Product Backlog

Elijah Hauber, Pritesh Kadiwala, Brian Rhee, Han Wang, Jaeyeon Kim

Problem Statement

Currently at Purdue there are many students who find it difficult to figure out where the events from organizations/groups are taking place. As for now, there is Boilerlink which provides information for events taking place around the globe for students but not particularly on campus. With Boilerlink students have an arduous time finding new activities specifically on campus. Our app will provide a better experience for Purdue students as this app will be focused on informing students about the events taking place around campus, as well as providing a platform for students to setup events from scratch. In this way, students will be able to register for the event quicker without having the hassle to contact someone personally to know more about the event.

Background Information

While there are other apps like BoilerLink to provide students with the information about several events taking place, it is still not convenient for the students to find these events on campus which are organized by the student organizations and students themselves. So our goal is to reduce the difficulty for registration for events and for the students to have a clear idea about the events where information such as the place of the event, who have registered for it and any changes that take place prior to the event.

Environment

Rally will be a standalone web application. The application will be a web application that is accessible through desktop browsers and mobile devices. The backend will keep track of valid user logins and the users' friends. The frontend GUI for Rally will be designed using HTML/CSS, Vue and VuetifyJS. User data would be backed up both on a firebase backend and in the form of a JSON file to the user's local machine.

Requirements

Functional

ID	Function Requirement	Hours	Status
1	As a user, I would like to open the application through different internet browsers	15	Planned for Sprint 1
2	As a user, I would like to Sign up	10	Planned for Sprint 1
3	As a user, I would like to Login	5	Planned for Sprint 1
4	As a user, I would like to Logout	5	Planned for Sprint 1
5	As a user, I would like to have Settings function	10	Planned for Sprint 1
6	As a user, I would like to delete my account	10	Planned for Sprint 1
7	As a user, I would like to change my password	5	Planned for Sprint 1
8	As a user, I would like to have my own personal profile for application	15	Planned for Sprint 1
9	As a user, I would like to have a tutorial on the usage of the app	20	Planned for Sprint 2
10	As a user, I would like to view the events I am registered for	20	Planned for Sprint 2
11	As a user, I would like to filter the events by time	20	Planned for Sprint 2
12	As a user, I would like to filter the events by tag	10	Planned for Sprint 2
13	As a user, I would like to view available events	10	Planned for Sprint 2
14	As a user, I would like to be able to login with Facebook	20	Planned for Sprint 1
15	As a user, I would like to be able to login with Twitter	20	Planned for

			Sprint 1
16	As a user, I would like to get notifications if there changes in the events	15	Planned for Sprint 2
17	As a user, I would like to be able to donate through PayPal	10	Planned for Sprint 1
18	As a user, I would like to be able to login with Google	15	Planned for Sprint 1
19	As a user, I would like to be able to organize an event	15	Planned for Sprint 2
20	As a user, I would like to be able to edit the event information	20	Planned for Sprint 2
21	As a user, I would like to be invite a friend through email	20	Planned for Sprint 2
22	As a user, I would like to have a helper tool when I do something wrong	30	Planned for Sprint 2
23	As a user, I should be able to store personal information such as Date of birth and etc.	10	Planned for Sprint 1
24	As a user, I would like to edit my personal information such as username and etc.	10	Planned for Sprint 1
25	As a user, I would like to upload image for my profile.	10	Planned for Sprint 2
26	As a user, I would like to upload image for the event	10	Planned for Sprint 2

Non-Functional

- Application will work with multiple browsers.
- Application will be interesting, instinctive, and visually appealing user interface.
- Application will be easy to use and robust.
- Application will have pop ups if any error occurs.
- Application will have cleanly organized and well-documented code.
- Application will be maintainable and testable.
- Application will store the required information safely and securely.

Usability

The app will be visually appealing on most browsers which would relate to the CSS framework we use for making the app and it will be accessible for all users. As we are using Vuetify, a framework similar to Bootstrap, it would make the app easy to use for all users as they wouldn't have a complicated procedure for any task they perform on the website. We will make sure that there is a seamless integration with Firebase so that the user can send and retrieve the information easily.

Security

Security is not critical for Rally as the details stored about the innocent user in their profile will not allow an adversary to steal an innocent user's identity. However, we will take steps to make sure only the innocent user can access their account. Firebase is hosted on a Secure Sockets Layer and establishes an encrypted link between the User and us, the host. We will expatiate the Firebase rules to ensure the security of our Firebase data. When the user forgets their password we will not tell them if it was the password or username that failed the verification.The failed verification message to display will be vague like "Invalid username and/or password." This prevents attackers from brute-forcing valid usernames without knowing their passwords.

Scalability

The system should be able to handle about 8,000 user efficiently where the system would have fast response time and bandwidth. Due to us using Vue.JS, we will have a fast response time as this framework uses a lightweight virtual DOM, so it functions asynchronously which makes it faster. Additionally, Firebase is hosted on Google servers and this will allow for fast access as well.

