SW Engineering CSC648/848 Summer 2021
July 8, 2021

Globetrotter Travel Assistant

Milestone 2

Team 2

Role	Name	Email
Team Lead/Github Master	Taylor Artunian	tartunian@mail.sfsu.edu
Front-End Lead	Kajeme Cheneque	
Back-End Lead	Luis Alfaro	
Front End	Ruja Rajbhandari	
	Yi Liu	
	Jesus Correa	
	Robin Fernando	

Revision History

Version	Date Submitted	Date Revised
M2V2		
M2V1	07/08/21	

Table of Contents

Table of Contents **2**

Data Definitions **4**

Prioritized Functional Requirements **5**

Priority 1 5

Registration 5

Login 5

Account Management 5

In-Progress Trips 5

Route Planner 6

Adding Services 6

Adding Guests 6

Checkout 6

Purchased Trips 7

UI Mockups & Storyboards **8**

Planning a Vacation Schedule 8

Inviting Guests on a Trip 9

Business Trip 10

Activity Itinerary 11

Login/Sign-Up Process 14

Social Media 15

Photo Gallery 16

High Level Database Architecture & Organization **18**

Business Rules for Photo Albums 18

User 18

File 18

Album 18

Photo Album 18

Photo 18

Entity Relationship Diagram (Photo Albums) 19

High Level APIs & Main Algorithms **20**

API Endpoints 20

Algorithms 21

High Level UML Diagrams **22**

Data Model 22

High Level Application Network & Deployment Diagrams **23**
Network Diagram 23
Deployment Diagram 24
Request Processing Sequence 25

Key Project Risks **26**
Skill Risks 26
Schedule Risks 26
Teamwork Risk 26
Technical Risks 26
Legal Risks 27

Project Management Strategy **28**
Present 28
Future 28

Team Contributions **29**
Jesus 29
Ruza 29
Yi 29
Robin 29
Luis 29
KJ 29
Taylor 29

Data Definitions

1. **Datetime** - A combination of a date and a time.
2. **Timespan** - A combination of a *start* **Datetime** and an *end* **Datetime**.
3. **Trip** - A **Trip** is the container that holds the **Activities** that a **User** has grouped together and are generally close together temporally. It has one **Owner** but may have many **Guests**.
4. **Location** - A **Location** is a specific place that is represented as either an address or GPS coordinate.
    a. Hotel/Restaurant: GPS coordinate and/or an address
    b. Airports: IATA code (i.e. LAX, SFO, SJO)
5. **Activity** - An activity represents things that a traveler can do. It occurs at a *start* **Location**, but also has an optional *destination* **Location**, during a **Timespan** and may have an associated **Service**. **Users** add **Activities** to their itinerary.
6. **Service** - A **Service** refers to a product that may be purchased from a **Service Provider** as part of an **Activity**. Examples include a hotel or restaurant reservation, or an airline ticket.
7. **Service Provider** - A service provider is a third party that a user may purchase **Services** from. Examples include a hotel, a restaurant, or an airline.
8. **Service Record** - A **Service Record** refers to the details obtained from a **Service Provider** regarding the purchased **Service**. Examples include reservation confirmation numbers, hotel room numbers, receipts, etc.
9. **User** - A **User** is anyone that uses the Globetrotter app.
10. **Unregistered User** - A **User** that has not registered an account and cannot save or checkout their in progress trips.
11. **Registered User** - This is a User that has created an account with Globetrotter.
12. **Trip Owner** - This is the **User** that created the **Trip**. They have full control over the **Trip**. They are allowed to invite **Guests** to the **Trip**.
13. **Guests** - These are **Users** that have been invited by either the trip **Owner** or other **Guests**. If the **Guest** is unregistered they will be required to register.
14. **Photo Album** - A container for showcasing photos. **Users** with edit permissions to the album may upload photos to that album. A **Photo Album** is private at first but the owner of the Album may add **Registered Users** and give them either view or edit permissions.
15. **Review** - Users are allowed to leave **Reviews** of **Service Providers** from which they have purchased **Services**. A **Review** contains an overall rating from 0-100 and a message.

Prioritized Functional Requirements

Priority 1

Registration

1. Unregistered users will be able to create an account.
2. Unregistered users shall choose a username when creating an account.
3. Unregistered users shall choose a password when creating an account.
4. Unregistered users shall provide their email when creating an account.
5. Users will be sent a confirmation email after completing registration.

Login

6. Registered users can login using the ‘Login’ page.
7. ‘Login’ page will have a ‘forgot password?’ button.
8. Registered users can reset their password from the ‘Login’ page by sending a reset link to the account’s email.
9. ‘Login’ page will have a ‘forgot password?’ button.
10. Registered users can reset their password from the ‘Login’ page by sending a reset link to the account’s email.

Account Management

11. Registered users can access the ‘Account Management’ page.
12. Registered users can change their password.
13. Registered users can change their email.

In-Progress Trips

14. Registered users may view trips that they previously created.
15. Registered users may edit trips that they previously created.
16. Registered users may delete trips that they previously created.
17. Registered users may view trips to which they were added as a guest.
18. All guests may view trips on which they are guests.
19. Unregistered users may not edit trips on which they are guests.
20. Registered users may upload pictures of their trip.
21. Registered users may delete pictures they uploaded.
22. Registered users may add a photo description when uploading a picture.
23. Registered users may edit their photo descriptions.
24. Registered users shall give a title when uploading a picture.
25. Registered users may change titles of pictures they uploaded.
26. Registered users may delete their photo descriptions.

Route Planner

27. All users can change the displayed currency.
28. The displayed currency will default to the preferred currency of a logged in user.
29. All registered guests can add at least one starting location.
30. All registered guests can add at least one destination.
31. All registered guests can add multiple waypoints.
32. All registered guests can edit the starting location of a trip.
33. All registered guests can remove waypoints.
34. All registered guests can change the starting location of a trip.
35. All registered guests can change the destination of a trip.
36. All registered guests can remove the starting location of a trip.
37. All registered guests can remove the destination of a trip.
38. All registered guests shall specify the dates of departure.
39. All users shall specify a budget.
40. All registered guests can add guests to the trip.
41. All users will be able to view weather information for each location.
42. Registered users can navigate to the ‘Account Management’ page.
43. Dates added to a trip may not overlap.

Adding Services

44. All users may add hotel reservations to a particular day of a trip.
45. All users may add entertainment activities to a particular day of a trip.
46. All users may add transportation reservations to a particular day of a trip.
47. All users can search for flights.
48. All users can choose from the available flights.
49. All users will be able to view arrival times for the available flights.

Adding Guests

50. All registered guests may add other registered users to the trip.
51. All registered guests may add unregistered guests to the trip.
52. All guests added to the trip will be sent an email containing a link to the trip.
53. All registered guests can only add up to one registered user at a time.
54. All registered guests can only add up to nine registered users to the trip.

