FRANKFURT
UNIVERSITY
OF APPLIED SCIENCES

Project Documentation
Blood Pressure Logs Web App

By:
Muhammad Ehsan-Ul-Haq
Matriculation: 1098587
Date: 30/07/2015

Submitted To**:**
Mr. Damir Dobric
Prof. Dr. Andreas Pech

Software Engineering SS2015
Master Information Technology
Frankfurt University of Applied Sciences

Contents

1 Introduction

An estimated 1 in 4 adults suffer from hypertension. Managing and monitoring hypertension can help minimize the risk of its many complications such as heart failure, heart disease and stroke.

Blood Pressure logs app is dedicated to helping you manage high blood pressure. This web app provides you with a simple method of logging and tracking your blood pressure readings that is an important part of choosing and monitoring treatment with your doctor. Moreover, Smart Blood Pressure can help you track your progress and manage all your blood pressure measurements with an overall goal of improving your blood pressure.

Features

- ✓ It records your systolic, diastolic blood pressure along with pulse rate. Indeed you can add as many health related variables as you want.
- ✓ Quickly add notes (e.g. before dinner) and measurement information (e.g. sitting, left arm) using tags.
- ✓ Store and access all your blood pressure measurements anywhere and anytime online with just logging in to your account or using open authentication.
- ✓ Analyse the progress you are making using intuitive graphs and statistics (variability for a period of last seven days to last month). Filter the data analysed based on tags which can prove useful in analysing how well you are doing and if any lifestyle modifications or medications have proven effective.

Disclaimer

- ✓ This app may only be used as a tool to record, share and keep track of blood pressure measurements. This cannot measure blood pressure.
- ✓ This is NOT a substitute for a doctor or professional healthcare or advice. Any health related information provided is for informational purposes only and should not be used to replace the advice of healthcare professionals.
- ✓ This is meant to track your blood pressure records where each record is the systolic, diastolic, pulse, weight, date/time and notes/tags.

2 Objectives

2.1 Background

Blood pressure is the force exerted by circulating blood on the walls of blood vessels. The blood pressure is typically recorded as two numbers, written in a ration like - 115/75 which is read as "115 over 75 millimeters of mercury".

Systolic-The top number (115 as per the above sample), measures the pressure in the arteries when the heart beats (when the heart muscle contracts).

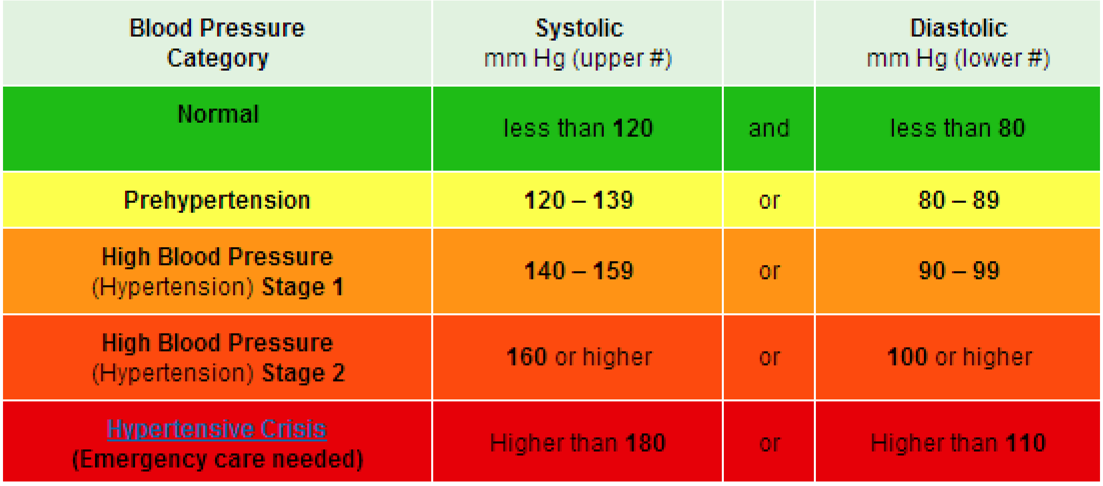
Diastolic-The bottom number (75 as per the above sample), measures the pressure in the arteries between heartbeats (when the heart muscle is resting between beats and refilling with blood). The blood pressure readings are categorized and considered to be normal , high or low based on the below readings.

Normal Blood Pressure - Blood pressure reading is below 120/80 is considered to be normal.

High Blood Pressure - Blood pressure of 140/90 or higher is considered to be high blood pressure. If either of the numbers are usually high, you have high blood pressure.

Low Blood Pressure - Blood pressure that is too low is known as hypotension. Some experts define low blood pressure as readings lower than 90 systolic or 60 diastolic.

Pulse rate- A normal resting heart rate for adult ranges from **60 to 100 beats a minute**. Generally, a lower heart rate at rest implies more efficient heart function and better cardiovascular fitness. For example, a well-trained athlete might have a normal resting heart rate closer to 40 beats a minute.



Blood Pressure Category	**Systolic** mm Hg (upper #)		**Diastolic** mm Hg (lower #)
Normal	less than **120**	and	less than **80**
Prehypertension	**120 – 139**	or	**80 – 89**
High Blood Pressure (Hypertension) **Stage 1**	**140 – 159**	or	**90 – 99**
High Blood Pressure (Hypertension) **Stage 2**	**160** or higher	or	**100** or higher
Hypertensive Crisis **(Emergency care needed)**	Higher than **180**	or	Higher than **110**

Fig: Blood pressure chart

3 Architecture

3.1 Abstract

This is a single page web app based on technologies like AngularJS, HTML05, CSS3 and Firebase cloud service. First of all it prompts the user to login to the page with user name and password. If user is not already registered then it asks the user to register with a valid email and password. It also provides the possibility to log in via third party accounts e.g. facebook, gamil, github, twitter etc. by using open authentication (OAuth). Once user is logged in it takes him to the main app where he can see the history of logs. There is a dialogue box where user can enter different values regarding blood pressure and pulse rate. User can add as many health related variables as he want by going in to projects, he can add/delete different projects. Moreover, he can analyse his blood pressure history in a graphical way by going into statistics.

3.2 Technical Architecture

3.2.1 General Structure

This project is mainly based on AngularJS and Firebase. **AngularJS** (commonly referred to as "**Angular**") is an open-source web application framework maintained by Google and by a community of individual developers and corporations to address many of the challenges encountered in developing single-page applications. It aims to simplify both the development and the testing of such applications by providing a framework for client-side model–view–controller (MVC) and model-view-viewmodel (MVVM) architectures, along with components commonly used in rich Internet applications.

The AngularJS library works by first reading the HTML page, which has embedded into it additional custom tag attributes. Angular interprets those attributes as directives to bind input or output parts of the page to a model that is represented by standard JavaScript variables. The values of those JavaScript variables can be manually set within the code, or retrieved from static or dynamic JSON resources.

General architecture of angularjs is as following:

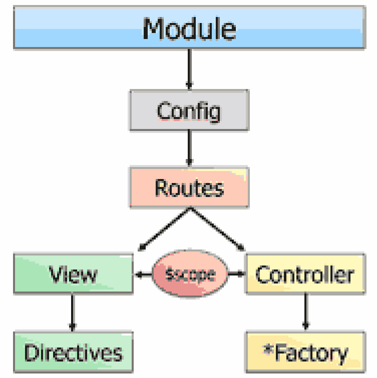
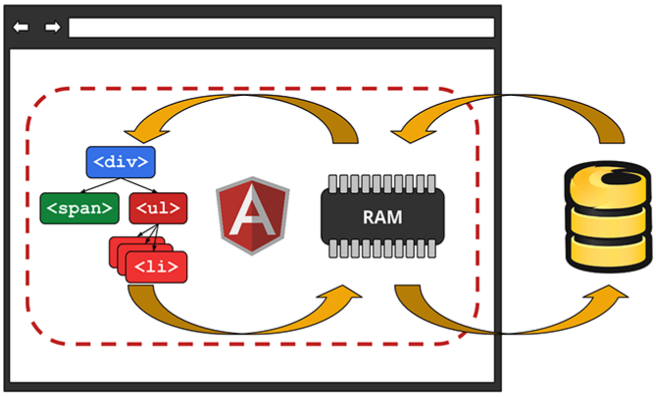


Fig: AngularJS structure diagram

AngularJS provides two way data binding. Angular's $scope service watches for changes in the model and modifies the DOM. Similarly, any updates to the bound elements in the DOM are immediately reflected in the model.

