uCal Backlog

Andrew Bostock, Ben Collier, Matt Kramer, Eli Scherrer, Maneesh Tewani, Hector Trevino

Problem Statement:

People have very hectic lives, and current calendars are only good at doing a subset of things such as adding and removing events. In addition, when creating events with other people, it's difficult to find a common time to meet. People need a better way to keep track of all of the events going on in their life. Our application provides a way for people to find available times for others to meet, the ability to add and remove events, and much more.

Background Information

When working on group projects or simply trying to find common free time for social gatherings, sometimes it can get difficult and frustrating to get everyone's schedules to align. Currently with calendar applications people can create and set events and reminders for themselves but when it comes to a group or collaborative meetings, not many applications deal with these problems.

Environment

We will use HTML and CSS to write and design the static pieces and Angular 5.X for dynamic parts of our front-end. On the backend, we plan on using the Node.js server framework, express and MongoDB to store information.

Functional Requirements

Backlog ID	Functional Requirement	Hours
1	As a user, I would like to be able to import my Google Calendar.	2
2	As a user, I would like to be able to import my iCalendar.	2
3	As a user, I would like to be able to add new events.	4
4	As a user, I would like to be able to create multiple calendars.	3
5	As a user, I would like to be able to create a group.	3
6	As a user, I would like to be able to manage my group.	5
7	As a user, I would like to be able to share my calendars.	3
8	As a user, I would like to be able to schedule group meetings.	4

9	As a user, I would like to be able to see the best times that work for all group members.	5
10	As a user, I would like to be able to remove events.	2
11	As a user, I would like to be able to login using Google.	3
12	As a user, I would like to be able to login using Facebook.	3
13	As a user, I would like to be able to logout.	2
14	As a user, I would like to be able to invite people to my events	3
15	As a user, I would like to be able to RSVP for events.	4
16	As a user, I would like to be able to edit events.	2
17	As a developer, I would like to test and inspect the code and UIs.	100
18	As a user, I would like to be able to add locations to events.	5
19	As a user, I would like to find information about uCal	3
20	As a user, I would like to go to a settings page to modify settings	3
21	As a user, I would like to view my calendar at a yearly level	3
22	As a user, I would like to view my calendar at a monthly level	2
23	As a user, I would like to view my calendar at a weekly level	2
24	As a user, I would like to view my calendar at a daily level	2
25	As a user, I would like to view profile and group information	3
26	As a user, I would like this application to be a progressive web app	4
27	As a user, I would like the app to be responsive on mobile	5
28	As a user, I would like to get reminders about events	2
29	As a user, I would like to be able to share events with people	2
	Total	186

Non-Functional Requirements

Backlog ID	Non-Functional Requirement	Hours
30	As a user, I would like my information to be private and secure.	10
31	As a developer, I would like to learn how to use Angular	10
32	As a developer, I would like to learn how to use MongoDB	10
33	As a developer, I would like to learn how to use TypeScript	10

Use Cases

Case: Google Login

Action	**System Response**
1.Click Google button	2. Open Google login
3. Input credentials	4. Login user
	5. Close Google login

Case: Facebook Login

Action	**System Response**
1.Click Facebook button	2. Open Facebook login
3. Input credentials	4. Login user
	5. Close Facebook login

Case: Logout

Action	**System Response**
1.Navigate to login screen	3. Pull up user login screen
2.Select sign out button	

Case: Create Event

Action	**System Response**
1. Fill in Event Information	
2. Click add Event	3. Validate Login
	4. Store Event in the Database

Case: RSVP

Action	**System Response**
1. Accept/Decline RSVP	2. Update Event Registry

Case: Access Help Page

Action	**System Response**
1. User selects Help Page	2. Help page is properly displayed

Case: Access Settings Page

Action	**System Response**
1. User selects Settings Page	2. Settings page is properly displayed

Case: Edit Event

Action	**System Response**
1. User selects event to edit	2. Event is pulled up
3. User begins to modify text	4. Text boxes for all editable values are on page
5. User selects save command	6. Information is saved and updated to reflect changes

Case: Event Reminder

Action	**System Response**
1. User turns on reminders and sets a date / time for an event	2. The system sends the user a notification reminding them about the event

