

---

# **Mobile Application Development Spec**

**for**

## **Project 1 - 4**

**Version 1.0 approved**

**Prepared by Daniel Graham PhD**

**University of Virginia**

**January 31 2019**

# Table of Contents

<b>Table of Contents</b>	<b>ii</b>
<b>Revision History</b>	<b>iii</b>
<b>1. Introduction</b>	<b>1</b>
1.1 Purpose	1
1.2 Document Conventions	1
1.3 Collaboration and external sources.	1
1.4 Check INs	1
1.5 Version control and bug tracking	1
1.6 Integrated Development Environment.	2
1.7 Ensuring code quality	2
1.8 Evaluating User stories	2
<b>2. App Design</b>	<b>2</b>
<b>3. Server and System Requirements</b>	<b>2</b>
<b>4. Splash screen (125 points - Group)</b>	<b>3</b>
4.1 Responsive Splash. (20 points)	3
4.2 Button Feedback. (5 points)	3
4.3 Animation (25 points)	4
4.4 Internationalization (15 points)	4
4.5 Navigation Getting Started (10 points)	4
4.6 Second Time Bypass (10 points)	4
4.7 Oath based login and registration (40 points)	4
<b>5. Home Screen (80 points – Group, 175 points – Individual)</b>	<b>5</b>
5.1 Display a single Invitation Card (50 points)	5
5.2 List of pending invitations (5 Points) (Individual)	6
5.3 Invitations and Firebase (25 Points) (Individual)	6
5.4 Pending Carousel (45 Points) (Individual)	6
5.5 Display the Events for today, tomorrow & the next day (30 points)	7
5.6 Vertical Scrolling Event Section (Individual) (40 points)	7
5.7 Horizontal Swiping by Month (Individual) (40 points)	7
5.8 Horizontal Swiping by Calendar (Individual) (20 points)	7
<b>6. Invitation Details Screen (125 points - Individual)</b>	<b>8</b>
6.1 Click on the invitation card. (20 points)	8
6.2 Display Map with location of the event marked. (25 points)	8
6.3 Display your current location on map. (30 points)	9
6.4 Display a single Invitation Card (50 points)	9
<b>7. Create Event (320 points – Individual)</b>	<b>10</b>
<b>10</b>	
7.1 Add dinner time (40 points - Individual)	10
7.2 Mapping (100 points - Individual)	11
7.3 Display send invitation screen (160 points – Individual)	11
7.4 The back button (20 points - Individual)	12

<b>8. Your Event Details Screen (110 Points - Group)</b>	<b>13</b>
8.1 Display newly created event (30 points)	13
8.2 The edit button works (20 points)	14
8.3 The cancel button works: (20 points)	14
8.4 Pop-up confirmation (30 points)	14
8.5 The back button works (10 points)	14
<b>Appendix A: Points Maps</b>	<b>14</b>
<b>Individual Points to Grade Map</b>	<b>14</b>
<b>Group Points to Grade Map</b>	<b>14</b>

## Revision History

Name	Date	Reason For Changes	Version

# 1. Introduction

## 1.1 Purpose

The purpose of this application is to allow people to schedule and attend dinners. The application will be a react-native application that will share a common code base but run on both Android and IOS devices.

## 1.2 Document Conventions

This document contains several user stories and each user story has a point value next to it. These point values indicate the number of points that you will receive if you implement the user story. Your grade will depend on the total number of points you amass. The appendix contains the information on the grades level that correspond to the total number of points that you receive. There are two components to your grade: an individual section a group section. User stories marked individual can only count toward your individual grade, while section marked group can only count for the group section of the project. The appendix also contains the threshold for the individual portion of the project. All items not marked individual should be completed as a group.

The remainder of the document is structured as follows. Each Section describes a screen in the application. User stories will generally follow the format below

As a < type of user >, I want < some goal > so that < some reason >.

