1. Overview

This document will serve as a summary of Lab 1. The summary includes the expected deliverables of the lab, as well as the detailed write-up of the respective deliverables.

2. Deliverables

The following defines the deliverables to be submitted to the Lab TA prior to Lab 2:

2.1 Team formation and team leader assignment

The team will choose an appropriate team name. The team will also appoint a member of the team as the team leader. The team must fill in the Team Registration Form as instructed by Lab TA.

2.2 Documentation of Functional Requirements (FR) and Non-Functional Requirements (NFR)

The team will proceed to brainstorm a project idea which will serve as guidance for the team project. The team will also formulate the target audience of the project. The team will proceed to liaise with the stakeholders to elicit FR and NFR. The team will then formulate a documentation which contains the FR and NFR.

The team will use SRS_Template.doc, under sections *4. System Features* and *5. Other Nonfunctional Requirements* as references for the template.

2.3 Data Dictionary

The team will create a data dictionary which includes important terms used throughout the project. The team will ensure that the data dictionary contains attributes of, and the relationship between each term.

2.4 Use Case Models

The team will identify certain use cases based off the FR. The team will proceed to depict the use cases using a Use Case Diagram.

The team shall make sure that each use case is accompanied by a use case description. The description will explain the interaction between an end-user and the system to carry out a functionality.

2.5 User Interface (UI) Mock-ups

The team will use relevant software tools, or hand-drawn methods (if applicable) produce a mock-up of the project's preliminary user interface.

3. Project Ideas

The following are the proposed project ideas:

3.1 Carpark availability application

The web application will provide a heatmap of the carparks around Singapore. The heatmap will depict the number of available parking spots in each carpark.

3.1.1 Feedbacks

The complexity of the initial project idea is not sufficient to demonstrate the work of a 5-man project.

3.2 Taxi availability application

The web application will provide a heatmap of the live location of all taxi around Singapore.

3.2.1 Feedbacks

A web application for taxi hailing may not be appropriate. The team considered that a taxi hailing application should be a mobile application instead

3.3 COVID-19 cases heatmap

The web application will provide a heatmap of the live COVID-19 cases within Singapore.

3.3.1 Feedbacks

The team agrees that the project idea is overused and lacks innovation.

3.4 Dengue fever heatmap

The web application will provide a heatmap of the live Dengue fever cases within Singapore.

3.4.1 Feedbacks

The team agrees that the project idea is overused and lacks innovation.

3.5 Music recommendation web application

The web application will allow the user to compile a list of favourite music. The list shall then be used to recommend the user other music of which the system deem the user may be interested in.

3.5.1 Feedbacks

A web application that allows a user to compile a list of music is redundant, as ordinary music players such as Spotify contains said features. The project idea can retain as a supplementary feature and should be built on top of an alternate project idea.

3.6 AI Shopping Tracking System (Selected)

The web application will track the user's search pattern. Based on the search pattern, the system will recommend the user a list of items that the system predicts the user may be interested in. The web application will also perform a cross-platform check to find the best deals of the items. The best deal includes the cheapest price and rebates, whichever is applicable.

The team has decided to select *3.6 AI Shopping Tracking System* as the project idea.

4. Team Name

The following are the proposed team names:

4.1 FindR (Selected)

The name *FindR* mimics the pronunciation of *Finder* and provides a catchy feeling to the customers.

Since there are no alternate suggestions or proposals, the team has unanimously decided to select *4.1 FindR* as the team's name.

5. Target Audience

Based on the selected project idea, the following characteristics shall define the target audience:

5.1 Users who lack time to perform shopping physically.

We are targeting workers who have packed schedules. We strongly believe the AI shopping recommendation system will help reduce time taken to shop for goods. The cross-platform comparison feature will also help to save the time needed to navigate through multiple stores or platforms to find the best deals.

5.2 Users who live far away from physical convenience stores.

We are targeting customers whose location are inconvenient for physical shopping. An online AI shopping recommendation system will aid the customers in their daily online shopping. The cross-platform comparison feature will further serve to aid the customers in finding the best deals of their everyday online shopping.

5.3 Users who are home-bound or have mobility issues.

We are targeting elderlies or disabled customers. The customers could already be engaged with online shopping activities. Thus, an online AI shopping recommendation system with cross-platform comparison feature will certainly aid the customers to make better and informed decisions.

