Team 41
Shaun Keene
Evan Borszem
Jared Brown
Joey Foutty
Nitin Murthy
Zack Kilgus

1. Purpose

Reviews are broken. Between bots, review bombing, false reviews from the competition, and inflated reviews by the owners, it's getting more difficult to know what to trust. On top of that, they're impersonal and often vague, making it difficult to know if you and the reviewer have the same sense of taste. More often than not, people prefer to get advice from those close to them. They want recommendations from people they know and whose opinions they trust, but they aren't going to poll their entire contact list every time they're considering a new place. That's where Famigo can help. Famigo is a social review app that allows users to see reviews of places from accounts they follow and plan trips with friends. Whether you're looking for your family's opinions on the new restaurant in town, need help organizing a group trip, or are curious about what celebrities recommend you see at your next vacation destination, Famigo will help you travel smarter.

Functional

1. As a user, I would like to be able to create an account so that I can save my information
2. As a user, I would like to be able to edit my password and my username

3. As a user, I would like to be able to recover my password by email so that I avoid losing my account
4. As a user, I would like to be able to delete my account
5. As a user, I would like, if I am considered credible by other users (judged by the number of followers, likes, and quality of reviews according to the moderation team) to have my reviews and likes have more weight
6. As a user, I would like to follow other users so that I can see their reviews
7. As a user, I would like to block other users so that I no longer see their reviews and they cannot post comments on mine
8. As a user, I would like to be able to add likes and dislikes to reviews
9. As a user, I would like to be able to see how many reviews someone has written
10. As a user, I would like to be able to read reviews even without an account
11. As a user, I would like to view other people’s profiles
12. As a user, I would like to be able to message friends so that I can coordinate schedules
13. As a user, I would like to be able to edit my messages
14. As a user, I would like to be able to delete my messages
15. As a user, I would like to send media like images in messages
16. As a user, I would like to be able to add rich text like bold and italics to messages
17. As a user, I would like to add friends as people who follow me and I follow back

18. As a user, I would like to see the reviews categorized with gold reviews being friends, silver reviews being people I follow, and bronze reviews being everyone else
19. As a user, I would like to see the credibility of users’ reviews
20. As a user, I would like the UI to scale to computers and phones of all sizes for ease of use
21. As a user, I would like to have a personalized page for people to check out and see my stats
22. As a user, I would like to hide my profile from searches
23. As a user, I would like to restrict my profile to only friends or friends of friends
24. As a user, I would like to be able to see a feed of the most recent posts of people I follow
25. As a user, I would like to be able to send files through messages to other people
26. As a user, I would like to be able to share calendars with other people
27. As a user, I would like to be able to report inappropriate content so that a moderator’s attention will be drawn to it (separate than marking reviews as offensive, also covers comments and chats)
28. As a user, I would like to be able to invite people with an email link to join my travel group
29. As a user, I would like to be able to remove people from my travel group

30. As a user, I would like ownership of a trip to transfer to the member of the trip who joined second if the trip creator leaves or is removed from the product or allow the trip planner to designate a new group leader
31. As a user, I would like to be able to see a consensus on how people feel about a location (if time permits)
32. As a user, I would like to be able to tag other users with @ or use # to search for categories and I would like to send clickable hyperlinks in messages and reviews
33. As a user, I would like to be able to see a map of all the places I have reviewed (if time permits)
34. As a user, I would like to be able to sort the reviews based on the star rating so that I can see the worst and best experiences
35. As a user, I would like a light mode and a dark mode
36. As a trip planner, I would like to create groups so that I can plan trips with them
37. As a trip planner, I would like to be able to create trips so that the different schedules can be separated and I would like to be able to delete those trips
38. As a trip planner, I would like to create events for my trip on a specific day so that I can organize my trip
39. As a trip planner, I would like to be able to see those events in a calendar so that I can visually see where I need to be and at what time

40. As a trip planner, I would like to be able to see the locations on a map so that I can quickly find where I need to go
41. As a trip planner, I would like to be able to create reminders for events so that I know when to leave
42. As a trip planner, I would like to be able to export those events to another calendar so that I have multiple methods of keeping track of my schedule
43. As a reviewer, I would like to be able to add places to review so that others can read about reviews for these places
44. As a reviewer, I would like to be able to add reviews (containing text and a number of stars) to places so that others can learn from my experiences there
45. As a reviewer, I would like to be able to edit my reviews so that they contain my accurate opinions
46. As a reviewer, I would like to be able to see my edit history for reviews so that I can revert to an earlier version if I liked it more (if time allows)
47. As a reviewer, I would like to add media like images to my reviews to give a better idea of what the place is like
48. As a commenter, I would like to add comments to reviews so that I can argue for or against others’ statements
49. As a reviewer, I would like to delete my reviews, which also delete their associated comments

50. As a commenter, I would like to delete my comments, but if my comment has any replies, they will still be visible and my comment's author and text will simply change to "[deleted]"
51. As a reviewer, I would like to see statistics on how many reviews I have written, what my average number of likes is, and other data
52. As a reviewer, I would like to see all of the reviews I have posted on my page
53. As a moderator, I would like to ban users who are spreading inappropriate content
54. As a user, I would like to appeal a ban if I receive one that I believe is unfair
55. As a moderator, I would like to be able to unban users who make an appeal and seem genuine about changing their ways
56. As a moderator, I would like to be able to see and message other moderators so that communication is simpler
57. As a moderator, I would like to be able to remove inappropriate comments or reviews
58. As a moderator, I would like to see reports with the reporting user, report reason, and attached content
59. As a moderator, I would like to be able to see who has specifically liked, disliked, and flagged reviews while regular users can only see totals
60. As an owner of a location, I would like to be able to claim ownership of that location

61. As the owner of a location, I would like to be able to add links to my website or a phone number so that people can contact me and book

Non-Functional

We aim to:

- Accommodate 50,000 users simultaneously
- Have a delay of less than 1s per request
- Achieve 99% uptime
- Encrypt private user data (not publicly shared text)
- Have 80% testing coverage

We want this program to operate as a small-scale test-bed for a new social media style, capable of expanding to a larger audience in the future. We will provide optimizations and usability expansions in the future, should this endeavor prove successful, but this program is meant to serve as a proof of concept. This is the reasoning behind the relatively lax requirements for simultaneous user count and response time for a social media program. We are focusing on functionality first, with the ability to expand and improve speed as necessary. However, we will be aiming for 99% uptime, as would be expected from a peer service of our caliber. We feel this, as a core requirement of the software, should be implemented at the ground level to ensure functionality and compatibility with our other functions. We plan to encrypt private and friend-only (if applicable) data to prevent abuse but see no reason to slow down our software securing public information. Despite the computational overhead, we feel this is necessary to grow trust in our brand from the outset. Finally, our test cases aim to cover 80% of the functionality of this program. This is, in our opinion, a reasonable standard to publish under to ensure a quality brand image and build trust with our users.

2. Design Outline

We plan to construct both a web-based and Android app-based UI to interact with the product. We will be using a server-client model with an SQL database in the backend. Many clients will send data to the same server via the website or the app and it will process the information, storing anything necessary in the database. We will be using Spring Boot to create the connection between the server and the client for the website.