## 1.3 Collaboration and external sources.

The individual portion of the project must be completed on your own. Your implementation should not contain code that you got online (with one exception you can use a tool for generating your JSX elements from the sketch file that is provided with the project). For the group portion of the project you are allowed to work in groups of two. You can also to complete the entire assignment on your own.

## 1.4 Check INs

There will be 7 TA checks available throughout the semester. There are point benchmarks for each check in. These point benchmarks are simple guidelines and are not strong requirements. Your final group and individual grades will be calculated at the end of the semester.

## 1.5 Version control and bug tracking

You will be required to use the Visual Studio Team Foundation Tool for project management and version control. <https://visualstudio.microsoft.com/tfs/> 'Please add me to all over repositories:

[dggraham@email.wm.edu](mailto:dggraham@email.wm.edu). This is how you will submit your assignment. (We will look at your final repository and check how your application performs)

## 1.6 Integrated Development Environment.

Please feel free to use whatever you want. I have listed a couple good ideas below:

1. Snack.expo.io
2. Visual studio code
3. Visual studio

The visual studio team foundation server supports git. Please use this as your version control system so that we can look at the commit logs to ensure that all members of the team contributed equally.

## 1.7 Ensuring code quality

All of the code developed should follow the JSX & JavaScript style guide originally proposed by the team at AIRBNB. <https://github.com/airbnb/javascript/tree/master/react>

The ESLint linter tool should be installed and used to enforce the style during development. The link below provides a great description on how you can configure VS Code to ensure the rules in the style guide are followed.

<https://travishorn.com/setting-up-eslint-on-vs-code-with-airbnb-javascript-style-guide-6eb78a535ba6>

## 1.8 Evaluating User stories

The TA and I will run your application and test the user stories that you have completed. If the user story behaves as expected you will receive full points for implementing that user story. However, if the application crashes, errors occur or the user story does not perform as expected you will not receive credit of that story.

# 2. App Design

All of the design files and artifacts have been supplied and can be found on Collab. We have provided the ordinal sketch files. (If you don't have a mac you can view and modify the sketch files using Avocode (Avocode.com)). It is important to note that you don't need to use sketch to create designs, I have already done that for you. We also provided slice repository with all of the sketch files. We have provided a collection of design files that contain two version of the

application. Please feel to use which ever design you like more.

There are several tools that determine exact pixel values for the artifacts in the design. Feel to use a tool of your choice: I recommend

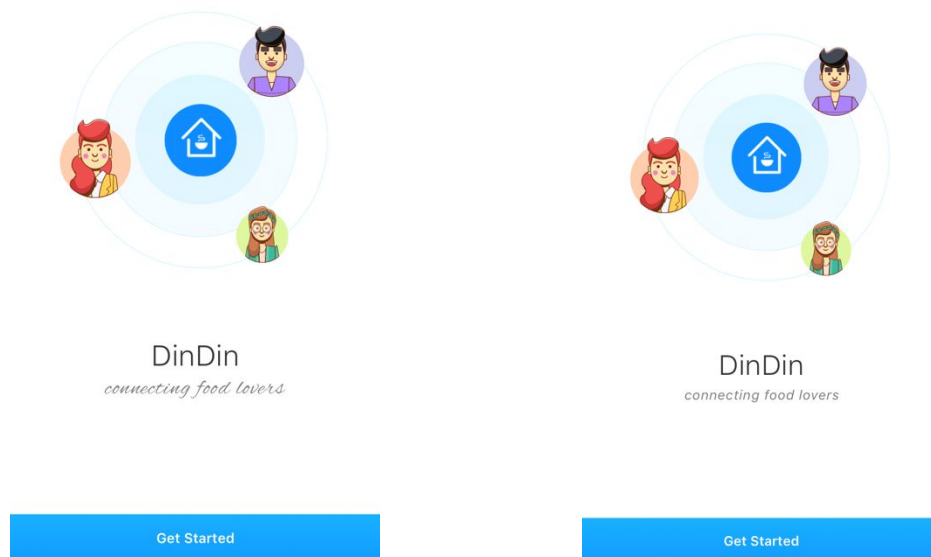
1. <http://utom.design/measure/>

### 3. Server and System Requirements

We will use the firebase Realtime database to store all the data related to the application. If you want to use server static data, feel free to use your home directory. We will not cover scalable serverless architectures but if you want to implement a cloud function/ lambda functions/ Azure functions feel free to do so, but this is not a requirement. As the developer you have full control the structure of the JSON objects.