5.4 Tech-savvy users.

We are targeting young adults who are mostly familiar with navigating the online world. Our online AI shopping recommendation system will aid the customers to make informed decisions in their purchases.

6. Functional Requirements (FR)

The following are the proposed preliminary FR:

6.1 The user must be able to register for an account with our system and login subsequently.

6.2 If the user has forgotten his/her login credentials, he/she must be able to seek help to recover the lost account.

6.2 When the user searches for an item, our system must be able to recommend at least three other items to the user.

6.3 The system must be able to retrieve all information of the searched items such as price, rating, delivery fee, payment methods available and any relevant rebates from the e-commerce platform.

6.4 The system must be able to provide a set of parameters such as price, number of purchases, form of rebate, delivery fee, payment methods and rating which can be tweaked by the user to sort the result.

6.5 The system must be able to provide the best deal for the user according to the parameters set by the user.

6.6 The user must be able to compile a wish list of items which must be sold on at least one e-commerce platform.

6.7 The user must be able to send another user a friend request using their usernames.

6.8 The user must be able to accept or reject the friend request received.

6.9 The user must be able to view his/her friends' wish list and their respective birthdays.

7. Non-Functional Requirements (NFR)

The following are the proposed preliminary NFR:

7.1 The system must be able to successfully register for an account for the user after the user fills in all the details required within 15 seconds.

7.2 The system must be able to retrieve and display the search result of the user within 30 seconds.

7.3 The system must be able to support searches from at least three e-commerce platform.

7.4 The system must not be down for more than three hours in one year.

7.5 The system must hash the user's credential information using Secure Hash Algorithm (SHA) before storing them on the database.

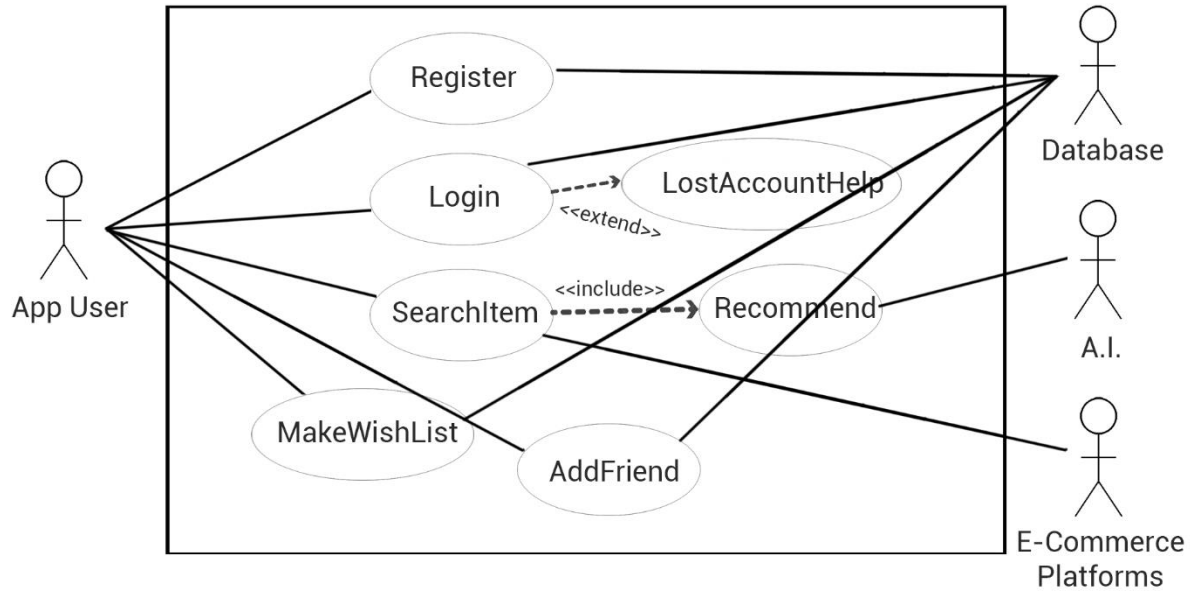
7.6 The system must be able to display FAQ information in the local language of the user based on the user's location.

