Software Requirements Specification

for

GoWhere

Version 1.0 approved

Prepared by

Keith Lim En Kai (U2220506C)

Lee Seungju (U2221620J)

Jesica Tjan (U2221011J)

Jerrell Yeoh Shao-En (U2220449D)

Lim Shaojun (U2221218F)

CodeCrafters

28 Jan 2024

Table of Contents

Revision History

Name	Date	Reason For Changes	Version

1. Introduction

1.1 Purpose

This Software Requirement Specification (SRS) document is created for the “GoWhere” web application, version 1.0. The primary goal of this SRS document is to outline the requirement specifications for the HawkerHub web application, streamlining the development and production processes for all stakeholders involved. Every aspect including: system features, limitations, non-functional requirements and interface specifications of the web application is documented within this SRS document.

2. System Features

<This template illustrates organizing the functional requirements for the product by system features, the major services provided by the product. You may prefer to organize this section by use case, mode of operation, user class, object class, functional hierarchy, or combinations of these, whatever makes the most logical sense for your product.>

2.1 Create Account

2.1.1 Description and Priority

Users can register for an account using their email and password at the Create Account Page..
Priority: High

2.1.2 Stimulus/Response Sequences

Flow of Events:	1. User clicks on the “Create Account” button to create an account. 2. The user will be redirected to the Account Creation page to register for a new account. 3. They will be prompted to enter their email and password, where their email will act as their username 4. A verification email will be sent to their email.
Alternative Flows:	AF-S1: User clicks on the “Sign in” button 1. The user will be redirected to the Sign In page to log into their account. (2.2 Account Login) 2. They will have the option to click “Go back” which returns them back to the homepage. 3. User clicks on the “Create Account” button to create an account. 4. The user will be redirected to the Account Creation page to register for a new account. 5. They will be prompted to enter their email and password, where their email will act as their username 6. A verification email will be sent to their email. 7.

2.1.3 Functional Requirements

1. The User must be able to create a new account using the sign-up page.
    a. When the user registers, they must provide an email and a password.
        i. The system must verify that the email provided is valid.
            1. The email provided must have a string before and after the “@” character.

2. The email provided must end with a “.com” or “.sg”

ii. The password provided must be an alpha-numeric string of length greater than 5 and less than 30 characters long.

iii. The system must verify if the email provided is already in the database

1. If the email is already in the database, the system will prompt the user to enter another email.

iv. The system must send a verification email to the user’s email, which must then be verified before they are able to log in.

2.2 Account Login

2.2.1 Description and Priority

Users are able to login to their account using their registered email and password.
Priority: High

2.2.2 Stimulus/Response Sequences

Flow of Events:	1. User launches GoWhere Web Application. 2. User clicks on the “Sign In” button at the top right hand corner. 3. User enters their email and password and clicks the “Sign In” button to login. 4. User must be brought to the Search Page of the application.
Alternative Flows:	AF-S2: User clicks on the “Create Account” 1. The user will be redirected to the Account Creation page to register their account. (2.1 Create Account) 2. They will have the option to click “Go back” which returns them back to the homepage . 3. User clicks on the “Sign In” button. 4. User enters their email and password and clicks the “Login” button to login. 5. User must be brought to the homepage of the application. AF-S3: User enters a invalid email or password 1. User must be prompted to enter a valid email or password 2. Once a valid email and password is provided, the user must be brought to the homepage of the application.

2.2.3 Functional Requirements

1. The User must be able to login to the system using their Account.
   a. The User must provide their email and password to login

    i. If the email or the password entered is invalid, an error message will be shown on screen to prompt the user to re-enter a valid email.

b. The system must verify that the email and password provided are correct

    i. Both the email and password provided by the User must match the existing user details in the database.

    ii. If it matches, the user must be able to view the homepage of the application.

2.3 Random Search

2.3.1 Description and Priority

Users will be posed a series of questions to help them get a location suggestion based on the preferences they provide.
Priority: High

2.3.2 Stimulus/Response Sequences

Flow of Events:	The questions will be flashed one by one, and questions are changed upon clicking one the choice picked. 1. User to pick 1 category (Food, Shopping, Activities, Entertainment). 2. User to tick on the exclusions based on preferences. 3. User to pick from the price ranges. 4. User to pick distance limit. 5. The user will be brought to a new page where a random location which fulfills the above criterias will be generated. 6. If the user does not like the location generated, the user is able to click on the “I’m feeling lucky” button. One more random location will be generated for the user to consider.
Alternative Flows:	AF-S1: User selected the wrong criteria, but the page went to the next question. 1. The user can click on the “Go Back” button at the bottom of the current question”. 2. The question now will be changed to the previous question. 3. The user is now able to edit the choices and continue on with the questionnaire.
Exceptions	EX1: No location which meets all criterias is found. 1. A pop-up “No location can be found! Time to edit your criteria!” will be shown. 2. The user will close the pop-up. 3. The user can then proceed to edit his/her criteria through the drop-downs to edit their criterias. 4. Click “I’m feeling lucky” to generate a new location.

2.3.3 Functional Requirements

1. The web application shall automatically change to the next question after a choice has been indicated by the user.
2. When the “Go Back” button is being clicked, the page shall change the question to the previous question.
3. For exclusion questions, the user does not have not to tick on any choices. However, they are able to also tick on at least one of the choices.
4. For exclusions questions, the user will only move on to the next question by clicking on the “Next” button.

5. After filling in the questionnaire, the web application shall change its page to the interactive map page. (next section 2.4)
    a. If at least one location meets the criterias, the location will be pinned, shown in the map page.
    b. If no location which meets all criteria is found, the web application shall display a pop-up message “No location can be found! Time to edit your criteria!”.

1. Users must be able to search for the category they desire.
    a. Users must input keywords into the search bar to search.
    b. System must filter the list of locations based on the keywords given.
        i. System must display the filtered list of locations back to the User
        ii. If no results are available, a “No results” message will be displayed on the screen.
    c. Categories include:
2. Users must be able to exclude specific categories.
3. Users must be able to choose the budget.
4. Users must be asked permission for the System to use their current location.
5. Users must be able to choose the proximity of the location given from their current location.

2.4 View Map

2.4.1 Description and Priority

Users would be able to view the locations of the specified category on the graphical map.
Priority: High

2.4.2 Stimulus/Response Sequences

Flow of Events:	1. The User signs in to their account and is brought to the search page and load the map. 2. The User will be able to zoom in and out of the map by clicking a plus or minus button respectively. 3. User will be able to click onto a place of interest on the map to see the details of that place.
Alternative Flows:	AF-S3: The user feels that the criterias are not suitable and wants to change them. 1. The user can click on the various drop-downs for the different categories. 2. The user then selects their new criterias by clicking on the choices in the drop-down. 3. The user can press on “I’m feeling lucky” to generate another location.

2.4.3 Functional Requirements

1. The system must display an interactive map that shows the locations of the selected category.
2. The User must be able to click and select any of the locations suggested to see its details.
3. The Map must be able to grow and shrink around the User’s location.
4. When current location is to be used, a pop-up shall be shown to ask for user permission.
5. Drop-downs shall be clickable.
6. Options in the drop-downs shall be clickable.

2.5 Save Locations

2.5.1 Description and Priority

Users would be able to save and unsave locations they want to revisit in the future.
Priority: High

2.5.2 Stimulus/Response Sequences

Flow of Events:	1. User views the details of a specific location 2. User clicks on “Save” button. 3. System adds the saved location to the User’s list of saved locations. 4. “Save” button changes to “Unsave”
Alternative Flows:	AF-S2: The User has already saved the location to their saved list. 1. The “Save” button has been changed to “Unsave” 2. The User clicks on the “Unsave” button. 3. The System removes the location from the User’s saved locations list. 4. “Save” button changes to “Unsave”.