Client

- The user will interact with the client to use the product
- The user will be able to send their usernames and passwords to be authenticated, reviews and comments to be posted, and messages to the server using fetch so that they can be handled
- The client will receive a response from the server, which will cause the UI to update the view of the client

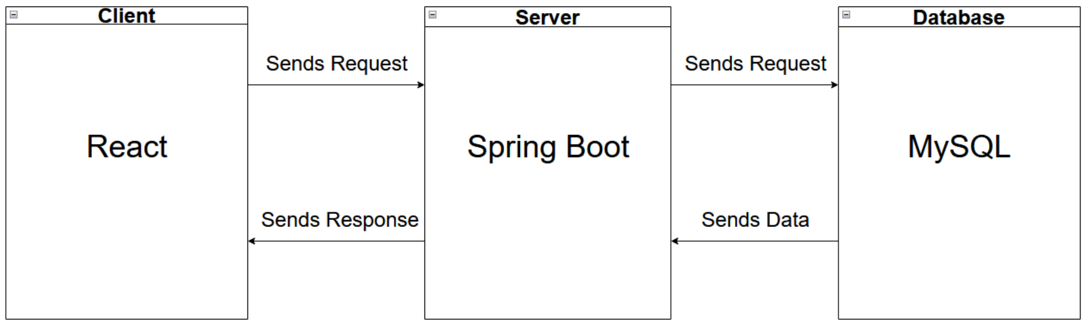
Server

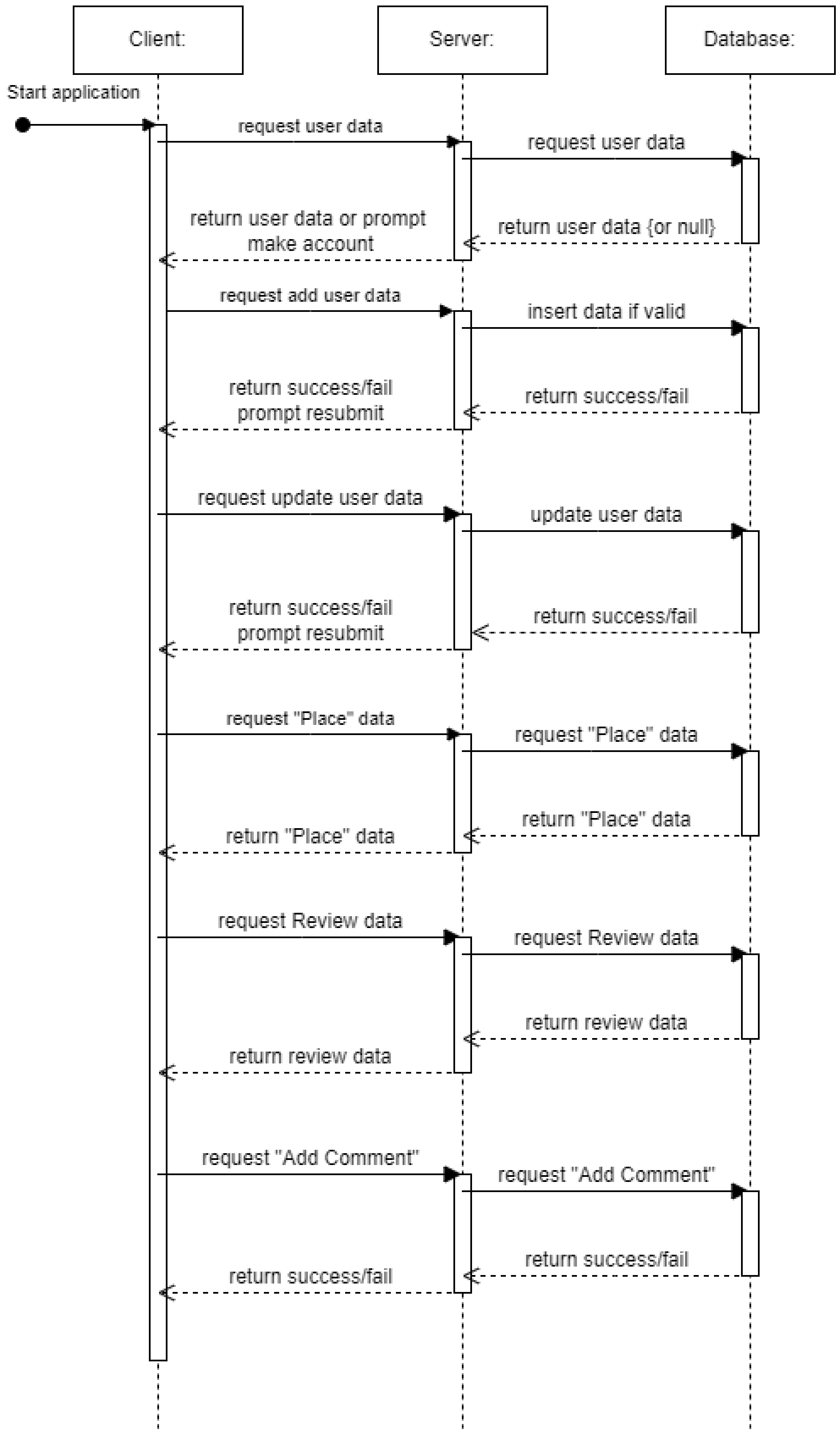
- The server receives requests from the client and handles them accordingly such as login requests or a request to add a comment

- If the server is sent new information like a new account, a new message, or a new review, it will store that by sending it to the database
- If there is a request for data like a message history, the server will ask the database and return the response to the client
- The server will then send a response or a redirection using Spring Boot

Database

- Receives data like login information, messages, reviews, etc. from the server to store in the database via SQL
- Returns any data requested by the server using SQL




Client:
Server:
Database:
Start application
request user data
request user data
return user data or prompt make account
return user data {or null}
request add user data
insert data if valid
return success/fail prompt resubmit
return success/fail
request update user data
update user data
return success/fail prompt resubmit
return success/fail
request "Place" data
request "Place" data
return "Place" data
return "Place" data
request Review data
request Review data
return review data
return review data
request "Add Comment"
request "Add Comment"
return success/fail
return success/fail

3. Design Issues

Functional

1. Issue: Review bots making fraudulent reviews for destinations

   Option 1: Check every review before its posted to ensure it is not a botted review

   Option 2: Limit users' ability to review multiple places in a given span of time

   Option 3: Ask people to not bot our site

   Solution: Add email verification before users can post reviews

   Justification: While stopping bots entirely is not possible, we want to significantly deter bots from our application. Adding email verification will deter botters by adding an extra layer of security and difficulty, so we expect much less of them. Checking every review would be cumbersome to the moderators and is subject to misuse by moderators. Limited users in artificial ways could also make them dissatisfied with the service; additionally, asking people not to bot will likely not lead to a solution.

2. Issue: Users will have no data when starting an account

   Option 1: Simply do not show many things in the user's feed until they have done things to provide the app with data on their interests. That way, the user will only be shown things they are interested in.

   Option 2: The user will simply be shown a general feed that is not tied to their interests. Over time, as they add friends, followers, and post about their experiences, their feed adjusts to show their interests more.

Option 3: Users will be asked to input some data to help the app get to know them faster. The data is optional, and might include approximate location, topics of interest, or immediately adding friends.

Solution: When a user creates an account, we will ask them about some general data, and use that data to show a combination of personalized content and general content. So when the user first signs up, they will not be bored, but they will also have control over what they immediately see.

Justification: It makes no sense to show users nothing when they start a new account. In giving new users a somewhat personalized and somewhat general feed, we will help jumpstart the account's following opportunities. This will save the user a ton of time having to individually find the people and topics they want to follow.

3. Issue: Some people always rate low no matter the experience

   Option 1: Do nothing

   Option 2: ban people for consistently low ratings

   Solution: Users will have pages that will publicly show some important data (average review score, number of reviews).

   Justification: Being able to see what a user is rating places is important when looking at the authenticity of a review. In today's age, a three star review is sub-par; however, it is not bad if it comes from someone who does not give anything above a two. Showing average scores ensures that people will be able to spot those who just review things lower on average. Banning or not addressing this issue could cause harm to users by not allowing them to express themselves or trust the site.

4. Can users access our service without creating an account/logging in?

   Option 1: Require account creation and login to view and create content

   Option 2: Allow users to view without account

   Option 3: Allow users to view and create without an account

   Decision: We chose to require a user account to view or create content on our services. Our main goal is to create a more personal environment that allows users to get information from and give advice to those they know, as well as discouraging bot-reviews. Allowing reviews or comments from non-users invites the possibility of botting, and viewing content would make us like any other review site. If you want personalized reviews, you must have an account to follow people whose reviews you trust

5. What platform should we make this program for?

   Option 1: A website (mainly for computers)

   Option 2: An app for IOS devices

   Option 3: An app for Android devices

   Decision: We decided to go with a website. Websites are accessible from all three platforms, giving it more versatility, and gives us the option to move towards app integration in the future if we desire.

6. Do we prioritize friends over followed in sorting, ratings, and other aggregates?

   Option 1: Friends are considered separate from followers and given priority

Option 2: Since "Friends" are just users who mutually follow each other, they are considered equally

Decision: Friends are a separate, prioritized category given higher regard in aggregates. Separating friends and followers allows more flexibility and user control in sorting and selecting, allowing them to more easily find reviews from certain users. Friends are given priority between the two because our focus as a platform is about personalization, and users who mutually follow each other are more likely to have a personal relationship off of the platform.