Firebase + Angular = 3 way data binding

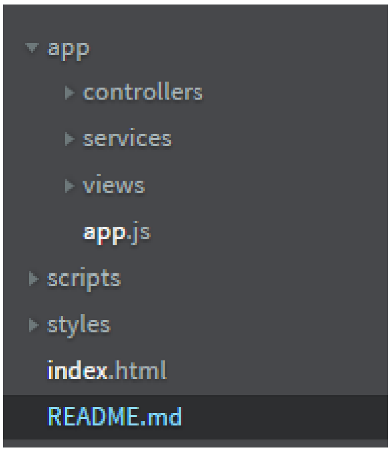
Firebase is cloud based real time JSON **database** for storing and sync the app's data. It works really well with angular and provides 3 way data binding. When data changes in one client, those updates are immediately persisted to Firebase and rendered across all clients. This three-way data binding is represented in the diagram below:



3.2.2 Folder/File Structure

Inside the Blood Pressure Logs project, I created a file structure with:

- ✓ **Index.html** as the main AngularJS file (i.e. this is the single-page application file)
- ✓ **App.js** contain module and roots to the different views of the app
- ✓ **controllers** folder contains java script files and handle the data of the app
- ✓ **services** folder contains factories that organise and share code across app and communicate with firebase database
- ✓ **views** contain html files for different views of the app
- ✓ **scripts** folder contains all the bootstrap files
- ✓ **styles** folder contain all the css files



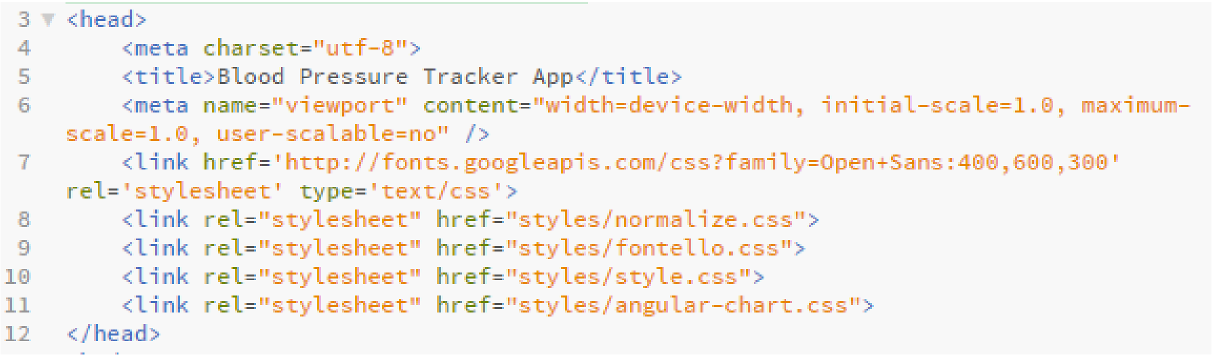
```
▾ app
    ▸ controllers
    ▸ services
    ▸ views
      app.js
▸ scripts
▸ styles
  index.html
  README.md
```

Index.html

This file is loaded directly by the browser which is made of:

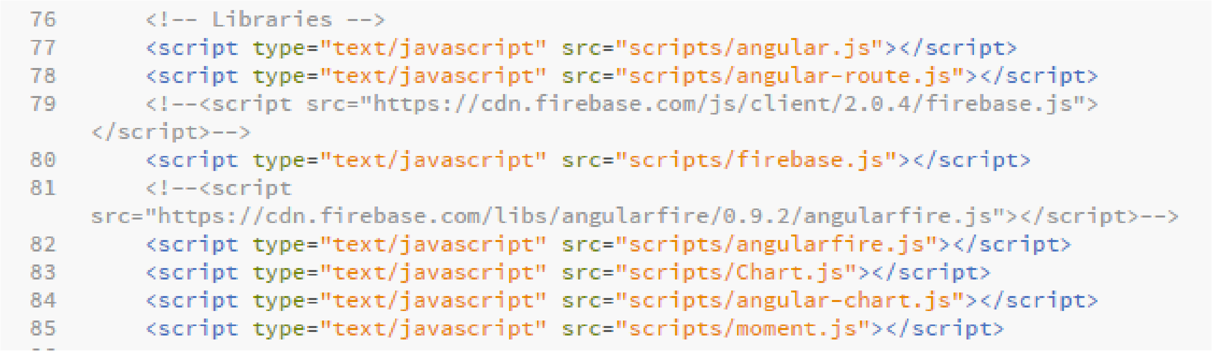
- ✓ External/Framework javascript includes: angular.js, angular-route.js, firebase.js, angularfire.js, bootsrap.min.css, chart.js, angular-chart.js, moment.js etc.
- ✓ AngularJS javascript includes (for the current application): app.js, appController.js, trackerController.js, logsController.js, projectsController.js, values.js etc.
- ✓ CSS link to bootstrap.min.css

CSS files references



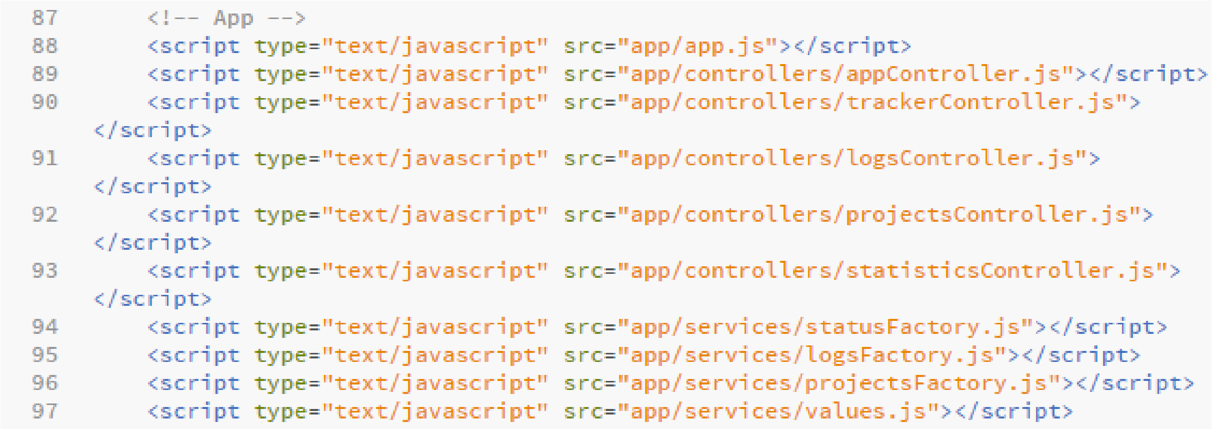
```
<head>
    <meta charset="utf-8">
    <title>Blood Pressure Tracker App</title>
    <meta name="viewport" content="width=device-width, initial-scale=1.0, maximum-scale=1.0, user-scalable=no" />
    <link href='http://fonts.googleapis.com/css?family=Open+Sans:400,600,300' rel='stylesheet' type='text/css'>
    <link rel="stylesheet" href="styles/normalize.css">
    <link rel="stylesheet" href="styles/fontello.css">
    <link rel="stylesheet" href="styles/style.css">
    <link rel="stylesheet" href="styles/angular-chart.css">
</head>
```

External Java script library files references



```
    <!-- Libraries -->
    <script type="text/javascript" src="scripts/angular.js"></script>
    <script type="text/javascript" src="scripts/angular-route.js"></script>
    <!--<script src="https://cdn.firebase.com/js/client/2.0.4/firebase.js"></script>-->
    <script type="text/javascript" src="scripts/firebase.js"></script>
    <!--<script src="https://cdn.firebase.com/libs/angularfire/0.9.2/angularfire.js"></script>-->
    <script type="text/javascript" src="scripts/angularfire.js"></script>
    <script type="text/javascript" src="scripts/Chart.js"></script>
    <script type="text/javascript" src="scripts/angular-chart.js"></script>
    <script type="text/javascript" src="scripts/moment.js"></script>
```

