Team logistics and evaluation

Team name

- 2Cool 2Cook

Team member names and roles

- Ruisi Jian (Expert coder/Full stack)
- Yueting Liao (Team leader)
- Yuchen Wang (Front-end/UI design)
- Yifan Wang (Back-end/Database design)
- Xiaoyu Zhang (Back-end/Search algorithm)

A summary of how you will run each meeting and where minutes will be recorded and stored (No changes from Team Deliverable 1)

- Summarize and evaluate what we did in the last meeting and conclude our inadequacy.
- Each member will be reporting their progress of tasks and potential problems from the last meeting
- Discuss the progress made so far and evaluate for potential improvements.
- Provide some constructive thinking and set a list of tasks for the next meeting.
- Assign appropriate works for each team member.
- Use Google Doc to record the minutes from the meeting (One Doc for all meetings)
- https://docs.google.com/document/d/1mJW8UWMUnrlh8FBEFf_GY9MLaYU6fGM3n-BjViskeRc/edit?usp=sharing (updates every meeting)

Success criteria (which includes a one-sentence description as to why you view each criterion as being important)

1. Attending each meeting on time.
    - Ensure the progress moving forward within the team plan.

2. Finishing the assigned task before the assigned deadline.
    - Ensure that the work schedule is followed.

3. Asking current problems and sharing feedback to team members.
    - Ensure the project is building in maximum efficiency and all team members understand the issue.

4. Planning out the unified process/brief schedule for the project.
    - Ensure team members could keep track of each milestone of the project.

5. Helping teammates on their task if they feel the risk of passing the deadline.
    - Ensure that the team is cohesive and each teammate takes responsibility not only for their separate part but for the whole project.

Project status reports

Write a project status report that summarizes the current state of your project.

The current state of the project:

- All members had a project environment setup

  Since the team decided to develop the website on Node-Red, each team member needed to set up their local Node-Red platform and link it with their local instance of MySQL database. Ruisi helped the team finish the set up during the team meeting by demonstrating the install process through screen sharing on October 5.

- Setup the database schema with the front-end that will be used in the project

  To move forward with the organization of recipe data, the team needed to design a database schema that would fit well with the need of the search engine and front-end display. Yifan is responsible for designing the schema and transferring data from the JSON file to the database.

- Setup the project structure and establishes connections between the main pages

  This process is considered as design in the software development lifecycle. Ruisi, Yuchen and Xiaoyu are responsible for building a general project structure such as pages setup, page redirects, components in the front-end. A general UI design is also determined by them.

- Search function for farmer markets

  The issue of solving this search function is how to use the database in the back-end and display the result to the front-end/web. By writing SQL sentences in Javascript for the search algorithm of farmer markets and creating serval web features for the search page. Yueting is responsible for this page and had these functions done before October 20.

Old risk:

- Risk of dietary allergy
- ~~Risk of putting people in danger if they might forget to turn off the stove when they failed to handle concurrency~~
- Users might upload inappropriate data into the recipe database
- Copyright risks of the recipes

New risk:

- The current data file contains over one million recipes, the database may not be able to handle the large data size

Your project status report also must include a summary of how each team member contributed to the team deliverable 2. This helps ensure that each team member is taking an active role in each aspect of the project.

Xiaoyu Zhang: Responsible for search algorithms and finished basic required code.

Yuchen Wang: Responsible for project structure and connections between the main page.

Yifan Wang: Responsible for databases management and design and searching of databases.

Ruisi Jian: Responsible for setting up the development environment for the team project structure setup on the Node-Red and finding useful data to build the database out of.

Yueting Liao: Responsible for creating the farmers market feature of the project, handling both front-end and back-end of the feature.