Use Cases

Case 1: Open the application

Action	System Response
1. Navigate to application link	2. Application opens, takes user to login if not logged in and profile if logged in

Case 2: Create an account (Sign up)

Action	System Response
1. Input credentials to text boxes	2. Confirms signup, takes user to tutorial
Username must be a valid email address between 3-254 characters. Username must not already be in use. Password must be 8-30 characters and the password confirmation field must match. Email must not contain “@”, “.”, “ “.	Otherwise displays an error message

Case 3: Login to account

Action	System Response
1. Input credentials	2. Confirms successful login
Username must be a valid email address between 3-254 characters. Username must be an existing account.. Password must be 8-30 characters.	Otherwise displays an error message

Case 4: Logout of account

Action	System Response
1. Click the logout button under account in the navigation bar	2. User is logged out and taken back to login page

Case 5: Settings function in personal profile

Action	System Response
1. Click the Settings button under account page in the navigation bar	2. User gets navigate to Settings

Case 6: Delete account

Action	System Response
1. Choose delete account from account in the navigation bar	2. Open a prompt for confirmation of account deletion
3. Confirm account deletion	4. Confirms account is deleted, takes user to login page

Case 7: Change password

Action	System Response
1. Click change password under account in the navigation bar	2. Takes user to change password page
3. Input login, current password, and new password	4. Confirms current password is correct and acknowledges a new password was set Otherwise displays an error message
Username must match current user. Old password must match current password. New password must be 8-30 characters.	

Case 8: View personal profile

Action	System Response
1. Click profile in the navigation bar	2. Display user's information; e.g. age and favorite hobby
User may choose to leave some fields blank.	

Case 9: Launch tutorial for entire app

Action	System Response
1. User logs in for the first time or if the user clicks the tutorial button in the navigation bar	2. Takes user to the tutorial
3. User clicks play on the tutorial video	4. Video begins to play

Case 10: View events registered for

Action	System Response
1. Click on My Events in the navigation bar	2. Display all events that users has signed up.
If user has no event registered, My Event is blank.	

Case 11: Filter events by time

Action	System Response
1. Click on recent week in the filter bar	2. Filter events within a week's time
3. Click on recent month in the filter bar	4. Filter events within a month's time
5. Click on recent year in the filter bar	6. Filter events within a year's time

Case 12: Filter events by tag

Action	System Response
1. Type in tag in the filter bar	2. Filter events by typed in tag

A valid tag must have been previously created by the user.

Case 13: View available events

Action	System Response
1. Click on available events	2. Filters all filtered events that are still available for sign up.
3. Click on available events again	3. Unfiltered all filtered events that are both available and unavailable.

Event must have been previously created by the user.

Case 14: Login with Facebook

Action	System Response
1. User clicks login with Facebook button	2. System accesses Facebook login API

Facebook user must exist.

Case 15: Login with Twitter

Action	System Response
1. User clicks login with Twitter button	2. System accesses Twitter login API

Twitter user must exist.

Case 16: Receive notifications when there are changes to an event

Action	System Response
1. User edits event information	2. System sends notification to all registered users that an event was changed.

Case 17: Donate through Paypal

Action	System Response
1. User clicks the donate through Paypal button	2. System shows a Paypal donation form
3. User enters an amount and hits send	4. System talks to Paypal api to transfer donation to Rally.

Case 18: Login with Google

Action	System Response
1. User clicks login with Google button	2. System accesses Google login api

Google user must exist.

Case 19: Organize/Host an event

Action	System Response
1. User clicks host an event button	2. System shows a event creation form
3. User enter name, time, location, description, number of slots and clicks Create Event	4. Event information is created and uploaded.

Case 20: Edit event information

Action	System Response
1. User (Host) clicks the edit button for their event	2. Editing form appears for event
3. User (Host) clicks the update button	4. System updates event database

Case 21: Invite friend through email

Action	System Response
1. User clicks on invite through email button	2. System uses EmailJS.com to send email to target email

Case 22: Helper tool that assists user

Action	System Response
1. User attempts an invalid action	2. Bot appears in UI with a suggestion or reason why the action was invalid

Case 23: Store personal information as a user

Action	System Response
1. User enters personal information	2. System shows the user their personal information on their account page
3. User clicks update	4. System stores and updates user database

Case 24: Edit personal information as a user

Action	System Response
1. Clicks edit personal information button	2. Editing Personal Information form appears
3. User enters and submits form	4. System updates user database with updated information

Case 25: Upload images as a user

Action	System Response
1. Clicks upload image for profile	2. Image upload prompt for browsing computer appears
3. User clicks browse computer	4. System shows computer search browser
5. User selects image and clicks upload	6. System uploads image to database and updates user database

Case 26: Upload images for an event

Action	System Response
1. Clicks upload image for profile	2. Image upload prompt for browsing computer appears
3. User clicks browse computer	4. System shows computer search browser
5. User selects image and clicks upload	6. System uploads image to database and updates event database