2.5.3 Functional Requirements:

1. Users must be able to save a location to their list of saved locations.
2. Users must be able to unsave a location if the location has already been saved before.
    a. The System must change the “Save” option to the “Unsave” option.
3. The System must ensure that the User is valid first.
    a. The User must be logged into the Application before they can save/unsave their locations.

2.6 View Saved Locations

2.6.1 Description and Priority

Users would be able to view previously saved locations.
Priority: High

2.6.2 Stimulus/Response Sequences

Flow of Events:	1. User clicks on "View saved locations" 2. Users will be brought to a page where they can view the list of their saved locations. Each location will have an "Unsave" button beside it. 3. If a user clicks on the saved location, it will open up the details of that location.
Alternative Flows:	1. If a user clicks on the "Unsave" button, the saved location must be removed from the list.

2.6.3 Functional Requirements:

1. Users must be able to access the list of saved locations, as well as remove saved locations from the list.

2.7 View Location Details

2.7.1 Description and Priority

Users can view more details about a location if they want to know more about the location.
Priority: High

2.7.2 Stimulus/Response Sequences

Flow of Events:	1. User clicks on the information card. 2. System will call the Google Places API and pull the reviews for the specified location. 3. System will then display the additional information pulled from the Google Places API in another larger information card.
Alternative Flows:	-

2.7.3 Functional Requirements:

1. Users must be able to view more details of a specific location.
   a. These details include: Reviews, Ratings and Description.
      i. System must pull the details using the Google Places API using the name of the location.

2.8 View Nearby Places by Category

2.8.1 Description and Priority

Users would be able to view places of interests in a specified category within a set radius of the User’s location.
Priority: High

2.8.2 Stimulus/Response Sequences

Flow of Events:	1. When the User is using the map, they click on the my location icon to 2. The System requests to use the User’s location using Google’s Geolocation API. 3. The system will display a pin icon to represent the User’s current location. 4. When the User searches places of interest using the search bar, the System will use the User’s location to locate the locations of interest in the specified category within 2km of the User’s location. 5.
Alternative Flows:	-

2.8.3 Functional Requirements

1. Users must be able to select their ‘starting point’
2. Users must be able to input their travel distance
3. Users must be able to see the places of interest nearby the ‘starting point’
4. Users must be able to view the details of said places of interest upon clicking into them

2.9 View Account

2.9.1 Description and Priority

Users would be able to view the locations of the specified category on the graphical map.
Priority: High

2.9.2 Stimulus/Response Sequences

Flow of Events:	1. User clicks on “Profile” 2. Application loads in the users account page along with their details 3. If the user clicks on “Sign out”, the Application will sign them out of their account and exit the profile page.
Alternative Flows:	AF-S2: User clicks on a detail 1. Application jumps to 2.11 Update Account Details

2.9.3 Functional Requirements

1. Users must be able to view their own account
    a. 1.1. User Accounts must display their account information: Name,
    b. 1.2. User Accounts must display a list of the Users Saved Locations
2. Users must be able to sign out of their Account

2.10 Update Account Details

2.10.1 Description and Priority

Users would be able to edit and change their account's details.
Priority: High

2.10.2 Stimulus/Response Sequences

Flow of Events:	1. At account page, user clicks on details they wish to edit 2. User types in their changes 3. User clicks on "Update" button 4. Application reads in changes into the database
Alternative Flows:	-

2.10.3 Functional Requirements

1. Application must ensure that it is able to allow users to update their details in a secure and easily accessible manner
2. Application must propagate changes to the database without issue

2.11 Prompt User to Bring Umbrella

2.11.1 Description and Priority

Users would be prompted to bring an umbrella if the weather forecast predicts rainy weather at the time the User is given their randomised location.
Priority: Low

2.11.2 Stimulus/Response Sequences

Flow of Events:	1.
Alternative Flows:	-

2.11.3 Functional Requirements

3.

3. Non-functional Requirements

Non-Functional Requirements describe the properties the system must have, that is not directly related to the functional behavior of the system.

3.1 Performance Requirements

1. The landing page must provide a 8-second or less response time in a desktop browser.
2. Each search result of a user's input must be displayed within 5 seconds.
3. The application must return the correct data to the user based on the selected value of inputs.

3.2 Portability Requirements

1. The web application must be functional on the latest versions of any browser type and in the operating system MacOS, Microsoft Windows, Android and iOS.

3.3 Security Requirements

1. Upon account registration, the system must send a verification letter to the email account selected by the user.
2. The system must accept only a password with a minimum one for each special character, upper case and lower case, number.
3. The system must ensure a user enters a valid email address and password to log into the account.
4. Users must be logged in using their registered email before using the search functions of the application.

3.4 Availability

1. The web application must maintain at least a 98% daily uptime for users located in Singapore, under normal network conditions.

3.5 Localization

1. The date in the system should follow Singapore Standard Time (UTC+8:00).
2. The default language of the web application should be English.

3.6 Usability

1. Users must retrieve an output from the system within 1 minute after logging into the web application, under the assumption that no error has occurred.
2. The system must return relevant results with an accuracy of at least 95% when using a searching feature. Relevance is determined based on the parameters of distance, rating, and price chosen by the user.

3. Users must receive information regarding maintenance schedules through a dedicated screen that appears upon accessing the web application.

3.7 Reliability

1. The system must not exceed an error rate of 5% for completion of common tasks (eg. selecting customized inputs used to randomly search restaurants).

3.8 Maintainability

1. The system must maintain a mean time between failures (MTBF) of at least 30 days.
2. The system must have a maximum mean time to repair (MTTR) of 3 hours for critical failures.
3. The system must not exceed a maintenance time of more than 2 hours when the automated email verification services become unavailable.

3.9 Testability

1. The system must be capable of executing a comprehensive set of automated tests covering at least 90% of the codebase within an hour following any modifications made to the codebase.

3.10 Reusability

1. At least 60% percent of the entire codebase shall be reusable in the future projects requiring the same API.

4. Data Dictionary

Term	Definition
Places of Interest	The places that the user often frequents.
Category	The group of places that serve similar purposes or services. E.g. restaurants, supermarkets, clothing stores, etc… Type.
Area	The circular area around the chosen location that the user is willing to travel to.
Rating	The rating of the place provided by users of google maps
Budget	The amount of money that the user is willing to spend.
User	The person who is currently using the application.
Stakeholders	The organizations or people who have a vested interest in the application.
System	The application that is being used.
Login System	The part of the application that focuses on logging in.
Food Places	Places that serve food. Eateries such as Restaurants, Cafes, Hawker Centres, Food Courts, Canteens
Requirements	The baseline necessities to be met by the application.
Homepage	The page of the application that is first viewed by the user upon logging in.

