Name: Syeda Sara Ferdousi
CS410: Project Report

As I have paid a number of visits to my doctor's office lately, I realized, I hardly remember anything about my health. Every now and then I measured my blood pressure but did not log it in anywhere. Thus, I am clueless, when my doctor wants to know how was my blood pressure in the last one year that I did not visit him. Similarly, when I try to lose weight I do not have any track of my weights over the years. Hence, I came up with the idea of the Health Monitor web application. The purpose of this application is simply to track health information. The ability to track health information and analyze them allows having a clear idea on one's overall health and early diagnosis of any diseases. In this case this application is certainly useful to detect hypertension and obesity. Currently, the Health Monitor web application is only capable of tracking body mass index (BMI), weight and blood pressure. Over time I would like to add more trackable health data such as body fat percentage, cholesterol.

I wanted to implement the project in a model view controller (MVC) format. Since, most of the control and model data are placed in the Server.js file, I only have the views folder to keep all the html files and pictures in it. I used Bootstrap for the UI and CSS to change the overall appearance of the pages. Web page functionalities are implemented in JavaScript. However, I have not put the CSS and JavaScript codes in separate files. They are placed in individual html files. Therefore, when I made CSS changes I had to make same changes in multiple files. For example, CSS code for buttons and wells. In future, I plan to put all the CSS codes in a single file and link it in the html files. The application is capable of taking user input via forms. The forms are validated in the client side. If valid data such as username, password, weight and others are inserted, they get stored in the database tables. I have used SQLite as my database. Database manipulation codes are placed in the Server.js file. To implement charts I followed Chart.js documentation for line charts. Although charts are working as expected, I still need to figure out how to add legend labels and how to make the date gap in the X-axis to reflect the actual date gap rather than a fixed gap. I want to get a better datepicker plugin for the application. Initially I wanted to add time, however, I realized that at this stage only date is sufficient for the application. I look forward to enable this application to get user health data from other software such as Fitbit through API.