### 4. Splash screen (125 points - Group)

This the first screen that the users will see. It is screen that will establish your products brand. This also the first impression that users will get of the application. The design for the splash screen is shown in figure both Figure 2 and 3. The third variation can be found in the design files that have been supplied with this assignment.



#### **4.1 Responsive Splash. (20 points)**

The splash screen is required to be responsive. (This means that it adjusts its layout elegantly regardless of the screen size that it is run on).

**4.1.1 As a user I am able rotate view the screen both in horizontal and vertical mode. (10 points)**

**4.1.2 As a user I am able to view the contents of the screen on both a tablet and a smartphone. (10 points)**

#### **4.2 Button Feedback. (5 points)**

Feedback is a critical component of any application. Know when a button is pressed.

**4.2.1 As a user I want to click the button causing it to darken (It's Red and Green values increase by 50%)**

#### **4.3 Animation (25 points)**

Animation provides your application with a more polished look. This user story requires you to animate the people around the logo, by having each of them slowly fade in and out of view over the period of 1 second (each)

**4.3.1 As a user I the avatars slowly and randomly fade in and out of view. (25 points)**

#### **4.4 Internationalization (15 points)**

Your application will run in several countries across the world. You need to ensure your application translates itself to the default language. (You will only need to support one other language Arabic)

**4.4.1 As a user I am able to read slogan on screen because the application translates the string to my default language. (15 points)**

#### **4.5 Navigation Getting Started (10 points)**

Your application will contain multiple pages. You need to be able to navigate from one page to the next.

**4.5.1 As a user when I click the ‘getting started’ button the app navigates to the home screen. (10 points)**

#### **4.6 Second Time Bypass (10 points)**

Since the getting started screen is simply a branding screen. We don't want users to see the screen the second time they open the application.

**4.6.1 As a user when I open the app from the second time, the getting started screen is skipped. (10 points)**

#### **4.7 Oath based login and registration (40 points)**

Ideally you want to be able to allow your users to log in to your app using Oath so that the user can use other logins that they already have. We will focus on one form of authentication that is currently really popular which is Facebook login.

**4.7.1 As a user I am able to login into the DinDin application using my Facebook login. (20 points)**

When implementing this feel free to connect to Facebook login directly or use the firebase authentication service.

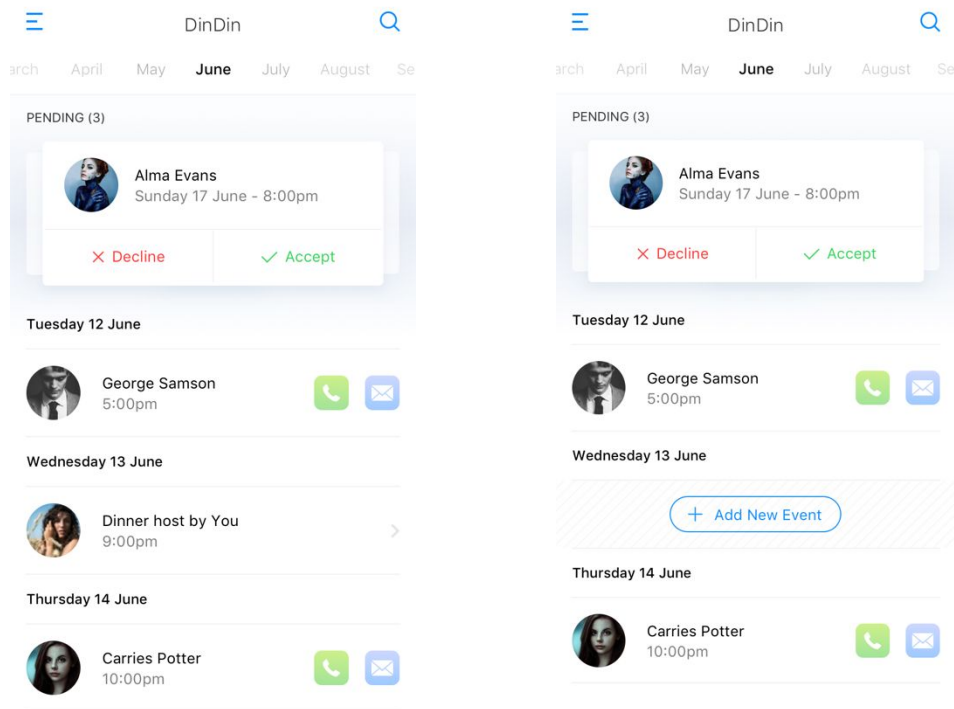
**4.7.2 As a user once I login, I will never have to login in again since the system remembers your login. (20 points)**

