VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

SCHOOL OF COMPUTER SCIENCE ENGINEERING

APRIL 2019

HOSPITAL MANAGEMENT SYSTEM

PROJECT REPORT INTERNET AND WEB PROGRAMMING

Submitted by

VISHISHTH TALWAR	**16BCE0357**
STUTI AGARWAL	**16BCE0963**
VIPUL GOEL	**16BCI0015**

Under the guidance of faculty:

Prof. Balasubramaniam V

ABSTRACT

Population control has gone beyond the limit and public places are heavily crowded. One such place is Hospital's now. Waiting in line to get the number and visit the doctor is very tiresome since these places are overflowed with patients which creates problems for patients who suffers from serious illness and need immediate treatment.

In our project we gives a web application solution where patient can register himself online and can see free number of rooms free, so the rooms can be allowed to the respective patient's at ease.

The management has the ability to see the number of patients in the waiting list and can assign a room for the patient. The patient diagnosed with the higher threat of health will be given higher priority based on degree of seriousness of disease. So the patient can be given treatment immediately, while if the score is less the system alerts the management about the patient.

1. INTRODUCTION

1.1 AIM:

To design and develop a web application for managing hospitals rooms and determining the patient's priority based on seriousness of disease to fast track the treatment process and to give patients for more privacy to give better treatment. The app provides a centralised hub for managing the patients and planning their distribution across hospital’s rooms.

It allows management personnel’s to keep track of the patients recovery and their diseases in real time and to have an overview over the patients and rooms, and better manage the rooms assignment across patients.

1.2 OBJECTIVE:

Our main objective is to make process of treatment and patients management hassle free.To accomplish this, we will make a database for the Hospital. The purpose of it is to give the management an ease to find the information about patients, rooms availability, type of disease & score of the disease at one place. It provides a complete database of the whole hospital. It is for the efficient working and for getting all information at one place with only a few clicks. We used nodejs to create a database in MongoDb which links the back-end with the front-end design. To create front-end we use HTML, CSS and JavaScript.