8. Data Dictionary

Created By:	Lee Juin	Last Updated By:	Lee Juin
Date Created:	19th August 2022	Date Last Updated:	30th August 2022

App User	An individual who holds an account with the FindR web application. The account must be formally registered via the FindR web application and must be retrievable from the Database. The individual is then entitled to use all the services provided within the web application, which includes but not limited to, adding other App Users, searching for an item using keywords and adding an item to a wish list, subject to the Terms of Use which the individual has agreed upon registration.
E-commerce platform	An API that provides all relevant information about an item sold on a particular platform, which includes the name, the price, the rating, the delivery fee, the payment methods available and any relevant rebates. The API returns the information upon requested (searched) by the App User.
A.I.	A machine learning model which is trained using the data retrieved by the API to recommend an App User three other relevant items based on the searched item. The model tracks the App User search pattern and predicts what the App User may be interested in.
Wish list	A list which contains all the items that are added by the App User. The items in the wish list are defined as preferred items by the App User to be bought as gift by his/her friends. The wish list is publicly accessible by the App User and his/her friends.
One-time Password (OTP)	A six-digit combination which is sent to the App User via his/her registered email address. The OTP serves as an additional layer of security in the event where an App User forgets his/her login credentials. The OTP sent will expire within five minutes. An App User may request for an additional OTP after every 60 seconds.
Username	A unique identification set by an App User which serves as a locator of him/her. An App User may find other App Users by searching for their username.
Database	An online spreadsheet which contains all information of each App User such as their username, hashed registered email address, hashed password, hashed name, hashed birthday, and wish list items. The sensitive personal details are hashed using Secure Hash Algorithm (SHA).
Delivery fee	A column under each item which displays the cost of delivering an item if the App User purchases it. The delivery fee column is displayed in Singapore Dollar (SGD) currency. The delivery fee will not be displayed if the item is sold out.
Rating	A scaling system implemented by individual e-commerce platforms for past customers to rate the bought items. The scale of the system is from one star to five star, where one star represents poorest experience, and five star represents best experience. The rating column displays the average of all rates given by the past customers.
Rebate	A promotional code that is applicable to a particular item which is sold on a particular e-commerce platform. Rebates are usually issued by the e-commerce platform during grand sales or by the individual sellers. Thus, rebates are not cross-platform compatible. The rebate column displays any potential rebates which can be applied to the item, provided either by the platform or the seller. Rebates are presented either in a ratio-based discount (i.e., 20% off) or flat-based discount (i.e., $ 20 off). In the case of flat-based discount, the discount is in Singapore Dollar (SGD) currency.

9. Use Case Diagrams



10. Use Case Description

Use Case ID:	001		
Use Case Name:	Register		
Created By:	Lee Juin	Created By:	Lee Juin
Date Created:	25th August 2022	Date Created:	30th August 2022

Actor:	App User (Initiating), Database
Description:	The App User can register for an account that is stored in the Database using this use case.
Preconditions:	1. The Database must be up and online. 2. The App User must be connected to the Internet.
Postconditions:	The App User has successfully registered an account for the web application with a unique username and password. *Or* The App User is notified of the reason(s) why the registration of an account is unsuccessful.
Priority:	
Frequency of Use:	
Flow of Events:	1. At the home page of the website, the App User clicks on "Sign up" and is redirected to the registration page. 2. The App User inputs a valid email, a username, a password that contains at least an upper-case letter, a lower-case letter and a digit, and the repeated password into the respective fields in the submission form. 3. The App User checks the tick box of "I agree to the Terms of Use and Privacy Policy" and clicks on "Sign Up". 4. The system verifies the username is unique and the password satisfies the constraints. 5. The App User inputs a One-Time Password (OTP) that is sent to his/her email inbox by the system. 6. The system stores the App User's information in the database securely. 7. The App User is notified that the registration is successful.
Alternative Flows:	AF-1: The App User did not check the tick box of "I agree to the Terms of Use and Privacy Policy". 1. When the App User clicks on "Sign Up", the system displays the message "Please tick the checkbox for acknowledging our Terms of Use and Privacy Policy!" above the submission form. 2. The system returns to Step 2 and waits for the App User inputs. AF-2: The App User left input field(s) blank. 1. When the App User clicks on "Sign Up", the system displays the message "Please ensure all fields have been filled up before submitting!" above the submission form. 2. The system returns to Step 2 and waits for the App User inputs. AF-3: The App User inputs a taken username. 1. The system displays the message "Username has been taken. Please try again!" above the submission form. 2. The system returns to Step 2 and waits for the App User inputs. AF-4: The App User inputs a password that do not satisfy the given requirements.