5. Initial Use Case Model

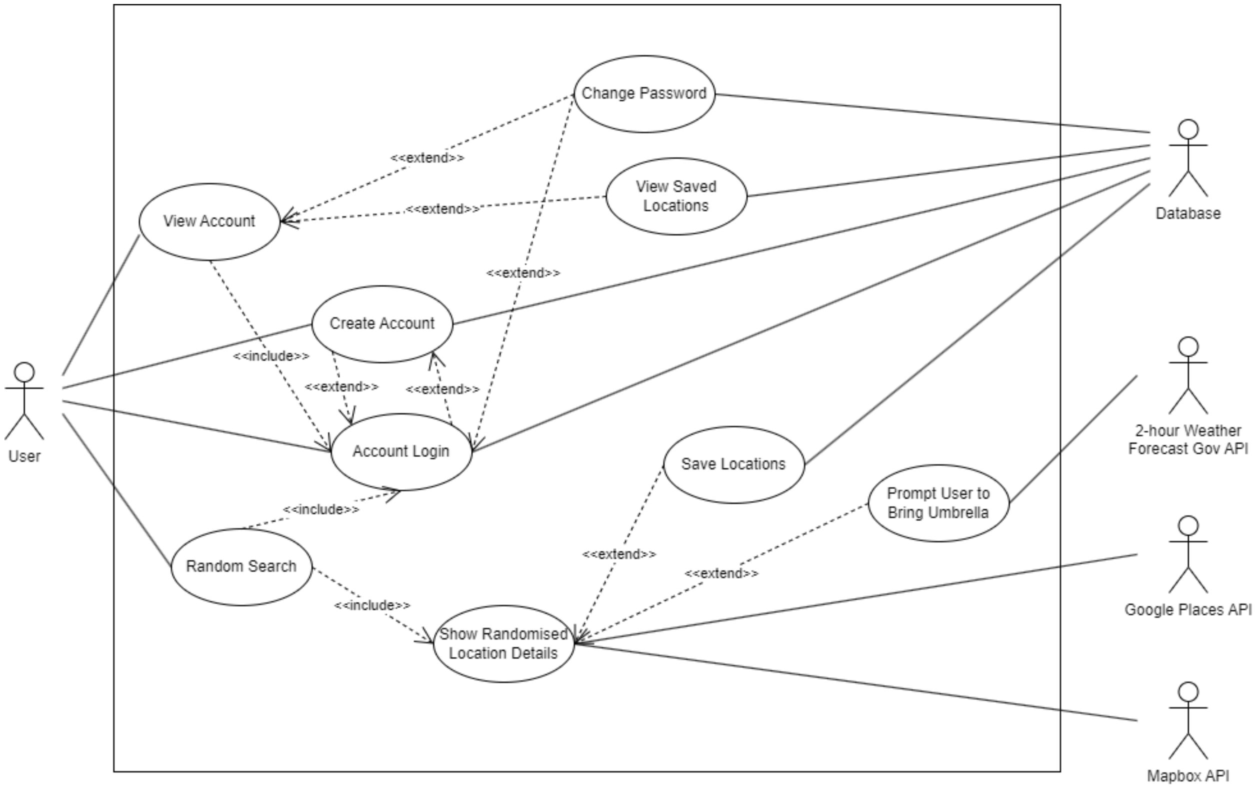


Figure 1: Use-Case Diagram

Use Case ID:	1		
Use Case Name:	Create Account		
Created By:	Shaojun	Last Updated By:	Shaojun
Date Created:	28/1/2024	Date Last Updated:	3/2/2024

Actor:	● User ● Database
Description:	Users can register for an account using their email and password.
Preconditions:	● User must provide an email and password in order to create account ● Password provided must be an alpha-numeric string of length greater than 5 and less than 30 characters long
Postconditions:	● User account created ● Re-directed to login page
Priority:	High
Frequency of Use:	High
Flow of Events:	1. User clicks on the "Create Account" button to create an account. 2. The user will be redirected to the Account Creation page to register for a new account. 3. They will be prompted to enter their email and password, where their email will act as their username 4. A verification email will be sent to their email.
Alternative Flows:	AF-S1: User clicks on the "Sign in" button 1. The user will be redirected to the Sign In page to log into their account. (2.2 Account Login) 2. They will have the option to click "Go back" which returns them back to the homepage. 3. User clicks on the "Create Account" button to create an account. 4. The user will be redirected to the Account Creation page to register for a new account. 5. They will be prompted to enter their email and password, where their email will act as their username 6. A verification email will be sent to their email AF-S3: User has an existing account with the email they provided 1. System prompts User to use a different email

Exceptions:	NIL
Includes:	NIL
Extends:	Account Login
Special Requirements:	NIL
Assumptions:	NIL
Notes and Issues:	NIL

Use Case ID:	2		
Use Case Name:	Account Login		
Created By:	Shaojun	Last Updated By:	Shaojun
Date Created:	28/1/2024	Date Last Updated:	3/2/2024

Actor:	● User ● Database
Description:	Users are able to login to their account using their registered email and password.
Preconditions:	● User has a GoWhere account
Postconditions:	● User is directed to the questionnaire page
Priority:	High
Frequency of Use:	High
Flow of Events:	1. User launches GoWhere Web Application. 2. User clicks on the “Sign In” button at the top right hand corner. 3. User enters their email and password and clicks the “Sign In” button to login. 4. User must be brought to the Search Page of the application.
Alternative Flows:	AF-S2: User clicks on the “Create Account” 1. The user will be redirected to the Account Creation page to register their account. (2.1 Create Account) 2. They will have the option to click “Go back” which returns them back to the homepage . 3. User clicks on the “Sign In” button. 4. User enters their email and password and clicks the “Login” button to login. 5. User must be brought to the homepage of the application. AF-S3: User enters a invalid email or password 1. User must be prompted to enter a valid email or password 2. Once a valid email and password is provided, the user must be brought to the homepage of the application.
Exceptions:	NIL
Includes:	NIL
Extends:	Create Account, Change Password
Special Requirements:	Email and Password are compulsory
Assumptions:	NIL

Notes and Issues:	NIL

Use Case ID:	3		
Use Case Name:	View Account		
Created By:	Shaojun	Last Updated By:	Shaojun
Date Created:	3/2/2024	Date Last Updated:	3/2/2024

Actor:	● User ● Database
Description:	User can view the email and password associated with their account, and choose to change their password or view their saved locations.
Preconditions:	● User has a GoWhere account ● User is signed into their account
Postconditions:	● User is directed to the change password page ● User is directed to the saved locations page
Priority:	High
Frequency of Use:	Medium
Flow of Events:	1. User clicks on “Profile” 2. Application loads in the users account page along which contains their email and password. 3. If the user clicks on change password, they will be redirected to the change password page. 4. If the user clicks on saved locations, they will be redirected to the saved locations page.
Alternative Flows:	NIL
Exceptions:	NIL
Includes:	Account Login
Extends:	View Saved Locations, Change Password
Special Requirements:	NIL
Assumptions:	NIL
Notes and Issues:	NIL

Use Case ID:	4		
Use Case Name:	Change Password		
Created By:	Shaojun	Last Updated By:	Shaojun
Date Created:	3/2/2024	Date Last Updated:	3/2/2024

Actor:	● User ● Database
Description:	Users would be able to edit and change their account's password.
Preconditions:	● User must have a GoWhere account ● User must be signed into their account if they were directed to this page from the account profile page
Postconditions:	● User's password will be changed in the database ● User will be redirected to the home page
Priority:	High
Frequency of Use:	Low
Flow of Events:	1. User keys in their old and new password. 2. User's new password replaces the old password in the database
Alternative Flows:	NIL
Exceptions:	NIL
Includes:	NIL
Extends:	NIL
Special Requirements:	NIL
Assumptions:	NIL
Notes and Issues:	NIL

Use Case ID:	5		
Use Case Name:	View Saved Locations		
Created By:	Shaojun	Last Updated By:	Shaojun
Date Created:	3/2/2024	Date Last Updated:	3/2/2024

Actor:	● User ● Database
Description:	Users would be able to view previously saved locations.
Preconditions:	● User must be signed into their account
Postconditions:	NIL
Priority:	High
Frequency of Use:	Medium
Flow of Events:	1. User clicks on “View saved locations” 2. Users will be brought to a page where they can view the list of their saved locations. Each location will have an “Unsave” button beside it. 3. If a user clicks on the saved location, it will open up the details of that location.
Alternative Flows:	1. If a user clicks on the “Unsave” button, the saved location must be removed from the list.
Exceptions:	NIL
Includes:	NIL
Extends:	NIL
Special Requirements:	NIL
Assumptions:	NIL
Notes and Issues:	NIL

Use Case ID:	6		
Use Case Name:	Random Search		
Created By:	Shaojun	Last Updated By:	Shaojun
Date Created:	28/1/2024	Date Last Updated:	3/2/2024

