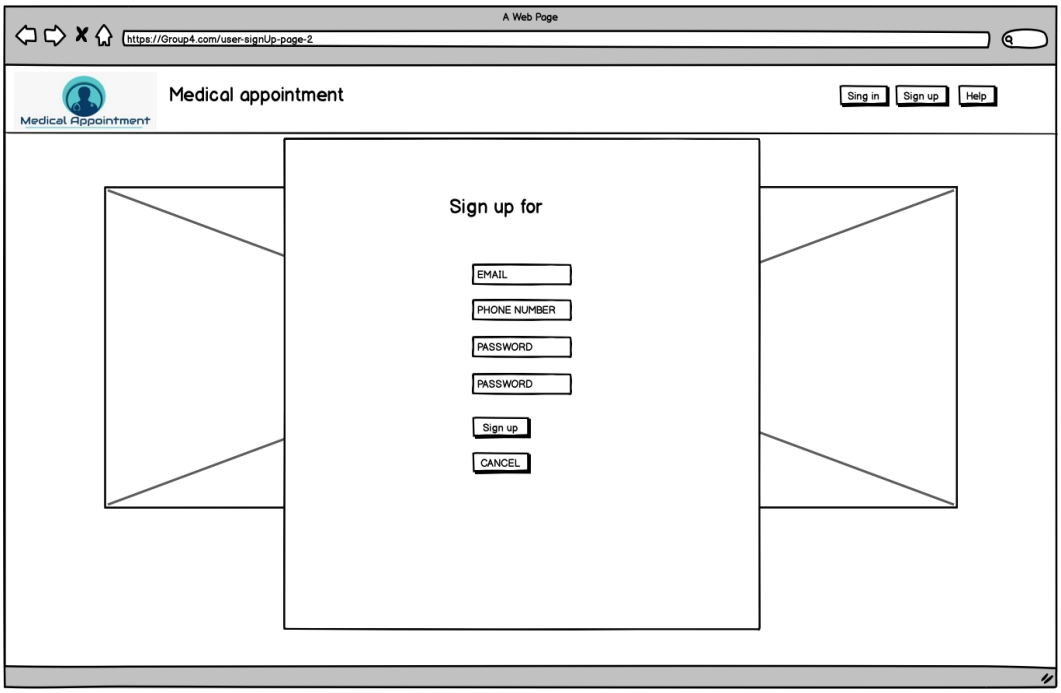
In the above diagram, the navigation bar consists of the application name which is ‘Medical Appointment’ and the logo on the left side. On the right hand side, we see the Sign in, Sign up and Help buttons which will be further discussed. The page also has a button for Online Covid-19 test and a footer.

2. The user can take a self test for covid-19 on the website. The user answers a set of questions about his health and symptoms using which a response is shown and suggestions are given.

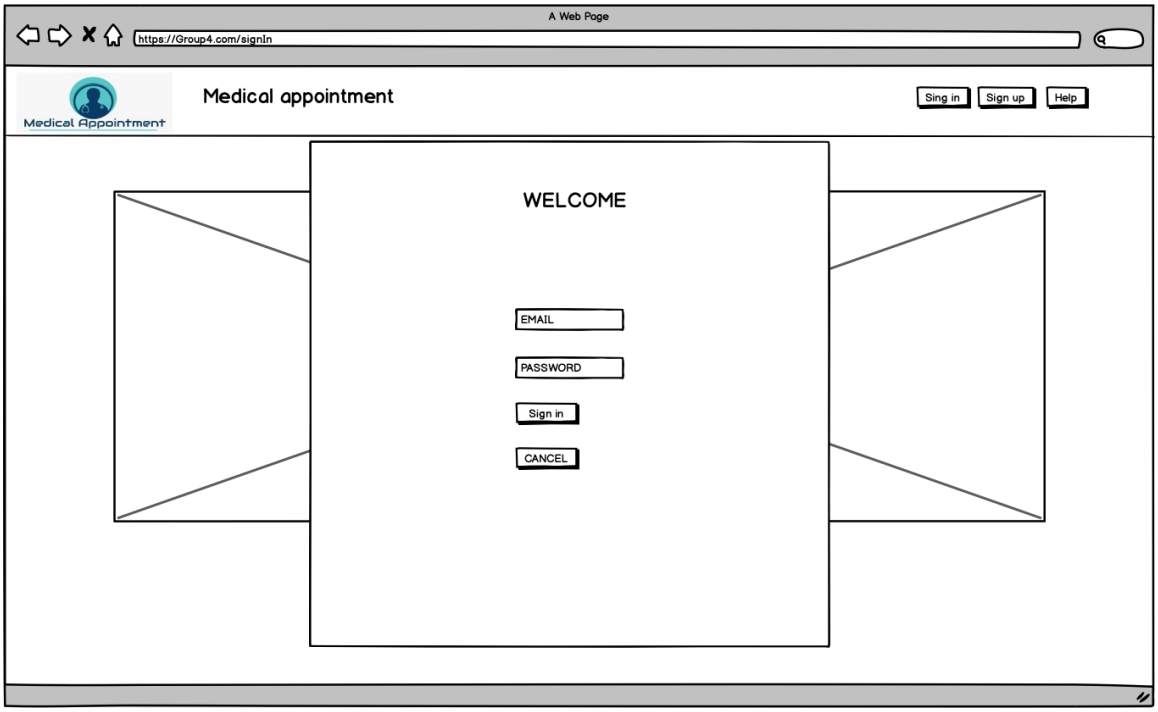


3. If the user is a new member, he can register with the application by clicking the sign up button. The user enters his personal details like Full name, email address, personal number etc. The user details are stored in the database and will be validated for future login.

