BucketLyst

Team 1

Marie Shimizu[1], Arielle Riray[2], Aaron Kuo[3], Tommy Truong[4],
Kenneth Gee, Rabin Karki, Francis Quang

[1] Team Lead mshimizu4@sfsu.edu
[2] Front End Lead
[3] Back End Lead
[4] Github Master

Milestone 4

Milestone	Date
M4V2	05/25/2023
M4V1	5/18/2023
M3V2	5/5/2023
M3V1	4/27/2023
M2V2	4/24/2023
M2V1	4/3/2023
M1V2	4/2/2023
M1V1	3/2/2023

Table Of Contents

1. Product Summary

Name:

    BucketLyst

Major Functions:

1. Guest User:
   a. Accessible Pages:
      i. Guest user can explore BucketLyst landing page
      ii. Guest user can go to create account page
      iii. Guest user can go to signin page

   b. Account Creation:
      i. Guest user can create a new account with BucketLyst
      ii. Guest user can sign up a new account using Google
      iii. Guest user can sign up a new account using Facebook

   c. Signing in:
      i. Guest user can sign in with Google account
      ii. Guest user can sign in with Facebook account
      iii. Guest user can sign in with registered details
      iv.

2. Registered user:
   a. Locations:
      i. Registered user can search a location on a map
      ii. Registered user can view the details of location on a map
      iii. Registered user can click navigation button (and jump to navigation screen on google map)

    iv.   Registered user can save a location to an existing list

    v.   Registered user can save a location to a new list

    vi.   Registered user can save a location rating

    vii.   Registered user can save a location description

    viii.   Registered user can save location tags

    ix.   Registered user can edit a location rating

    x.   Registered user can edit a location description

    xi.   Registered user can edit location tags

    xii.   Registered user can remove a location from a list

b. Lists:

    Creating a list:

    i.   Registered user can create a new list from scratch

    ii.   Registered user can add a title of a new list

    iii.   Registered user can add a description of a new list

    iv.   Registered user can add a privacy setting of a new list

    Edit a list:

    v.   Registered user can edit a title of a list

    vi.   Registered user can edit a description of a list

    vii.   Registered user can edit a privacy setting of a list

    Viewing lists:

    viii.   Registered user can view the list of lists that they own

    ix.   Registered user can view the details of the list by clicking a specific list

    x.   Registered user can view the list of lists that the people they follow own

    xi.   Registered user can view the public list of lists that other people have

    xii.   Registered user can view the public list of lists that they saved

    Manipulate the list:

    xiii.   Registered user can search list by keyword in each section

  xiv. Registered user can search sort list by added date in each section

  xv. Registered user can see how many locations saved on a specific list

  xvi. Registered user can delete a list that they own

  xvii. Registered user can filter locations in a list by rating

  xviii. Registered user can filter locations in a list by price range

  xix. Registered user can filter locations in a list by radius

  xx. Registered user can filter locations in a list by open now

  xxi. Registered user can sort locations in a list by added date

  xxii. Registered user can sort locations in a list by price

  xxiii. Registered user can sort locations in a list by rating

  xxiv. Registered user can share their list

 Saving lists:

  xxv. Registered user can save other people's public lists

  xxvi. Registered user can unsave saved lists

c. Interacting with Other Users:

  i. Registered user can search other user by their username

  ii. Registered user can see the following status

  iii. Registered user has access to public lists

  iv. Registered user can follow other user

  v. Registered user can block other user

  vi. Registered user can unfollow other user

  vii. Registered user can unblock other user

  viii. Registered user can request to follow other users if their account is private

d. Profile:

  i. Registered user can view their profile

  ii. Registered user can add their profile description

  iii. Registered user can edit their profile description

  iv. Registered user can upload their profile picture

      v.   Registered user can change their profile picture

      vi.   Registered user can edit account privacy

     vii.   Registered user can delete account

    viii.   Registered user should be able to reset the password

      ix.   Registered user can sign out

e. Notification:

       i.   Registered user gets notifications when their followings create a new list

      ii.   Registered user gets notifications when someone follow them(request to follow them)

     iii.   Registered user can turn on notifications

     iv.   Registered user can turn off notifications

Unique Features:

BucketLyst is the ultimate tool for saving and sharing your favorite locations. Create personalized lists of places you've been or want to visit and easily share them with others. Explore public lists and those from influencers you follow, gaining insider access to curated recommendations. BucketLyst seamlessly integrates with Google Maps, allowing you to effortlessly search for nearby locations or find places that are open. It's your all-in-one guiding platform. We're collaborating with influencers and YouTubers to bring you their favorite places, ensuring you have trustworthy recommendations. Trust the word-of-mouth marketing that fuels our community. Join BucketLyst today and never forget your favorite spots by creating your own personalized map of memories!

Product URL: BucketLyst

2. Usability test plan

The followings are the five major functions that we have chosen as our superior feature to be tested for usability test:

1. **Make List**
2. **Follow other user**
3. **Search List**
4. **Add Location**
5. **Save List**

Selected Function-1 For Testing Usability: Make List

a. **Test Objective** : The objective of testing the "Make List" function is to assess the ease and user-friendliness of creating a new list using the "Make List" function. The test will evaluate the process of creating a new list to ensure that it is straightforward, easy, intuitive without confusion or errors. Feedback from users will be gathered to identify potential improvements and enhance the overall user experience.

b. **Test Description:**

    -   **System Set-Up:** The system setup process is not that complicated–it is very simple and efficient. From the user's perspective, there would be no installation required. The user can launch a web browser on their computer or laptop, and make sure that the internet is connected.Our system is designed to be responsive to mobile versions as well. This means that users can access and use the BucketLyst website seamlessly on their mobile devices, such as smartphones or tablets. Currently, we support the latest two versions of Google Chrome and Mozilla Firefox as the web browsers. By typing the URL in the web browser, the user will be able to visit our BucketLyst website.

    -   **Logging In/Signing In:**Before accessing the functionality to create a new list, users need to log in or sign in to their registered account. On the BucketLyst website, users will find a login/sign-in option where they can enter their credentials, such as username and password. Once logged in, users will be directed to their personalized dashboard page.

- **Starting Point**: After signing into the website, the starting point for making a new list is the user's BucketLyst's dashboard page. User should click "Lists" button right below the "Home" button of the dashboard in order to create a new list. After that, User should see a prominent button called "Create a new list" that will guide the user to create a new list. After clicking the button, the user should be able to enter the name, location, description of the new list, and any other required information. Once all the information is entered, the user should be able to save the list and view it on their dashboard.
- **Intended User:** The intended user for the "Make List" function will be for those individuals who enjoy organizing their tasks and events but do not have much technical skill. We only allow regular users types i.e. registered, and admin users to make lists. We do not allow business users to create a new list in our application.
- **URL of the system to be tested:**
  https://master.d38dj1lszxpibn.amplifyapp.com/ownlist

Selected Function-2 For Testing Usability: Searching List
a. **Test Objectives:** The objective of testing the "Searching List" function is to evaluate the ease of searching for a specific list. The users should be able to search for a list quickly and easily, without any confusion or errors. The user's feedback on the process will help us identify any potential improvements and bring a better user experience.

b. **Test Description:**
- **System Setup:** The system setup process for searching list is similar to above, there would be no installation required. The user can launch a web browser on their computer or laptop, making sure that the internet is connected. Our system is designed to be responsive to mobile versions as well. This means that users can access and use the BucketLyst website seamlessly on their mobile devices, such as smartphones or tablets. We support the latest two versions of Google Chrome and Mozilla Firefox. By typing the URL in the web browser, the user will be able to visit our website.

