Team 19 - Sprint 1 Planning Document

CS 30700 - Team 19

Samrat Reddy Nalla
Seohyun Ahn
Youngjoon Park
Kuan-Ting Wu

Table of Contents

Sprint Overview

In this sprint, our goal is to produce 8 of the user stories drafts. where most of it is based on the looks of each page as the user goes into the app. He or she will be able to login using Facebook or Google. After logging in the user will be able to search for all the available parties around him or her. Also, there will be constant updates which will give the user information about other users that have joined a party. A user will be able to look for restaurants based on selected tastes and search up reviews about them.

Scrum Master

SeoHyun Ahn

Meeting Schedule

Tuesday - 1:30 pm to 3:00 pm
Wednesday - 5:30 pm to 7:00 pm
Thursday - 1:30 pm to 3:00 pm
Friday - 10:30 pm to 11:30 pm

Risks/Challenges

Most of our team members are not familiar with the technologies that we need to use for this project such as XCode, Swift, Firebase, and SQL. For our first sprint, we will work towards familiarizing ourselves with the new languages in order to work efficiently. Since some members do not have experience working as a group we will coordinate with each other in order to work systematically during this sprint.

Current Sprint Details

User Story #1

As a user, I would like to login to the System with my existing Google and Facebook account.

#	Task description	Time	Owner
1	Pre-environment setup	6 hr	Arthur
2	Set up back end server	4 hr	Arthur
3	Design the login page	5 hr	SeoHyun
4	Connecting back end and login page	6 hr	Youngjoon
5	Testing	3 hr	Arthur
	Total	24 hrs	

Acceptance Criteria

→ Given that the login page, environment and the server have been successfully setup, the user will be able to successfully login with his or her Google and Facebook account.
→ Given that the login success, it should automatically store the user's information to the server.
→ Given that the login fails, a proper error message should show up.

User Story #2

As a user, after logged in, I would like to be able to have a main page to see all the available parties around me

#	Task description	Time	Owner
1	Implement Google Map Api in the main page	7 hr	SeoHyun
2	Implement function to get coordinate data from the server	6 hr	Youngjoon
3	Implement function to unpack data and load them into variables	7 hr	Samrat
4	Testing	4 hr	Arthur
	Total	24 hrs	

Acceptance Criteria

→ Given that the Google Map implementation is correct, when the users view the map they would be able to see what their current location is.

→ Given that the application is receiving information from the server, the application should check every variable within the view that has data in it.

→ Given that the user is viewing the main page, user will be able to see the parties that are available around the user.

→ Given that the user is viewing the map page, when he or she clicks on other parties, each party's data can be viewed

ok

User Story #3

As a user, I would like to see different restaurants around me filtered by tastes, and view reviews about them

#	Task description	Time	Owner
1	Implement Yelp API into the app	8 hr	SeoHyun
2	Retrieve data from the server	5 hr	Youngjoon
3	Unpack data and load them into variables	8 hr	Arthur
4	Connecting data to the UI	8 hr	Arthur
5	Testing	2 hrs	Samrat
	Total	31 hrs	

Acceptance Criteria

→ Given that the Yelp API is functional, when the server receives the data from Yelp, the ratings of the restaurants will be visible.

→ Given that the data from the server is processed properly, the data should be ordered by ratings, names, places and tastes.

→ Given that the page is implemented, when a user views the page, the page is listed out via ratings from the Yelp API.

User Story #4

As a user, I would like to freely chat within the group and individual parties

#	Task description	Time	Owner
1	Implement functionally for communicate with the back end communicating server	7 hr	Samrat
2	Implement ability to automatically sync with every client within the chatroom	6 hr	Youngjoon
3	Implement options for users to freely use within the group chat room including carpool, photos, or announcements	8 hr	SeoHyun
4	Implement functionality for a user to add another user as friend	8 hrs	SeoHyun
5	Testing	2 hr	Youngjoon
	Total	31 hrs	

Acceptance Criteria

→ Given that the back end server is set up, when there is a request to send a message within the party, that information will appear on the server.

→ Given that the chat room has been set up, users should be able to view any incoming messages within the proper time frame and will be able to upload any kind of photos within the chat room.

→ Given that both users are inside the chat room, a user will be able to add another user to his or her friend list.

→ Given that the chat room has been set up, a user will be able to share the party code to their friend in order to join the party.