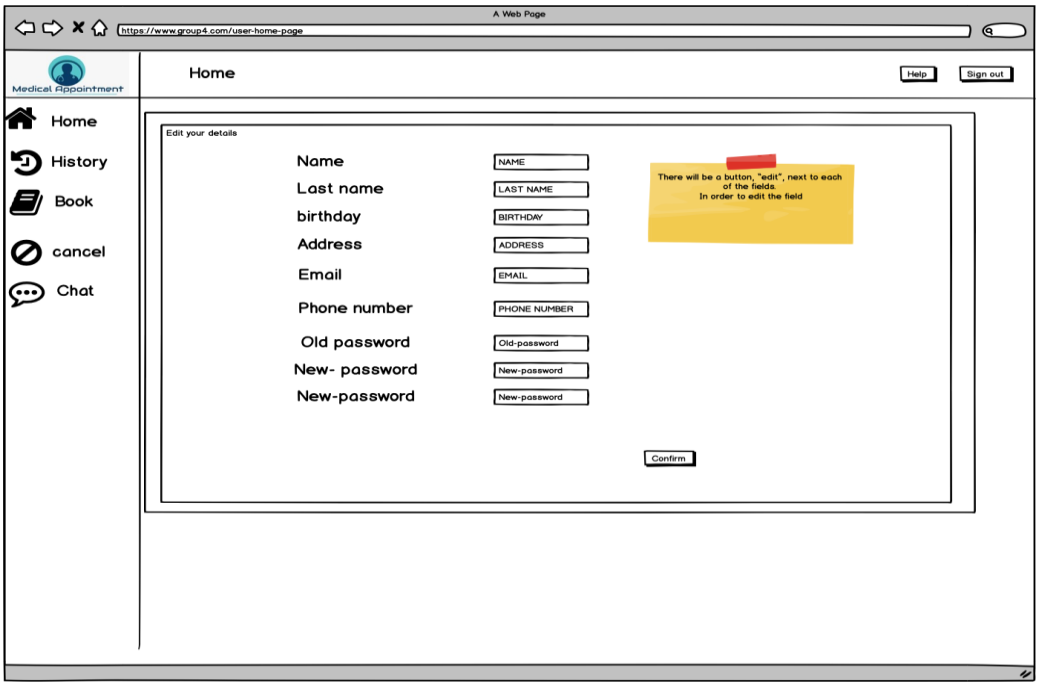




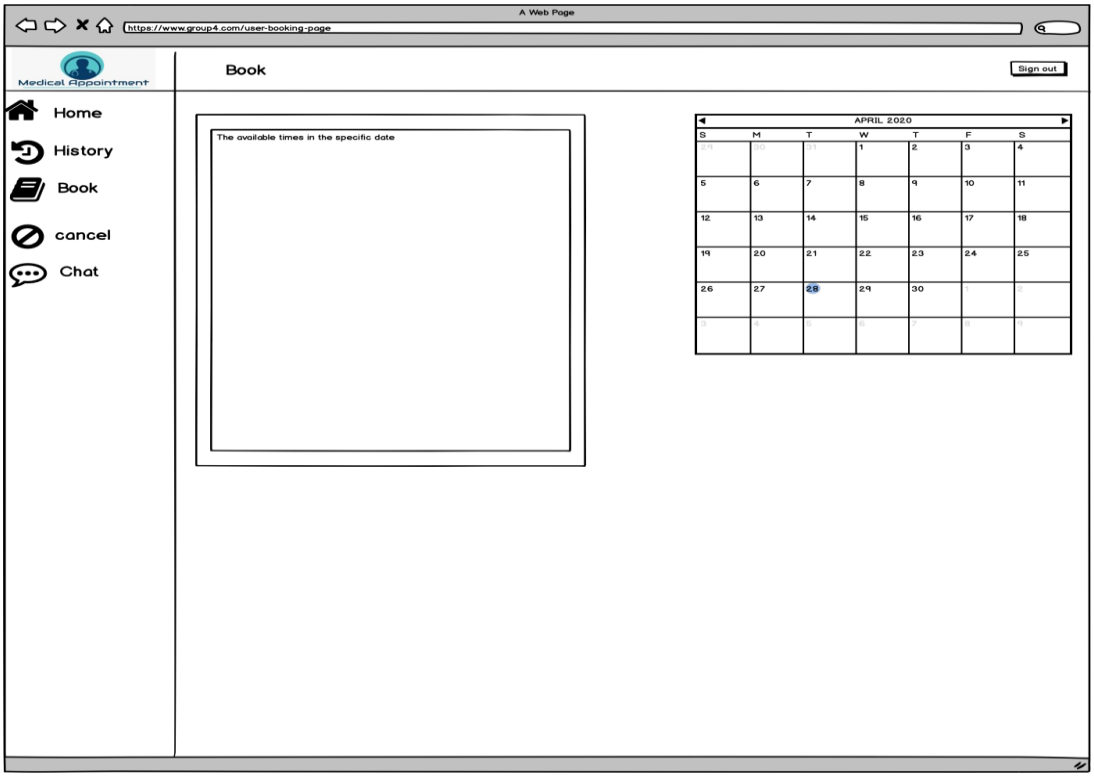
4. The user can sign in to his account to make a booking or make other changes if required. He does this by clicking the ‘Sign In’ button on the homepage.
   The user enters his login details like email id and password.
   The details are validated with the database and he is logged in if the correct details are enterd.