2. REQUIREMENTS

2,1 BACKEND

- Nodejs
- Database-MongoDB

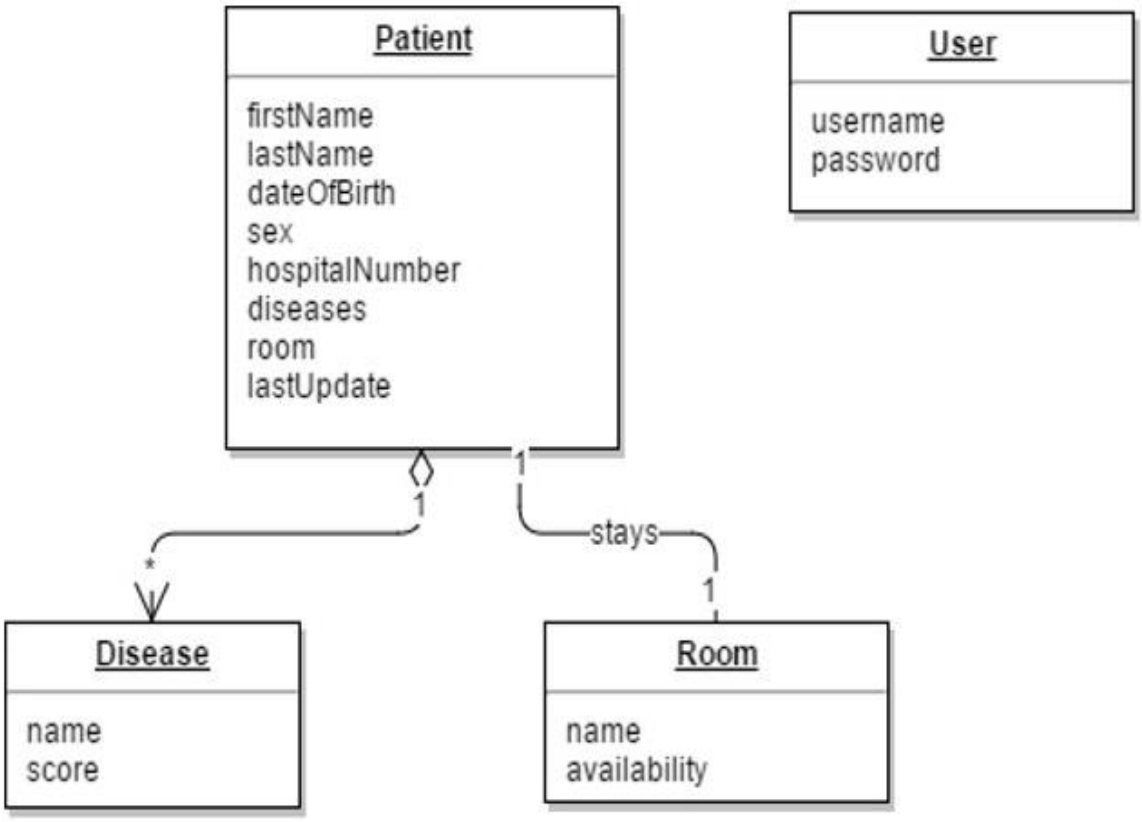
2.2 FRONTEND

- HTML
- CSS
- JQuery

3. FUNCTIONALITIES

Functionalities	Patients	Diseases	Rooms
- login system	- add / delete patients	- add / delete diseases	- assign rooms to patients
- add users	- update patient's diagnosis	- assign disease to patients	- add / remove rooms
-view dashboard	- view patient's page		
-	-retrieve patient's information		

DATABASE SCHEMA



CODE

addPatient.hsb

```
<link rel="stylesheet" type="text/css" href="/css/dataTables.bootstrap.min.css">
<link rel="stylesheet" href="/css/jquery.dataTables.min.css">
<div class="col-md-6 col-lg-6">

   <form action="/app/addpatient"
      method="post"> <!-- First name input -->

      <div class="form-group row">
         <label class="col-sm-3 col-md-3 col-lg-3 col-form-label">First Name</label>
         <div class="col-sm-9 col-md-9 col-lg-9">

            <input id="first-name" class="form-control" style="width: 100%;"
type="text" name="firstName" required="true" placeholder="Enter first name...">

         </div>
      </div>
      <!-- Last name input -->
      <div class="form-group row">
         <label class="col-sm-3 col-md-3 col-lg-3 col-form-label">Last
         Name</label> <div class="col-sm-9 col-md-9 col-lg-9">

            <input class="form-control" style="width: 100%;" type="text"

name="lastName" required="true" placeholder="Enter last name...">

         </div>

      </div>

      <!-- Hospital number input -->


      <div class="form-group row">

         <label class="col-sm-3 col-md-3 col-lg-3 col-form-label">Hospital
         no.</label>

         <div class="col-sm-9 col-md-9 col-lg-9">
```

```
<input class="form-control" style="width: 100%;" type="text" name="hospitalNumber"
required="true" placeholder="Enter Hospital no...">

                </div>
            </div>
            <!-- Date of birth -->
            <div class="form-group row">

                <label class="col-sm-3 col-md-3 col-lg-3 col-form-label">Date of birth</label>

                <div class="col-sm-9 col-md-9 col-lg-9">

                    <input class="form-control" style="width: 100%;" type="text"
                    name="dateOfBirth" required="true" placeholder="DD/MM/YYYY">

                    </div>

            </div>

            <!-- Sex -->

            <div class="form-group row">

                <label class="col-sm-3 col-md-3 col-lg-3 col-form-label">Sex</label>
                <div class="col-sm-9 col-md-9 col-lg-9">

                <div class="btn-group" data-toggle="buttons">

                    <label class="btn btn-primary active" style="margin:0px;">
                      <input id="male-radio" type="radio" name="sex"
                      value="male"autocomplete="off" checked>Male

                    </label>
                    <label class="btn btn-primary" style="margin:0px;">
                      <input id="female-radio" type="radio" name="sex"value="female"
        autocomplete="off"> Female

                    </label>
                  </div>
```

```
        </div>

      </div>

        <table id="add-new-patient" class="table table-striped table-
bordered" cellspacing="0" width="100%">

        <!-- JavaScript works here! -->

        </table>

        <input class="btn btn-success" style="float: right;" type="Submit"
        name="Submit" value="Add patient">

   </form>

</div>

<script type="text/javascript" src="/js/jquery.dataTables.min.js"></script>
<script type="text/javascript" src="/js/dataTables.bootstrap.min.js"></script>
<script type="text/javascript" src="/scripts/addpatient.js"></script>
```