Checkout

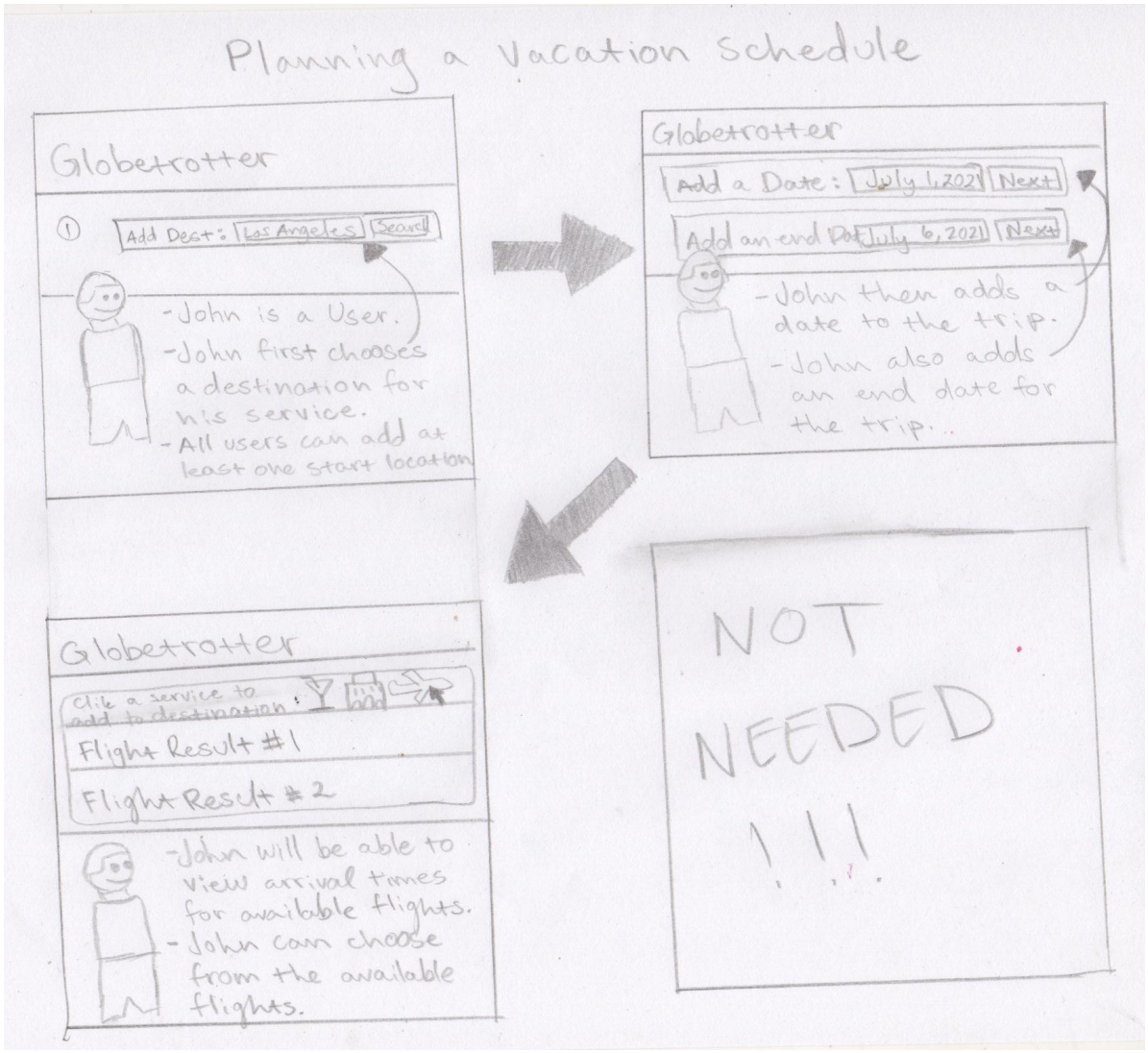
55. Registered users may purchase their ‘In-Progress’ trips.
56. Registered guests may purchase trips on which they are guests.
57. Unregistered guests may not purchase trips on which they are guests.
58. Registered users will receive a confirmation email after checkout.
59. Registered guests will receive a confirmation email after checkout.

Purchased Trips

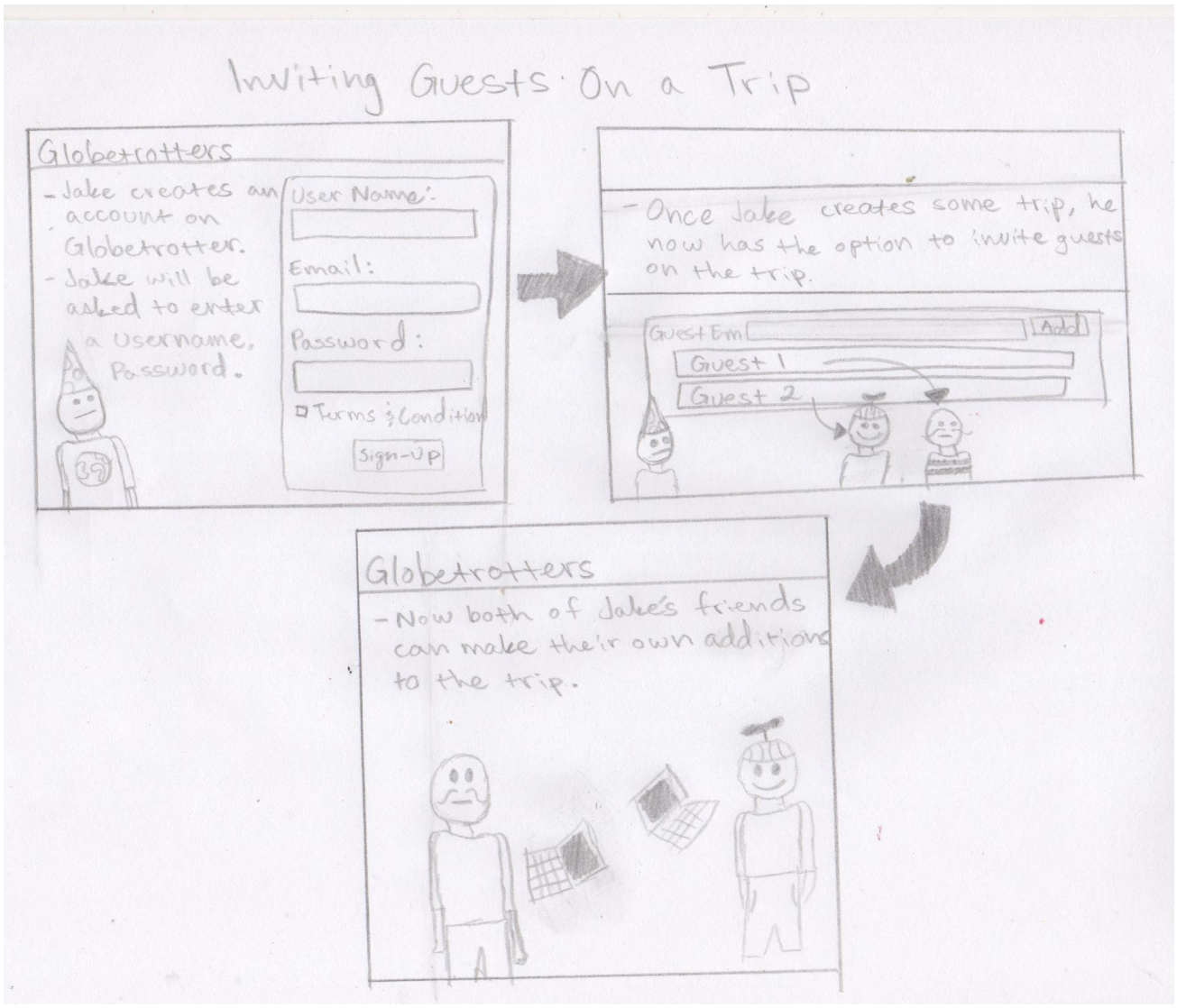
60. Registered users may cancel flights which they purchased.
61. Registered users can be issued a refund for flights cancelled within 48 hours of departure.
62. Registered users can receive credit for flights cancelled within 48 hours of departure.
63. Unregistered users shall be directed to the ‘Registration’ page.

