Final Project for SW Engineering Class CSC648-848 Section 01 Summer 2021

Product Name: Globetrotter Travel Assistant

August 5, 2021

PREPARED FOR

Jose Ortiz

PREPARED BY TEAM 2

Taylor Artunian tartunian@mail.sfsu.edu
Jesus Correa
Luis Alfaro
Ruja Rajbhandari
Yi Liu yliu22@mail.sfsu.edu
Robin Fernando
Kajeme Cheneque

PRODUCT URL

http://172.93.54.146:3000/

Dear Jose Ortiz,

Please find enclosed our Milestone 5 document.

Yours Truly,

Team 2

1. Product Summary

Product Name: Globetrotter Travel Assistant

Product URL: http://172.93.54.146:3000/

At Globetrotter, we are aware that the internet has endless opportunities. That being said, we plan to use the internet as our main source of marketing. One way we plan to market is through Youtube. I'm sure we have all seen those ads that youtubers put into their videos. The beauty of those ads is that a viewer must watch a certain amount of the ad before they can skip it. Getting an ad, with information about our product, to play in between videos would ensure that people will see it. We could also make a deal with youtubers that have a huge audience and have them talk about our product at the beginning of their video. Millions of viewers, of all ages, would see their favorite content creators talk about or product. This is great for our product because youtubers with huge followings can easily persuade their fans.

Priority 1 Functional Requirements

Account Management

1. Registered users can edit their birthday.
2. Registered users can edit their email.
3. Registered users can edit their home city
4. Registered users can edit their home postal code.
5. Registered users can edit their home state
6. Registered users can edit their home street address.
7. Registered users can edit their password.
8. Registered users can edit their primary phone number.
9. Registered users can edit their profile photo.
10. Registered users can edit their secondary phone number.
11. Registered users can edit their username.
12. Registered users can view the ‘Account Management’ page.
13. Unregistered users cannot access Account Management.

Login

14. Registered users can login using the “Login” page.
15. Unregistered users can view the “Registration” page.
16. Users shall provide their password when logging in.
17. Users shall provide their username when logging in.

18. Registered users will be able to reset their password by providing their username, email, and new password.
19. The password will be obscured in the Login page.

Photo Album

20. Registered users can upload pictures of their trip.
21. Unregistered users cannot access Photo Albums.

Route Planner

22. All registered users can add at least one destination to their trip..
23. All registered users can add multiple Activities to a Trip.
24. All registered users can edit the destination of a trip.
25. All registered users can edit the starting location of a trip.
26. All registered users can remove the destination of a trip.
27. All registered users can remove the starting location of a trip.
28. All registered users can remove Activities from a Trip.
29. All registered users shall specify the dates of departure for each flight.
30. Unregistered users cannot access the Route Planner.

Saved Trips

31. Registered users can view trips that they have saved.
32. Unregistered users cannot access Saved Trips.

Checkout

33. Unregistered users cannot access Checkout.

Previous Trips

34. Unregistered users cannot access Previous Trips.

Registration

35. Unregistered Users shall provide their date of birth when creating an account
36. Unregistered users shall provide a password when creating an account.
37. Unregistered users shall provide a primary phone number when creating an account.
38. Unregistered users shall provide a username when creating an account.
39. Unregistered users shall provide their email when creating an account.
40. Unregistered users shall provide their first name when creating an account.
41. Unregistered users shall provide their home city when creating an account.

42. Unregistered users shall provide their home postal code when creating an account.
43. Unregistered users shall provide their home state when creating an account.
44. Unregistered users shall provide their home street address when creating an account.
45. Unregistered users shall provide their last name when creating an account.
46. Unregistered users will be able to create an account.
47. Unregistered users shall provide their gender when creating an account.

Meet the Team

48. All users can contact the team.
49. All users can view the “About Us” page to find out more information about the team.

Website

50. The website will be easy for all Users to maneuver through.

Product Uniqueness

Our product is unique in that it has a photo gallery feature. Based on our research, no website that allows users to book flights has a photo gallery feature. In our website, once a user has created and purchased a trip, they can go back and upload images to previous trips they have been on. This benefits the user because they have a place where they can store some of their favorite images that they took while on that specific trip. Some of our future plans for this feature include being able to invite guests that went on the trip with the owner, to the photo gallery so that both the owner and the guests can upload images. We decided to go with this unique feature because a lot of us can agree that memories are priceless and we wanted to give the users the option to save any memories they would like to keep forever.

2. Milestone Documents

Milestone 4 V2

PREPARED FOR

Jose Ortiz

PREPARED BY

Team 2

Globetrotter

JUL 30, 2021

SW Engineering CSC648/848 Summer 2021

July 30, 2021

Globetrotter Travel Assistant

Milestone 4

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
M4V1	07/30/21	08/02/21
M3V2	07/30/21	-
M3V1	07/22/21	07/30/21
M2V2	07/22/21	-
M2V1	07/08/21	07/22/21
M1V2	07/08/21	-
M1V1	06/22/21	07/08/21

Table of Contents

1. Product Summary

Product Name: Globetrotter Travel Assistant

At Globetrotter, we are aware that the internet has endless opportunities. That being said, we plan to use the internet as our main source of marketing. One way we plan to market is through Youtube. I'm sure we have all seen those ads that youtubers put into their videos. The beauty of those ads is that a viewer must watch a certain amount of the ad before they can skip it. Getting an ad, with information about our product, to play in between videos would ensure that people will see it. We could also make a deal with youtubers that have a huge audience and have them talk about our product at the beginning of their video. Millions of viewers, of all ages, would see their favorite content creators talk about or product. This is great for our product because youtubers with huge followings can easily persuade their fans.

Priority 1 Functional Requirements

Account Management

1. Registered users can edit their birthday.
2. Registered users can edit their email.
3. Registered users can edit their home city
4. Registered users can edit their home postal code.
5. Registered users can edit their home state
6. Registered users can edit their home street address.
7. Registered users can edit their password.
8. Registered users can edit their primary phone number.
9. Registered users can edit their profile photo.
10. Registered users can edit their secondary phone number.
11. Registered users can edit their username.
12. Registered users can view the ‘Account Management’ page.
13. Unregistered users cannot access Account Management.

Login

14. Registered users can login using the “Login” page.
15. Unregistered users can view the “Registration” page.
16. Users shall provide their password when logging in.
17. Users shall provide their username when logging in.
18. Registered users will be able to reset their password by providing their username, email, and new password.

19. The password will be obscured in the Login page.

Photo Album

20. Registered users can upload pictures of their trip.
21. Unregistered users cannot access Photo Albums.

Route Planner

22. All registered users can add at least one destination to their trip..
23. All registered users can add multiple Activities to a Trip.
24. All registered users can edit the destination of a trip.
25. All registered users can edit the starting location of a trip.
26. All registered users can remove the destination of a trip.
27. All registered users can remove the starting location of a trip.
28. All registered users can remove Activities from a Trip.
29. All registered users shall specify the dates of departure for each flight.
30. Unregistered users cannot access the Route Planner.

Saved Trips

31. Registered users can view trips that they have saved.
32. Unregistered users cannot access Saved Trips.

Checkout

33. Unregistered users cannot access Checkout.

Previous Trips

34. Unregistered users cannot access Previous Trips.

Registration

35. Unregistered Users shall provide their date of birth when creating an account
36. Unregistered users shall provide a password when creating an account.
37. Unregistered users shall provide a primary phone number when creating an account.
38. Unregistered users shall provide a username when creating an account.
39. Unregistered users shall provide their email when creating an account.
40. Unregistered users shall provide their first name when creating an account.
41. Unregistered users shall provide their home city when creating an account.
42. Unregistered users shall provide their home postal code when creating an account.
43. Unregistered users shall provide their home state when creating an account.

44. Unregistered users shall provide their home street address when creating an account.
45. Unregistered users shall provide their last name when creating an account.
46. Unregistered users will be able to create an account.
47. Unregistered users shall provide their gender when creating an account.

Meet the Team

48. All users can contact the team.
49. All users can view the “About Us” page to find out more information about the team.

Website

50. The website will be easy for all Users to maneuver through.

Our product is unique in that it has a photo gallery feature. Based on our research, no website that allows users to book flights has a photo gallery feature. In our website, once a user has created and purchased a trip, they can go back and upload images to previous trips they have been on. This benefits the user because they have a place where they can store some of their favorite images that they took while on that specific trip. Some of our future plans for this feature include being able to invite guests that went on the trip with the owner, to the photo gallery so that both the owner and the guests can upload images. We decided to go with this unique feature because a lot of us can agree that memories are priceless and we wanted to give the users the option to save any memories they would like to keep forever.

Visit our Home Page At: http://172.93.54.146:3000/

2. Usability Test Plan

Outline

Objectives

In these usability tests, our goal is to assess whether the various parts of our website are easy to find and use. The five parts or features that we have determined should be tested include: *searching for a flight*, *viewing a previous trip*, *reopening a saved trip*, *uploading an image*, and also *learning about the team members*. In checking the flight search feature, we want to ensure that users can easily find the input fields, that the airport autocomplete is easy to use, and that the flight results show enough information. With viewing a previous trip, we want to know that a user can find the Previous Trips page and find a specific trip that they have taken. From here we will test the usability of the Photo Gallery feature. The test will determine how easy it is to find the Photo Gallery for a particular trip as well as if there is any difficulty in uploading an image. Next, we will have the user reopen a saved trip. Here, we will test whether the user can find the Saved Trips page, view the details of the saved trip, and figure out how to reopen the saved trip. Lastly, we will assess how accessible the Meet the Team pages are, how easily a user can access them, and also find information on each team member.

Test Description

To perform the usability tests, there are not very many requirements. The user should use a desktop or laptop computer with internet access to perform the steps listed below. When performing the test, the user does not need to return home or start from a specific page in order to move on to the next test.

To begin the test users should navigate to our website at: http://172.93.54.146:3000/login

At the login page, users may use the built-in account
username: **globetrotter**
password: **globetrotter**

Testing Steps

Save a Trip

1. The user will first attempt to navigate to the Route Planner.
2. The user will attempt to input the airport from which they would like to depart and select from the autocomplete results.
3. The user will attempt to input the airport to which they would like to travel and select from the autocomplete results.
4. The user will attempt to select the desired date of departure.
5. The user will attempt to search for flights matching the parameters.
6. The user will wait for the flight results to load.
7. The user will view the results and decide which result is best for them.
8. The user may add additional activities using the 'Add Activity' button and repeat steps 2 through 7.

1	Task	Description
	Task	Save a Trip
	Machine State	In the Route Planner.
	Successful Completion	Trip will show up in the "Saved Trips" page.
	Benchmark	30 seconds

View a previous trip - KJ

1. The user will attempt to navigate to the Previous Trips page.
2. The user will attempt to find a Previous saved Trip.
3. The user will click the file picker button.
4. The user will select a file from their computer.
5. The user will click the upload image button.

2	Task	Description
	Task	View a previous trip.
	Machine State	On the home page, already logged in.
	Successful Completion	The user will be able to see previously created trips.
	Benchmark	20 seconds

Upload an image - Jesus

1. The user will attempt to find the Previous Trips page.
2. The user will have to click on the photo gallery option of the previous trip.
3. The user will have to click the “Choose a File” button.
4. The user will then have to choose the image they would like to upload.
5. The user will click the “Upload” button to upload the image.

3	Task	Description
	Task	Upload an image.
	Machine State	On the home page, already logged in.
	Successful Completion	The photo gallery will have the users uploaded photos.
	Benchmark	60 seconds

Reopen a saved trip - Ruza

1. The user will attempt to find the Saved Trips page.
2. The user will attempt to open a specific saved trip.
3. The user will view the details of the saved trip.

4	Task	Description
	Task	Reopen a saved trip.
	Machine State	On the home page, already logged in.
	Successful Completion	The user will be in the route planner, with the details of the reopened trip on the screen.
	Benchmark	30 seconds

Learn about the team members - Yi

1. The user will attempt to find the About page.
2. The user will have to click on a member's photo or name to get the member's details.
3. The user will attempt to access each member's information page.

5	Task	Description
	Task	Learn about the team members.
	Machine State	On the home page.
	Successful Completion	The user will be reading information about one of the team members from their individual "About Me" page.
	Benchmark	60 seconds

Questionnaire

For each of these questions, please rate your response as one of the following: *Strongly Disagree*, *Disagree*, *Neutral*, *Agree*, *Strongly Agree*

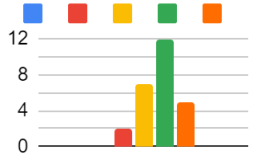
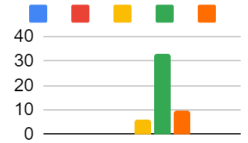
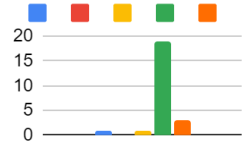
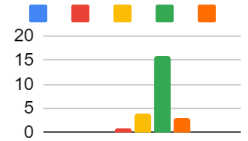
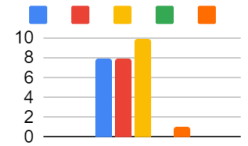
1. It is easy to find the list of activities in the Route Planner.
2. The airport search feature was easy to use.
3. The flight results show enough information for me to pick a flight in the Route Planner.
4. It is easy to find the Previous Trips page.
5. It is easy to see previous saved trips.
6. It is easy to find all previous trips.
7. It is easy to click the file picker button.
8. It is easy to select an image to upload from the computer.
9. It is easy to click on the upload image button.
10. It is easy to find the Saved Trips page.
11. It is easy to open a specific saved trip.
12. It is easy to find details of the saved trip.
13. The process of uploading an image is simple.
14. It is easy to find the photo gallery.
15. I was able to upload more than one image.
16. It is easy to find the About page.
17. It is easy to find all team members' photos and names.
18. It is easy to find a specific team member's detailed information.

Questionnaire Results

Question #	Participant								
	1	2	3	4	5	6	7	8	9
1	Agree	Agree	Agree	Agree	Agree	Neutral	Neutral	Agree	Agree
2	Agree	Strongly Agree	Agree	Agree	Agree	Strongly Agree	Neutral	Agree	Neutral
3	Neutral	Strongly Agree	Disagree	Neutral	Neutral	Strongly Agree	Agree	Strongly Agree	Disagree
4	Agree	Strongly Agree	Agree	Agree	Agree	Strongly Agree	Agree	Strongly Agree	Agree
5	Neutral	Agree	Agree	Agree	Agree	Strongly Agree	Agree	Agree	Agree
6	Neutral	Agree	Neutral	Neutral	Agree	Strongly Agree	Agree	Agree	Neutral
7	Agree	Agree	Agree	Agree	Neutral	Agree	Agree	Agree	Agree
8	Agree	Agree	Agree	Strongly Agree	Agree	Strongly Agree	Agree	Strongly Agree	Agree
9	Agree	Agree	Agree	Agree	Agree	Strongly Agree	Agree	Strongly Agree	Agree
10	Agree	Agree	Agree	Agree	Agree	Strongly Agree	Agree	Strongly Disagree	Agree
11	Agree	Agree	Agree	Agree	Agree	Strongly Agree	Agree	Agree	Agree
12	Neutral	Agree	Agree	Agree	Agree	Strongly Agree	Agree	Agree	Agree

13	Disagree	Agree	Agree	Agree	Neutral	Strongly Agree	Agree	Agree	Agree
14	Neutral	Agree	Agree	Neutral	Neutral	Strongly Agree	Agree	Agree	Agree
15	Agree	Agree	Agree	Agree	Agree	Strongly Agree	Agree	Agree	Agree
16	Neutral	Neutral	Strongly Disagree	Neutral	Neutral	Strongly Agree	Strongly Disagree	Disagree	Neutral
17	Disagree	Disagree	Strongly Disagree	Neutral	Neutral	Strongly Disagree	Strongly Disagree	Disagree	Disagree
18	Neutral	Disagree	Strongly Disagree	Neutral	Neutral	Strongly Disagree	Strongly Disagree	Disagree	Disagree

Usability Testing Results

Test Name	% Completed	Errors	Comments	% Time	Response Distribution*
Save a trip	90%	Some UI elements glitched	Date is missing from flight results. Saving does not work.	30s	
View a previous trip	100%	*None.*	Only one trip available to view	20s	
Reopen a saved trip	100%	*None.*	No saved details	30s	
Upload an image	100%	*None.*	Should be accessible elsewhere	60s	
Learn about the team members	30%	Broken link	Have to use another port	60s	

* Charts include combined responses for all questionnaire questions belonging to that Test (blue=strongly disagree, red=disagree, orange=neutral, green=agree, light red=strongly agree)

3. QA Test Plan

QA Testing Steps

Test Name: Secured Password in Registration

Purpose: See if the user's password is encrypted in database

Step #	Step Description
1	Fill out the registration form and submit (or send a POST request to /users/register).
2	Search the database for registered users (SELECT username, password_hashed FROM registered_user).
3	Check if the password_hashed column contains a cleartext password or a hash value.

Test Name: User Sessions at Login

Purpose: Check if a user session is created after login

Step #	Step Description
1	Fill out the login form and submit.
2	Check the right-most part of the navbar for the logged in user's username.

Test Name: Flight Searching via Amadeus API

Purpose: To see if correct flights are retrieved based on parameters

Step #	Step Description
1	Make a GET request to our flight search endpoint: /planner/flights?originLocationCode=LAX&destinationLocationCode=SFO&departureDate=2021-09-02&adults=1&currencyCode=USD&max=10
2	Change the query parameters to match the desired test.
3	Continue to the page.
4	Ensure that the results returned match the parameters inputted in the previous step.

Test Name: Check Access Control to Pages

Purpose: To make sure that pages which require login redirect anonymous users to the login page

Step #	Step Description
1	Go to the website and logout if already logged in.
2	Try navigating to a restricted page such as /planner or /accountmanagement.
3	Ensure that you are redirected to the login page.

Test Name: No Location Restriction

Purpose: Make sure that the website is accessible from other countries.

Step #	Step Description
1	Send the website link to someone outside of the US (or use a VPN/proxy to access the website).
2	Check that the website loads under these conditions.

QA Test Results

#	Test	Purpose	Input/User Action	Expected Result	Pass/Fail
1	Secured password in registration	See if the user's password is encrypted in database	Registration post request to /users/register	The password is encrypted before sending to database	Pass
2	User sessions at login	Check if a user session is created after login	Successful authentication	The user's username will be pulled from the session data and displayed in the navbar	Pass
3	Flight searching via Amadeus API	To see if correct flights are retrieved based on parameters	var response = await amadeus.shopping.flightOffersSearch.get({ originLocationCode: 'LAX', destinationLocationCode: 'SFO', departureDate: '2021-08-04', currencyCode: 'USD', adults: 1, nonStop: true, max: 5 });	Results should be displayed like this: 0 LAX to SFO jetBlue PT1H23M USD 38.40 1 LAX to SFO jetBlue PT1H26M USD 38.40	Pass

4	Access Control to Certain Pages	Check if expected pages are inaccessible to users that are not logged in	Manually visiting /savedtrips	Redirect to /login	Fail
5	No Location Restrictio n	To ensure users can access the website outside of the US	Visit our URL http://172.93.54.146:3000/ from Iceland	The home page should display	Pass

4. Code Review

Coding Style

Capitalization: camelCase

Indentation Style: “One True Brace” - braces go on the same line as function declarations, if-else, etc.

(email correspondence begins on next page)

Email Correspondence

From: Taylor Artunian <tartunian@mail.sfsu.edu>
Sent: Wednesday, July 28, 2021 10:13 PM
To: Luis Adrian Alfaro <lalfaro4@mail.sfsu.edu>
Subject: CSC648 Team2 Code Review

Hey Luis,

I am submitting a portion of my code from the Route Planner to you so that you can perform a code review. Attached is the client-side JavaScript for the Route Planner. Please comment on any part of the code, check for consistency with our coding style, and share your feedback with me when you have some time.

Thanks,

Taylor

From: Luis Adrian Alfaro <lalfaro4@mail.sfsu.edu>
Sent: Thursdaysday, July 29, 2021 2:34 PM
To: Taylor Artunian <tartunian@mail.sfsu.edu>
Subject: Re: CSC648 Team2 Code Review

Good Afternoon Taylor,

Good job on following our coding standards in the code you submitted for code review. I did notice some issues, most of them being comments, and have highlighted my comments regarding those issues. In the attached document you shall see my comments around the parts where I did notice some problems. Other than that everything else was good!

Best Regards,

Luis Alfaro

Code & Comments From Attached Document

```
//No comment explaining the fetchURL function. -L.A.
async function fetchURL(method, endpoint, parameters) {
    var url = new URL(getURLBase() + endpoint),
        params = parameters
    Object.keys(params).forEach(key => url.searchParams.append(key, params[key]));
    var response = await fetch(url, {
        method: method,
        headers: new Headers({
            // 'Authorization': `Bearer ${token}`,
//Will new Headers always be empty, since there is commented out code inside of new Headers? -L.A.
        });
    });
    return response;
}
/*****************************************************************************************
* Event handler for planner-result-item-button when it is within an Activity
//Comment not describing what this event handler is going to do, only describes a
//condition. -L.A.
*****************************************************************************************/
function removeFlightClickHandler(e) {
    // Get the parent result item of the button
    var resultItem = e.target.closest('.planner-result-item');
    resultItem.remove();
```

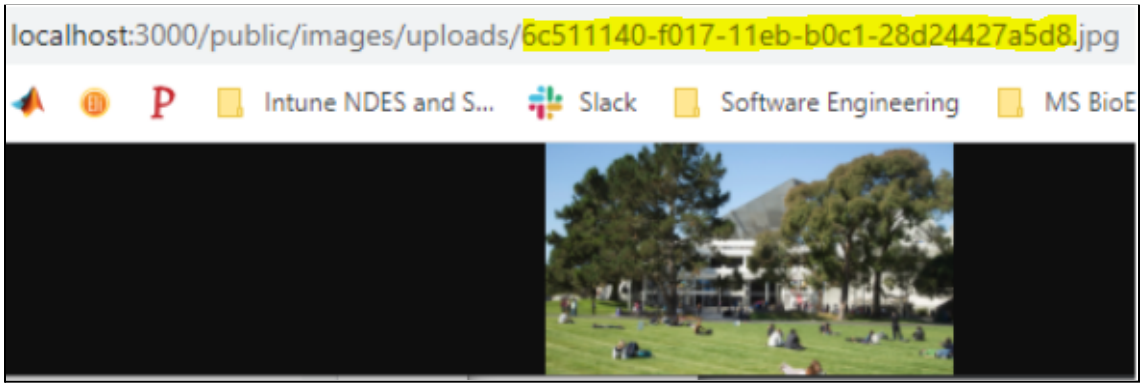
```
}

/*****************************************************************************************

* Event handler for planner-result-item-button when it is in the result list

//Comment not describing what this event handler is going to do, only describes a
//condition. -L.A.

*****************************************************************************************/

function addFlightClickHandler(e) {


    // Get the parent result item of the button

    var resultItem = e.target.closest('.planner-result-item');


    // Get the currently selected Activity and child container

    var selectedActivity =
document.querySelector('input[name="planner-selected-activity"]:checked').parentNode.pa
rentNode;

    var selectedActivityChildContainer =
selectedActivity.querySelector('.planner-activity-child-container');


    // Get the results container

    var resultsList = document.getElementById('planner-results-container');


    // Make a copy of the flight offer

    var copyOfFlightOffer = resultItem.cloneNode(true);

    var flightOfferButton = copyOfFlightOffer.querySelector('.planner-result-item-button')

    flightOfferButton.value = 'Reset';

    flightOfferButton.addEventListener('click', removeFlightClickHandler);
```

```
    //typo in under -L.A
    // Move the result item (new flight offer) uner the selected Activity
    if (selectedActivityChildContainer.childElementCount == 0) {
        selectedActivityChildContainer.innerHTML = '';
        selectedActivityChildContainer.appendChild(copyOfFlightOffer);
    } else {
        selectedActivityChildContainer.replaceChild(copyOfFlightOffer,
selectedActivityChildContainer.children[0]);
    }
}
```

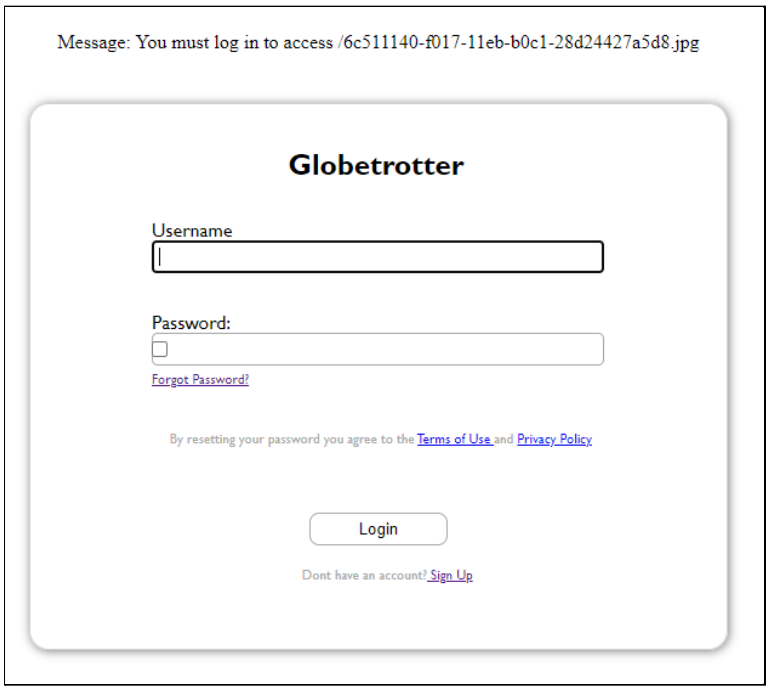
5. Self-Check on Best Practices

Images

When photos are uploaded to our website, random UUID's are used for file names. This is done to avoid revealing a user's name or other personal information which may be in the original file name.



Additionally, we are protecting the static images of the website which should not be publicly accessible. If a user is not logged in, they can no longer access photos uploaded to the Photo Gallery even if they manually enter the URL.



Passwords

We hash the password using the .hash( ) function from the bcrypt library before storing into the database. As you can see in the createUser( ) function, line 147, we hash the password. We store the hashed password into a variable and pass it into an array that will be used as a parameter for the query. We then store the password on line 153 and at no point does the Globetrotter team know what the password of the newly created user is.

```
139
140  /*****************************************************************************
141   * Create a user in the database (all fields)
142   *****************************************************************************/
143  async function createUser(email, username, password, firstName, lastName, birthday, gender,
144      preferredCurrency, homeLocationAddressLine1, homeLocationAddressLine2,
145      homeLocationCity, homeLocationState, homeLocationCountry, homeLocationPostalCode,
146      primaryPhoneCountryCode, primaryPhoneNumber) {
147      var passwordHash = await bcrypt.hash(password, 10);
148      var query = 'CALL usp_register_user(?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, @user_id);';
149      var params = [email, username, passwordHash, firstName, lastName, birthday, gender,
150          preferredCurrency, homeLocationAddressLine1, homeLocationAddressLine2,
151          homeLocationCity, homeLocationState, homeLocationCountry, homeLocationPostalCode,
152          primaryPhoneCountryCode, primaryPhoneNumber];
153      var result = await runQuery(query, params);
154      return result;
155  }
```

```
+--------------+--------------------------------------------------------------+
| username     | password_hashed                                              |
+--------------+--------------------------------------------------------------+
| globetrotter | $2b$10$4TF8lwcJBfFW.tgew44ubOfQQCR7oZNxAg.UJXI.zwgdW5xMZd6US |
| GLOBETROTTED | $9b$99$9TF8lwcJBfFW.ftqd44ubOfQQCR7oZNxAg.UJXI.zwgdW5xMZd9EU |
+--------------+--------------------------------------------------------------+
2 rows in set (0.06 sec)

mysql>
```

Input Validation

Registration

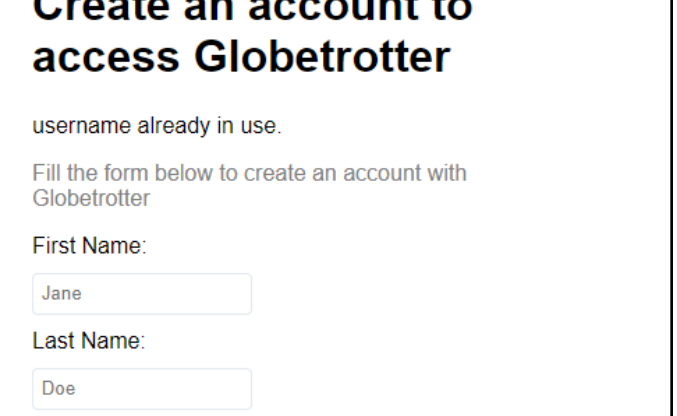
All of the fields in our registration form are specified as required with the *required* HTML tag.