Project deliverables

A brief (one paragraph?) description of how your project has changed from the first team deliverable assignment; this should describe any changes in stakeholders, feasibility, scope, project risks

Project changes from the first team deliverable:

- Stakeholders:
    - Removed stakeholder [Chefs who want to learn new recipes]
    - Removed stakeholder [People who have little time to cook]
    - Add stakeholder [Staff who maintains the recipe and account database]

- Feasibility:
    - Removed [Remote Server]
    - Removed [Service may need to be on several platforms]
    - Removed [Work may not spread out evenly throughout the 12 weeks]

- Scope (things we do not plan to implement):
    - Removed [Chatting or comments for recipes on the website]
    - Add [Service may need to be on several platforms]

- Project risk:
    - Add [Current data set may be too large to prototype with]
    - Add [Work may not spread out evenly throughout the 12 weeks]
    - Removed [Risk of putting people in danger if they might forget to turn off the stove when they failed to handle concurrency]

Fully dressed use cases; you should have approximately 80% of your use cases completed for this deliverable

Use Case Title: Search for a Recipe

Primary Actor: User

Stakeholders and Interests: User would like to find a recipe they want to cook

Preconditions: N/A

Main Success Scenario:

1. User navigates to the search bar
2. User types in the keywords of a recipe
3. User get a list of potential recipes through keywords;
   if displaying nothing, return to step 2
4. User chooses one of the recipes from the displayed list
5. User gets a detailed view of the chosen recipe

Extensions:
2a. If the user leaves the search bar fields blank, then show blank.
2b. If the user types in unidentified keywords, then show blank.

Minimal Guarantees: N/A

Success Guarantees: Users able to get all recipes

Use case references by which you reference another use case by underlining the given task: N/A

Use Case Title: User Creates a Login Account

Primary Actor: User

Stakeholders and Interests: User would like to upload their favorite recipes

Preconditions: N/A

Main Success Scenario:

1. User navigates to “create account” button
2. Fill in the personal account information
    a. Username (required)
    b. Date of Birth
    c. Password (required)
    d. Confirmed password (required)
    e. Email address (required)
    f. Allergies
    g. Dietary preferences
3. User automatically logs into the new account

Extensions:
2a. If User leaves a required blank or empty, then shows errors
2b. If the username has already existed, then shows errors
2c. If the confirmed password does not match the password, then shows errors

Minimal Guarantees: N/A

Success Guarantees: Users own a new account

Use case references by which you reference another use case by underlining the given task:
N/A

Use Case Title: User Uploads a New Recipe

Primary Actor: User

Stakeholders and Interests: User uploads a new recipe to the database. Everyone can search and see the new recipe

Preconditions: User must be authenticated and logged in

Main Success Scenario:

4. User clicks “upload a new recipe” button
5. User enters the new recipe name (required)
6. User enters all the necessary ingredients (required)
7. User enters all steps of recipes (required)
8. User uploads an image of the recipe
9. User uploads the new recipe to the database successfully.

Extensions:
2a. If the user leaves a blank, then shows errors
3a. If the user leaves a blank, then shows errors
4a. If the user leaves a blank, then shows errors

Minimal Guarantees: User uploads the recipe to the database

Success Guarantees: User’s recipe is valid after verified. Everyone can see the new recipe

Use case references by which you reference another use case by underlining the given task
Staff verifies that new recipe is valid

Use Case Title: User Creates a Daily Menu

Primary Actor: User

Stakeholders and Interests: User would like to plan out what they cook for meals on a daily basis

Preconditions: User must be authenticated and logged in

Main Success Scenario:

10. User navigates to today's menu
11. User select which meal (breakfast, lunch, dinner) to add
12. User search a recipe for that meal
13. User add recipes to that meal on the daily menu
14. User clicks save button to finish create a menu

Extensions:
3a. If the user couldn't get a recipe from add, user can choose to upload a new recipe

Minimal Guarantees: N/A

Success Guarantees: A new daily menu is created

Use case references by which you reference another use case by underlining the given task
The users search for a recipe
The users upload a new recipe (optional)

Use Case Title: Apply Filters by Allergy

Primary Actor: User

Stakeholders and Interests: User would like eliminate recipes based their dietary allergy

Preconditions: Log into the account; Enter recipe information in the search bar

Main Success Scenario:

15. User login to the account
16. User updates the their dietary allergy in their profile
17. User navigates to the search bar
18. User enters the keywords for a recipe
19. Allergy filter will be applied to the recipes returned
20. Recipes with the keywords will be displayed if it does not contain allergy ingredient

Extensions:
2a. If users already have their dietary allergy entered in their profile, skip this step.
4a. If the user leaves the search bar fields blank, then show blank.
4b. If the user types in unidentified keywords, then show blank.

