Computer Science and Engineering

Eco-Foodie

System Requirements (and Analysis) Specification (SRS)

Version 1.0

Document Number: SRS-001

Project Team Number: B23

Project Team Members: Nathan Cantu snc387

Vijay Kallem vrk248

Jonathan Xu jx990

REVIEW AND APPROVALS

Printed Name and Title	Function (Author, Reviewer, Approval)	Date	Signature
Nathan Cantu	Author, Reviewer, Approval	09/26/2022	Nathan Cantu
Vijay Kallem	Author, Reviewer, Approval	09/26/2022	Vijay Kallem
Jonathan Xu	Author, Reviewer, Approval	09/26/2022	Jonathan Xu

REVISION LEVEL

Date	Revision Number	Purpose
September 26, 2022	Version 1.0	Initial Release

TABLE OF CONTENTS

1. INTRODUCTION

Purpose

The purpose of the Software Requirements Specification is to specify the requirements and analysis for Eco-Foodie that will meet the expectations of the client. The intended audience of this document is the client, the software quality group, and other team members assigned to this project.

2. SCOPE

EcoFoodie will originally be limited to the confines of New York City in order to perform hands-on practical testing and to determine potential implementation issues. Those within NYC can share their prepared food offerings and those within NYC can accept the meals. Once a supplier has uploaded an offering, every user within NYC will be notified. If anyone is interested, they can accept the proposal using the app and will have a designated amount of time to go to the location of the supplier and pick the food up. It must be picked up within this amount of time or else the offering will be made available on the app again. Location sharing must be enacted by a user if that user wants to be notified of new local food offerings being posted. All users for the initial testing phase will be within NYC. A supplier may use Eco-Foodie to upload meal offerings accompanied by pictures of the food, a video of the preparation of the food, and a description of ingredients and allergy information. Consumers may use it to reserve food offerings, leave reviews, add other users as friends and create groups, and file complaints against suppliers. Users can send out directed alerts to groups and avoid encountering strangers. All users and suppliers have access to a terms of conduct page dictating the rules that every supplier must follow in the preparation of and distribution of their food.

2.1 Identification

EcoFoodie, SRS-001, Revision 1.0

2.2 Bounds

For the initial release, only the city of New York will be supported. For the initial stage, every user and every supplier must be within the city of New York. For the time being, we will only be developing a mobile application, meaning no website. The service requires a phone running IOS or Android and location sharing to use. Every supplier must have a valid SSN in order to register. Due to the nature of a time limit, not every offering will be available to every user if the location distance is too far to meet the timeline.

2.3 Objectives

The priority of this service is to allow users to share their leftover produce, leftover unopened food items, leftover ingredients, and homemade meals with their community and neighborhood. Every offering by a supplier (a user who has registered their SSN) is free of cost to every other user. The goal is to help mitigate food insecurity across the city of New York, broadening wider as the service progresses, for families or individuals who are low income but do have an internet capable device. This service is high priority and will be delivered directly to the client. The service follows an agile incremental model.

Future objectives will be to broaden our service to those who are low income and do not have access to an internet capable device. Current workshop ideas are screens positioned around the city that anyone can use to order the offering followed by a transportation service to/from that location for drop off of the offering. Ways to pay for those expenses must be workshopped heavily.

Deliverable Dates:

Software Analysis/Requirements Specification (SRS) : 9/26/22

Software Project Management Plan (SPMP) : 10/3/2022

Project Description : 10/11/2022

Initial Software Design Description : 10/17/22

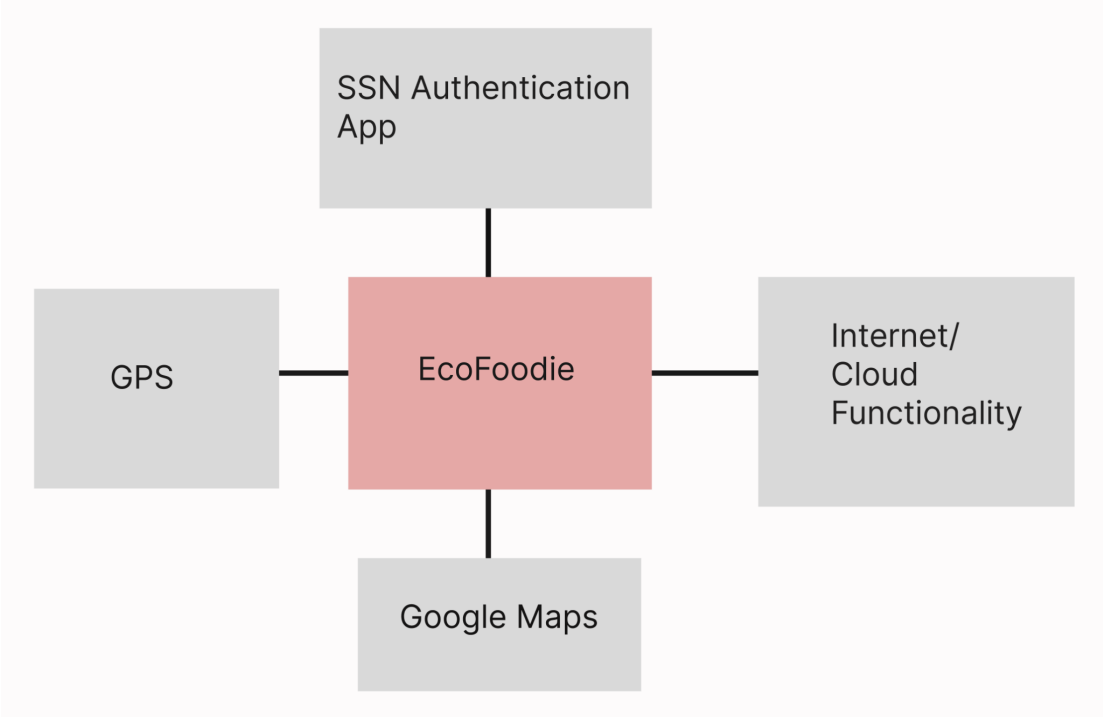
Final Design Document : 11/28/22 -12/14/2022

Implementation: 11/28/22 -12/14/2022

Presentation and Demonstration: 11/28/22 -12/14/2022

3. OVERALL SYSTEM OVERVIEW

3.1 Context Diagram



3.2 Additional Descriptive Items

The product functions of the application include the ability of a supplier to post a new listing along with video and pictures, and a notifier that rings everyone within a certain radius of the supplier when a supplier submits a post. This will be the main function of the application as it is how users will be able to see as well as receive their food. Another important function would be the account creation and login, since it is important that the users are all verified to prevent fraud or other misdoings.

General characteristics of the intended users can vary widely. Any user using the app strictly for obtaining food offerings requires only knowledge and experience of intercity travel and following directions. Users require basic technical expertise on how to interface with a mobile device. Any supplier using the app should know how to record a video, how to take pictures, and how to attach said files to posts. Any supplier preparing food should have basic knowledge of food preparation and sanitation.

Constraints on the system include safety issues especially with contaminated food. The video system requirement, the picture requirement, registering the suppliers SSN, and giving the users a shortcut to a complaint box will help ensure that the supplier is held liable but does not prevent the incident from occurring in the first place. Security is also a risk factor since you have strangers meeting up at a potentially dangerous location.

We will work under the assumption that the software that we will use remains compatible with each other and that any changes to the Android or IOS operating system will not inhibit the application. If we are not able to develop for either platform, then we would try creating a web application.

A mobile device running Android or IOS and having the GPS permission on is required. For the supplier, the mobile device also has to be capable of taking and recording video.

4. DOCUMENT OVERVIEW

The rest of the SRS contains:

- The Reference Documents: project documents already completed.
- The Business Requirements: any Business Drivers of the System and any Business Risks that the System might run into
- The Functional Requirements: requirements about what the system components should provide
- The NonFunctional Requirements: requirements that impact the system as a whole including constraints
- The Analysis: consists of diagrams such as Class, Event, and Sequence
- System Plan Requirements: planned ways to test the system after the system has been implemented
- Qualification Provisions: how this document is reviewed for quality
- The Requirements Traceability: demonstrates how each requirement is connected to another
- The Document Evolution: expectations about changes that could be made during implementation
- The personal schedule and defect tracking charts: tracking each team member’s estimated versus actual time and defect statistics
- The Dictionary: showcases all of the Classes, Members, and Attributes of the System.

5. REFERENCE DOCUMENTS

B23, Project Proposal, Version 1.0, 9/22/2022

6. BUSINESS REQUIREMENTS

6.1 Technology

A separate mobile interface will need to be built to allow the user to post content, and to show content posted by other users in their neighborhood using GPS.