- **Starting Point:** After signing in to the website, the starting point for searching a list is the user's dashboard page. Once the User clicks the "My List" button right below the "Home" button at the top left corner, they should see a search bar which will guide the user to search for a specific list. After typing in the keyword, the user should be able to view the search results and select the desired list.
- **Intended User:** The intended user for the "Searching List" function will be for those users who want to quickly find a specific item in their lists. We only allow registered, and admin users. We do not allow business users to search lists.
- **URL of the system to be tested:**
- https://master.d38dj1lszxpibn.amplifyapp.com/ownlist

Selected Function-3 For Testing Usability: Adding Location

a. **Test Objective:** The objective of testing the "Add Location" function is to evaluate the ease of adding a new location to an existing list. The users should be able to add a new location from google map quickly and easily, without any confusion or errors. The user's feedback on the process will help us identify any potential improvements and bring a better user experience.

b. **Test Description:**
- **System set-up:** Similar to above, the system setup process is fast and simple. As previously stated, there would be no installation required. The user can launch a web browser on their computer or laptop, making sure that the internet is connected. Our system is designed to be responsive to mobile versions as well. This means that users can access and use the BucketLyst website seamlessly on their mobile devices, such as smartphones or tablets. Currently, we support the latest two versions of Google Chrome and Mozilla Firefox as web browsers. By typing the URL in the web browser, the user will be able to visit our website.
- **Starting Point:** After signing in to the website, the starting point for adding a new location to an existing list is the user's dashboard page. Once the user clicks the "Home" button at the top left side of the dashboard page, they should see a

"Search" button at the top of the map. The information such as location name of the searched location would be automatically fed. Once the user decides to add to their list–all they would need to do is click the "Save" button. After selecting the list, the user should be able to add a new location by providing the necessary details such as rating, and description. Once all of the information is entered, the user should be able to save the location and view it on their list.

- **Intended User**: The intended user for the "Add Location" function will be for those registered and admin users who want to add a location for their list to organize their visited places. Only users who have created a list can add a location to that list.

- **URL of the system to be tested:**
  https://master.d38dj1lszxpibn.amplifyapp.com/home

Selected Function-4 For Testing Usability: Follow other user

a. **Test Objectives:** The objective of testing the "**Follow other user**" is to show if following users feature is easy and intuitive enough for regular users. This also tests to confirm that this function works fine.

b. **Test Description:**

- **System Setup:** The system setup process for following other users is not that complicated–it's fast and simple. From the user's perspective, there would be no installation required. Our system is designed to be responsive to mobile versions as well. This means that users can access and use the BucketLyst website seamlessly on their mobile devices, such as smartphones or tablets. The user can launch a web browser on their computer or laptop, making sure that the internet is connected.

- **Starting Point:** After signing in to the website, the starting point for following other users is Dashboard. From the Dashboard, on the side navigation bar, click the bottom icon for Users. After landing on the user page, search/click the desirable user profile, go to their profile and click the follow button.

- **Intended Users:** The intended users for the "follow other user" function will be for those individuals who like to keep up with other people's list and recommendations.
- **URL of the system to be tested:**
https://master.d38dj1lszxpibn.amplifyapp.com/users

Selected Function-5 For Testing Usability: Save List

a. **Test Objectives:** The objective of testing the "Save List" function is to evaluate the ease of saving a list after a search. The users should be able to save a list quickly and easily, without any confusion or errors. The user's feedback on the process will help us identify any potential improvements to bring a better user experience.

b. **Test Description:**
- **System Setup:** The system setup process for saving lists is the same as above– there would be no installation required. The user can launch a web browser on their computer or laptop, making sure that the internet is connected. Our system is designed to be responsive to mobile versions as well. This means that users can access and use the BucketLyst website seamlessly on their mobile devices, such as smartphones or tablets. Currently, we support the latest two versions of Google Chrome and Mozilla Firefox as web browsers. By typing the URL in the web browser, the user will be able to visit our website.
- **Starting Point**:After signing in to the website, the starting point for saving a list is the user's dashboard page. Once users enter into the dashboard, they should click the "My List" button at the second top left side of the dashboard. Once users decide to save their follower's lists, they should click the desired list they want to save into their own lists. After clicking a desired list, users should be able to save their follower's lists in their existing lists by clicking the "Save" button at the top of the list. Once the list is saved, the user should be able to view it on their dashboard.
- **Intended User**: The intended user for the "Save List" function will be for those registered and admin users who want to save their followers lists into their own

lists for future use. We do not allow business users to Save locations into lists directly from google map

- **URL of the system to be tested:**

  https://master.d38dj1lszxpibn.amplifyapp.com/listsaved

Usability test tables

Effectiveness

Test/Use case	% Completed	Errors	Comments
Create a new list with description	100 %	None	Created a new list successfully.
Add a location to a list.	80%	None	Added a location into a existing list successfully
Find a specific location name in a list.	80%	Locations do not found by created_time.	Does not support search by created time.
Follow your favorite friend in BucketLyst.	100%	None	Sent follow request successfully
Save a list for future use .	80%	None	Saved a list successfully .