dashBoard.hsb

```
<link rel="stylesheet" type="text/css" href="/css/dataTables.bootstrap.min.css">
<link rel="stylesheet" href="/css/jquery.dataTables.min.css">

<!-- <link rel="stylesheet" type="text/css" href="/css/bootstrap.min.css"> -->

<div class="row">
   <div class="col-lg-4 col-md-4">
      <div class="panel panel-primary">
         <div class="panel-heading"
            id="patientsInRooms"> <div class="row">
               <div class="col-xs-3">

                  <i class="fa fa-hotel fa-5x"></i></div>
                  <div class="col-xs-9 text-right">
```

```
                <div id="patients-with-rooms-live" class="huge"></div>
                <div>Patients with rooms</div>

                </div>

                </div>

                </div>
    </div>
</div>
<div class="col-lg-4 col-md-4">
    <div class="panel panel-red">
      <div class="panel-heading">
          <div class="row">
            <div class="col-xs-3">
              <i class="fa fa-child fa-5x"></i>
            </div>
            <div class="col-xs-9 text-right">
              <div id="patients-waiting-live"class="huge"></div>
              <div>Patients waiting</div>

            </div>

          </div>

       </div>

    </div>

</div>
<div class="col-lg-4 col-md-4">
    <div class="panel panel-orange">
       <div class="panel-heading" id="freeRooms">
          <div class="row">
            <div class="col-xs-3">
              <i class="fa fa-hospital-o fa-5x"></i>
            </div>
```

```
                    <div class="col-xs-9 text-right">
                        <div id="free-rooms-live" class="huge"></div>
                        <div>Free rooms</div>
                    </div>
                    </div>
                    </div>
                    </div>
                    </div>
                    </div>
        <!-- Table with patients in rooms-->

        <div style="padding-left:0px;" class="col-lg-6 col-md-6">

        <table id="patients-in-hospital" class="table table-striped table-bordered"
        cellspacing="0" width="100%"></table> </div>

        <!-- Table with patients waiting to get a room->
        <div class="col-lg-4 col-md-4">
<table id="patients-waiting" class="table table-striped table-bordered"cellspacing="0"
width="100%"></table>

</div>

        <!-- Table with free rooms-->

        <div style="padding-right:0px;" class="col-lg-2 col-md-2">

            <table id="free-rooms" class="table table-striped table-bordered" cellspacing="0"
            width="100%"></table>

            </div>
        <!-- <script type="text/javascript" src="/js/jquery-1.12.4.js"></script> -->
        <script type="text/javascript" src="/js/jquery.dataTables.min.js"></script>
        <script type="text/javascript" src="/js/dataTables.bootstrap.min.js"></script>
        <script type="text/javascript" src="‘/scripts/dashboard.js"></script>
```

Login.hbs

```
<!DOCTYPE html>

<html>

   <head>

      <meta charset="utf-8">

      <title>LoginPage</title>

      <link rel="stylesheet" href="/css/stylelogin.css">

      <link rel="stylesheet" href="/css/bootstrap.css" />

      <!-- <link rel="stylesheet" href="/css/style.css" /> -->

      <!-- <link rel="stylesheet" href="/css/login.css"> -->

   </head>

   <body>

      <div class="container-fluid">

         <div class="card card-container">

            <img id="profile-img" class="profile-img-card" src="/images/nhs.png"/>
            <p id="profile-name" class="profile-name-card"></p>

            <form method="post" action="/login" class="form-signin">
               <span id="reauth-email" class="reauth-email"></span>
               <input type="text" id="inputEmail" class="form-control"
               name="username" placeholder="username" required
               autofocus>

               <input type="password" id="inputPassword" class="form-control"
name="password" placeholder="Password" required>

               <button class="btn btn-lg btn-primary btn-block btn-signin"
               type="submit">Sign in</button>

            </form><!-- /form -->

         </div><!-- /card-container -->

      </div><!-- /container -->
```

```
    </body>

</html>
```