7. How should we default to ordering Reviews?

   Option 1: By date posted/modified (most recent to oldest)

   Option 2: By interactions (comments, likes)

   Option 3: By positivity (likes - dislikes)

   Option 4: By user relations primarily (friends, following, first degree relations, others), then by date posted/modified

   Option 5: By user relations primarily (friends, following, first degree relations, others), then by interactions

   Option 6: By user relations primarily (friends, following, first degree relations, others), then by positivity

   Option 7: By user relations primarily (friends, following, others), then by date posted/modified

   Option 8: By user relations primarily (friends, following, others), then by interactions

   Option 9: By user relations primarily (friends, following, others), then by positivity

Decision: We chose to default to sorting by user relations with the user's friends and following lists primarily, then by the number of interactions a review gets. Our goal as an app is to prioritize personalization, so we chose to show reviews from friends and followers over reviews from other users. We had debated including "first degree relations" for a time (meaning the friends and followed users of the individuals on your friends and following lists) to help alleviate problems arising from few to no personal reviews for locations, but we eventually decided that that undermined the personalization we were striving for. Finally, we chose to sort secondarily by positivity because we felt users would rather see information that other users found helpful rather than more recent or more eye catching content.

8. How does review removal effect comments on it

   Option 1: The comments remain on a blank review

   Option 2: All comments are deleted

   Option 3: Reviews cannot be removed

   Decision: All comments are deleted. We felt this was the most practical solution, as a blank review is not useful and the comments on it will no longer be relevant. Furthermore, we felt it was necessary to be able to remove reviews, both so users could retract statements and to remove inappropriate content. Besides, not allowing users to remove reviews, but allowing them to edit them would result in them deleting the content of the review but leaving a blank and useless review. Storage also comes into consideration, as this allocated space could be better used on current reviews and discussions

9. How does comment removal effect comment chains

Option 1: A blank or “comment removed” label takes it’s place

Option 2: All subsequent comments are removed

Option 3: Comments cannot be deleted

Decision: We chose to put a “comment removed” label in place of any deleted comment, so comment chains would not be disturbed by deletion. Unlike reviews, comment chains can still be relevant after the original comment is gone. Our reasoning for allowing comment deletion is the same as for review deletion.

10. Should we include an “edited” notation on changed messages, reviews, and comments

    Option 1: Yes, we will include an edited label to let users know when content has been changed

    Option 2: No, this is an unnecessary feature

    Decision: We will include an edited label so that users can transparently see when content has been changed retroactively. Many social media services have this feature as a standard and it gives users context to comments referring to information in previous messages that are no longer relevant.

Non-Functional

1. Which service will we use for this project?

   Option 1: GitHub

   Option 2: GitLab

   Option 3: SourceHut

   Decision: GitHub, because it is the service we are most familiar with.

2. Where will we run the server for this project?

   Option 1: On a team member’s computer

   Option 2: Pay a company for a server (AWS)

   Option 3: Use an free online sever (Firebase, Heroku)

   Decision: We will be using Firebase to run our server. Since we do not need a lot of server power, it does not make sense to pay for a bulky server to run our application. Additionally, hosting our server on a team member’s computer would make the service unreliable at best. Because of these facts, we will use one of the free hosting services for this application.

3. What will the website domain name be?

   Option 1: famigo.bin

   Option 2: knockofftravelsite.net

   Option 3: famigoonvacation.com

   Decision: famigo.bin. Since our product is called famigo, we should call the website something very similar to make sure users can identify the product. The .bin at the end is due to the domain famigo.com is already taken by a different service.

4. What color scheme will we use?

   Option 1: Calm, Professional Blue and muted orange style

   Option 2: Darker, designed like younger popular apps

   Option 3: White, fairly standard and unobtrusive

   Decision: We chose to go with a professional design. Blue and orange are complementing colors, and the muted tone conveys a put together and calm demeanor that is more welcoming than the darker UI of some newer programs, while not feeling as bright and stale as white.

5. What database should we use?

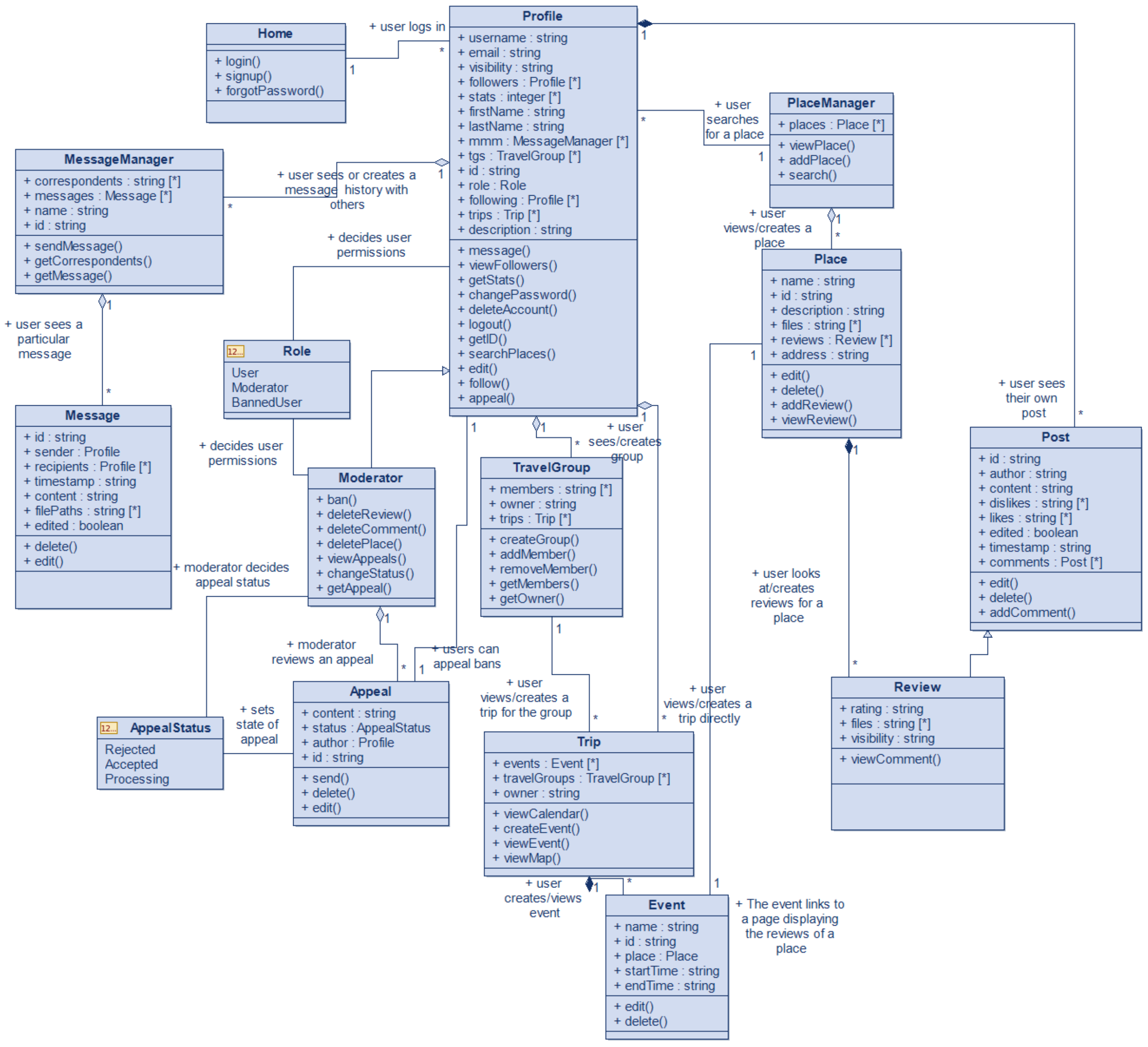
   Option 1: MySQL

   Option 2: MongoDB

   Option 3: Oracle Database

   Decision: SQL (specifically MySQL). The widespread use of SQL by companies and websites for their databases proves its reliability. We will use SQL, as it is proven to be a good database management system. SQL's scalability will also help with the growth of user data as the website is used by more people.