Efficiency

Test/Use Cases	Average Time to Complete (in sec)	Average Time of Users Who Completed (in sec)	Average Number of Steps taken to complete	Number of Screens	Number of Pages of Instructions

Create a new list with a description.	5	5	4	2	0
Add a location to a list.	6	6	5	1	0
Find a specific location name in a list.	4	4	3	1	0
Follow other users to view their public lists.	4	4	3	1	0
Save a list for future use.	4	3	3	2	0

- Likert questionnaire: Below is a Likert-Scale Questionnaire, with questions inquiring user's satisfaction after performing the above tasks. Each "question" is a statement, with five choices (from **Strongly Agree** to **Strongly Disagree**) scaling the user's response.

	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree

The process of creating a new list was easy to understand.				✅	
I was able to add items to the list quickly and easily.			✅		
The list creation process was intuitive and user-friendly.				✅	
The process of following other users was straightforward and easy to follow.					✅
I was able to follow other users quickly.					✅
I was successfully able to follow favorite people without encountering any errors.		✅			
Adding a new location to an existing list was easy to understand and navigate.		✅			

The added location was accurately displayed on the list.			✅		
The process of adding a new location was intuitive and user-friendly.			✅		
The process of saving a list was straightforward and easy to follow.			✅		
The saved list was accurate and contained all of the necessary information.				✅	
The search function was easy to use.				✅	
The search function provided accurate results.				✅	
The search function helped me find what I was looking for.				✅	

Your Comments:

Overall the UI/UX is good, but the error shows up and the database was not working sometimes. When I shrink the window, it does not response well.

3. QA Test Plan

QA Test #1

Objective: Password Requirement

Feature to be Tested: Account will be created when password is accepted

Input: Create password

Output: Check if password passes all requirements of minimum length, at least 1 capital letter, at least 1 number, and at least one symbol.

Number	Description	Test Input	Expected Output	Pass/Fail
1	Test Password Requirement	"purple7"	Password Rejected	Pass
2	Test Password Requirement	"purple789"	Password Rejected	Pass
3	Test Password Requirement	"purple!"	Password Rejected	Pass
4	Test Password Requirement	"purple789!"	Password Rejected	Pass
5	Test Password Requirement	"Purple789!"	Password Accepted	Pass

QA Test #2

Objective: Cross-Browser Compatibility

Feature to be Tested: Style of website is uniform among different browsers

Input: Use any browser to access https://master.d38dj1lszxpibn.amplifyapp.com/

Output: Browser displays the landing page to BucketLyst

Number	Description	Test Input	Expected Output	Pass/Fail
1	Test Browser Compatibility	Microsoft Edge	Working	Pass
2	Test Browser Compatibility	Google Chrome	Working	Pass
3	Test Browser	Firefox	Working	Pass

	Compatibility			
4	Test Browser Compatibility	Opera	Working	Pass
5	Test Browser Compatibility	Safari	Working	Pass

QA Test #3

Objective: Efficiency

Feature to be Tested: User with basic computer skills, after 15 minutes of training, shall complete the task of Creating and Sharing a list within 5 minutes with no more than one error

Input: Create and share a list

Output: User successfully creates and shares a list

Number	Description	Test Input	Expected Output	Pass/Fail
1	Efficiency	User1	Successfully Created and Shared list	Pass
2	Efficiency	User2	Successfully Created and Shared list	Pass
3	Efficiency	User3	Successfully Created and Shared list	Pass

QA Test #4

Objective: Privacy

Feature to be Tested: Application will not collect more data than what the user explicitly provided