By doing this, we reduce the number of calls to our registration endpoints and prevent needless page reloads for the user.

```
<form method="POST" action="/users/register" class="center">
    <h1>Create an account to access Globetrotter</h1>
    <div>{{message}}</div>
    <p class="light-text">Fill the form below to create an account with Globetrotter</p>

    <div class="row">
        <label>First Name:</label>
        <input type="text" name="firstName" placeholder=" Jane" required>
    </div>
    <div class="row">
        <label>Last Name:</label>
        <input type="text" name="lastName" placeholder=" Doe" required>
    </div>
```

When a user submits the registration form, we check our database first to see if the username or email are already in use.

```
// Check if username is already in use
var user = await database.getUserByUsername(username);
if (user) {
    log('username already in use', 'fail');
    return res.redirect(`/registration?message=username already in use.`);
}

// Check if email is already in use
user = await database.getUserByEmail(email);
if (user) {
    log('email already in use', 'fail');
    return res.redirect(`/registration?message=email already in use.`);
}
```

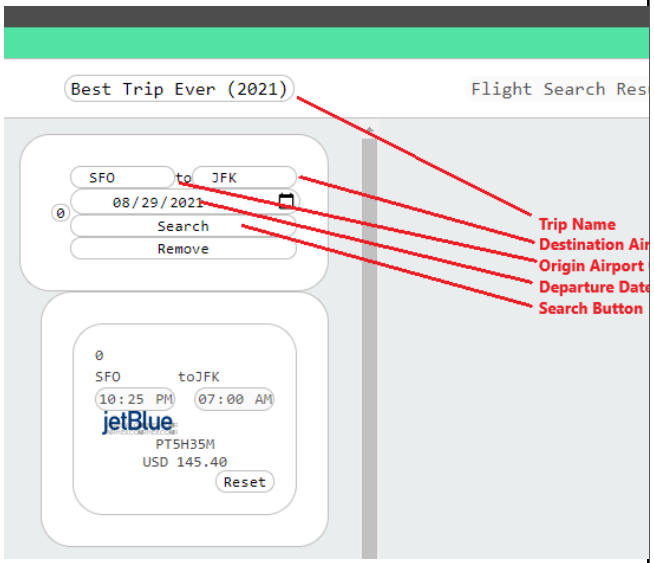


This is the Registration page. If the registration is unsuccessful, we display a message to the user explaining why.

Route Planner

Before a user can search for flights, all three inputs must be valid. The code below checks whether the fields are filled in, whether the departure date is valid, and whether the airport codes are valid options from the autocomplete list.

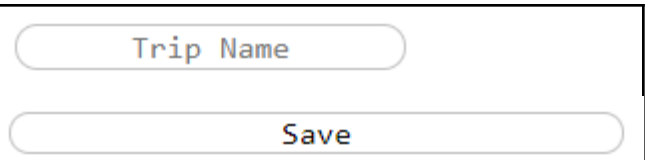
```
----------------------------------Excerpt from
planner.js--------------------------------
if(isIncompleteActivity(activity)) {
        alert('Some search parameters are invalid.');
        return;
 }
if (!validateAirportInput(originInputValue, originInputDatalist) ||
!validateAirportInput(destinationInputValue, destinationInputDatalist)) {
        alert('Please input a valid airport.');
        return;
}

----------------------------------Excerpt from
validator.js--------------------------------
function validateAirportInput(inputValue, datalistElement) {
    for (var option of datalistElement.options) {
        var airportCode = option.value;
        if (inputValue == airportCode) {
            return true;
        }
    }
    return false;
}
function isIncompleteActivity(activity) {
    var originCode =
activity.querySelector('.planner-activity-origin-input').value;
    var destinationCode =
activity.querySelector('.planner-activity-destination-input').value;
    var departureDate =
activity.querySelector('.planner-activity-departure-date').value;
    if (originCode == '' || destinationCode == '' ||
isNaN(Date.parse(departureDate))) {
        return true;
    }
    return false;
}
```



This is the Route Planner. The code files referenced are client-side JavaScript code used on this page. "*planner.js*" is the main script for loading the data from the server and displaying the UI elements. "*validator.js*" is a separate script with just the two functions shown and is used to validate the UI elements/inputs.

When the user saves the Trip, the same code as above is run, but this time for all Activities in the Trip. Additionally, the Trip Name field is validated for empty values.

```
if(tripName=='') {
      alert('Please enter a valid name for your trip.');
      return;
 }
```



6. Non-Functional Requirements

Category	Non-Functional Requirement	Status
Functionality	The website shall be hosted on SSD Nodes.	DONE
Functionality	The website shall be user friendly.	DONE
Functionality	The website shall be easy to navigate.	DONE
Security	User login credentials will be validated against the user database.	DONE
Security	Changes to access controls in the database will be reflected immediately in the user interface.	DONE
Security	The user's passwords will be encrypted in the database.	DONE
Security	User passwords will be case sensitive.	DONE
Security	User passwords shall have at least 8 characters.	DONE
Security	User passwords shall have at least one upper case letter.	DONE
Security	User passwords shall have at least one special character.	DONE
Security	The website will use sessions to track logged in users.	DONE
System Requirements	The website shall function on at least version 91.0.4472.101 of Google Chrome.	DONE
System Requirements	The website shall function on at least version 91.0.864.48 of Microsoft Edge.	DONE
System Requirements	The website shall function on at least version 14.1 of Apple Safari.	DONE
System Requirements	The website shall function on at least version 89.0.1 of Mozilla Firefox.	DONE
Marketing	The website shall use the company logo displayed.	DONE
Marketing	The company shall have engaging content.	DONE

Content	The website shall have a navigation bar.	DONE
Content	The website shall have a consistency of its design on every page.	ON TRACK
Content	The flights returned from API will match parameters entered by the user.	DONE
Coding Standards	The code will be portable.	DONE
Coding Standards	The code will have a consistent indentation style.	ON TRACK
Coding Standards	The code shall be understandable.	ON TRACK
Coding Standards	The code shall contain comments so that team members can quickly learn what different pieces of code do without having to dive deep into the code.	ON TRACK
Coding Standards	There shall be a limit to the line length of the code to avoid horizontal scrolling.	ON TRACK
Coding Standards	The team lead shall decide if code is at the agreed upon standard.	ON TRACK
Coding Standards	The team lead shall decide whether code can be pushed to the main branch.	DONE
Coding Standards	The coding will be done in a modular way.	DONE
Coding Standards	Code modules will be reused as often as possible.	DONE
Coding Standards	CSS will be applied to classes.	DONE
Coding Standards	Scripts will be in their own files	DONE
Coding Standards	NodeJS modules will be installed in the ‘node_modules’ folder.	DONE
Coding Standards	The website will use a public and private folder to serve web content.	DONE

Coding Standards	Stylesheets will be in their own files.	DONE
Coding Standards	HTML files will be in the HTML folder.	DONE
Coding Standards	CSS files will be in the CSS folder.	DONE
Coding Standards	Script files will be in the Script folder.	DONE
Coding Standards	Express routers will be placed in their own folder.	DONE
Coding Standards	Handlebars partials will be in their own folder.	DONE
Coding Standards	Handlebars templates will be in their own folder.	DONE
Coding Standards	Handlebars layouts will be in their own folder.	DONE
Coding Standards	Images will be in their own folder.	DONE
Coding Standards	The main branch for frontend code will be the 'frontend-development' branch.	DONE
Coding Standards	The main branch for backend code will be the backend-development' branch.	DONE
Coding Standards	Code that is ready to be tested will be moved from the 'development' branch to the 'testing' branch for the frontend code.	ISSUE, no testing branch
Coding Standards	Code that is ready to be tested will be moved from the 'development' branch to the 'testing' branch for the backend code.	ISSUE, no testing branch
Coding Standards	Code will only be pushed after pulling first.	DONE
Availability	The website will not restrict users based on their geographic location.	DONE
Availability	The website's URL shall be reliable to access its	DONE

	contents.	
Availability	The website's URLs shall be simple and short.	ON TRACK, some URL's use query parameters which increase their length
Availability	The website will load in at least 20 seconds unless there is a system error.	DONE
Scalability	The website will be designed such that it can be migrated to higher-performance hardware if demand increases.	DONE
Fault Intolerance	A backup of the virtual server will be performed daily.	DONE
Fault Intolerance	The virtual server will be restored from a backup in the event of a total system failure.	DONE
Expected Load	The website is expected to handle 10-20 users at a time.	ON TRACK
Privacy	The website shall not save the users CVV number during the checkout process.	ON TRACK, checkout process is priority 2
Privacy	The website shall not save the users CVV number after the checkout process.	ON TRACK, checkout process is priority 2
Privacy	This website shall not sell user personal information to third parties.	DONE
Privacy	The website will not share private user information with other users unless where explicitly done by the user.	DONE
Privacy	The website shall not store any user medical information.	DONE
Privacy	The website will not store users' IP addresses.	DONE

Privacy	The website will not share users' IP addresses.	DONE
Legal	The website will not purposely publish any false or fraudulent information in any form.	DONE
Legal	The website shall have Terms and Conditions which users must consent to before logging in.	DONE
Legal	The website shall notify the user when the Terms and Conditions have been changed.	ISSUE, we would have to develop a separate process to send emails out
Legal	The website shall not let the user continue until they have accepted the updated Terms and Conditions.	DONE, there is a message on the Registration Page explaining this agreement
Look and Feel	The website will use a uniform color palette on all pages of the site.	ON TRACK
Look and Feel	The website will have a responsive design.	ON TRACK
Look and Feel	The website will scale to the width of the screen.	ON TRACK
Look and Feel	The website will scale to the height of the screen.	ON TRACK
Look and Feel	The website will be optimized for tablet clients.	ISSUE, we too have many style files
Look and Feel	The website will be optimized for desktop clients.	ON TRACK
Look and Feel	The website will be optimized for mobile clients.	ISSUE, same as above

Look and Feel	The website will have readable text font size.	ON TRACK
Look and Feel	The website will have uniform font styles.	ON TRACK
Content	The website will have no typos.	ON TRACK
Content	The website will have alt tags for images.	ON TRACK
Look and Feel	The website will accurately display currency to two decimal places.	DONE

7. Contributions

Taylor:

- DB procedures, Server code for request handling and DB interaction, Route Planner client and server
- Testing Plan for Search for A Flight

Yi Liu:

- Testing plan for Learn About the Team members

Luis Alfaro:

- Made updates to the database procedures and model, fixed DB errors
- Integrated DB procedures into server code and request handlers

Robin Fernando:

Ruja Rajbhandari:

- Configured the navbar
- Testing Plan for Reopen a Saved Trip

Kajeme Cheneque:

- Performed testing on the webpages
- Testing Plan for View A Previous Trip
- Fixed broken links on Registration and Reset Password

Jesus Correa:

- Configured request handlers for About Us, Individual about pages, Terms and Conditions, Privacy page
- Designed Home Page and Photo Gallery, About Us, Individual about pages, Terms and Conditions, Privacy page
- Testing Plan for Upload an Image

Milestone 3 V2

PREPARED FOR

Jose Ortiz

PREPARED BY

Team 2

Globetrotter

JUL 22, 2021

Globetrotter Travel Assistant

Milestone 3

Team 2

Role	Name	Email
Team Lead/Github Master	Taylor Artunian	tartunian@mail.sfsu.edu
Front-End Lead	Yi Liu	
Back-End Lead	Luis Alfaro	
Front End	Ruja Rajbhandari	
	Kajeme Cheneque	
	Jesus Correa	jcorrea2@mail.sfsu.edu
	Robin Fernando	

Revision History

Version	Date Submitted	Date Revised
M3V2	07/30/21	-
M3V1	07/22/21	07/30/21
M2V2	07/22/21	-
M2V1	07/08/21	07/22/21
M1V2	07/08/21	-
M1V1	06/22/21	07/08/21

Table of Contents

1. Data Definitions

Entity Name		Entity Description					
	Column Name	**Column Description**	**Data Type**	**Length**	**Primary Key**	**Nullable**	**Unique**
activity		This represents something that a user does during a trip or is planning to do.					
	activity_id	The id of the activity.	varchar	36	true	false	true
	end_location	The id of the location at which the activity ends.	varchar	36	false	true	false
	end_time	The timestamp at which the activity ends.	timestamp	19	false	false	false
	start_location	The id of the location at which the activity starts.	varchar	36	false	false	false
	start_time	The timestamp at which the activity starts.	timestamp	19	false	false	false
	trip	The id of the trip that the activity is part of.	varchar	36	true	false	false
	trip_owner	The id of the trip's owner.	varchar	36	true	false	false
airline		A type of service_provider that sells flights.					

	airline_code	The IATA code of the airline.	varchar	7	true	false	true
	service_provider	The id of the service_provider that the airline represents.	varchar	36	true	false	true
airport		A type of location that belongs to an airport.					
	iata_code	The IATA code of the airport.	varchar	3	true	false	true
	location	The id of the location of the airport.	varchar	36	true	false	true
currency_code		A collection of codes for currencies.					
	code	The ISO 4217 code for the currency.	varchar	3	true	false	false
	name	The name of the currency.	varchar	255	false	false	false
file		This is a file uploaded by a registered_user to our web server.					
	extension	The extension of the file.	varchar	5	false	true	false
	file_id	The unique id of the file.	varchar	36	true	false	true
	file_name	The name of the file.	varchar	255	false	false	false
	folder_path	The folder path of the file.	varchar	255	false	false	false
	full_file_path	The complete path of the file.	varchar	255	false	false	true
	owner	The owner of the fil.	varchar	36	true	false	false

flight_activity		This is a type of activity that represents a flight in the user's trip.					
	activity	The id of the activity.	varchar	36	true	false	true
	flight_offer_json_data	The json data of the selected offer from the airline.	varchar	4095	false	false	false
	trip	The id of the trip.	varchar	36	true	false	false
	trip_owner	The id of the trip owner.	varchar	36	true	false	false
guest_to_trip_association		Associates a user to a trip as a guest.					
	guest	The id of the guest's user entity.	varchar	36	true	false	false
	trip	The id of the trip.	varchar	36	true	false	false
	trip_owner	The id of the trip owner.	varchar	36	true	false	false
home_location		Associates a user with a location as their home.					
	location	The id of the location which represents the registered_user's home.	varchar	36	true	false	false
	user	The id of registered_user which the home_location belongs to.	varchar	36	true	false	true
hotel_activity		This is a type of activity which holds information about a planned hotel stay.					
	activity	The id of the activity.	varchar	36	true	false	true

	hotel_offer_json_data	The json data of the selected offer from the hotel.	varchar	4095	false	false	false
	trip	The id of the trip.	varchar	36	true	false	false
	trip_owner	The id of the owner.	varchar	36	true	false	false
location		An address, a GPS coordinate or both.					
	city	The name of the city.	varchar	255	false	true	false
	country	The name of the country.	varchar	255	false	true	false
	latitude	The GPS latitude coordinate.	decimal	10.7	false	true	false
	location_id	The id of the location.	varchar	36	true	false	false
	location_name	The name of the location.	varchar	255	false	true	false
	longitude	The GPS longitude coordinate.	decimal	10.7	false	true	false
	postal_code	The postal code.	varchar	255	false	true	false
	state	The name of the state.	varchar	255	false	true	false
photo		This is a type of file which is also an image.					
	description	The description of the photo.	varchar	255	false	false	false
	file	The file which the photo represents.	varchar	36	true	false	false
	file_owner	The owner of the file.	varchar	36	true	false	false

	is_profile_photo	Tells whether the photo is a registered_user's profile photo.	bit	1	false	true	false
	photo_id	The unique id of the photo.	varchar	36	true	false	true
	title	The title of the photo.	varchar	255	false	false	false
photo_album		A photo album is created for each trip and is owned by a user. It is a collection of photos.					
	photo_album_id	The unique id of the photo.	int	10	true	false	false
	trip	The id of the trip which the photo_album belongs to.	varchar	36	true	false	true
	trip_owner	The id of the owner of the trip.	varchar	36	true	false	false
photo_to_photo_album_association		Associates a photo to a photo_album.					
	owner	The owner of the photo file.	varchar	36	true	false	false
	photo	The photo which is part of the album.	varchar	36	true	false	false
	photo_album	The album which the photo is part of.	varchar	36	true	false	false
registered_user		User that has registered. They may own trips, files, photos, photo albums, and reviews.					

	password_hashed	The bcrypt hash of the registered_user's password.	varchar	255	false	false	false
	preferred_currency	The registered_user's preferred currency.	varchar	3	false	true	false
	primary_phone_country_code	The registered_user's primary phone country code.	varchar	2	false	true	false
	primary_phone_number	The registered_user's primary phone number.	varchar	10	false	true	false
	secondary_phone_country_code	The registered_user's primary phone number.	varchar	2	false	true	false
	secondary_phone_number	The registered_user's secondary phone number.	varchar	10	false	true	false
	user	The id of the user entity.	varchar	36	true	false	true
	username	The username of the registered_user.	varchar	255	false	false	true
review		A review of a service_provider written by a registered_user.					
	content	The text content of the review.	varchar	255	false	false	false
	reviewer	The id of the registered_user who made the review.	varchar	36	true	false	false

	service_provider	The id of the service_provider that the review is about.	varchar	36	true	false	false
service_provider		A company that sells services to users such as an airline.					
	location	The id of the service_provider's location.	varchar	36	false	true	false
	service_provider_id	The id of the service provider.	varchar	36	true	false	false
	service_provider_name	The name of the service_provider.	varchar	255	false	false	false
service_record		This is a record of a service purchased from a service provider such as a flight or hotel reservation.					
	activity	The id of the activity.	varchar	36	true	false	false
	json_data	The json data provided by the service_provider.	varchar	510	false	true	false
	service_provider	The id of the service_provider.	varchar	36	true	false	false
	service_record_id	The id of the service_record.	varchar	36	true	false	true
	trip	The id of the trip.	varchar	36	true	false	false
	trip_owner	The id of the trip owner.	varchar	36	true	false	false
trip		A trip holds activities and is named.					

	owner	The id of the trip's owner.	varchar	36	true	false	false
	trip_id	The unique id of the trip.	varchar	36	true	false	true
	trip_name	The unique name of the trip.	varchar	255	false	false	true
user		Unregistered user. They may be guests of trips.					
	email	The unique email of the user.	varchar	255	false	false	true
	user_id	The unique id of the user.	varchar	36	true	false	true
usersessions		Holds session information for logged in users.					
	data	The data associated with the logged in user's session.	mediumtext	0	false	true	false
	expires	The timestamp of when the session expires.	int	10	false	false	false
	session_id	The id of the user_session.	varchar	128	true	false	false

2. Functional Requirements

Priority 1

Account Management

- Registered users can edit their birthday.
- Registered users can edit their email.
- Registered users can edit their home city
- Registered users can edit their home postal code.
- Registered users can edit their home state
- Registered users can edit their home street address.
- Registered users can edit their password.
- Registered users can edit their primary phone number.
- Registered users can edit their profile photo.
- Registered users can edit their secondary phone number.
- Registered users can edit their username.
- Registered users can view the ‘Account Management’ page.
- Unregistered users cannot access Account Management.

Checkout

- Unregistered users cannot access Checkout.

Login

- Registered users can login using the ‘Login’ page.
- Unregistered users can view the ‘Registration’ page.
- Users shall provide their password when logging in.
- Users shall provide their username when logging in.
- Registered users will be able to reset their password by providing their username, email, and new password.
- The password field will be obscured on the Login page.

Meet the Team

- All users can contact the team.
- All users can view the “About Us” page to find out more information about the team.

Photo Album

- Registered users can upload pictures of their trip.
- Unregistered users cannot access Photo Albums.

Previous Trips

- Unregistered users cannot access Previous Trips.

Registration

- Unregistered Users shall provide their date of birth when creating an account
- Unregistered users shall provide a password when creating an account.
- Unregistered users shall provide a primary phone number when creating an account.
- Unregistered users shall provide a username when creating an account.
- Unregistered users shall provide their email when creating an account.
- Unregistered users shall provide their first name when creating an account.
- Unregistered users shall provide their home city when creating an account.
- Unregistered users shall provide their home postal code when creating an account.
- Unregistered users shall provide their home state when creating an account.
- Unregistered users shall provide their home street address when creating an account.
- Unregistered users shall provide their last name when creating an account.
- Unregistered users will be able to create an account.
- Unregistered users shall provide their gender when creating an account.

Route Planner

- All registered users can add at least one destination.
- All registered users can add multiple Activities to a Trip.
- All registered users can edit the destination of a trip.
- All registered users can edit the starting location of a trip.
- All registered users can remove the destination of a trip.
- All registered users can remove the starting location of a trip.
- All registered users can remove Activities from a Trip.
- All registered users shall specify the dates of departure for each flight.
- Unregistered users cannot access the Route Planner.

Saved Trips

- Registered users can view trips that they have saved.
- Unregistered users cannot access Saved Trips.

Website

- The website will be easy for all Users to maneuver through.

Priority 2

Checkout

- Registered guests may purchase trips on which they are guests.
- Registered guests will receive a confirmation email after checkout.
- All registered users can view the cost of the trip when checking out.
- All registered users can view the flights they are purchasing when checking out.

Guests

- All guests added to the trip will be sent an email containing a link to the trip.

Photo Album

- Registered users can delete photos they uploaded.
- Registered users can add a photo description when uploading a picture.
- Registered users can edit their photo descriptions.
- Registered users can edit titles of pictures they uploaded.
- Registered users shall give a title when uploading a picture.

Registration

- Users will be sent a confirmation email after completing registration.

Reviews

- Registered users can create a review of their flight.
- Registered users can delete their reviews.
- Registered users can edit their reviews.
- All users can look at reviews of Airlines.

Route Planner

- All guests can view trips on which they are guests.
- All users can choose from the available flights.
- All users will be able to view arrival times for the available flights.
- All users may add entertainment activities to a particular day of a trip.
- All users may add hotel reservations to a particular day of a trip.
- All users will be able to view weather information for each location.
- Dates added to a trip may not overlap.

Saved Trips

- Registered users can delete trips that they previously created.

- Registered users can edit trips that they previously created.
- Registered users can purchase their Saved Trips.
- Registered users can edit trips on which they are guests.
- Registered users may cancel flights which they purchased.

Priority 3

Account Management

- Registered users can edit their preferred currency.

Guests

- All registered guests can add guests to the trip.
- All registered guests can add multiple flight stops in between the start and destination locations (waypoints).
- All registered guests can add multiple flight stops in between the start and destination locations.
- All registered guests can add other registered users to the trip.
- All registered guests can add unregistered guests to the trip.
- All registered guests can edit the destination of a trip.
- All registered guests can edit the starting location of a trip.
- All registered guests shall specify the dates of departure.
- All registered guests can only add up to nine registered users to the trip.
- All registered guests can only add up to one registered user at a time.
- All registered guests can remove the destination of a trip.
- All registered guests can remove the starting location of a trip.
- All registered guests can remove waypoints.
- Registered users will receive a confirmation email after checkout.
- Unregistered guests may not purchase trips on which they are guests.

Hotels

- Registered users can reserve other registered guests to hotels.

Route Planner

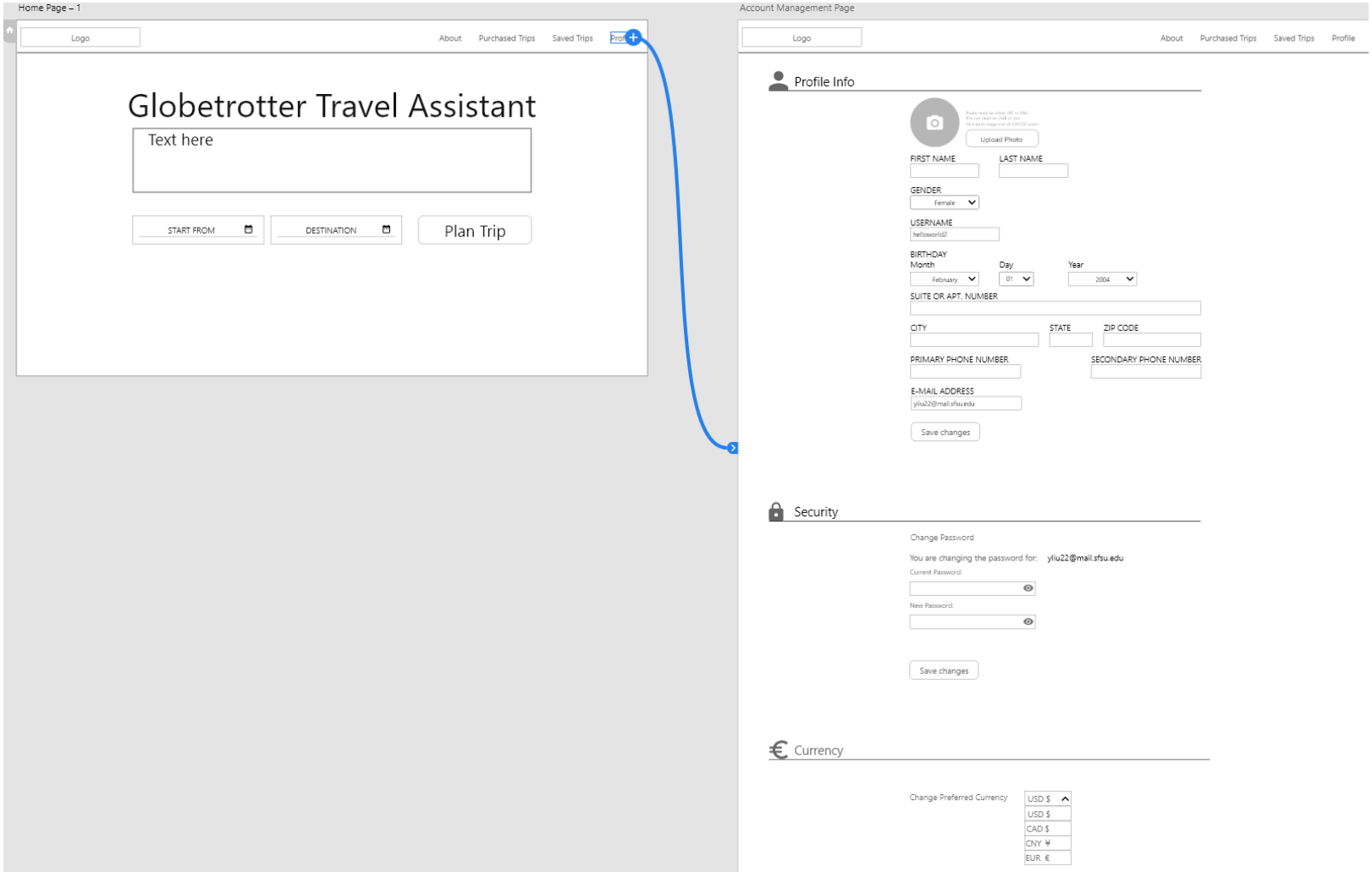
- All registered guests can add at least one starting location.
- All users can edit the displayed currency.
- All users shall specify a budget.

