

Product Backlog

Team 14: CoconutWatr

January 26, 2017

Team Members

Kate Wang, Dominic Valentine, Wilson Husin, Jiho Choi, Kyle Leaf

Problem Statement

Students have trouble finding people in the same class to study with, and they need an application that will connect users with other students taking the same class for study buddies.

Background Information

Students procrastinate in their responsibilities. A lot of students go through college with studying while working and they tend to neglect the things they should do at home such as homework, studying for quizzes, etc.

For a lot of students, their way to keep up with the school work is to have friends in the same boat, with the same schedule and plan. Having a study group would encourage them to do their assignments.

Unless you always say “Hi” to whoever is sitting next to you in class, there isn’t an efficient way to find a study group. We provide the easy, harmless way to communicate with classmates and organize study sessions.

Environment

We are planning to build a web application to handle user information and connect different users with front-end implemented using HTML5/CSS3 with AngularJS, and Bootstrap framework. We will handle user data and system information using MongoDB and Node.js with Express in back-end implementation. GitHub will be utilized as our version control and code backup tool. We will be using Stormpath to handle user accounts and user information.

Functional Requirements

ID	Functional Requirement	Hours	Status
1	As a user, I would like to login with Google Account.	7	Planned for sprint 1

2	As a user, I would like to be able to see a summary of what has been going on in my classes (something like news feed).	15	Planned for sprint 1
3	As a user, I would like to be able to add my classes.	8	Planned for sprint 1
4	As a user, I would like to be able to create a class if it does not exist.	6	Planned for sprint 1
5	As a user, I would not want to see history of previous semester of a class.	8	Planned for sprint 2
6	As a user, I would like to be able to remove class from my feed (in case I drop the class).	6	Planned for sprint 1
7	As a user, I would like to be able to see the classmates in the same class.	6	Planned for sprint 1
8	As a user, I would like to be able to communicate with classmates in the same class (public channel of the class).	10	Planned for sprint 1
9	As a user, I would like to be able to create a subgroup to chat with certain users in the same class.	11	Planned for sprint 2
10	As a user, I would like to be able to leave a subgroup.	9	Planned for sprint 2
11	As a user, I would like to be able to invite classmates to the subgroup.	10	Planned for sprint 2
12	As a user, I would like to be able to accept the invitation of a subgroup.	7	Planned for sprint 2
13	As a user, I would like to be able to post a message in a specific conversation.	6	Planned for Sprint 1
14	As a user, I would like to be able to post reminder of an assignment that will be due.	16	Planned for sprint 2
15	As a user, I would like to be able to create public events with important information.	10	Planned for sprint 1
16	As a user, I would like to be able to edit the public events that I made.	5	Planned for sprint 1
17	As a user, I would like to be able to delete the public events that I made.	7	Planned for sprint 1
18	As a user, I would like to see the public event of others	6	Planned for sprint 1

	had created.		
19	As a user, I would like to be able to RSVP to events.	8	Planned for sprint 1
20	As a user, I would like to be able to see who is attending an event.	6	Planned for sprint 1
21	As a user, I would like to be able to see who created an event.	4	Planned for sprint 1
22	As a user, I would like to be able to log in using my Facebook account. (If time permits)	9	Planned for sprint 2
23	As a user, I would like to have a help section in the application to provide more detailed explanations of functionality. (If time permits)	15	Planned for sprint 2

Non-Functional Requirements

1. As a user, I want a visually appealing interface.
2. As a user, I want a web application to be high responsive.
3. As a user, I want an intuitive and easily navigable interface.
4. As a user, I want the app to run smoothly without crashing.
5. As a developer, I want the code to be reusable.
6. As a developer, I want the user data to be securely stored and transferred.
7. As a developer, I want the code to be organized and maintainable.
8. As a developer, I want the code to be properly commented on each function definition.

Use cases

Case: Login in with a Google account

Action:	System Response:
1. Click "Login with Google account button"	2. Google login window opens
3. Enter credentials	
4. Click "Login"	5. Login window closes and brings you to dashboard

Case: Show summary

Action:	System Response:
1. Put login information and click “Login” button	2. Retrieve corresponding information and Display the information in “News Feed” section.

