cluster

Team 15 – Sprint 1 Planning Document

Frederick Li, Shane Li, Shai Mohan, Joseph Singer, David Tong, Andrew Tully

Sprint Overview

Overview

For our first sprint, we will prepare our backend/frontend servers and our database and ensure all components communicate properly. Most of our time will be spent setting up environments, servers, etc. Our frontend team will be creating a landing page and implementing the log in/sign up feature via Facebook. Our backend team will be implementing the database and API routes for information storage and transfer.

Risks/Challenges

Our team is relatively inexperienced with the languages and tools that we will be utilizing. As such, a large portion of the time will be spent learning and developing our skills in these areas. Setting up our servers and database and allowing them to communicate with each other is a critical portion of our project, and will likely occupy the majority of our first sprint. In order to complete all the tasks we have planned, we will need to follow our schedule closely and delegate our workload efficiently.

Scrum Master: Shane Li

Meeting Schedule: Mondays, Wednesdays, and Fridays from 11:30 a.m. - 12:30 p.m.

Current Sprint Detail

User Story #1

As a guest, I would like to be able to view existing clusters.

#	Task Description	Hours	Team	Owner
1	Set up backend server	5	Back	Shane
2	Set up frontend server	5	Front	Joey
3	Set up database	3	Back	Shane
4	Set up API routes	4	Back	Andrew
5	Set up web interface	5	Front	David
6	Implement API call to poll current clusters	3	Front	Shai
7	Display current clusters on landing page	4	Front	Shai
8	Implement MySQL with Spring	4	Back	Andrew
9	Create cluster information storage system	8	Back	Fred

Acceptance Criteria:

- Given that the frontend and backend servers are setup correctly, when the user goes to the website, then they will reach the landing page, which is the portal to the rest of the website.
- Given that the database, website, backend and frontend servers are setup correctly, the user should be able to see a list of current clusters on the landing page.
- Given that the database is setup correctly, when the logic controller makes a database query, then the database will be able to process them successfully.
- Given that the API service and routes are setup correctly, when an API request is made, the logic controller will be able to handle the request properly.

User Story #2

As a guest, I would like to be able to create an account using Facebook.

#	Task Description	Hours	Team	Owner
1	Connect backend with Facebook Authentication	4	Back	Fred
2	Implement Facebook Login Authentication with Spring Security	5	Back	Shane
3	Create a login button on website landing page	2	Front	Shai
4	Redirect users to Facebook Login page	1	Front	Joey
5	Implement frontend logic to send authentication information to backend	10	Front	David
6	Implement backend logic to verify authentication token	10	Back	Andrew
7	Create user information storage system	3	Back	Fred

Acceptance Criteria:

- Given that the backend is properly connected with Facebook Authentication and the landing page has been created, when a user visits the landing page, then they should be able to click a button to create an account using his/her Facebook account.
- Given that the frontend and backend can properly send and verify user authentication information, when a user clicks the “login” button, then they should be redirected to Facebook to sign in and authenticate their session.
- Given that the backend logic has been implemented properly and the database has been setup to store uses’ information, when a user makes an API call, then the backend should be able to verify that the they has been authenticated.

-

User Story #3

As a logged-in user, I would like to be able to view my profile.

#	Task Description	Hours	Team	Owner
1	Implement API to retrieve user profile	3	Back	Fred
2	Create profile page on website	4	Front	Joey

Acceptance Criteria

- Given that a user’s profile information is able to be retrieved and displayed correctly, when a user clicks the “Profile” tab at the top of the screen, then the user will be redirected to their profile page.

User Story #4

As a logged-in user, I would like to be able to start a cluster.