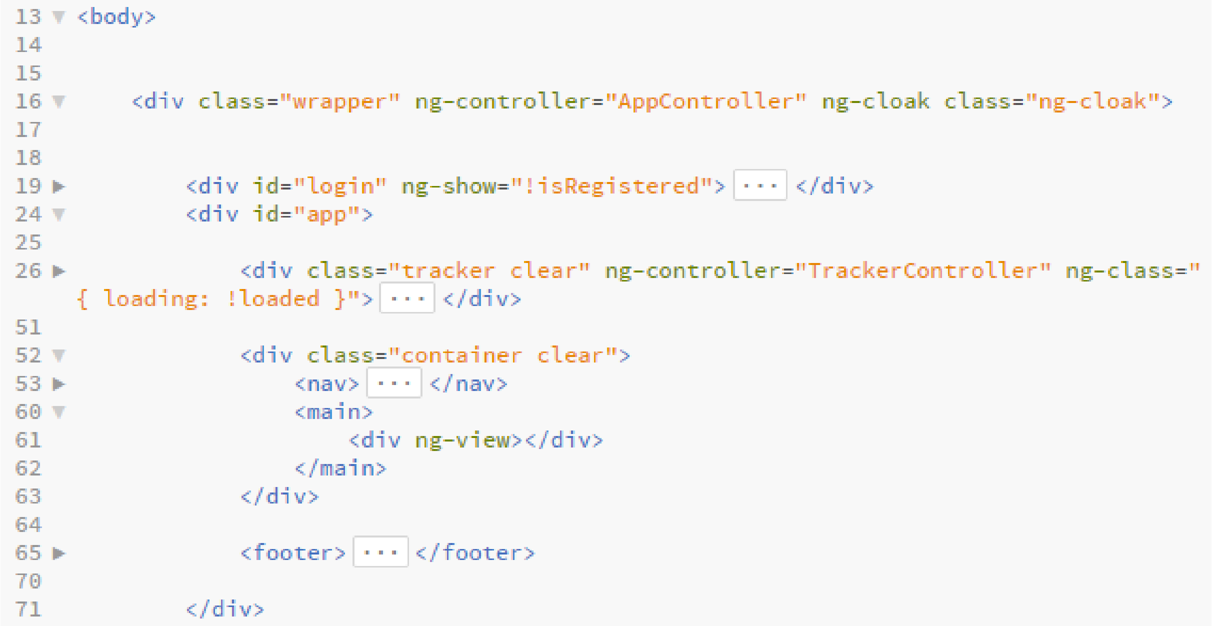
Internal Java script app files references



```
    <!-- App -->
    <script type="text/javascript" src="app/app.js"></script>
    <script type="text/javascript" src="app/controllers/appController.js"></script>
    <script type="text/javascript" src="app/controllers/trackerController.js"></script>
    <script type="text/javascript" src="app/controllers/logsController.js"></script>
    <script type="text/javascript" src="app/controllers/projectsController.js"></script>
    <script type="text/javascript" src="app/controllers/statisticsController.js"></script>
    <script type="text/javascript" src="app/services/statusFactory.js"></script>
    <script type="text/javascript" src="app/services/logsFactory.js"></script>
    <script type="text/javascript" src="app/services/projectsFactory.js"></script>
    <script type="text/javascript" src="app/services/values.js"></script>
```

Html body (containing the AngularJS **ng-app** directive) and:

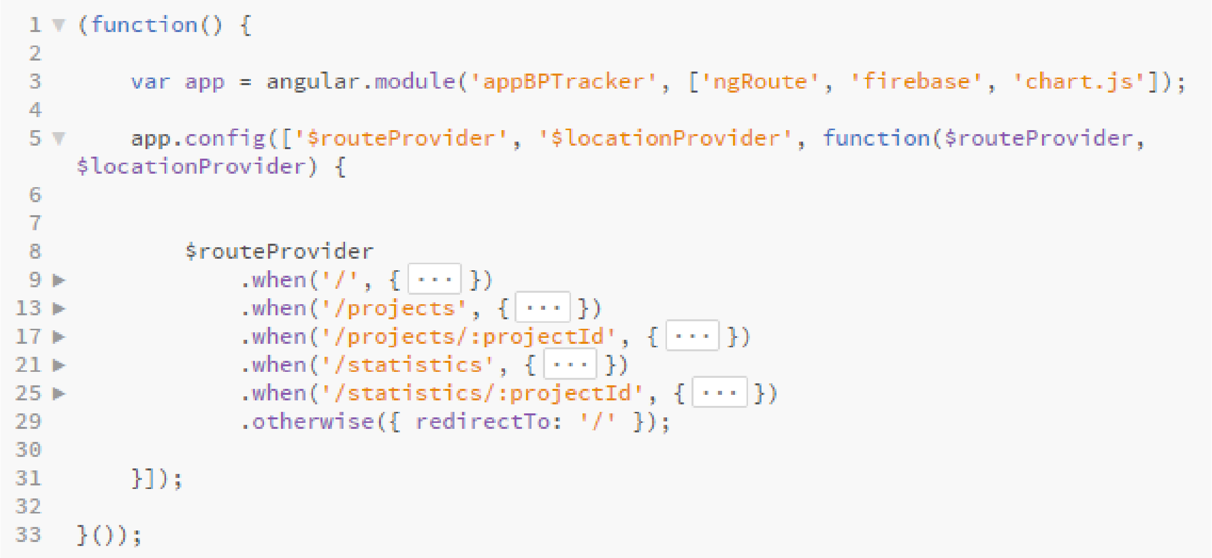
- ✓ div tag with container css class
- ✓ ng-controller directive
- ✓ ng-model directive
- ✓ ng-view directive
- ✓ ng-show directive



```
<body>


    <div class="wrapper" ng-controller="AppController" ng-cloak class="ng-cloak">


        <div id="login" ng-show="!isRegistered">···</div>
        <div id="app">

            <div class="tracker clear" ng-controller="TrackerController" ng-class="{ loading: !loaded }">···</div>

            <div class="container clear">
                <nav>···</nav>
                <main>
                    <div ng-view></div>
                </main>
            </div>

            <footer>···</footer>

        </div>
```

app.js

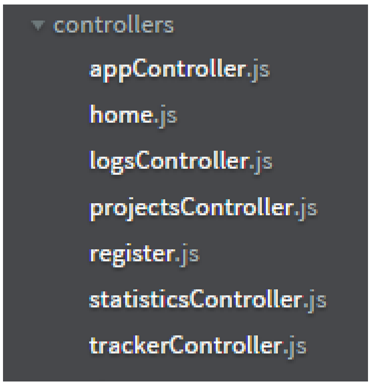
This is where the **project** module is created (with three dependencies **ngRoute**, **firebase, chart.js**).Since I moved the controllers, factories and directives into their own separate .js files, there wasn't much to do here, apart from creating routes for different views.



```
(function() {

    var app = angular.module('appBPTracker', ['ngRoute', 'firebase', 'chart.js']);

    app.config(['$routeProvider', '$locationProvider', function($routeProvider, $locationProvider) {


        $routeProvider
            .when('/', { ··· })
            .when('/projects', { ··· })
            .when('/projects/:projectId', { ··· })
            .when('/statistics', { ··· })
            .when('/statistics/:projectId', { ··· })
            .otherwise({ redirectTo: '/' });

    }]);

}());
```

Controllers

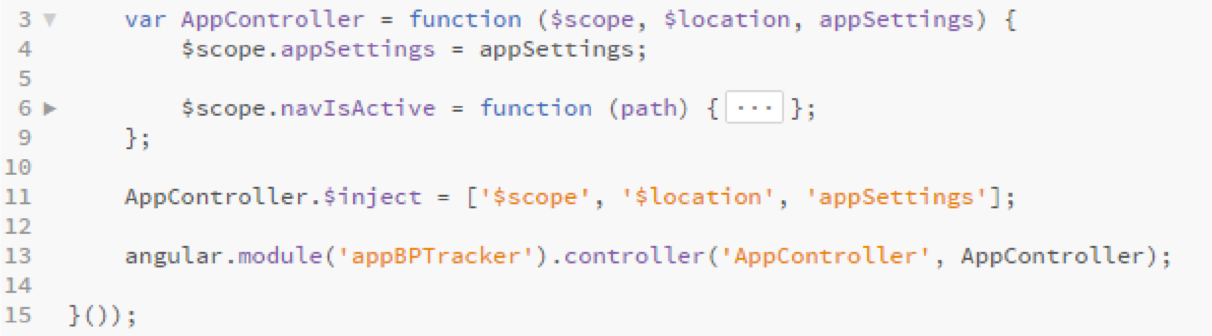
This folder contains all the all javascript files that contain different controllers used to handle the data of the app, following is the list of all controllers.



```
▾ controllers
    appController.js
    home.js
    logsController.js
    projectsController.js
    register.js
    statisticsController.js
    trackerController.js
```

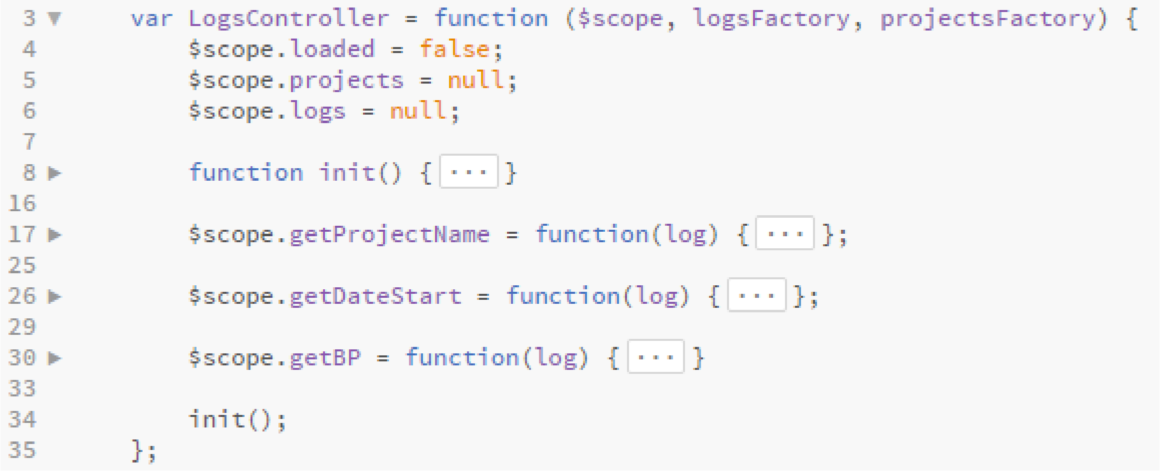
Each controller is used inside a different view.