Feel free to use expo local storage option to store the token. (Encrypting your local storage would be a plus)

### **5. Home Screen (80 points – Group, 175 points – Individual)**

Clicking the *getting started* button gets you to the home screen this is the most involved screen in application. It includes an overview of everything that you need to plan and attend your dinners. The figures below show two designs for the home screen. One design shows the ‘*add event*’ option which is shown if there are no dinners on that day.





## 5.1 Display a single Invitation Card (50 points)

You are required to implement a custom component for this section. Do not use any existing npm packages to implement your card.

**5.1.1 As a user I am able to see single pending invitation card. (5 points)**

**5.1.2 As a user I can see the avatar of person who is inviting me to dinner. (5 points)**

**5.1.3 As a user I can see the name of the person who is inviting me to dinner. (5 points)**

**5.1.4 As a user I can see the date and time of the dinner. (5 points)**

The date time should be in the following format <Day of Week> <Day number> <Month Name> - <Time am/pm>

**5.1.5 As a user I can click the accept button to accept an invitation. (10 points)**

Once the user clicks the accept button the Invitation should flash green for 1 second and then disappear. After clicking accept/reject, the next pending (if any) invitation replaces the original invitation. (similar to 5.4.1 but in the home screen without opening up a list of invitations)

#### **5.1.6 As a user I can click the reject an invitation. (10 points)**

Once the user clicks the *reject invitation* button the invitation should flash red and then disappear.

#### **5.1.7 As a user I only see the calendar section if there are no invitations. (10 points)**

If there are no pending invitations, the invitation section should disappear.

### **5.2 List of pending invitations (5 Points) (Individual)**

Since only one invitation is visible at a time. We would like to be able show the user a list of pending invitations.

#### **5.2.1 As a I user I can see a list of pending invitations. (5 points)**

Fetch a list of invitations from firebase endpoint. Display account of invitations

### **5.3 Invitations and Firebase (25 Points) (Individual)**

When a user accepts or rejects an invitation the appropriate firebase objects are updated or deleted (If you complete this task include a description of invitation JSON object in your presentation)

### **5.4 Pending Carousel (45 Points) (Individual)**

In an ideal world we would not have to accept or reject an invitation in order to see the next invitation. It would be great if we could look at all the invitations and then decide.

#### **5.4.1 As a user I can swipe through a list of my pending invitations. (45 points)**

Feel to use npm or other open source react native packages to implement this.

## **5.5 Display the Events for today, tomorrow & the next day (30 points)**

Once you have accepted an event it gets added to your event calendar. The event calendar should display event cards for the current day and the next two days.

**5.5.1 As a user I see an ‘add event’ button if there are no events schedule on that day. (15 points)**

**5.5.2 As a user I can see the event cards for today, tomorrow and the day after tomorrow. (15 points)**

## **5.6 Vertical Scrolling Event Section (Individual) (40 points)**

Only displaying the events for the next 3 days is limiting. A user might want be able to view all the events in month by scrolling up and down in that section.