Website

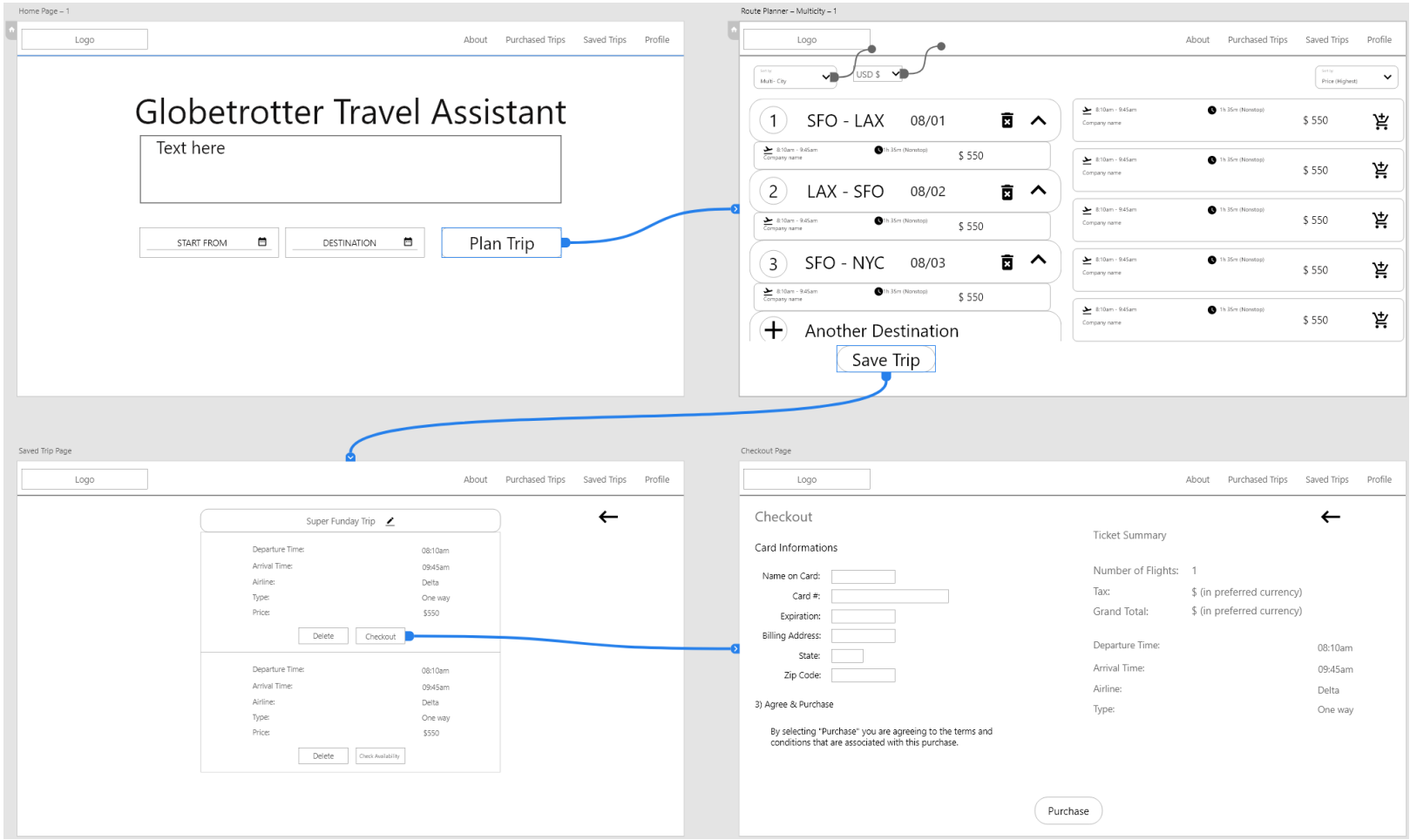
- The displayed currency will default to the preferred currency of a logged in user.

3. UI Wireframes

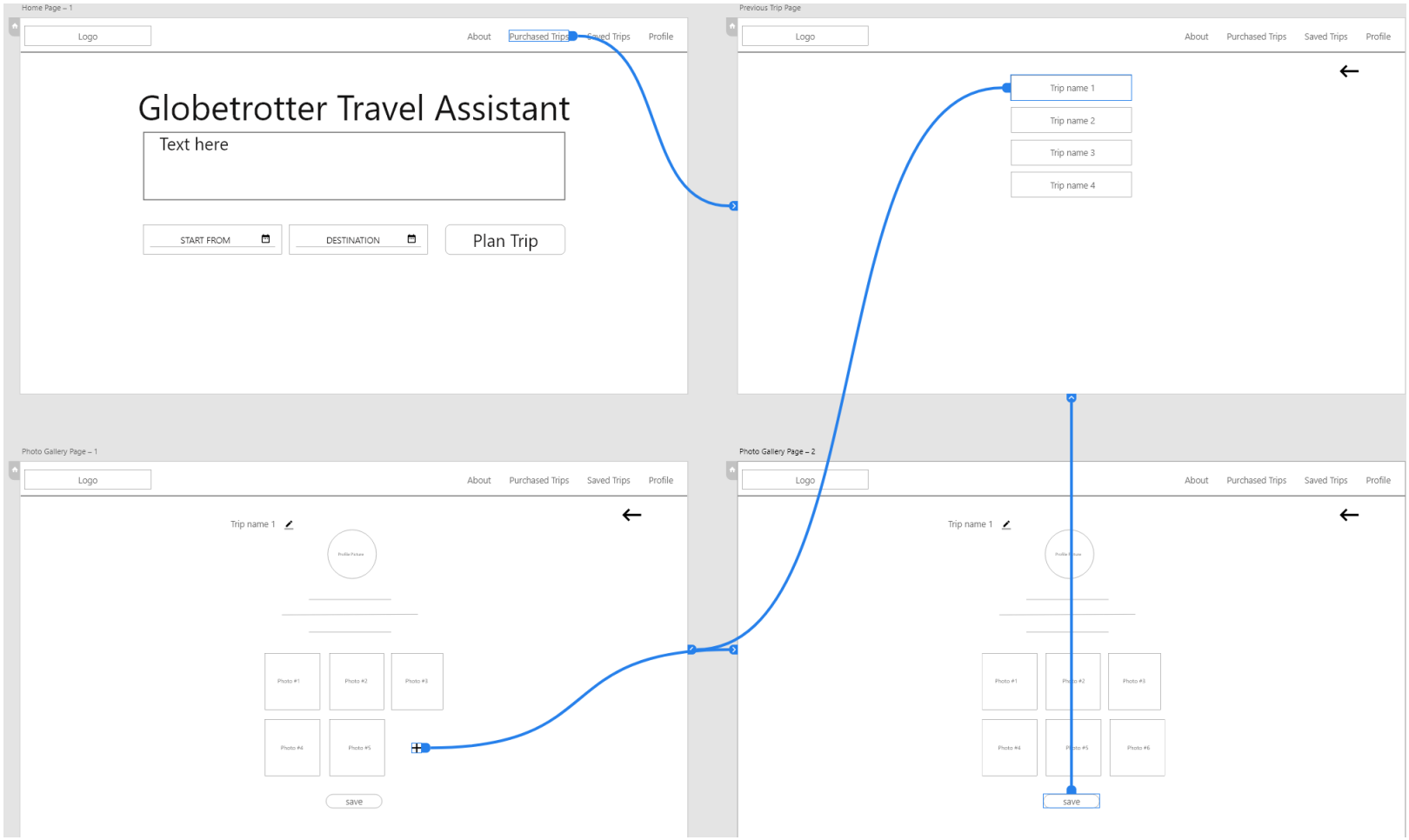
1: Go to Profile Info Page (logged in)



2. Full process of search, add and checkout.



3. Access to Previous (Purchased) Trips and Photo Gallery



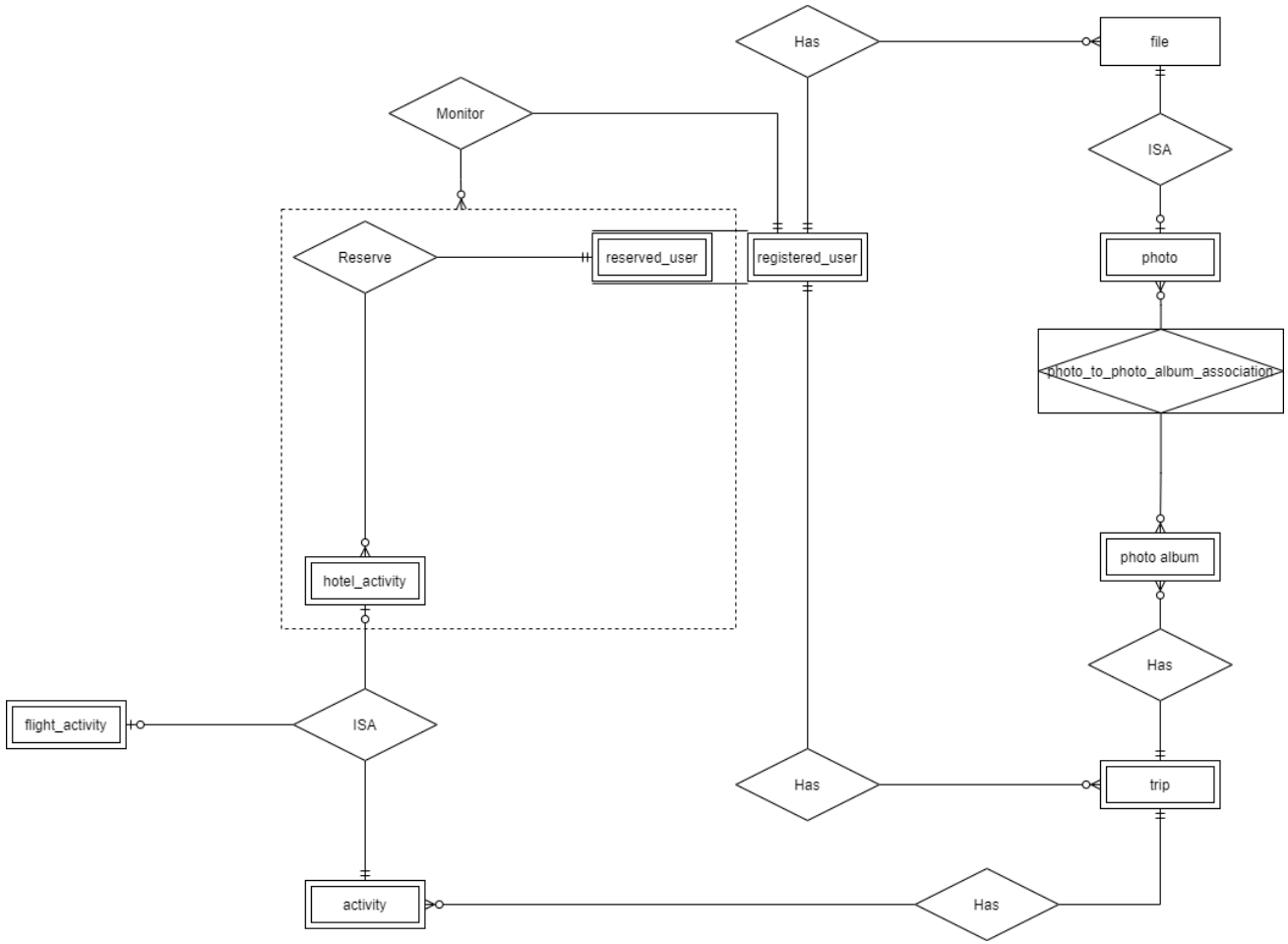
4. Database Architecture and Organization

Business Rules for Photo Albums

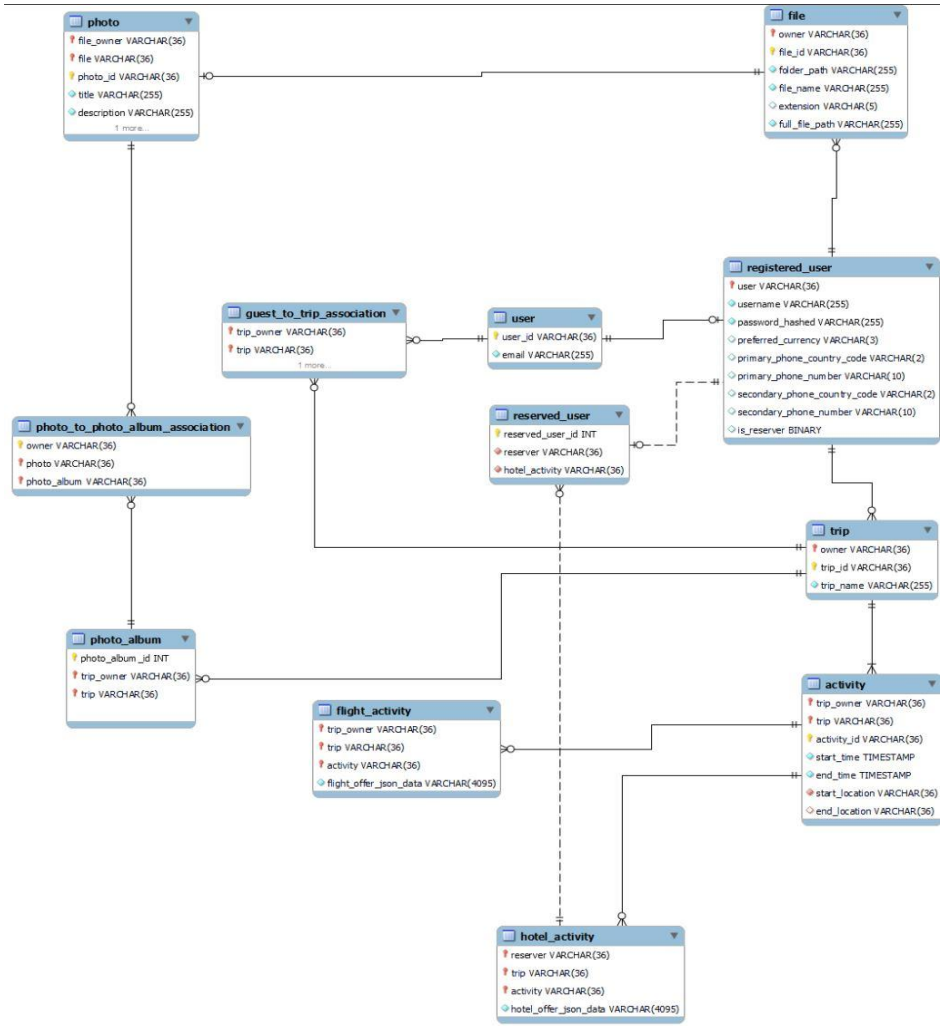
Entities, Attribute, Relation, Quantitative

1. Registered User
    a. A Registered User has many files.
    b. A Registered User shall have many Trips.
2. File
    a. A File can be a Photo.
3. Photo
    a. A Photo ISA File.
    b. A Photo shall be photo_to_photo_album_association by many Photo Albums.
    c. A Photo shall have one name.
    d. A Photo shall have one description.
4. Photo Album
    a. An Album shall photo_to_photo_album_association many Photos
    b. An Album shall have only one Trip.
5. Activity
    a. An Activity is either a flight activity or a hotel activity.
6. Flight Activity
    a. A Flight Activity ISA Activity.
7. Hotel Activity
    a. A Hotel Activity ISA Activity.

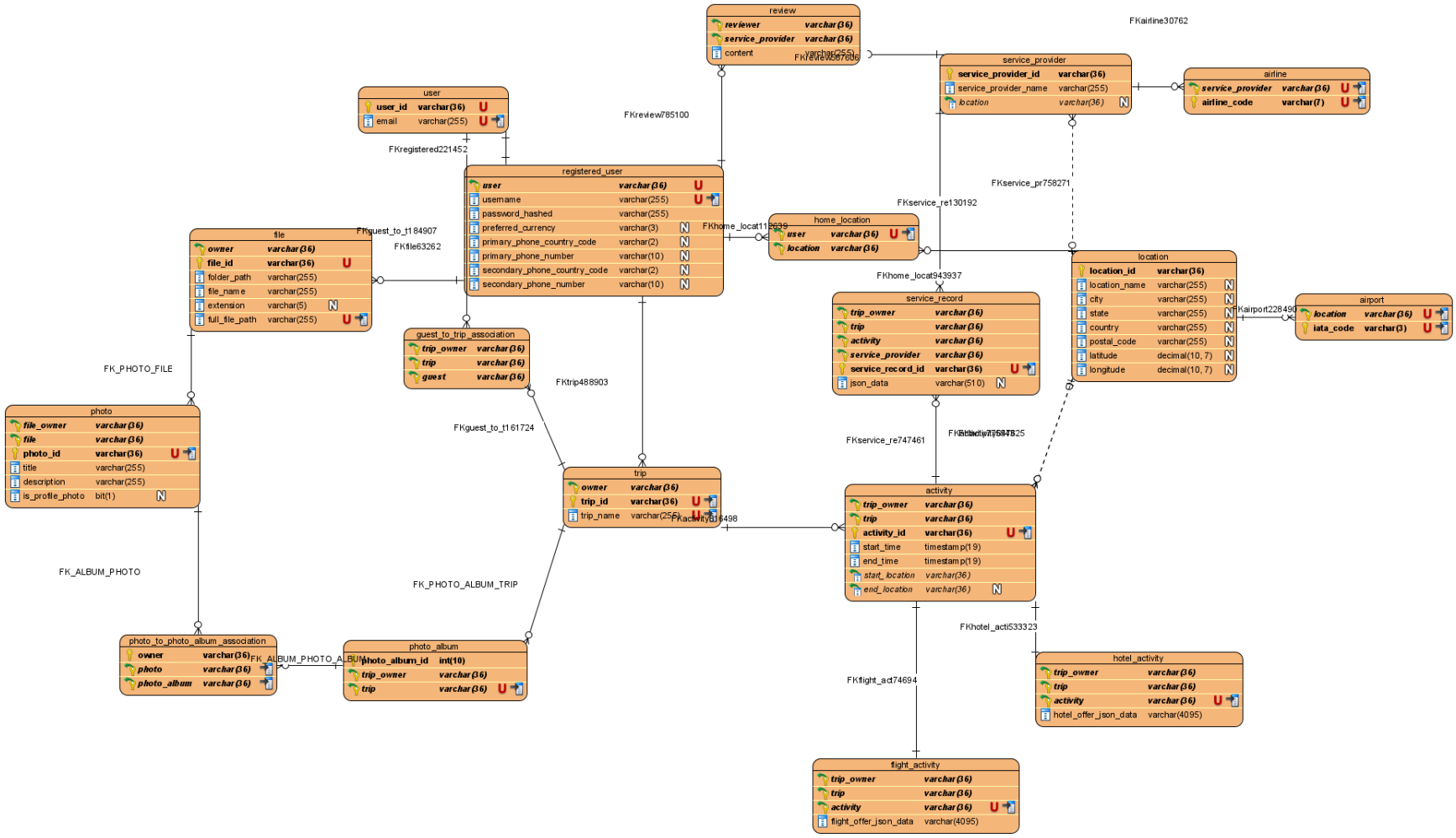
Entity Relationship Diagram (Photo Albums)



Database Model (Photo Albums)



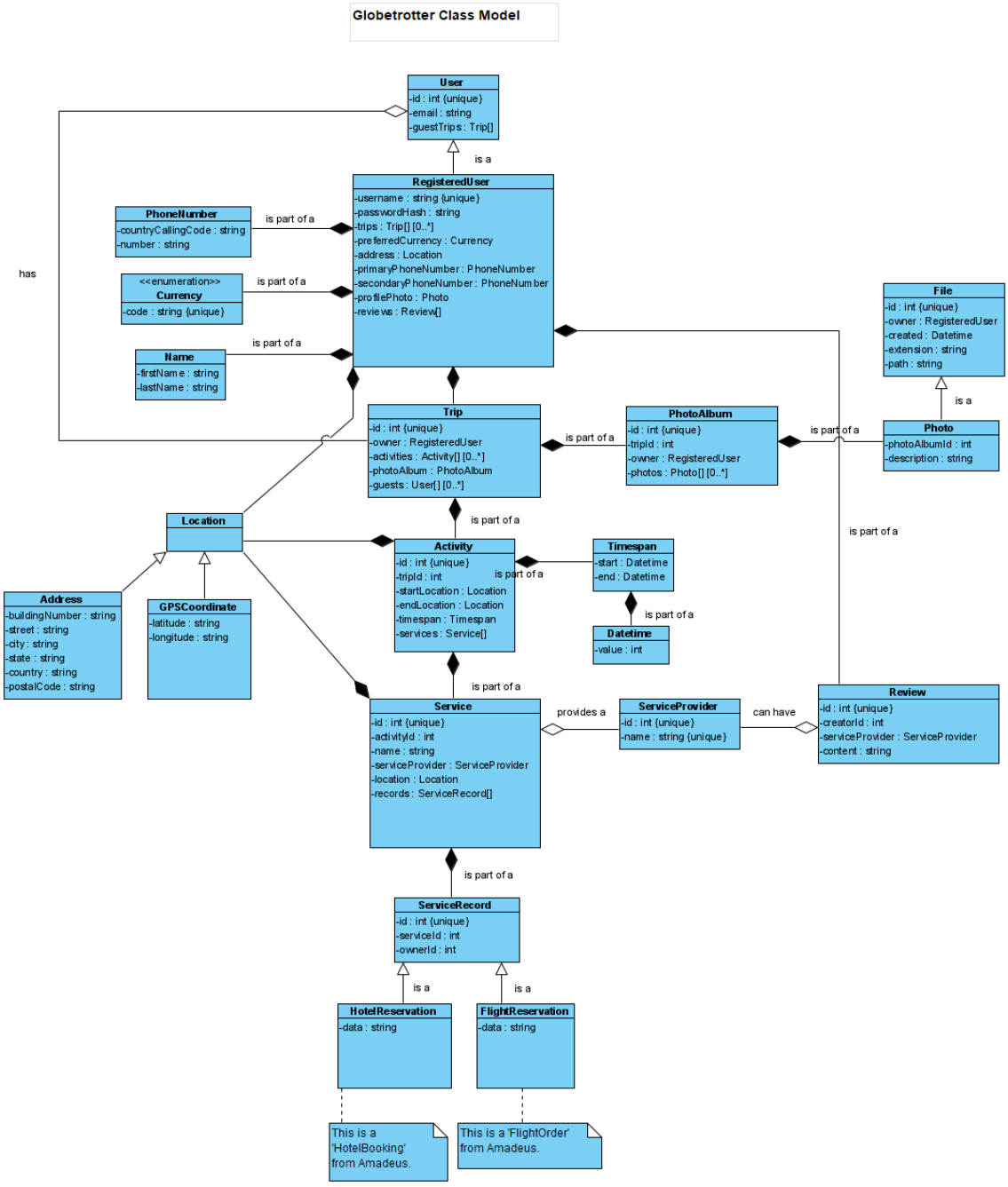
Database Model (Full)



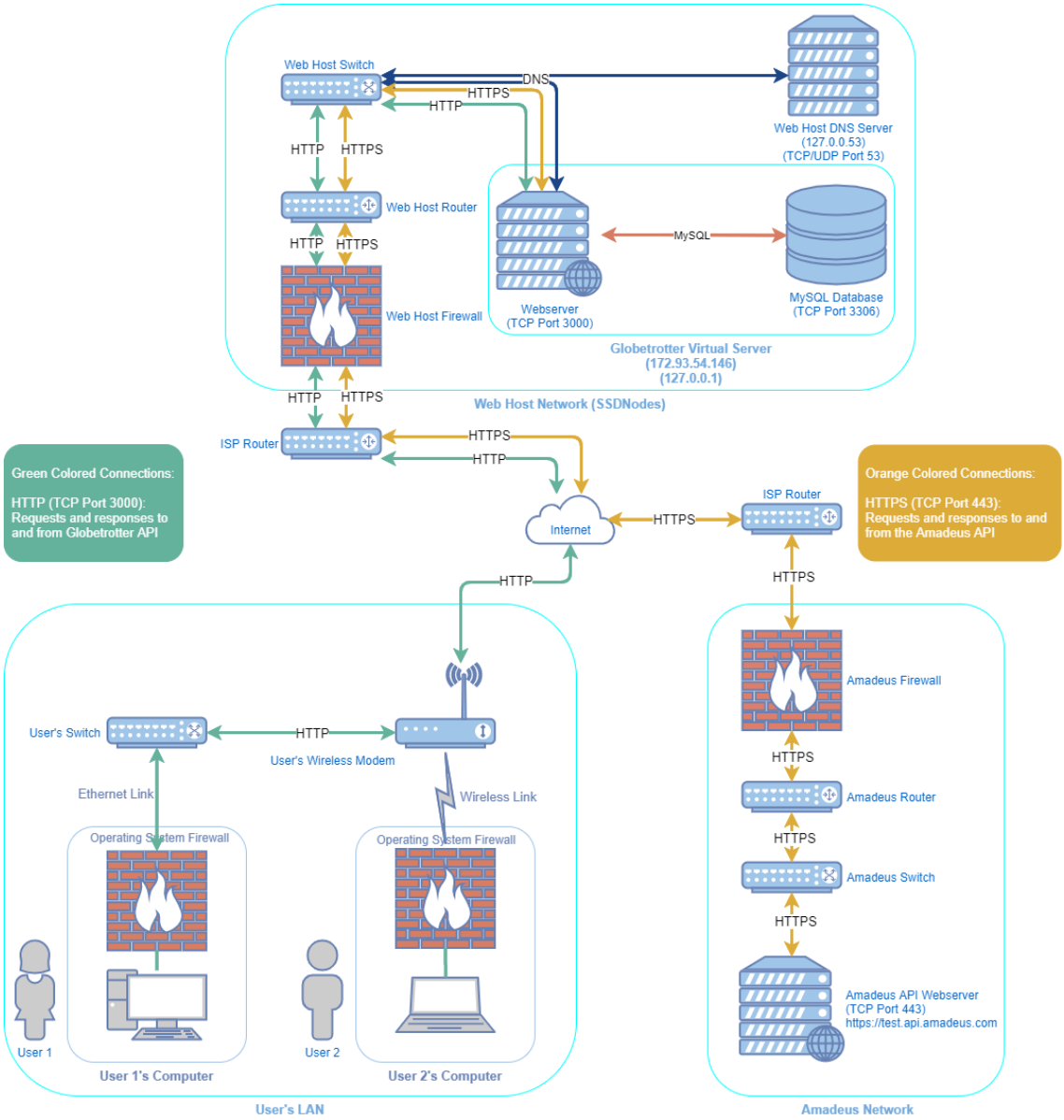
Search Implementation

- We will be using MySQL as our database management system because some of our team members already have some experience with MySQL.
- Photos will be kept in a file system and not stored as blobs.
- The user will be able to search for users by Username or Email.
    - Users will have a dropdown choice os Username and Email to choose from and confirm their choice witht the “next” button.
    - After they have confirmed what they are going to be inputting, the user will input their text in the textbox and press submit to perform the search.
    - SELECT * FROM GlobetrotterV1.users WHERE username LIKE '%${searchString}% will be used to search for users by username where “searchString” is the input of the user.
    - `SELECT * FROM GlobetrotterV1.users WHERE email LIKE '%${searchString}%'` will be used to search for users by email where “searchString” is the input of the user.