Input: Peruse through the website

Output: No output expected

Number	Description	Test Input	Expected Output	Pass/Fail
1	Privacy	User1		Pass

2	Privacy	User2		Pass
3	Privacy	User3		Pass

QA Test #5

Objective: Improve User Experience

Feature to be Tested: Application will collect user data including: name, email, and list of saved places to improve User experience without exploiting user's information

Input: Peruse through the website

Output: No output expected

Number	Description	Test Input	Expected Output	Pass/Fail
1	User Experience	User1		Pass
2	User Experience	User2		Pass
3	User Experience	User3		Pass

4. Code Review

Code Style:

In our development process, we adhere to the following coding style guidelines:

1. Separation of Responsibilities: We ensure that different functions have distinct responsibilities by organizing them into separate files. This approach promotes modularity and makes it easier to maintain and understand the code in the future.

2. Descriptive Function Names: We use descriptive function names to enhance code readability. This practice helps us better understand the purpose and functionality of each function, especially during debugging or code enhancement phases.

3. Proper Alignment: We maintain consistent and proper alignment of code elements, such as control structures (e.g., loops, conditionals), to improve code legibility. Properly aligned code allows us to quickly identify which statements are controlled by specific control structures, aiding in comprehension and debugging efforts.

4. Grouping of Statements: We group related statements together to enhance code organization and readability. By logically grouping statements, we establish a clear structure that makes it easier to comprehend the code's flow and intent. Additionally, we place a blank line between groups of statements to visually separate them, further improving code clarity.

<u>Review from the same team:</u>

```javascript
    if (notification_type === "like") {
        message = sender_name + " liked a list!";
    } else if (notification_type === "follow" && receiverIsPublic) {
        message = sender_name + " followed you!";
    } else if (notification_type === "follow" && !receiverIsPublic) {
        message = sender_name + " requested to follow you!";
    } else if (notification_type === "create") {
        message = sender_name + " created a new list!";
    } else if (notification_type === "edit") {
        message = sender_name + " edited their list!";
    }
    let query_insert_notification =
        "INSERT INTO db2.Notifications (sender_fk, receiver_fk, content, is_read) "
        + "SELECT " + user_fk + ", Followers.following_fk, '" + message + "', 0 "
        + "FROM db2.Followers WHERE Followers.followed_fk = " + user_fk + ";";

    con.query(query_insert_notification, function (err, result, fields) {
        if (err) {
            res.json( body: {error: err});
            return;
        }
        res.json( body: {success: 'post call succeed!', url: req.url, result: result, err: err, fields: fields});
    });
```

Original Code: Aaron / Reviewer: Marie

Review:

1) Overall structure and logics are clear

2) Naming is descriptive, easy to understand and follow

3) Formatting and alignment looks good and consistent

4) If you can add more inline comment for readability, that would be great

5) If-else statements can be improved

20

Review from the different team:

```javascript
app.get('/home', function (req, res) {
  // Add your code here
  const params = req['query'];
  const username = params['username'];
  const email = params['email'];
  const mysql = require('mysql');
  let userid = undefined;

  const con = mysql.createConnection({
    host: "bucketlystdb.cnzomfczmn9g.us-west-1.rds.amazonaws.com",
    user: "team1",
    password: "921382797",
    port: 3306,
    multipleStatements: true
  });

  console.log(username);
  console.log(email);

  con.connect(function (err) {
    if (err) {
      console.log("Error has occured at get request home API");
      throw err;
    }
    con.query("SELECT id from db2.PersonalUsers WHERE username = '" + username + "';",
      function (err, result, fields) {
        console.log("result is:", result[0]);

        userid = result[0];


        if (userid === undefined) {
          con.query("INSERT INTO db2.PersonalUsers (username, email, is_public) VALUES ('" +
            username + "', '" + email + "', 0); SELECT LAST_INSERT_ID();", function (err, result, field) {
              if (err) {
                console.log("error on second query");
                throw err;
              }
              userid = result;
              con.query("INSERT INTO db2.Profiles (name, user_fk, photo) VALUES ('"
                + username + "', " + userid + ", 'https://dummyimage.com/100/787878/000000.png&text=USER');", function () {
                  if (err) {
                    console.log("error on third query");
                    throw err;
                  }
                  res.json({ success: 'get call succeed! 1st pattern', url: req.url, body: userid });
                  con.end();
                });
            })
        } else {
          res.json({ success: 'get call succeed!', url: req.url, body: userid });
          con.end();
        };

      };
  });
```