4. Design Details



Home

This class contains functions to log in and sign up for an account and serves the first web pages that you see. If you forget your password, there is a recovery option presented in the same class. Once you log in, you are taken to the profile page.

Profile

This class is essentially the hub of the site. It grabs all the relevant user information from the database to be at the disposal of other classes. It obtains the username, email, name, visibility level (who can see your profile and reviews), which conversations you have, your travel groups, the user’s id (which is the only constant used for identification across the system), the people you follow and the people who follow you, your current trips, and a profile description, if you have one. You can edit the personal information fields in this class, can create new conversations with message(), can follow people if viewing another person’s profile, can search for places to add to a trip or read reviews, and you can view statistics on how many reviews you have written, how many likes and dislikes you have received, and more.

MessageManager

This class is the class that contains all the messages in one conversation with any number of users. There is an id for the conversation, a name for the conversation, the messages sent, and the other members. You can send a new message from this class, look at specific people with getCorrespondents(), and look at specific messages with getMessage().

Message

This class represents one message. It has an id to distinguish it from others, a sender, recipients, a timestamp of when it was sent, the contents of what the message said, the paths of any files that were sent, and a boolean indicating if the message has been edited or not. You can edit or delete a message from this class.

Role

The Role enum will be used to determine what permissions certain users have. If they are moderators, they will get access to the moderator class with special functions.

Moderator

This class, inherited from Profile, contains special functions for moderators. They can ban users who violate community guidelines, judge appeals, and can delete comments, reviews, and places.

Appeal

This class represents an appeal against being banned that can be submitted from the Profile class. It contains the content saying why the user should not have been banned, the status of whether the appeal has been approved or rejected, an author of the appeal, and an id to distinguish it from others.

AppealStatus

This enum contains the three states for an appeal representing the judgements a moderator can pass on an appeal: rejected, accepted, or processing (meaning they haven’t gotten to it yet).

TravelGroup

This class represents a travel group: a group of users planning trips. It contains members of the group as their ids, an owner of the group that will transfer to another user if the owner loses their account, and a trips field that contains all the trips the group is planning. A user can create a group, see who the other users are, and see the owner. The owner can add and remove people.

Trip

This class represents one trip that users can go on. The events field represents all the events on the trip people are traveling to, the travelGroups field indicates all the travel groups using this trip, and the owner field is the owner of the trip. The owner can add events, see a calendar representation of the events, and see a map representation of the events planned.

Event

This class represents a single event in a trip. The name field is the title of the event, the id is the identifier used everywhere to represent it, the place is the location of the event using the Place class, and the start and end times are the duration of the event. The trip owner, who created the trip, can make changes to it.

PlaceManager

This class contains all the places that have been created by users to review. You can see an individual place, add a place, and can search for places.

Place

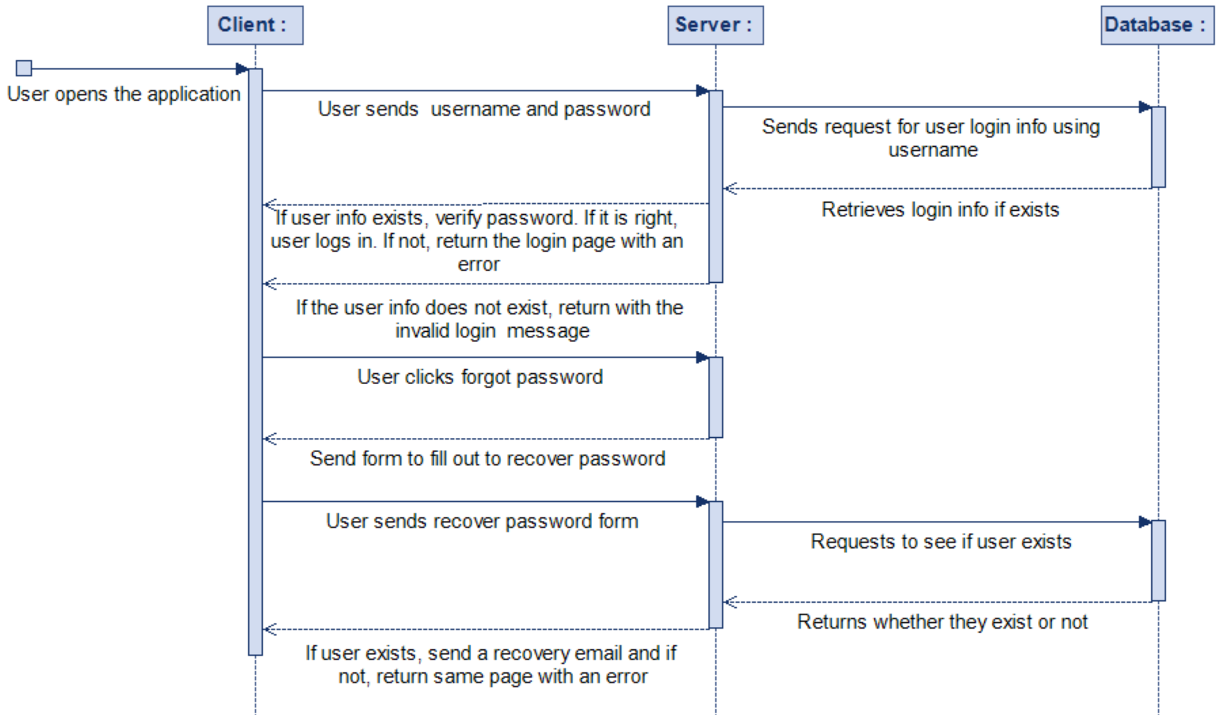
This class is a single location that can be reviewed. It should be unique and duplicates will be removed. A place has a name, an id that is associated with it in case the name changes, a description of what the place is, the paths of files which could be photos or videos of the place, and an address showing where it is. Moderators can edit or delete the place, and users can add reviews and see the reviews for the place.