	1. The system displays the message "Password does not meet the required standards" above the submission form. 2. The system returns to Step 2 and waits for the App User inputs. AF-5: The App User inputs mismatched passwords. 1. The system displays the message "Passwords do not match" above the submission form. 2. The system returns to Step 2 and waits for the App User inputs. AF-6: The App User inputs an incorrect OTP. 1. The system displays the "Security Check" submission form again. 2. The system displays the message "Incorrect OTP! Please try again!" above the submission form. 3. The system returns to Step 5 waits for the App User inputs.
Exceptions:	EX-1: The App User repeatedly attempts to register for an account for more than ten times despite errors in input. 1. On the 11th attempt, when the App User clicks on "Sign Up", the system displays the message "Too many attempts! Please try again in 10 minutes." above the submission form. 2. The "Sign Up" button is unavailable for ten minutes. 3. The system only accepts registration from the App User's IP address after ten minutes. EX-2: The App User did not receive the OTP in his/her email inbox. 1. The "Resend another OTP" button is available after 60 seconds. 2. The App User clicks on the "Resend another OTP" button to resend another OTP to his/her email inbox. 3. The system returns to Step 5 and waits for the App User inputs. EX-3: The App User requests for more than three resent of OTP. 1. The system displays the message "Please try again with a different email" above the submission form. 2. The system returns to Step 2 and waits for the App User inputs.
Includes:	
Special Requirements:	
Assumptions:	
Notes and Issues:	

Use Case ID:	002		
Use Case Name:	Login		
Created By:	Lee Juin	Created By:	Lee Juin
Date Created:	25th October 2022	Date Created:	30th October 2022

Actor:	App User (Initiating), Database
Description:	The App User can login to his/her account with the correct credentials that are stored securely in the Database.
Preconditions:	1. The Database must be up and online. 2. The App User must be connected to the Internet. 4. The App User has a registered account.
Postconditions:	The App User has successfully logged into his/her application account. *Or* The App User is notified of the reason(s) why he/she is unable to login to his/her account.
Priority:	
Frequency of Use:	
Flow of Events:	1. At the home page of the website, the App User clicks on "Log in" and is redirected to the login page. 2. The App User inputs his/her username and password. 3. The App User clicks on "LOGIN". 4. The system verifies the credentials provided with the Database. 5. When the information is verified, the App User is redirected to his/her account dashboard.
Alternative Flows:	<u>AF-1: If the App User inputs an incorrect username or password</u> 1. When the App User clicks on "LOGIN", the system displays the message "Invalid username and/or password!" above the submission form. 2. The system returns to Step 2 and waits for the App User inputs. <u>AF-2: The App User left input field(s) blank</u> 1. When the App User clicks on "LOGIN", the system displays the message "Please ensure all fields have been filled up before submitting!" above the submission form. 2. The system returns to Step 2 and waits for the App User inputs.
Exceptions:	<u>EX-1: The App User inputs incorrect username or password for more than five times</u> 1. On the sixth attempt at logging in, when the App User clicks on "LOGIN", the system displays the message "Account suspended. Please try again after 10 minutes" above the submission form. 2. The "LOGIN" button is unavailable for ten minutes. 3. The system only accepts registration from the App User's IP address after ten minutes. <u>EX-2: The App User forgot his/her login credentials</u> 1. The App User clicks on "Forget Password?" on the login page. 2. The App User can recover his/her account using the extended use case *LostAccountHelp*.