UI Mockups & Storyboards

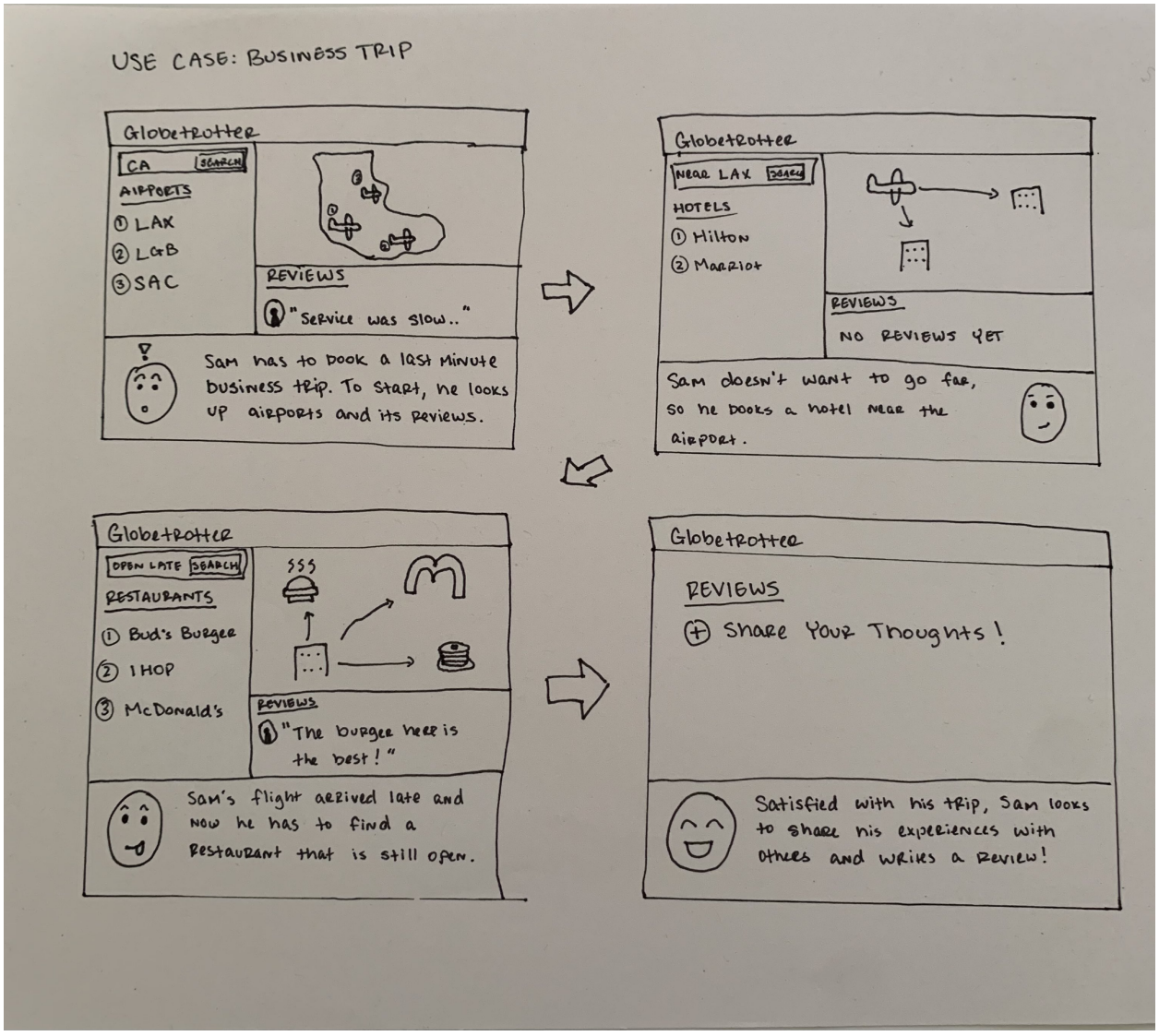
Planning a Vacation Schedule



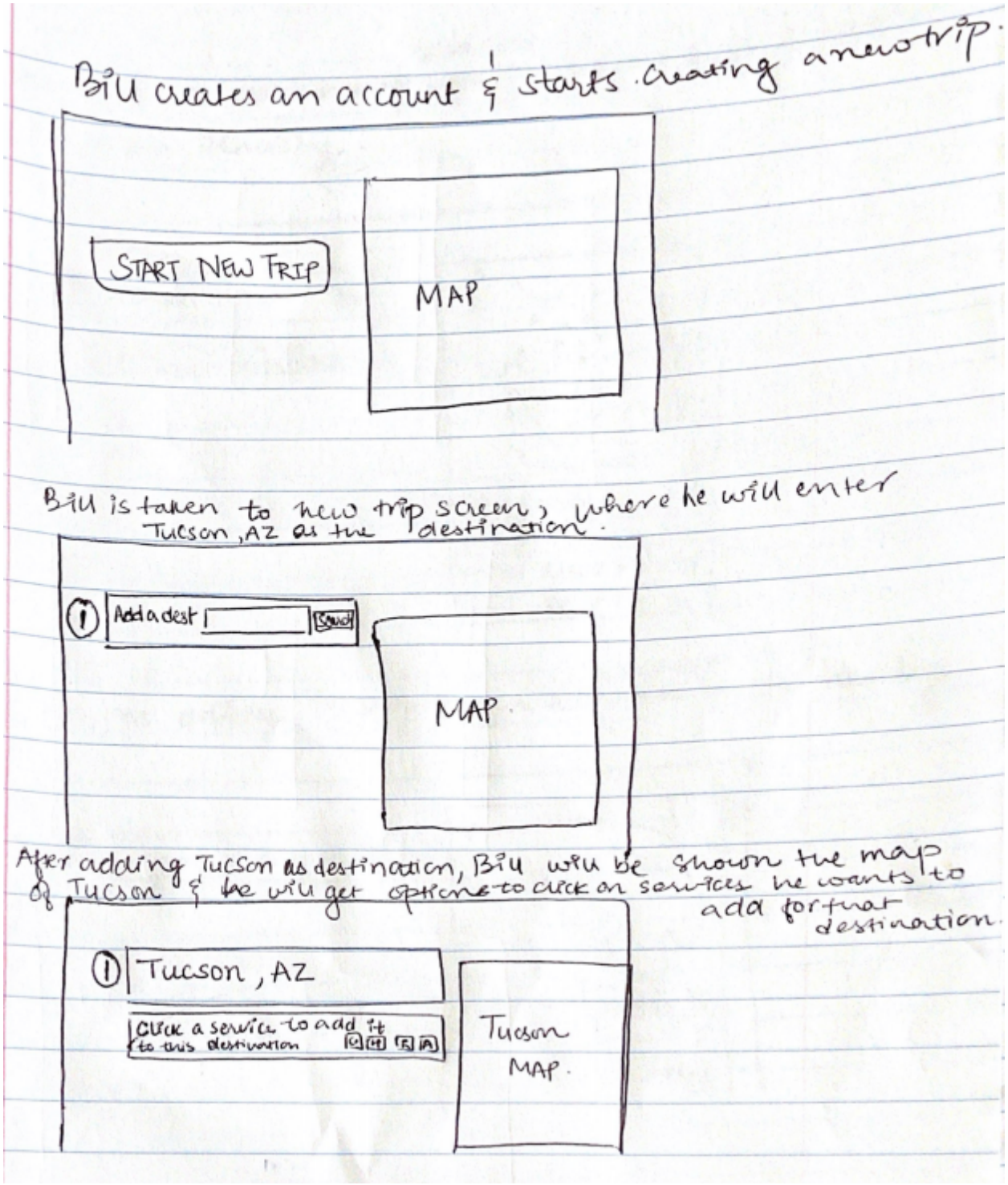
Inviting Guests on a Trip



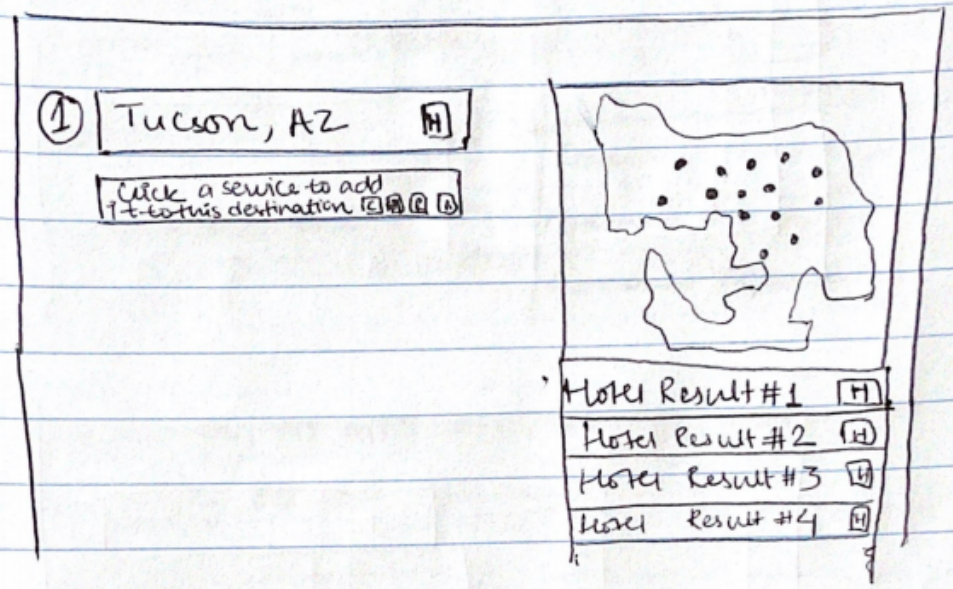
Business Trip



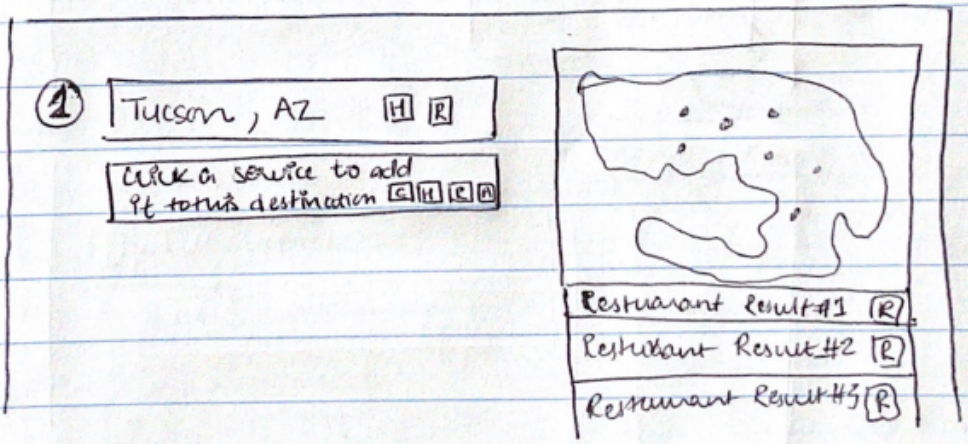
Activity Itinerary



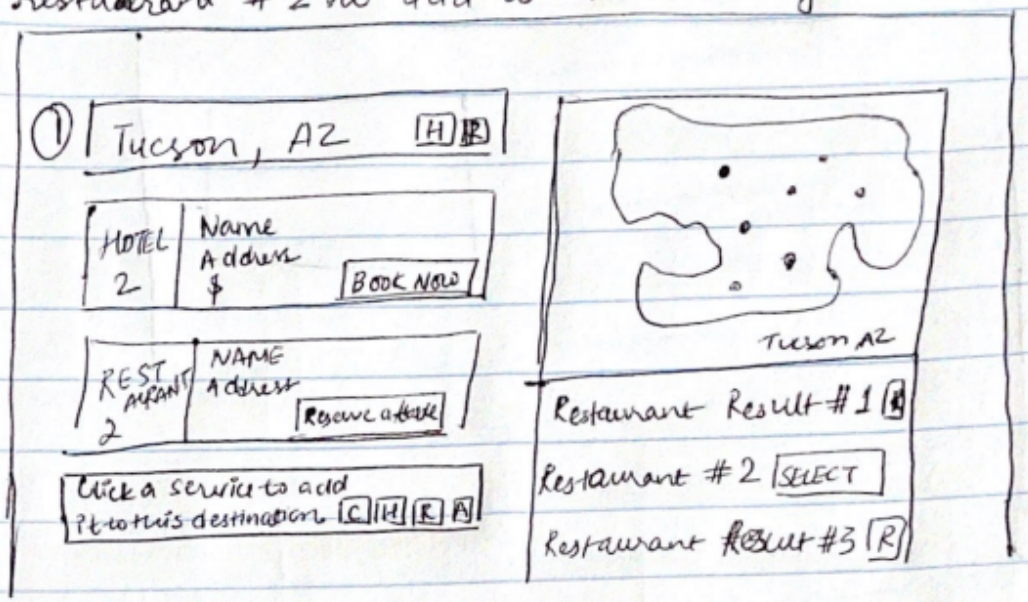
Bill decides to ~~cookie his hoo~~ ~~book~~ search for hotels first & add it to his itinerary.



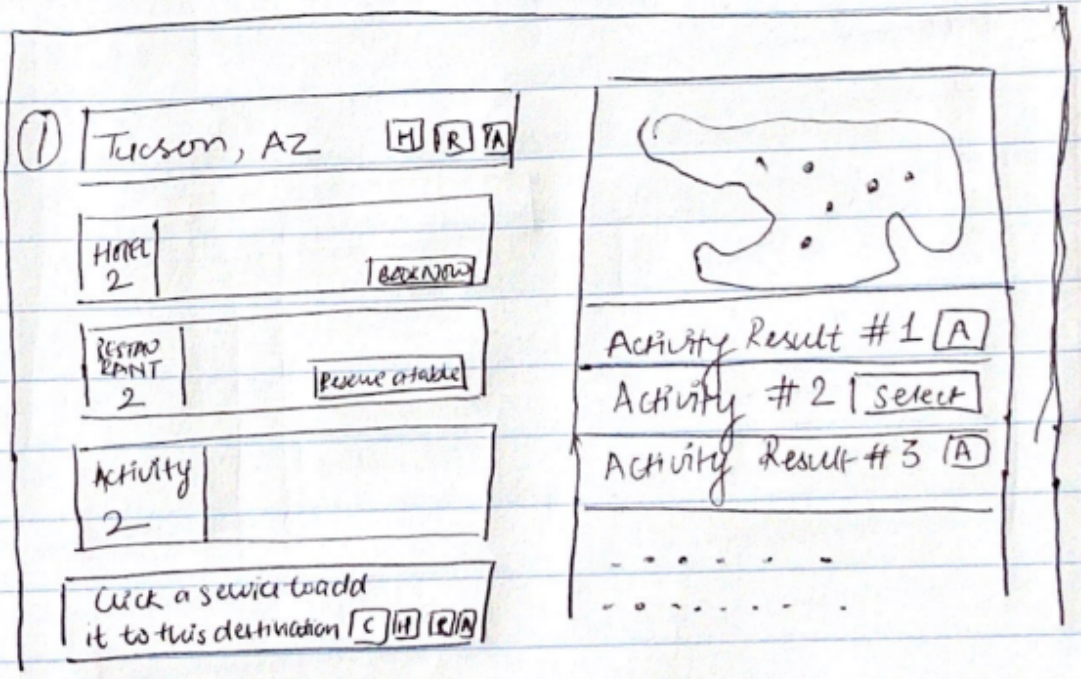
Then he decides to search for ~~restaurants~~ nearby