Case: Add a class

Action:	System Response:
1. Click “Manage Classes” button	2. A dialogue window pops up
3. Click “Add Class”	4.. Add class dialogue window opens
5. Enter Class name into “Class name” box	
6. Enter semester into “Semester” box	
7. Click “Add” button	8. Dialogue window opens saying “Class added”

Case: Create a class

Action:	System Response:
1. Click “Manage Classes” button	2. A dialogue window pops up
3. Click “Add Class”	4.. Add class dialogue window opens
5. Enter Class name into “Class name” box	
6. Enter semester into “Semester” box	
7. Click “Add” button	8. Dialogue window opens saying “Class does not exist. Would you like to create it?”
9. Click “Create” button	9. Dialogue window opens saying “Class Created”

Case: User exits a class

Action:	System response
1. Click “Manage Classes” button	2. A dialogue window pops up
3. Click on the “X” next to the group you want to leave	4. A dialogue window opens saying “Are you sure you want to leave this class?”
5. Click on “Leave”	6. Class is removed from your groups list and you

	can no longer see group activity
--	----------------------------------

Case: Checking classmates in the class

Action:	System response
1. Click “Classes” button	2. Drop-down menu, list of classes
3. Click on a specific class	4. Redirect user to specified class
5. Click on members on the right side of the page	6. Redirect user to a page that shows all members

Case: Create a subgroup

Action:	System Response:
1. Click on “Manage Groups”	2. A dialogue window pops up
3. Click on “Create Subgroup”	
4. Enter subgroup data and add classmate	
5. Click “Submit”	6. Dialogue window opens saying “Subgroup Created”

Case: Leave subgroup

Action:	System Response:
1. Click on “Manage Groups”	2. A dialogue window pops up
3. Click on the “X” next to the group you want to leave	4. A dialogue window opens saying “Are you sure you want to leave this group?”
5. Click on “Leave”	6. Group is removed from your groups list and you can no longer see group activity -- groups with 0 members should be automatically deleted.

Case: Invite classmates to subgroup

Action:	System Response:
1. Click on “Manage Groups”	2. A dialogue window pops up
3. Click on the “Invite Classmates”	4. A dialogue window opens showing a list of classmates.
5. Click names of classmates to invite to the	

subgroup.	
7. Click on “Invite”	8. Selected users receive invitations to your subgroup

Case: Accept Invitation to subgroup

Action:	System Response:
1. Invitation shows up on sidebar of "News Feed"	2. Hover to the logo/text
3. Click Accept Invitation	

Case: Post a message in a chat

Action:	System response
1. Click on group you want to message on sidebar	2. Current chat will change to chat you selected
3. Click on select the message bar and begin typing your message	4. Your message will appear as you type it in the message bar
5. Click the send button or press enter	6. Message will show up in the chat and members of that group will be able to view it

Case: Add a reminder of an assignment that is due

Action:	System response
1. Click on group you want to message on sidebar	2. Current chat will change to chat you selected
3. Click on “Add assignment reminder”	4. Create Assignment window opens
5. Enter assignment name	
6. Enter assignment description	
7. Enter link to assignment (if applicable)	
8. Click “Create Assignment”	9. A dialogue window opens saying “Assignment Created”. The assignment can now be found in the group’s assignment section and can be viewed by all group members

Case: Add an event

Action:	System Response:
1. Click “Add Event”	2. A popup box appears
3. Enter event data	
4. Click “Create Event”	5. Dialogue window opens saying “Event Created”

Case: Edit an event

Action:	System response
1. Click “Edit event”	2. A popup box appears
3. Enter edited event data	
4. Click “Edit Event”	5. Dialogue window opens saying “Event Edited”

Case: Delete an event

Action:	System Response:
1. Click the “Delete Event” on the event page	2. A dialogue window pops up saying “Ok” and “Cancel”
3. Click the “Ok” button	4. Dialogue window opens saying “Event Deleted”

Case: See public events

Action:	System response
1. Click on group you want to view events for	2. Current chat will change to chat you selected
3. Click on “View all Events” button on sidebar	4. Window will open displaying brief information about all of the upcoming events for that group

Case: RSVP to events

Action:	System response
1. Click on the group you want to view events for.	2. Current chat will change to chat you selected
3. Click on “View all Events” button on sidebar	4. Window will open displaying brief information about all of the upcoming events for that group

5. Click on the event name you would like to attend	6. Window will open displaying information about the event.
7. Click on the “RSVP” button	8. Add user to the list of classmates attending the event
	9. Add event to user’s upcoming events on their personal dashboard

Case: See who is attending an event

Action:	System response
1. Click on the group you want to view events for.	2. Current chat will change to chat you selected.
3. Click on event name you would like to view who is attending.	4. Window will open displaying brief information about all of the upcoming events for that group including who will be attending.

Case: See who created an event

Action:	System response
1. Click on group you want to view events for	2. Current chat will change to chat you selected
3. Click on “View all Events” button on sidebar	4. Window will open displaying brief information about all of the upcoming events for that group.
5. Click on the event name you would like to see information about.	6. Window will open displaying information about the event including who created it.

Case: Login with Facebook

Action:	System response
1. Click “Login with Facebook account button”	2. Facebook login window opens
3. Enter credentials	
4. Click “Login”	5. Login window closes and brings you to dashboard

Case: Help section of application

Action:	System response
1. At the bottom of the page, click help	2. Redirect to help menu page

3. Pick a topic, click on it	4. Expand explanation
5. Or send an email to the developers	6. "mailto:" link