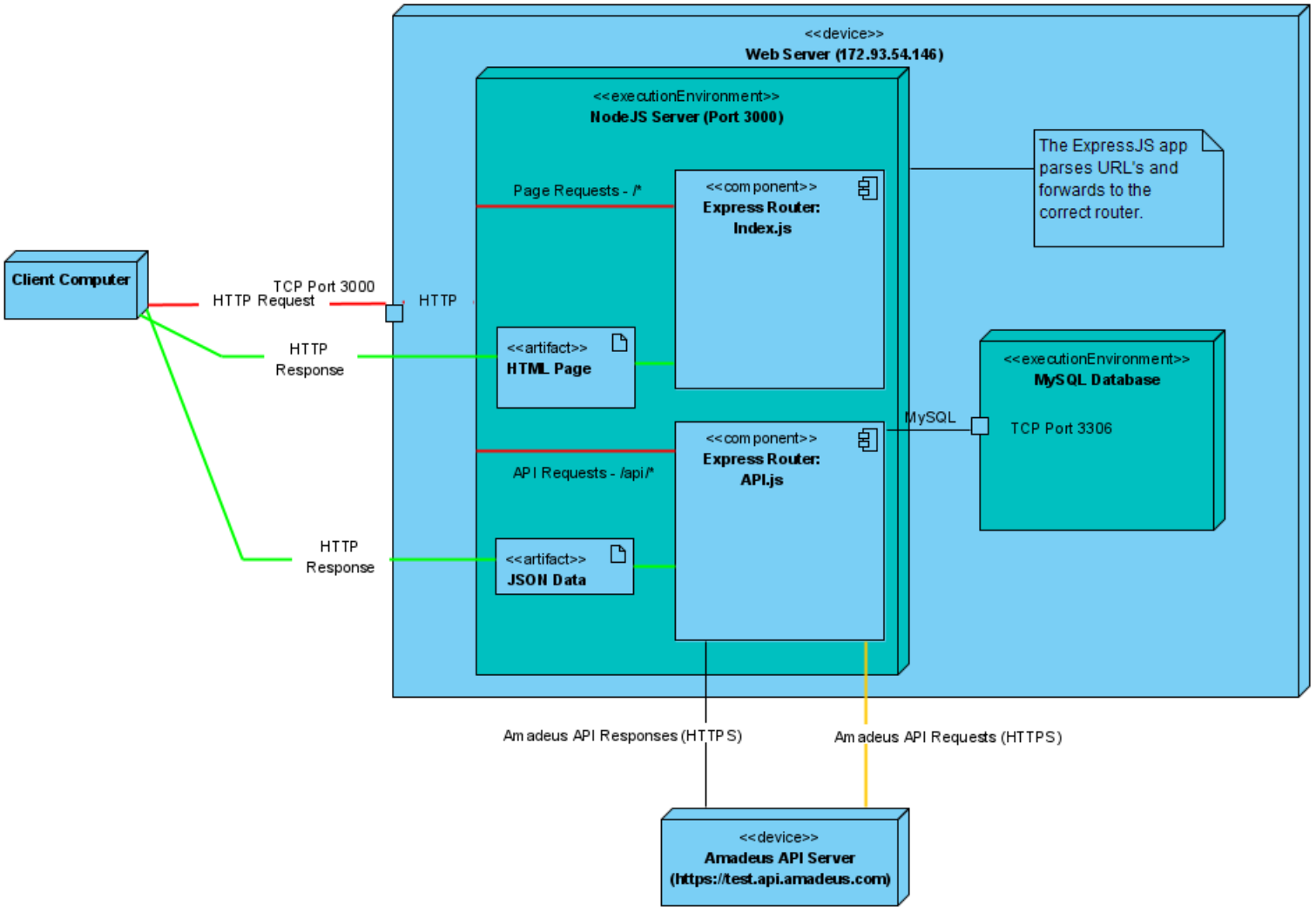
5. UML Diagrams



Network Diagram



Deployment Diagram



6. Contributions

- Ruza
    - Wireframes: Login, Registration, Home, Landing, Flights Results, Saved Flights, Review Flights Detailed, Travelers Info
    - Prototype Implementation: Previous Trips, Registration
- Jesus
    - Wireframes: Photo Gallery, Reset Password, Individual About Page
    - Prototype Implementation: Home, Photo Gallery, Reset Password
- Yi
    - Wireframes: Home, Login, Reset Password, Account Management, Route Planner
    - Prototype Implementation: Login, Account Management
    - Joined all wireframes into master wireframe document
- Kajeme
    - Helped with Implementing ‘Account Management’
    - Helped with ‘Data Definitions’
- Robin
    - Wireframes: Checkout, Saved Trips
    - Prototype Implementation: Checkout, Saved Trips
- Luis
    - Database Architecture
    - Entity Relationship Diagrams
- Taylor
    - Data Definitions
    - UML diagrams

Milestone 2

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	Yi Liu	
Back-End Lead	Luis Alfaro	
Front End	Ruja Rajbhandari	
	Kajeme Cheneque	
	Jesus Correa	
	Robin Fernando	

Revision History

Version	Date Submitted	Date Revised
M2V2	07/22/21	
M2V1	07/08/21	07/22/21
M1V2	07/08/21	
M1V1	06/22/21	07/08/22

Table of Contents

Data Definitions

Entity Name		Entity Description					
	Column Name	**Column Description**	**Data Type**	**Length**	**Primary Key**	**Nullable**	**Unique**
activity		This represents something that a user does during a trip or is planning to do.					
	activity_id	The id of the activity.	varchar	36	true	false	true
	end_location	The id of the location at which the activity ends.	varchar	36	false	true	false
	end_time	The timestamp at which the activity ends.	timestamp	19	false	false	false
	start_location	The id of the location at which the activity starts.	varchar	36	false	false	false
	start_time	The timestamp at which the activity starts.	timestamp	19	false	false	false
	trip	The id of the trip that the activity is part of.	varchar	36	true	false	false
	trip_owner	The id of the trip's owner.	varchar	36	true	false	false
airline		A type of service_provider that sells flights.					

	airline_code	The IATA code of the airline.	varchar	7	true	false	true
	service_provider	The id of the service_provider that the airline represents.	varchar	36	true	false	true
airport		A type of location that belongs to an airport.					
	iata_code	The IATA code of the airport.	varchar	3	true	false	true
	location	The id of the location of the airport.	varchar	36	true	false	true
currency_code		A collection of codes for currencies.					
	code	The ISO 4217 code for the currency.	varchar	3	true	false	false
	name	The name of the currency.	varchar	255	false	false	false
file		This is a file uploaded by a registered_user to our web server.					
	extension	The extension of the file.	varchar	5	false	true	false
	file_id	The unique id of the file.	varchar	36	true	false	true
	file_name	The name of the file.	varchar	255	false	false	false
	folder_path	The folder path of the file.	varchar	255	false	false	false
	full_file_path	The complete path of the file.	varchar	255	false	false	true
	owner	The owner of the fil.	varchar	36	true	false	false

flight_activity		This is a type of activity that represents a flight in the user's trip.					
	activity	The id of the activity.	varchar	36	true	false	true
	flight_offer_json_data	The json data of the selected offer from the airline.	varchar	4095	false	false	false
	trip	The id of the trip.	varchar	36	true	false	false
	trip_owner	The id of the trip owner.	varchar	36	true	false	false
guest_to_trip_association		Associates a user to a trip as a guest.					
	guest	The id of the guest's user entity.	varchar	36	true	false	false
	trip	The id of the trip.	varchar	36	true	false	false
	trip_owner	The id of the trip owner.	varchar	36	true	false	false
home_location		Associates a user with a location as their home.					
	location	The id of the location which represents the registered_user's home.	varchar	36	true	false	false
	user	The id of registered_user which the home_location belongs to.	varchar	36	true	false	true
hotel_activity		This is a type of activity which holds information about a planned hotel stay.					
	activity	The id of the activity.	varchar	36	true	false	true

	hotel_offer_json_data	The json data of the selected offer from the hotel.	varchar	4095	false	false	false
	trip	The id of the trip.	varchar	36	true	false	false
	trip_owner	The id of the owner.	varchar	36	true	false	false
location		An address, a GPS coordinate or both.					
	city	The name of the city.	varchar	255	false	true	false
	country	The name of the country.	varchar	255	false	true	false
	latitude	The GPS latitude coordinate.	decimal	10.7	false	true	false
	location_id	The id of the location.	varchar	36	true	false	false
	location_name	The name of the location.	varchar	255	false	true	false
	longitude	The GPS longitude coordinate.	decimal	10.7	false	true	false
	postal_code	The postal code.	varchar	255	false	true	false
	state	The name of the state.	varchar	255	false	true	false
photo		This is a type of file which is also an image.					
	description	The description of the photo.	varchar	255	false	false	false
	file	The file which the photo represents.	varchar	36	true	false	false
	file_owner	The owner of the file.	varchar	36	true	false	false

	is_profile_photo	Tells whether the photo is a registered_user's profile photo.	bit	1	false	true	false
	photo_id	The unique id of the photo.	varchar	36	true	false	true
	title	The title of the photo.	varchar	255	false	false	false
photo_album		A photo album is created for each trip and is owned by a user. It is a collection of photos.					
	photo_album_id	The unique id of the photo.	int	10	true	false	false
	trip	The id of the trip which the photo_album belongs to.	varchar	36	true	false	true
	trip_owner	The id of the owner of the trip.	varchar	36	true	false	false
photo_to_photo_album_association		Associates a photo to a photo_album.					
	owner	The owner of the photo file.	varchar	36	true	false	false
	photo	The photo which is part of the album.	varchar	36	true	false	false
	photo_album	The album which the photo is part of.	varchar	36	true	false	false
registered_user		User that has registered. They may own trips, files, photos, photo albums, and reviews.					

	password_hashed	The bcrypt hash of the registered_user's password.	varchar	255	false	false	false
	preferred_currency	The registered_user's preferred currency.	varchar	3	false	true	false
	primary_phone_country_code	The registered_user's primary phone country code.	varchar	2	false	true	false
	primary_phone_number	The registered_user's primary phone number.	varchar	10	false	true	false
	secondary_phone_country_code	The registered_user's primary phone number.	varchar	2	false	true	false
	secondary_phone_number	The registered_user's secondary phone number.	varchar	10	false	true	false
	user	The id of the user entity.	varchar	36	true	false	true
	username	The username of the registered_user.	varchar	255	false	false	true
review		A review of a service_provider written by a registered_user.					
	content	The text content of the review.	varchar	255	false	false	false
	reviewer	The id of the registered_user who made the review.	varchar	36	true	false	false

	service_provider	The id of the service_provider that the review is about.	varchar	36	true	false	false
service_provider		A company that sells services to users such as an airline.					
	location	The id of the service_provider's location.	varchar	36	false	true	false
	service_provider_id	The id of the service provider.	varchar	36	true	false	false
	service_provider_name	The name of the service_provider.	varchar	255	false	false	false
service_record		This is a record of a service purchased from a service provider such as a flight or hotel reservation.					
	activity	The id of the activity.	varchar	36	true	false	false
	json_data	The json data provided by the service_provider.	varchar	510	false	true	false
	service_provider	The id of the service_provider.	varchar	36	true	false	false
	service_record_id	The id of the service_record.	varchar	36	true	false	true
	trip	The id of the trip.	varchar	36	true	false	false
	trip_owner	The id of the trip owner.	varchar	36	true	false	false
trip		A trip holds activities and is named.					

	owner	The id of the trip's owner.	varchar	36	true	false	false
	trip_id	The unique id of the trip.	varchar	36	true	false	true
	trip_name	The unique name of the trip.	varchar	255	false	false	true
user		Unregistered user. They may be guests of trips.					
	email	The unique email of the user.	varchar	255	false	false	true
	user_id	The unique id of the user.	varchar	36	true	false	true
usersessions		Holds session information for logged in users.					
	data	The data associated with the logged in user's session.	mediumtext	0	false	true	false
	expires	The timestamp of when the session expires.	int	10	false	false	false
	session_id	The id of the user_session.	varchar	128	true	false	false

Prioritized Functional Requirements

Priority 1

1. Registered users can edit their birthday.
2. Registered users can edit their email.
3. Registered users can edit their home city
4. Registered users can edit their home postal code.
5. Registered users can edit their home state
6. Registered users can edit their home street address.
7. Registered users can edit their password.
8. Registered users can edit their preferred currency.
9. Registered users can edit their primary phone number.
10. Registered users can edit their profile photo.
11. Registered users can edit their secondary phone number.
12. Registered users can edit their username.
13. Registered users can view the ‘Account Management’ page.
14. Unregistered users cannot access Account Management.
15. All registered users can view the cost of the trip when checking out.
16. All registered users can view the flights they are purchasing when checking out.
17. Unregistered users cannot access Checkout.
18. Registered users can login using the ‘Login’ page.
19. Users shall provide their password when logging in.
20. Users shall provide their username when logging in.
21. Registered users can reset their password from the ‘Login’ page by sending a reset link to the account’s email.
22. Unregistered users can view the ‘Registration’ page.
23. Login page will have a ‘forgot password?’ button.
24. The password field will be obscured on the Login page.
25. All users can contact the team.
26. All users can search for flights.
27. All users can view the “Meet the Team” page to find out more information about the team.
28. Registered users can delete photos they uploaded.
29. Registered users can upload pictures of their trip.
30. Unregistered users cannot access Photo Albums.
31. Unregistered users cannot access Previous Trips.
32. Unregistered Users shall provide their date of birth when creating an account
33. Unregistered users shall provide a password when creating an account.

34. Unregistered users shall provide a primary phone number when creating an account.
35. Unregistered users shall provide a username when creating an account.
36. Unregistered users shall provide their email when creating an account.
37. Unregistered users shall provide their first name when creating an account.
38. Unregistered users shall provide their home city when creating an account.
39. Unregistered users shall provide their home postal code when creating an account.
40. Unregistered users shall provide their home state when creating an account.
41. Unregistered users shall provide their home street address when creating an account.
42. Unregistered users shall provide their last name when creating an account.
43. Unregistered users will be able to create an account.
44. All registered users can add at least one destination.
45. All registered users can add multiple Activities to a Trip.
46. All users can choose from the available flights.
47. All registered users can edit the destination of a trip.
48. All registered users can edit the starting location of a trip.
49. All users can edit the displayed currency.
50. All registered users can remove the destination of a trip.
51. All registered users can remove the starting location of a trip.
52. All registered users can remove Activities from a Trip.
53. All Registered Users can specify a return date.
54. All registered users shall specify the dates of departure for each flight.
55. All users will be able to view arrival times for the available flights.
56. Unregistered users can access the Route Planner.
57. Registered users can delete trips that they previously created.
58. Registered users can edit trips that they previously created.
59. Registered users can purchase their Saved Trips.
60. Unregistered users cannot access Saved Trips.
61. Registered users can view trips that they previously created.
62. The displayed currency will default to the preferred currency of a logged in user.
63. The website will be easy for all Users to maneuver through.

Priority 2

64. Unregistered users shall provide their gender when creating an account.
65. Registered guests may purchase trips on which they are guests.
66. Registered guests will receive a confirmation email after checkout.
67. All guests added to the trip will be sent an email containing a link to the trip.
68. All users will be able to click on individual team images to find out more information about the specific member.

69. Registered users can add a photo description when uploading a picture.
70. Registered users can edit their photo descriptions.
71. Registered users can edit titles of pictures they uploaded.
72. Registered users shall give a title when uploading a picture.
73. Users will be sent a confirmation email after completing registration.
74. Registered users can create a review of their flight.
75. Registered users can delete their reviews.
76. Registered users can edit their reviews.
77. All users can look at reviews of Airlines.
78. All guests can view trips on which they are guests.
79. All users may add entertainment activities to a particular day of a trip.
80. All users may add hotel reservations to a particular day of a trip.
81. All users will be able to view weather information for each location.
82. Dates added to a trip may not overlap.
83. Registered users can edit trips on which they are guests.
84. Registered users may cancel flights which they purchased.
85. Registered users can be issued a refund for flights cancelled within 48 hours of departure.
86. Registered users can receive credit for flights cancelled within 48 hours of departure.
87. Users may export the schedule prepared by Globetrotter to a file.
88. Users may print the schedule prepared by Globetrotter.

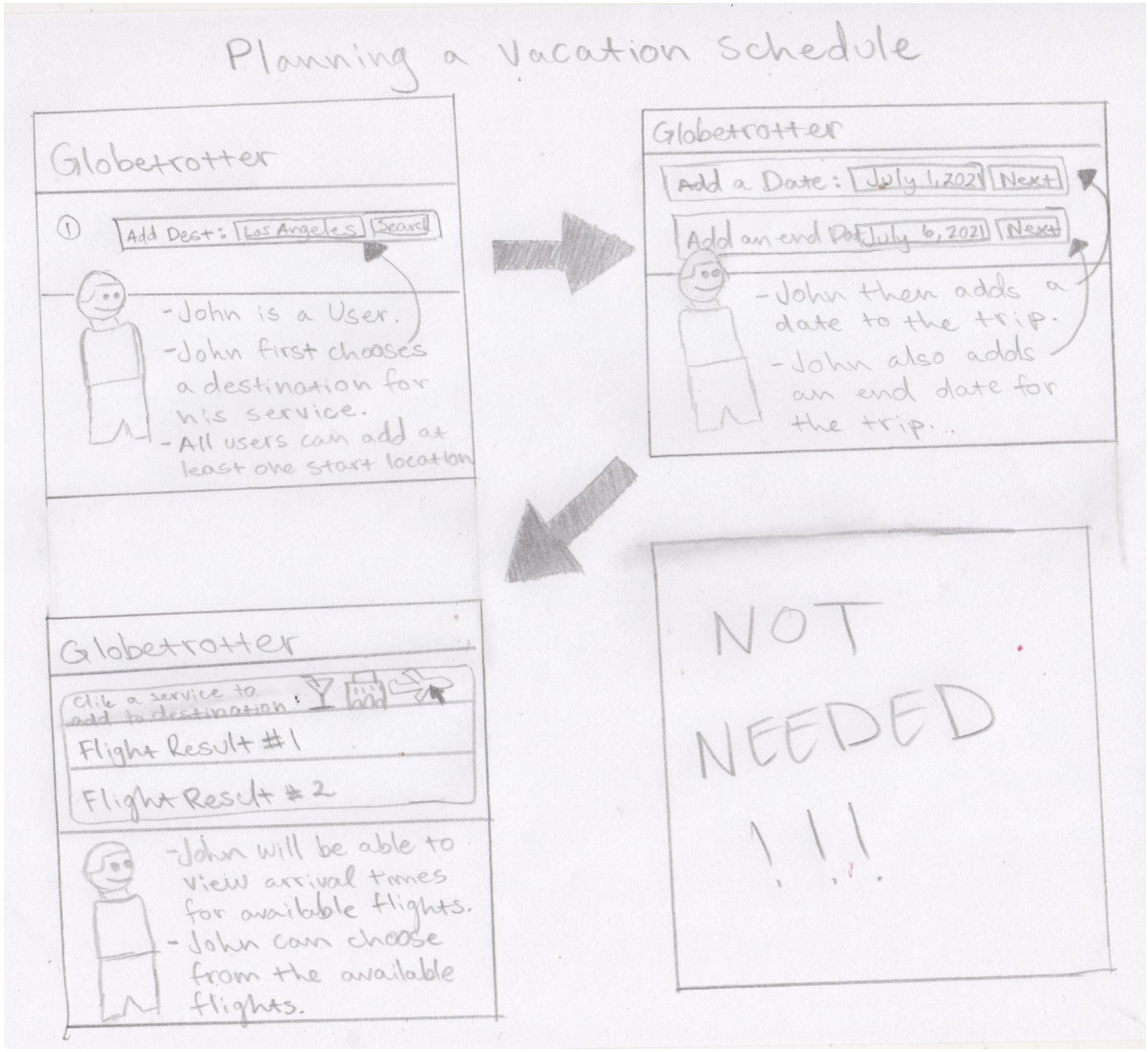
Priority 3

89. All registered guests can add guests to the trip.
90. All registered guests can add multiple flight stops in between the start and destination locations (waypoints).
91. All registered guests can add multiple flight stops in between the start and destination locations.
92. All registered guests can add other registered users to the trip.
93. All registered guests can add unregistered guests to the trip.
94. All registered guests can edit the destination of a trip.
95. All registered guests can edit the starting location of a trip.
96. All registered guests shall specify the dates of departure.
97. All registered guests can only add up to nine registered users to the trip.
98. All registered guests can only add up to one registered user at a time.
99. All registered guests can remove the destination of a trip.
100. All registered guests can remove the starting location of a trip.
101. All registered guests can remove waypoints.
102. Registered users will receive a confirmation email after checkout.
103. Unregistered guests may not purchase trips on which they are guests.

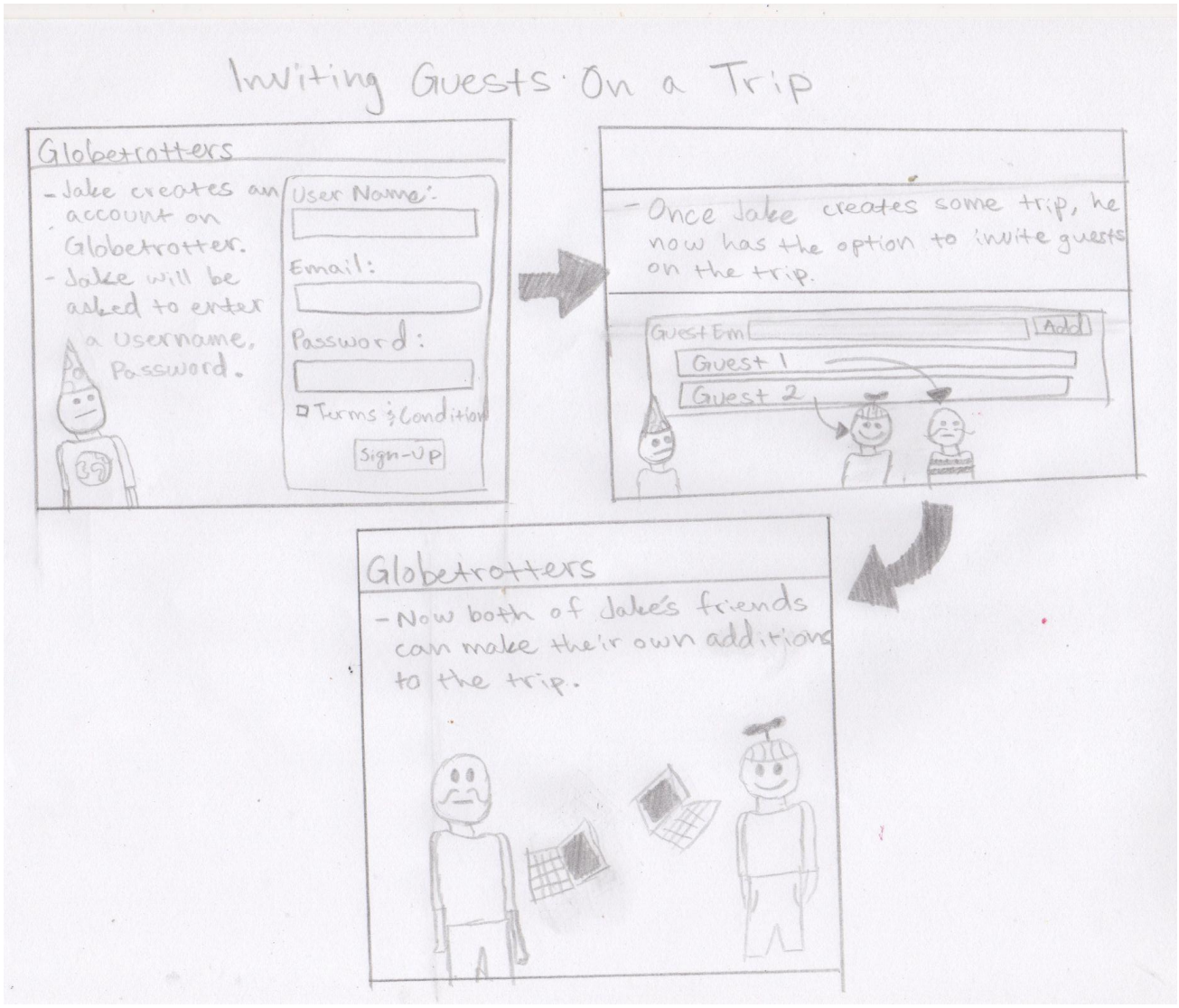
104. All registered guests can add at least one starting location.
105. All users shall specify a budget.
106. Registered users can reserve other registered guests to hotels.

UI Mockups & Storyboards

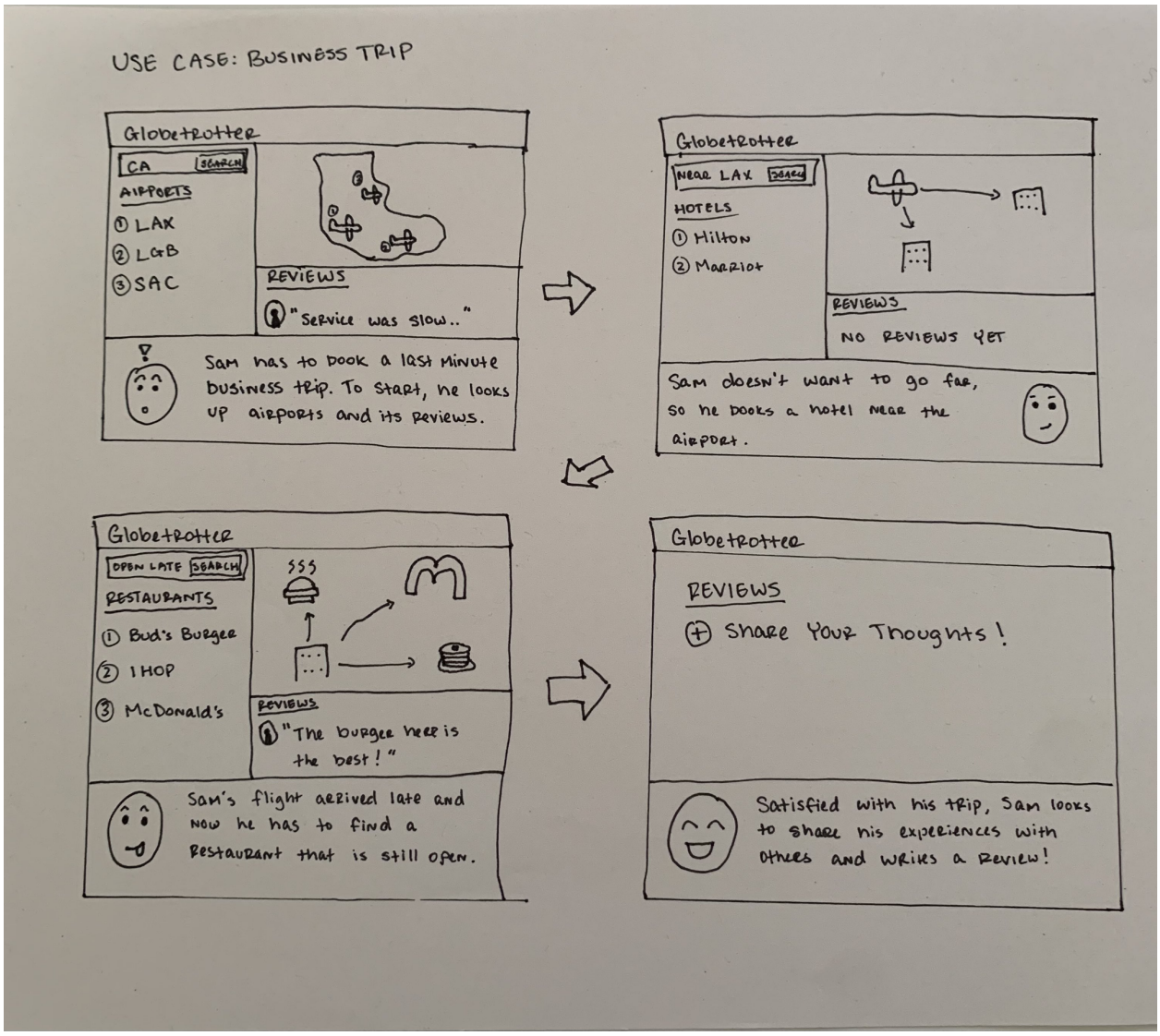
Planning a Vacation Schedule



Inviting Guests on a Trip



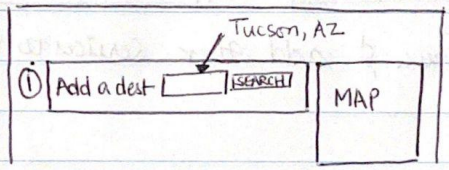
Business Trip



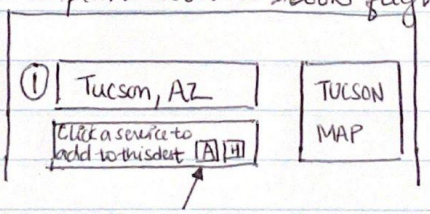
Activity Itinerary (Focused on booking/adding flights to itinerary)

Bill creates an account & starts creating a new trip.