Actor:	● User
Description:	Users will be posed a series of questions to help them get a location suggestion based on the preferences they provide.
Preconditions:	● User must have an account with GoWhere to log in and use this function
Postconditions:	● User will be directed to a map with details of a randomised location.
Priority:	High
Frequency of Use:	High
Flow of Events:	The questions will be flashed one by one, and questions are changed upon clicking one the choice picked. 1. User to pick 1 category (Food, Shopping, Activities, Entertainment). 2. User to pick from the price ranges. 3. User to pick distance limit. 4. The user will be brought to a new page where a random location which fulfills the above criterias will be generated. 5. If the user does not like the location generated, the user is able to click on the "I'm feeling lucky" button. 3 more random locations will be generated for the user to consider.
Alternative Flows:	AF-S1: User picked the food category. (or other categories) 1. Exclusions (not yet done) AF-S2: User selected the wrong criteria, but the page went to the next question. 1. The user can click on the "Go Back" button at the bottom of the current question". 2. The question now will be changed to the previous question. 3. The user is now able to edit the choices and continue on with the questionnaire.
Exceptions:	NIL
Includes:	Account Login, Show Randomised Location Details
Special Requirements:	NIL
Assumptions:	NIL
Notes and Issues:	NIL

Use Case ID:	7		
Use Case Name:	Show Randomised Location Details		
Created By:	Shaojun	Last Updated By:	Shaojun
Date Created:	3/2/2024	Date Last Updated:	3/2/2024

Actor:	● User ● Google Places API ● Mapbox API
Description:	User will see details of the randomised place they have been given, such as its name, ratings and reviews, address, and location on the map.
Preconditions:	● Location given must be within Singapore
Postconditions:	● User can choose to be directed to Google Maps for directions. ● User can choose to save this location.
Priority:	High
Frequency of Use:	High
Flow of Events:	1. Users answer the questionnaire or select the customisation options on the map page 2. Users are allocated a randomised location based on their customisations. 3. Users can click on the Google Maps link to find directions. 4. Users can click on the bookmark icon to save this location in their account.
Alternative Flows:	AF-S3/4: Users click on the “I’m feeling lucky” button 1. Users will be provided with a new randomised location
Exceptions:	NIL
Includes:	NIL
Extends:	Save Locations, Prompt User to Bring Umbrella
Special Requirements:	NIL
Assumptions:	NIL
Notes and Issues:	NIL

Use Case ID:	8		
Use Case Name:	Save Locations		
Created By:	Shaojun	Last Updated By:	Shaojun
Date Created:	3/2/2024	Date Last Updated:	3/22024

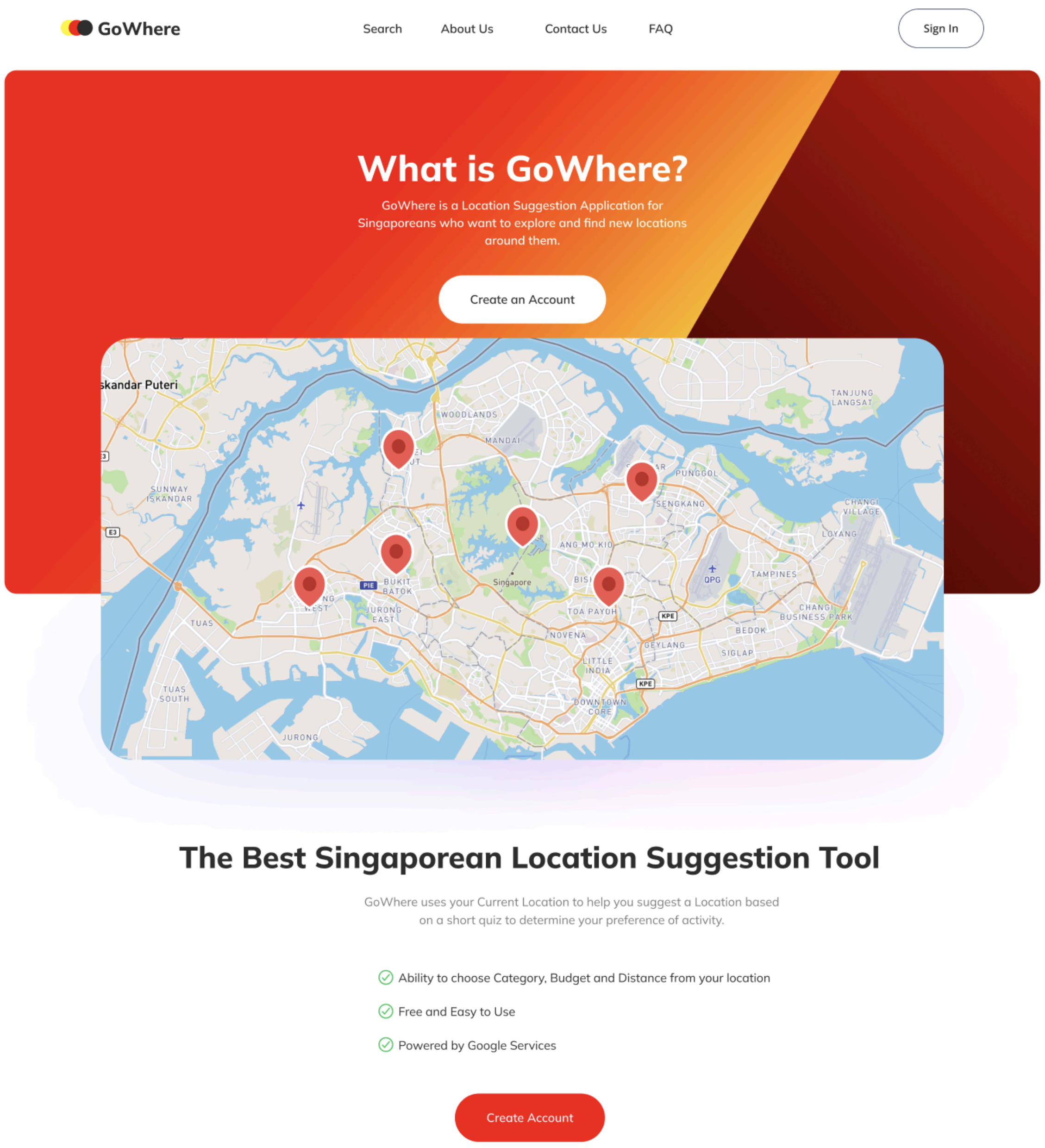
Actor:	● User ● Database
Description:	Users would be able to save and unsave locations they want to revisit in the future.
Preconditions:	● User must be signed into their account
Postconditions:	● Location details will be added to Saved Locations page
Priority:	High
Frequency of Use:	Medium
Flow of Events:	1. User views the details of a specific location 2. User clicks on “Save” button. 3. System adds the saved location to the User’s list of saved locations. 4. “Save” button changes to “Unsave”
Alternative Flows:	AF-S2: The User has already saved the location to their saved list. 1. The “Save” button has been changed to “Unsave” 2. The User clicks on the “Unsave” button. 3. The System removes the location from the User’s saved locations list. 4. “Save” button changes to “Unsave”.
Exceptions:	NIL
Includes:	NIL
Special Requirements:	NIL
Assumptions:	NIL
Notes and Issues:	NIL

Use Case ID:	9		
Use Case Name:	Prompt User to bring Umbrella		
Created By:	Shaojun	Last Updated By:	Shaojun
Date Created:	3/2/2024	Date Last Updated:	3/2/2024

Actor:	● User ● 2-hour Weather Forecast Gov API
Description:	User will be prompted to bring an umbrella for their trip if the weather forecast indicates that it will rain soon.
Preconditions:	● User has been given a randomised location on the map
Postconditions:	● User is prompted to bring an umbrella
Priority:	Low
Frequency of Use:	High
Flow of Events:	1. If the system detects that there is going to be or is raining through the weather API, the system will prompt the User to bring an Umbrella.
Alternative Flows:	AF-S1: If the system does not detect that there will be rain, they will not send a prompt
Exceptions:	NIL
Includes:	NIL
Special Requirements:	NIL
Assumptions:	NIL
Notes and Issues:	NIL