#	Task Description	Hours	Team	Owner
1	Implement API to start a cluster	3	Back	Fred
2	Implement logic to update cluster in database	2	Back	Andrew
3	Implement logic to create a cluster instance	2	Back	Andrew
4	Create "Start a Cluster" page on website	3	Front	David
5	Implement logic to take user input and pass information from frontend to backend	4	Front	Joey

Acceptance Criteria

- Given that the logic to create a cluster instance and update a cluster in the database is implemented correctly, when a user attempts to form a cluster, the cluster will be correctly logged on the backend server and stored in the database.
- Given that the "Start a Cluster" page is implemented correctly, when a user attempts to form a cluster, then they will be redirected to a page containing a form for cluster information and preferences.
- Given that the logic to take user input and pass the information from the frontend to backend is implemented correctly, when a user inputs information into the form on the "Start a Cluster" page, the data is correctly stored in the database.

User Story #5

As a logged-in user, I would like to be able to log out.

#	Task Description	Hours	Team	Owner
1	Implement API to receive sign out requests	4	Front	Shai
2	Implement logic to deauthenticate user	2	Back	Shane
3	Create sign out button	1	Front	Joey

Acceptance Criteria:

- Given that the API for receiving logout requests is implemented correctly, when the user attempts to log out then they will click a "Sign Out" button.
- Given that the "Sign Out" button is implemented correctly, when the user clicks the button, then they will be deauthenticated and treated as a guest.

User Story #6

As a logged-in user, I would like to be able to view and join existing clusters.

#	Task Description	Hours	Team	Owner
1	Implement all APIs to send all information of clusters	4	Back	Shane
2	Implement API to receive join requests	5	Back	Andrew
3	Implement logic to properly update cluster and user information in database	2	Front	David
4	Create page to show all existing clusters with basic information	1	Front	Shai
5	Create page to show current cluster information after user clicks join	4	Front	Joey

Acceptance Criteria:

- Given that the APIs are implemented properly, when a user selects a cluster to join, the backend should add the user to the cluster and provide information for the frontend to generate a view for the user.
- Given that the logic is implemented properly, when changes to user and clusters are made, the changes will be written into the database and updated properly.
- Given that the frontend is implemented properly, when the user goes to cluster information, the user will see all the information necessary.

User Story #7

As a logged-in user, I would like to be able to leave a cluster

#	Task Description	Hours	Team	Owner
1	Implement all APIs to receive leave requests	4	Back	Shane
2	Implement logic to properly remove a user from a cluster	5	Back	Fred
3	Create a button to leave a cluster and redirect to the dashboard	2	Front	Joey

Acceptance Criteria:

- Given that the website properly implements a leave button, when the user clicks the button, then it will remove him from the cluster and return him/her to the list of cluster groups.
- Given that a user is properly removed from a cluster, when the cluster has zero members, the cluster will be deleted.

Remaining Backlog

By the conclusion of Sprint 1, we intend to complete 7 of 26 user stories.

As a guest, I would like to be able to:

- ~~view existing clusters.~~
- ~~create an account using Facebook.~~

As a logged-in user, I would like to be able to:

- ~~view my profile.~~
- ~~view and join existing clusters.~~
- sort clusters by newest, distance, or alphabetically by restaurant name.
- view information about a cluster and its members.
- ~~request to start a cluster.~~
  ~~i) specify where I would like to pickup the order.~~
  ~~ii) choose a restaurant to order from.~~
  ~~iii) set minimum cluster size.~~
  ~~iv) set maximum cluster size.~~
  ~~v) set a duration time for the cluster.~~
- ~~log out.~~

As a cluster leader, I would like to be able to:

- *have all the capabilities of a cluster member.*
- approve/reject users joining cluster.
- edit information about my cluster.

As a cluster member, I would like to be able to:

- *have all the capabilities of a logged-in user.*
- view restaurant information about my cluster.
- view pickup/meeting location for my cluster.
- view information about all other members of my cluster.
- leave ratings for other members of my cluster.
- ~~leave a cluster.~~

As an administrator, I would like to be able to:

- disband clusters.
- review order history.
- review cluster history.
- view number of clusters active.
- view number of users online.
- view total number of orders completed.
- view total number of orders not completed.
- view user information.
- ban users (if time permits).