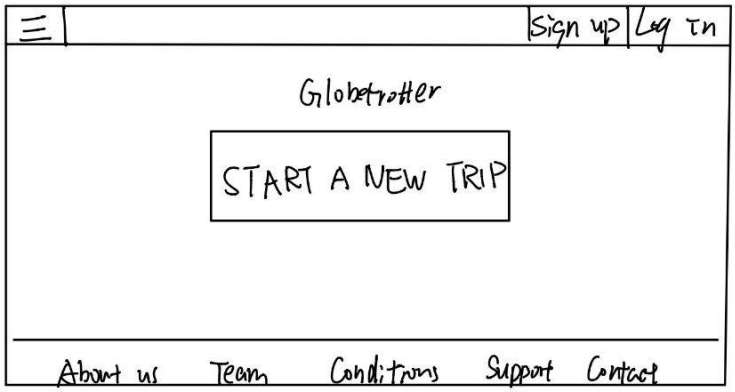
Bill decides to look for restaurants nearby & selects restaurant # 2 to add to his itinerary.



Bill wants to add an activity and looks up activities to do nearby.



Login/Sign-Up Process



Main Page

Sign up Page

Creating Account

First name
Last name
Address :
State :
Zip :
County:

Account number:
password:
verify password:

email address :

☐ show password

☐ I agree with terms and conditions..

Create account

Log in Page

Log in

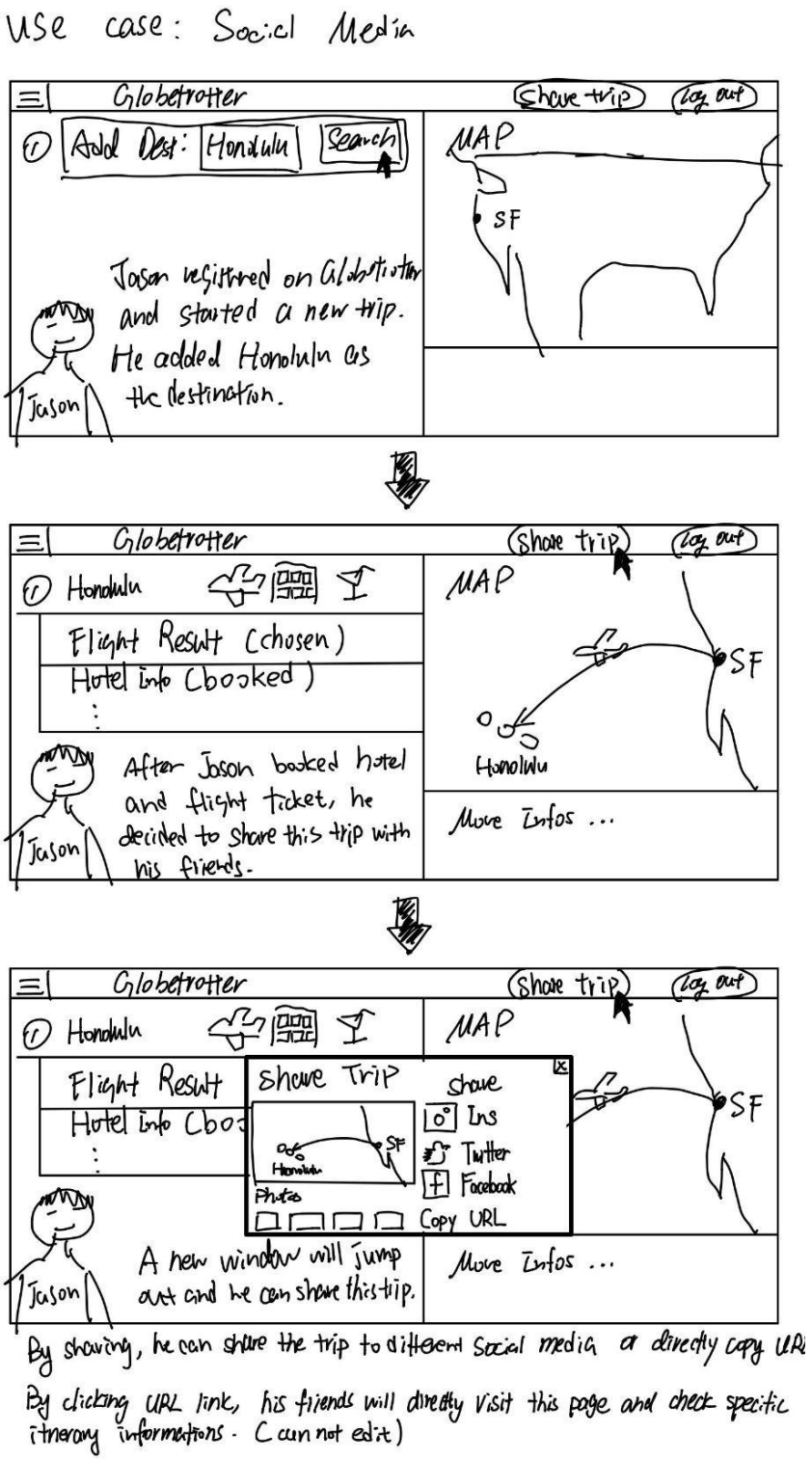
Account num :
Password :