6. UI Mockups

6.1 Home Page



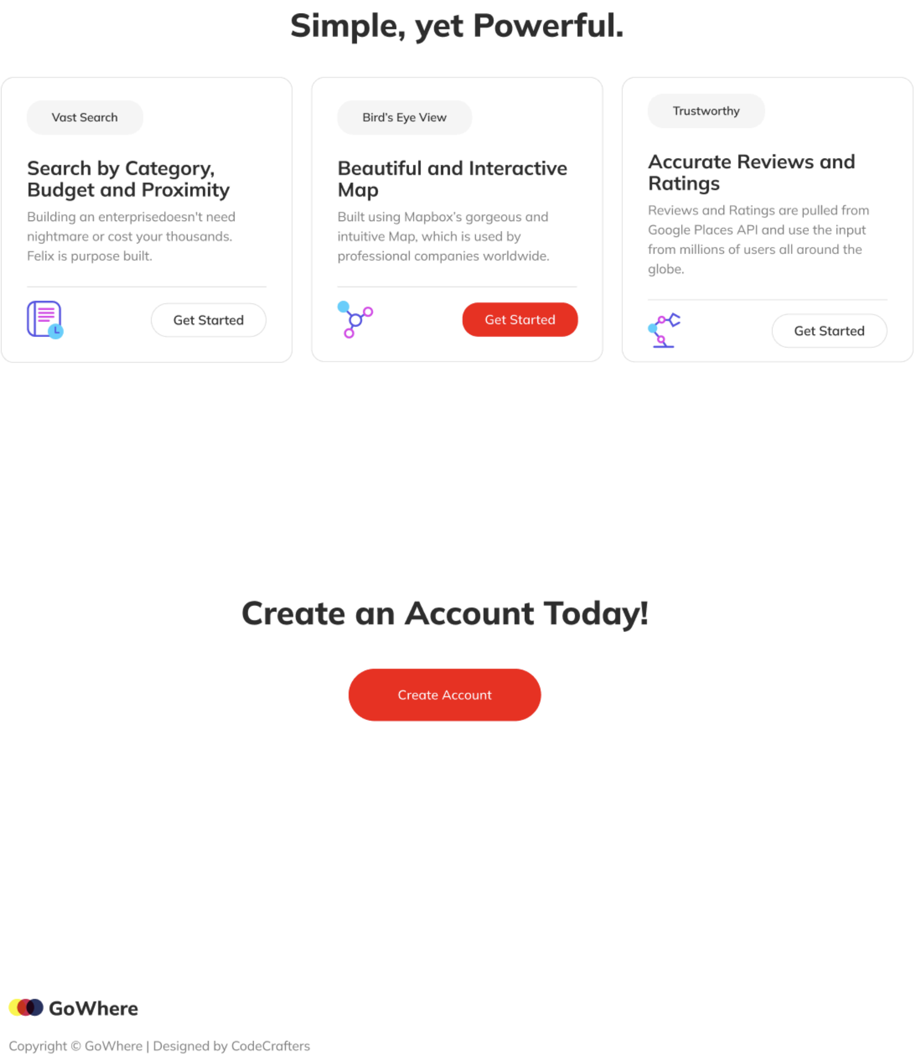
What is GoWhere?

GoWhere is a Location Suggestion Application for Singaporeans who want to explore and find new locations around them.

The Best Singaporean Location Suggestion Tool

GoWhere uses your Current Location to help you suggest a Location based on a short quiz to determine your preference of activity.

- ✓ Ability to choose Category, Budget and Distance from your location
- ✓ Free and Easy to Use
- ✓ Powered by Google Services



Simple, yet Powerful.

Vast Search

Search by Category, Budget and Proximity

Building an enterprisedoesn't need nightmare or cost your thousands. Felix is purpose built.

Bird's Eye View

Beautiful and Interactive Map

Built using Mapbox's gorgeous and intuitive Map, which is used by professional companies worldwide.

Trustworthy

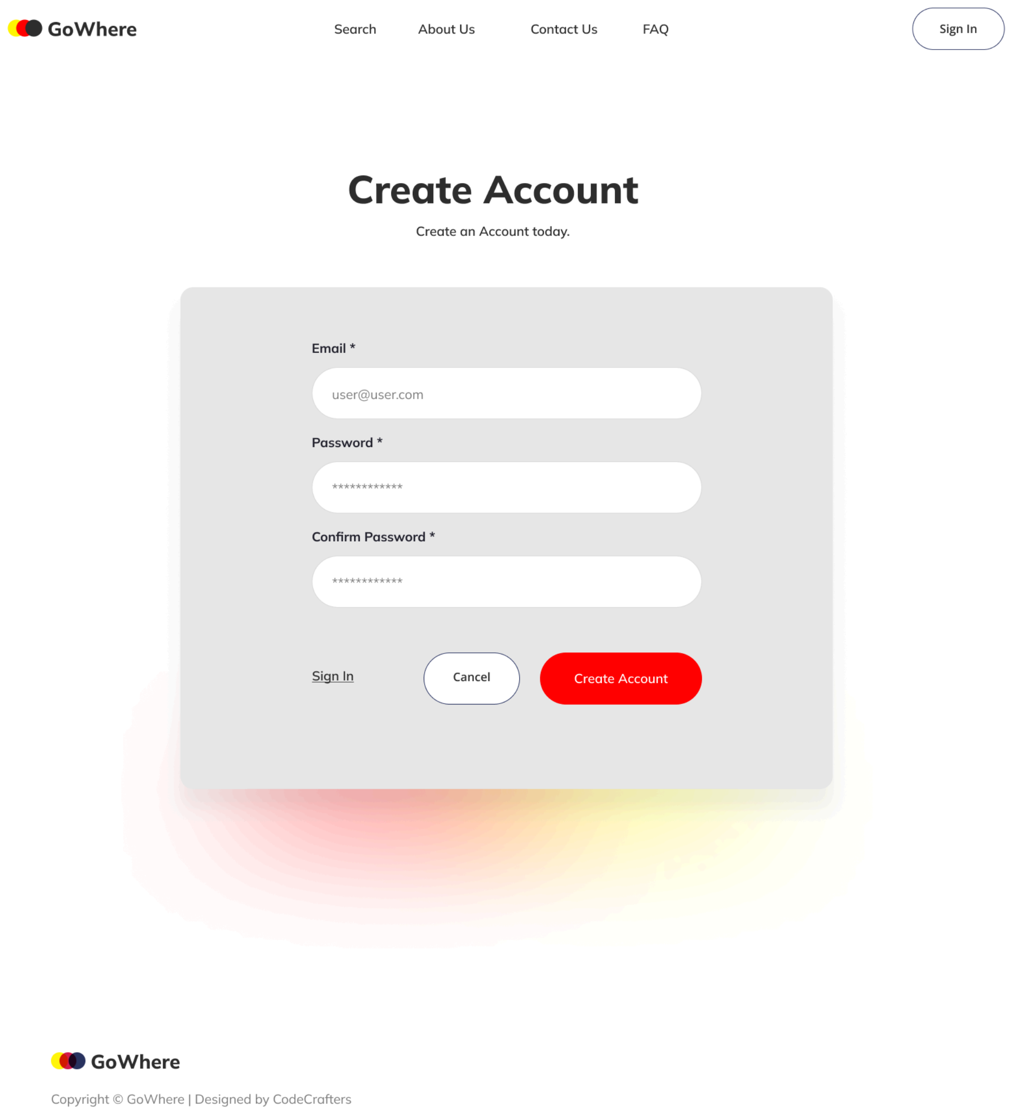
Accurate Reviews and Ratings

Reviews and Ratings are pulled from Google Places API and use the input from millions of users all around the globe.

Create an Account Today!

GoWhere

6.2 Create Account Page



Create Account

Create an Account today.