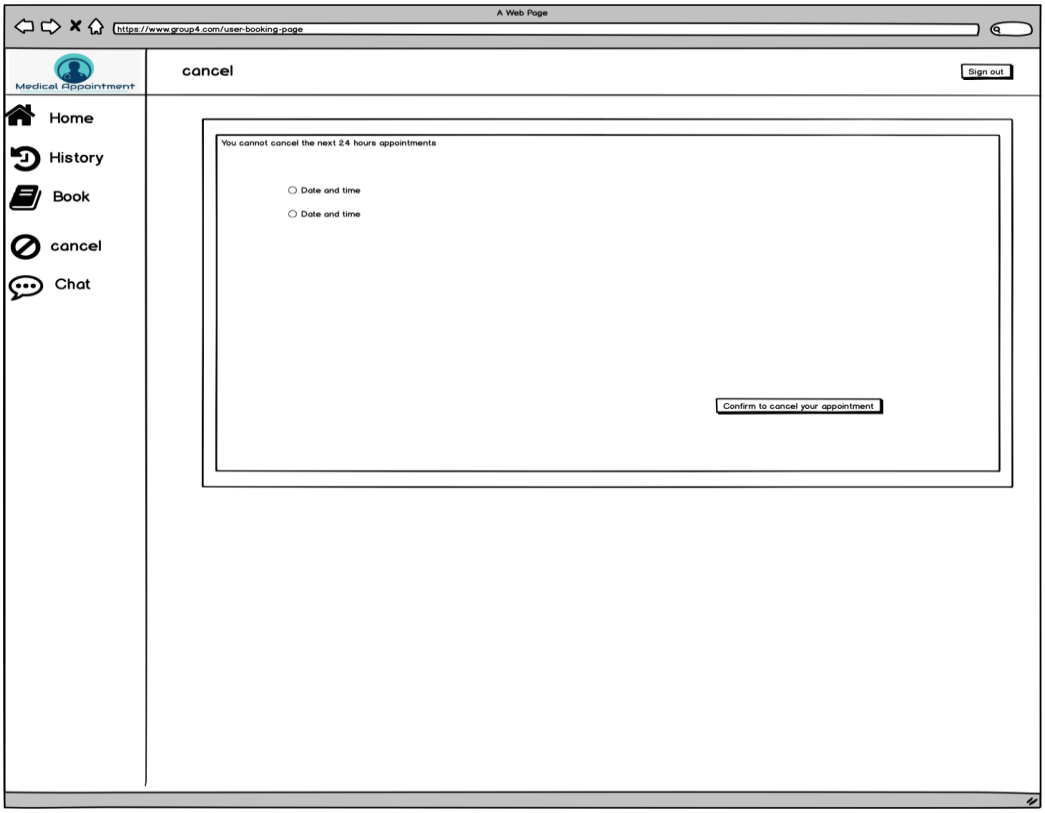
5. The user is navigated to the dashboard upon logging in to his account where he can make a booking, cancel a previous booking, view the history of his booking or edit his personal details. The user can edit his personal details in the ‘Home’



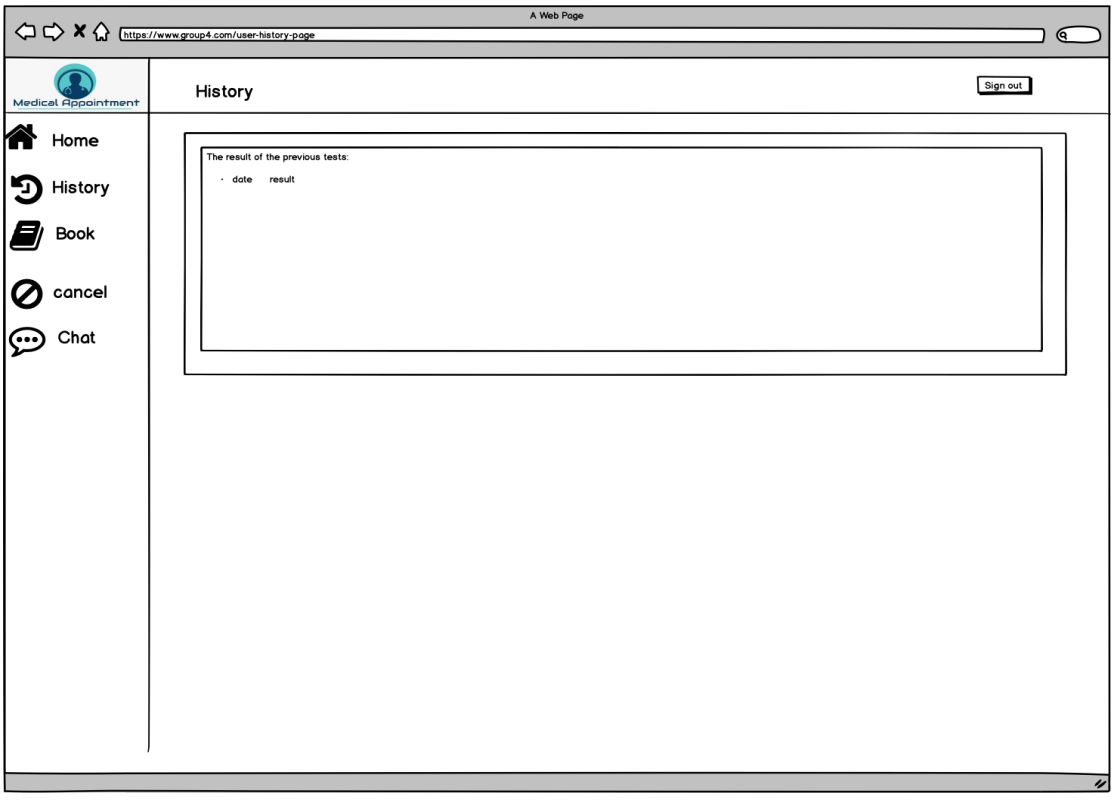
6. When the user selects the Book option, he is presented with a calendar containing a color scheme where red stands for slots full, yellow for almost full and green indicates that none of the slots are booked.
    - He then selects a day of his choice and he is presented with the available time slots in a box beside the calendar.
    - The user selects a slot and finally his booking is made.