6.2 Economics

The app will be free for users to download and use. Per our release, we will not have any monetization options available due to time constraints. Later, the app will be monetized using advertisements. Ads will be personalized for each user. The GPS feature also will allow us to show ads to a user related to their neighborhood, town or city.

6.3 Regulatory and Legal

The developers must be covered for any liability caused by malicious practices done through the app. Thus an extensive terms and conditions contract is required. Any laws regarding the handling of citizen's SSN will also be followed.

6.4 Market Considerations

The product has potential to scale up to a large number of users. Every household is our target customer.

6.5 Risks and Alternatives

Type of Risk	Description	Probability	How discovered	Responsible Party	Status	Mitigation Plan
Business	Advertisements may not be enough to cover the costs. Especially if the project evolves into having screens placed around the city, those costs plus transportation costs for drivers to head to those locations may be too expensive	Very likely	If the total cost starts exceeding the total profits	Product team	Not currently a threat	Introduce more methods of monetization, currently undetermined
Operational	Users can become ill, sick, or fatally harmed due to poisoned or contaminated dishes offered up on the app	Extremely likely	If someone complains on the app or directly to the team or if law enforcement notifies the team	Product Team and the perpetrator	Major current problem	Uploading the video required of them preparing the food alongside pictures of the food, the complaint feature, and suppliers having to register their SSN, should make it easier to hold the supplier liable. This accountability will hopefully prevent suppliers from acting out.
Technology	One risk is if someone hacked into the system and sent out false alerts to everyone in the area. They would not be registered with a SSN and they could use this feature for extreme harm towards the users.	Possible	Law enforcement notification, seeing it through our servers	Product team and hacker	Always very importa nt	The mitigation plan is to implement security cryptographic measures to ensure that nobody can reasonably hack into the system
Economic	Due to the operational risk of one of the users being injured by the contaminated food or by technology risk of being lured by ill intentioned people, the service could be sued.	Likely	Notification by an attorney that the company is being sued	The product team but the prosecution is the one suing	Of concern currently	Have users agree to a terms and service contract that does not place liability on the service

Human Resources and Training

As there are only three team members, the human resources requirement is minimal. Primarily an overview of expected respect, and proper gender nouns.

7. SPECIFIC REQUIREMENTS (DESCRIPTIVE FUNCTIONAL REQUIREMENTS)

7.1 Functional Descriptive Detailed Requirements

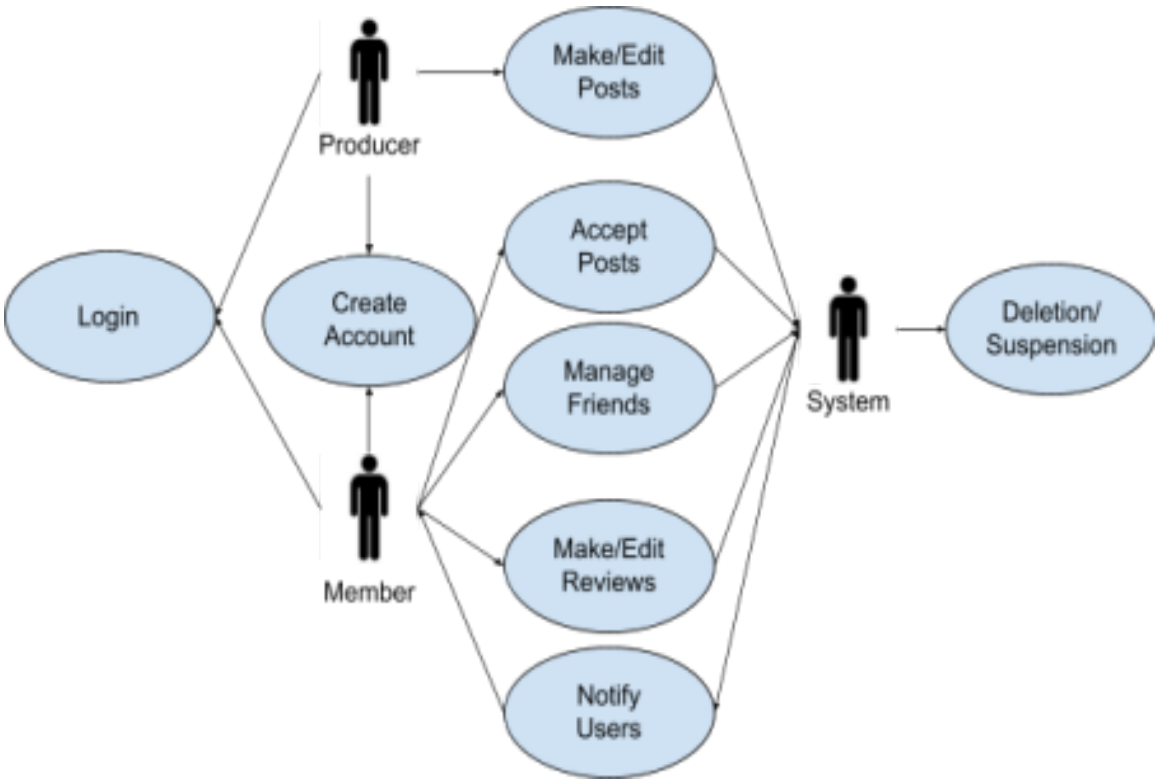
1. The system shall send out local notifications to devices within NYC boundaries
2. The system shall allow for access to the GPS location of a users device in order to send out local alerts to every user in the area (currently within NYC is the locality for all posts)
3. The system shall allow users to set up profiles with their name and any additional information they want to include in a text box
4. The system shall allow users to send friend requests and accept friend requests from other users
5. The system shall allow users to create multiple groups consisting of some or all of the user's friends
6. The system shall allow users the option on each post before publication to only notify selected groups or group
7. The system shall only allow users to be able to publish posts if they are registered as a producer and their SSN has been registered and verified
8. The system shall interact with the Social Security Number Verification Service (SSNVS) in order to check a users SSN verification
9. The system shall check that photos, an address, a video, a recipe, and allergy information are all included on each individual post before it is allowed to be published
10. The system shall have interactions with the devices camera system in order to upload pictures or videos and a timer system to track the listing take down reupload schedule
11. The system shall have reserved storage for file uploads
12. The system shall only allow one user to reserve a listing at a time
    a. The system shall take down the listing temporarily and reupload the listing after a designated amount of time if the producer has not marked it as complete

    b. The system shall remove the user who had reserved the order from the reservation list upon timer completion

13. The system shall allow producers to delete or edit their listings after publication
14. The system shall allow users to leave public reviews available to every user on a producer's account with a star rating system out of 5 and the ability for the user to publish text along with the star rating
15. The system shall include a complaint box page available to all users
    a. The page is a text box where you can send an email to the customer support team
    b. Other text boxes include the account the user wants to complain about and the order or orders they wish to complain about
16. The system shall direct users who leave a 1 star review to the complaint box page
17. The system shall allow for a post to be deleted by and for an account to be suspended or permanently deleted by the developers
18. The system shall include a terms of service page available to every user detailing the rules, regulations and conduct standards that every producer must abide by

7.2 Requirement Use Cases

7.2.1 Use Case Diagrams



7.2.2 Use Case Descriptions

Log in		
Description	The user/producer puts in their username and password in order to access their account.	
Pre- Conditions	Must already have an account.	
Flows	Basic or Normal Flows	1. User inputs account username/password 2. Profile loads up

	Alternative Flows	1. Incorrect Information is put in 2. User is prompted to retry 3. Correct info is finally entered
Post Conditions	Can now view profile and see if any posts are out	
Special Requirements	Must access database to confirm account information	
Extension Points	Users are taken to a main page where they can view posts, manage friends, etc	

Create Account		
Description	User fills in their personal information and username/password and chooses whether they are making a producer/member account. Producers have to put in extra information to ensure that they are trustworthy and don't have ill intentions.	
Pre- Conditions	Must have access to the application	
Flows	Basic or Normal Flows	1. User chooses member account 2. User fills out member information 3. Information is saved to database
	Alternative Flows	1. User chooses producer account 2. User fills out producer information 3. Information is saved to database
Post Conditions	If a producer account is created, they will have some extra privileges. Can now sign in.	
Special Requirements	Checks with Social Security Number Verification Service when creating a producer account.	
Extension Points	Now will be sent to the login page	