AppController.js - this is the most important controller that controls the app



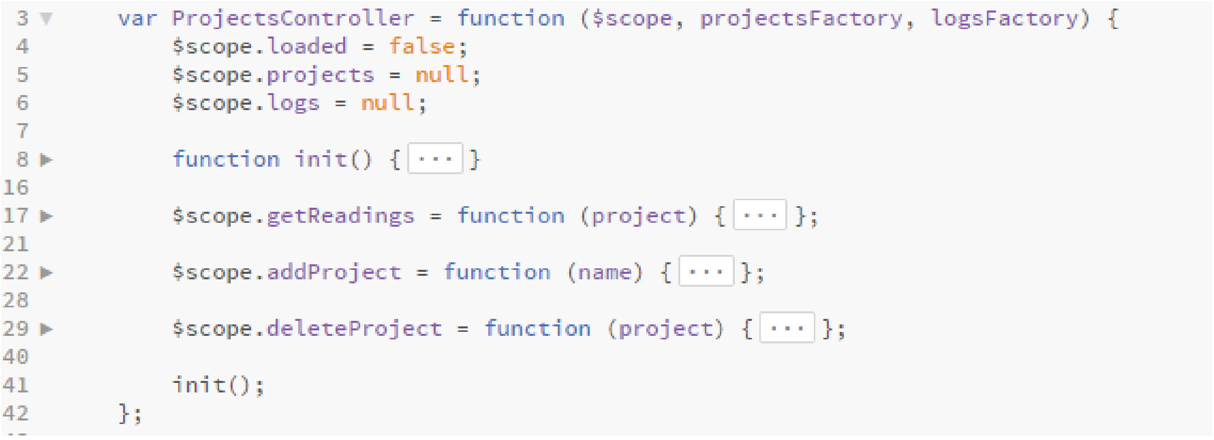
```
    var AppController = function ($scope, $location, appSettings) {
        $scope.appSettings = appSettings;

        $scope.navIsActive = function (path) { ··· };
    };

    AppController.$inject = ['$scope', '$location', 'appSettings'];

    angular.module('appBPTracker').controller('AppController', AppController);

}());
```

LogsController – used inside logs.html & contains the functions that control the blood pressure logs



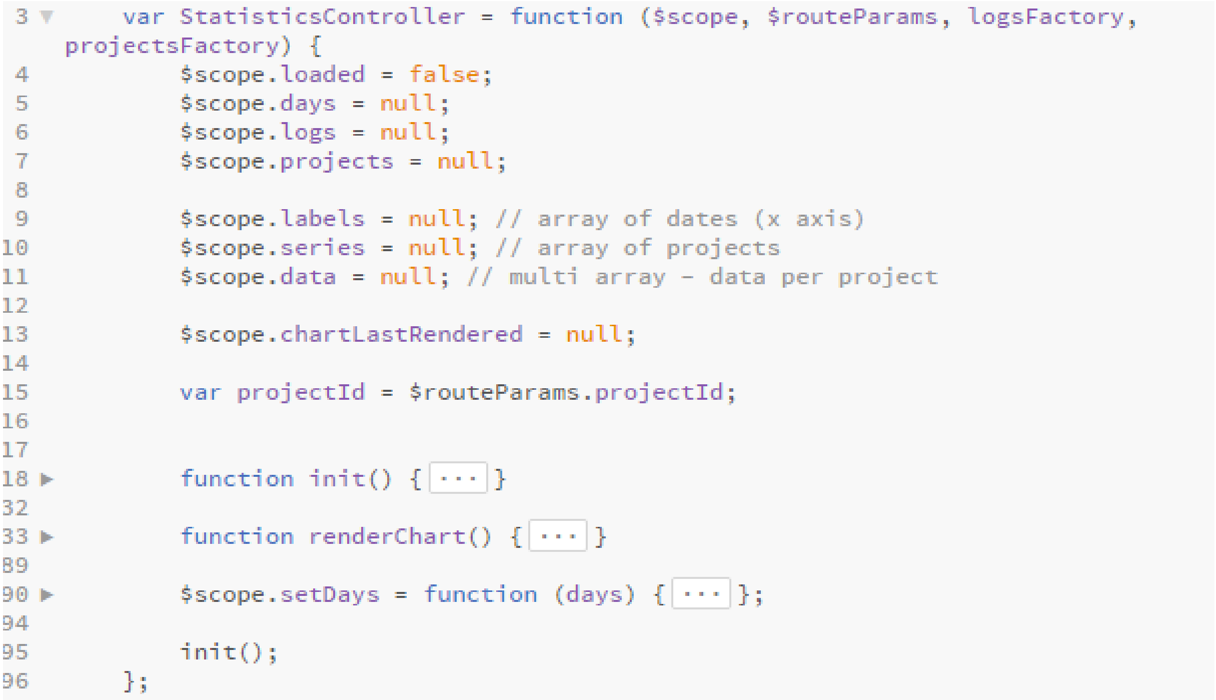
```
var LogsController = function ($scope, logsFactory, projectsFactory) {
    $scope.loaded = false;
    $scope.projects = null;
    $scope.logs = null;

    function init() {···}

    $scope.getProjectName = function(log) {···};

    $scope.getDateStart = function(log) {···};

    $scope.getBP = function(log) {···}

    init();
};
```

ProjectsController – used inside projects.html and contains the functions for getting readings and adding/deleting projects like systolic, diastolic blood pressure and impulse rate etc.



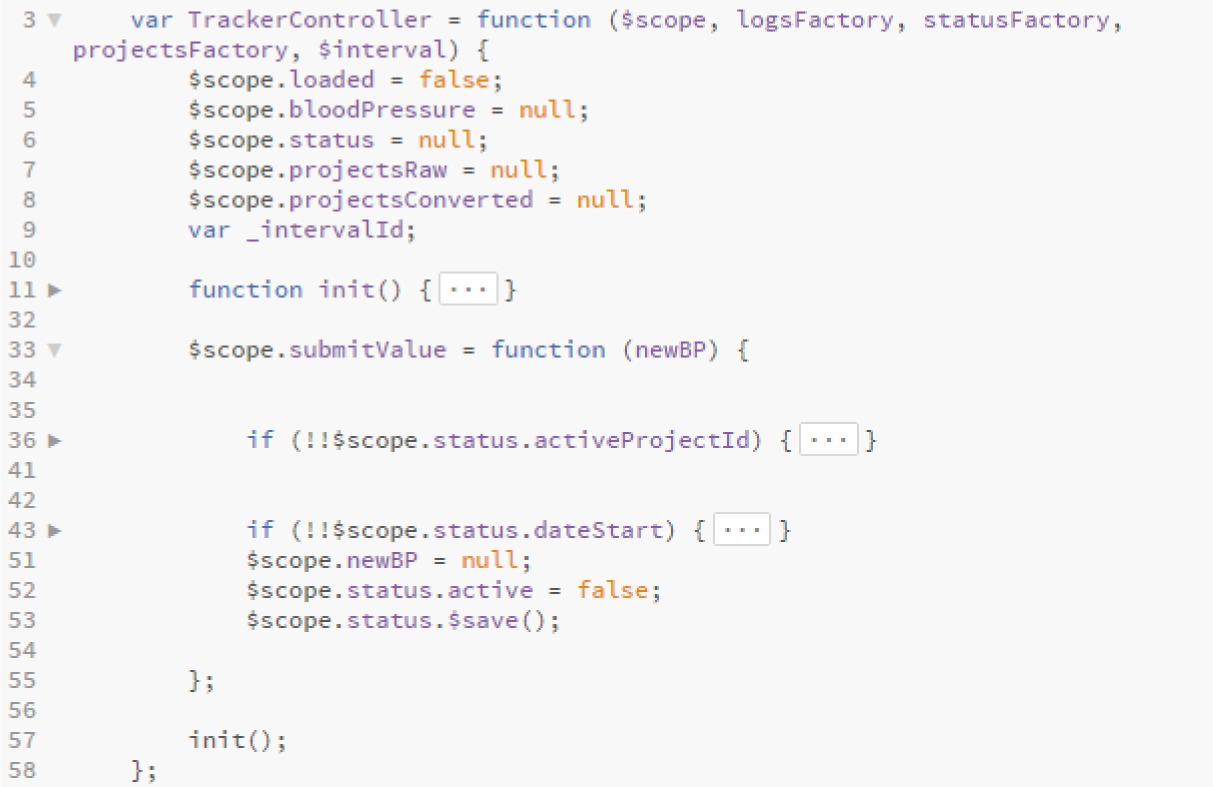
```
var ProjectsController = function ($scope, projectsFactory, logsFactory) {
    $scope.loaded = false;
    $scope.projects = null;
    $scope.logs = null;

    function init() {···}

    $scope.getReadings = function (project) {···};

    $scope.addProject = function (name) {···};

    $scope.deleteProject = function (project) {···};

    init();
};
```