Log in

forgot password ?

Don't have an account? create

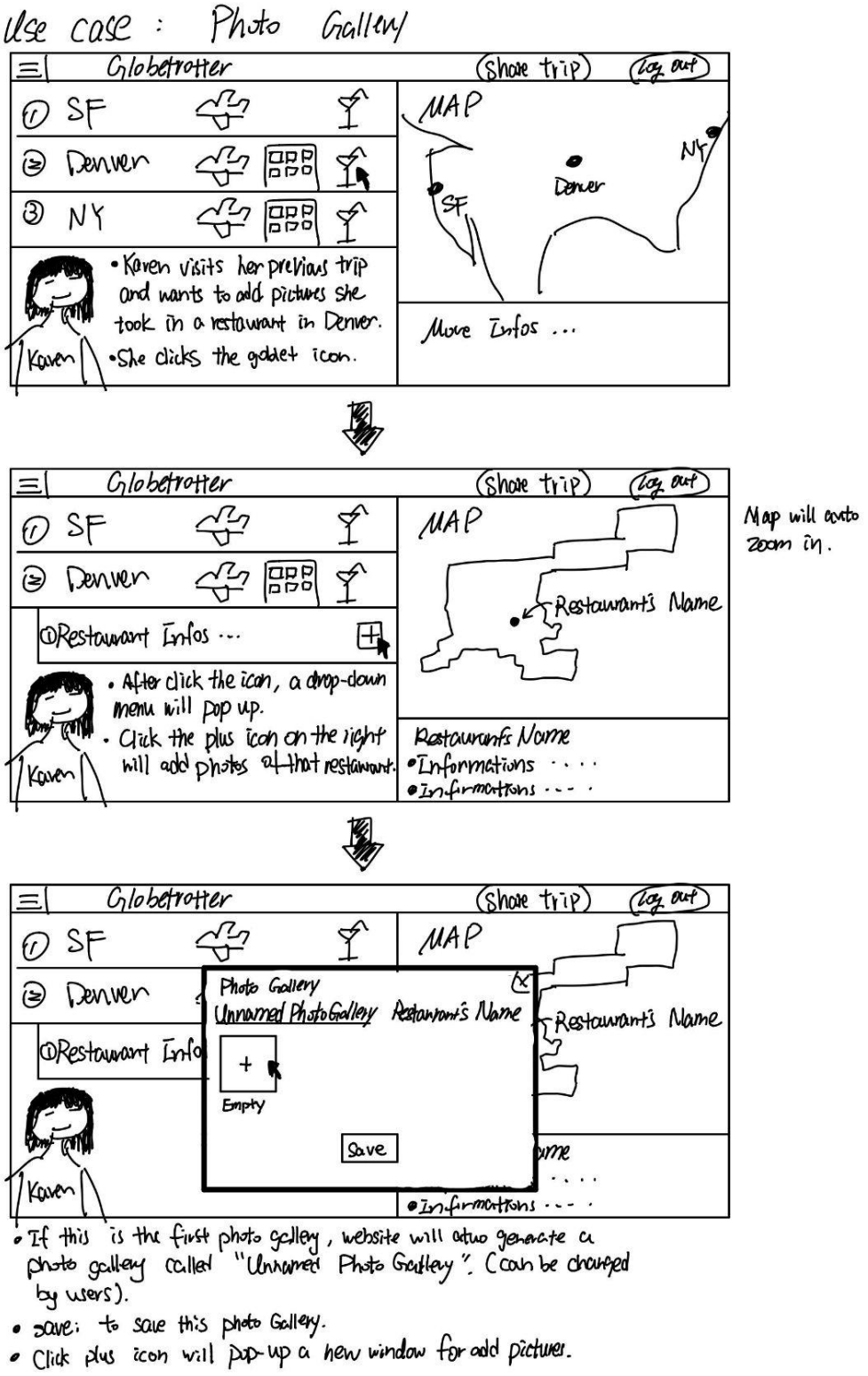
Social Media

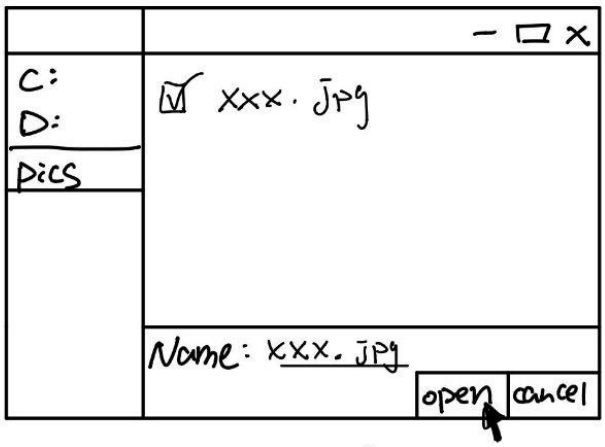


By sharing, he can share the trip to different social media or directly copy URL

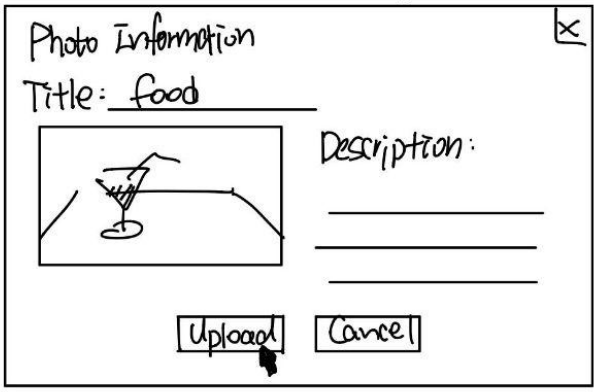
By clicking URL link, his friends will directly visit this page and check specific itinerary informations. (Can not edit)

Photo Gallery

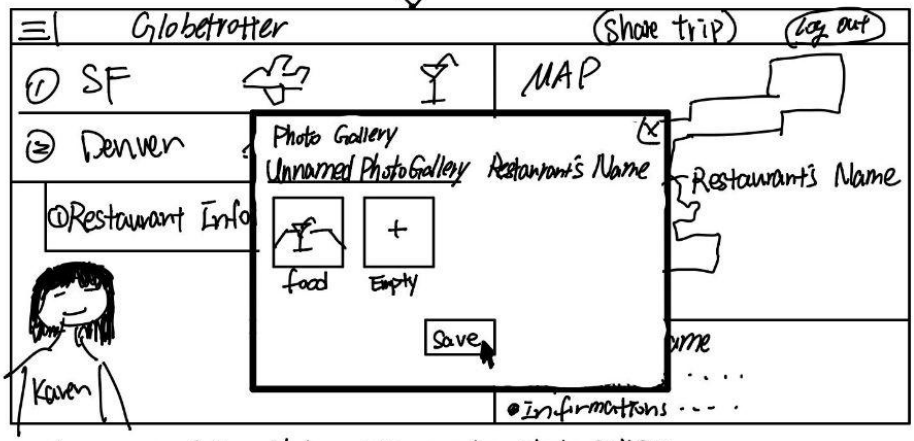




- Then Karen can browse her PC folders and choose picture to upload.