User Story #5

As a user, I would like to be able to create a party

#	Task description	Time	Owner
1	Implement the party setting page	10 hrs	Samrat
2	Implement the ability to insert new party requests into the server	8 hrs	Youngjoon
3	Implement the error handler for failing request	7 hrs	Samrat
4	Testing	2 hrs	Arthur
5	Total	27 hrs	

Acceptance Criteria

→ Given that the party setting page has been set up, the users should be able to create a new party.

→ Given that the party page has been set up, the users should be able to join the room that exists.

→ Given that the communication between the application and the server is connected properly, user should not be joined into more than 1 parties in the same time.

→ Given that the party has ended, the party chat room should be closed within a given time frame.

User Story #6

As a user, I would like to be able to see my friends' location

#	Task description	Time	Owner
1	Implement the UI	8 hrs	SeoHyun
2	Retrieve coordinates data from the server	6 hrs	Youngjoon
3	Implement google map api	6 hrs	Arthur
4	Connecting coordinates with Google Map Api	6 hrs	Samrat
5	Testing	2 hrs	Samrat
	Total	28 hrs	

Acceptance Criteria

→ Given that the coordinate data has been built in the server, the program should retrieve those data successfully.

→ Given that the data has been received from the server, the application should represent the given data properly.

→ Given that the data has been received properly, the user will be able to view information of other users.

→ Given that the data of the users are stored properly, it is the user’s choice whether to share his or her coordinates.

Remaining Backlog

Functional Requirements

- As a user, I would like to be able to reserve a table at the restaurant.
- As a user, I would like to be able to talk in a live chat room so that my questions can be answered quickly.
- As a user, I would like to be able to know the total price of the meal including tax.
- As a user, I would like to be able to make comments on the foods I've eaten.
- As a user, I would like to be able to read reviews for the restaurant.
- As a user, I would like to be able to rate the restaurant.
- As a user, I would like to be able to get notifications about new menu items.
- As a student, I would like to be able to get recommendations for food within various menus.
- As a student, I would like to be able to save my preference for the items I have from the restaurant.
- As a student, I would like to be able to get contact information of the restaurant.
- As a student, I would like to be able to turn on and turn off the tracking service.
- As a student, I would like to be able to get a series of data showing where I dine the most.
- As a student, I would like to be able to add people to my contact by just scanning their unique QR code.
- If time allows, I would like to be able to introduce some secret menu on the restaurant.
- If time allows, I would like to be able to split the bill with friends who go with me.
- If time allows, I would like to be able to get notifications about promotions or events of restaurants.
- If time allows, I would like to be able to have a feature that will prevent me from calling or texting when I'm drunk.
- If time allows, I would like to be able to see other people's most recent activated time so that we can contact them.
- If time allows, I would like to be able to track my monthly spending when I use the app to pay.
- If time allows, I would like to be able to get real-time feedback from people who are in certain restaurants (space, waiting time, etc...).

- If time allows, I would like to be able to get notifications about any coupons.
- I would like to be able to carpool with people who might be going to the same restaurants which are far from campus.
- If time allows, I would like to be able to pay with Apple Pay directly.

Non-Functional Requirements

Architecture and Performance:

We are building an app that is based on an iOS platform by using Swift as the core programming language. We will be using the MVC structure. We expect the time to search for people around you to be finished in 5 to 10 seconds without the user specifying (and if they do, we can set the upper bound to around 30 seconds), and the time for the login to execute should be less than 5 seconds. We also expect the restaurant search to complete in less than 10 seconds. We will be mainly separating our app into two parts, the front-end, and the back-end. For the front-end, we will be using the auto-layout offered by Xcode, and also some of the icons that are free and provided by the various website. As for the back-end, we are planning to use Firebase to create our user database, Yelp's API to gather information about restaurants, and Google Maps' API for the location services.

Security:

Foodie's security is mainly dependent on the login as well as the tracking of a contact's location. We are planning to build a login system which can link accounts by using Facebook or Google accounts. On top of that, we are going to allow users to switch on or switch off their current locations whenever they'd like. If the user switches off his or her current location then the locations of friends or colleagues cannot be seen as well. We will also implement a series of tasks for the user to finish before they can access their phone when they are drunk. We will add the mechanism of checking the identity (such as checking the fingerprint before paying or accessing the app) just in case if someone robbed a phone from our user, with the user already being logged into his or her account.

Usability:

The interface should be easy for users to navigate through applications and be simple enough for the users to understand. As there are various features of our application, it is important to know which tab offers what. We will provide tutorial on how the app functions when opened for the first time to make the app user-friendly.

Hosting/Deployment:

Using firebase as a database, there should be continuous communication between the server. Also, there should be another server for the number of people who went into dining courts.