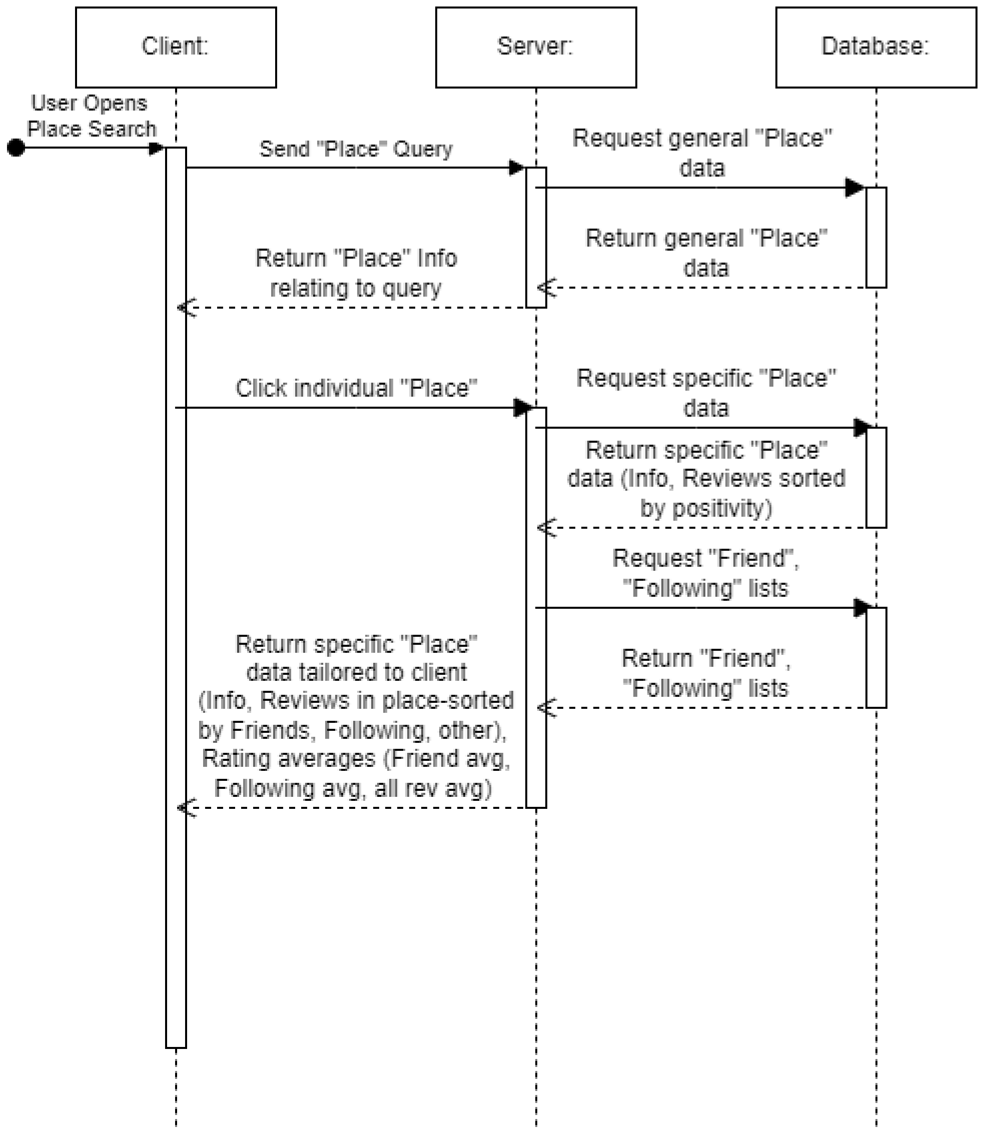
Post

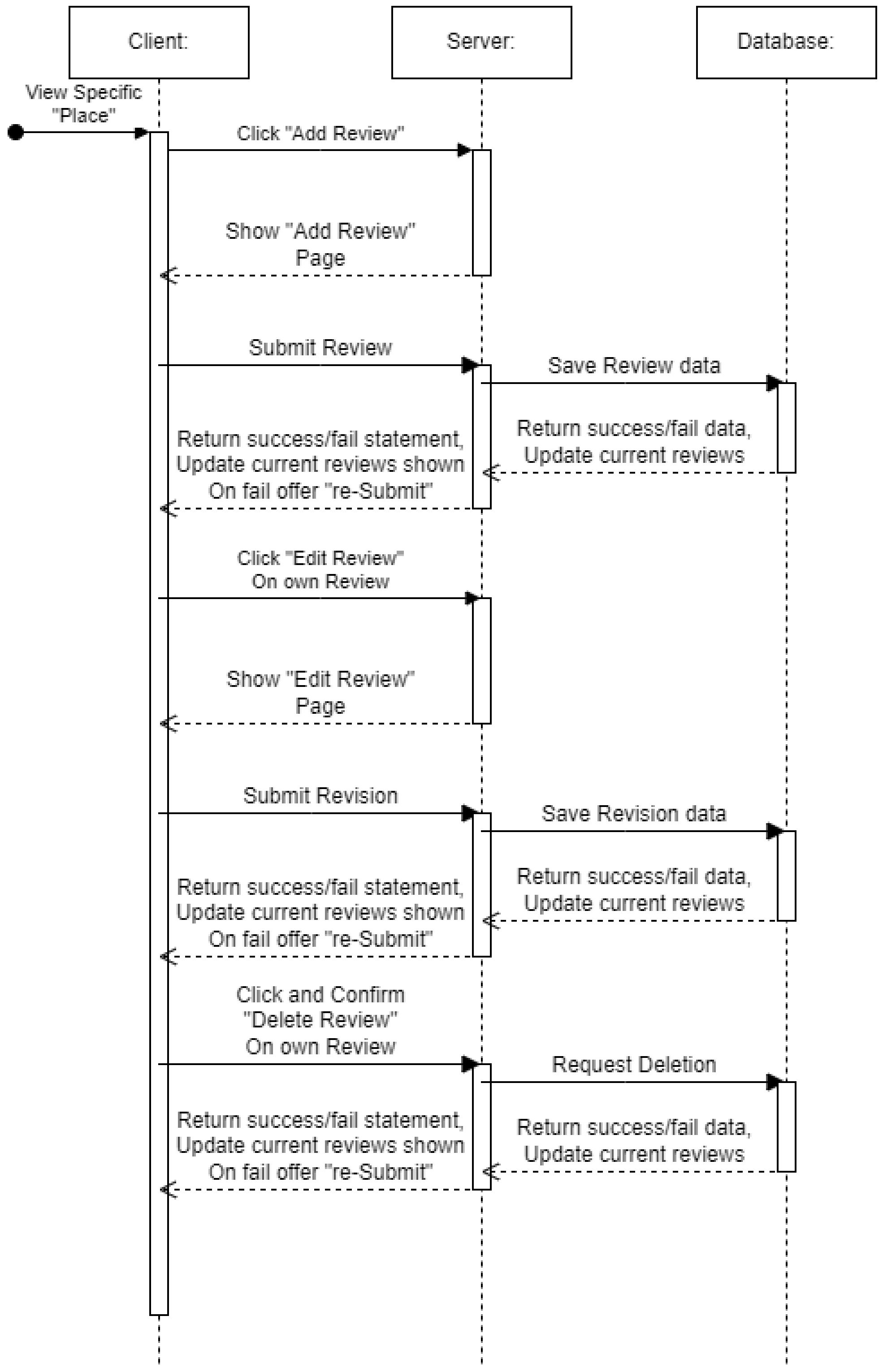
This class is a superclass containing all the information needed to create a comment and most of a review, which is inherited from it. Any post will have a unique id, an author, some content, dislikes and likes, a boolean value indicating if the post has been edited, a timestamp indicating when it was posted, and comments which can be attached to it.