Team 02's review:

First, the code looks clean and easy to read. However, I recommend removing the console.log statement for printing data. If you follow the Airbnb code style for JavaScript, it's highly recommended to use camelCase for variable names (e.g., userId). To include variables in strings, you can use the ${variable} syntax. Here's an example:
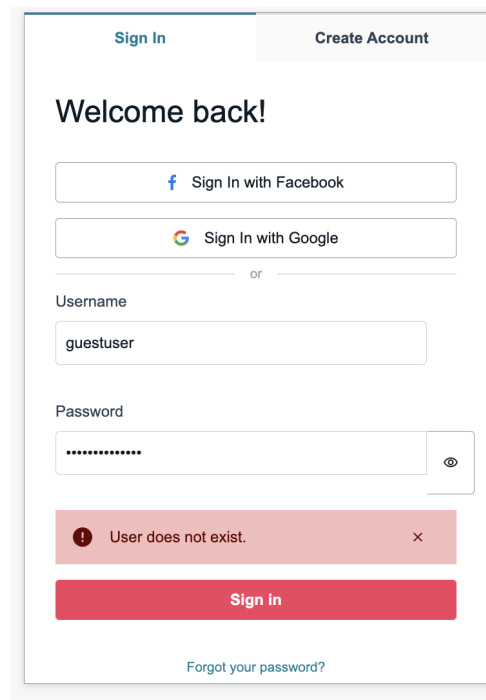
let myPet = 'seahorse';
console.log(`My favorite animal is the ${myPet}.`);

5. Self-check on best practices for security

The major assets we're protecting are primarily user login credentials. Our user login credentials are handled using AWS Cognito as part of our Amplify application. When a user signs up for BucketLyst, our front end calls on Authentication to send a request to AWS Cognito servers to add this user and also send a verification email. Upon verification the user is able to log into our website, which is verified through AWS Cognito. All the passwords and emails are stored and secured by AWS Cognito. As such, all our encryption is done in the cloud and we have no access to the raw encoded data. For more information visit AWS Cognito: https://docs.aws.amazon.com/cognito/latest/developerguide/data-protection.html.

Below are screenshots of validation for user login.

An unregistered user is stopped by validation run by AWS Cognito

Passwords have restrictions and are confirmed during registration

Email validation run sent by the Authentication API

User information stored in AWS cognito
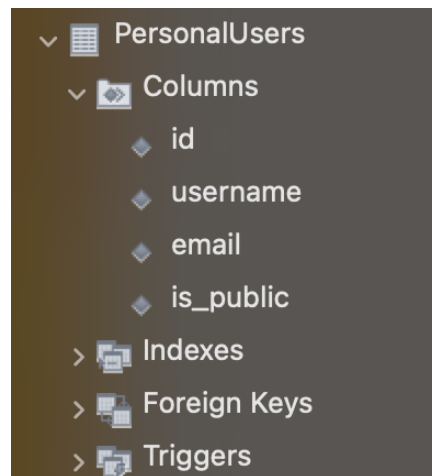
In our database schema, it does not contain password information
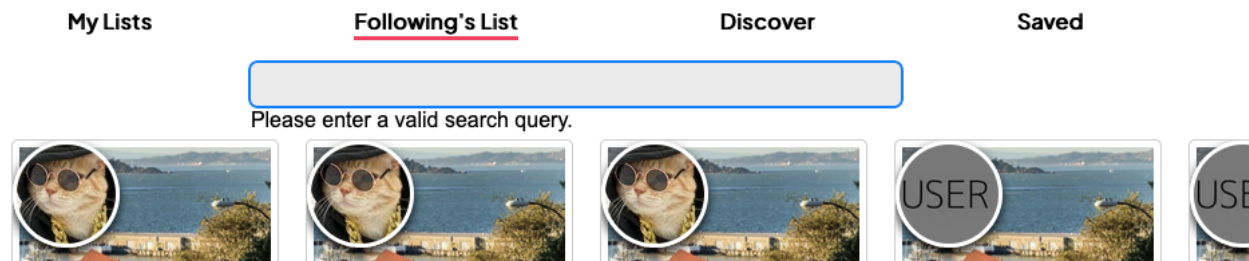
Search bars filter out trailing and leading spaces that would throw off search results. Also requires a text entry.