Email *

user@user.com

Password *

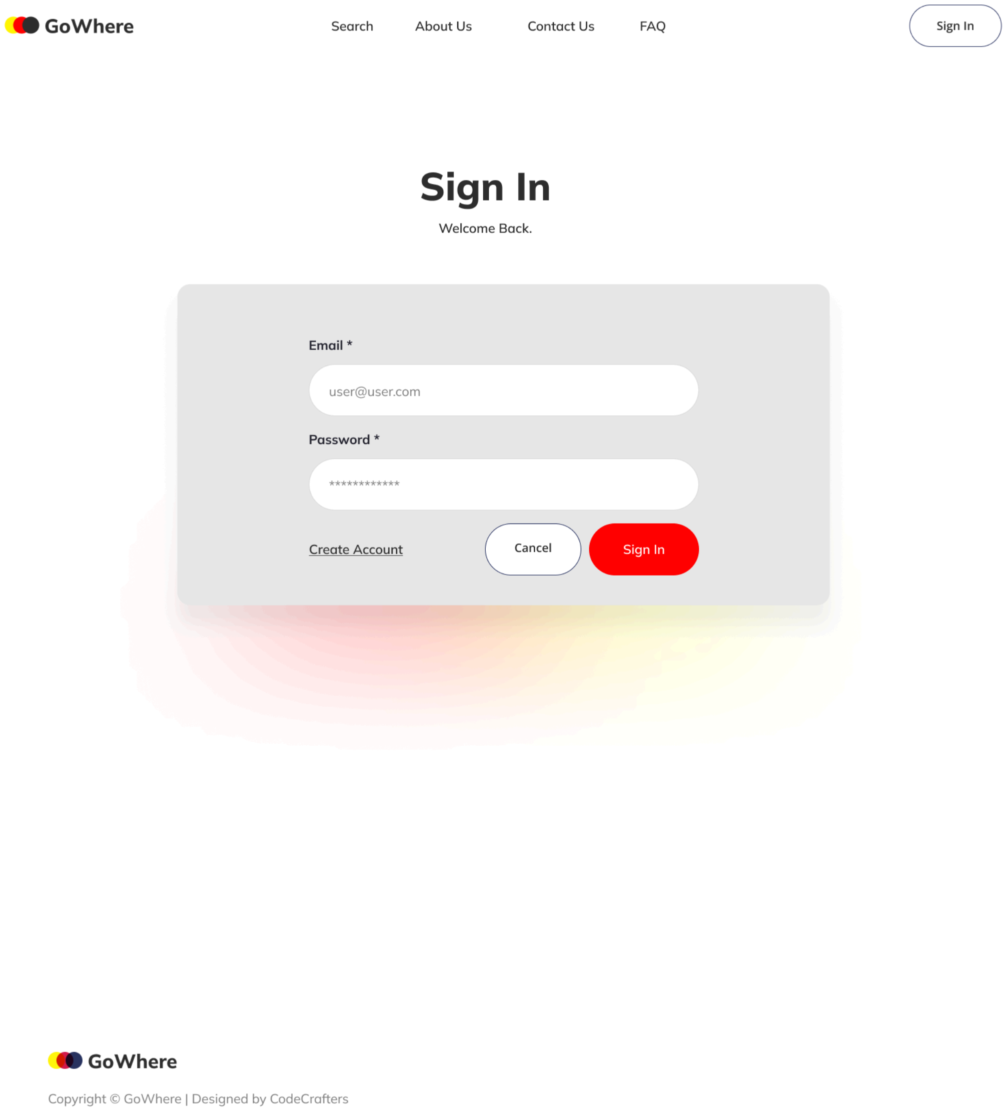
****

Confirm Password *

****

Copyright © GoWhere | Designed by CodeCrafters

6.3 Login Page



Sign In

Welcome Back.