**5.6.1 As a user I can vertically scroll through all the days in the month. (40 points)**

Your solution should not load all of the events at once. It should instead do adaptive loading and load the events only when they are needed. You should also handle the cases relating to variable number of days in each month.

## **5.7 Horizontal Swiping by Month (Individual) (40 points)**

Wouldn't it be great if users were also able to scroll horizontally? So, they could see the dinner calendar months in advance.

**5.7.1 As a user I can swipe horizontally to view the events in the upcoming months. (40 points)**

You should not load all of the events for the year. But instead you should do adaptive loading and only load the events for the next and previous month.

## **5.8 Horizontal Swiping by Calendar (Individual) (20 points)**

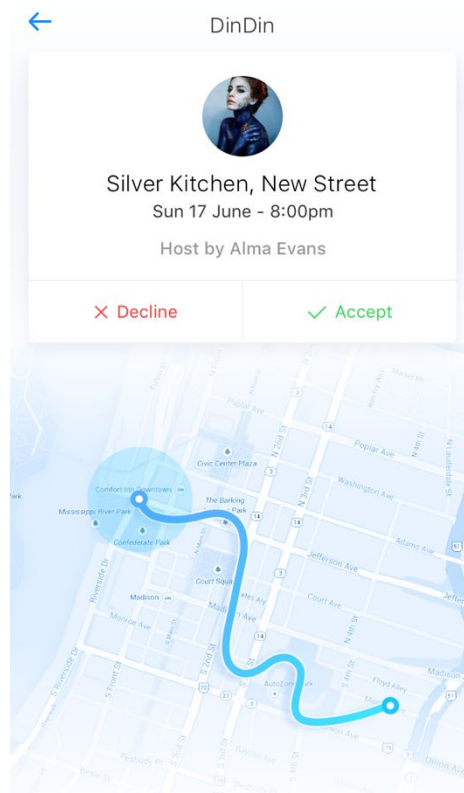
This task can only be completed if the previous task is also completed. It would be amazing if users could see a top calendar section that lets them know what month are currently viewing.

**5.8.1 As a user I can see a bar at the top that indicates the month I am currently viewing, the bar also shows the previous two months and the next two months. (20 points)**

The previous two months and the next two months should be greyed out.

## 6. Invitation Details Screen (125 points - Individual)

The screen below shows the design for the invitation screen. This is the screen that the user will see when they click on any part of the invitation card (not including the Accept or Decline sections). The screen is intended to provide the user with an overview of the events as well as a map including directions on how to get to the event.



### 6.1 Click on the invitation card. (20 points)

Clicking on the invitation card on the home screen should take you the invitation details screen above.

**6.1.1 As a user I can navigate to the invitation screen by clicking on the invitation card on the home screen. (20 points)**

## **6.2 Display Map with location of the event marked. (25 points)**

To make it easier for a user to decide if they will accept a dinner invitation, we would like to show them a map of location relative their current location. You can assume that there is a web endpoint that will return longitude and latitude of the event.

**6.2.1 As a user I can see a marker on the screen displaying the location of the current event. (25 points)**

## **6.3 Display your current location on map. (30 points)**

In addition to displaying the location of the event. It would also be useful, if the user can see the current location.

**6.3.1 As a user I can see my current location along with location of the event on a map. (30 points)**

## **6.4 Display a single Invitation Card (50 points)**

You need to extend your custom invitation card view from the home screen.