System.hhbs

```
<link rel="stylesheet" type="text/css" href="/css/dataTables.bootstrap.min.css">
<link rel="stylesheet" href="/css/jquery.dataTables.min.css">

    <!-- Diseases manager -->

    <div class="col-md-4 col-lg-4" style="padding-right: 30px; padding-left:
        30px;"> <div class="row">

            <!-- Add new disease -->

            <h4 style="margin-top: 0px;">Add new disease in the
            system</h4> <form action="/app/adddisease" method="post">

                <div class="form-group row">

                    <label class="col-sm-5 col-md-5 col-lg-5 col-form-label">Disease
name</label>

                    <div class="col-sm-7 col-md-7 col-lg-7">

                        <input class="form-control" style="width: 100%;" type="text"
name="diseaseName" required="true" placeholder="Enter new disease name...">

                    </div>

                </div>

                <!-- Last name input -->

                <div class="form-group row">

                    <label class="col-sm-5 col-md-5 col-lg-5 col-form-label">Disease
score</label>

                    <div class="col-sm-7 col-md-7 col-lg-7">

                        <input class="form-control" style="width: 100%;" type="number"
name="diseaseScore" required="true" placeholder="Enter new disease score...">

                    </div>
```

```
                    </div>
<input class="btn btn-success" style="float: right;" type="Submit" name="Submit"
value="Add disease">
                </form>
            </div>
            <hr style="border-top: dotted 1px;" />
            <div class="row">
            <!-- Delete diseases -->
                <h4 style="margin-top: 0px;">Delete diseases from the
                system</h4> <form action="/app/deletediseases" method="post">
                    <table id="diseases-table" class="table table-striped table-bordered"
        cellspacing="0" width="100%">
                        <!-- JavaScript works here! -->
                    </table>
<input class="btn btn-danger" style="float: right;" type="Submit" name="Submit"
value="Delete diseases">
</form>
</div>
</div>
<!-- Rooms manage -->
            <div class="col-md-4 col-lg-4" style="padding-right: 30px; padding-left:
                30px;"> <!-- Add new room -->
                <div class="row">
                    <h4 style="margin-top: 0px;">Add new room in the system</h4>
                    <form action="/app/addroom" method="post">
```

```
                    <div class="form-group row">

                <label class="col-sm-5 col-md-5 col-lg-5 col-form-label">Room name</label>

<div class="col-sm-7 col-md-7 col-lg-7">

<input class="form-control" style="width: 100%;" type="text" name="roomName"
required="true" placeholder="Enter new room name...">

</div>

</div>

<input class="btn btn-success" style="float: right;" type="Submit" name="Submit"
value="Add room">

</form>

</div>

<hr style="border-top: dotted 1px;" />

<!-- Delete rooms -->

<div class="row">

<h4 style="margin-top: 0px;">Delete rooms from the system</h4> <form
action="/app/deleterooms" method="post">

<table id="rooms-table" class="table table-striped table-bordered" cellspacing="0"
width="100%">
<!-- JavaScript works here! -->
</table>
<input class="btn btn-danger" style="float: right;" type="Submit" name="Submit"
value="Delete rooms">
</form>
</div>
</div>
```

```
<div class="col-md-4 col-lg-4" style="padding-right: 30px; padding-left:
    30px;"> <div class="row">

        <h4 style="margin-top:0px">Add a new user</h4>
        <form method="post" action="/app/adduser"
          center="true"> <div class="form-group row">

            <label class="col-sm-3 col-md-3 col-lg-3 col-form-
            label">Username</label>

            <div class="col-sm-9 col-md-9 col-lg-9">

                <input class="form-control" style="width: 100%;" type="text"
name="username" required="true" placeholder="Enter username...">

            </div>
          </div>
          <div class="form-group row">
            <label class="col-sm-3 col-md-3 col-lg-3 col-form-
            label">Password</label>
            <div class="col-sm-9 col-md-9 col-lg-9">
<input class="form-control" style="width: 100%;" type="password" name="password"
required="true" placeholder="Enter password...">
            </div>

          </div>

<button type="submit" style="float: right;" class="btn btn-primary">Add user</button>

</form>

</div>

</div>

<script type="text/javascript" src="/js/jquery.dataTables.min.js"></script> <script
type="text/javascript" src="/js/dataTables.bootstrap.min.js"></script>
```

<script type="text/javascript" src="/scripts/systemSettings.js"></script>

Disease.js

```
/*mongoDB Schema for diseases*/
const mongoose = require ('mongoose');
var DiseaseSchema = mongoose.Schema({
    name: {
        type: String,
        unique: true,
        required: true
    },
    score: {
        type: Number,
        required: true,
        default: 0
    }
});
var Disease = mongoose.model('Disease', DiseaseSchema);/*
Default diseases in the system

-> those will be added as soon as the system is live

-> if they are deleted from the system, and the system restarts, then they will be added
again in the system
*/

var scoreOfDisease = {}; // empty map
```

```
module.exports = {scoreOfDisease, Disease};
```