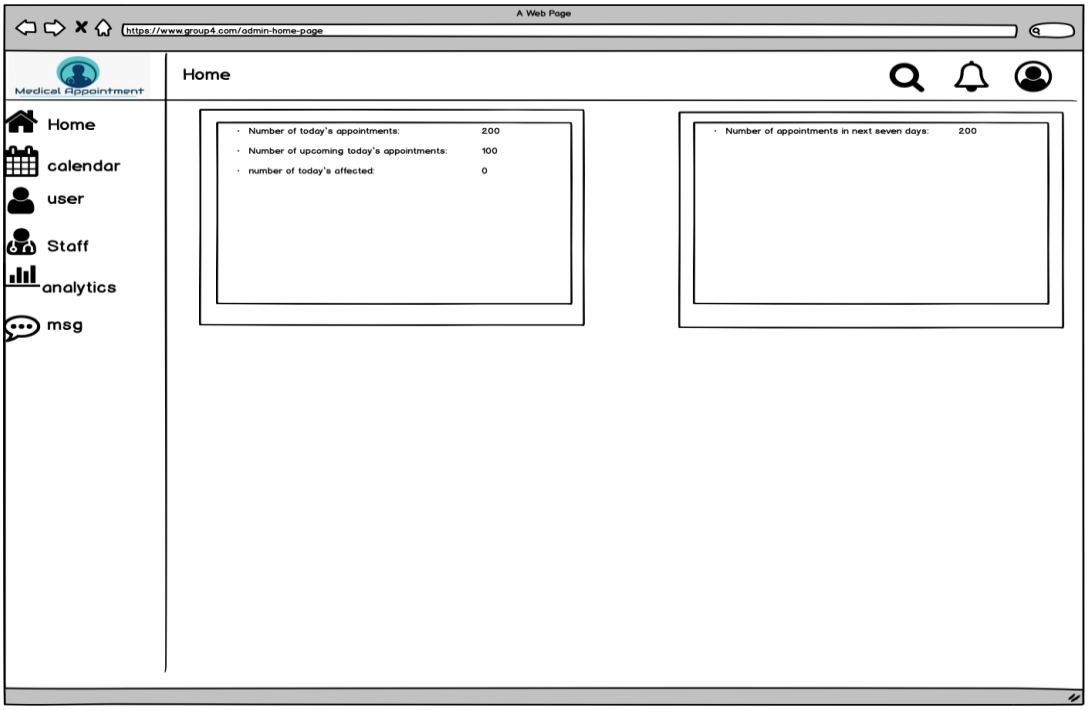
7. The user can cancel a previously made booking if he changes his mind by selecting the ‘Cancel’ button. A radio button is shown next to the date and time of the booking. He selects the desired booking and clicks the ‘Confirm to cancel your appointment’ button.



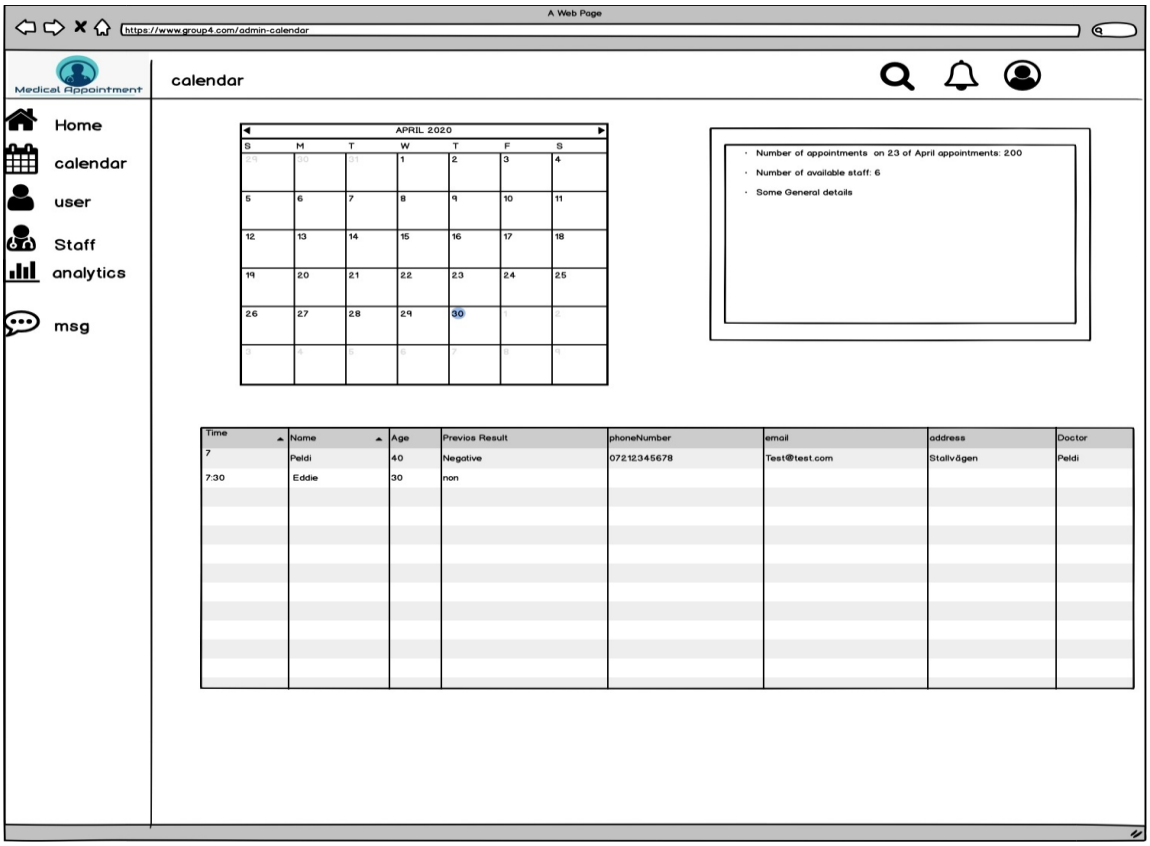
8. The user can see a history of his bookings by selecting the ‘History’ option.