Minimal Guarantees: Normal search result will be displayed if the filter failed in an unexpected manner

Success Guarantees: A list of recipes with no allergic ingredients will be displayed after the filter.

Use case references by which you reference another use case by underlining the given task
The users create an account and log in
The users search for a recipe

Use Case Title: User feedback

Primary Actor: User

Stakeholders and Interests: User would like to give some feedback on their experience with this website

Preconditions: Log into the account

Main Success Scenario:

1. User login to the account
2. User navigates to the feedback page
3. User check from a list of frequent issues (required)
4. User types in more detailed comments in textbox

Extensions:
3a. If the user did not select any of the check from the list, the submit button is disabled

Minimal Guarantees: N/A

Success Guarantees: A success submission page will be displayed to the user

Use case references by which you reference another use case by underlining the given task
The users create an account and log in

Use Case Title: Favourite Recipes

Primary Actor: User

Stakeholders and Interests: User would like to save recipes that they like for future reference

Preconditions: Log into the account

Main Success Scenario:

1. User login to the account
2. User navigates to the search bar
3. User enters the keywords for a recipe
4. Recipes with the keywords will be displayed
5. User would select the “add to favourite” button

Extensions:
3a. If the user leaves the search bar fields blank, then show blank.
3b. If the user types in unidentified keywords, then show blank.
5a. If the recipe is already added to favourite, click again to remove from favourite

Minimal Guarantees: N/A

Success Guarantees: Recipe saved to their list of favourite recipes

Use case references by which you reference another use case by underlining the given task
The users create an account and log in
The users search for a recipe

User Case: Verify Validity of Uploaded Recipes

Primary Actor: Staff

Stakeholders and Interests: Staff would like to make sure every recipe waiting to be added to the database is feasible

Preconditions: Staff log into the account

Main Success Scenario:

1. Read through the list of recipes uploaded by user that are waiting to be added to database
2. Update the recipe into the database if this recipe looks reasonable
3. Send notification to the users that their recipe successfully went through

Extensions:
2a. If the recipe looks bad, staff would notify users that their recipe is invalid

Minimal Guarantees: N/A

Success Guarantees: The recipe will be either passed verification or not.

Use case references by which you reference another use case by underlining the given task
N/A

User Case: Rate the Recipe

Primary Actor: User

Stakeholders and Interests: User would like to rate each recipe

Preconditions: Log into the account

Main Success Scenario:

1. User login to the account
2. User navigates to the search bar
3. User enters the keywords for a recipe
4. User navigates to the recipe
5. User click rate the recipe button and give a score

Extensions:
3a. If the user leaves the search bar fields blank, then show blank.
3b. If the user types in unidentified keywords, then show blank.
5a. If user has already rated, rating button disabled

Minimal Guarantees: N/A

Success Guarantees: Recipe score is updated with the rating input

Use case references by which you reference another use case by underlining the given task
The users create an account and log in
The users search for a recipe

User Case: Sort the Search Result by Rating

Primary Actor: User

Stakeholders and Interests: User would like to see recipes with higher rating up front

Preconditions:

Main Success Scenario:

1. User navigates to the search bar
2. User enters the keywords for a recipe
3. Recipes with the keywords will be displayed in order of sorting

Extensions:
2a. If the user leaves the search bar fields blank, then show blank.
2b. If the user types in unidentified keywords, then show blank.

Minimal Guarantees: N/A

Success Guarantees: Recipes will automatically be ordered based on the rating from highest to lowest

Use case references by which you reference another use case by underlining the given task
The users search for a recipe
The users rate the recipe

User Case: Ranking List for Top 10 Recipes for Each Week

Primary Actor: User

Stakeholders and Interests: User would like to see the best rating recipes

Preconditions: N/A

Main Success Scenario:

1. User navigates to the ranking page
2. Ranking is updated based on current recipe rating at the beginning of each week

Extensions: N/A

Minimal Guarantees: N/A

Success Guarantees: Display top 10 recipes weekly.

Use case references by which you reference another use case by underlining the given task
The users rate the recipe

User Case: Search for Nearby Farmers Market for to buy Ingredients

Primary Actor: User

Stakeholders and Interests: User would like to see nearby farmer market to buy ingredients

Preconditions: Log into the account

Main Success Scenario:

1. User login to the account
2. User navigates to search farmers market page
3. User enters information in the text box
4. The farmers market result will be displayed

Extensions:
3a. If the user leaves all text box fields blank, then show all.
3b. If the user types in unidentified keywords, then show blank.