Email *

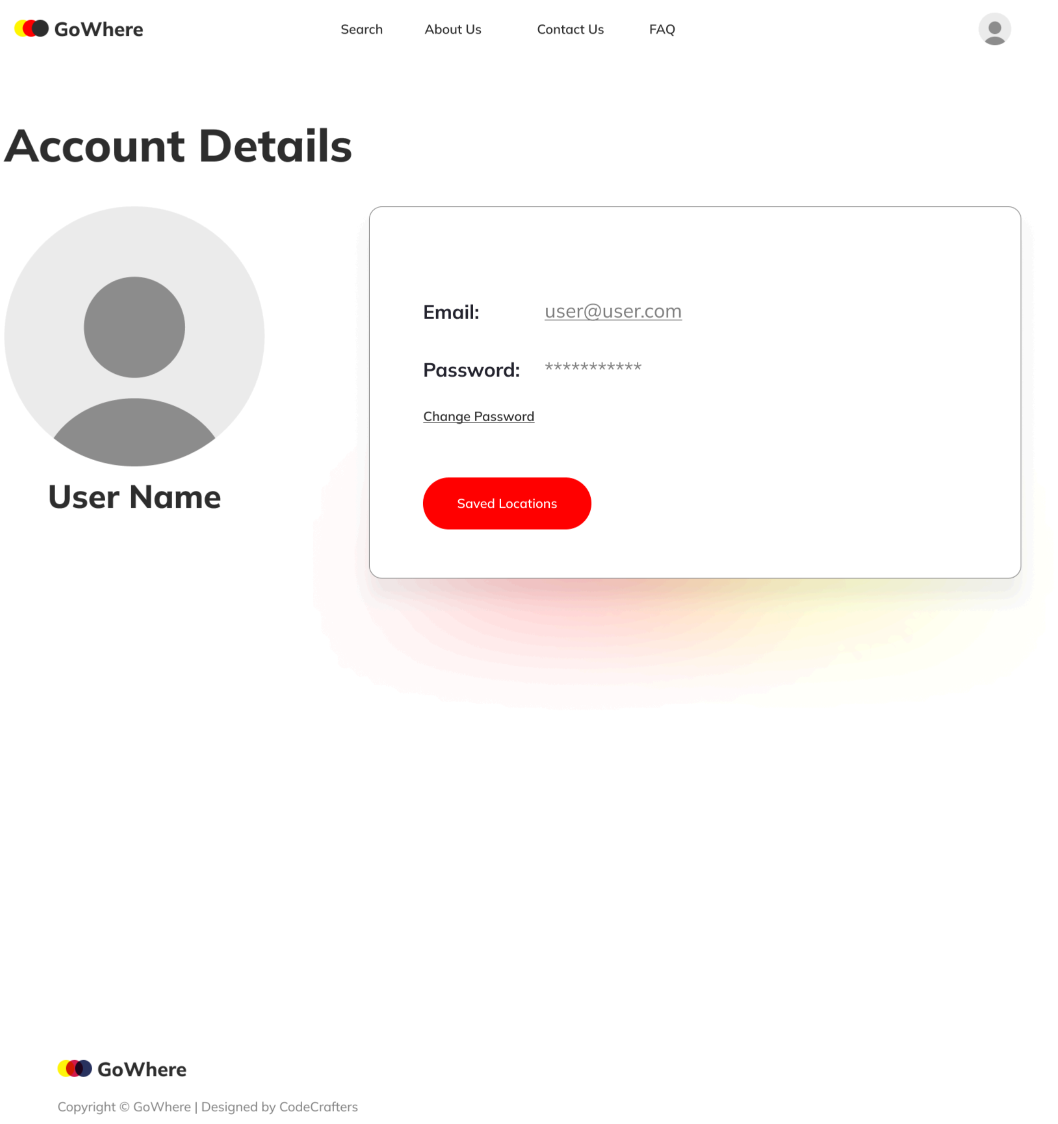
user@user.com

Password *

****

Copyright © GoWhere | Designed by CodeCrafters

6.4 Account Details Page



Account Details

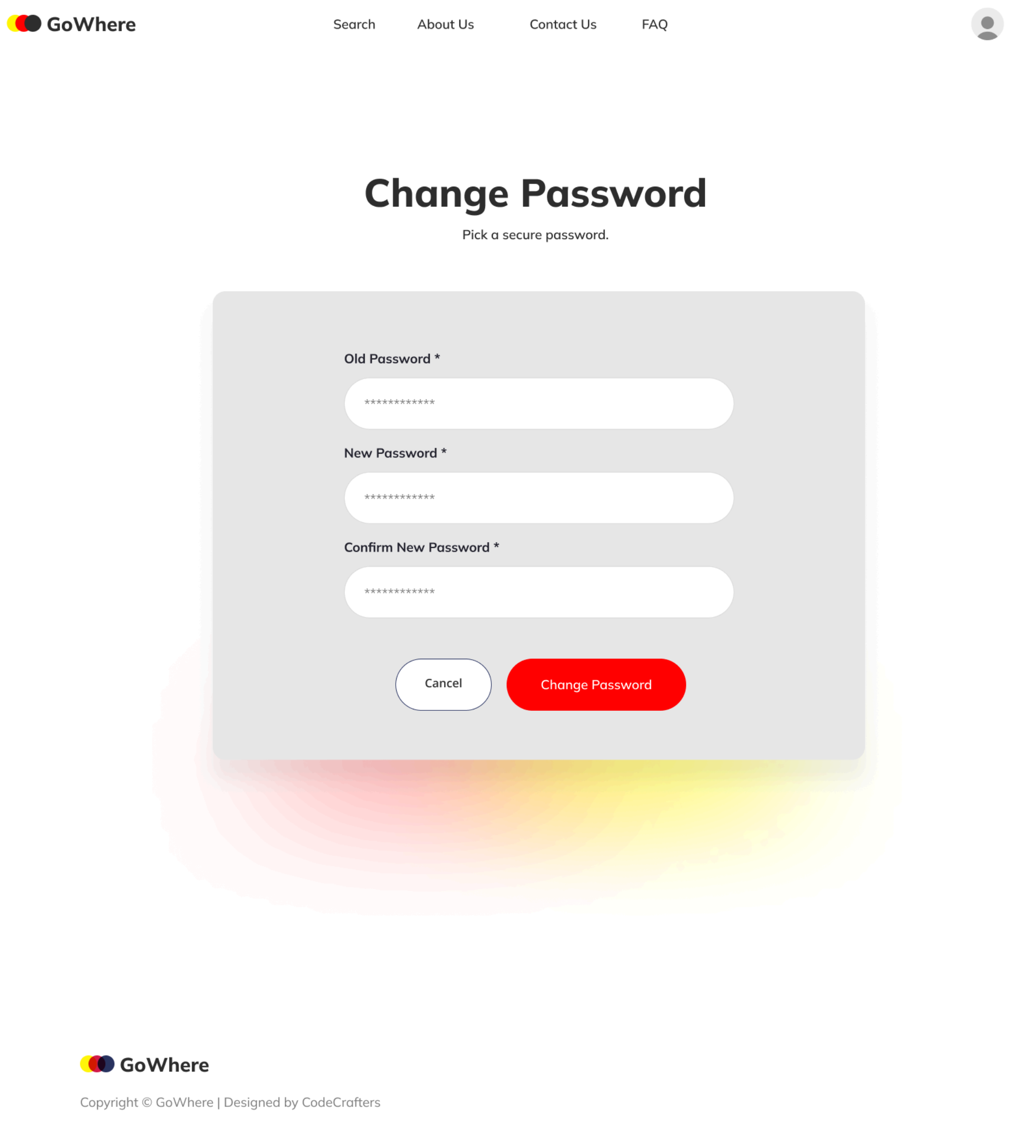
User Name

Email: user@user.com

Password: ***********

Copyright © GoWhere | Designed by CodeCrafters

6.5 Change Password Page



Change Password

Pick a secure password.