StatisticsController – used to control the data for different graphs



```
var StatisticsController = function ($scope, $routeParams, logsFactory,
projectsFactory) {
    $scope.loaded = false;
    $scope.days = null;
    $scope.logs = null;
    $scope.projects = null;

    $scope.labels = null; // array of dates (x axis)
    $scope.series = null; // array of projects
    $scope.data = null; // multi array - data per project

    $scope.chartLastRendered = null;

    var projectId = $routeParams.projectId;


    function init() {···}

    function renderChart() {···}

    $scope.setDays = function (days) {···};

    init();
};
```

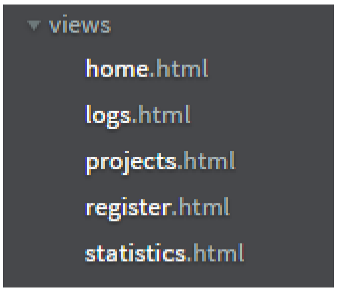
TrackerController – controls the record against different dates and time



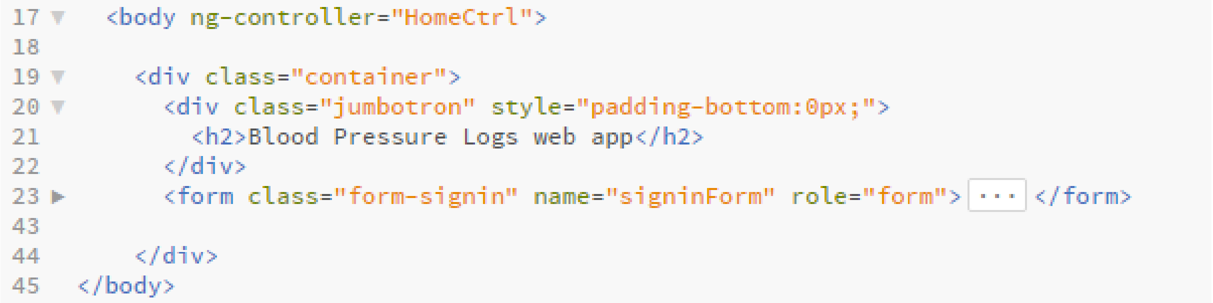
```
    var TrackerController = function ($scope, logsFactory, statusFactory,
projectsFactory, $interval) {
        $scope.loaded = false;
        $scope.bloodPressure = null;
        $scope.status = null;
        $scope.projectsRaw = null;
        $scope.projectsConverted = null;
        var _intervalId;

        function init() {···}

        $scope.submitValue = function (newBP) {


            if (!!$scope.status.activeProjectId) {···}


            if (!!$scope.status.dateStart) {···}
            $scope.newBP = null;
            $scope.status.active = false;
            $scope.status.$save();

        };

        init();
    };
```

Views

This folder contains all html files which generate different views of the app

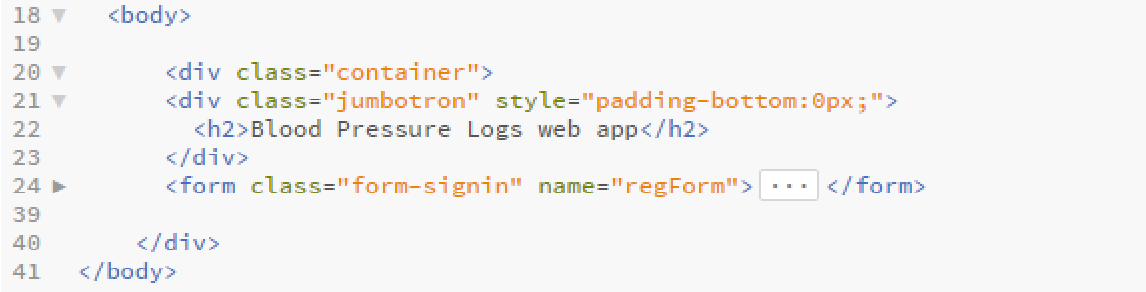


home.html – contains login view page with different fields



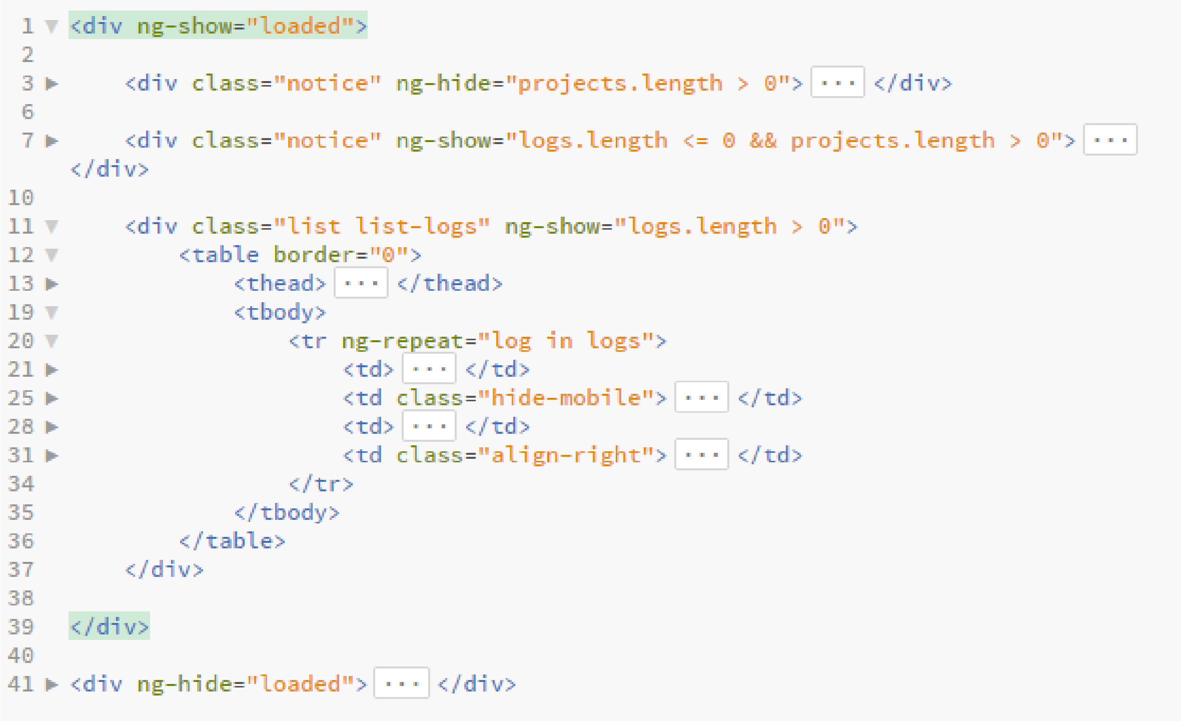
```
  <body ng-controller="HomeCtrl">

    <div class="container">
      <div class="jumbotron" style="padding-bottom:0px;">
        <h2>Blood Pressure Logs web app</h2>
      </div>
      <form class="form-signin" name="signinForm" role="form">···</form>

    </div>
</body>
```

register.html – contains signup view page for user registration



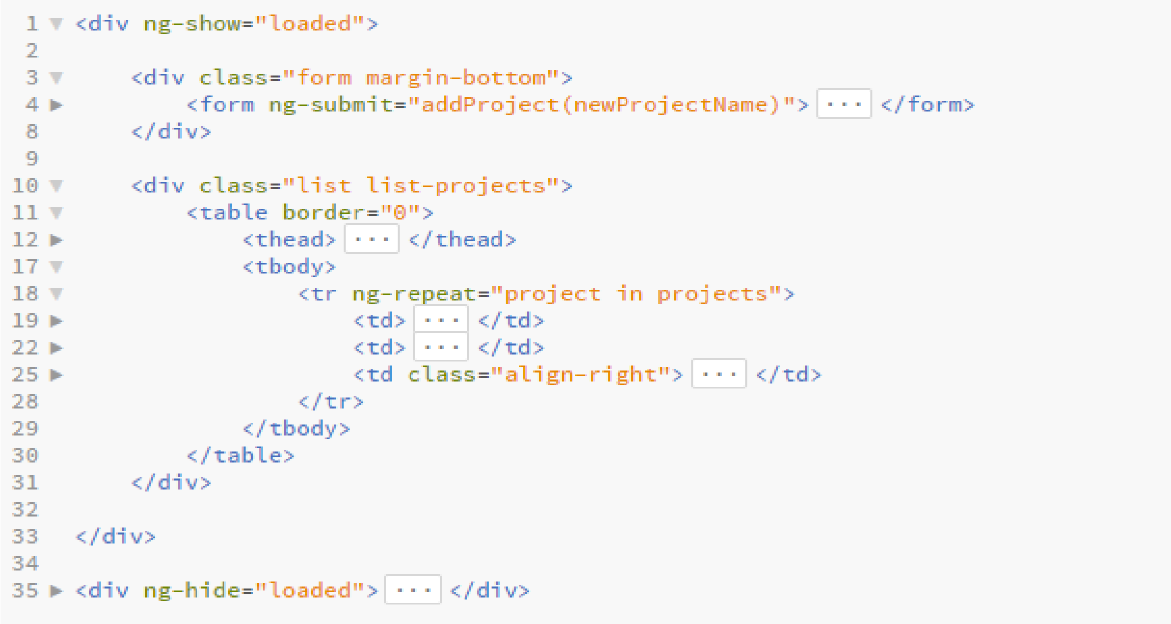
```
  <body>

      <div class="container">
      <div class="jumbotron" style="padding-bottom:0px;">
        <h2>Blood Pressure Logs web app</h2>
      </div>
      <form class="form-signin" name="regForm"> ··· </form>

    </div>
</body>
```