Make/Edit Posts		
Description	User makes a post about leftovers that they have ready, taking pictures and putting down information on the food that has been prepared	
Pre- Conditions	User must be in producer account and have valid information	
Flows	Basic or Normal Flows	1. User prepares the food 2. User takes picture of food 3. User fills out information on the post
	Alternative Flows	1. User makes mistake in post 2. User edits post 3. User re uploads post
Post Conditions	Member users can now see the post and react to it	
Special Requirements	System ensures that all requirements of the posts are met. GPS location is tracked to see where the close members are	
Extension Points	The post then goes out to the system which activates the notification	

Accept Posts		
Description	User can accept the post if they decide to take the leftovers from the producer/vendor	
Pre- Conditions	Must have member account	
Flows	Basic or Normal Flows	1. User looking through posts 2. User sees a post that catches his eye 3. User hits accept and gets their food
	Alternative Flows	1. User accepts a post and changes their mind 2. User revokes the claim 3. Post goes back out into the system
Post Conditions	Post disappears so other members can’t accept it	

Special Requirements	Activates GPS to see where the food is located
Extension Points	Post is then removed from the system and later user is prompted to make a review

Manage Friends		
Description	User can add other users as friends as well as delete and can create group chats with them	
Pre- Conditions	Must have account already	
Flows	Basic or Normal Flows	1. User looks up another user's username 2. Checks to confirm profile is the one they want to add 3. Add person as friend and they can decide whether or not to accept
	Alternative Flows	1. Can highlight various friends from friends list 2. Pick the group option 3. New group is created with all the highlighted friends
Post Conditions	Either have a new friend, new friend group, or less friends.	
Special Requirements	Database must be accessed and new friends list/group must be updated	
Extension Points	Updated friends list is shown in the system and the user has the option to make changes again	

Make/Edit Reviews		
Description	After accepting a post from any producer, the user has the option to either write a good or bad review describing the experience as well as the food quality	
Pre- Conditions	Must be a member and have accepted a post	
Flows	Basic or Normal Flows	1. User finishes food and enjoys it 2. User writes a good review for the producer 3. Future users get to see the review and producer has a higher chance of getting posts accepted
	Alternative Flows	1. User finishes food and dislikes it 2. User write a very negative review for the producer 3. If review was bad they are also given the option to file a complaint
Post Conditions	Producer's profile will be updated to show the average rating since their last post	
Special Requirements	If a complaint is filed, the contents of the complaint get emailed to a support team that will handle their case	
Extension Points	After making the review, get sent back to the main menu where you have the option to accept posts, manage friends, etc.	

Notify Users		
Description	Once a post is made, the system immediately sends out a notification to all users in the area	
Pre- Conditions	A post had to have been made in order for the notification to appear	
Flows	Basic or Normal Flows	1. Producer makes a post 2. System notices and checks the gps location 3. All member users in the general location get notified
	Alternative Flows	1. Producer makes a post, but it was canceled 2. System notifies, but then the notification says canceled 3. Users see that it's canceled and look for other posts

Post Conditions	Notification disappears quickly as well as the post if it was canceled. Otherwise, notification appears to everyone in the area
Special Requirements	All members in the area are affected
Extension Points	New post appears on the main page when application opens

Deletion/Suspension		
Description	Once a post is left untouched for long enough, it will get automatically deleted by the system. The same goes for accounts, except they first get suspended and would require the user to put in their information to ensure nothing changed	
Pre- Conditions	Must have account/post that has been out for too long (1 year for account, 1 day for post)	
Flows	Basic or Normal Flows	1. Post left unaccepted for 1 day 2. System automatically deletes the post 3. Post is no longer shown on users' home pages
	Alternative Flows	1. Account not logged into for at least a year 2. System automatically suspends it 3. User no longer shown on friends lists or has privileges
Post Conditions	Post will no longer be shown and account will no longer be useable until user verifies all the information on the profile if the user ever comes back	
Special Requirements	If need be, the post/account could also be deleted directly by developers	
Extension Points	Once the account is suspended (from inactivity), user is taken back to sign in page to verify information if they come back to use the application	

8. NON-FUNCTIONAL DESCRIPTIVE DETAILED REQUIREMENTS

Product requirements:

1. The system shall use an end to end encrypted method to receive a users SSN and send to a SSN verification API to avoid exposing a user's SSN
2. The system shall use an end to end encrypted method to receive and track a user's GPS location to avoid location leakage

Organizational requirements:

1. If a user is found to break the terms of service and their action resulted in the physical or mental harm of a user, then that user and their SSN will be reported to the authorities
2. Users of the application shall authenticate themselves using their user email and password
3. Those who are part of the development team or associated teams of the application shall authenticate themselves using their employee work email and password

External requirements:

1. The system shall implement and follow data privacy protections as determined under U.S. regulations

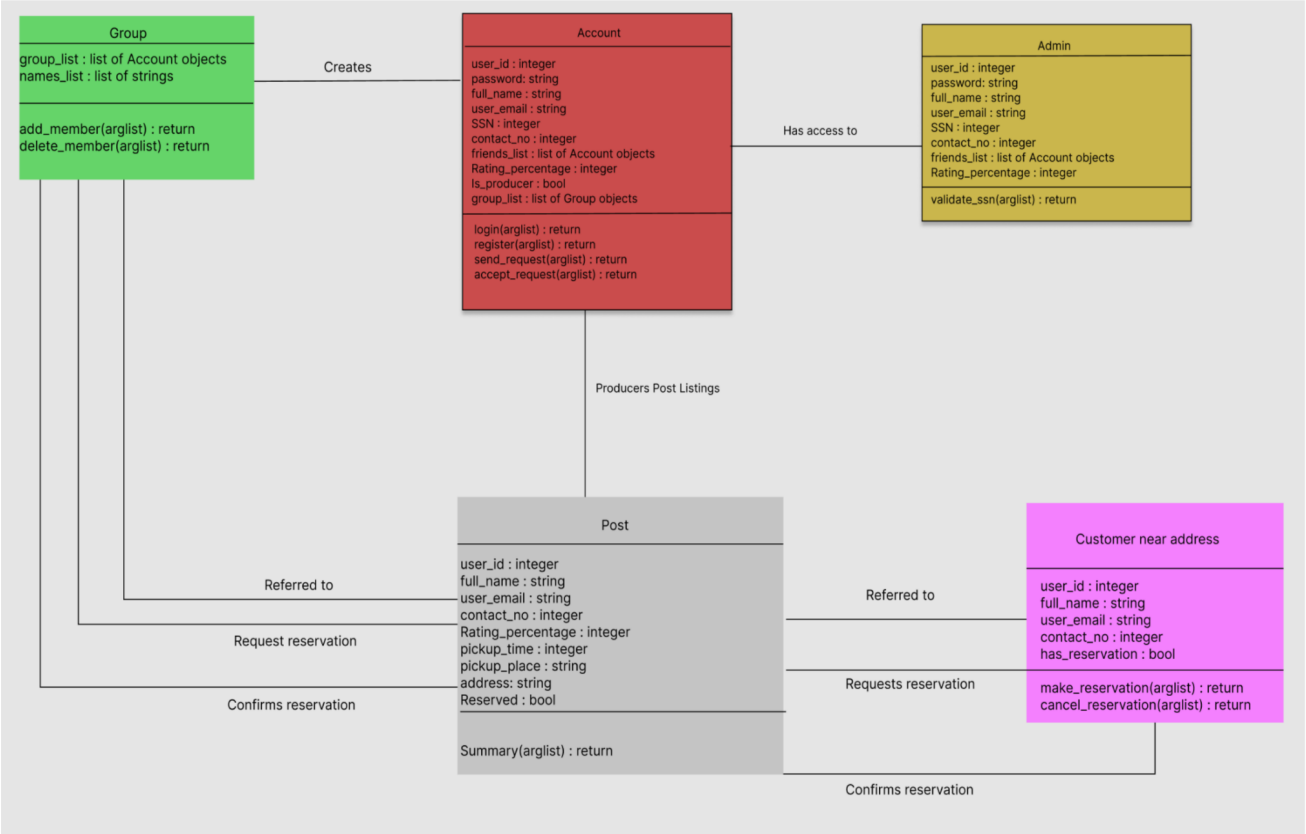
9. ANALYSIS

9.1 Component (Component/Package/Subsystem) Architecture

- Classes: Account, Group, Admin, Post, Customer Near Address
- Account object creates Group object

- Admin object can delete/suspend Account object
- Account object can create a Post object
- Post object can send notifications to Account members of the Group object
- Post object can send notifications to Account members of the Customer near address object

9.2 Component Architecture Diagram



9.3 Component Descriptions

Account: The account stores all the data of a user. It requires login credentials to access that data. The data allows users to accept and send connection requests. It maintains the list of groups that the user is active in. It stores and gives the user access to all the personal data and account data.