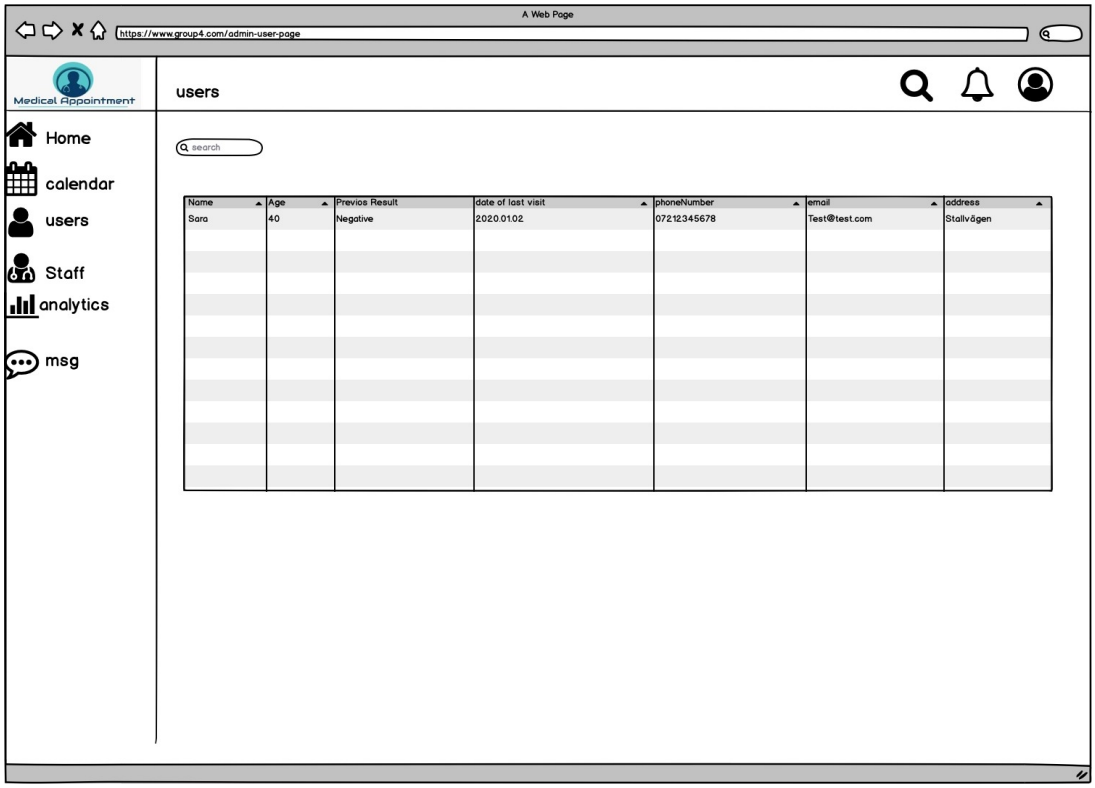
9. The admin can login to his account by selecting the ‘Sign in’ button on the homepage. The admin details are validated with the database and he is successfully logged in if the correct details are entered
    - The admin home consists of appointments and they are sorted by date.



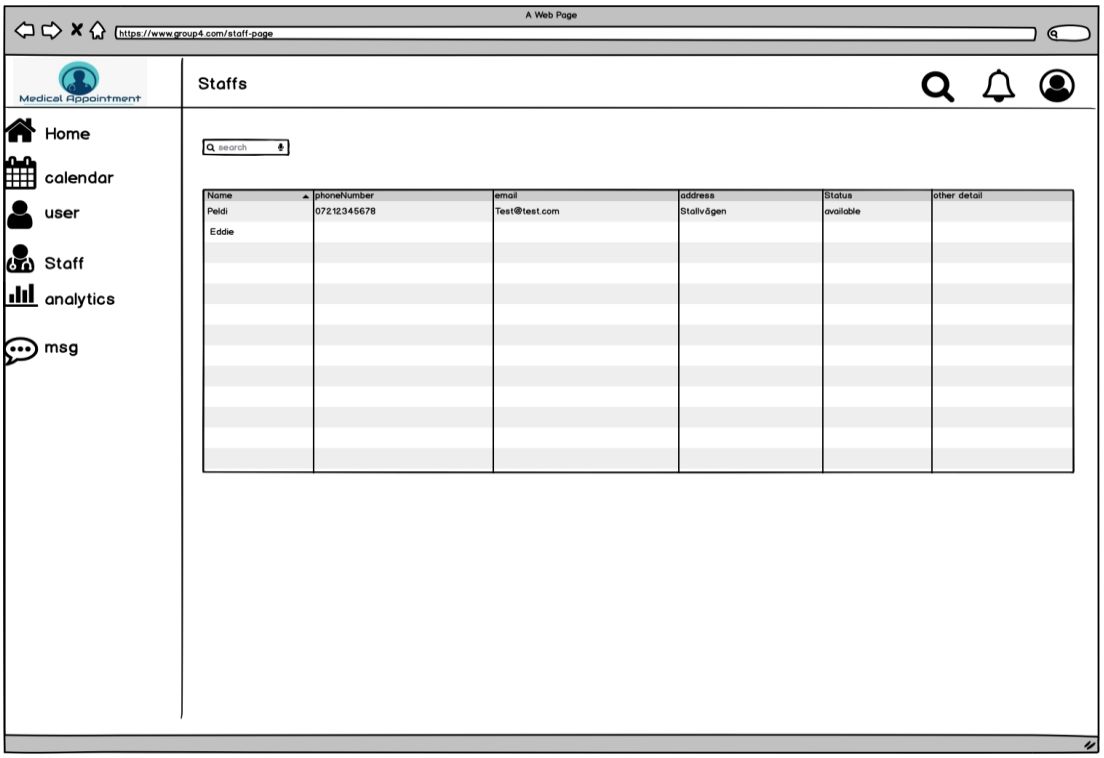
10. By selecting the calendar button the admin is presented with an organised view of the calendar, number of booking made and the details of the patients who made the bookings. He can assign a doctor to a patient as well.