Includes:	
Extends:	*LostAccountHelp*
Special Requirements:	
Assumptions:	
Notes and Issues:	

Use Case ID:	003		
Use Case Name:	LostAccountHelp		
Created By:	Lee Juin	Created By:	Lee Juin
Date Created:	25th October 2022	Date Created:	30th October 2022

Actor:	App User (Initiating), Database
Description:	The App User can request for help if he/she lost access to his/her account using this use case.
Preconditions:	1. The Database must be up and online. 2. The App User must be connected to the Internet. 3. The App User has forgotten his/her login credentials.
Postconditions:	The App User has successfully recovered his/her account by changing his/her credentials. *Or* The App User has contacted support to seek further assistance.
Priority:	
Frequency of Use:	
Flow of Events:	1. The App User clicks on "Forgotten?" on the login page. 2. The system displays the recover account page. 3. The App User inputs his/her registered email and clicks on "Recover Account". 4. The system displays a "Security Check" submission form. 5. The App User inputs the one-time password (OTP) that has been sent to his/her email inbox. 6. The system displays a "Change Security Details" submission form. 7. The App User inputs a new set of username and password and clicks on "Change". 8. The system verifies that the username is unique, and the password satisfies the given requirements before updating the App User's information in the database securely. 9. The App User is informed of the successful change in credentials and is redirected back to the login page.
Alternative Flows:	<u>AF-1: The App User did not receive the OTP in his/her email inbox</u> 1. The "Resend another OTP" button is available after 60 seconds. 2. The App User clicks on the "Resend another OTP" button to resend another OTP to his/her email inbox. 3. The system returns to Step 5 and waits for the App User inputs. <u>AF-2: The App User entered an incorrect, but registered email address</u> 1. When the App User realises that he/she has inputted an incorrect email address, the App User clicks on "Not email@serviceprovider.com?". 2. The system returns to Step 3 and waits for the App User inputs. <u>AF-3: The App User entered an incorrect, and not registered email address</u> 1. When the App User clicks on "Recover Account", the system displays the message "Email not registered!" above the submission form. 2. The system returns to Step 3 and waits for the App User inputs
Exceptions:	<u>EX-1: The App User forgot his/her registered email.</u> 1. The App User clicks on "Contact Support".

	2. The system redirects the App User to the FAQ page that contains the support email address. 3. The App User contacts the support via email to retrieve access of his/her account.
Includes:	
Special Requirements:	
Assumptions:	
Notes and Issues:	

Use Case ID:	004		
Use Case Name:	SearchItem		
Created By:	Jerick Lim Kai Zheng	Last Updated By:	Jerick Lim Kai Zheng
Date Created:	22nd August 2022	Date Last Updated:	30th August 2022

Actor:	App User (Initiating), AI
Description:	The App User will be able to search for items with keywords. The AI will retrieve the information of the items from the e-commerce platforms for the App User to sort the search results based on parameters given to provide the best deal.
Preconditions:	1. The AI is up and online. 2. The App User is connected to the Internet. 3. The App User registered for an account with the Database. 4. The App User has logged in to his/her account.
Postconditions:	The App User obtained a list of searched items based on the keywords and parameters inputted. *Or* The App User is unable to obtain a search result based on the keywords inputted.
Priority:	
Frequency of Use:	
Flow of Events:	1. The App User types a keyword in the search box and clicks on the search icon. 2. The AI searches, based on the keyword, for items sold on the e-commerce platforms. 3. The AI retrieves the information of the searched items from the e-commerce platforms such as rating, price, number of items sold, payment method, rebates, and delivery fee. 4. The AI displays the searched items along with the searched keyword and the number of results retrieved. 5. The AI recommends at least three other items to the App User using the included use case *Recommend*. 6. The AI provides a set of parameters such as price, number of purchases, form of rebate, delivery fee, payment methods and rating to sort the results. 7. The App User sorts the results using the parameters and clicks on "Apply". 8. The AI re-displays the list of times based on the parameters set. 9. The AI computes the best deal for the App User based on the parameters set.
Alternative Flows:	<u>AF-1: The AI returns a list of items which includes sold out items.</u> 1. The AI displays a list of items sold on the e-commerce platforms based on the keyword inputted by the App User. 2. A "Sold Out" symbol is further displayed over the items which are sold out. 3. The AI returns to Step 1 and waits for the App User to input another keyword. <u>AF-2: The App User inputs nothing and clicks on the search icon.</u> 1. The AI displays a list of 100 random items. 2. The AI displays "Random items" instead of the searched keywords. 3. The AI displays 100 as the number of results.