Paitent.js

```
const mongoose = require('mongoose');
const _ = require('lodash');

var {scoreOfDisease, Disease} = require('./diseases.js');
var rooms = require('./rooms.js');

// User Schema

var PatientSchema = mongoose.Schema({

        firstName: {

                type: String,

                required: true

        },

        lastName: {

                type: String,

                required: true

        },

        dateOfBirth: {

                type: String,


                required: true,},

        sex: {

                //true = male
                //false = female
                type: Boolean,
                required: true,
                default: true
        },

        hospitalNumber: {
```

```
                type: String,
                required: true,
                unique: true},
        diseases: {
      type: Array,
      default: []
    },

    score: {
      type: Number,
          required: true,
          default: 0
    },
        room: {
                type: String,
                required: true,
                default: 'noroom'
        },
        lastUpdate: {
                type: Number,
                required: true}});
                /*
        function to update the diseases and the score of a patient
        *Requires the patient to have the diseases already saved in the databases*/
PatientSchema.methods.updateScore = function () {
        var patient = this;
```

```
        //promise to get the patient object inside the diseases callback
        var promise = new Promise(function(resolve, reject) {
                resolve(patient);

                reject(patient);})
        Promise.all([promise.then(function (patient) { return patient;
}), Disease.find({})])

      .then((data) => {

        var patient = data[0];

        var diseases = data[1];

        var scoreOfDisease = {};

        var score = 0;

                 if (! _.isEmpty(diseases) && _.isArray(diseases)) {

          //create a hashmap with the diseases and their scores
          for (var i = 0; i < diseases.length; ++i) {
            scoreOfDisease[diseases[i].name] = diseases[i].score;}

                for (var i = 0; i < patient.diseases.length; ++i) {

                 if (scoreOfDisease[patient.diseases[i]] > score) {

                 score = scoreOfDisease[patient.diseases[i]];

                 }

                }}

        patient.score = score;

        patient.save().catch((err) => {

        console.log(err);
```

```
                    });
        }).catch((err) => {
            console.log(err);
            })
}
var Patient = mongoose.model('Patient',
PatientSchema); module.exports = {Patient};
const mongoose = require('mongoose');
const _ = require('lodash');
var {scoreOfDisease, Disease} = require('./diseases.js');
var rooms = require('./rooms.js');
// User Schema
var PatientSchema = mongoose.Schema({
        firstName: {
                type: String,
                required: true,
        },
        lastName: {
                type: String,
                required: true
        },
        dateOfBirth: {
                type: String,
                required: true,
        },
        sex: {
                //true = male
                //false = female
                type: Boolean,
```

```
                required: true,
                default: true,},
            hospitalNumber: {
                type: String,
                required: true,
                unique: true
            },
diseases: {
            type: Array,
            default: []
          },
          score: {
            type: Number,
              required: true,
              default: 0,
          },
            room: {
                type: String,
                required: true,
                default: 'noroom',
            },
            lastUpdate: {
                type: Number,
                required: true,
            }
    });
    /*
        function to update the diseases and the score of a patient
        *Requires the patient to have the diseases already saved in the databases
```

```
        */

        PatientSchema.methods.updateScore = function () {
            var patient = this;

            //promise to get the patient object inside the diseases callback
            var promise = new Promise(function(resolve, reject) {
                resolve(patient);

                reject(patient);})

            Promise.all([promise.then(function (patient) { return patient;
        }), Disease.find({})])

            .then((data) => {

              var patient = data[0];

              var diseases = data[1];

              var scoreOfDisease = {};

              var score = 0;

if (! _.isEmpty(diseases) && _.isArray(diseases)) {

                  //create a hashmap with the diseases and their scores
                  for (var i = 0; i < diseases.length; ++i) {
                     scoreOfDisease[diseases[i].name] = diseases[i].score;

                  }

for (var i = 0; i < patient.diseases.length; ++i) {

if (scoreOfDisease[patient.diseases[i]] > score) {

score = scoreOfDisease[patient.diseases[i]];

}

}

}
```

```
patient.score = score;
patient.save().catch((err) => {
console.log(err);
});
}).catch((err) => {
console.log(err);
});
}
```

```
var Patient = mongoose.model('Patient',
PatientSchema); module.exports = {Patient};
```

Room.js

```
/*
   mongoDB Schema for rooms
   false = free room
   true = occupied room
*/
const mongoose = require ('mongoose');
var RoomSchema = mongoose.Schema({
name: {
type: String,
unique: true,
required: true,
},
availability: {
```

```
type: Boolean,
required: true,
default: false
}
});
var Room = mongoose.model('Room', RoomSchema);
var rooms = {};
rooms["noroom"] = false;
/*Function to put the default diseases in the system*/
        function populateDatabase () {
            for (prop in rooms) {
                var room = Room({
                    name: prop,
                    availability: rooms[prop]});
                            //simply save the default room in the
                system room.save().then((disease) => {
                                    //do nothing
                            }, (err) => {


                                    // do nothing});}}
```