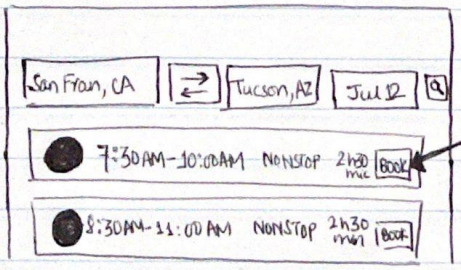
Bill is taken to a new trip screen, where he will enter Tucson, AZ as the destination



After adding Tucson as destination, Bill will be shown the map of Tucson & he will get options to click on services he wants to add for that destination. He clicks the airplane icon to ~~book~~ for flights to the destination (Tucson)

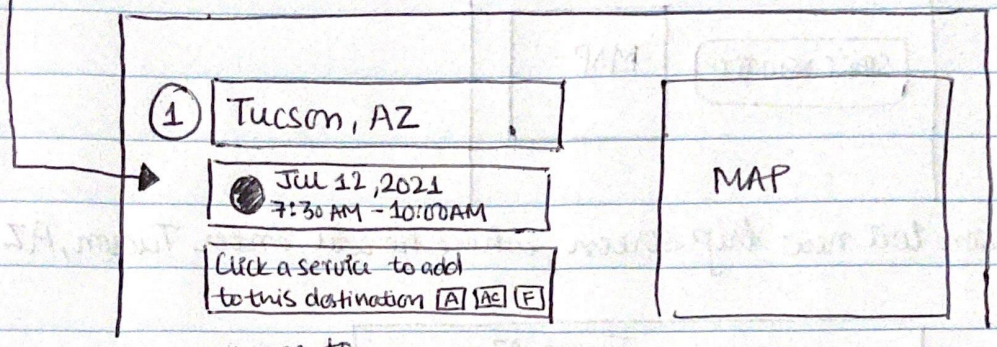


He is taken to add a flight screen, where he will get to enter the arrival & departure destination & dates. He will be able to see all the available flights & he can choose to book flights from the ~~results~~ displayed.



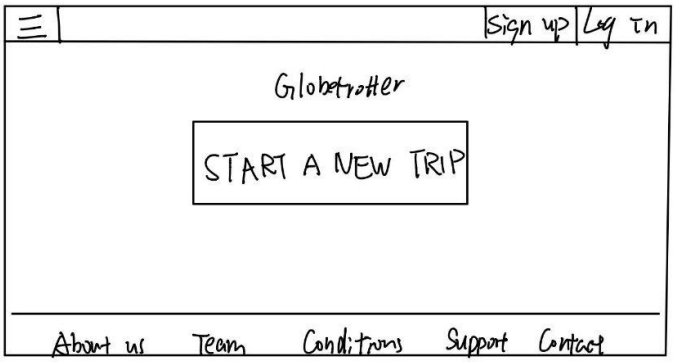
He books the 7:30AM flight.

After he books his desired flight, he will be able to see the flight with details on his itinerary screen.



Now Bill can choose to continue & add other services to his trip.

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

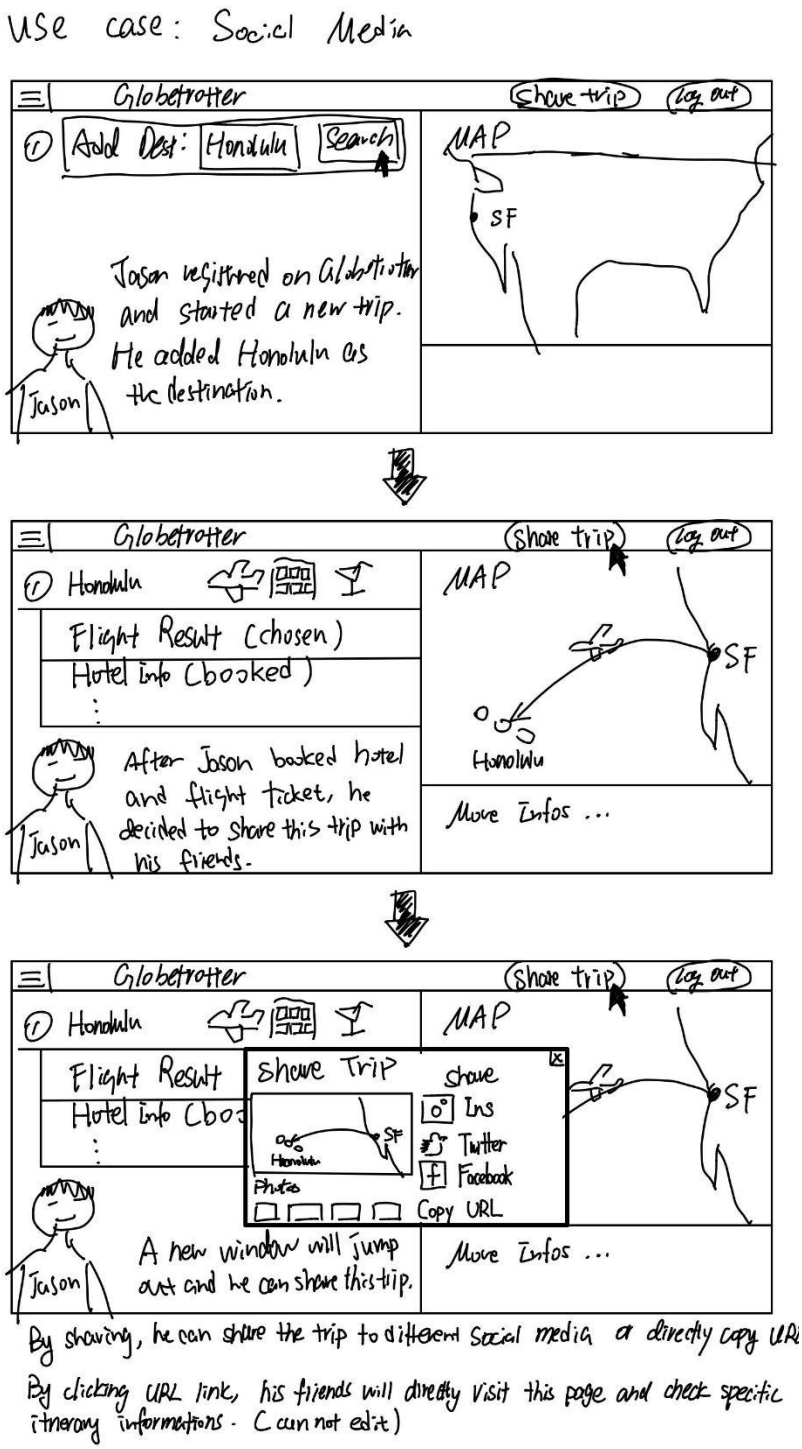
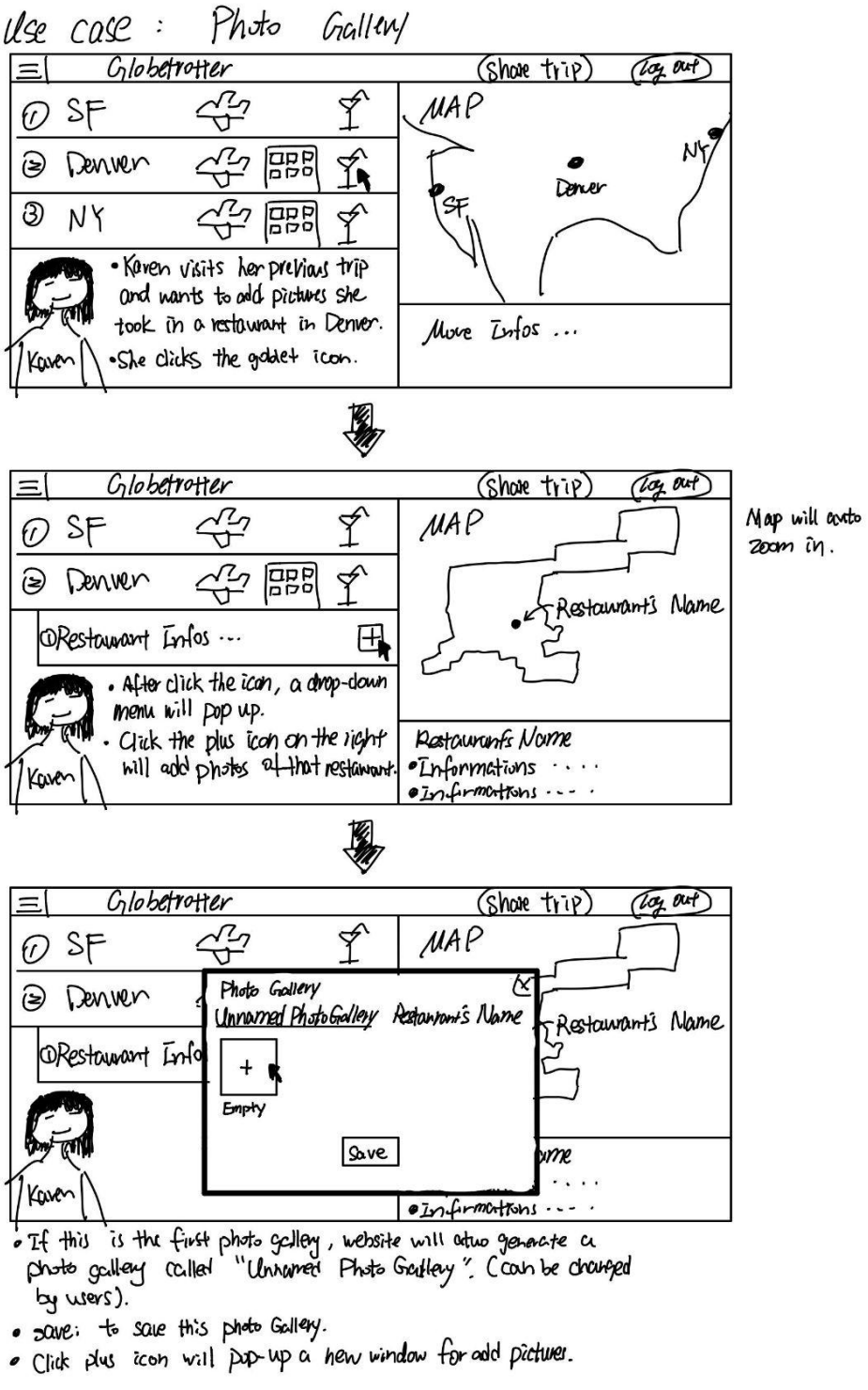
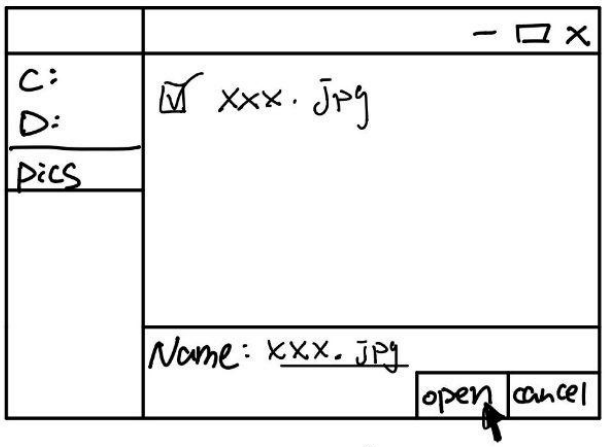
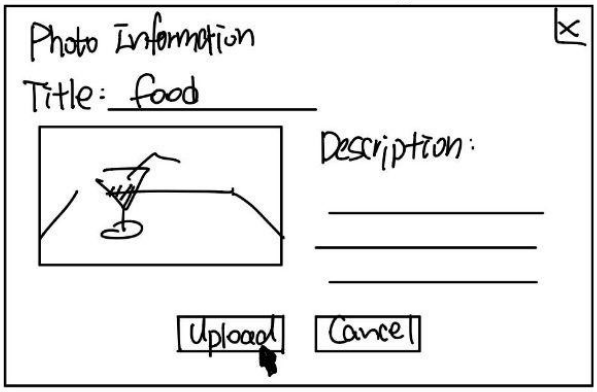


Photo Gallery

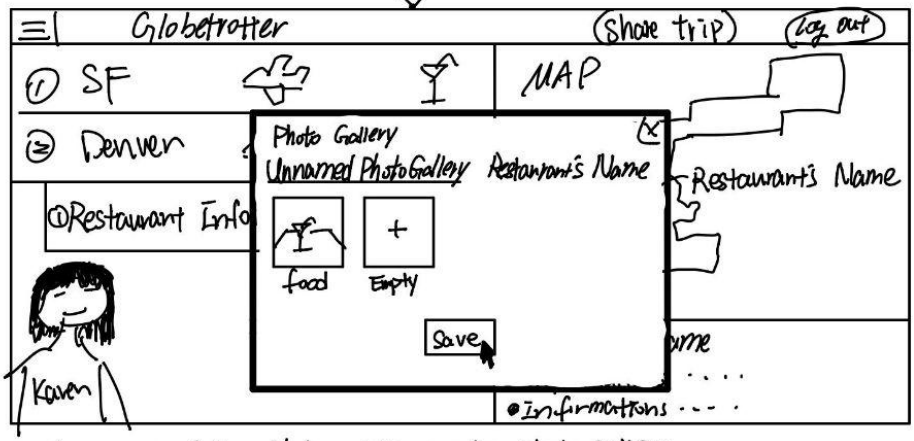




- Then Karen can browse her PC folders and choose picture to upload.



- 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

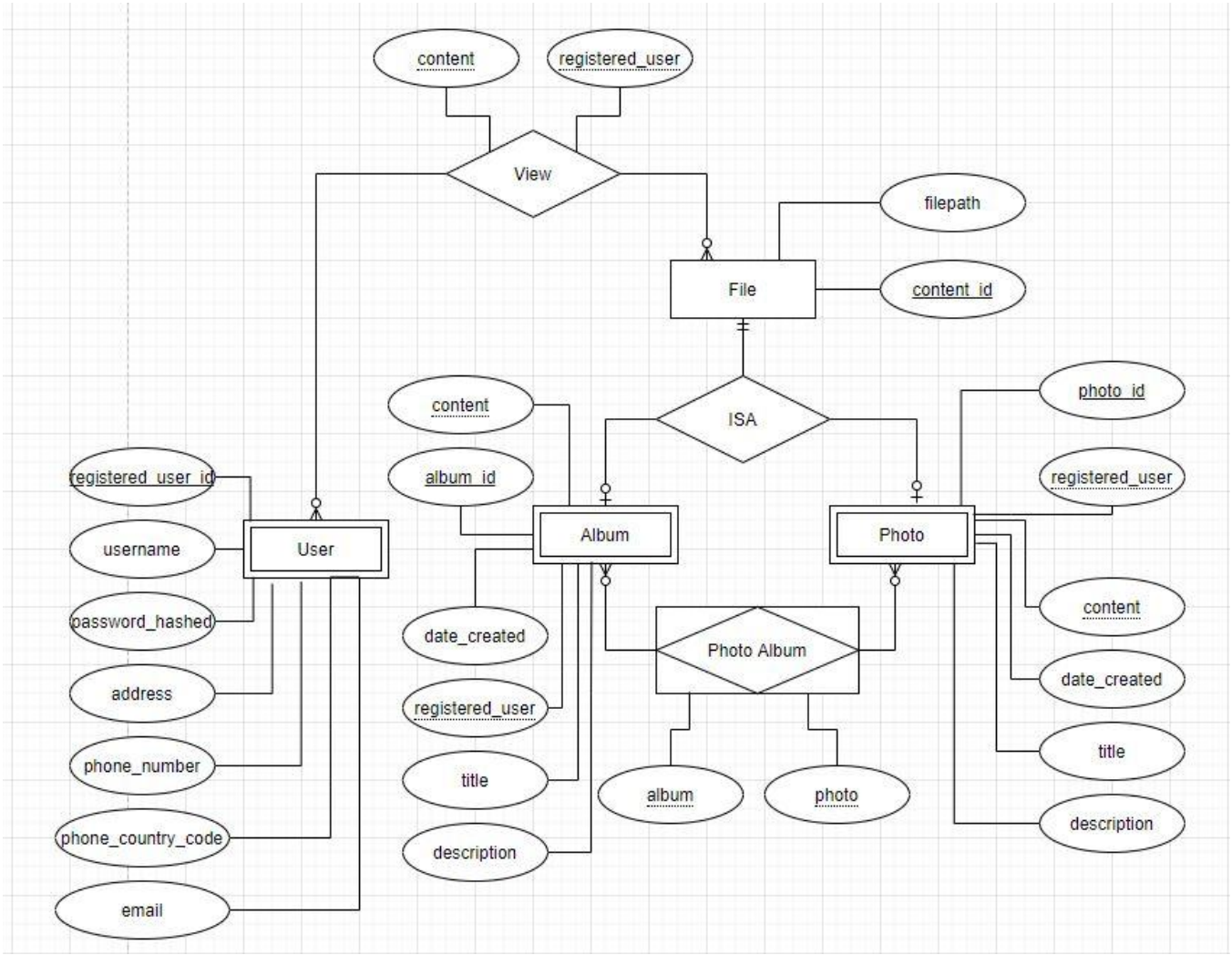
High Level Database Architecture & Organization

Business Rules for Photo Albums

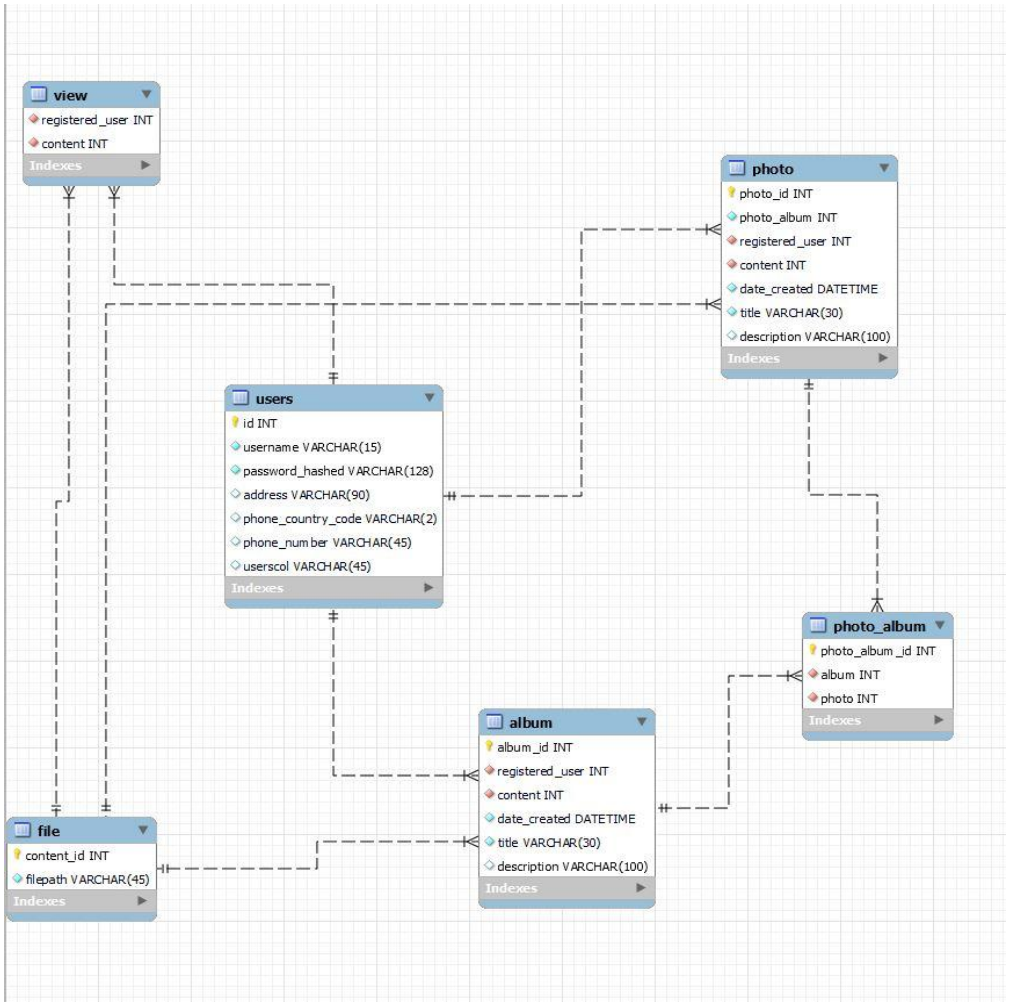
Entities, Attribute, Relation, Quantitative

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.

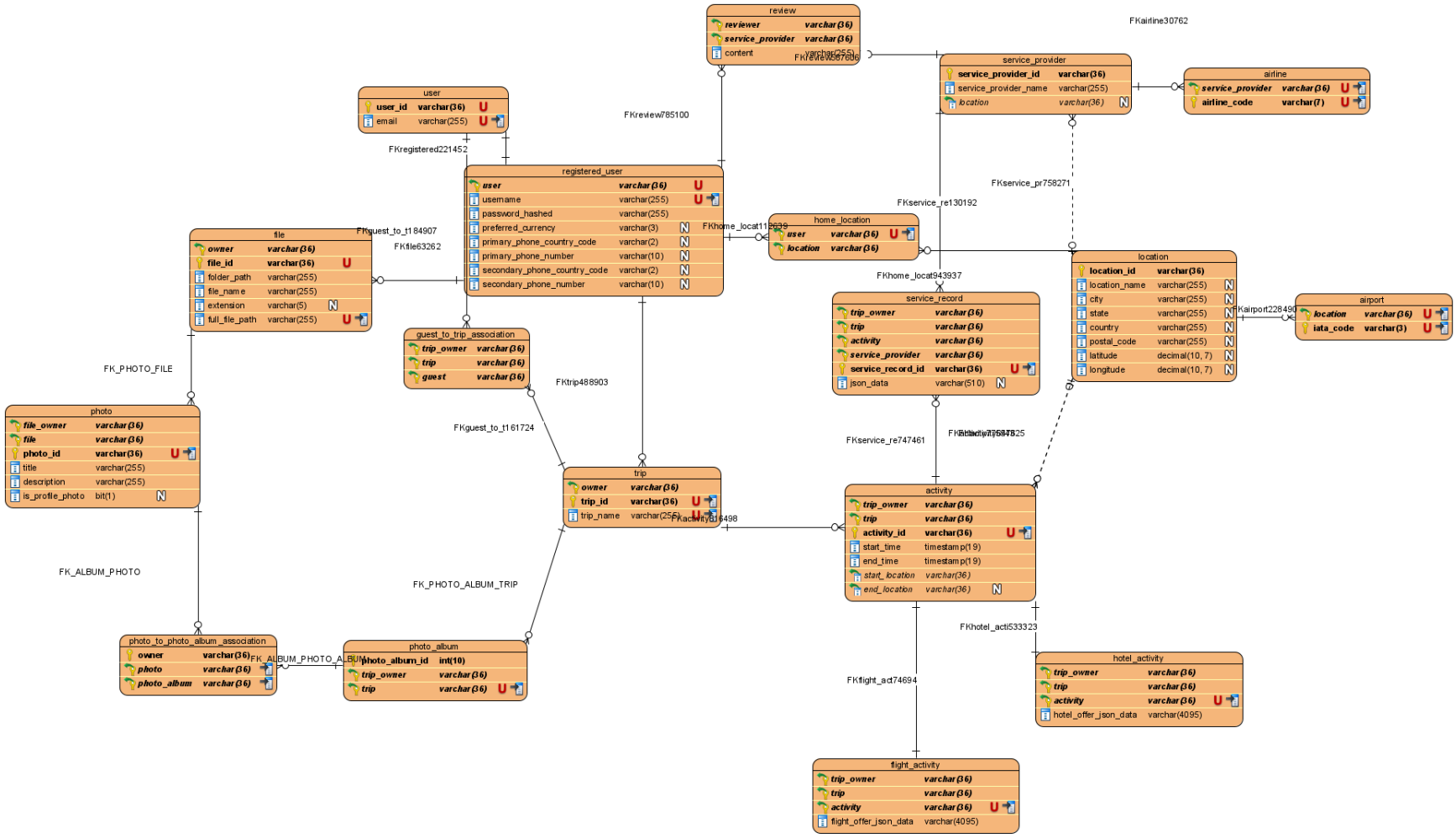
Entity Relationship Diagram (Photo Albums)



Database Model (Photo Albums)



Database Model (Full)



Search Implementation

- We will be using MySQL as our database management system because some of our team members already have some experience with MySQL.
- Photos will be kept in a file system and not stored as blobs.
- The user will be able to search for users by Username or Email.
    - Users will have a dropdown choice os Username and Email to choose from and confirm their choice witht the “next” button.
    - After they have confirmed what they are going to be inputting, the user will input their text in the textbox and press submit to perform the search.
    - SELECT * FROM GlobetrotterV1.users WHERE username LIKE '%${searchString}% will be used to search for users by username where “searchString” is the input of the user.
    - `SELECT * FROM GlobetrotterV1.users WHERE email LIKE '%${searchString}%'` will be used to search for users by email where “searchString” is the input of the user.

High Level APIs & Main Algorithms

Our API's

Auth

- Registration:
    - POST Request**:** The unregistered user will be asked to enter a user name and password for their account. The password will be hashed and the username, along with the hashed password, will be stored in the database.
- Login:
    - POST Request**:** All users will be required to enter a username and a password. If the username exists in the database and the hash of the supplied password matches the 'password_hashed' field of that user, the user will be given a token that will serve as proof that they have been authenticated. A user session will be created for the user in the database.

Users

- Search Users
    - GET Request: The Registered User will supply either a username or email and will be returned a list of Registered Users with matching values.
- Get Me
    - GET Request: The Registered User will supply their access token which the server will use to identify them. After the user has been identified, they will be returned their user object.

Trips

- My Trips
    - GET Request: The Registered User will supply their Access Token which the server will use to identify them. After the user has been identified, all Trips belonging to the User will be returned.
- Get Guests
    - GET Request: The Registered User will supply their Access Token as well as a Trip ID. After the Trip is found in the database, all User objects for Guests of that Trip will be returned.
- Add Guest

    - POST Request: The Registered User will supply their Access Token, a Trip ID, and a User ID. This will add the User with the supplied User ID as a Guest to the Trip with the provided Trip ID.

Reviews

- My Reviews
    - GET Request: The Registered User will supply their Access Token, and be returned a list of Reviews which they have made.
- Create Review
    - POST Request: The Registered User will supply their Access Token, a ServiceProvider ID, and a string. This will add a new Review object in the database which will be associated with the Service Provider.
- Delete Review:
    - DELETE Request: The Registered user will be able to delete a review. They will first have to get the request that they wish to delete. The database will return the review that they searched for and then they will have the option to delete their review.

Photo Albums

- Create Photo Album for Trip
- Get Photo Album for Trip
- Create Photo in Photo Album
- Get Photo from Album
- Delete Photo from Album

Flights

- Search
    - GET Request**:** All users will be able to search for flights. The users will be asked to enter some information, such as: Origin Location Code, Destination Location Code, departure date, number of passengers, currency code, and the number of results desired. After the information is acquired, it will be sent to the Amadeus “Flight Offer Search” API which will return a list of flights based on the criteria given by the user.

Third Party API’s

The APIs that follow are used to obtain information about hotel rooms and flights, as well as to perform booking.

Flights

We decided to go with the Amadeus Air APIs because these APIs returned a lot of information that would be good for the user to know. It also took care of certain things that we would not have to implement on our own. For example, if a return date is specified, the API would return round trips instead of us having to filter out the trips we want to display. The results are already filtered when being returned by the API.

Amadeus AIR APIs

- Flight Offer Search
    - GET Request**:** Information entered by the User, when they do a GET request, will be sent to the Amadeus Flight Offer API. This API will return a list of flights based on the criteria given by the user.
- Flight Offers Price
    - GET Request**:** Once the user searches for flights using the Flight Offer Search endpoint, a GET request will be made to the Flight Offers Price endpoint automatically. This endpoint will return the price for any flights that were returned by the Flight Offers Search endpoint. Once the prices have been retrieved, the flights, along with their prices may be displayed.