Group: The group has a functionality to cluster together a group of users to make sharing the post easy. The user could create a group and add people to it so the user can send posts exclusively to intended people easily.

Admin: Admin is like a motherboard of the application. It maintains the details of the user and Admin gives the access to everything for a user to do on the app.

Post : The post's functionalities lets users take a picture of the food, add a description to it and send to other users.

Customers nearby : It retrieves and gives the data of all the users nearby the user who is posting a post using GPS technology which locates the user.

9.4 Class Diagrams

9.4.1 Individual Class Diagrams

9.4.2 Class Relationship/Interaction Diagrams

See Section 9.2

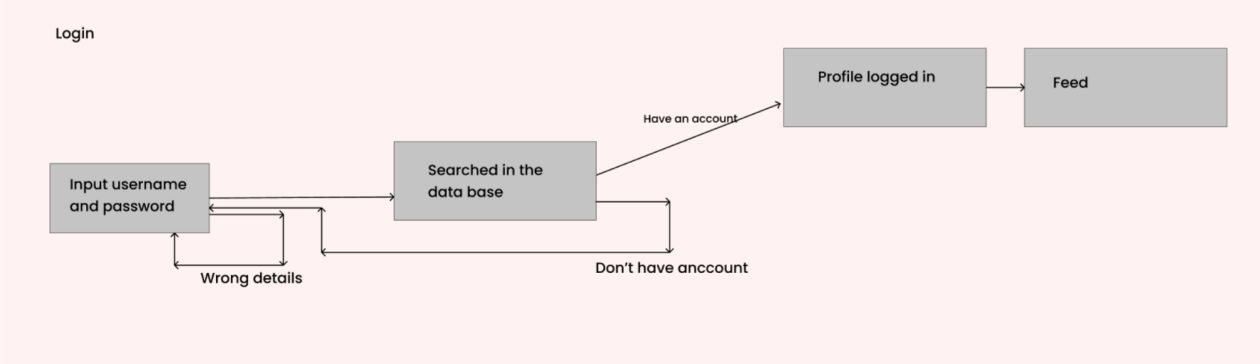
9.5 Events

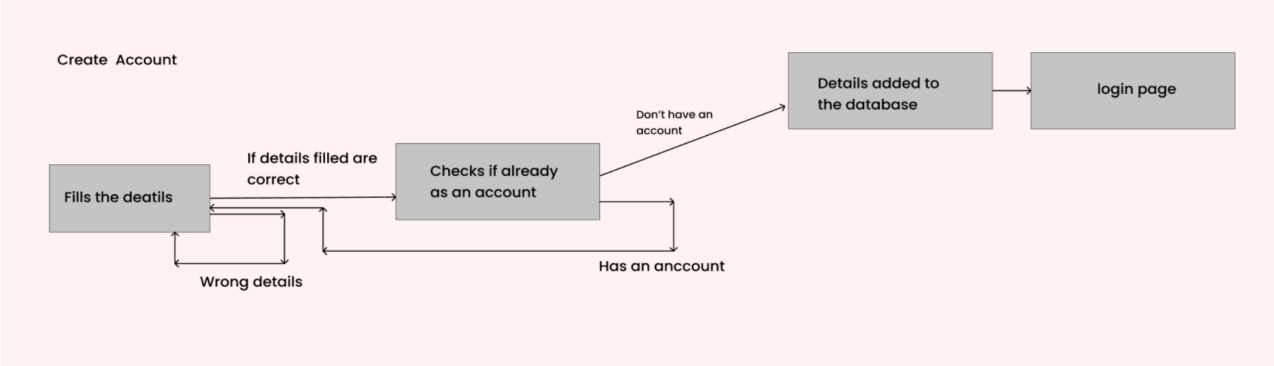
Login, Creating an account, Notifying a user, making/editing a review, adding/removing friends, creating friend groups, making/editing posts, suspending/deleting a post or an account

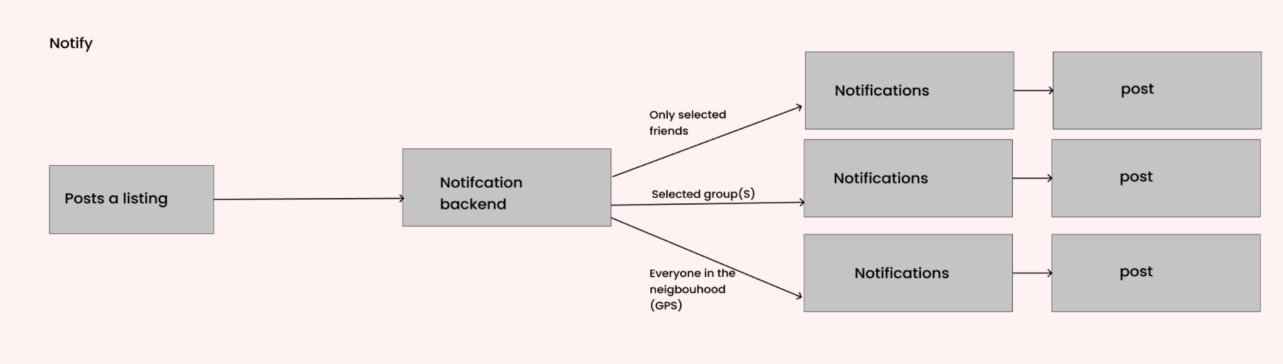
9.5.1 Motives

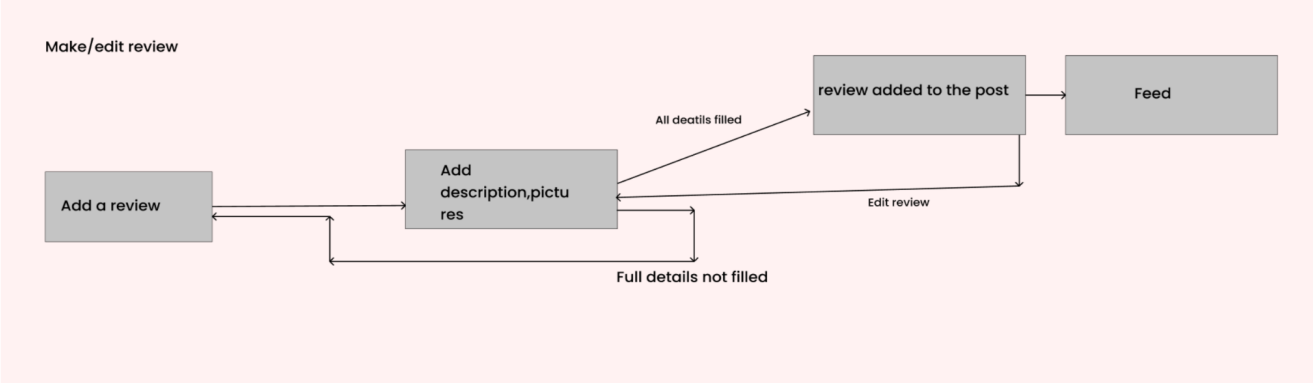
- Creating an account: User must make an account to save GPS location, profile details, and credit information and be authorized to post listings after their SSN is verified
- Login: User must log in to access saved personal data, access saved account profile, and be able to post listings/pick up listings
- Notifying a user: Users must be notified of new listings in their area in order for them to be able to reserve the listing
- Making/editing a review: Review structure must be implemented in order to dictate harmful suppliers and/or bad food conditions
- Adding/removing friends: Friends can be added/removed in order to create groups
- Creating friend groups: This allows users to send specific listing notifications to a group list, avoiding strangers
- Making/editing posts: This is needed so suppliers can upload and send notifications to others of their new food offering
- Suspending/deleting a post or an account: Needed to ensure that harmful accounts are deleted from the account and needed to ensure listings do not remain in effect past an expiration time to ensure safety