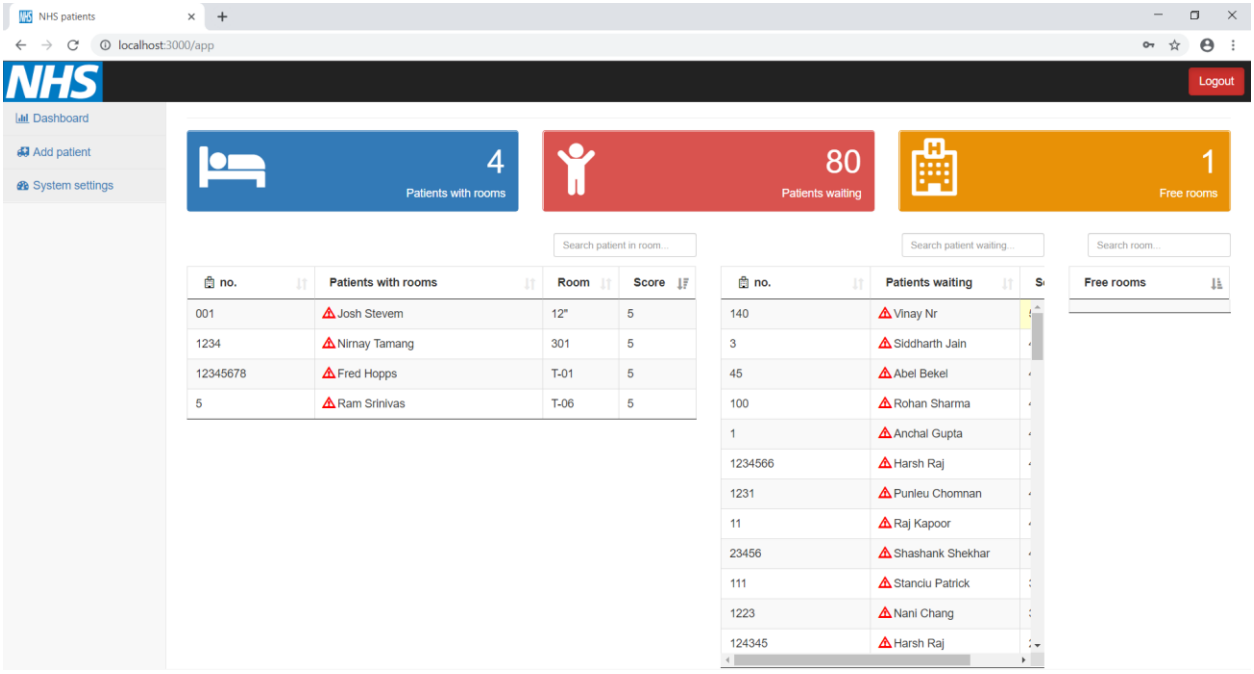
```
populateDatabase();

module.exports = {rooms, Room};
```