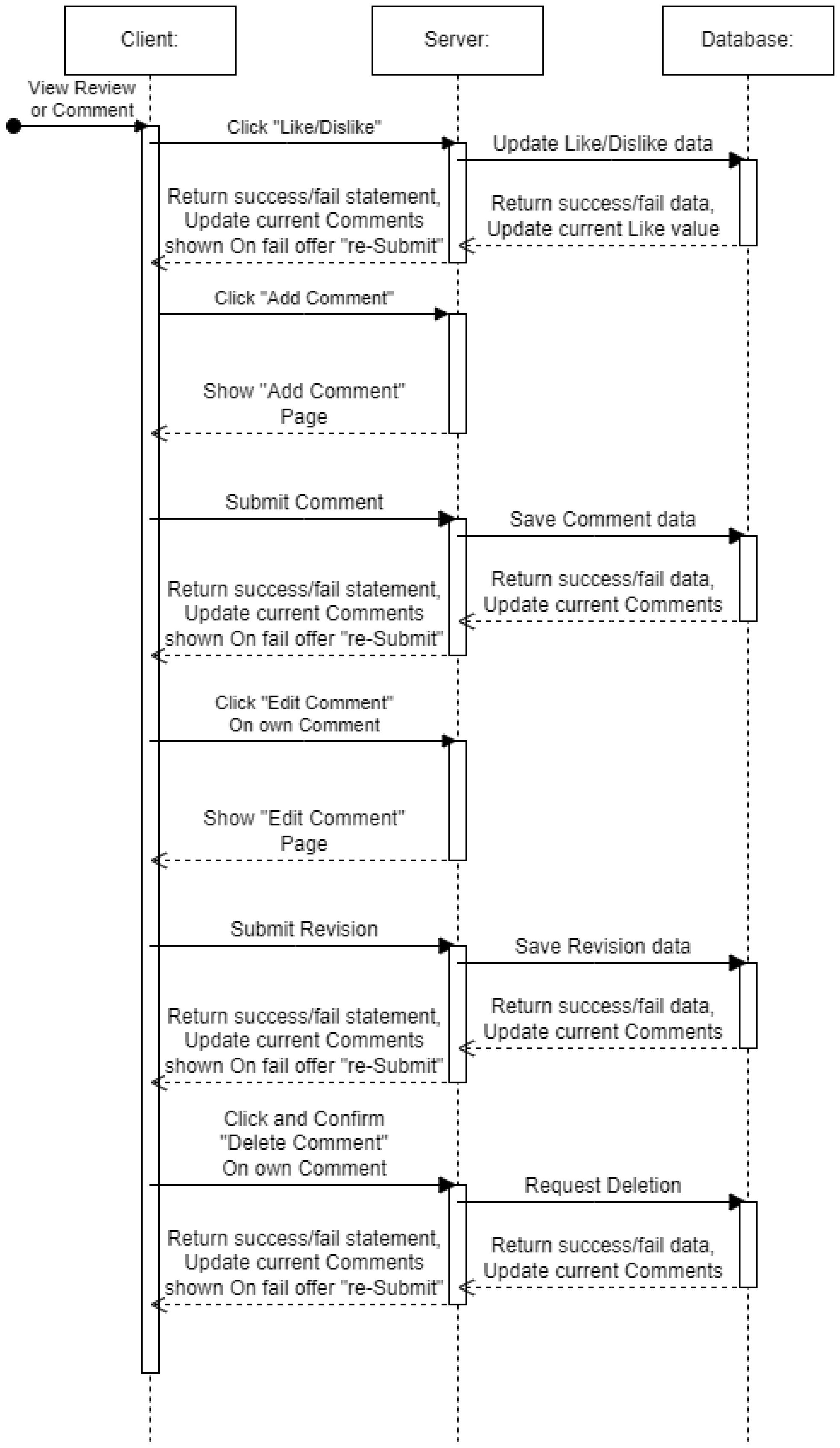
Review

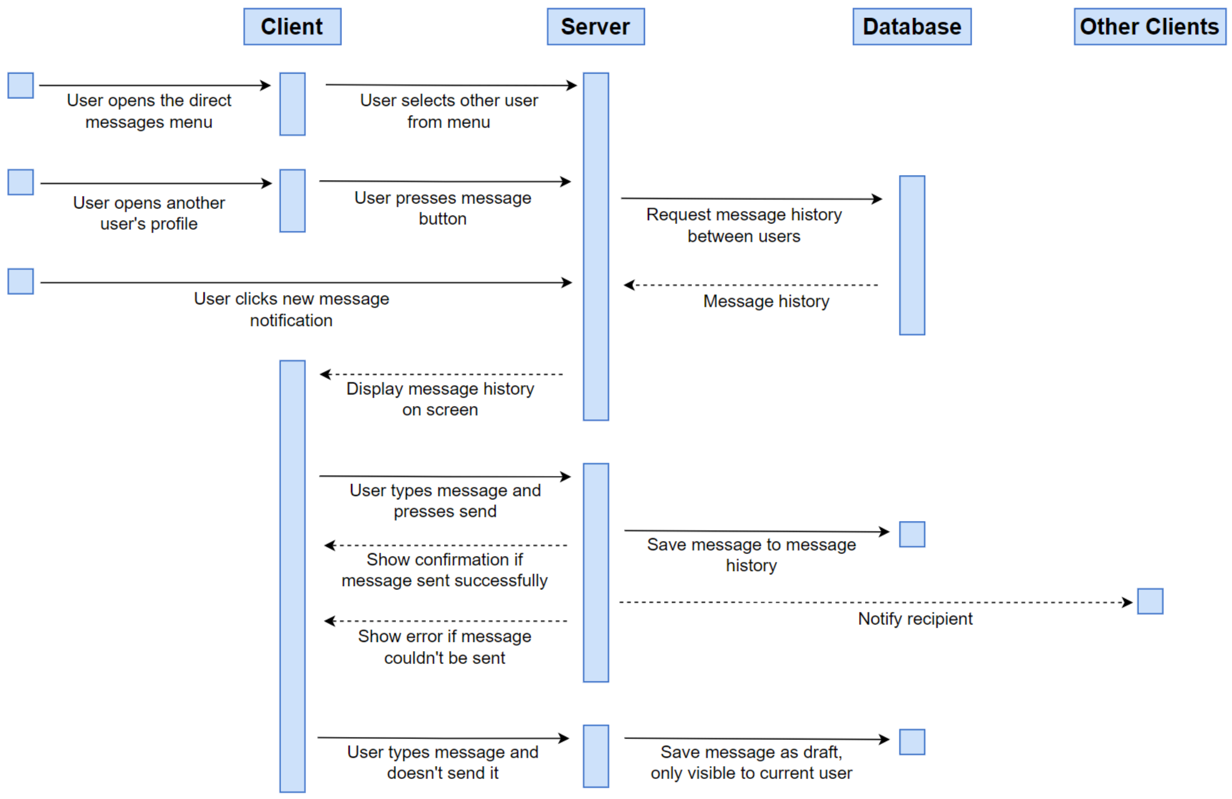
This class, inherited from Post, contains all the information for a review. A review will have a rating of the place with a certain number of stars, file paths containing images or videos showing what the place looks like, and whether the review is publicly available, only for friends, or followers.

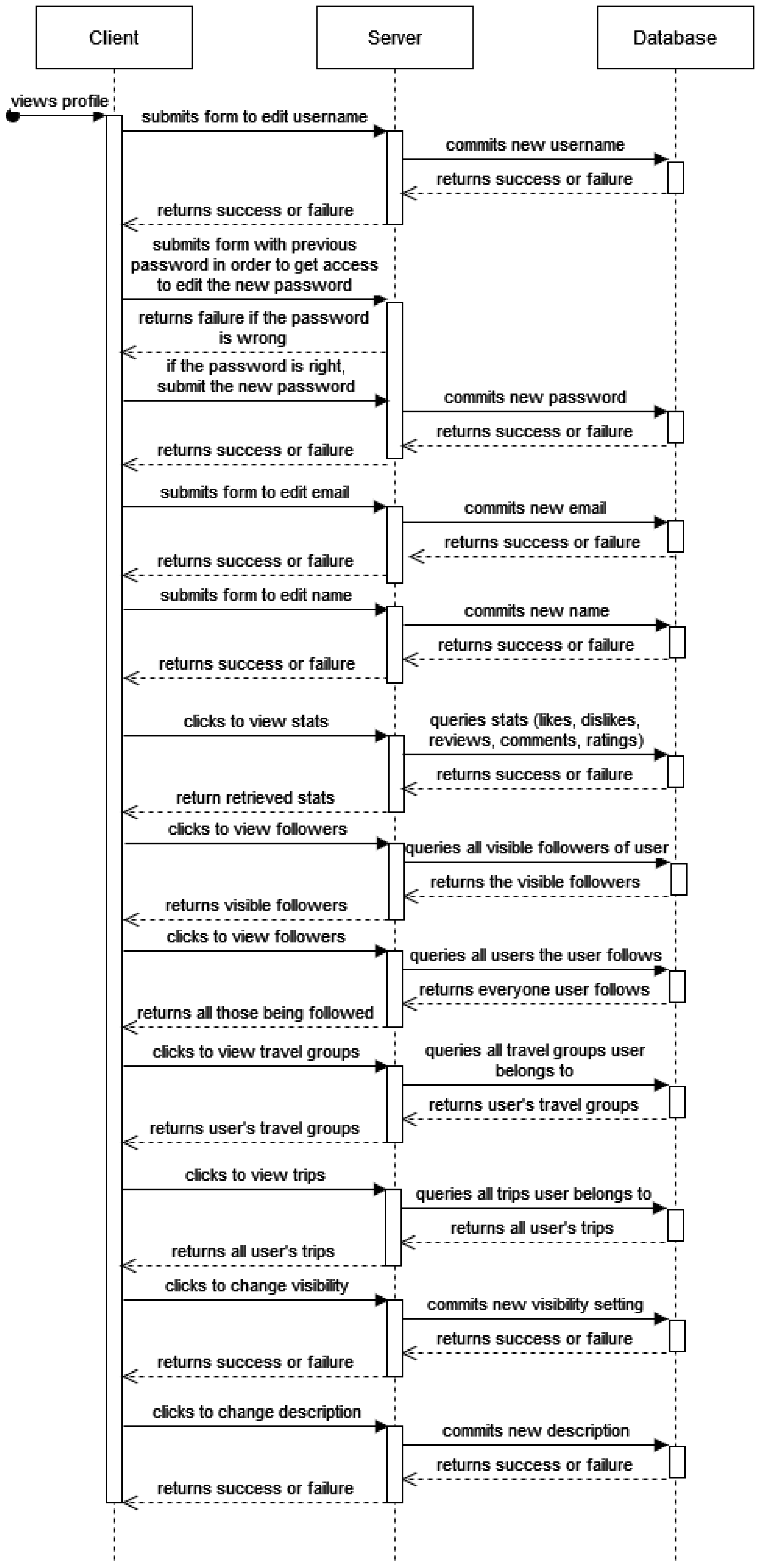

Client :
Server :
Database :
User opens the application
User sends username and password
Sends request for user login info using username
Retrieves login info if exists
If user info exists, verify password. If it is right, user logs in. If not, return the login page with an error
If the user info does not exist, return with the invalid login message
User clicks forgot password
Send form to fill out to recover password
User sends recover password form
Requests to see if user exists
Returns whether they exist or not
If user exists, send a recovery email and if not, return same page with an error