Hotels

We chose the Amadeus Hotel APIs because they are the most complete hotel searching APIs that we found. Using these APIs we are able to get detailed information about room availability, ratings, and prices, for a variety of hotels.

Amadeus Hotel APIs

- Hotel Search
    - GET Request: When calling this API the following parameters are supplied: either a city code or GPS coordinate, a search radius, check-in and check-out dates, room quantity, a currency code, as well as the number of adults. After the request is made, the API returns a list of **Hotel Offers**, each containing a list of room offers from a particular hotel, as well as information about the hotel.
- Hotel Booking
    - POST Request: When calling this API, the request body should contain the **Hotel Offer** ID, a list of guests, and payment information. The API will return an array for each successful booking, each containing a ‘providerConfirmationId’ and an array of ‘associatedRecords’.

Amadeus SDK

To simplify the process of using the Amadeus APIs, we intend to use the Amadeus Node SDK. This SDK is updated by Amadeus and hosted on Github at https://github.com/amadeus4dev/amadeus-node

Main Algorithms

- The app will mainly use sorting and filtering algorithms to present the data. These will appear when:
    - Viewing flights in order of price
    - Filtering flight ticket results by date and time

-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.

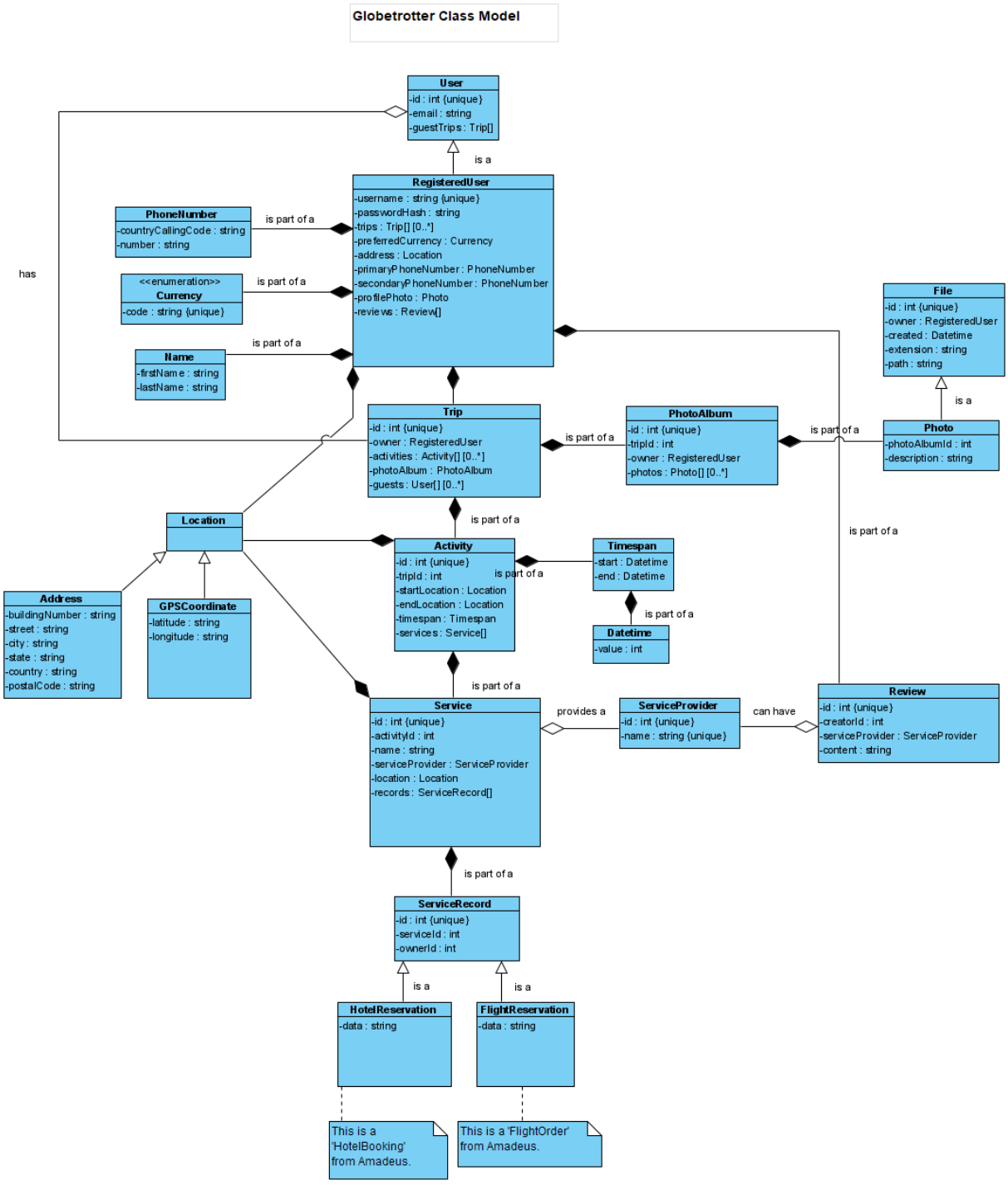
-If the User searches for a Trip that was created previously, the backend will show a Trip that matches the searched input given by the User.

-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.

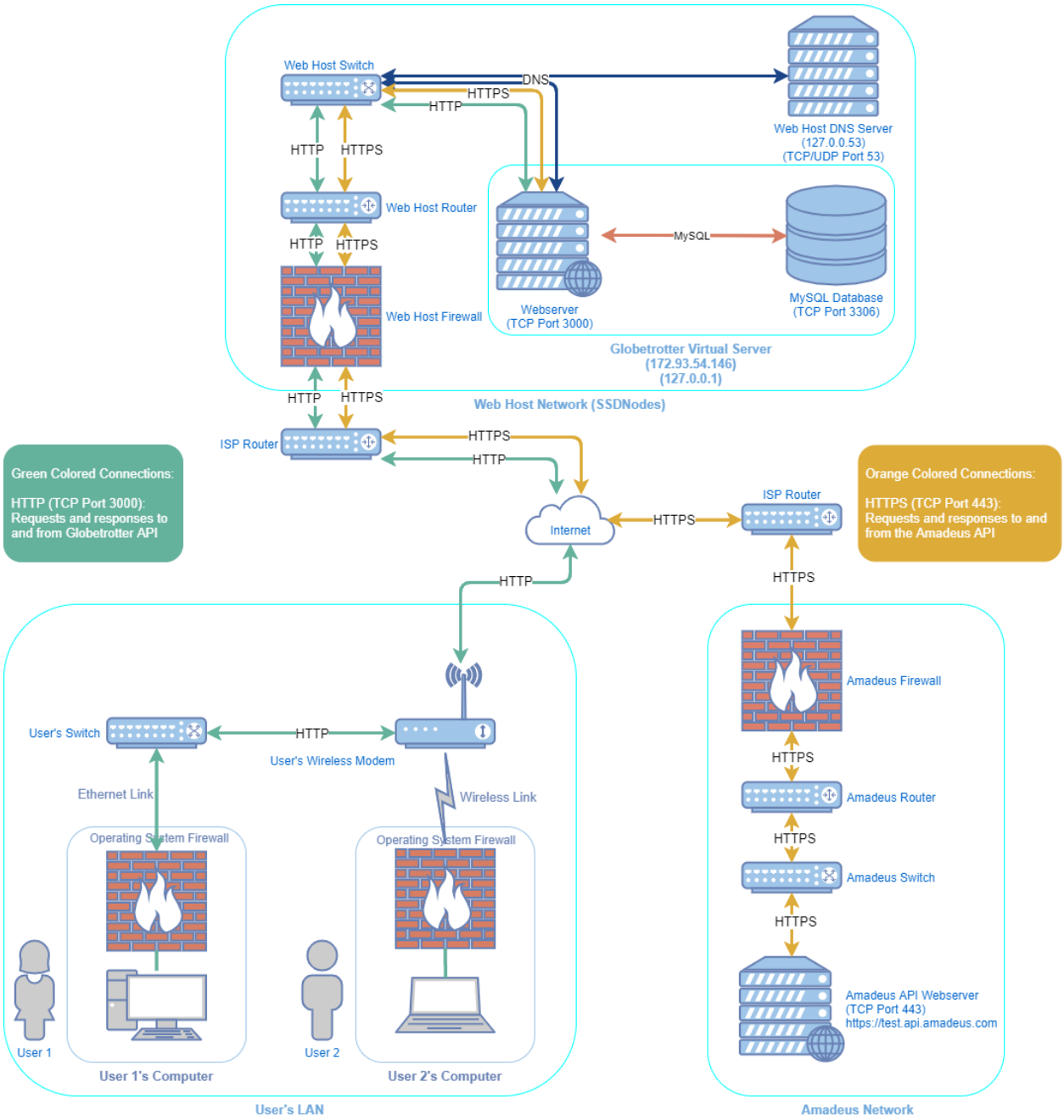
High Level UML Diagrams

Data Model

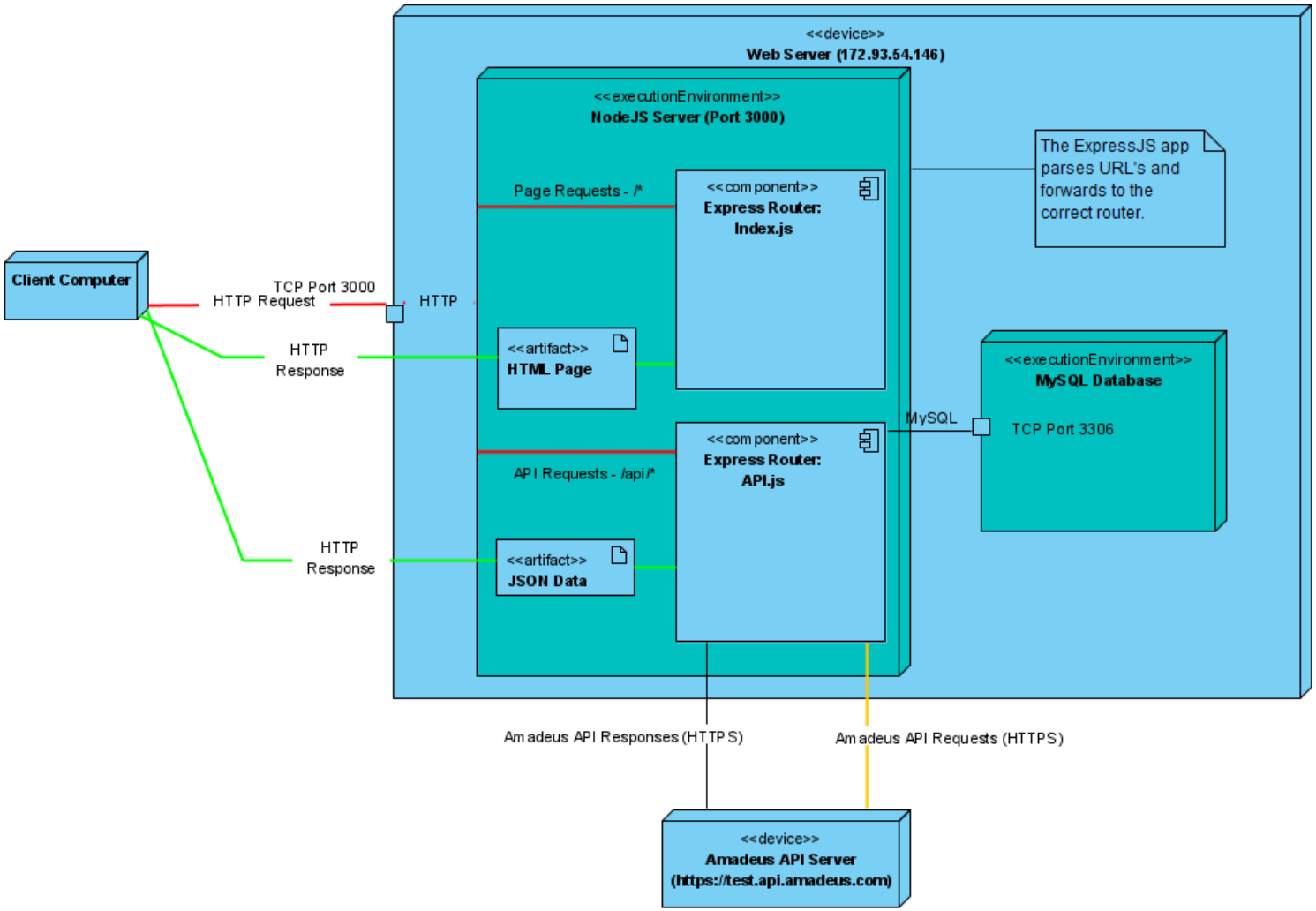


High Level Application Network & Deployment Diagrams

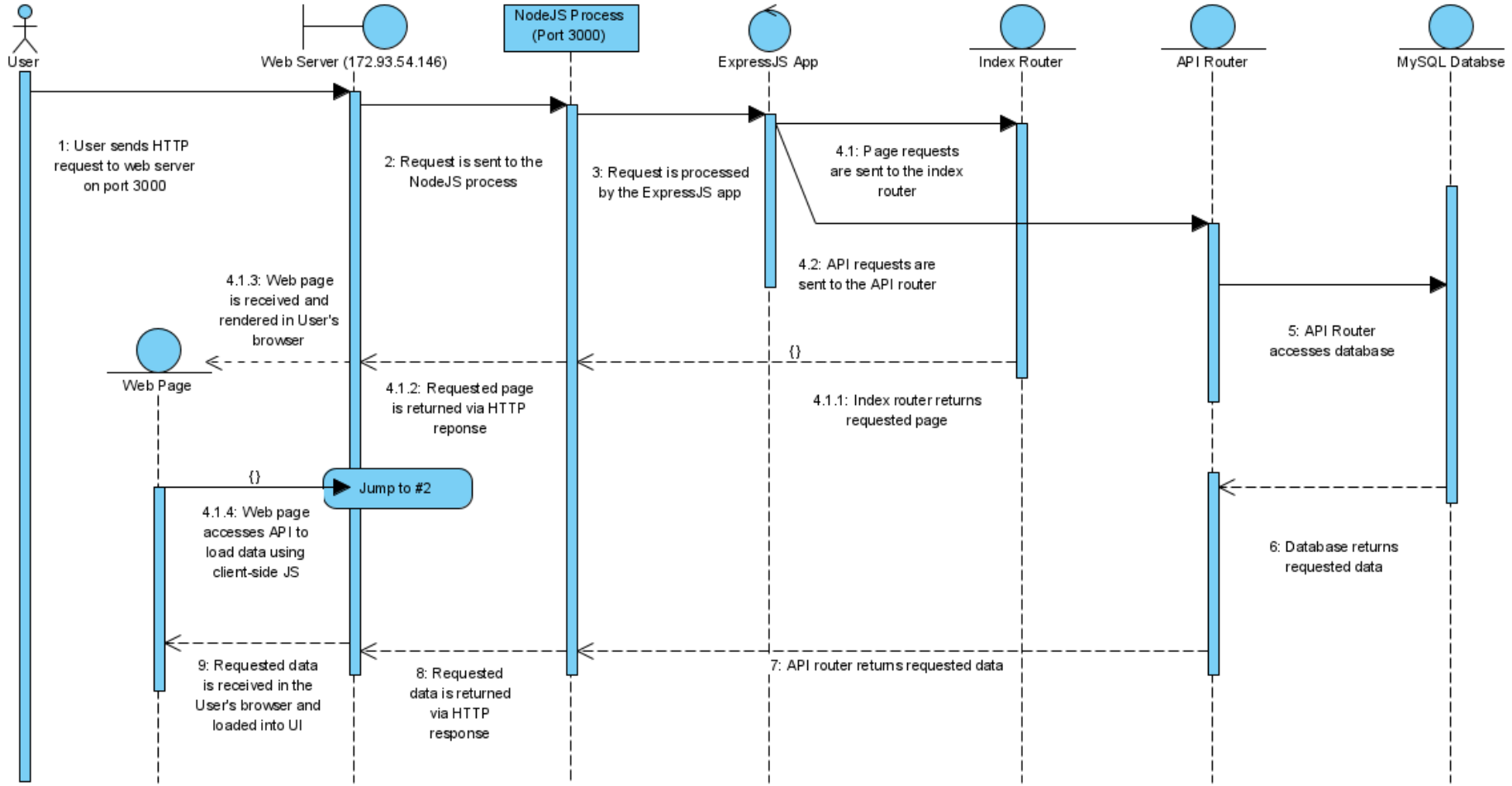
Network Diagram



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

Ruja

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

Milestone 1

Team 2 – Summer ‘21
CSC648
June 22, 2021

Globetrotter Travel Assistant

Milestone 1

Role	Name
Team Lead/Github Master	Taylor Artunian
Front-End Lead	Kajeme Cheneque
Back-End Lead	Luis Alfaro
Front End	Ruja Rajbhandari
	Yi Liu
	Jesus Correa
	Robin Fernando

Revision History

Version	Date	Notes
M1V2	07/08/22	
M1V1	06/22/21	07/08/22

Table Of Contents

Executive Summary

Motivation

Travel has become more convenient for the average person in recent decades. Services like Uber and Lyft have allowed travelers to summon a taxi just about anywhere, OpenTable has made it easier to find a table at a restaurant without calling around, and apps from the major airlines have made it possible to book a plane ticket from a smartphone or computer. The result is that a traveler may choose either plan their travel reservations months, hours, or even minutes before they need them.

For the extended trip though, one that is planned farther into the future, organizing these reservations can be tricky. A multi-week trip may have four or five hotel bookings, several restaurant reservations, and multiple Uber rides each day. Traditionally, a travel agent would fill this role, making the arrangements in advance and preparing an itinerary for the traveler. However, hiring a travel agent today is still cost prohibitive for the average traveler. Many, if not most, would benefit from an affordable online service that lets them book and organize their travel reservations in one place.

How it works

Our plan with *Globetrotter* is to create a user interface for planning, booking, and organizing travel arrangements. This interface will be centered around a visual schedule in which users can insert their destinations, dates of stay, and services for which they will make reservations. With this strategy, the user will be able to see their entire trip chronologically making them feel confident and organized.

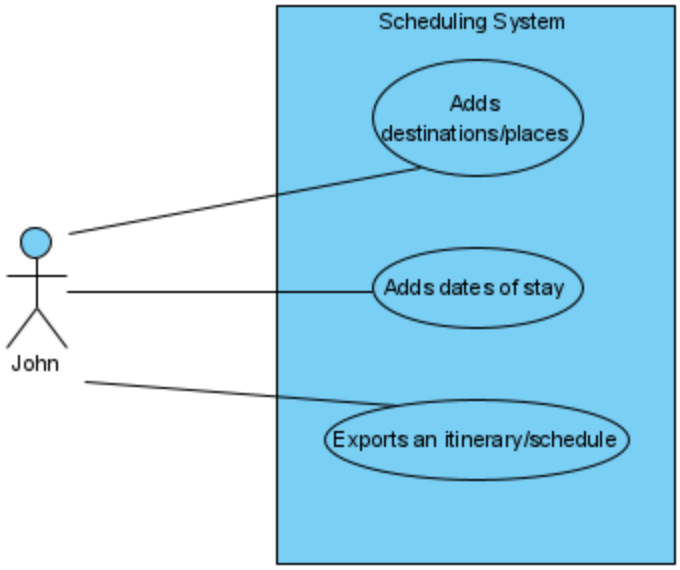
In addition to the improved user interface, Globetrotter will also be feature-rich. Many of the previously mentioned services (such as Uber and OpenTable) provide APIs which make them easy for developers to integrate with. Globetrotter will be designed such that it is easy to provide integrations for more third-party services over time.

Business Plan

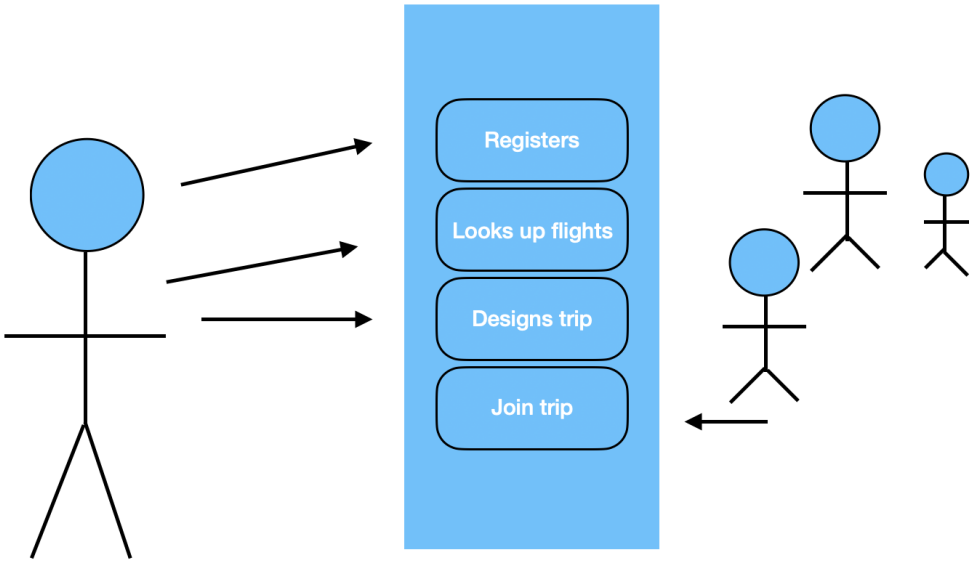
The business plan for Globetrotter will use a mixture of enticing users with free features and charging for other features. This is used in many software applications to create a base of consistent users of the system. Some users will only use the system for the free features. Eventually, though, many of them are willing to pay for the ones that enhance the experience of using the application. By selling features a la carte, the user remains in control of paying only for the features they use. However, we would also add on the ability to purchase a full-featured subscription for users that wish to have access to all the features. This would be priced higher than two or three features on their own but less than the price of all the features combined.

Use Cases

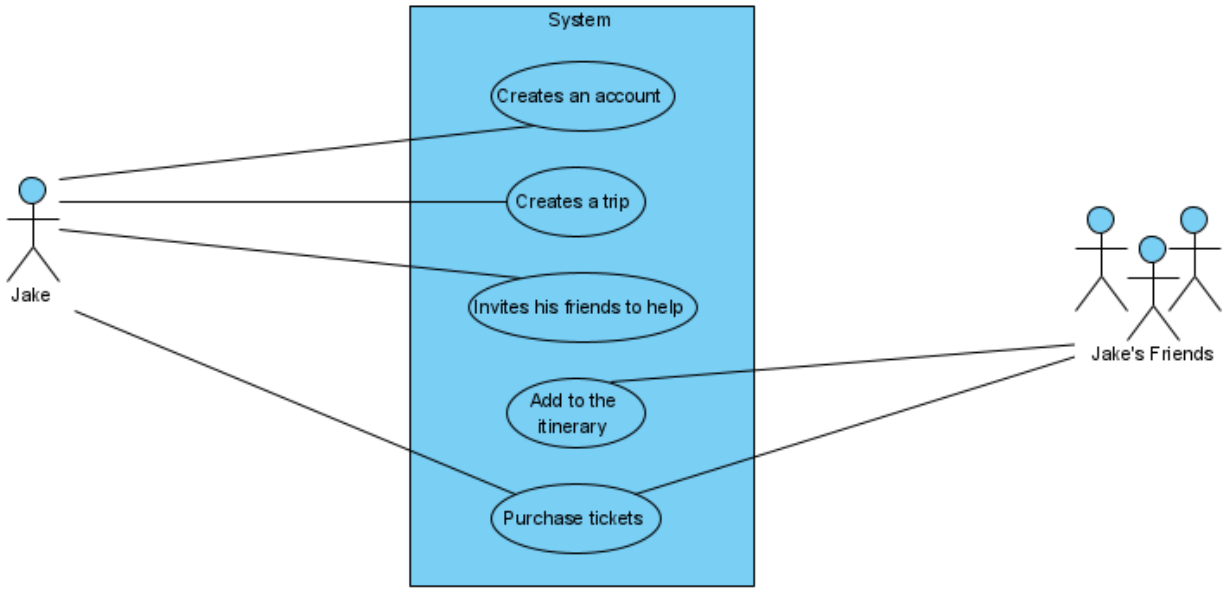
Title:	Planning a Vacation Schedule
Actors:	John
Description:	John is planning a vacation to Northern California for his family of four. They plan to fly from Los Angeles to San Jose on the morning of July 1 and visit the Intel Museum. After staying overnight in San Jose, they would like to check out San Francisco State University because the oldest child is considering attending there for school. They would like to stay somewhere close to the school for one night. On the third day, the family would like to visit the Golden Gate Bridge and also see a Warriors basketball game at Oracle Park. Since this is a trip spanning multiple days and including multiple people, John wants to stay organized with all the services he needs to use. He needs one application to plan the trip's schedule, and make each day's reservations for hotels, restaurants, etc. In searching for an online trip planning service, though, he finds that many of them have only some of the features that he wants. However, when he finds the Globetrotter website, John is surprised to find that he can create a schedule, add activities to each day, and export the whole trip as a printable itinerary. John decides to go to the route planner section of the website. John adds a destination for his service, aka his starting location. Then he enters the departure and return date for the trip. Lastly, John chooses the types of service that he is interested in and some results show up based on his input. Not only was he able to book the flight, he was also able to book certain activities and hotel rooms by following a similar process. Fast forward a few weeks and Jake and his family return happy from their trip. They were able to accomplish everything they were hoping to.


Scheduling System
Adds destinations/places
Adds dates of stay
Exports an itinerary/schedule
John

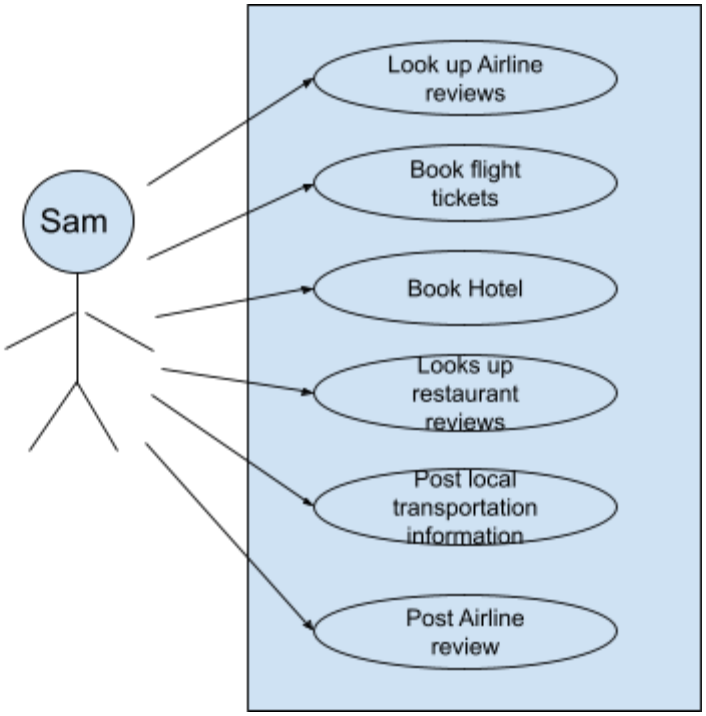
Title:	Transportation
Actors:	Steve Miller and schoolmates
Description:	Steve and his friends have decided to take a trip out of the state to celebrate one of their friends' new promotions. They decided to fly to their destination because Steve does not want to put too many miles on his new car very quickly. Since none of his friends had any vehicles too reliable for long drives, Uber is too expensive to use multiple times a day, they were immediately presented with the dilemma of transportation once they arrived at their destination. During the process of purchasing the flight tickets on Globetrotter, they noticed that they had the option to look at some rental vehicles. They were able to find a vehicle that they all fit in and was very gas efficient. After they returned the car they were happy to have been able to find a car they could use for transportation on the same website where they bought the flight tickets.