Minimal Guarantees: N/A

Success Guarantees: All the farmers market will be displayed depend on user input

Use case references by which you reference another use case by underlining the given task
The users create an account and log in

ADD TWO NEW USES CASES

Use Case Title: Set timer for certain steps that require timing

Primary Actor: User

Stakeholders and Interests: User would like to time for certain steps according to the recipe instruction to improve the quality/taste of the recipe

Preconditions: User must be authenticated and logged in, user must add the recipe into his/her menu

Main Success Scenario:

1. User navigates to “record time” button
2. Fill in the value in the timer bar
3. User clicks the start time button
4. A web label will be displayed to notify the user once the time is up

Extensions:
2a. If User leaves a required blank or empty, then start button is disabled

Minimal Guarantees: A timer is set up

Success Guarantees: User correctly times the step according to the instruction

Use case references by which you reference another use case by underlining the given task:
User creates a daily menu
User creates a login account

Use Case Title: Deactivate user account

Primary Actor: Staff

Stakeholders and Interests: Users' accounts will be banned if they violate user agreement

Preconditions: Staff login as admin

Main Success Scenario:

1. Staff login as admin
2. Navigate to list of all users
3. Deactivate the selected user
4. The user is removed from the database

Extensions:
1a. If staff enters wrong authentication information, then shows errors
3b. If staff clicks "Cancel" button to the confirmation pop up, no action will take place

Minimal Guarantees: N/A

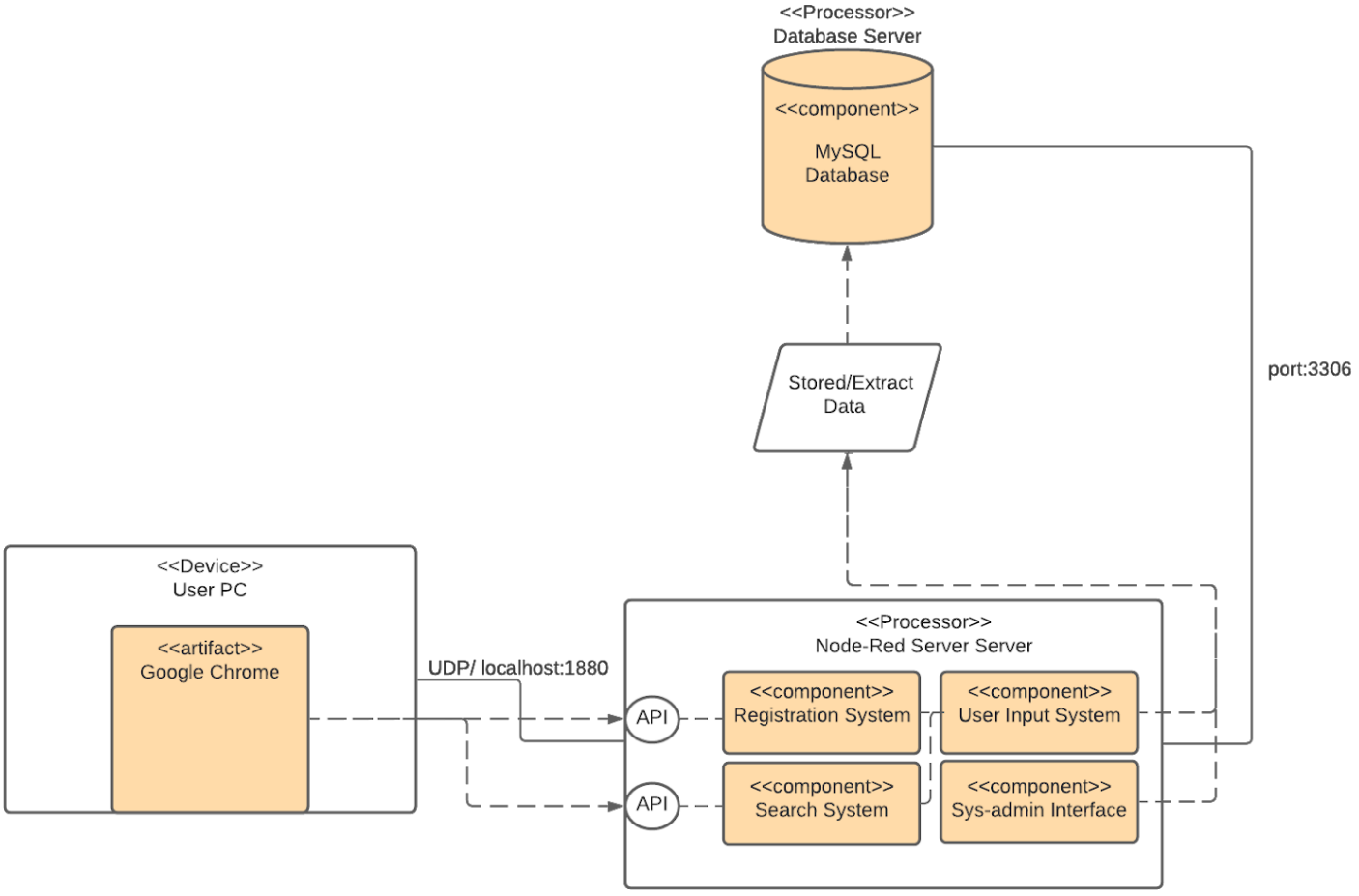
Success Guarantees: Deactivated user account cannot be logged in