Old Password *

****

New Password *

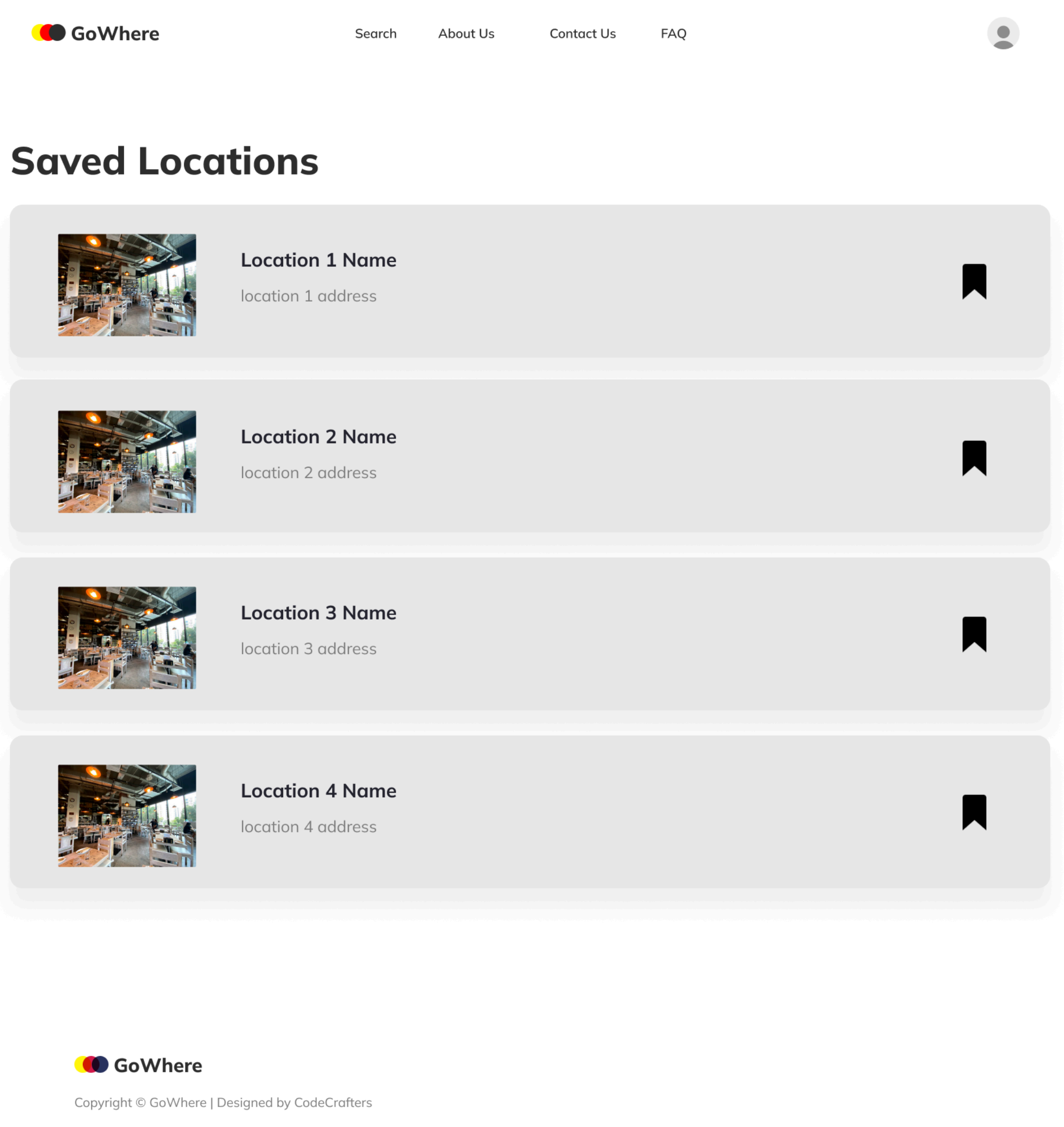
****

Confirm New Password *

****

Copyright © GoWhere | Designed by CodeCrafters

6.6 Saved Locations Page



Saved Locations

Location 1 Name

location 1 address

Location 2 Name

location 2 address

Location 3 Name

location 3 address

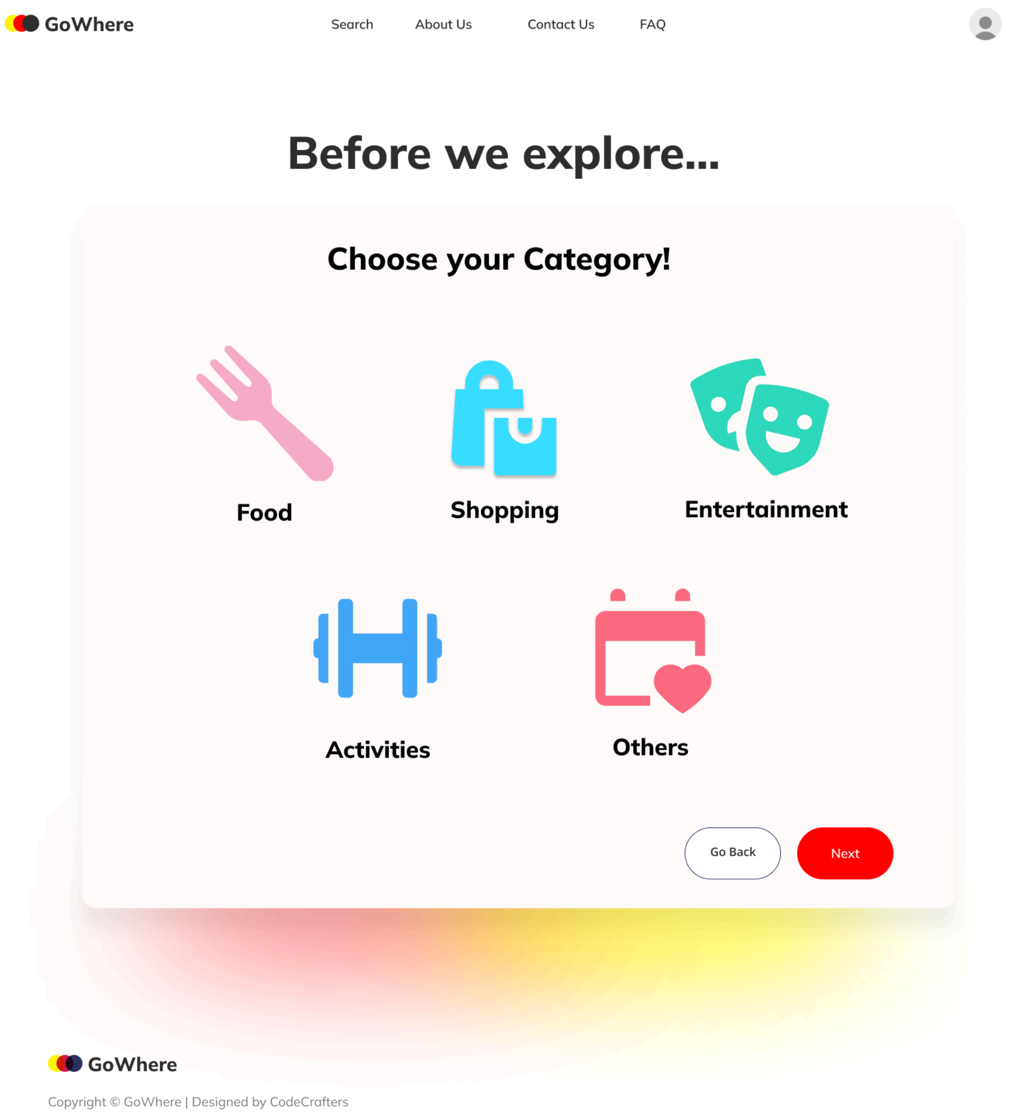
Location 4 Name

location 4 address

GoWhere

Copyright © GoWhere | Designed by CodeCrafters

6.7 Questionnaire Page

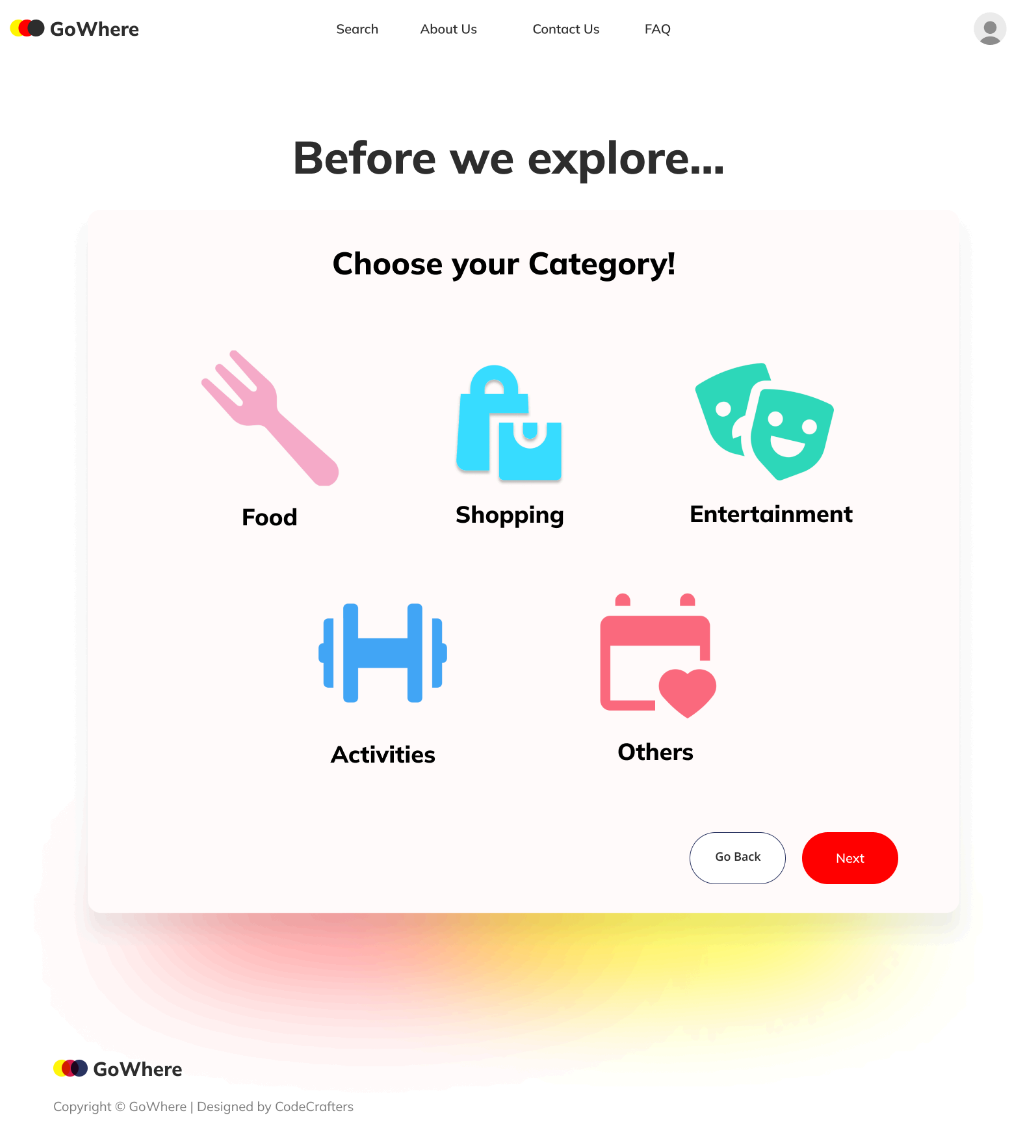


Before we explore...

Choose your Category!

Copyright © GoWhere | Designed by CodeCrafters

6.8 Map Page



Before we explore...

Choose your Category!

Copyright © GoWhere | Designed by CodeCrafters