logs.html – contains the view for all blood pressure logs



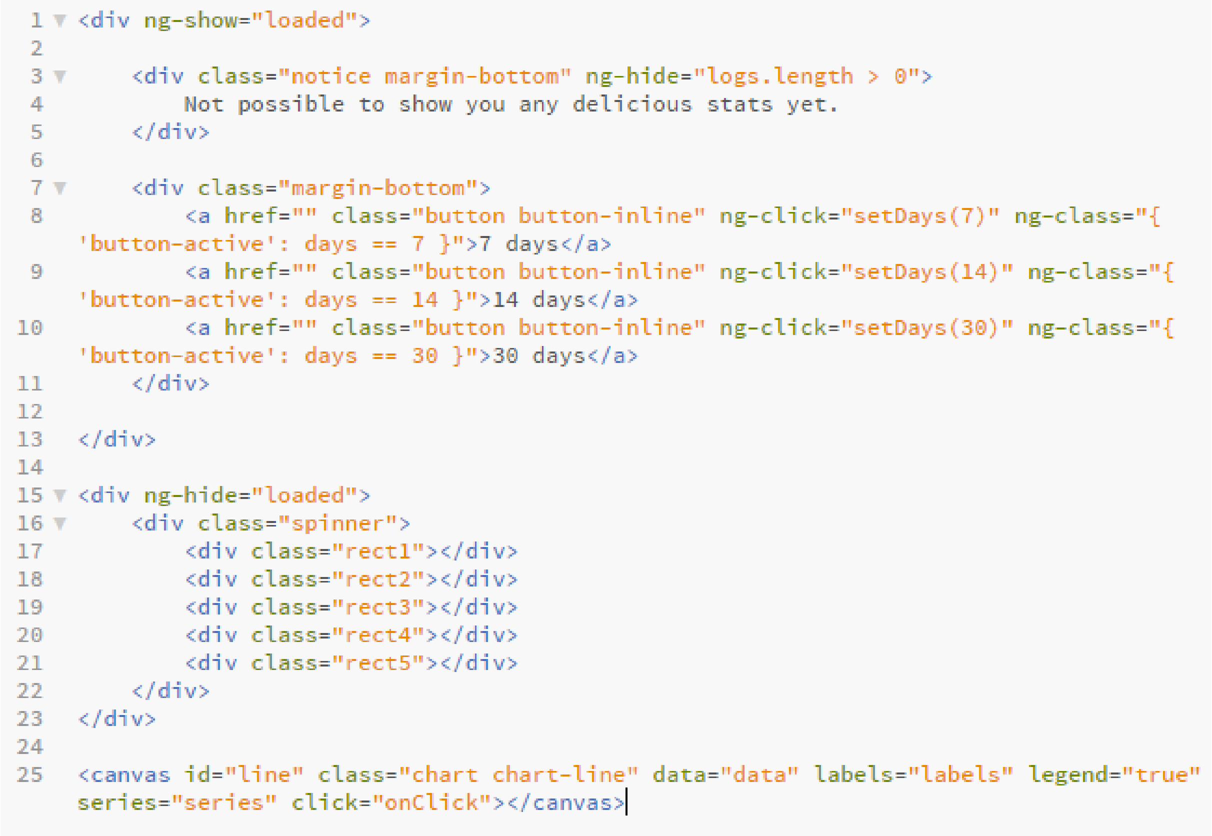
```
<div ng-show="loaded">

    <div class="notice" ng-hide="projects.length > 0"> ··· </div>

    <div class="notice" ng-show="logs.length <= 0 && projects.length > 0"> ···
</div>

    <div class="list list-logs" ng-show="logs.length > 0">
        <table border="0">
            <thead> ··· </thead>
            <tbody>
                <tr ng-repeat="log in logs">
                    <td> ··· </td>
                    <td class="hide-mobile"> ··· </td>
                    <td> ··· </td>
                    <td class="align-right"> ··· </td>
                </tr>
            </tbody>
        </table>
    </div>

</div>

<div ng-hide="loaded"> ··· </div>
```

projects.html – contains the view for different projects systolic, diastolic and impulse rate



```
<div ng-show="loaded">

    <div class="form margin-bottom">
        <form ng-submit="addProject(newProjectName)"> ··· </form>
    </div>

    <div class="list list-projects">
        <table border="0">
            <thead> ··· </thead>
            <tbody>
                <tr ng-repeat="project in projects">
                    <td> ··· </td>
                    <td> ··· </td>
                    <td class="align-right"> ··· </td>
                </tr>
            </tbody>
        </table>
    </div>

</div>

<div ng-hide="loaded"> ··· </div>
```

statistics.html – contains the view for generating charts for different values of blood pressure



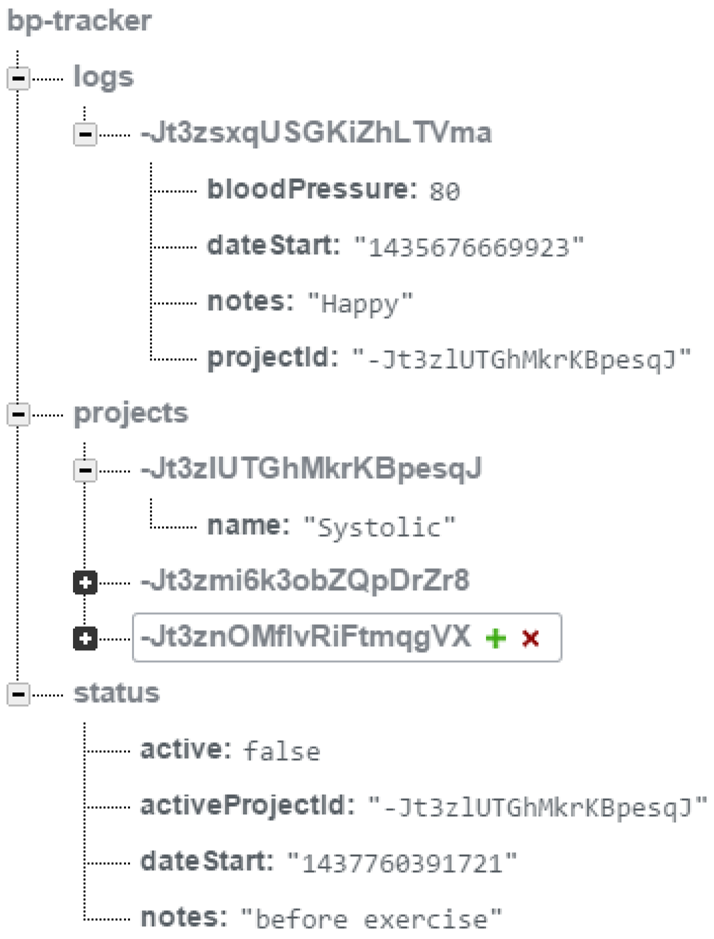
```html
<div ng-show="loaded">

    <div class="notice margin-bottom" ng-hide="logs.length > 0">
        Not possible to show you any delicious stats yet.
    </div>

    <div class="margin-bottom">
        <a href="" class="button button-inline" ng-click="setDays(7)" ng-class="{ 'button-active': days == 7 }">7 days</a>
        <a href="" class="button button-inline" ng-click="setDays(14)" ng-class="{ 'button-active': days == 14 }">14 days</a>
        <a href="" class="button button-inline" ng-click="setDays(30)" ng-class="{ 'button-active': days == 30 }">30 days</a>
    </div>

</div>

<div ng-hide="loaded">
    <div class="spinner">
        <div class="rect1"></div>
        <div class="rect2"></div>
        <div class="rect3"></div>
        <div class="rect4"></div>
        <div class="rect5"></div>
    </div>
</div>

<canvas id="line" class="chart chart-line" data="data" labels="labels" legend="true" series="series" click="onClick"></canvas>
```

4 Services

4.1 Firebase

Firebase is a cloud based real time Database. Data in **Firebase** database is stored as JSON and synchronized in real time to every connected client. When you build cross-platform apps with our Android, iOS, and JavaScript SDKs, all of your clients share one **Firebase** database and automatically receive updates with the newest data.
AngularFire is the officially supported AngularJS binding for Firebase. The combination of Angular and Firebase provides a three-way data binding between your HTML, your JavaScript, and the Firebase database.