- If file opened successfully, this window will show up on her website.
- Click upload will upload this picture and information to the photo gallery.



- Karen successfully added a picture in her photo gallery.
- Click save icon will save the whole photo gallery

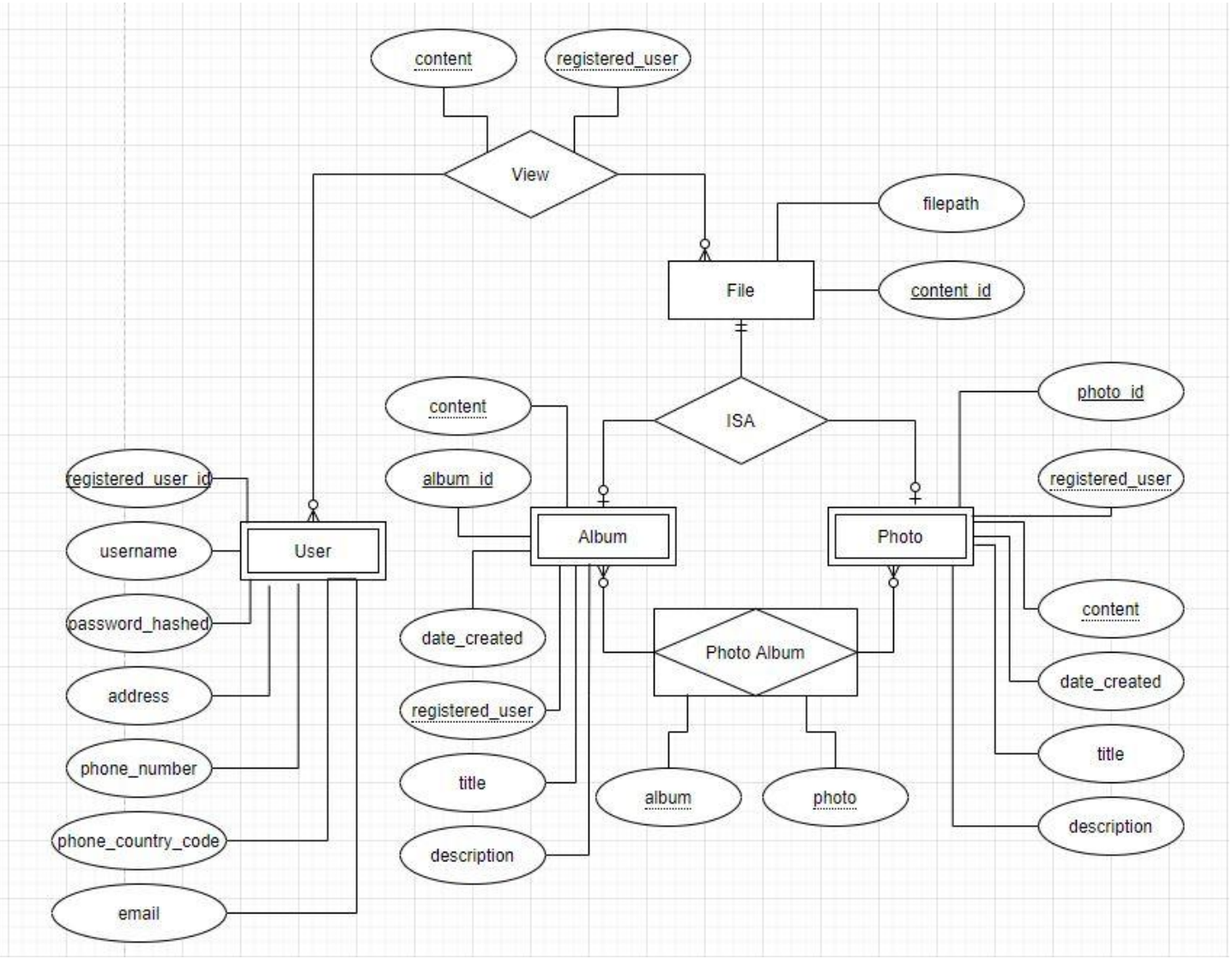
High Level Database Architecture & Organization

Business Rules for Photo Albums

Entities, Attribute, Relation, Quantitative

1. User
    a. A User shall be able to view many user_content
2. File
    a. A File shall either be a Photo or an Album.
    b. A File shall be viewed by many Registered Users
3. Album
    a. An Album shall Photo_Album many Photos
4. Photo Album
    a. A Photo Album shall have many photos.
    b. A Photo Album shall be viewed by many users.
5. Photo
    a. A Photo shall be Photo_Albumed by many albums.
    b. A Photo shall have one name.
    c. A Photo shall have one description.

Entity Relationship Diagram (Photo Albums)



High Level APIs & Main Algorithms

API Endpoints

Note: All endpoints begin with /api. Example: the authenticate endpoint is http://server:3000/api/authenticate.

Endpoint Name	HTTP Method	Required Parameters	Required Roles	Purpose	Implemented
/authenticate	POST	• username • password		Returns the user object and token if successful.	✓
The endpoints below require an additional 'Bearer' authorization header, which the server uses to identify the requester, know that they have been authenticated, and check whether they have permission to access the endpoint.					
/users	GET	• None.		Returns all users.	✓
	PUT	• username • email • password		Creating and updating Users.	✓
/users/search	GET	• username or email		Returns users with a username or email that partially matches the parameter sent.	✓
/users/me	GET	• None.		Returns the user object of the requester.	✓
/users/trips	GET	• userId		Returns a list of tripId's for a given User.	✗
/trips	GET	• tripId		Returns the details of a single Trip.	✗
	PUT	• tripObject?		Modify, Update Trip	✗
	DELETE	• tripId		Remove User's Trip	✗
/trips/guests	GET	• tripId		Return guests from a Trip	✗
	PUT	• tripId • userId • allowEdit		Modify guests from a Trip	✗
	DELETE	• tripId • userId		Remove guests from a Trip.	✗
/reviews	GET	• serviceProviderId		Returns a list of reviews for a Service Provider.	✗
	PUT	• reviewObject?		Modify, Update reviews on a service from a Trip.	✗

	DELETE	• reviewId		Remove a review from a Trip.	✗
/flights/search	GET	• originLocationCode • destinationLocationCode • departureDate • adults • currencyCode • max		Retrieve a list of flights from the Amadeus 'Flight Offer Search' API.	✓
/flights/book	POST	• flightOfferObject?		Return confirmation of flight booking.	✗

Algorithms

- The app will mainly use sorting algorithms to present the data. These will appear in several places
    - Viewing flights in order of price
    - Viewing flights

-When the user starts a search, it will either be a valid or invalid search.

-If User searches for a non-existent Flight, Restaurant, Activity, Trip, then the backend will send a message saying that the search is not valid to the fronted UI.

-If the User searches for a Flight from a valid airport and to a valid airport(destination), then the backend will send a list of airlines displaying the following departures of flights.