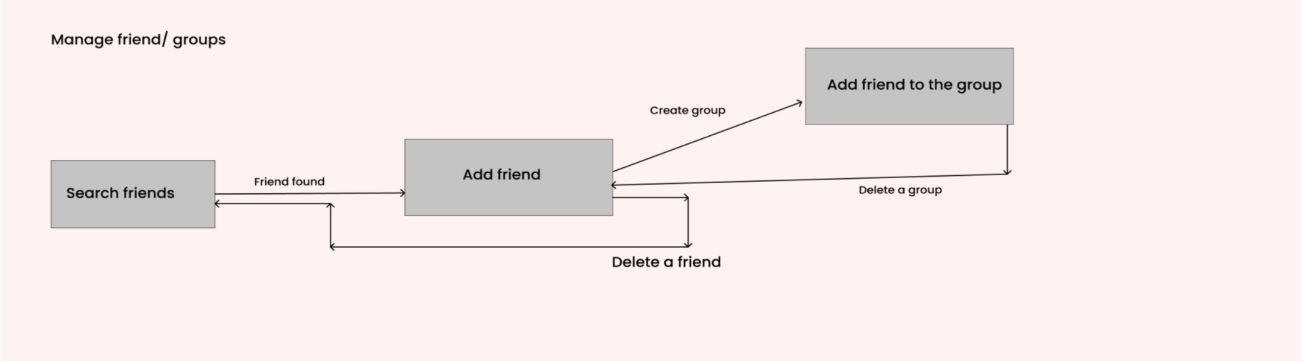
9.5.2 Event Diagrams

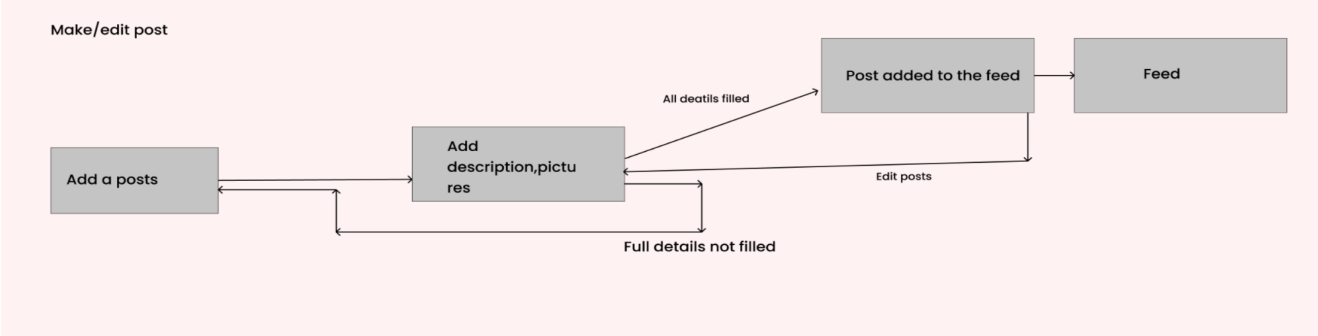



Create Account
Fills the deatils
If details filled are correct
Checks if already as an account
Don't have an account
Details added to the database
login page
Wrong details
Has an anccount

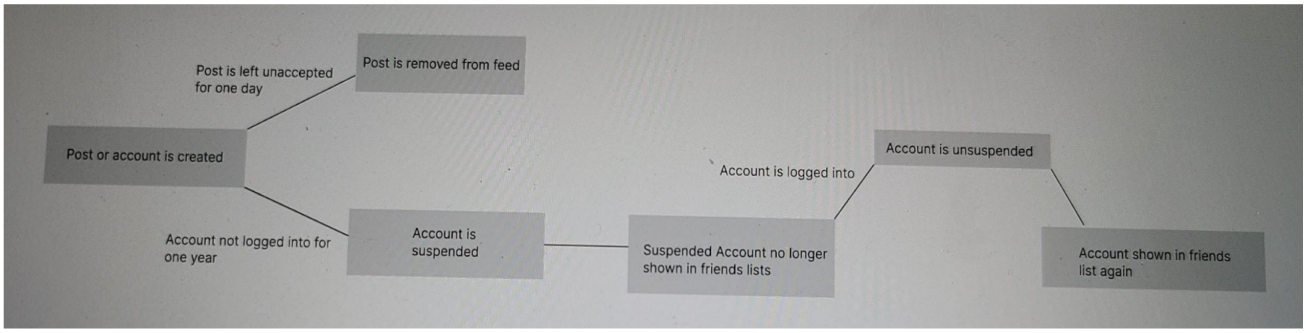

Notify
Posts a listing
Notifcation backend
Only selected friends
Selected group(S)
Everyone in the neigbouhood (GPS)
Notifications
Notifications
Notifications
post
post
post


Make/edit review
Add a review
Add description,pictures
All deatils filled
review added to the post
Feed
Edit review
Full details not filled

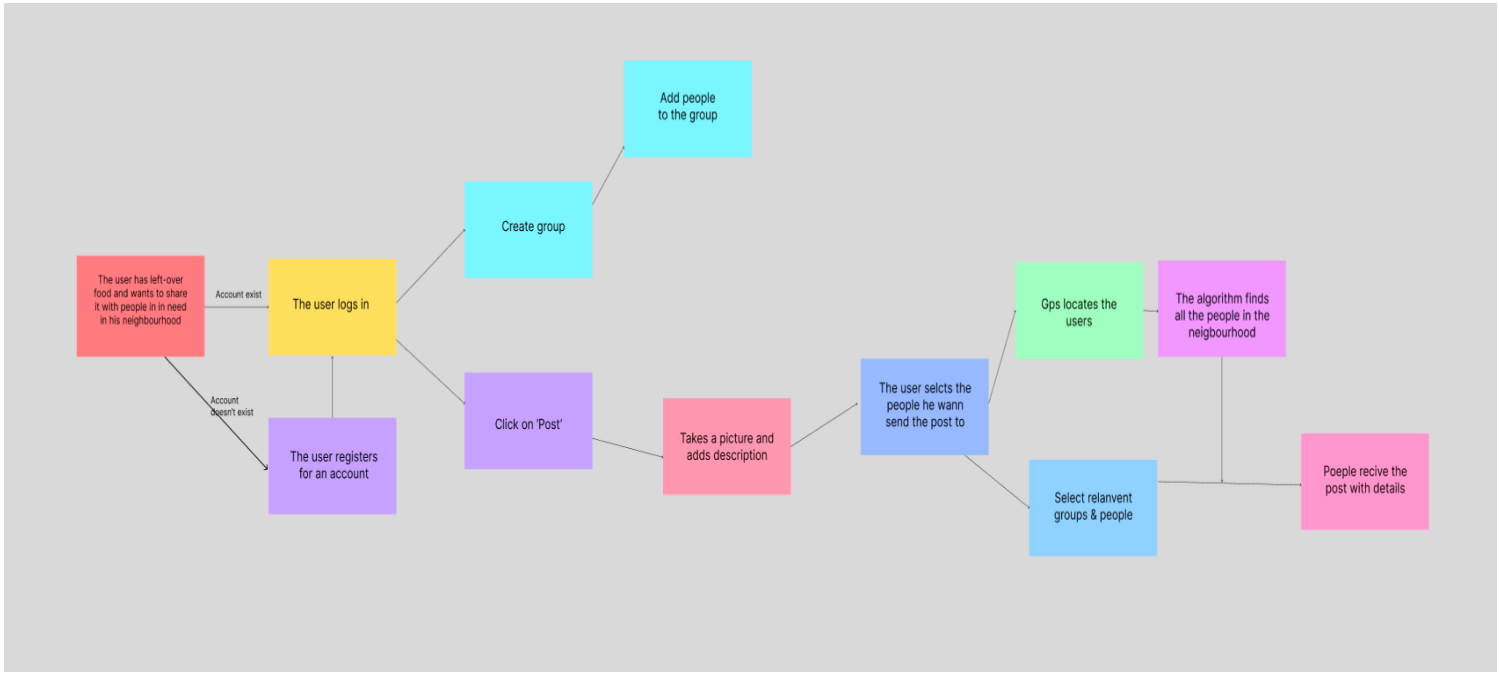

Manage friend/ groups
Search friends
Friend found
Add friend
Create group
Add friend to the group
Delete a group
Delete a friend