	4. The AI returns to Step 1 and waits for the App User to input another keyword. AF-3: The parameters set by the App User does not match any searched items 1. The AI will not display any search results. 2. The system will display a message "No items that matches the filters set!" instead. 3. The AI returns to Step 7 and waits for the App User to re-adjust the parameters. AF-4: There is no relevant rebate of the searched item 1. The AI will not show any rebates. 2. A message that says, "No relevant rebates." is displayed instead at the rebate column.
Exceptions:	EX-1: The AI System is unable to retrieve any items based on the keyword. 1. The AI will not display any search results. 2. The system will display a message "No results found! Sorry we cannot find any results for your search item." instead. 3. The AI will not allow the App User to adjust the parameters. 4. The AI will return to Step 1 and waits for the App User to input another keyword.
Includes:	*Recommend*
Special Requirements:	
Assumptions:	
Notes and Issues:	

Use Case ID:	005		
Use Case Name:	Recommend		
Created By:	Jerick Lim Kai Zheng	Last Updated By:	Jerick Lim Kai Zheng
Date Created:	22nd August 2022	Date Last Updated:	30th August 2022

Actor:	App User (Initiating), AI
Description:	Based on the searched items, the AI recommends the App User other similar items using this use case.
Preconditions:	1. The AI System is up and online. 2. The App User is connected to the Internet. 3. The App User registered for an account with the Database. 4. The App User has logged in to his/her account.
Postconditions:	App User obtains a list of recommended items based on the searched items.
Priority:	
Frequency of Use:	
Flow of Events:	1. The App User inputs a keyword and clicked on the search icon. 2. The AI returns a list of items based on the keyword. 3. The AI recommends at least three other related items based on the searched item. 4. When the App User scrolls to the bottom of the page, the App User can view the section of "You may also like:" which displays the recommended items.
Alternative Flows:	AF-1: The AI is unable to retrieve any items based on the keyword. 1. The AI display three items based on the keyword instead as recommended items to the App User. 2. The AI returns to Step 1 and waits for the App User to input another keyword.
Exceptions:	EX-1: The App User inputs nothing and clicked on the search icon. 1. When the App User inputs nothing, the AI will display 100 random items. 2. The AI will not display any recommended items.
Includes:	
Special Requirements:	
Assumptions:	
Notes and Issues:	

Use Case ID:	006		
Use Case Name:	AddFriend		
Created By:	Oi Yeek Sheng	Last Updated By:	Oi Yeek Sheng
Date Created:	22nd August 2022	Date Last Updated:	30th August 2022

Actor:	App User (Initiating Actor), Database
Description:	The App User can send, accept, and reject friend request using this use case. The App User can also receive birthday notification of his/her friends using this use case.
Preconditions:	1. The App User must be logged in to an account. 2. The App User must have enabled the Receive Birthday Notification feature.
Postconditions:	The App User successfully sent a friend request to another App User. *Or* The App User accepts the friend request of another App User. *Or* The App User rejects the friend request of another App User. *Or* The App User is prompted about his/her friends' birthday 7 days in advance
Priority:	
Frequency of Use:	
Flow of Events:	1. The App User adds friends by searching the unique username of each App User. 2. If the username searched is available in the Database, the App User may send a friend request to that user. 3. When the other App User accepts the friend request, the App User can now navigate to his/her friend's Profile page and view the friend's wish list and birthday date. 4. The App User may opt in to the Receive Birthday Notification feature. 5. When the App User connects with another App User, the App User is notified about his/her friend's birthday 7 days in advance.
Alternative Flows:	<u>AF-1: If the other App User rejects the friend request</u> 1. The system sends an email to the App User about the friend request being rejected. <u>AF-2: The App User has not set up his/her birthday details</u> 1. When the App User navigates to his/her Profile page for the first time, the App User is asked to set up his/her personal details, including birthday. 2. If the App User chooses not to set up his/her personal details and attempts to Step 4, the system displays a message "Please set up your personal details prior to opting in for this feature!" to the App User.
Exceptions:	<u>EX-1: If the searched username is not available</u> 1. The system displays the message "The searched username is not available. Please insert a valid username."