Use case references by which you reference another use case by underlining the given task:
N/A

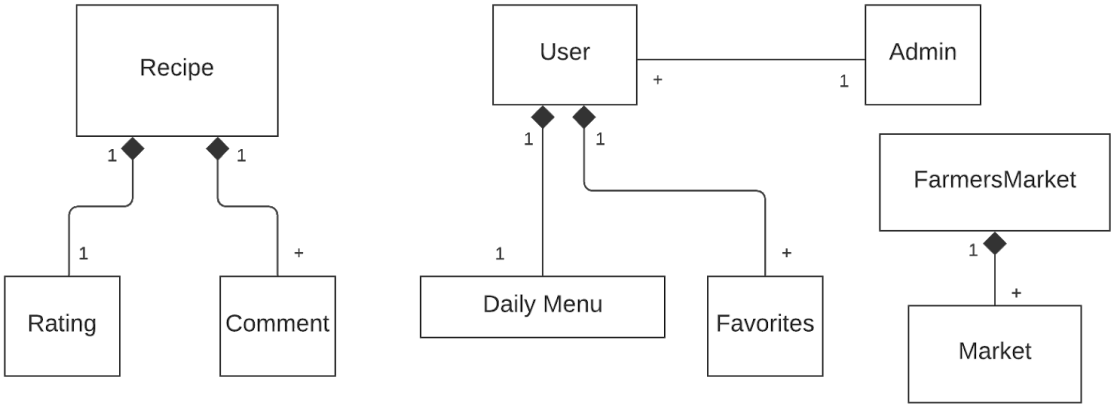
Table of requirements; as with Mini-Assignment 2, you should include a well-organized spreadsheet of requirements; each requirement must be uniquely numbered and include the five columns shown in the lecture

ID	Requirement	Priority	Stakeholder	Feasibility
1	Search for a recipe	high	Cooking novice; Indecisive users	Phase 1
2	Smoothly navigate through different pages with a logical flow	high	All users	Phase 1
3	Search for a farmers market	medium	All users	Phase 2
4	Comment on the recipes	low	All users	Phase 3
5	Rating for the recipes	low	All users	Phase 3
6	Verifying the recipes	low	Staff	Phase 3
7	Set timer for certain steps that require timing	low	Cooking novice	Phase 3
8	Update Rating weekly	low	All users	Phase 3
9	Add dietary preferences	medium	All users	Phase 2
10	Drop dietary preferences	medium	All users	Phase 2
11	Add allergy to user profile	medium	All users	Phase 2
12	Set up a daily menu	high	Cooking novice	Phase 1
13	User login account created	high	All users	Phase 1
14	User feedback	low	All users	Phase 3
15	User account deactivate	low	Staff	Phase 3
16	Add a favourite recipe to user profile	medium	All users	Phase 2
17	Remove favourite recipe from the user profile	medium	All users	Phase 2
18	Displays weekly top 10 recipes	medium	All users	Phase 2
19	Sort the recipe searching result by recipe rating	medium	All users	Phase 2
20	Check off steps on the procedure when completed	low	Cooking novice	Phase 3