Suspension/Deletion of Account/Post



9.6 Activity/State (Scenario) Sectional

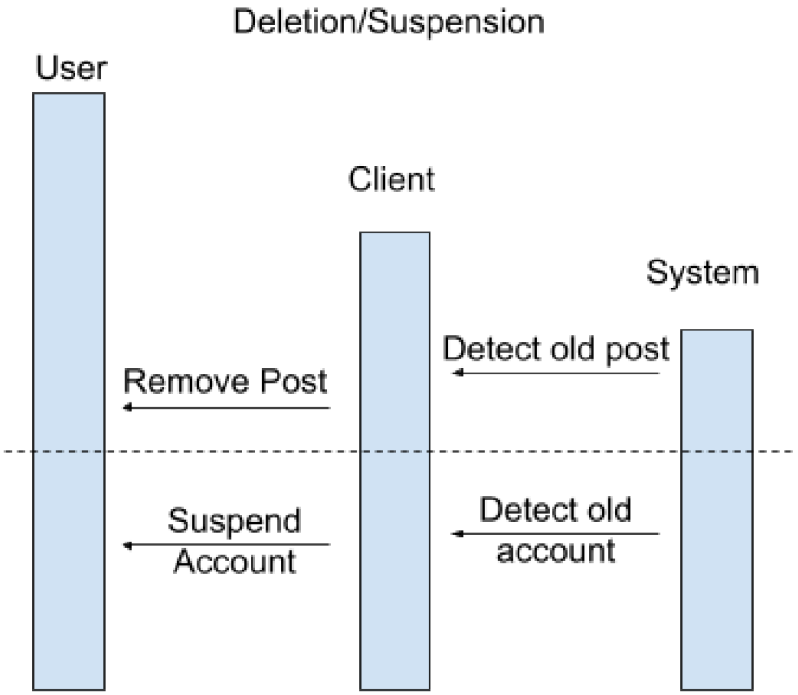


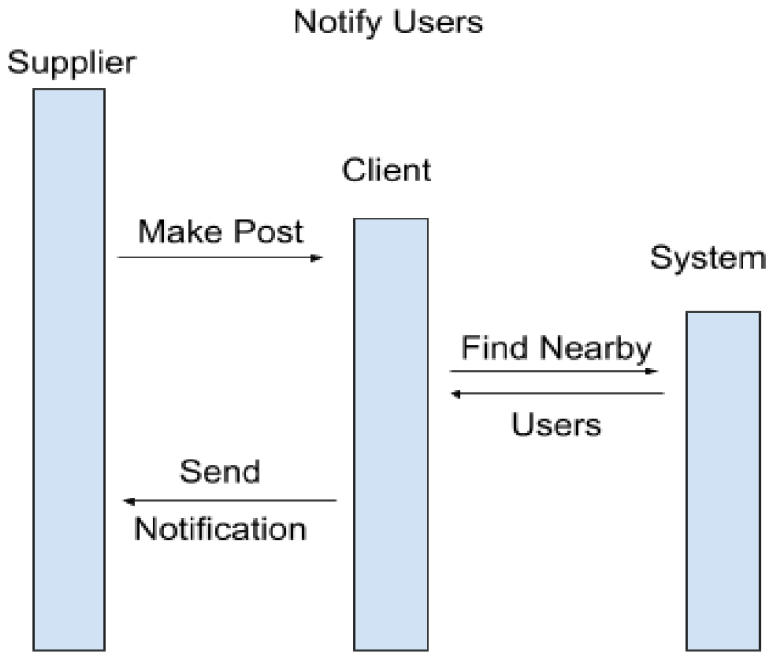
9.7 State logic (Started in analysis and completed in design)

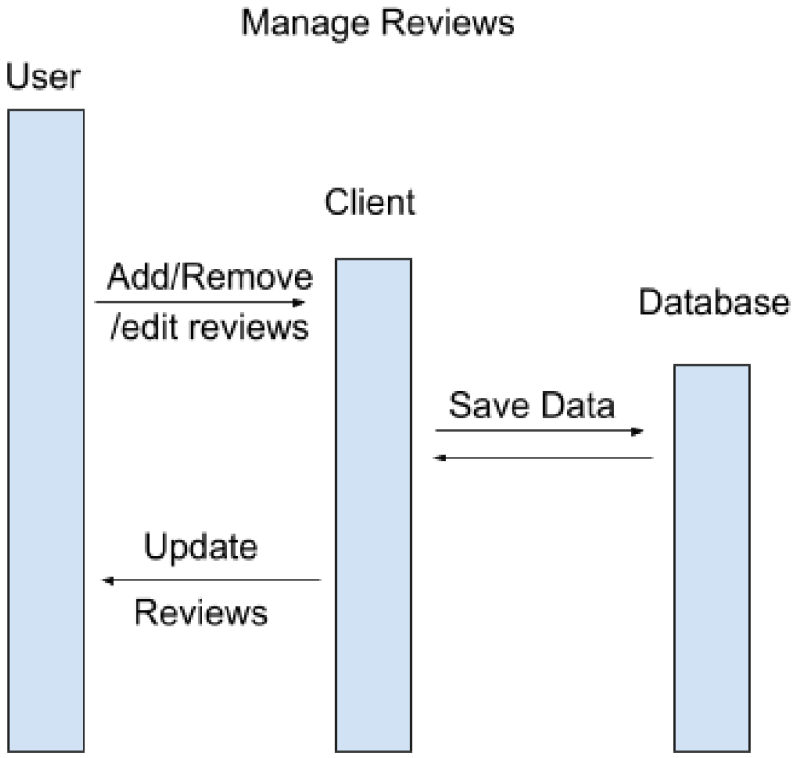
State	Description
Searching	App is searching for available vendors nearby
Active	Whenever a new vendor appears, you will receive a notification if in the area
Disabled	Account is disabled and will not receive any notifications and will automatically delete itself after a certain amount of time Post deletes itself after a certain amount of time
Posting	User is either creating a post or reviewing a certain vendor. Mostly any user interaction will be in this state.