Case: Yearly View

Action	**System Response**
1. User clicks on view dropdown.	2. Dropdown of different view options shows up
3. User clicks on the “Yearly” button	4. The UI updates Showing all 12 months

Case: Monthly View

Action	**System Response**
1. User clicks on view dropdown.	2. Dropdown of different view options shows up
3. User clicks on the “Monthly” button	4. The UI updates Showing one month

Case: Weekly View

Action	**System Response**
1. User clicks on view dropdown.	2. Dropdown of different view options shows up
3. User clicks on the “Weekly” button	4. The UI updates Showing one week

Case: Daily View

Action	**System Response**
1. User clicks on view dropdown.	2. Dropdown of different view options shows up

3. User clicks on the “Dailey” button	4. The UI updates Showing one day

Case: Import Calendar

Action	**System Response**
1. User clicks on the import calendar button	2. The UI displays a screen that the user can put their information into to get the calendar
3. The user submits the information	4. The calendar view updates showing the imported events

Case: Sharing a Calendar

Action	**System Response**
1.User clicks on the share calendar button	2. The UI displays a screen that the user can put their information into to share the calendar
3. User submits the information	4. The calendar view updates showing a list of shared users

Case: Create a group

Action	**System Response**
1. The user clicks on the create group button	2. The UI shows a new screen that prompts the user to set basic information about the group, like the name and it’s members
3. The user inputs valid information	4. The group is created and the user is directed to the group page

Case: Manage groups

Action	**System Response**
1. The user clicks on the group they would like to edit	2. The UI displays a new screen with all of the group information

3. The user can edit the group members and events	4. The system responds accordingly to the user's edit request

Case: Log-in through Facebook

Action	**System Response**
1.On the log-in page the user selects the "Log-in with Facebook option"	2. A UI will pop up to the user that prompts them for facebook login information
3. User enters in their Facebook log-in information	4. Textboxes will reflect the typed in keys (with filler characters for password)
5. User clicks log-in button	6. With the provided credentials, a log-in request is made

Case: Log-in through Google

Action	**System Response**
1.On the log-in page, the user selects the "Log-in with Google" option	2. A UI will pop up to the user that prompts them for google login information
3. User enters in their Google log-in information	4. Textboxes will reflect the typed in keys (with filler characters for passwords)
5. User clicks log-in button	6. With the provided credentials, a login request is made

Case: Logout

Action	**System Response**
1. The user clicks the logout button	2. The UI returns to the login page

Case: Share an event with someone

Action	**System Response**
1. The user clicks on an event on their calendar	2. The UI shows event details

3. The user selects the “share” button	4. The UI prompts the user to select who they would like to share the event with
5. The user selects one (or more) people to share the event with	6. The server sends a request to the other users asking them to accept the event to their calendar

Case: Invite someone to an event

Action	**System Response**
1. The user clicks on an event on their calendar	2. The UI shows event details
3. The user selects the “share” button they also check the “Request RSVP” box	4. The UI prompts the user to select who they would like to invite to the event
5. The user selects one (or more) people to invite to the event	6. The server sends a request to the other users asking them to accept / respond to the event

Case: Reminders for Event

Action	**System Response**
1. When creating an event, user selects the reminders option	2. UI displays drop-down option for when to receive reminders about event and through push notifications or email
2. User selects when to receive reminders and notification choice	4. Computer sends reminder at selected time through push notifications or email depending on option selected

Case: Create multiple calendars

Action	**System Response**
1.User selects create new calendar option	2. UI prompts the user for information
3. User fills in all information and selects “Create Calendar”	4. New calendar is generated and it is added to the page
5. Buttons to view each calendar are displayed	6. If name of calendar is selected that is brought to the page for view/edit

Case: View profile and group information

Action	**System Response**
1. User clicks on the “Profile and Group pages”	2. The UI displays all of the User and group information

Case: Using the website on a phone

Action	**System Response**
1. User navigates to the website on their phone	2. The server delivers a responsive UI

Case: Using the website as a progressive web app on a compatible OS

Action	**System Response**
1. User navigates to the website on their phone	2. The server delivers a responsive UI
3. User clicks “Add to home screen”	4. The server responds with the webAPK and installs on the phone