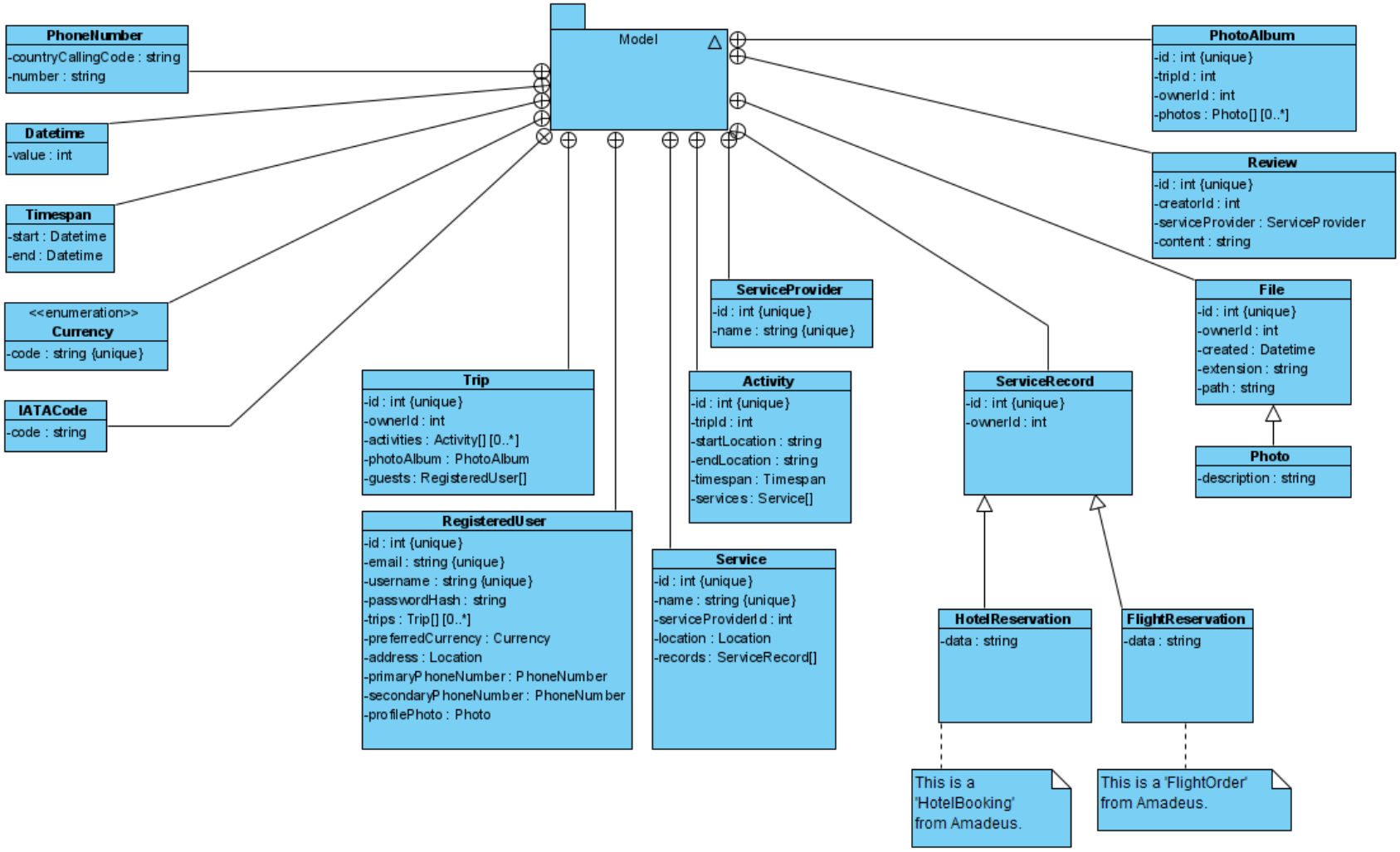
-If the User searches for a Trip that was created previously, the backend will show a Trip that matches the searched input given by the User.

-If User searches for activities around a certain region, the backend will display a list of things available for the User to view on the frontend UI.

-If the User searches for a service provider such as a restaurant, the backend will send a list of restaurants available in the region of interest to the fronted UI for the User to view.

