**Project Documentation**

**Marketing:**

People, specifically BU students, should be drawn to this app as a health diary. Nearly every college student struggles with being over or under weight. With common fears such as the “freshman fifteen”, this app can soothe such students with its accurate calorie counts, pulled directly from BU dining hall calorie statistics. BU students can keep track of their calories according to the dining hall menus. In the circumstance that students are not eating in a dining hall, they also have the option of entering the calorie counts through manual entry. This app can be constantly updated to reflect the changing BU dining hall menu, and also can be expanded upon with late night dining options and common restaurants around BU, such as Pavement, GSU, Chipotle and Starbucks.

**Front-end:**

When the app first opens the home view is displayed. This is the main centralized view of the app and has the option for the users to access the meal\_menu view and the calendar view, through their corresponding buttons in the center of the page labeled “Enter Meal” and “Calendar”. When the user clicks the enter meal button, they are taken to the meal menu. On this view and on all other views, except for the calendar page, there are two navigational buttons at the top. On the top left of these subsequent pages there is a calendar button which takes the user to calendar view, and on the top right there is a home button which takes the user to the home view. Meal\_menu contains these two navigational buttons at the top of its page but also has four buttons located in the center of the page from top to bottom, which takes an input for what meal was consumed by the user. The first three buttons read breakfast, lunch and dinner from top to bottom, respectively. All three of these buttons take the user to the dining hall view. The last button is a manual input for when the user does not eat at a dining hall and takes the user to the manual\_entry view.

In the dining hall view there are the two navigational buttons at the top, and three buttons listed down the page that display the locations of the three dining halls, which are Baystate, Warren and West. When the user selects any of the three buttons, the app inputs a location for the dining hall and proceeds to the enter\_food view.

The enter\_food view is a special view as it takes the user’s two previous inputs, which were time of meal and location, and reads the menu of the dining hall for that day. We spent many hours creating a working code that scraped the BU menu page for food items. We got the scraping functionality to work using JavaScript and JSON objects, but were unable to implement that code in the Java side of the app. The meal items are then displayed in a dropdown box which is located at the center of the page. Once the user decides on his or her meal they press the “Save” button located at the bottom of the page. Once the save button is pressed the app sends the user to the summary view to display their calories for the day. If the user did not select a meal time and instead decided to input the calories manually, the app would take the user to the manual\_entry view. This view has the two navigational buttons on the top and also has two textEdit boxes for the user to manually input the name of the meal and the amount of calories of their meal. When they have completed these two tasks, they press the “Save” button located at the bottom of the page which takes them to the summary view.

The summary view, which is accessed by the save button in either the enter\_food view or the manual\_entry view, displays the total calories and the meals consumed for the current day. After this, the user must press the navigation buttons at the top of the page, either taking them back to the home view or the calendar view. When the user goes to the calendar view, either through any of the navigational buttons or the calendar button on the home view, the user is brought to a calendar display. Through this calendar view the user is able to return home by pressing a home navigational button similar to other view pages and is able to select a date by pressing on the date on the calendar. When the user selects a date they can press the “Set” button on the bottom of the page that brings the user to the calorie display view for that corresponding date.

**Back-end:**

Originally, our JavaScript web-scraping code accessed the BU menu’s server, which then requested data from the BU dining hall menu page. It then pulled the HTML data and sent it back to the JavaScript code. Then the Cheerio module was used to parse the data and place it in a multidimensional array formatted as a JSON object. Despite spending many hours be able to access the JSON data, the JavaScript was not able to run on the Android app. Instead of doing this we utilized basic meals for the user to select that are commonly found in their respective BU dining halls.

The calendar was created using an android widget called DatePicker. This widget can be used to easily select dates within a calendar. We used a DatePicker over CalendarView because CalendarView only showed the current day in the current month and was unable to select days, which was important for the calorie display view. Each of the views had its own class and XML file.

Global variables were an important part of the app. The class Global\_Variables was used to save char variables that represent what meal and what dining hall the user ate at from their input meal\_menu and dining\_hall classes. This is how enter\_food knows which options to choose from. Global\_Varibles is the class where the JSON objects would be stored.