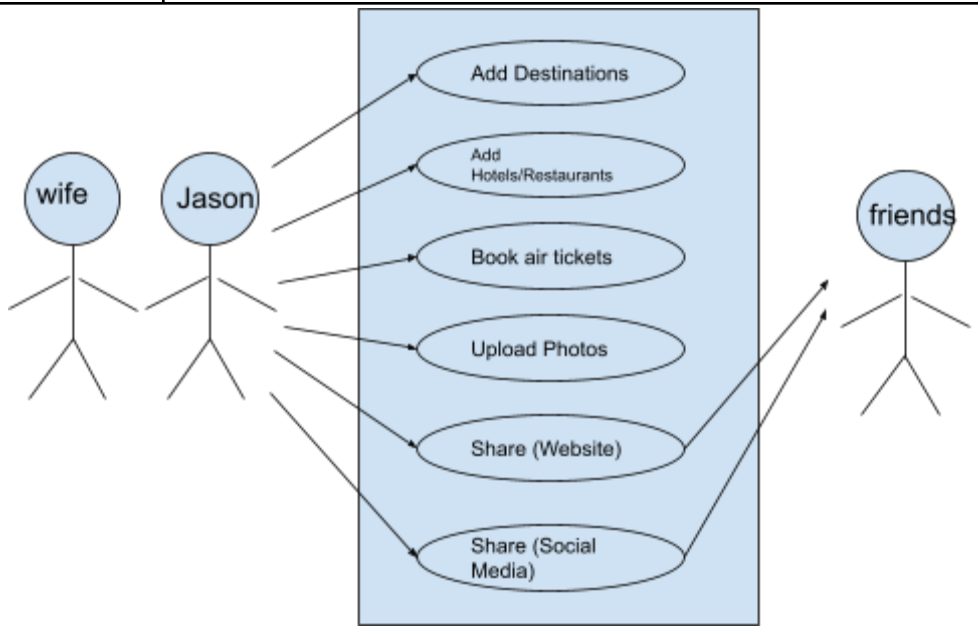
Title:	Inviting Guests On Trips
Actors:	Jake and his friends
Description:	Jake and his friends decided to take a trip out of state now that the country is beginning to open back up and they are all on summer vacation. Jake and his friends are having a hard time finding a time and place for all of them to meet and discuss the details of the trip. Jake begins to look for flight booking websites and eventually ends up on Globetrotter. He was easily able to create an account and save some trips that he created. Jake then realizes that he is able to invite guests to the trip he has created and invites his friends through Globetrotter. His friends can now easily look at the details of the trip at their earliest convenience and give input accordingly. Jake's friends then make their own additions to the trip. Once all the details of the trip have been agreed on, he purchases the flight. Through the Globetrotter website, Jake and his friends were able to create a trip without having to meet in person. They just showed up at the airport the day of and flew to their destination. On the way, they discussed how easy it was for them to plan the trip. A few days later, Jake and his friends make it home safe and are happy that they have created memories together that they will remember for years to come.



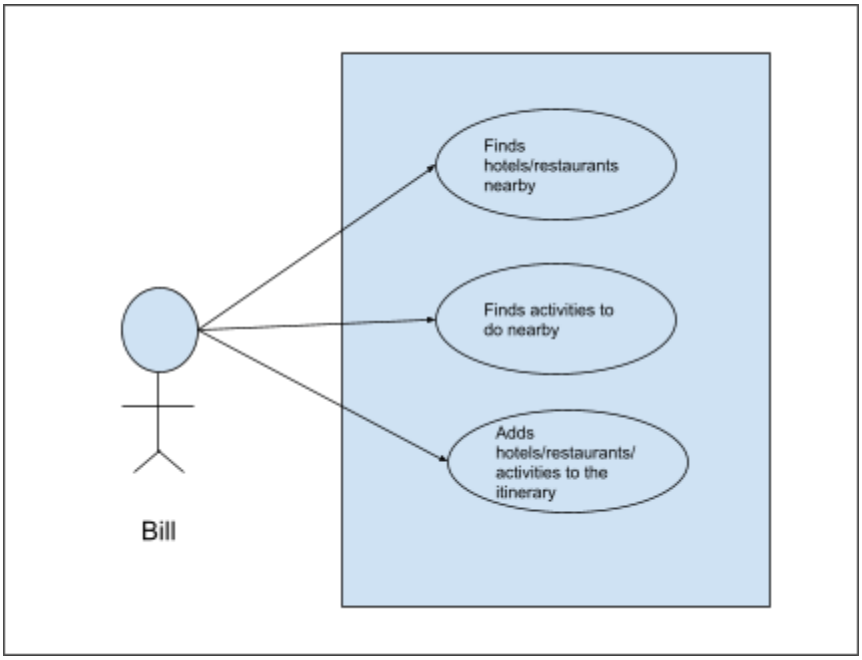
Title:	Business Meeting
Actors:	Sam
Description:	Sam has a business meeting in New York City that just came up and is planning on taking a last-minute 2-way flight from Los Angeles. Sam does not want to have a bad experience because they don't want bad emotions during the meeting so she decides to look up Airline reviews before purchasing. Sam then books a hotel that is near the airport. Sam does not have a lot of time to make it to the meeting. On the way to the airport, she plans on going to a fast-food restaurant. While on the plane, Sam is looking at nice restaurants to consider where the meeting should take place so she decides to look up restaurant reviews. Sam then adds Uber as a method of transportation to her In-Progress trips, for her travel within the city. Once Sam has arrived at JFK airport in NYC, she gets an Uber to take her to the restaurant where the meeting is held. After the meeting is finished, she takes another Uber to her hotel where she will rest until her flight back to Los Angeles. Sam takes an Uber to the JFK airport and takes her flight back home happy to have had a great meeting. After everything went smoothly, Sam decided to add a positive review to the Airline.


Sam
Look up Airline reviews
Book flight tickets
Book Hotel
Looks up restaurant reviews
Post local transportation information
Post Airline review

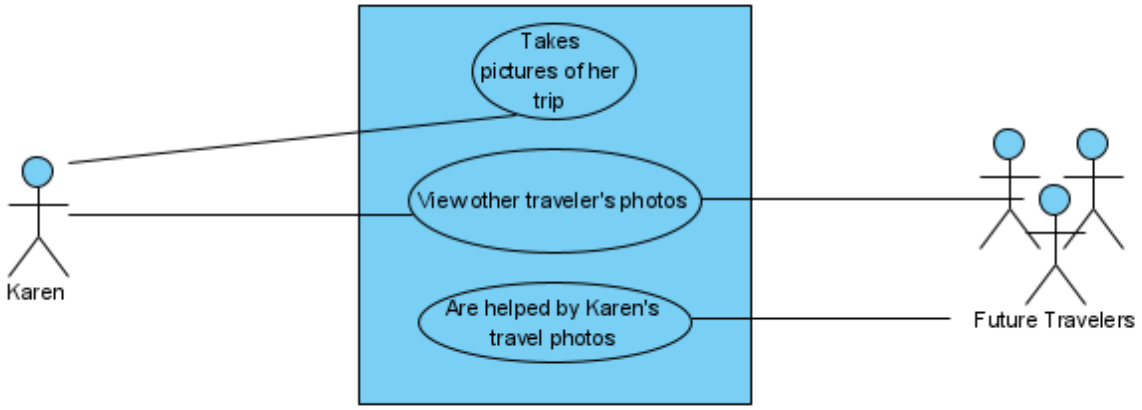
Title:	Social Media
Actors:	Jason, his wife and his friends
Description:	Jason, who lives in Los Angeles, and his newly married wife plan to spend an unforgettable honeymoon in Hawaii. Jason is a busy photographer, they only have 7 days, so they choose to play only on Oahu. They plan to fly from Los Angeles International Airport to Honolulu International Airport at noon on the first day. After check-in, they hope to book a Japanese restaurant for dinner near the hotel in advance. On the way, they plan to go to Waikiki Beach to sunbathe, watch the seaside scenery, visit the Macap Point Lighthouse, enjoy the scenery and take pictures, and go sightseeing in the historic pineapple plantation Dole Plantation. In the next few days, they hope to experience skydiving, experience shooting dolphins and whales by boat. Finally, they will take the return flight back to Los Angeles. Because of Jason's personal habits, he will take a lot of photos on the road. He hopes to not only share his itinerary and photos taken by himself on the website, but also let his friends see his itinerary on Instagram.



Title:	Activity Itinerary
Actors:	Bill
Description:	Bill is planning a Business trip to Tucson, Arizona. He has an exhibition he has to attend for 3 days there. After figuring out the dates of the exhibition, he wants to plan out his itinerary. He plans to fly there a day before it starts and fly back the day after it ends, concluding he needs to plan a trip for 5 days and 4 nights. He decides to book a hotel within 10 miles of his exhibition hall so travel time will not be an issue. He plans to meet his fellow exhibitors on the day he arrives and they have given him the tasks to find activities to do there and restaurants to go eat at. He decides to plan out an itinerary on Globetrotter, which allows him to search for activities to do nearby and make hotel/restaurant reservations so he can stay organized and to make sure he is utilizing his free time with his fellow exhibitors and gets to do as many activities as possible before work starts.



Title:	Photo Gallery
Actors:	Karen, Future Travelers
Description:	Aspiring food blogger, Karen, loves to travel often by herself. Her most recent trip included flying from San Francisco to New York City, with a layover in Denver in between. While traveling, she keeps track of her adventures by taking pictures of almost everything she does and eats. With the layover in Denver, Karen actually did not know what to do with her time there since she was not expecting to be there in the first place. She takes a risk and tries a local restaurant for lunch. Karen finds a bug in her food and is also met with horrible customer service, so this meal was probably the worst of her whole trip. She documents the unpleasant meal but has no one to share this warning with besides herself. Karen needs a way to share her knowledge and experience with others, while also learning from other people's experiences as well. With a photo gallery, people can share fun finds and things to avoid while traveling to areas you have never been before! Karen can post her fun pictures at the Statue of Liberty, hidden restaurant gems she finds, and the rude bug-serving restaurant so others can make judgements for their own future travels.



Application Entities

Roles

- Unregistered User - A user who has not registered an account with Globetrotter. All users begin with this role.
- Registered User - A user who has registered an account with Globetrotter.
- Guest - A user that has been added to a trip.
- Unregistered Guest - Someone who has been added to a trip but has not registered with Globetrotter.
- Registered Guest - Someone who has been added to a trip that already has a Globetrotter account.
- Owner - A registered user who has created a trip

Trips

A set of unconnected dates with locations.

Trip Day

Each day added to a trip has a location, a date, and activities.

Flights

Flights that the user has taken.

Airports

Origins and destinations of flights

Photo

Users are allowed to add photos to a Photo Album.

Photo Album

Functional Requirements

Account Management

1. General users can contact the team.
2. Registered users can access the ‘Account Management’ page.
3. Registered users can change their username.
4. Registered users can change their email.
5. Registered users can change their preferred currency.
6. Registered users can change their home address.
7. Registered users can change their primary phone number linked to their account.
8. Registered users can upload a profile picture.
9. Registered users can change their profile picture.
10. Registered users can change their primary phone number.
11. Registered users can change their secondary phone number.
12. Registered users can change their password.
13. All registered guests may add other registered users to the trip.
14. All registered guests may add unregistered guests to the trip.
15. All registered guests can only add up to one registered user at a time.
16. All registered guests can only add up to nine registered users to the trip.
17. All guests added to the trip will be sent an email containing a link to the trip.
18. All users can search for flights.
19. All users may add hotel reservations to a particular day of a trip.
20. All users may add entertainment activities to a particular day of a trip.
21. All users may add transportation reservations to a particular day of a trip.
22. All users can choose from the available flights.
23. All users will be able to view arrival times for the available flights.
24. All users can look at reviews of Airlines.
25. Adding a confirmation number to a trip in the Shopping Cart moves the trip to Purchased.
26. All registered users may add a confirmation number to a trip in their Shopping Cart.
27. Unregistered guests may not purchase trips on which they are guests.
28. Registered guests may purchase trips on which they are guests.
29. Registered guests will receive a confirmation email after checkout.
30. Registered users may purchase their ‘Shopping Cart’ trips.
31. Registered users will receive a confirmation email after checkout.
32. Registered users can login using the ‘Login’ page.
33. ‘Login’ page will have a ‘forgot password?’ button.

34. Registered users can reset their password from the ‘Login’ page by sending a reset link to the account’s email.
35. Unregistered users may choose to continue as an unregistered user.
36. Unregistered users may choose to go to the ‘Registration’ page.
37. All users shall be able to access the “Meet the Team” page to find out more information about the team.
38. General users will be able to click on individual team images to find out more information about the specific member.
39. All users shall be able to access the “Meet the Team” page.
40. Registered users can edit their reviews.
41. Registered users can delete their reviews.
42. Registered users may cancel flights which they purchased.
43. Registered users can be issued a refund for flights cancelled within 48 hours of departure.
44. Registered users can receive credit for flights cancelled within 48 hours of departure.
45. Registered users can post a review of their flight.
46. Unregistered users shall be directed to the ‘Registration’ page.
47. Unregistered users will be able to create an account.
48. Unregistered users shall choose a username when creating an account.
49. Unregistered users shall choose a password when creating an account.
50. Unregistered users shall provide their email when creating an account.
51. Unregistered users can specify a preferred currency for displaying prices when creating an account.
52. Unregistered users can record their home address when creating an account.
53. Unregistered users shall add a primary phone number.
54. Unregistered users can add a secondary phone number.
55. Users will be sent a confirmation email after completing registration.
56. All users can change the displayed currency.
57. All Registered Users can specify a return date
58. All registered users can add at least one destination.
59. All registered users can add multiple flight stops in between the start and destination locations (waypoints).
60. All registered users can edit the starting location of a trip.
61. All registered users can remove waypoints.
62. All registered users can change the starting location of a trip.
63. All registered users can change the destination of a trip.
64. All registered users can remove the destination of a trip.
65. All registered users shall specify the dates of departure.
66. All registered users can remove the starting location of a trip.
67. All registered guests can add multiple flight stops in between the start and destination locations (waypoints).

68. All registered guests can add multiple flight stops in between the start and destination locations.
69. All registered guests can edit the starting location of a trip.
70. All registered guests can remove waypoints.
71. All registered guests can change the starting location of a trip.
72. All registered guests can change the destination of a trip.
73. All registered guests can remove the starting location of a trip.
74. All registered guests can remove the destination of a trip.
75. All registered guests shall specify the dates of departure.
76. All registered guests can add guests to the trip.
77. All registered guests can add at least one starting location.
78. The displayed currency will default to the preferred currency of a logged in user.
79. All users shall specify a budget.
80. All users will be able to view weather information for each location.
81. Registered users can navigate to the ‘Account Management’ page.
82. Dates added to a trip may not overlap.
83. Registered users may view trips that they previously created.
84. Registered users may edit trips that they previously created.
85. Registered users may delete trips that they previously created.

Photo Album

86. Registered users may upload pictures of their trip.
87. Registered users may delete pictures they uploaded.
88. Registered users may add a photo description when uploading a picture.
89. Registered users may edit their photo descriptions.
90. Registered users may change titles of pictures they uploaded.
91. Registered users may delete their photo descriptions.
92. Registered users shall give a title when uploading a picture.
93. Users may export the schedule prepared by Globetrotter to a file.
94. Users may print the schedule prepared by Globetrotter.

Guests

95. Registered users may edit trips to which they were added as a guest.
96. Registered users may edit trips on which they are guests.
97. Registered users may view trips to which they were added as a passenger.
98. Unregistered users may not edit trips on which they are guests.
99. Any changes of the trip will be auto saved.
100. All guests may view trips on which they are guests.

Website

101. Website will notify the user of redirection to external websites.
102. The website will let the user know when they are hovering a clickable item.
103. The website will have a hover effect on clickable items.
104. The website will be easy to maneuver through.

Non-functional Requirements

Functionality

1. The website shall be hosted on SSD Nodes.
2. The website shall be user friendly.
3. The website shall be easy to navigate.

Security

4. Website will use https wherever possible.
5. User login credentials will be validated against the user database.
6. Changes to access controls in the database will be reflected immediately in the user interface.
7. The user's passwords will be encrypted in the database.
8. User passwords will be case sensitive.
9. User passwords shall have at least 8 characters.
10. User passwords shall have at least one upper case letter.
11. User passwords shall have at least one special character.
12. The website will use sessions to track logged in users.

System Requirements

13. The website shall function on at least version 91.0.4472.101 of Google Chrome.
14. The website shall function on at least version 91.0.864.48 of Microsoft Edge.
15. The website shall function on at least version 14.1 of Apple Safari.
16. The website shall function on at least version 89.0.1 of Mozilla Firefox.

Marketing

17. The website shall use the company logo displayed.
18. The website shall have a link to the company’s Instagram profile on each page.
19. The website shall have a link to the company’s Facebook profile on each page.
20. The website shall have a link to the company’s Twitter profile on each page.
21. The company shall have engaging content.

Content

22. The website shall have a navigation bar.
23. The website shall have a clear description of its functionality.
24. The website shall have a consistency of its design on every page.

25. The website shall have a destination search entry to search for flights.
26. The flights returned from API will match parameters entered by the user.
27. The website shall have a from location search entry to search for flights.
28. The website shall have a date of flight search entry to search for flights.
29. The website shall have a drag bar for air ticket budgets.
30. The website shall be able to convert currencies.

Coding Standards

31. The code will be portable.
32. The code will have a consistent indentation style.
33. The code shall be understandable.
34. The code shall contain comments so that team members can quickly learn what different pieces of code do without having to dive deep into the code.
35. There shall be a limit to the line length of the code to avoid horizontal scrolling.
36. The team lead shall decide if code is at the agreed upon standard.
37. The team lead shall decide whether code can be pushed to the main branch.
38. The coding will be done in a modular way.
39. Code modules will be reused as often as possible.
40. CSS will be applied to classes.
41. CSS will not be applied to ID's.
42. Scripts will be in their own files
43. NodeJS modules will be installed in the 'node_modules' folder.
44. The website will use a public and private folder to serve web content.
45. Stylesheets will be in their own files.
46. HTML files will be in the HTML folder.
47. CSS files will be in the CSS folder.
48. Script files will be in the Script folder.
49. Express routers will be placed in their own folder.
50. Handlebars partials will be in their own folder.
51. Handlebars templates will be in their own folder.
52. Handlebars layouts will be in their own folder.
53. Images will be in their own folder.
54. The main branch for code will be the 'dev' branch.
55. Code that is ready to be tested will be moved from the 'dev' branch to the 'testing' branch.
56. Code that has been tested will be moved from the 'testing branch' to the 'master' branch.
57. Code will only be pushed after pulling first.

Availability

58. The website will not restrict users based on their geographic location.

59. The website's URL shall be reliable to access its contents.
60. The website's URLs shall be simple and short.
61. The website will load in at least 20 seconds unless there is a system error.

Scalability

62. The website will be designed such that it can be migrated to higher-performance hardware if demand increases.

Fault intolerance

63. A backup of the virtual server will be performed daily.
64. The virtual server will be restored from a backup in the event of a total system failure.

Expected load

65. The website is expected to handle 10-20 users at a time.

Privacy

66. Globetrotter and its employees cannot be held responsible for any data leaks.
67. Globetrotter and its employees cannot be held responsible for any data loss.
68. The website shall not save the users CVV number during the checkout process.
69. The website shall not save the users CVV number after the checkout process.
70. This website shall not sell user personal information to third parties.
71. The website will not share private user information with other users unless where explicitly done by the user.
72. The website shall not store any user medical information.
73. The website will not store users' IP addresses.
74. The website will not share users' IP addresses.
75. The website will not publish any false or fraudulent information in any form.

Legal

76. The website shall have Terms and Conditions which users must consent to before logging in.
77. The website shall notify the user when the Terms and Conditions have been changed.
78. The website shall not let the user continue until they have accepted the updated Terms and Conditions.
79. Users agree not to harass Globetrotter employees using the website or email.
80. Users agree not to harass other users using the website or email.

81. Users agree not to post illegal content on the Globetrotter website.
82. Users agree not to DDoS the Globetrotter website.
83. Users agree not to tamper with the source code of the Globetrotter website.
84. Users agree not to upload invalid data, viruses, worms or others through the website.
85. Users agree to provide truthful and accurate content.
86. The website shall have a copyright notice.
87. All content of the website is released under GNU Public License v3 unless otherwise noted.
88. Website is not responsible for users' violation of any laws while using the Service.
89. Website reserves the right to suspend or terminate any user at any time.

Look and Feel

90. The website will use a uniform color palette on all pages of the site.
91. The website will have a responsive design.
92. The website will scale to the width of the screen.
93. The website will scale to the height of the screen.
94. The website will be optimized for tablet clients.
95. The website will be optimized for desktop clients.
96. The website will be optimized for mobile clients.
97. The website will have readable text font size.
98. The website will have uniform font styles.
99. The website will have no typos.
100. The website will have alt tags for images.
101. The website will have title tags for images.
102. The website will accurately display currency to two decimal places.

Competitive Analysis

Quick Comparison

	Tripadvisor	Road Trippers	TripIt	Kayak	Wander
Strengths	Lots of options and suggestions	Great map display and fascinating pictures	Automatically scans and imports travel plans from connected primary email's inbox.	Flexible methods for adding services (manual, search, import), intuitive UI	Variety of travel packages ready to go, many ways to search for the ideal trip
Weaknesses	Too many blogs and guides	Only support road trips	Cannot search for flights to book	No photo gallery	Website feels cluttered
Social Media	Daily posts to Instagram, twitter and facebook	Blog posts, Facebook, Twitter	Blog posts, Instagram, Twitter, YouTube	Frequent social media posts on Facebook and Twitter	Blog posts on Facebook, Twitter, Instagram
Pricing	Free/ $99/year (Plus)	Free/ $29.99/year (Plus)	Free / $49/year(Pro)	Free	Free

Feature Comparison

	Tripadvisor	Road Trippers	TripIt	Kayak	Wander	Globetrotter
Users can add travel destinations	+	+	+	+	+	+
Users can search for flights	+	-	-	+	+	+
Users can search for hotels	+	+	-	+	+	+
Users can search for restaurants	+	+	-	+	-	+
Users are given suggestions for flights	+	-	-	+	+	+
Exportable itinerary	+	+	+	+	+	+
Supports collaboration on trips	+	-	+	+	-	+
Personal Photo Gallery	-	-	-	-	-	++

Economic Plan

Most of the competitors use ads to generate revenue for their site. Potential sources of revenue for Globetrotter include:

- Limit the number of trips that a user can create for free; charge for additional
- Limit the types of activities that a user can add to their trip for free; charge for additional
- Sell a subscription option
- Sell advertisements to travel-related businesses

Personal Photo Gallery (Advanced Feature)

Registered Users will be able to upload photos to their personal photo album which they can share with other users. Photo album will be a collaborative feature where the owner of the photo album will be able to add other users to post photos in their album. Personal photo albums will only be viewable to the owner of the photo album and the people added to the album. Only the owner of the photo album will be able to rename, edit description, and delete the photo album. Only the owner of a photo uploaded to a photo album will be able to rename, edit description, and delete the photo. Compared to our competitors, our Personal Photo Gallery is an advanced feature because it gives the ability for users to have personal photo albums and collaboratively work on it.

Tools and Frameworks

Frameworks

- NodeJS
- Express
- Handlebars

IDE

- Visual Studio Code

APIs

- SkyScanner Flight Search
    - This service will allow us to get flight price estimates.
    - https://rapidapi.com/skyscanner/api/skyscanner-flight-search/
- Compare Flight Prices
    - This API returns flight prices from Expedia, Travelocity, etc.
    - https://rapidapi.com/obryan-software-obryan-software-default/api/compare-flight-prices/
- Travel Advisor
    - Travel Advisor allows for searching of places (restaurants, hotels, etc.) within a rectangular region or radius. The API returns the names of the places, price info, and ratings.
    - https://rapidapi.com/apidojo/api/travel-advisor/
- SendGrid
    - Adds the ability to send emails to users.
    - https://sendgrid.com

Checklist

- Team found a time slot to meet outside of class.

  DONE

- Github master chosen

  DONE

- Team decided and agreed together on using the listed SW tools and deployment server.

  DONE

- Team ready and able to use the chosen back and front end frameworks and those who need to learn are working on learning and practicing.

  ON TRACK

- Team lead ensured that all the team members read the final M1 and agree/understand it before submission.

  DONE

- Github is organized as discussed in class (e.g. master branch, development branch, folder for milestone documents, etc.)

  ON TRACK

List Of Team Contributions

Taylor

- Executive Summary
- Added some Functional/Non-Functional Requirements
- Wrote Use Case for ‘Planning a Vacation Schedule’
- Helped with competitive analysis for Kayak and Tripadvisor

Luis

- Wrote “Business Meeting” Use Case and drew Use Case Diagram
- Contributed to Functional/Non-Functional Requirements
- Contributed to Application Entities
- Contributed to competitive analysis for Tripadvisor and TripIt.
- Added SendGrip Api to our list

Yi

- Wrote ‘Social Media’ Use Case
- Contributed to Function/Non-Functional Requirements
- Contributed to competitive analysis for RoadTrippers.