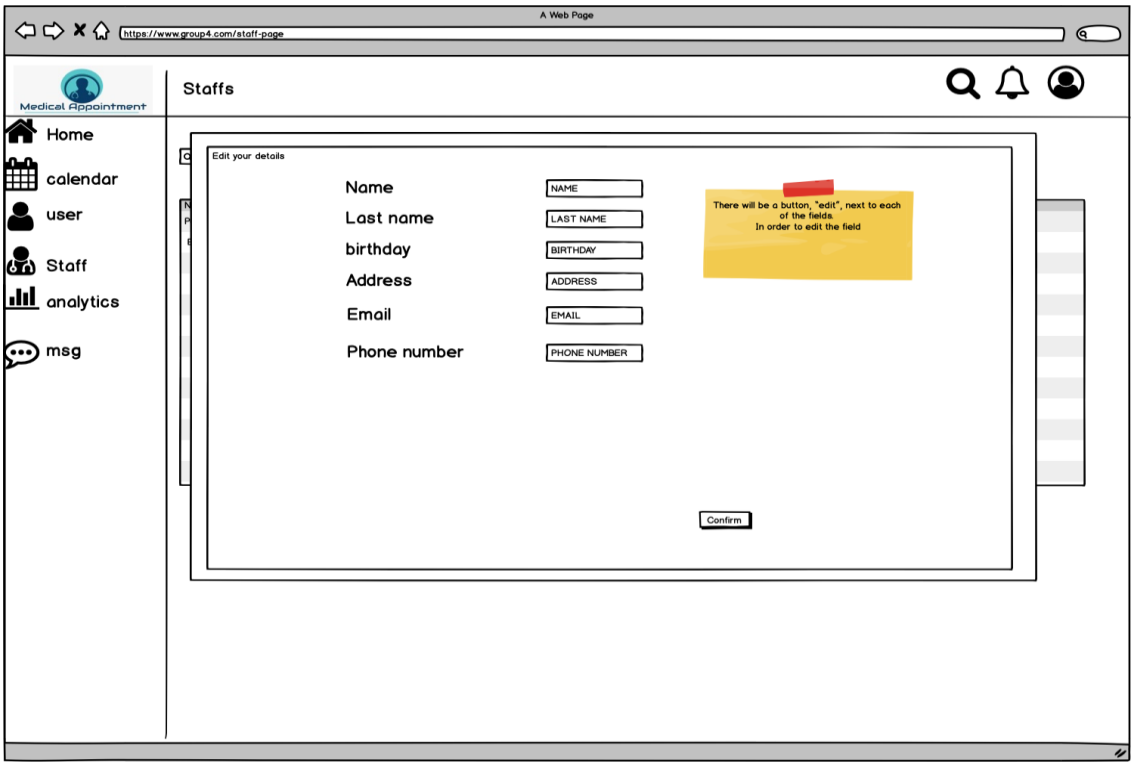
11. The Admin can view the list of users along with their personal details and the date of the booking they last made by selecting the ‘Users’ option.



12. By selecting the ‘Staff’ option, the Admin can view the list of the staff/ doctors working in the hospital along with their personal details.



Admin can also edit staff details by clicking on one of the names. A menu pops up and the Admin enters the new details and then selects the Confirm button. See the figure below.



Major design issues

Design Issues	Alternatives	Final decision
Platform	Should the medical appointment application be designed in Web or Android	Web Application
Database	Should the database be from Firebase or PostgreSQL	Firebase
Web Framework	Should it be Angular cli or React	React
Scalabilty	The web application was not scalable with the window.	It is fixed (scalable)

Software Architecture

This sections explains how the system behaves on a structural level, the set of decisions that were made and the design of the project.

- The framework used to create the web application was React js. React is a part of the JavaScript library for building user interfaces.
- The project does not contain Architectural pattern but we have divided the classes by creating sub directories for easy understanding, for switching between

classes and for code reusability. The src folder consists of components directory which contains the important components of the system.
The src folder also consists of other classes that deal with implementing the components in the web page like:

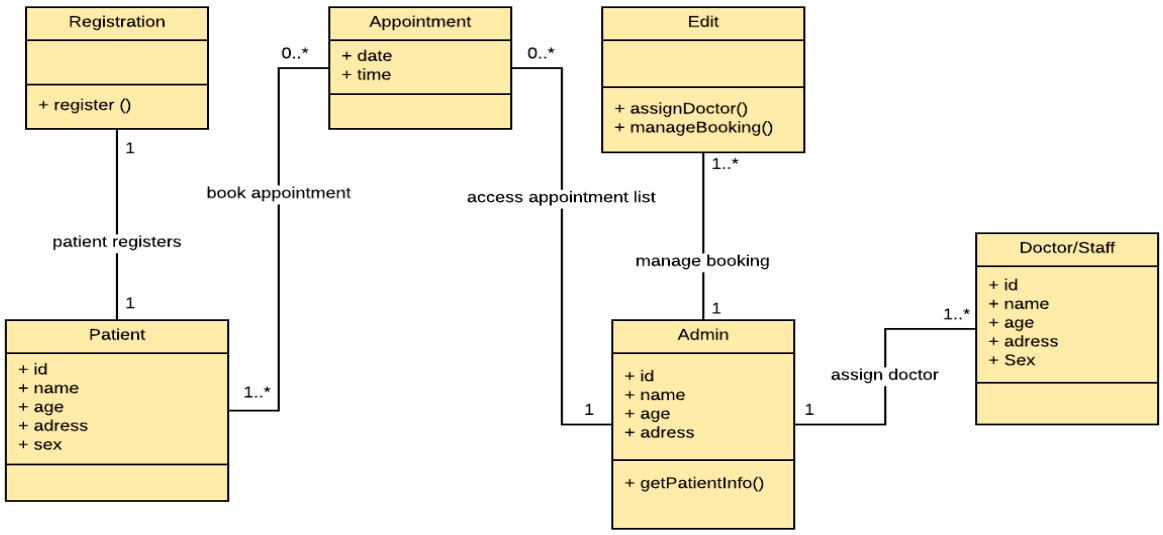
- App.js
- App.css
- Index.js
- Index.css

- The code is also easy to test while deploying the web application.
- The database used for this project is through Firebase. Firebase is form Google which provides applications with features like authentication, hosting of apps, cloud storage and realtime database.