**6.4.1 As a user I am able to see a single pending invitation card. The detailed view of an invitation should have a consistent state, i.e. if an invitation is accepted, in the home screen, the detailed view of that invitation should have status accepted; same with pending; but no detail view for rejected invitations. (20 points)**

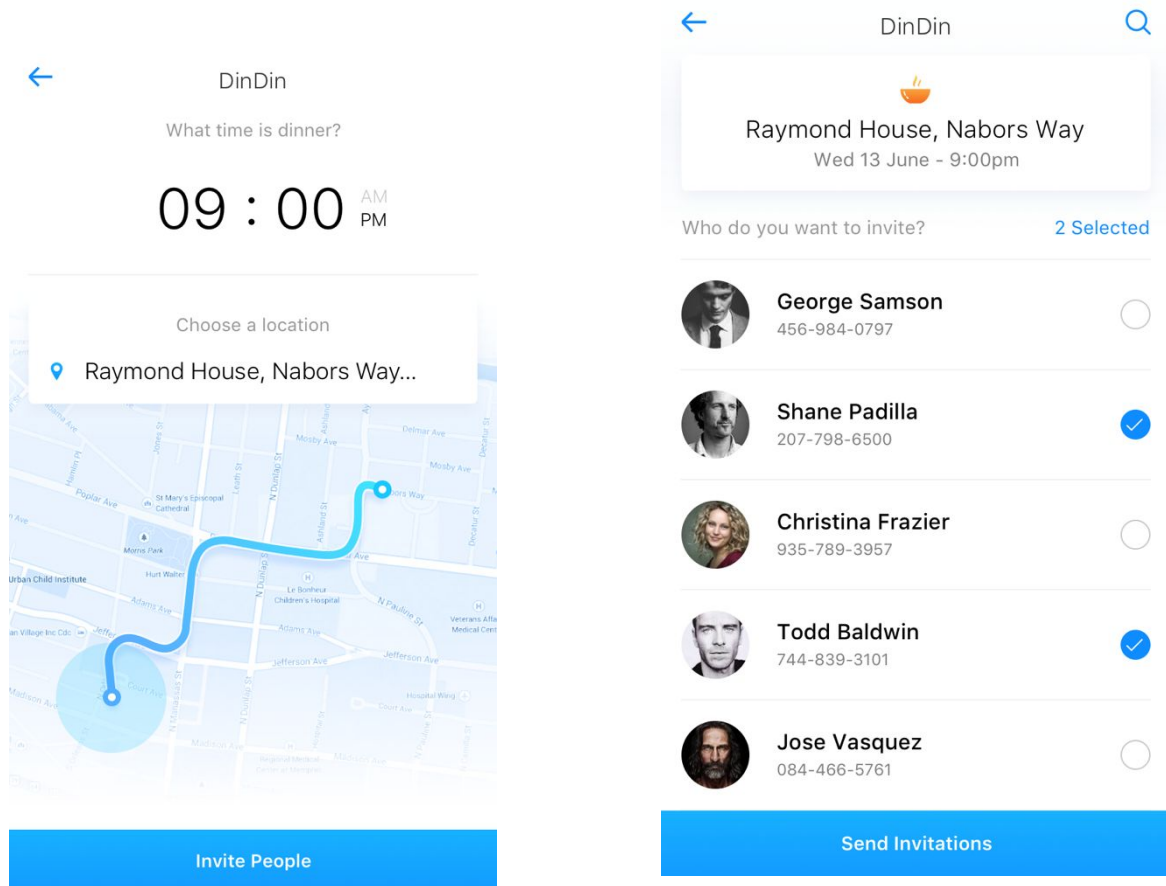
**6.4.2 As a user I can see the avatar of the person who is inviting me to dinner. (10 points)**

**6.4.3 As a user I can see the name of the person who is inviting me to dinner. (10 points)**

**6.4.4 As a user I can see the date and time of the dinner. (10 points)**

The date time should be in the following format <Day of Week> <Day number> < Month Name> - <Time am/pm>

## 7. Create Event (320 points – Individual)



One key component of the application is that users should be able to create new dinner events and invite their friends. In this section we discuss the requirements for the invitations.

### 7.1 Add dinner time (40 points - Individual)

A key part of being able to create a dinner event is being able to add the dinner time.