Deployment diagram; using proper UML notation, provide a diagram that shows the hardware and software platform(s) used to deploy your software



Domain model diagram; using proper UML notation. UML relationship/class



Recipe
- id : String - title : String - rating : Rate - comment : Comment - recipeInfo:ArrayList<String> = new ArrayList<String>(); - instructions:ArrayList<String> = new ArrayList<String>(); - ingredients:ArrayList<String> = new ArrayList<String>();
+ getRecipeInfo(id : String):ArrayList<String>(); + getInstructions(id: String):ArrayList<String>() + getIngredients(id: String):ArrayList<String>() + getRating(id: String):Rate + getComment(id: String):Comment

Rating
- Rate : Float; - id : String; - totalRating : Int; - totalRateNum : Float;
+ getRating(id : String) : Float; + giveRate(id : String, Int);

Admin
+ allusers : ArrayList<User>; + allrecipes : ArrayList<recipes>; + recipes_under_review: ArrayList<recipes>;
+ deactivate(username) + approve(Recipe) + disapprove(Recipe)

Comment
- comments : ArrayList<String> - id : String
+ getComments(id : String) : ArrayList(String); + giveComments(id : String, String);

Farmers Market
searchKey: String + marketList ArrayList<Market> = new ArrayList<String>();
+ searchMarket(searchKey : String):Market

User
+ dietary preferences : new ArrayList<String>(); + dietary allergy : new ArrayList<String>(); + Favorites + Daily Menu
+ addMenu(recipeid) + removeMenu(recipeid) + addFavorites(recipeid) + removeFavorites(recipeid) + addPreference(string) + removePreference(string) + addAllergy(string) + removeAllergy(string) + comment(recipeid) + rate(recipeid) + upload(Recipe) + deactivate()

Favorites
+ RecipeList : ArrayList<Recipes> = new ArrayList<Recipes>();
+ show(RecipeList):ArrayList<Recipes> + getRecipe(recipeid):Recipe + add (recipeid) +del (recipeid)

Market
+ FMID:int + marketName:String - website:String - street: String + county:String + city:String + state:String
+ getWebsite(searchKey: String):String + getStreet(searchKey: String):String

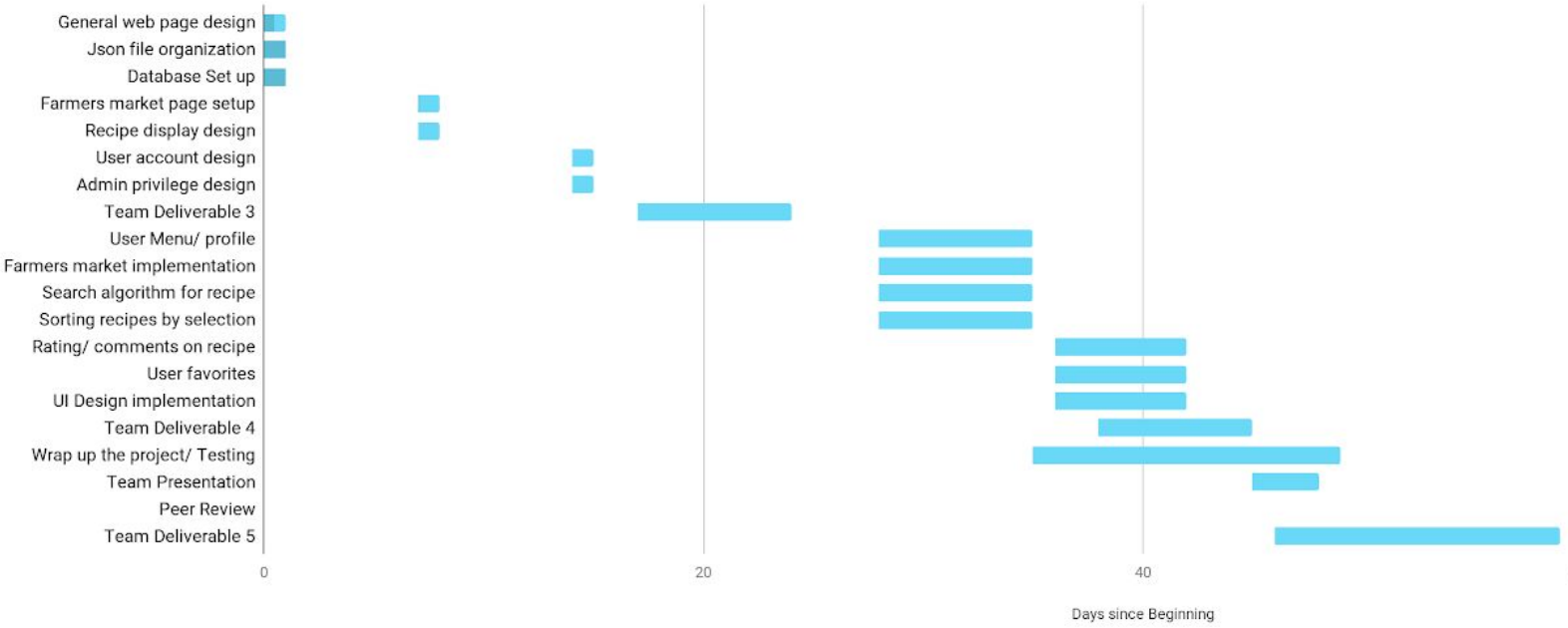
Daily Menu
+ RecipeList : ArrayList<Recipes> = new ArrayList<Recipes>(); + Past7D_menu : ArrayList<ArrayList<Recipes>> = new ArrayList<ArrayList<Recipes>>();
+ show(RecipeList):ArrayList<Recipes> + add (recipeid) +del (recipeid) + calAggregatedNutrition(RecipeList) + listInstructions(RecipeList) + listIngredients(RecipeList) + showPast7D(Past7D_menu)