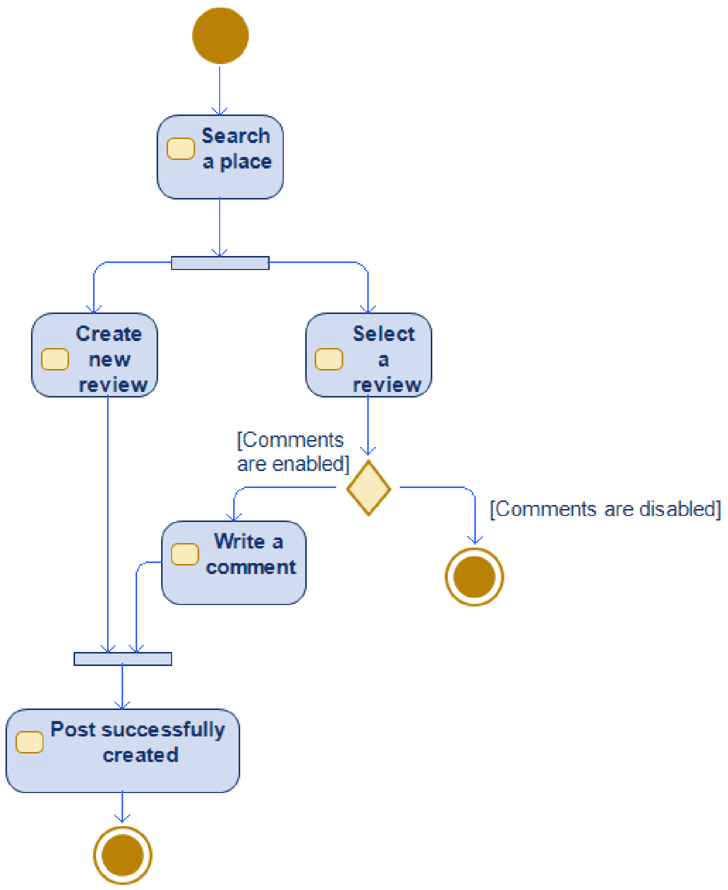

Client:
Server:
Database:
User Opens Place Search
Send "Place" Query
Request general "Place" data
Return general "Place" data
Return "Place" Info relating to query
Click individual "Place"
Request specific "Place" data
Return specific "Place" data (Info, Reviews sorted by positivity)
Request "Friend", "Following" lists
Return "Friend", "Following" lists
Return specific "Place" data tailored to client (Info, Reviews in place-sorted by Friends, Following, other), Rating averages (Friend avg, Following avg, all rev avg)

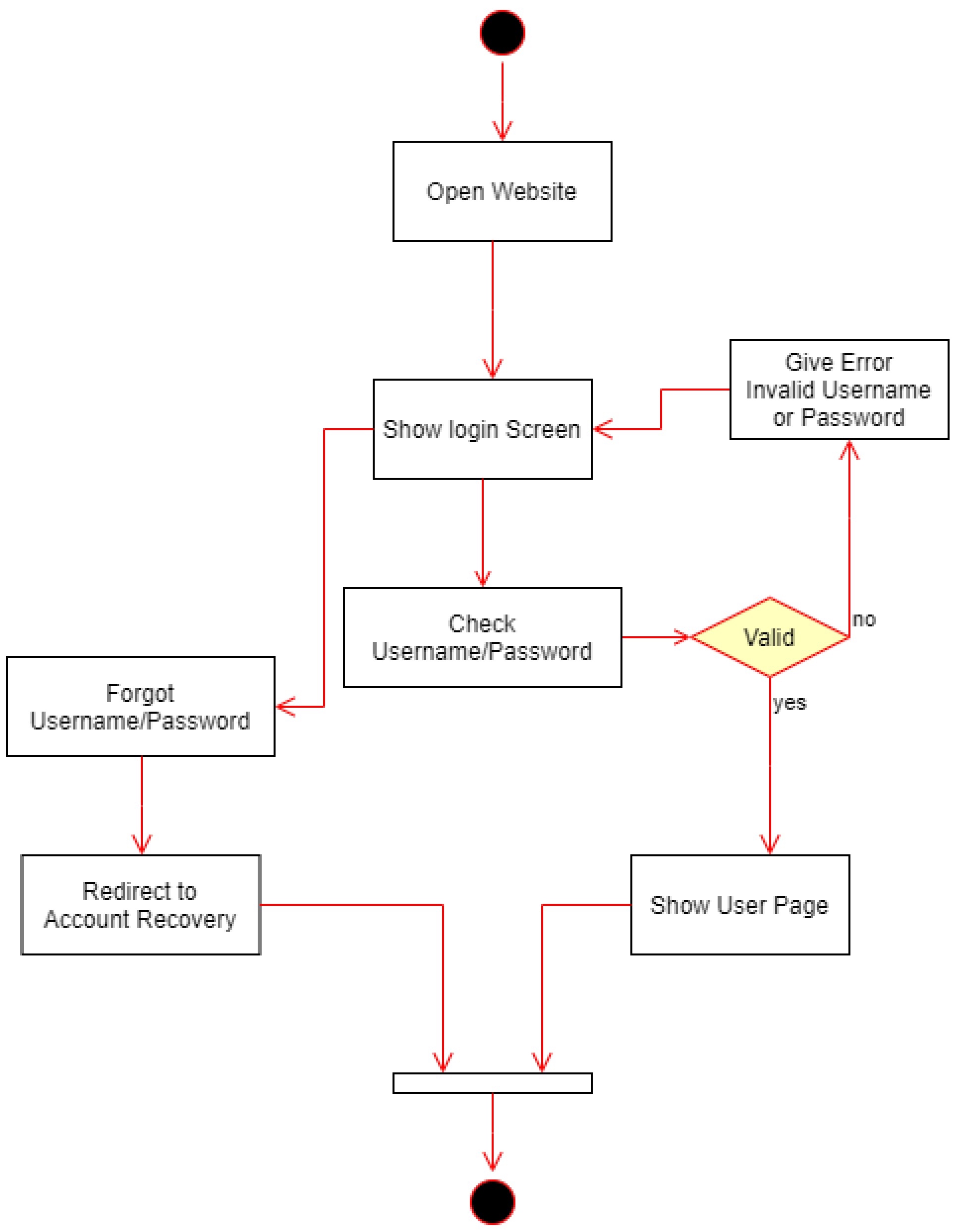

Client:
Server:
Database:
View Specific "Place"
Click "Add Review"
Show "Add Review" Page
Submit Review
Save Review data
Return success/fail data, Update current reviews
Return success/fail statement, Update current reviews shown On fail offer "re-Submit"
Click "Edit Review" On own Review
Show "Edit Review" Page
Submit Revision
Save Revision data
Return success/fail data, Update current reviews
Return success/fail statement, Update current reviews shown On fail offer "re-Submit"
Click and Confirm "Delete Review" On own Review
Request Deletion
Return success/fail data, Update current reviews
Return success/fail statement, Update current reviews shown On fail offer "re-Submit"


Client:
Server:
Database:
View Review or Comment
Click "Like/Dislike"
Update Like/Dislike data
Return success/fail data, Update current Like value
Return success/fail statement, Update current Comments shown On fail offer "re-Submit"
Click "Add Comment"
Show "Add Comment" Page
Submit Comment
Save Comment data
Return success/fail data, Update current Comments
Return success/fail statement, Update current Comments shown On fail offer "re-Submit"
Click "Edit Comment" On own Comment
Show "Edit Comment" Page
Submit Revision
Save Revision data
Return success/fail data, Update current Comments
Return success/fail statement, Update current Comments shown On fail offer "re-Submit"
Click and Confirm "Delete Comment" On own Comment
Request Deletion
Return success/fail data, Update current Comments
Return success/fail statement, Update current Comments shown On fail offer "re-Submit"