Following is the diagram of JSON tree created by firebase.



```
bp-tracker
  logs
    -Jt3zsxqUSGKiZhLTVma
      bloodPressure: 80
      dateStart: "1435676669923"
      notes: "Happy"
      projectId: "-Jt3zlUTGhMkrKBpesqJ"
  projects
    -Jt3zlUTGhMkrKBpesqJ
      name: "Systolic"
    -Jt3zmi6k3obZQpDrZr8
    -Jt3znOMflvRiFtmqgVX
  status
    active: false
    activeProjectId: "-Jt3zlUTGhMkrKBpesqJ"
    dateStart: "1437760391721"
    notes: "before exercise"
```

Fig: Firebase JSON tree diagram

5 Project Flow Diagram

Following is the project's flow diagram which shows how the things are connected

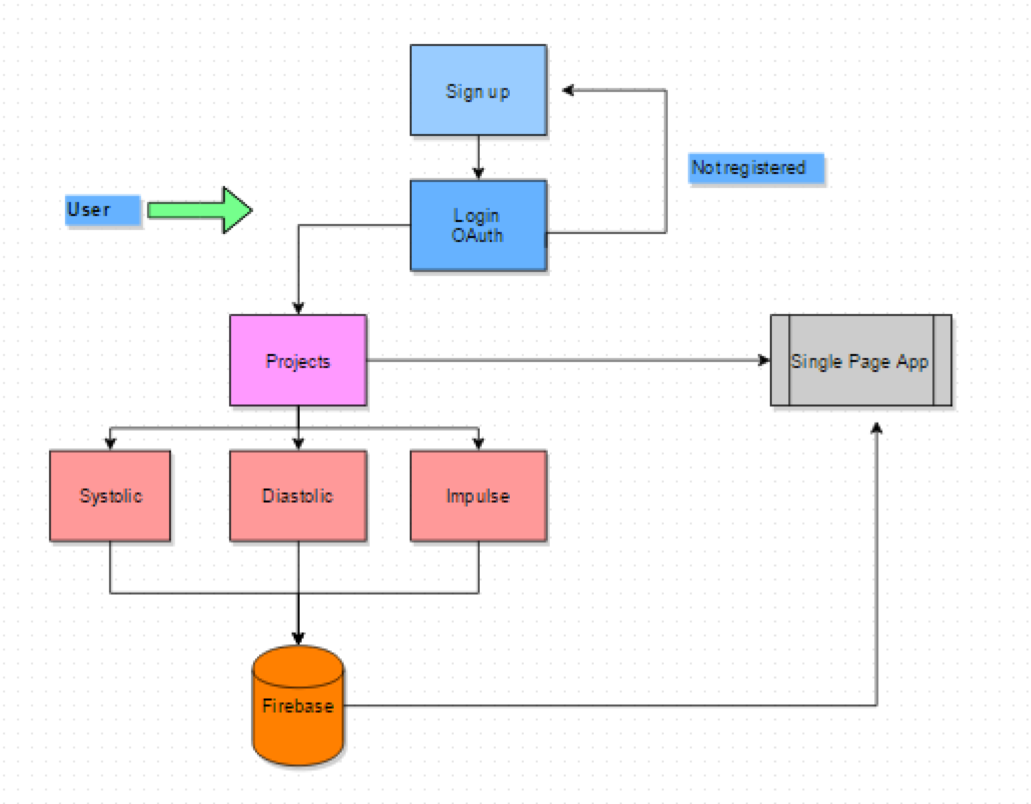
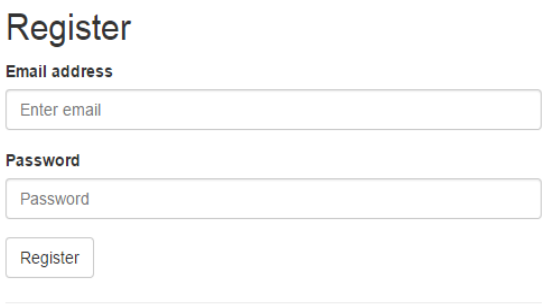


Fig: work flow chart of the project

First of all it prompts the user to login to the page with user name and password. If user is not already registered then it asks the user to register with a valid email and password. It also provides the possibility to log in via third party accounts e.g. facebook, gamil, github, twitter etc. by using open authentication (OAuth). Once user is logged in it takes him to the main app where he can see the history of logs. There is a dialogue box where user can enter different values regarding blood pressure and pulse rate. User can add as many health related variables as he want by going in to projects, he can add/delete different projects. Moreover, he can analyse his blood pressure history in a graphical way by going into statistics.

6 Design Views

6.1 Registration & Login



Register

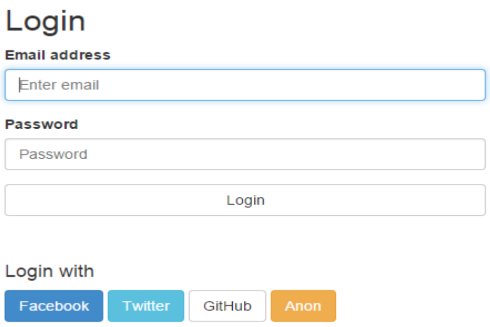
Email address

Enter email

Password

Password

Fig: registeration page



Login

Email address

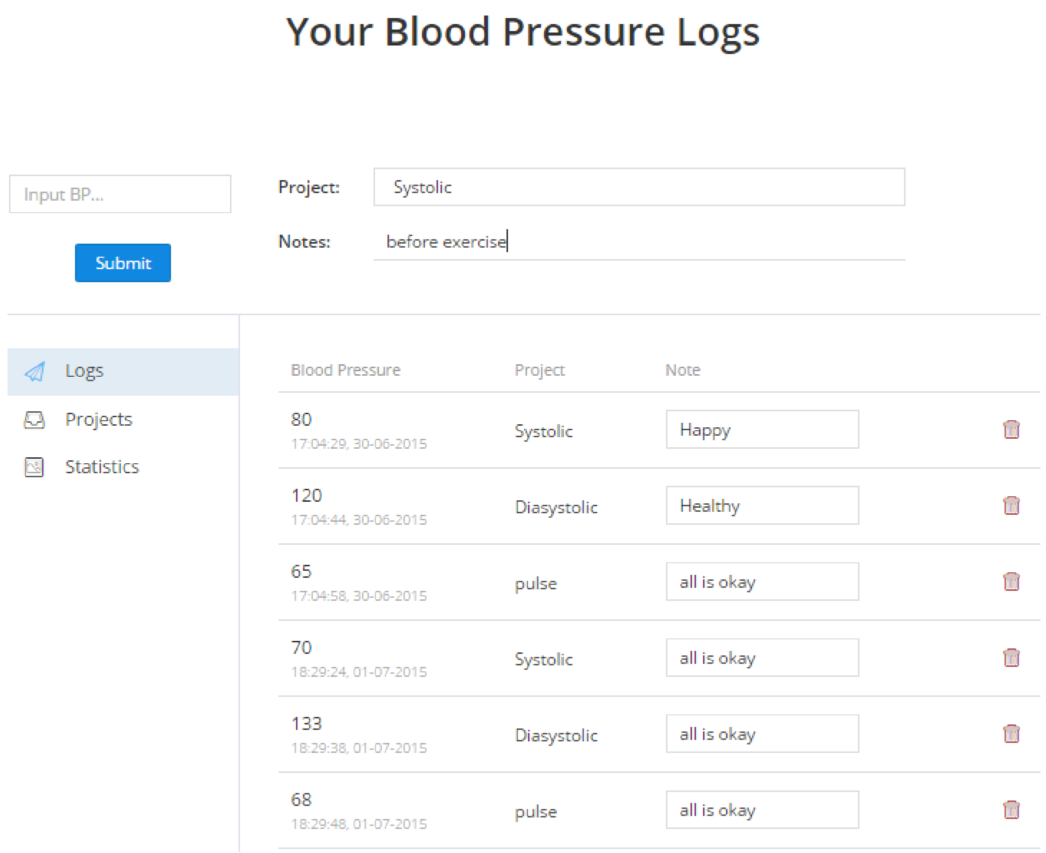
Enter email

Password

Password

Fig: login page

6.2 Blood Pressure Logs



Your Blood Pressure Logs

Input BP...