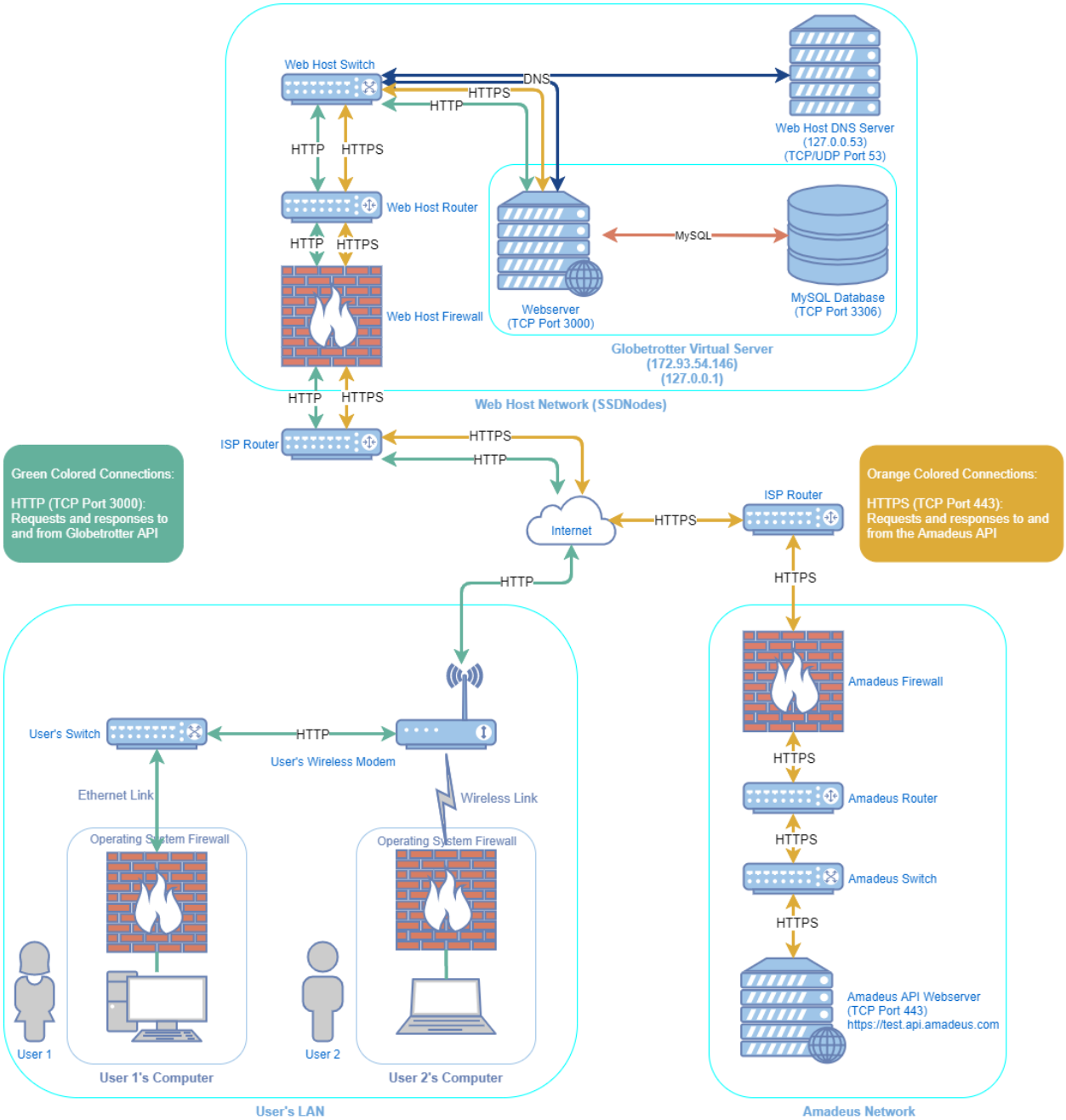
High Level UML Diagrams

Data Model

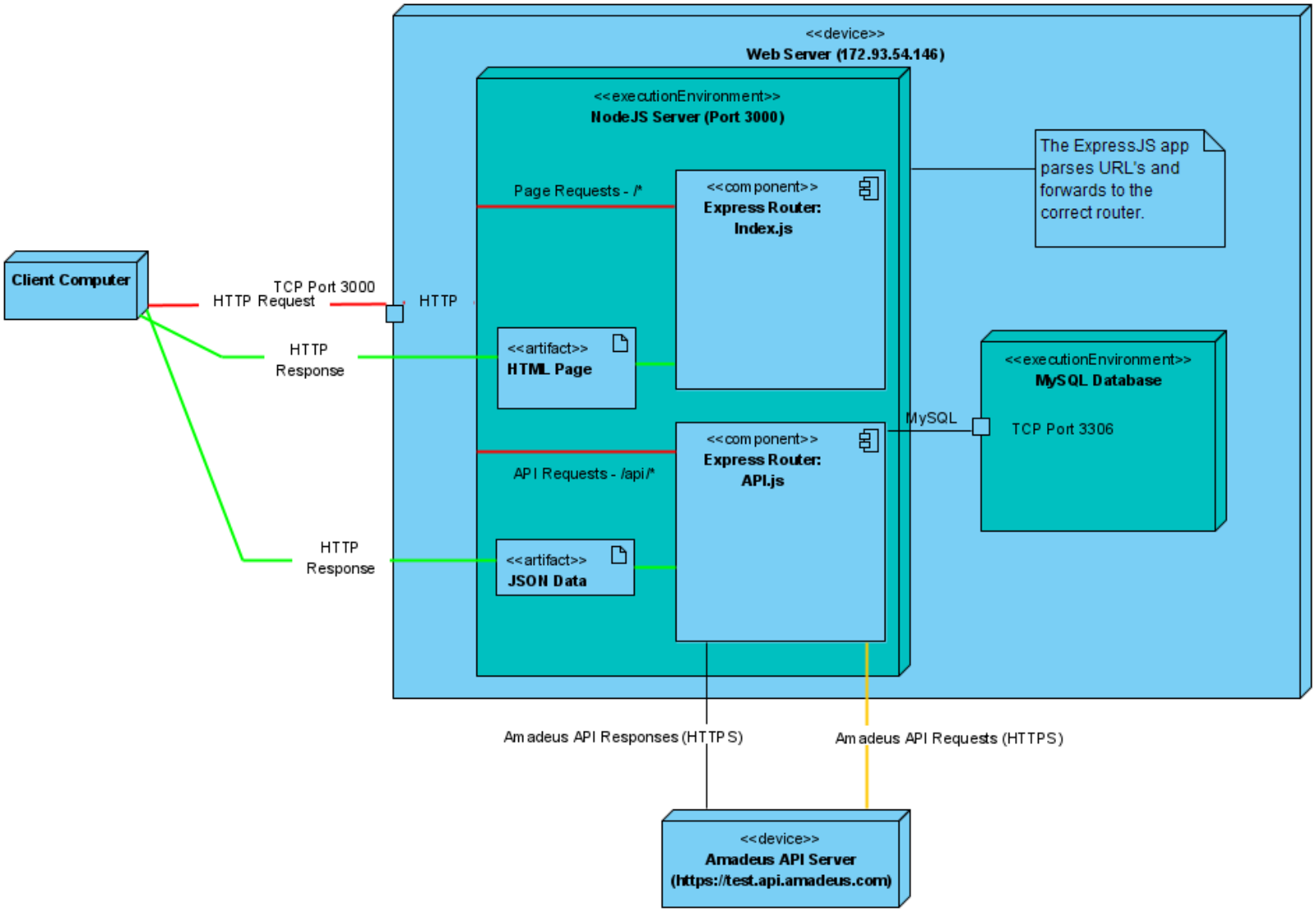


High Level Application Network & Deployment Diagrams

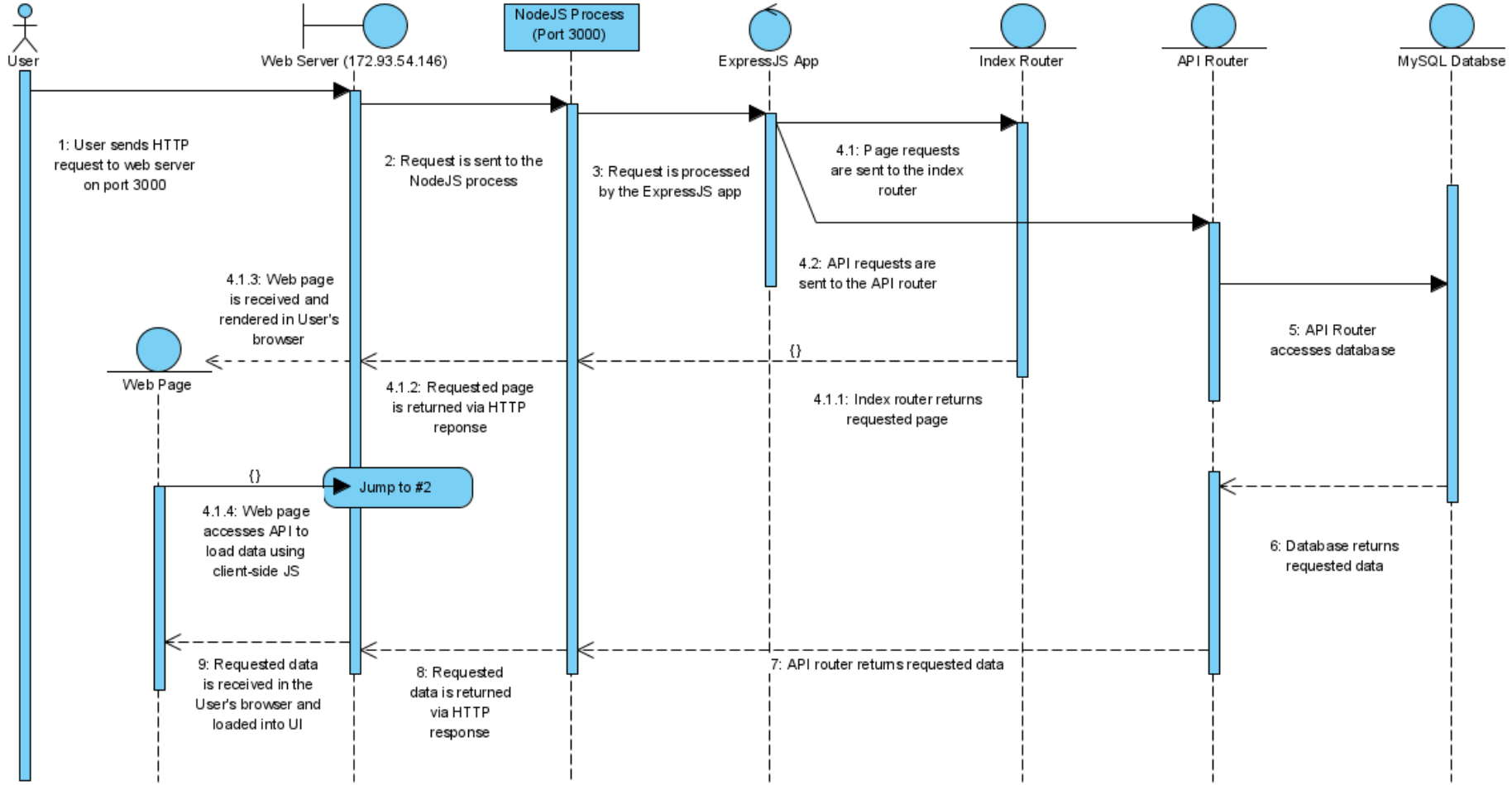
Network Diagram



Deployment Diagram



Request Processing Sequence



Key Project Risks

Skill Risks

- Not everyone is familiar with the software stack.
    - Solutions:
        - Provide an overview of the software stack.
        - Direct team members to online tutorials.
        - Recommend books that focus around the software stack.

Schedule Risks

- Implementing a checkout feature may take more time than the other parts of the system.
    - Solutions:
        - Focus more on other features while still trying to figure out a way to implement a checkout feature. Making one feature, such as flight planning and booking, very strong will make it easier to add other features in the future.
- Integrating hotels as a usable Service may take more time than we anticipated
    - Solutions:
        - Same as the checkout feature. If we focus on making Flights work well as a Service available to users, it may be easier to integrate hotel reservations in the future

Teamwork Risk

- Uneven distribution of front/back end team members
    - Solutions:
        - Attempt to distribute tasks as evenly as possible.
        - Offer assistance on tasks that team members might be behind on.
        - Reach out for help if tasks assigned are running behind schedule.

Technical Risks

- Need to develop access control for the API endpoints. Some of the endpoints should only be accessed by an Admin level user.
    - Solutions:
        - We can implement a Role column on each user in the database with an option of either Admin or User. The Role of a user can be checked in each API call.
- Cannot yet deserialize JSON objects returned from the database into typed-objects with methods.

    - Solutions:
        - Research JavaScript classes and prototypes. Creating prototypes or classes from the JSON data will allow us to have better control over the data.
- Overall, keeping track of all the data and parameters that go into and come out of the external API's is tricky.
    - Solution:
        - Same as above. We need to figure out how to create our UML data model in JavaScript using classes or prototypes. We will focus on making our data model more robust and learn how to model it in code.
- We are in need of a mapping and geocoding API for the front end.
    - Solution:
        - Team lead will assign members of the front end to experiment with mapping API's.

Legal Risks

- None at this time.

Project Management Strategy

Present

- We had more frequent meetings.
- We had separate and more detailed meetings for the front and back end teams.
- We began using Trello to divide and assign tasks to team members.
- Deadlines were set on tasks within Trello.
- We reviewed individual tasks and provided constructive feedback as a team.

Future

- Team lead will assign tasks with clearer objectives and more specific due dates.

Team Contributions

Jesus

Created storyboards for ‘Planning a Vacation Schedule’ and ‘Inviting Guests on a Trip’, Created front end of Vertical Prototype, Contributed to Data Definitions

Ruza

Developed interactive UI Prototype, Created storyboard for ‘Activity Itinerary’

Yi

Created storyboards for ‘Login/Sign-Up Process’, ‘Social Media’, and ‘Photo Gallery’

Robin

Created storyboard for ‘Business Trip’

Luis

Created the business rules for Photo Gallery, ERD diagram, Coded parts of the back end for Vertical Prototype, Contributed to Data Definitions

KJ

Added to API Endpoints table and Algorithms section

Taylor

API Endpoints table, Network and Deployment Diagrams, UML Diagram, contributed to Data Definitions