Component Interface

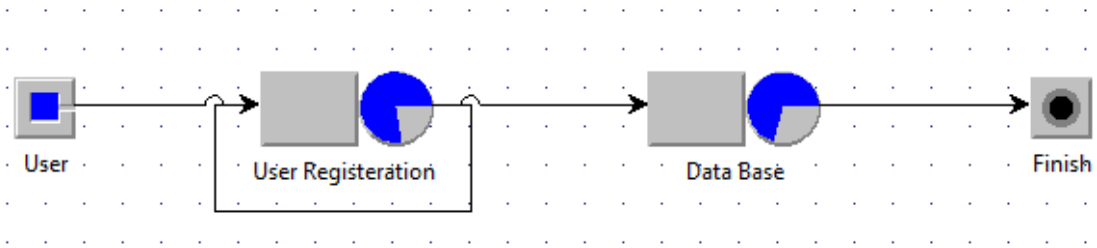
Function	**Component Interface**
Homepage containing the application style, layout and footer	Landing page
A navigation bar conaining the application name, logo and other options to getting started.	Navigation bar
User registration	Sign up
User log in	Sign in
User can take an online self assessment test to check for Covid-19	CovidTest
User can book an appointment for check up	Appointment
The admin can manage the booking and assign a doctor to the users (patients)	Edit
The application comes with a live chat to help users by answering their questions	LiveChat
The user data is stored in the database and is retrieved for validation	Database

UML Component Diagram



Performance modeling and evaluation

For this we considered the scenario where user can register with the system. The components involved here are user registration and database.
The performance model form JMT is shown below



The performance of the system is evaluated by making reasonal assumptions of the data.

User Registration

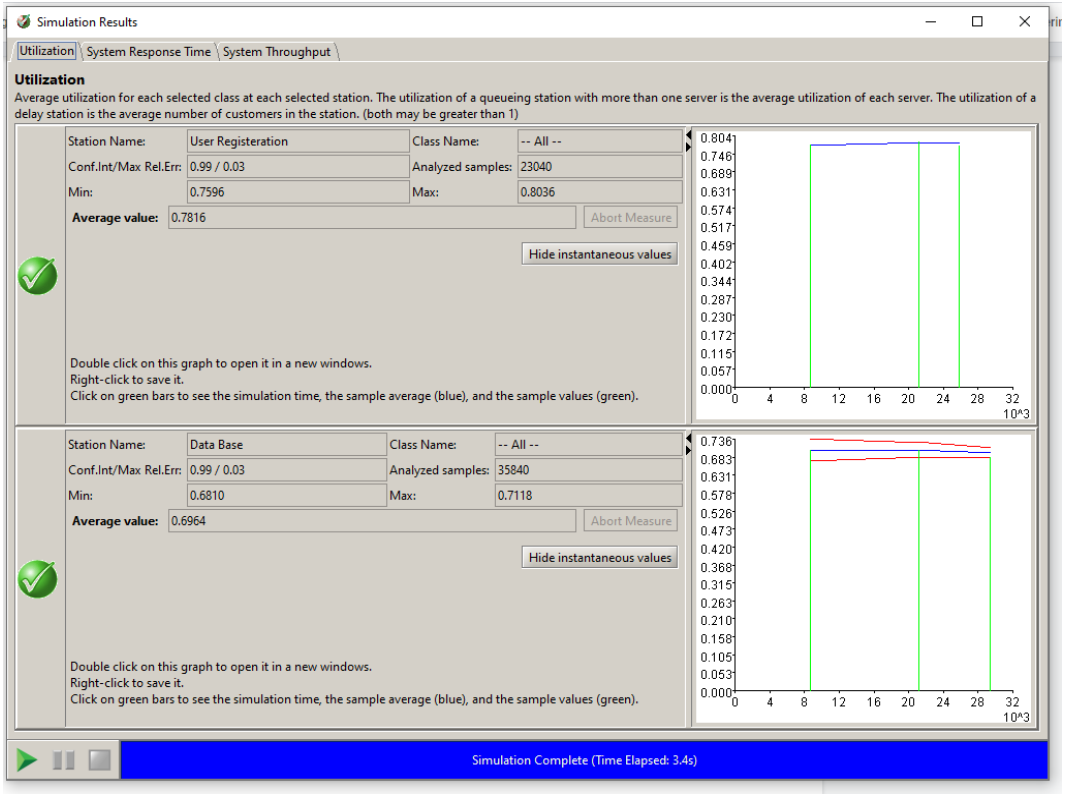
Arrival rate (λ)= 1/second
Service time (S)= 1.75
Resources= 1
Probabilistic routing= 0.9 towards Database if all details are correct
0.1 returns if there are errors in user details