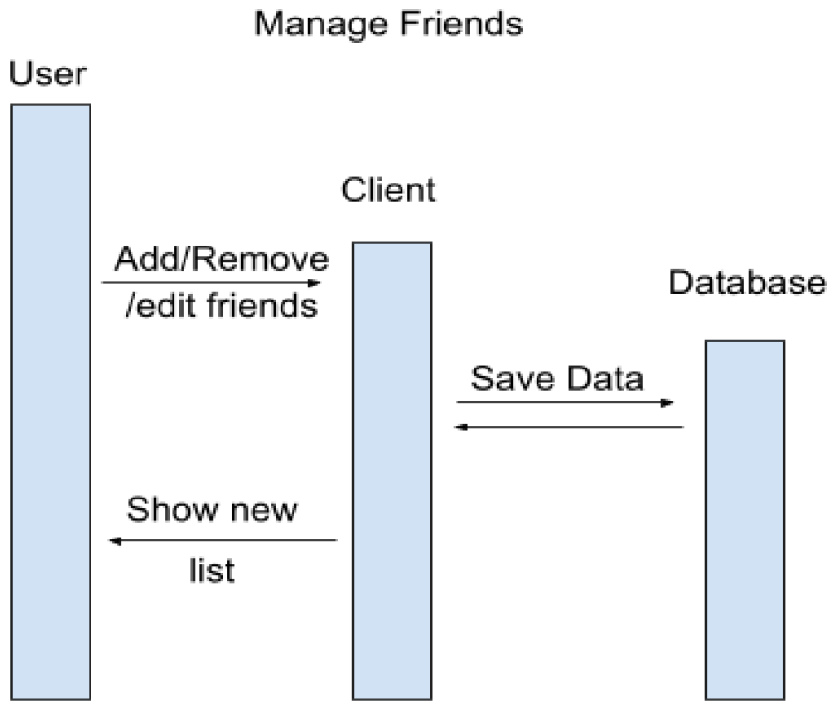
9.8 Behavior

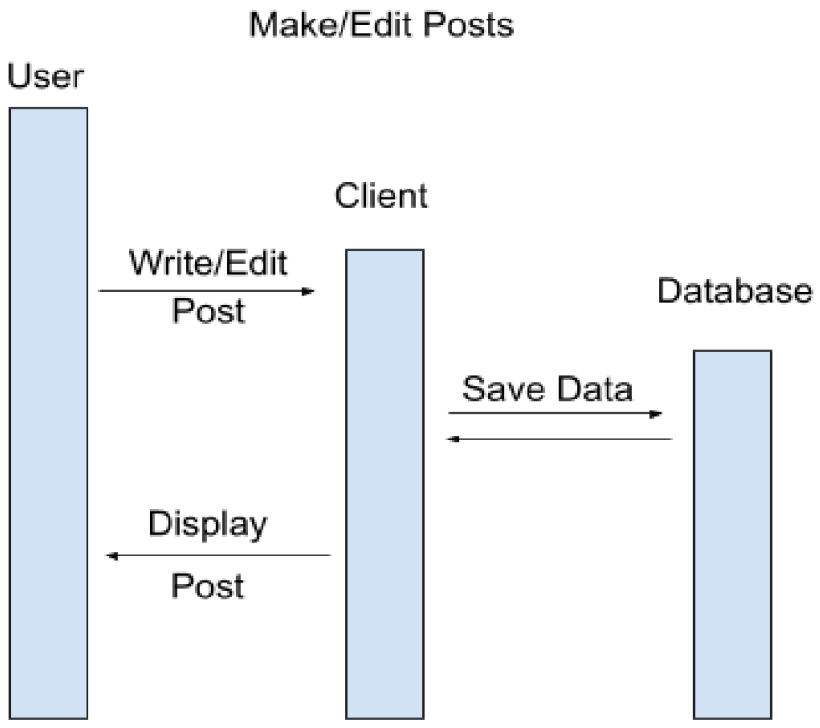
6.8.1 Sequence Diagrams

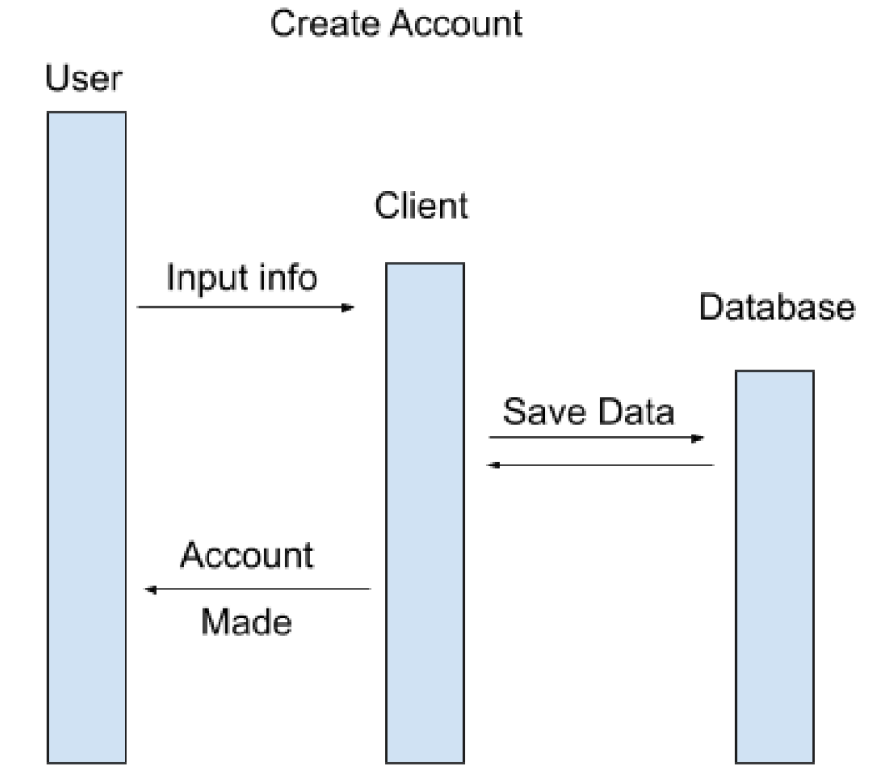


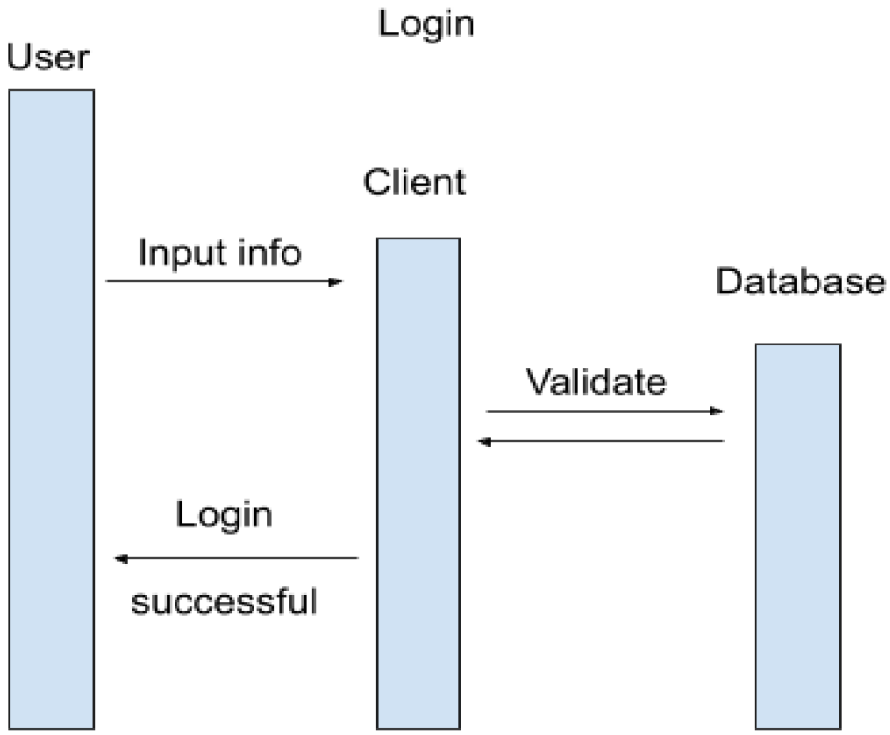

Notify Users
Supplier
Client
Make Post
System
Find Nearby
Users
Send
Notification


Manage Reviews
User
Client
Add/Remove
/edit reviews
Database
Save Data
Update
Reviews


Manage Friends
User
Client
Database
Add/Remove
/edit friends
Save Data
Show new
list


Make/Edit Posts
User
Client
Database
Write/Edit
Post
Save Data
Display
Post


Create Account
User
Client
Input info
Database
Save Data
Account
Made


Login
User
Client
Input info
Database
Validate
Login
successful

10 SYSTEM TEST PLAN REQUIREMENTS

System test cases/scenarios

Software Quality Assurance Testing Process:

1. If the app launches properly with all the features and functionality
2. If the user can register/login into the app
3. If the user can see the feed of all the pickups available
4. If multiple users can access the app simultaneously
5. If the user interface is seamless
6. If all the content on the app are properly aligned and well managed

Product Testing:

1. Test out the GPS by simulating an order
2. If the system sends out local notifications once an order is posted
3. Check if the user is able to set up profiles with their names and any additional information they want to include in a text box
4. Simulate friend request to see if it works
5. If the system allows the users to create multiple groups and add friends to those groups
6. Simulate and check if the system allows the users to post the order ton only selective groups or friends
7. If the users are able to post by checking there verifications status
8. If the system can only allow the post to be published if all the information is filled
9. If the application can access the camera and can take pictures and upload them to the post.
10. If the system only allows to uploads photos/videos that are under certain reserved storage
11. If the system allow to post one listing at a time
12. If the user able to delete or edit their listings after publication
13. Once the user picks up the food, is the user able to give a review to the user who posted it.
14. If the user able to submit a complaint about the order
15. If the app is directly users to the complaint box if he gives a 1 start review
16. If the post gets deleted if the account is suspended or deleted by the developers.
17. Check if the app has all the required content.

Acceptance Testing:

1. This testing is conducted as part of customer sign off on the finished software

2. If the user is satisfies after using the app or in other words user does nor find it difficult to use the application
3. If the the user is able to use it multiple times in a day
4. If the user likes the features and functionalities of the application

11 QUALIFICATION PROVISIONS

1. Once the document is complete, it will be checked for any spelling/grammatical errors
2. The developer will walkthrough the document along with an audience with no software engineering experience to make the document more understandable for audience with no software engineering experience
3. Check if all the requirements are complete
4. In advance inform the audience of the technical content of the document
5. Check for inconsistencies in the content
6. Ensure if the technical concepts are used correctly
7. Ensure if there correlation between topics
8. The document will be inspected by the developers to improve the quality of the document to find and remove defects efficiently, as early as possible
9. Ensure every topic is completely explained
10. Ensure any topic is easily traceable using the ‘table of contents’
11. Produce the documents with a higher level of quality by exchanging information among the inspection audience
12. Feedback will be taken from other experts and normal audience and changed things to make the document more presentable

12 REQUIREMENTS TRACEABILITY

Requirement:

7.1.1:

Forward: Requirements 7.1.6, 7.1.12

Backward: Requirements 7.1.2, 7.1.3, 7.1.7, 7.1.8, 7.1.9, 7.1.11

7.1.2:

Forward: Requirements 7.1.1, 7.1.4

Backward: Requirements 7.1.5

7.1.3:

Forward: Requirements 7.1.9, 7.1.1

Backward: Requirements 7.1.1, 7.1.4

7.1.4:

Forward: Requirements 7.1.6, 7.1.12

Backward: Requirements 7.1.6, 7.1.12

7.1.5:

Forward: Requirements 7.1.6

Backward: Requirements 7.1.1, 7.1.2, 7.1.3

7.1.6:

Forward: Requirements 7.1.9, 7.1.7

Backward: Requirements 7.1.4, 7.1.5

7.1.7:

Forward: Requirements 7.1.9

Backward: Requirements 7.1.6

7.1.8:

Forward: Requirements 7.1.9,

Backward: Requirements 7.1.7

7.1.9:

Forward: Requirements 7.1.10, 7.1.11, 7.1.12, 7.1.13

Backward: Requirements 7.1.8, 7.1.7

7.1.10:

Forward: Requirements 7.1.1, 7.1.2, 7.1.9, 7.1.12, 7.1.13

Backward: Requirements 7.1.3, 7.1.7, 7.1.8, 7.1.11

7.1.11:

Forward: Requirements 7.1.9, 7.1.10

7.1.12:

Forward: Requirements 7.1.1, 7.1.7

Backward: Requirements 7.1.10

7.1.13:

Forward: Requirements 7.1.1

Backward: Requirements 7.1.7

7.1.14:

Forward: Requirements 7.1.16, 7.1.17

Backward: Requirements 7.1.15, 7.1.18

7.1.15:

Forward: Requirements 7.1.14, 7.1.16, 7.1.17

Backward: Requirements 7.1.18

7.1.16:

Forward: Requirements 7.1.17

Backward: Requirements 7.1.15

7.1.17:

Backward: Requirements 7.1.16

7.1.18:

Forward: Requirements 7.1.14

13 EVOLUTION OF THE SRS

The document will evolve and change as the development process goes underway. A few things that would change might be the range/area that our app would cover. This could change depending on how well the application is doing whether it is reasonable to cover certain distances/areas. Another change could be in the technology, as our app may be able to cover other operating systems if testing goes well. Human resources and training, regulatory and legal, as well as market considerations are also subject to change depending on the reaction from users and government officials as the user base continues to grow.

14 APPENDICES

Schedule Tracking

Artifact or Deliverable	Who (individual or team)	Estimated	Actual	Difference
SRS	Nathan Cantu	6	10	4
	Jonathan Xu	5	8	3
	Vijay Kallem	5	6	1
	Summary for the entire team	16	24	8

Artifact or Deliverable	Who (individual or team)	Estimated	Actual	Difference
SPMP	For each team member			

	Summary for the entire team			

Artifact or Deliverable	Who (individual or team)	Estimated	Actual	Difference
SDD - Initial	For each team member			
	Summary for the entire team			

Artifact or Deliverable	Who (individual or team)	Estimated	Actual	Difference
SDD - Final	For each team member			
	Summary for the entire team			

Cumulative

Who (individual or team)	Estimated	Actual	Difference
Summary for the entire team			

Defect Tracking

Artifact or Deliverable	Who (individual or team)	Estimated	Actual	Difference
SRS	Nathan Cantu	5 per page	3 per page	2 per page
	Vijay Kallem	3 per page	8 per page	5 per page
	Jonathan Xu	4 per page	3 per page	1 per page
	Summary for the entire team	12 per page	14 per page	2 per page

Artifact or Deliverable	Who (individual or team)	Estimated	Actual	Difference
SPMP	For each team member			
	Summary for the entire team			

Artifact or Deliverable	Who (individual or team)	Estimated	Actual	Difference
SDD - Initial	For each team member			
	Summary for the entire team			

Artifact or Deliverable	Who (individual or team)	Estimated	Actual	Difference
SDD - Final	For each team member			
	Summary for the entire team			

Cumulative

Who (individual or team)	Estimated	Actual	Difference
Summary for the entire team			

15. Dictionaries

Classes:

Name	Description	Methods	Attributes
Account	User account information	login(arglist) : return register(arglist): return Send_request(arglist) : return accept_request(arglist) : return	user_id : integer password: string full_name : string user_email : string contact_no : integer SSN : integer friends_list : list of Account objects rating_percentage : integer is_producer : bool group_list : list of Group objects
Group	Group of friends that the user creates that the user can send notifications to instead of the general public	add_member(arglist) : return delete_member(arglist) : return	group_list : list of Account objects names_list : list of strings
Post	Publication of a listing made by a producer	summary(arglist) : return cancel_reservation(arglist) : return	user_id : integer full_name : string user_email : string contact_no : integer Rating_percentage : integer pickup_time : integer pickup_place : string Reserved : bool
Customer near address	A user within the same vicinity as the public listing is notified, customer can request the listing	make_reservation(arglist) : return cancel_reservation(arglist) : return	user_id : integer full_name : string user_email : string contact_no : integer has_reservation : bool
Admin	Administrative office that has access to every	validate_ssn(arglist) : return	user_id : integer password: string

	Account object, validates SSN to promote them to supplier		full_name : string user_email : string SSN : integer contact_no : integer friends_list : list of Account objects Rating_percentage : integer

Methods:

Name	Description	Class	Arguments
login()	Validates the user using email and password and gives the user access to their personal data	Account	User inputted email User inputted password
register()	Creates a new Account object	Account	User inputted email User inputted password
Send_request()	Sends a friend request to another user	Account	Account object friends_list
Add_member()	Adds a member to the group only if that member is in the friends list	Group	Account object Friends_list
delete_member()	Deletes a member from the group	Group	Account object
validate_ssn()	Validates the users SSN	Admin	SSN user_id Full_name

			is_producer

Summary()	Presents a summary of the listing information to be printed onto the listing	Post	Full_name Contact_no User_email Rating_percentage Pickup_time Pickup_place

make_reservation()	Accepts listing made by producer	Customer near address	User_id User_email Full_name Contact_no

cancel_reservation()	Producer or user cancels reservation	Customer near address	Reserved User_id user_email

Accept_request()	Accepts or declines a friend request sent by another user	Account	Whether or not the user accepts

Attributes:

Name	Description	Type	Complexity

user_id	User identification number	Integer	Simple

password	User login identification sequence of characters	String	Simple

full_name	Name that user wants other users to call them by	String	Simple

user_email	User contact and log in identification email address	String	Simple

Contact_no	User phone number for contact and log in identification	Integer	Simple

SSN	User’s social security number used to hold the user liable for wrongdoing	Integer	Simple

Friends_list	List of users that this user has either sent or accepted a friend request	List of Account Objects	Complex

rating_percentage	The percentage number calculated through reviews made on the user’s account	Integer	Simple

Is_producer	Whether or not this user is certified as a producer	Bool	Simple

Group_list	List of users who are members of the same group	List of Group objects	Complex

Names_list	List of people's names in	List of strings	Complex

	the Group		

Pickup_place	Address that the listing is located	String	Simple

Pickup_time	Time that the listing will expire by	Integer	Simple

Reserved	Whether or not this listing is reserved	Bool	Simple

Has_reservation	Whether or not the user has the reservation to pick up the food	Bool	Simple

Relationships:

Name	Description	From Class	To Class
Accessing information	Admin has access to every Account object's information	Account	Admin
Posting Listings	Producers can post listings	Account	Post
Refer public	Notifies those within the same area of a new listing	Post	Customer near address

Refer group	Notifies those within the same group of a new listing	Post	Group

Creating Group	User creates a group of friends	Account	Group

Public requests	User requests reservation	Customer near address	Post

Group requests	Group member requests reservation	Group	Post

Confirms public	Producer confirms reservation to user	Post	Customer near address

Confirms group	Producer confirms reservation to group member	Post	Group

Key Events:

Name	Description	Motive	Action	Pre-Conditions	Post-Conditions	State Change

User becomes producer	A user submits their SSN for verification in order to	User wants to start posting listings	User submits SSN	SSN field is empty Is_producer is false	SSN field is full Is_producer is true	User becomes a producer

	become a producer					

User joins group	A user is added to a group by the user who created it	Friend wants to join exclusive rights to notifications	User adds a friend to the group	Group_list initial Names_list initial	Account object is added to the group_list Friend's name added to the names_list	Friend joins group

Member leaves group	A user is deleted from the group	User does not want to have exclusive rights to notifications anymore	User deletes either themselves or another user from the group	Group_list initial Names_list initial	Account object is removed from the group_list Friend's name removed from the names_list	User leaves group

Customer gets listing reservation	A user requests reservation	User wants to acquire listing contents	User requests the listing	Has reservation is false Reserved is false	Has reservation is true Reserved is true	User acquires reservation

Customer loses reservation A user cancels reservation	A user cancels reservation or a producer cancels the listing	User does not want to acquire listing contents anymore	User cancels reservation Producer cancels listing	Has reservation is true Reserved is true	Has reservation is false Reserved is false	User loses reservation

Timer cancels reservation	The window that the user who has the reservation has for getting to the location	Since user probably won't be coming to get the food another user can acquire it	Timer cancels the reservation	Has reservation is true Reserved is true	Has reservation is false Reserved is false	User loses reservation

	is over and the listing becomes open again					