6. Self-check: Adherence to original Non-functional specs

1. Performance (assuming latency[1] is less than 1000 ms)

    1. Page loading time: less than 5000 ms - DONE
    a. Search bar loading time: less than 10000 ms – DONE
    b. Log-in verification time: less than 5000 ms - DONE
    c. List loading time: less than 10000 ms - DONE

2. Expected load
    a. No more than 100 active users - DONE
    b. No more than 10000 unique location entries - DONE
    c. No more than 1000 lists created - DONE
    d. No more than 100 locations in a given list - DONE

3. Fault tolerance
    a. Server downtime: no more than 5 business days - DONE

4. Security requirements
    a. HTTPS/2 protocol – DONE
    b. Unexpired SSL/TLS certificate –DONE
    c. User information won't be willingly distributed - DONE
    d. Minimum 8 character alphanumeric password with at least one special character – DONE
    e. Email validication for email – DONE

5. Availability
    a. During working hours on business days – ON TRACK
    b. The system should be available 24/7, with minimal downtime for maintenance and upgrades. – ON TRACK

6. Storage
    a. No more than 10000 GB on server storage - DONE
    b. No more than 10000 GB on database storage - DONE
    c. No more than 10 GB of client RAM - DONE
    d. No more than 10 GB of client storage - DONE
    e. The file size of images or media uploaded by users shall not exceed 2 MB. - DONE

7. User Privacy
   a. Application will not collect more data than what the user explicitly provided - DONE
   b. Application will collect user data including name, email, and list of saved places. The data shall be used only for improving the user experience and shall not be shared with any third parties without explicit user consent. - DONE

8. Compatibility
   a. The platform should be compatible with all major web browsers, including Chrome, Firefox, and Safari, as well as mobile device browsers running iOS and Android. – DONE

9. Usability
   a. The platform should be easy to use and intuitive, with a simple and clear user interface that is accessible to all users regardless of technical expertise. - DONE
   b. Users with basic computer skills, after 15 minutes of training, shall complete the task of creating and sharing a list within 5 minutes with no more than 1 error. – DONE

10. Scalability:
    a. The platform should be designed to scale easily as the user base grows, with the ability to add new features and functionality without affecting the performance of the system. - DONE

7. Team Contribution

Name	Role	Contribution
Arielle Riray 10/10	Frontend Lead	- Implementing the more difficult pages for the frontend from Figma. -Setting up workspaces and assigning react components for rest of frontend members to work off of - Worked on investigating and researching on using Google Maps API, this took the most debugging due to deprecated libraries and limited frontend access -Adjust routing for frontend pages
Francis Quang 3/10	Backend Engineer	- Worked on adherence to non-functional specs section document. - API deployment and testing on search query. - Created API for profile
Tommy Truong 7/10	Document Editor	- Worked on Settings page, share card, 2 import pages, business login, business membership, filter page, - UI fixes on on navigation bar, top bar, side bar, landing page, home page, and list page - Product Summary
Kenneth Gee 6/10	Frontend Engineer	- QA tests - Worked on components
Rabin Karki 9/10	Database Master	- Worked With Google Map API Front-End with Arielle - Worked with Usability Test Plan Section of the documentation - Worked with ER diagram -
Aaron Kuo 10/10	Github Master/ Backend Lead	- Listed out backend APIs and methodically assigned to the backend team - Set up Google Maps API connection - Provided Google Maps API key to front end team, and handed off documentation and guidelines - Reviewed backend team work, provided feedback accordingly - Facilitated knowledge transfer of search API