Database

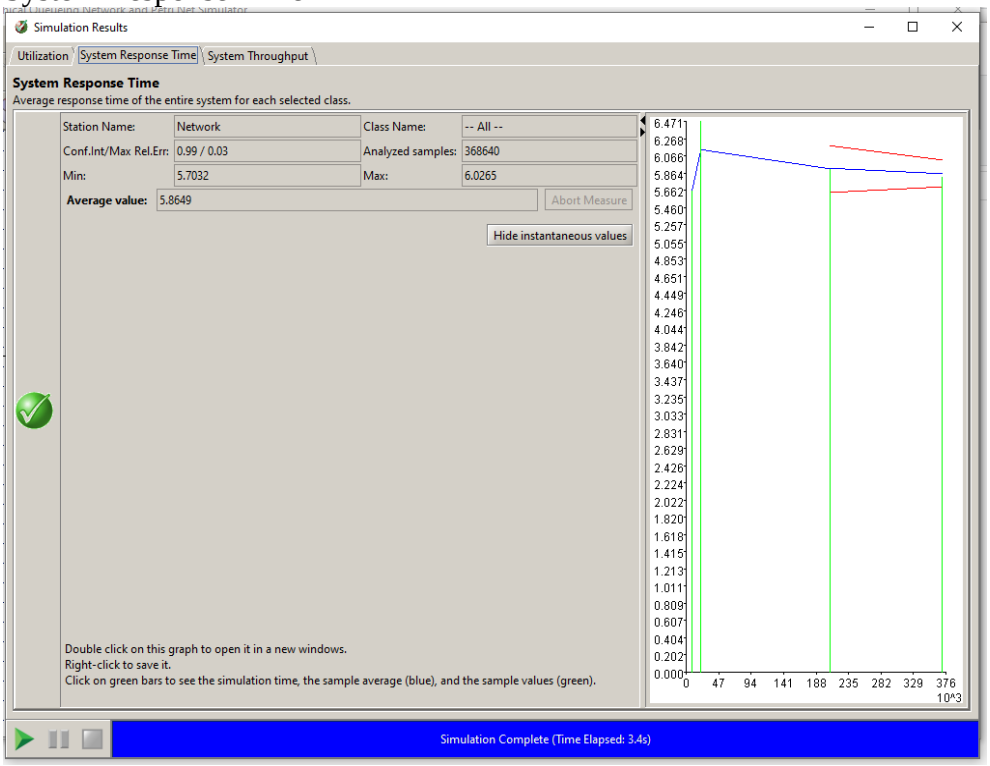
Arrival rate (λ)= 1/second
Service time (S)= 1.75
Resources= 1

The results after simulation form the JMT is shown below

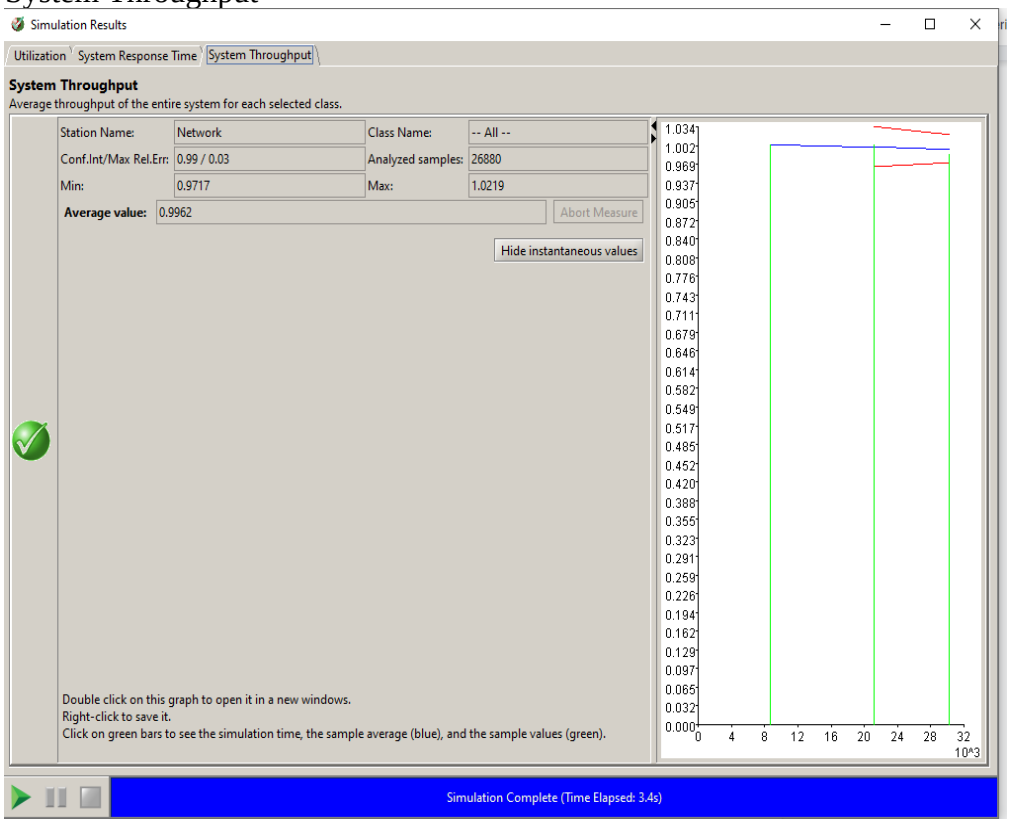
Utillization



System Response Time



System Throughput



Component Implementation

A few design principles were used to build the application. They are discussed below

- Design principle 1: Divide and Conquer
  The project was divided among the team members and each person received a component to work on. Hece, by dividing into subsystems it becomes much easier to understand the code.

- Design principle 2: Increase cohesion where possible
  The parts of the system that were related with each other were kept together in the same package for instance the App.js and the Index.js and the other components were separated. This improves understandability and is easier to change.

- Design principle 3: Reduce coupling
  The code does not contain coupling hence there are no interdependencies between one module and another.

- Design principle 4: Increase reusability where possible
  The design is simplified as much as possible and hence it is resuable.

UML class diagram