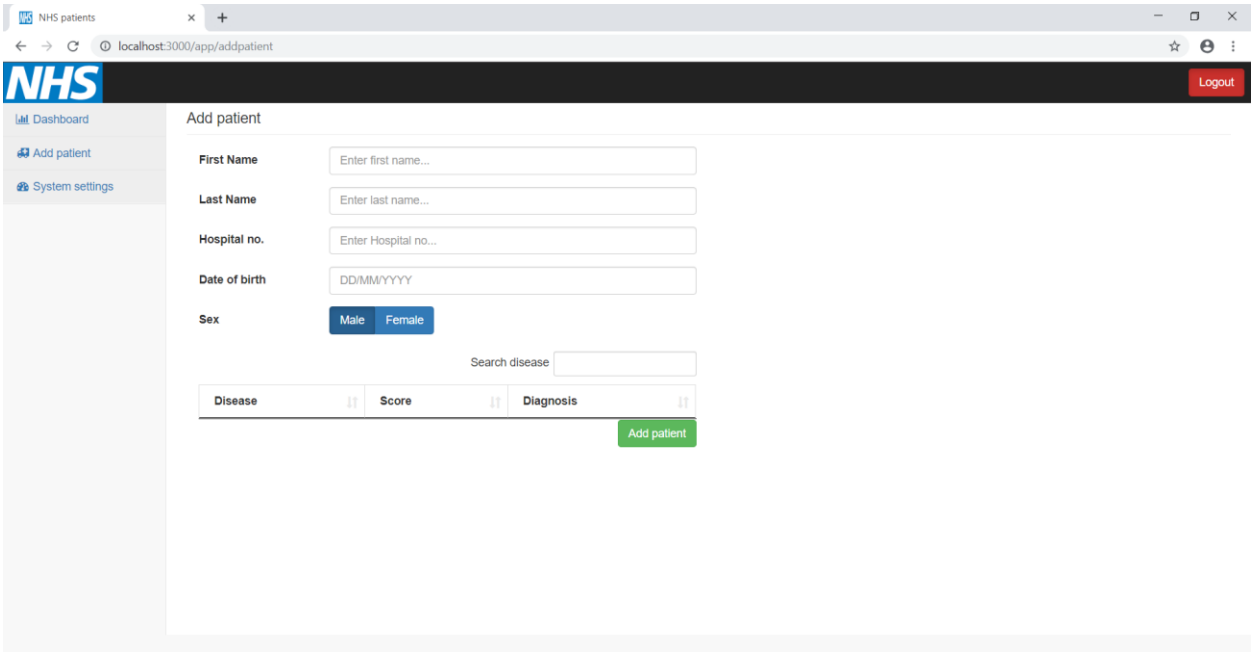
User.js

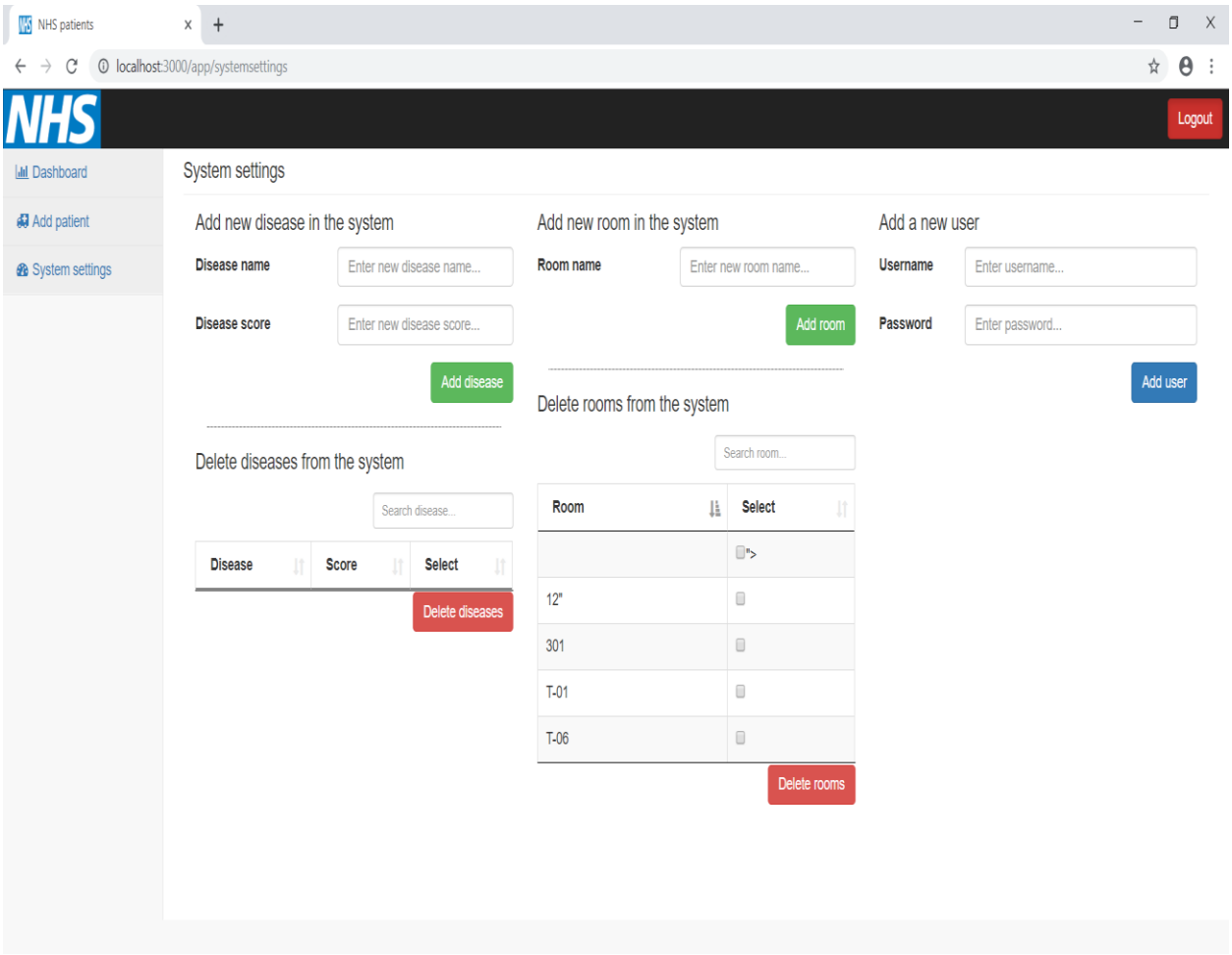
```
/*

        Define the authentification system and the user model.

        ***Password are stored encrypted using Bcrypt algorithm.

*/

const mongoose = require ('mongoose');

const bcrypt = require ('bcryptjs');
var UserSchema = mongoose.Schema({

        username: {

                type: String,

                required: true,

                unique: true

        },

        password: {

                type: String,

                required: true

        },
});
// define the model User to be added in the database

var User = module.exports = mongoose.model('User', UserSchema);
/*CREATE ADMIN ACCOUNT*/
```

```
var adminUser = new User({

        username: 'admin',

        password: 'admin'

});

createUser(adminUser, function (aux1, aux2)
        { // do nothing});

/*

        Create new User in the system

*/

//hash the password when creating a new user in the database
function createUser(newUser, callback) {
        bcrypt.genSalt(10, function(err, salt) {
                bcrypt.hash(newUser.password, salt, function(err, hash) {
                        //store hash newUser.password = hash;
                        newUser.save(callback); // the callback of save is
                        afunction(err, user)

                });

        });

}

module.exports.createUser = createUser;

module.exports.getUserByUsername = function(username, callback) {
        User.findOne({

username: username

        }, callback);

}

module.exports.getUserById = function(id, callback)
        { User.findById(id, callback);

}
module.exports.comparePassword = function(candidatePassword, hash, callback)
        { bcrypt.compare(candidatePassword, hash, function(err, isMatch) {

                if (err) {

                        throw err;}

                callback(null, isMatch);});};
```