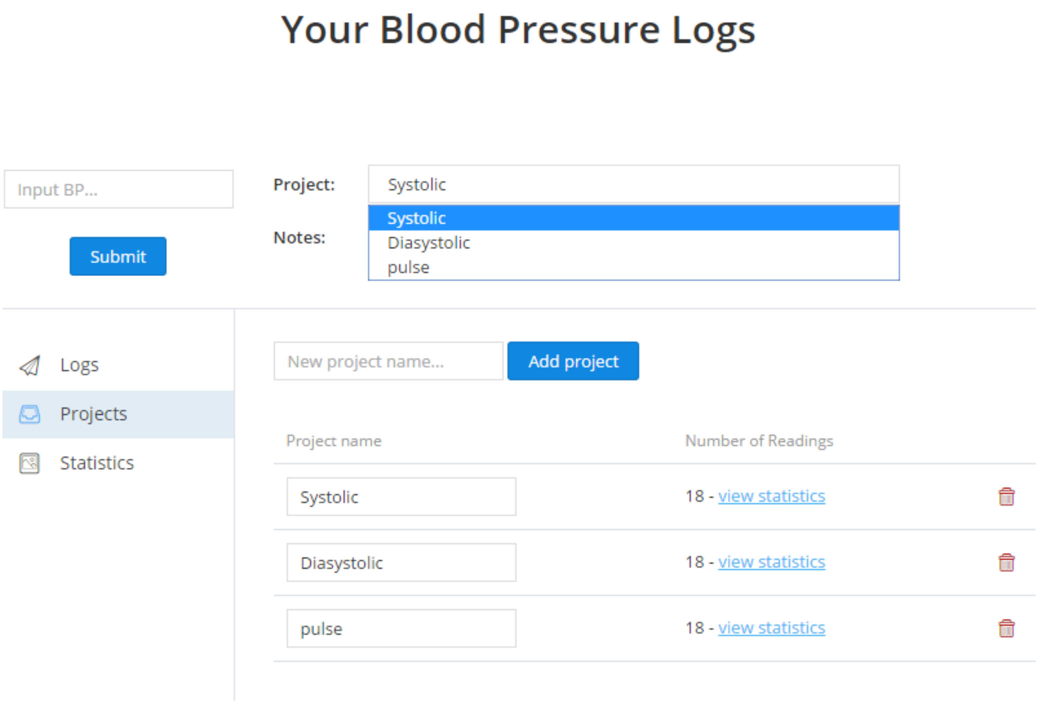
Project: Systolic

Notes: before exercise

Blood Pressure	Project	Note
80 17:04:29, 30-06-2015	Systolic	Happy
120 17:04:44, 30-06-2015	Diasystolic	Healthy
65 17:04:58, 30-06-2015	pulse	all is okay
70 18:29:24, 01-07-2015	Systolic	all is okay
133 18:29:38, 01-07-2015	Diasystolic	all is okay
68 18:29:48, 01-07-2015	pulse	all is okay

Fig: logs page

6.3 Blood Pressure Projects



Your Blood Pressure Logs

Input BP...

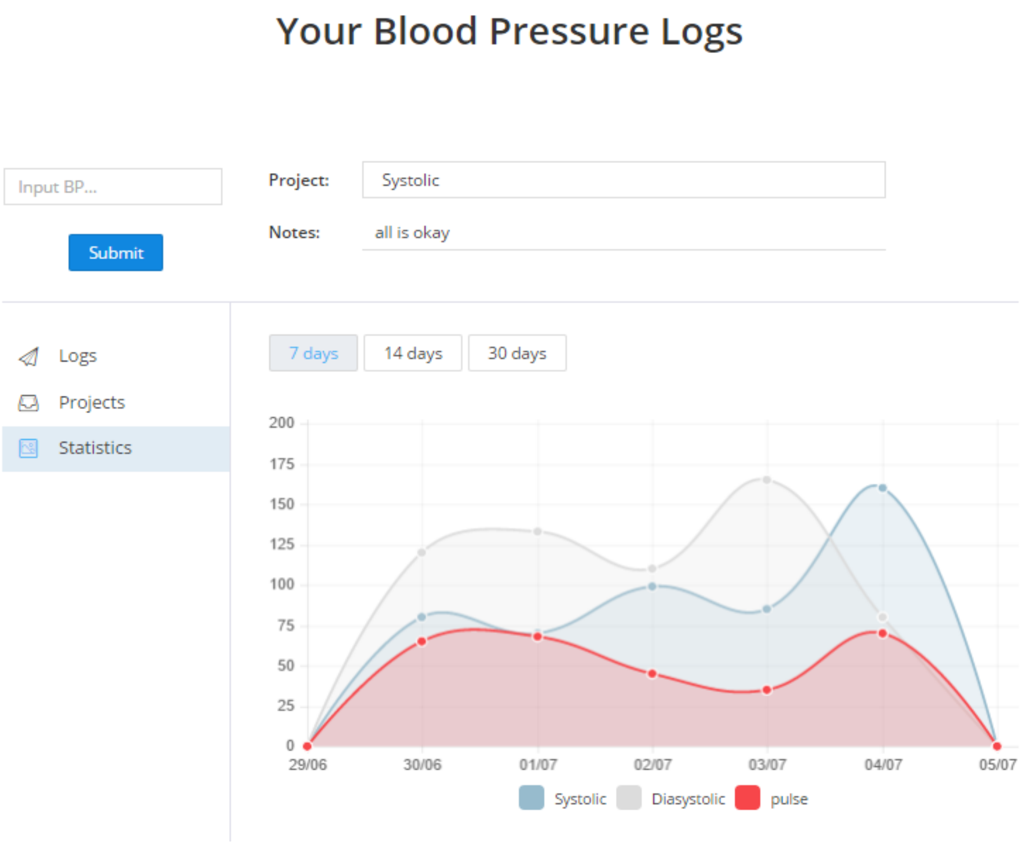
Project: Systolic

Notes:

Project name	Number of Readings
Systolic	18 - view statistics
Diasystolic	18 - view statistics
pulse	18 - view statistics

Fig: projects page

6.4 Blood Pressure Charts



Your Blood Pressure Logs

Input BP...

Project: Systolic

Notes: all is okay

Fig: charts page

7 Project Scope

7.1 Ease of access

Blood Pressure Logs will help you store and analyze your blood pressure and pulse readings made by your doctor, or at home. Now you can have your data at hand wherever you are, whatever you do. The app allows storing all crucial details, and for as many users as you like too, each having their own dataset. The details you can store include date and time, site and position, systolic and diastolic pressure, heart rate. Each entry can be stored with an optional comment and a set of text tags.

7.2 Graphical Analysis

The app also provides some essential analysis, presented in a quick view manner and a set of charts. The statistics summarises minimum, maximum and mean values over a pre-defined period of time, and time of day, with indication of long-term trend. As an extension to the basic analysis, the app provides multiple charts which can use all data points separately or daily averages.

7.3 Cloud based storage

This app uses cloud based storage which provides high level of security and availability. Moreover any update in browser will be immediately reflected in firebase or vice versa, thus providing three way data binding which is quite awesome. Moreover, cloud based storage is highly scalable.

7.4 Light weight and user friendly

AngularJs is a great way to build stylish, light weight and user friendly apps. Features like data binding, directives, filters help produce powerful apps which are simple and easily scalable. User friendly interface gives user a great experience and they can handle the app with least knowledge.

7.5 Single page application

By design, SPA applications are far more responsive. They can provide an experience that feels almost like a native app in the web. By rendering on the client they cut down load on the server as well as reduce network traffic – instead of sending a full page of markup, you can send a payload of data and turn it into markup at the client.

8 Further Enhancement

There are many future aspects related to web applications. One of the most attractive features that could be added is by using this web application we can automate the manual data entry process which requires the services of nursing staff in a hospital. This could be done by using the sensors which will automatically calculate the high/low blood pressure and heartbeat of the patient and store it in the file. Then the data of that file could be uploaded in Database and afterwards become accessible anywhere on the network.

Furthermore another feature can be added which will allow to generate and email the pdf format of all blood pressure readings to the concerned doctor.

Another feature of tips can be added which will automatically analyse your history, mean and average systolic/diastolic blood pressure values and based on that give health tips.

References

[1] Wikipedia https://en.wikipedia.org/wiki/Web_application

[2] Online tutorial for HTML http://www.w3schools.com/html

[3] Online tutorial for CSS http://www.w3schools.com/css/

[4] AngularJS https://angularjs.org/

[5] Firebase https://www.firebase.com/tutorial/#session/koyvdd94ojt

[6] Bootstrap http://getbootstrap.com/

[7] Tutorials https://www.codecademy.com/learn

[8] Tutorials https://www.codeschool.com/

[9] AngularJS Docs https://docs.angularjs.org/guide/controller

[10] Github https://github.com/angular-ui/bootstrap