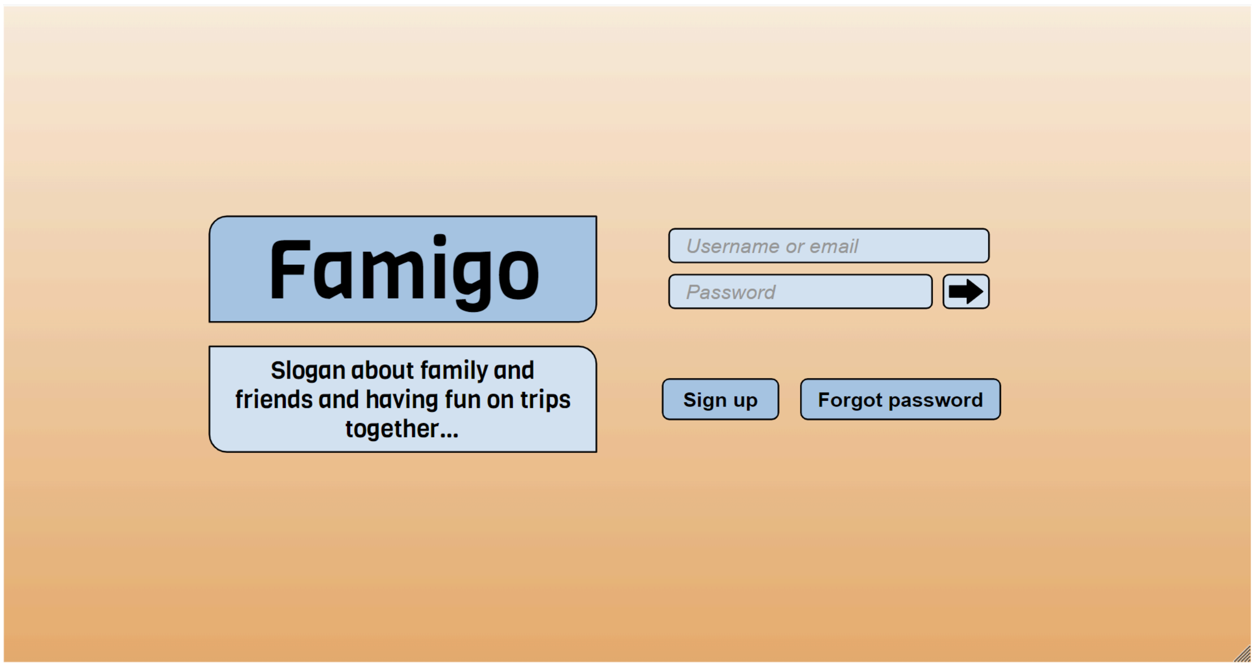

Client
Server
Database
Other Clients
User opens the direct messages menu
User selects other user from menu
User opens another user's profile
User presses message button
Request message history between users
User clicks new message notification
Message history
Display message history on screen
User types message and presses send
Save message to message history
Show confirmation if message sent successfully
Notify recipient
Show error if message couldn't be sent
User types message and doesn't send it
Save message as draft, only visible to current user


Client
Server
Database
views profile
submits form to edit username
commits new username
returns success or failure
returns success or failure
submits form with previous password in order to get access to edit the new password
returns failure if the password is wrong
if the password is right, submit the new password
commits new password
returns success or failure
returns success or failure
submits form to edit email
commits new email
returns success or failure
returns success or failure
submits form to edit name
commits new name
returns success or failure
returns success or failure
clicks to view stats
queries stats (likes, dislikes, reviews, comments, ratings)
returns success or failure
return retrieved stats
clicks to view followers
queries all visible followers of user
returns the visible followers
returns visible followers
clicks to view followers
queries all users the user follows
returns everyone user follows
returns all those being followed
clicks to view travel groups
queries all travel groups user belongs to
returns user's travel groups
returns user's travel groups
clicks to view trips
queries all trips user belongs to
returns all user's trips
returns all user's trips
clicks to change visibility
commits new visibility setting
returns success or failure
returns success or failure
clicks to change description
commits new description
returns success or failure
returns success or failure

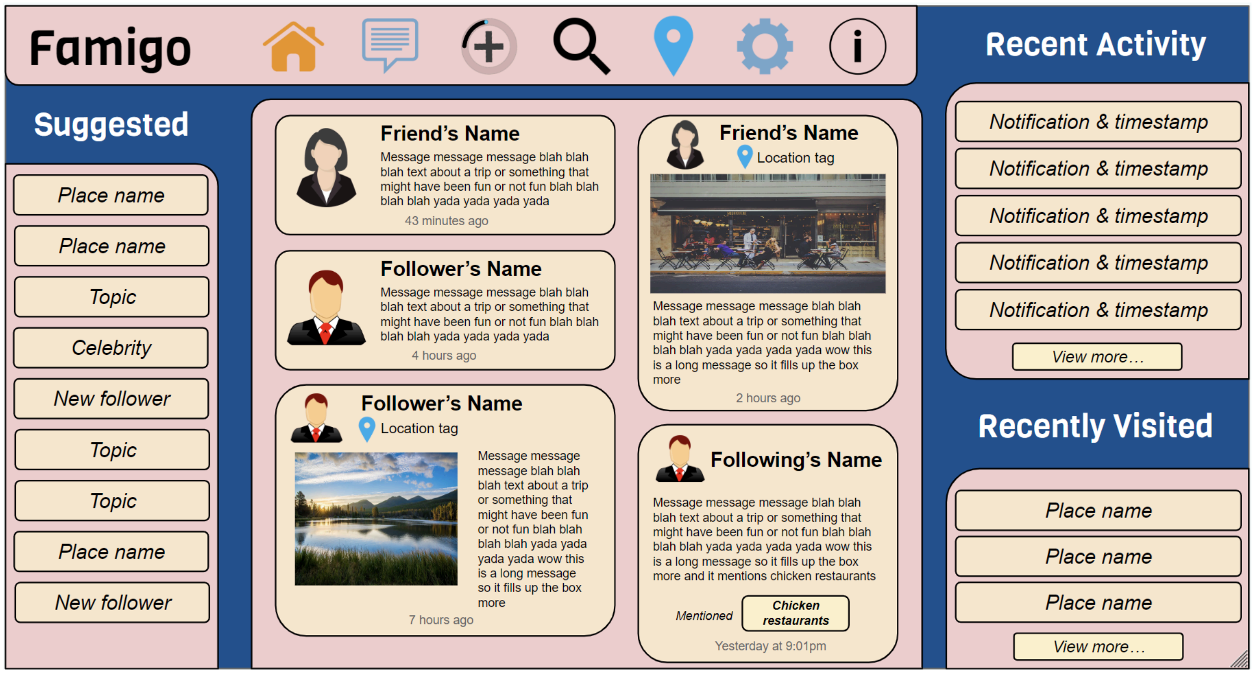

Search a place
Create new review
Select a review
[Comments are enabled]
[Comments are disabled]
Write a comment
Post successfully created


Open Website
Show login Screen
Give Error Invalid Username or Password
Check Username/Password
Valid
no
yes
Forgot Username/Password
Redirect to Account Recovery
Show User Page

Landing Page UI Concept


Famigo
Slogan about family and friends and having fun on trips together…
Username or email
Password
Sign up
Forgot password

Home Feed UI Concept



Friend's Name
Message message message blah blah blah text about a trip or something that might have been fun or not fun blah blah blah blah yada yada yada yada
43 minutes ago

Follower's Name
Message message message blah blah blah text about a trip or something that might have been fun or not fun blah blah blah blah yada yada yada yada
4 hours ago

Follower's Name
Location tag
Message message message blah blah blah text about a trip or something that might have been fun or not fun blah blah blah blah yada yada yada yada wow this is a long message so it fills up the box more
7 hours ago

Friend's Name
Location tag
Message message message blah blah blah text about a trip or something that might have been fun or not fun blah blah blah blah yada yada yada yada wow this is a long message so it fills up the box more
2 hours ago

Following's Name
Message message message blah blah blah text about a trip or something that might have been fun or not fun blah blah blah blah yada yada yada yada wow this is a long message so it fills up the box more and it mentions chicken restaurants
Mentioned Chicken restaurants
Yesterday at 9:01pm