Robin

- Wrote “Photo Gallery” Use Case
- Contributed to Functional/Non-Functional Requirements
- Contributed to competitive analysis for Wander

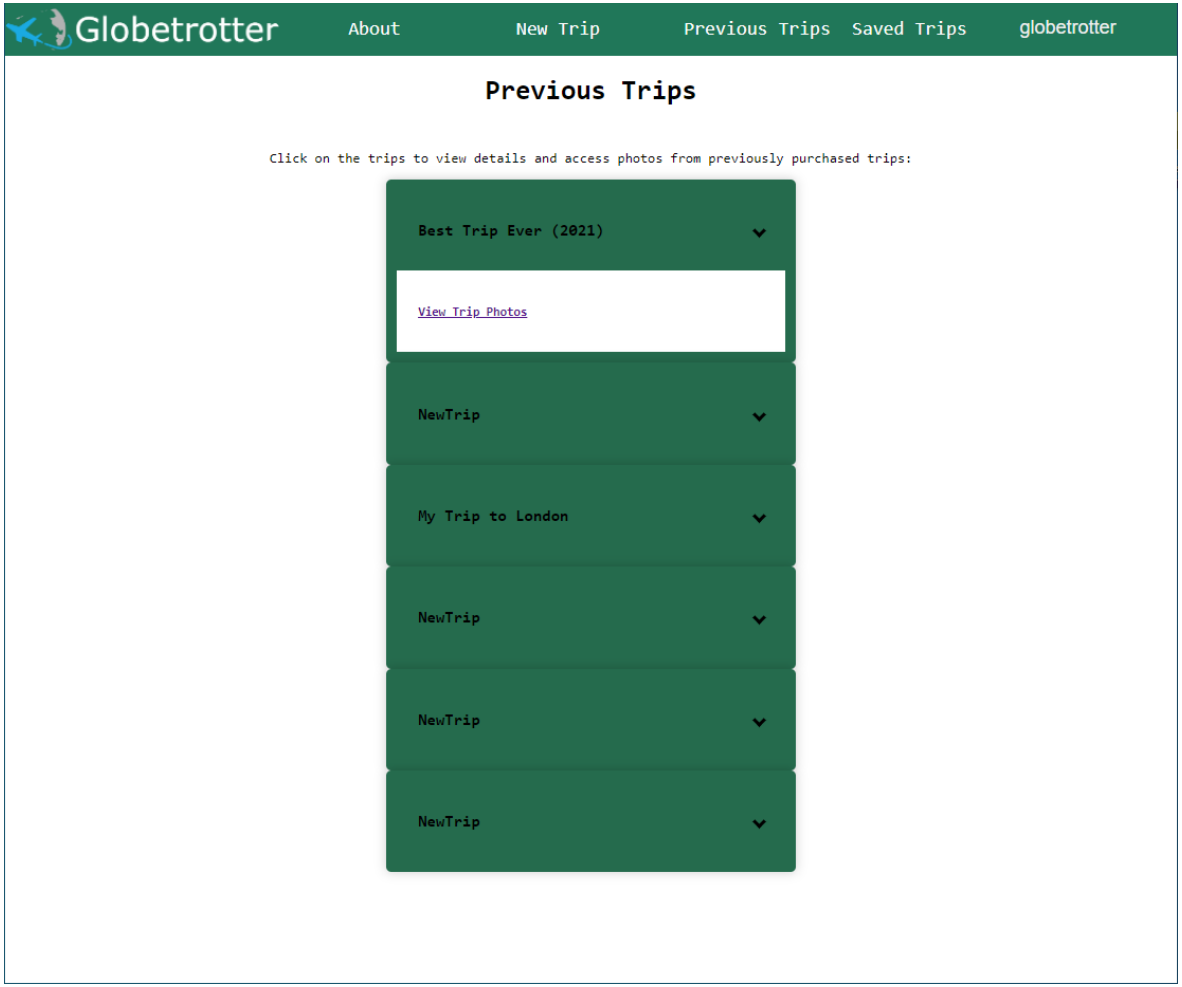
Kajeme

- Wrote Use Case and made a Use case diagram
- Contributed to Functional Requirements
- Contributed to Non-Functional Requirements
- Contributed to Competitive Analysis for Kayak

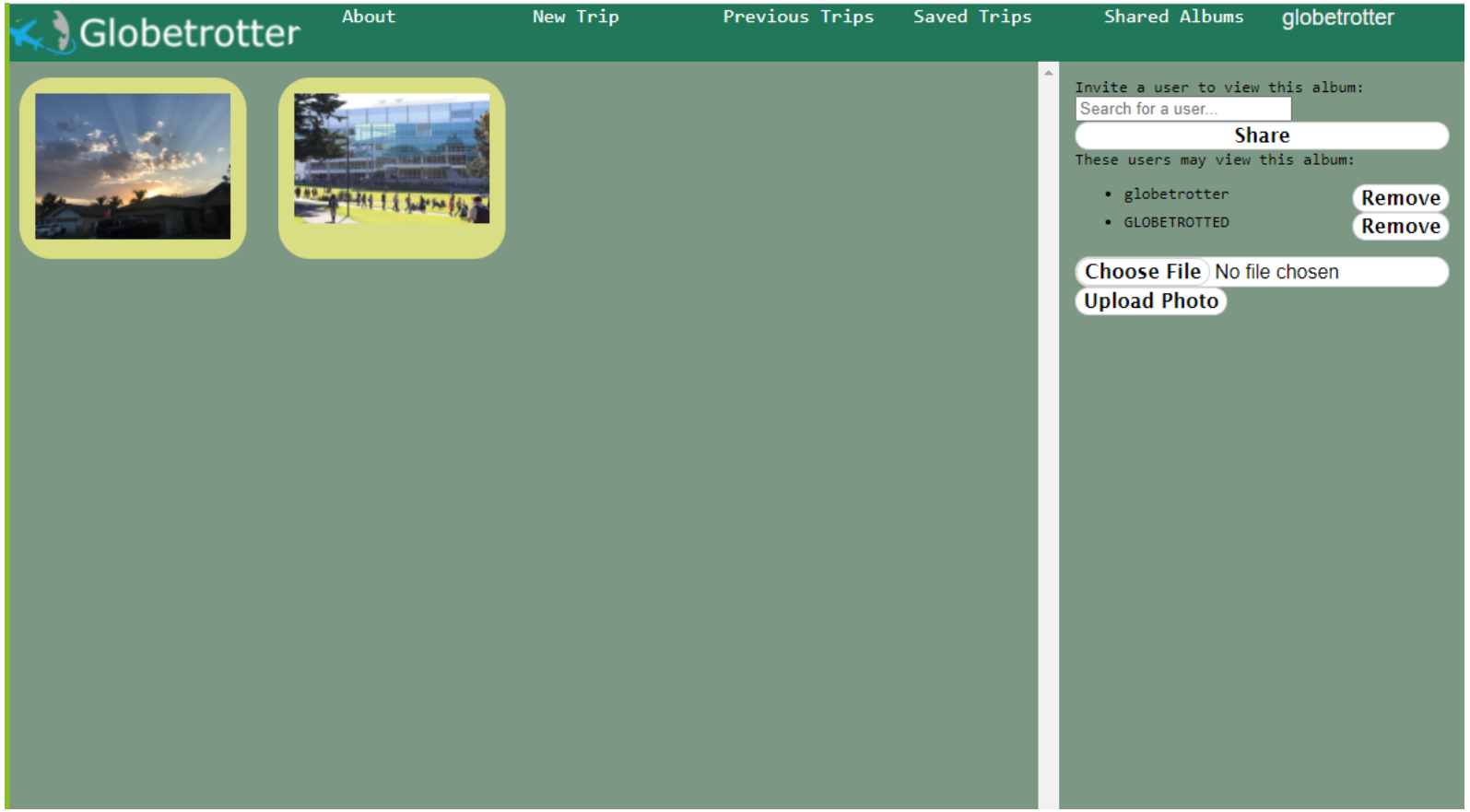
Jesus

- Wrote “Transportation” Use Case.
- Wrote “Inviting of Guests on Trips” Use Case.
- Contributed to Functional/Non-Functional Requirements.
- Contributed to competitive analysis for Wander.
- Contributed in creating the Checklist.

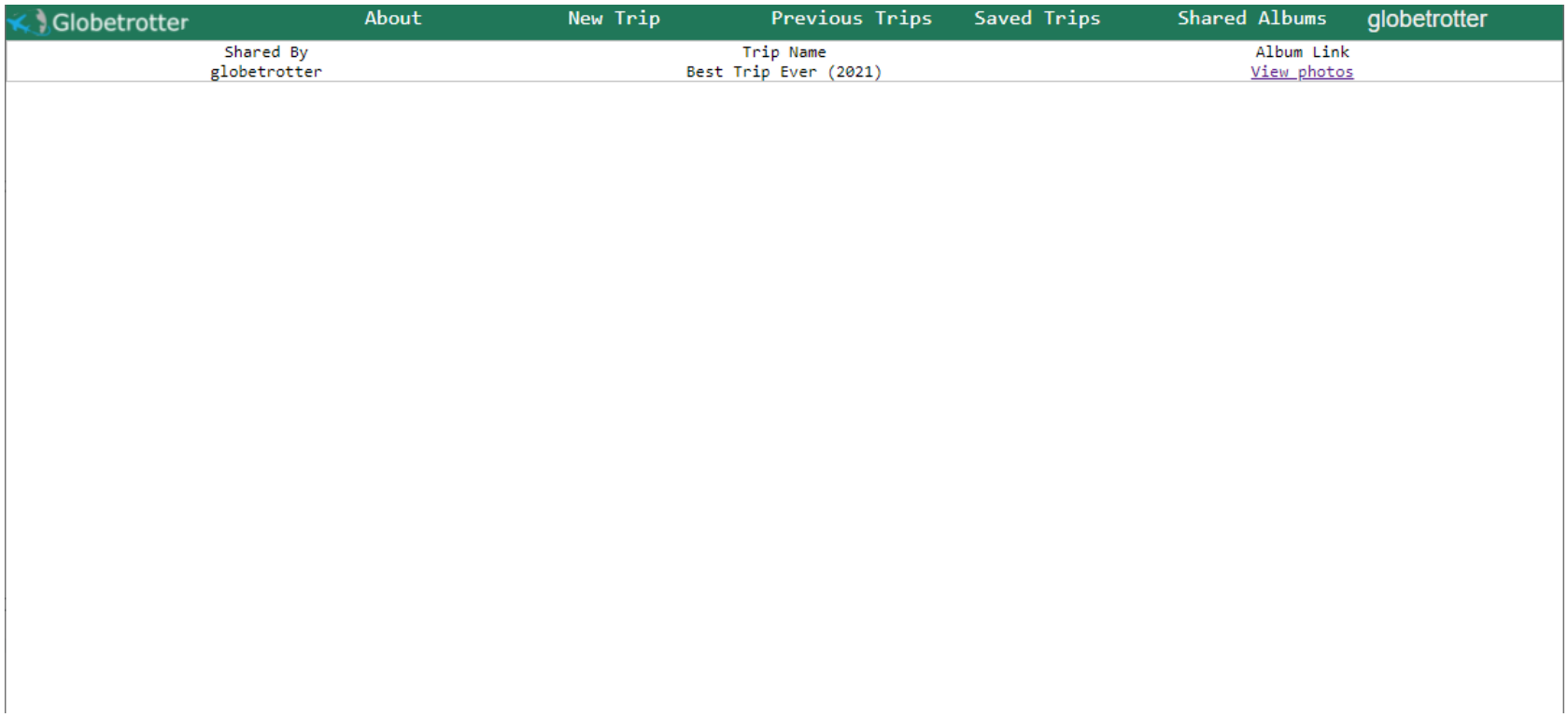
4. Product Screenshots



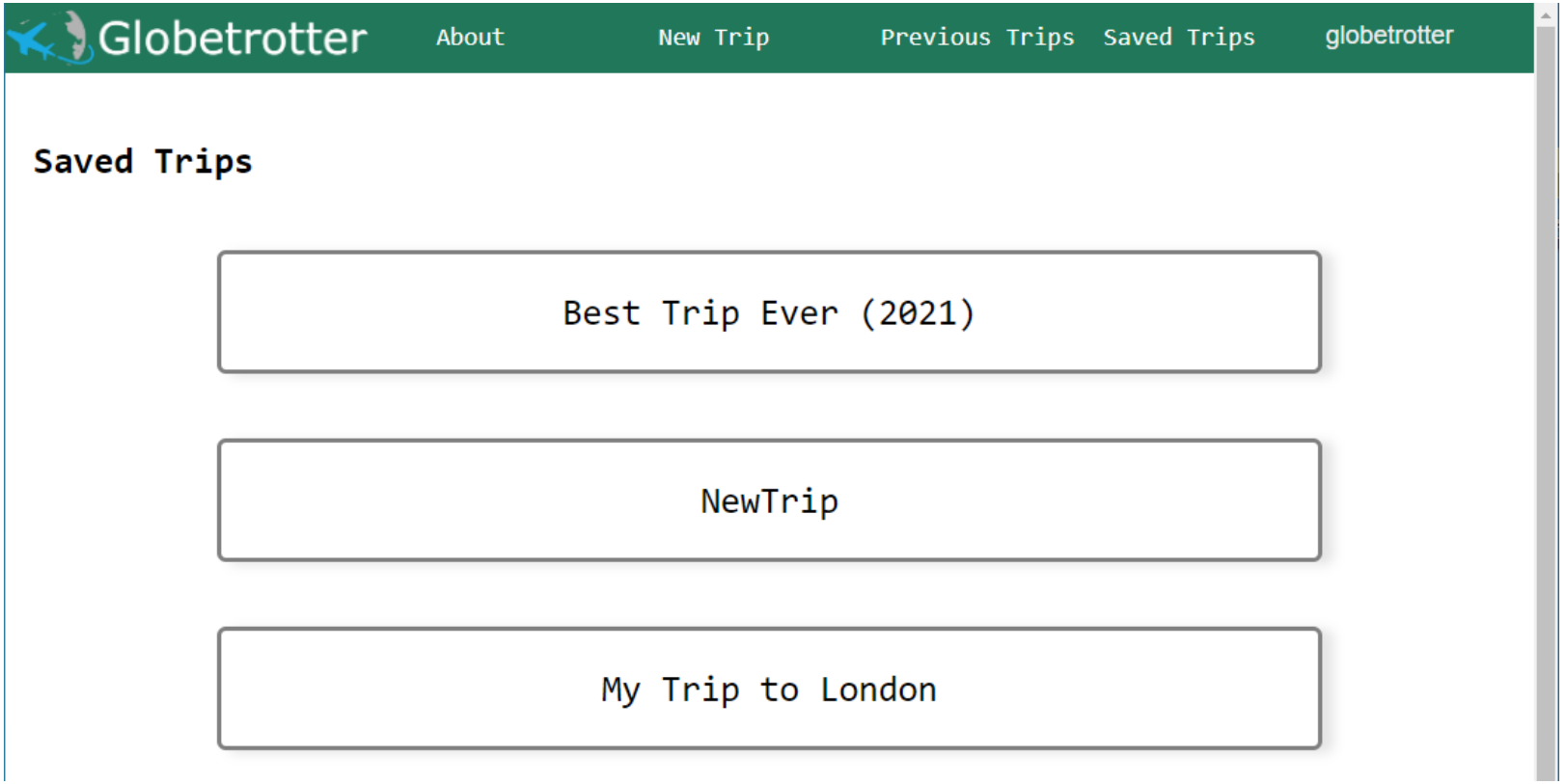
This image shows the Previous Trips page.



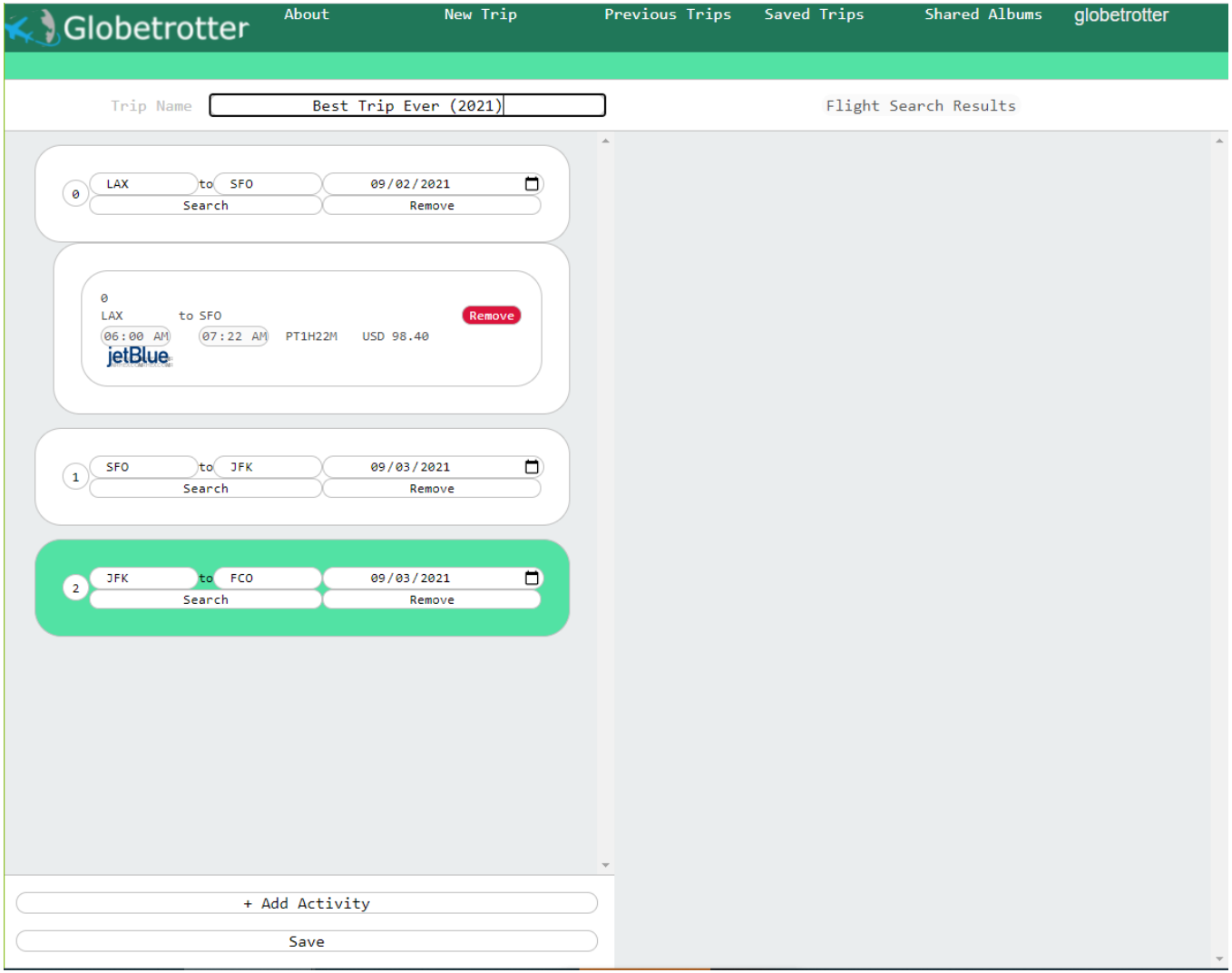
This image shows the Photo Gallery page.



This image shows the Shared Albums page.



This image shows the Saved Trips page.



This image shows the Route Planner page.

5. Database Screenshots

Table Name: registered_user

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
user	VARCHAR(36)	☑	☑	☐	☐	☐	☐	☐	☐	
username	VARCHAR(255)	☐	☑	☑	☐	☐	☐	☐	☐	
password_hashed	VARCHAR(255)	☐	☑	☐	☐	☐	☐	☐	☐	
first_name	VARCHAR(255)	☐	☐	☐	☐	☐	☐	☐	☐	
last_name	VARCHAR(255)	☐	☐	☐	☐	☐	☐	☐	☐	
birthday	DATE	☐	☐	☐	☐	☐	☐	☐	☐	
gender	VARCHAR(255)	☐	☐	☐	☐	☐	☐	☐	☐	
preferred_currency	VARCHAR(3)	☐	☐	☐	☐	☐	☐	☐	☐	NULL
primary_phone_country_code	VARCHAR(2)	☐	☐	☐	☐	☐	☐	☐	☐	NULL
primary_phone_number	VARCHAR(10)	☐	☐	☐	☐	☐	☐	☐	☐	NULL
secondary_phone_country_c...	VARCHAR(2)	☐	☐	☐	☐	☐	☐	☐	☐	NULL
secondary_phone_number	VARCHAR(10)	☐	☐	☐	☐	☐	☐	☐	☐	NULL

Table Name: trip

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
owner	VARCHAR(36)	☑	☑	☐	☐	☐	☐	☐	☐	
trip_id	VARCHAR(36)	☑	☑	☑	☐	☐	☐	☐	☐	
trip_name	VARCHAR(255)	☐	☑	☐	☐	☐	☐	☐	☐	
		☐	☐	☐	☐	☐	☐	☐	☐	

Table Name: activity

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
trip_owner	VARCHAR(36)	☑	☑	☐	☐	☐	☐	☐	☐	
trip	VARCHAR(36)	☑	☑	☐	☐	☐	☐	☐	☐	
activity_id	VARCHAR(36)	☑	☑	☑	☐	☐	☐	☐	☐	
start_time	TIMESTAMP	☐	☑	☐	☐	☐	☐	☐	☐	
end_time	TIMESTAMP	☐	☑	☐	☐	☐	☐	☐	☐	
start_location	VARCHAR(36)	☐	☐	☐	☐	☐	☐	☐	☐	
end_location	VARCHAR(36)	☐	☐	☐	☐	☐	☐	☐	☐	NULL
		☐	☐	☐	☐	☐	☐	☐	☐	

Table Name: flight_activity

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
trip_owner	VARCHAR(36)	☑	☑	☐	☐	☐	☐	☐	☐	
trip	VARCHAR(36)	☑	☑	☐	☐	☐	☐	☐	☐	
activity	VARCHAR(36)	☑	☑	☑	☐	☐	☐	☐	☐	
flight_offer_json_data	JSON	☐	☐	☐	☐	☐	☐	☐	☐	
		☐	☐	☐	☐	☐	☐	☐	☐	

Table Name: location

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
location_id	VARCHAR(36)	☑	☑	☐	☐	☐	☐	☐	☐	
location_name	VARCHAR(255)	☐	☐	☐	☐	☐	☐	☐	☐	NULL
address_line_1	VARCHAR(255)	☐	☐	☐	☐	☐	☐	☐	☐	NULL
address_line_2	VARCHAR(255)	☐	☐	☐	☐	☐	☐	☐	☐	NULL
city	VARCHAR(255)	☐	☐	☐	☐	☐	☐	☐	☐	NULL
state	VARCHAR(255)	☐	☐	☐	☐	☐	☐	☐	☐	NULL
country	VARCHAR(255)	☐	☐	☐	☐	☐	☐	☐	☐	NULL
postal_code	VARCHAR(255)	☐	☐	☐	☐	☐	☐	☐	☐	NULL
latitude	DECIMAL(10,7)	☐	☐	☐	☐	☐	☐	☐	☐	NULL
longitude	DECIMAL(10,7)	☐	☐	☐	☐	☐	☐	☐	☐	NULL

Table Name: airport

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
location	VARCHAR(36)	☑	☑	☑	☐	☐	☐	☐	☐	
iata_code	VARCHAR(3)	☑	☑	☑	☐	☐	☐	☐	☐	

Table Name: file

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
owner	VARCHAR(36)	☑	☑	☐	☐	☐	☐	☐	☐	
file_id	VARCHAR(36)	☑	☑	☐	☐	☐	☐	☐	☐	
folder_path	VARCHAR(255)	☐	☑	☐	☐	☐	☐	☐	☐	
file_name	VARCHAR(255)	☐	☑	☐	☐	☐	☐	☐	☐	
extension	VARCHAR(5)	☐	☐	☐	☐	☐	☐	☐	☐	NULL
full_file_path	VARCHAR(255)	☐	☑	☑	☐	☐	☐	☐	☐	

Table Name: photo

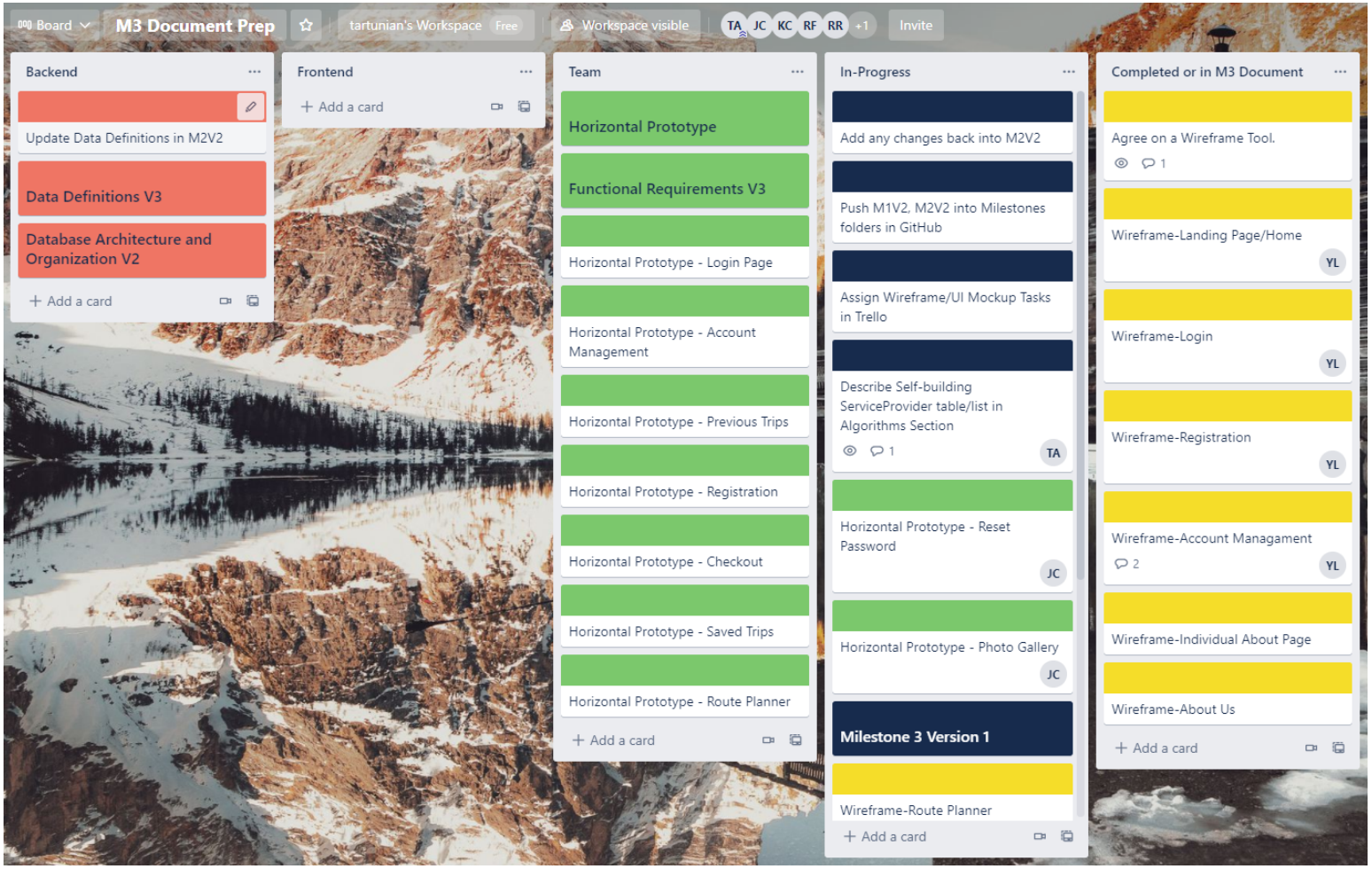
Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
file_owner	VARCHAR(36)	☑	☑	☐	☐	☐	☐	☐	☐	
file	VARCHAR(36)	☑	☑	☐	☐	☐	☐	☐	☐	
photo_id	VARCHAR(36)	☑	☑	☑	☐	☐	☐	☐	☐	
title	VARCHAR(255)	☐	☑	☐	☐	☐	☐	☐	☐	
description	VARCHAR(255)	☐	☑	☐	☐	☐	☐	☐	☐	
is_profile_photo	TINYINT	☐	☐	☐	☐	☐	☐	☐	☐	NULL

Table Name: photo_album

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
photo_album_id	VARCHAR(36)	☑	☑	☑	☐	☐	☐	☐	☐	
trip_owner	VARCHAR(36)	☑	☑	☐	☐	☐	☐	☐	☐	
trip	VARCHAR(36)	☑	☑	☑	☐	☐	☐	☐	☐	

6. Task Management Screenshots

Trello



7. Team Contributions

- Taylor Artunian: 9

  I think I did my best to lead the team in the right direction and take ownership of our project. I could have taken more time to learn about the team's strengths and weaknesses earlier on and modified our scope.

- Luis Alfaro: 8

  It was a little hard to contact Luis at the very beginning of the project but he ultimately showed ownership of the backend. When I made changes to the database he adapted and even found errors in my code.

- Kajeme Cheneque: 6

  We did not hear from or see KJ sometimes but he did find bugs and errors which was helpful.

- Jesus Correa: 8

  Jesus stepped up and provided leadership and communication to the team in every aspect of the project: frontend, backend, and documentation.

- Robin Fernando: 6

  The Previous Trips and Saved Trips pages were good but we could have heard from him more often.

- Yi Liu: 7

  Yi did a great job with combining all of the mockups from the frontend team into a set of wireframes that looked great. I would have liked to see him turn those into HTML.

- Ruja Rajbhandari: 7

  Ruja started off strong with creating the interactive prototype early in the project and did a great job with the navbar. I would have liked to see some initiative to create layouts or page templates because her code was good.

8. Post Analysis

This project was a great learning experience. We learned, through a ‘sink-or-swim’ approach, that in the SWE industry, there will not always be someone there to hold your hand and guide you to success. The project idea, scope, and implementation were all within our control which also made our success within our control as well.

As the team lead, I think my main shortcoming was not paying closer attention to the strengths and weaknesses of my team. We could have worked better together if I had led the team in practicing how to do the basics with the platform and frameworks (NodeJS and Express) that we chose earlier on. I believe this would have brought us all to a common point of technical understanding which we achieved, but later on. By not doing this, our separation of roles broke down and those with the most technical knowledge ended up dominating the project. In addition, I feel that I needed to bring a better understanding of git as we began the coding phase so that we could have been more efficient and worked independently.

In regards to the team, I think one issue that hindered our performance was that there was not a strong desire to own the product, at least early on. By this I mean that not too many members had strong opinions about what they wanted to see the final product and could also personally contribute to it. From my perspective, this was risky because it led to a scope that sounds great, but is lacking a plan for actual implementation. We also did not fully understand that we did not need 100+ functional and non-functional requirements at the end of the project which led to us thinking we had to deliver more requirements than we could.

Overall, I learned a lot about how to work in a team through this project. If I were to do a project like this again, I would definitely make fewer assumptions and do some things differently. We were up against a short schedule in the Summer and in a longer semester I think we would have had better communication opportunities among our team and with the professor, which would have helped with some of the challenges we faced. I had a great time working with everyone and am grateful for the experience we had.

-Taylor