Project schedule and work breakdown structure (WBS); document a projected schedule of all tasks associated with your project and who will do what; note that the 11/3 team deliverable focuses on design, whereas the 11/24 team deliverable focuses on implementation

TASK NAME	START DATE	DAYS since Beginning*	END DATE	DURATION* (WORK DAYS)	DAYS COMPLETE*	DAYS REMAINING*	TEAM MEMBER	PERCENT COMPLETE
General web page design	**10/17**	0	10/18	1	0.5	0.5		50%
Json file organization	**10/17**	0	10/18	1	1	0		100%
Database Set up	**10/17**	0	10/18	1	1	0		100%
Farmers market page setup	**10/24**	7	10/25	1	0.1	0.9		10%
Recipe display design	**10/24**	7	10/25	1	0	1		0%
User account design	**10/31**	14	11/1	1	0	1		0%
Admin privilege design	**10/31**	14	11/1	1	0	1		0%
Team Deliverable 3	**11/3**	17	11/10	7	0	7		0%
User Menu/ profile	**11/14**	28	11/21	7	0	7		0%
Farmers market implementation	**11/14**	28	11/21	7	0	7		0%
Search algorithm for recipe	**11/14**	28	11/21	7	0	7		0%
Sorting recipes by selection	**11/14**	28	11/21	7	0	7		0%
Rating/ comments on recipe	**11/22**	36	11/28	6	0	6		0%
User favorites	**11/22**	36	11/28	6	0	6		0%
UI Design implementation	**11/22**	36	11/28	6	0	6		0%
Team Deliverable 4	**11/24**	38	12/1	7	0	7		0%
Wrap up the project/ Testing	**11/21**	35	12/5	14	0	14		0%
Team Presentation	**12/1**	45	12/4	3	0	3		0%
Peer Review	**12/10**	54	12/10	0	0	0		0%
Team Deliverable 5	**12/2**	46	12/15	13	0	13		0%

Gantt Chart See Next Page

Gantt Chart


General web page design
Json file organization
Database Set up
Farmers market page setup
Recipe display design
User account design
Admin privilege design
Team Deliverable 3
User Menu/ profile
Farmers market implementation
Search algorithm for recipe
Sorting recipes by selection
Rating/ comments on recipe
User favorites
UI Design implementation
Team Deliverable 4
Wrap up the project/ Testing
Team Presentation
Peer Review
Team Deliverable 5
0
20
40
Days since Beginning