SCREENSHOTS




NHS patients
localhost:3000/app/addpatient
NHS
Logout
Dashboard
Add patient
System settings
Add patient
First Name
Enter first name...
Last Name
Enter last name...
Hospital no.
Enter Hospital no...
Date of birth
DD/MM/YYYY
Sex
Male
Female
Search disease
Disease
Score
Diagnosis
Add patient


NHS patients
localhost:3000/app/systemsettings
NHS
Logout
Dashboard
Add patient
System settings
System settings
Add new disease in the system
Disease name
Enter new disease name...
Disease score
Enter new disease score...
Add disease
Delete diseases from the system
Search disease...
Disease
Score
Select
Delete diseases
Add new room in the system
Room name
Enter new room name...
Add room
Delete rooms from the system
Search room...
Room
Select
12"
301
T-01
T-06
Delete rooms
Add a new user
Username
Enter username...
Password
Enter password...
Add user

CONCLUSION

Hospital Management System provides the benefits of streamlined operations, enhanced administration & control, superior patient care, strict cost control and improved profitability. Hospital Management System is powerful, flexible, and easy to use and is designed and developed to deliver real conceivable benefits to hospitals. More importantly it is backed by reliable and dependable support.

Hospital Management System is custom built to meet the specific requirement of the mid and large size hospitals across the globe. All the required modules and features have been particularly built to fit into the requirements. It covers all the required modules right from Patient Registration, add / delete diseases, assign rooms to patients, update diagnosis and add / remove rooms.

REFERENCES

1. McMahon, G. T., Gomes, H. E., Hohne, S. H., Hu, T. M. J., Levine, B. A., & Conlin, P. R. (2005). Web-based care management in patients with poorly controlled diabetes. *Diabetes care*, *28*(7), 1624-1629.
2. Kuhn, K. A., & Giuse, D. A. (2001). From hospital information systems to health information systems. *Methods of information in medicine*, *40*(04), 275-287.
3. Ngai, E. W., Poon, J. K. L., Suk, F. F. C., & Ng, C. C. (2009). Design of an RFID-based healthcare management system using an information system design theory. *Information Systems Frontiers*, *11*(4), 405-417.
4. Yellin, S., & Singer, W. (2005). *U.S. Patent Application No. 11/037,842*.
5. http://w3schools.com
6. https://nodejs.org/
7. https://jquery.com/
8. https://www.mongodb.com/