**7.1.1 As a user, I can select the hour and minute of my dinner. (10 points) (Individual)**

**7.1.2 As a user I can select the hour and minute using a scrolling wheel (30 points) (Individual)**

## **7.2 Mapping (100 points - Individual)**

Providing the user with visual feedback is great way to ensure that information is entered correctly

**7.2.1 As a user I can enter my address in the address box below to choose the location label (25 points) (Individual)**

**7.2.2 As a user I can click on the location pin to enter my current location. (30 points) (Individual)**

Typing an address is tedious. It would be great if a user was able to add the current address. There are several APIs that allow you to convert longitudes and latitudes to addresses.

**7.2.3 As a user, when I enter an address it automatically gets displayed on the MAP. (Individual) (20)**

**7.2.4 Clicking on the button leads to the invite people screen. (5 points) (Individual)**

**7.2.5 As a user, I am able to click on 'invite people' button and send invitations. (5 points) (Individual)**

**7.2.6 As a user, I am able to click on the back button to lead me back to the main screen. (5 points) (Individual)**

**7.2.7 As a user, I am able to navigate between the create event screen and main screen. (5 points) (Individual)**

## **7.3 Display send invitation screen (160 points – Individual)**

The create event screen should display a list of people to invite, each with a clickable select.

**7.3.1 As a user, I should be able to see the name, phone number and avatar of each possible guest. (10 points) (Individual)**

**7.3.2 As a user, I should be able to toggle each guest for selected/deselected status. And the selected count should update onChange. (15 points) (Individual)**

**7.3.3 As a user, I should be able to click on “magnifier” button to search for a specific guest. Search opens an input text-bar and the guests names satisfying the search criteria should display as a list under search. (optional maybe?) (30 points) (Individual)**

**7.3.4 The invite guest screen should have consistent info with the create event screen. (15 points) (Individual)**

**7.3.5** As a user, I should be able to see the previously entered event info from the ‘create event’ screen on the ‘send invitations’ screen. (15 points) (Individual)

**7.3.6** The send invitation button should update the database, prompt a successful/failure message and then redirect back to event detail screen if successful, and remain on send invitation/create event screen if failure. (15 points) (Individual)

**7.3.7** As a user, I should be able to click on the ‘send invitation’ button and it correctly updates guest info in firebase. (15 points) (Individual)

**7.3.8** As a user, I should see a success/failure message upon send invitation. (optional maybe) (15 points) (Individual)

**7.3.9** As a user, I should be redirected to the event details screen if invitations are sent successfully and should also be able to see newly created events in the calendar on main screen. (15 points) (Individual)

**7.3.10** As a user, I should remain on create event/send invitation screen if creating an event results in a failure (not likely but as a failsafe for weird errors). (15 points) (Individual)

## **7.4 The back button (20 points - Individual)**

**7.4.1** As a user, I should be able to click on back button to get back to create event screen and be able to edit the info previously entered. (20 points) (Individual)

## **8. Your Event Details Screen (110 Points - Group)**

### **8.1 Display newly created event (30 points)**

**8.1.1** As a user, I should be able to see my event info in event detail screen. (15 points)

**8.1.2** As a user, I should be able to see a list of invitations sent and its respective status (accepted/pending/rejected). (15 points)

### **8.2 The edit button works (20 points)**

**8.2.1** As a user, I should be able to edit event info by clicking on the edit button. The button should lead me to create event screen. (20 points)



### 8.3 The cancel button works: (20 points)

8.3.1 As a user, I should be able to cancel event by clicking on the cancel button. Clicking on button should completely delete this event both locally and in the database. (20 points)

### 8.4 Pop-up confirmation (30 points)

8.4.1 As a user, when I click on cancel, I should see a warning message that asks me to confirm/discard. Confirm deletes the event while discard leads back to event detail screen. (30 points)

### 8.5 The back button works (10 points)

8.5.1 As a user, I should be able to go back to the main screen by clicking on back button. (10 points)

## Appendix A: Points Maps

### Group Points to Grade Map

Points	Percentage
175	100%
150	90%
125	80%
100	70%
75	60%
50	50%

### Individual Points to Grade Map

Points	Percentage
200	100%
175	90%
150	80%
125	70%
100	60%
75	50%