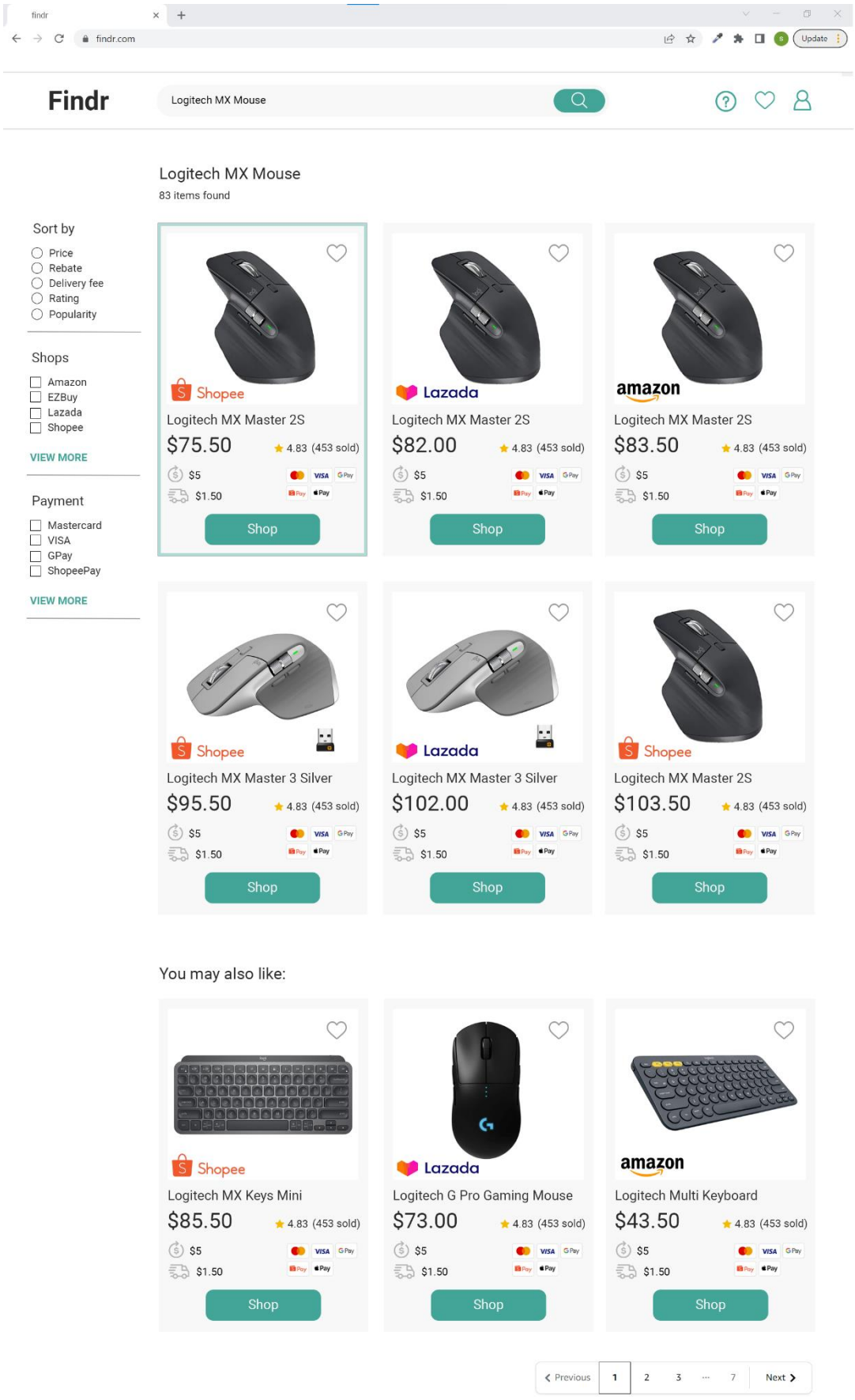
	EX-2: The App User's friend has not set up his/her birthday details 1. The App User will not be notified of his/her friend's birthday.
Includes:	
Special Requirements:	
Assumptions:	
Notes and Issues:	

Use Case ID:	007		
Use Case Name:	MakeWishList		
Created By:	Oi Yeek Sheng	Last Updated By:	Oi Yeek Sheng
Date Created:	22nd August 2022	Date Last Updated:	30th August 2022

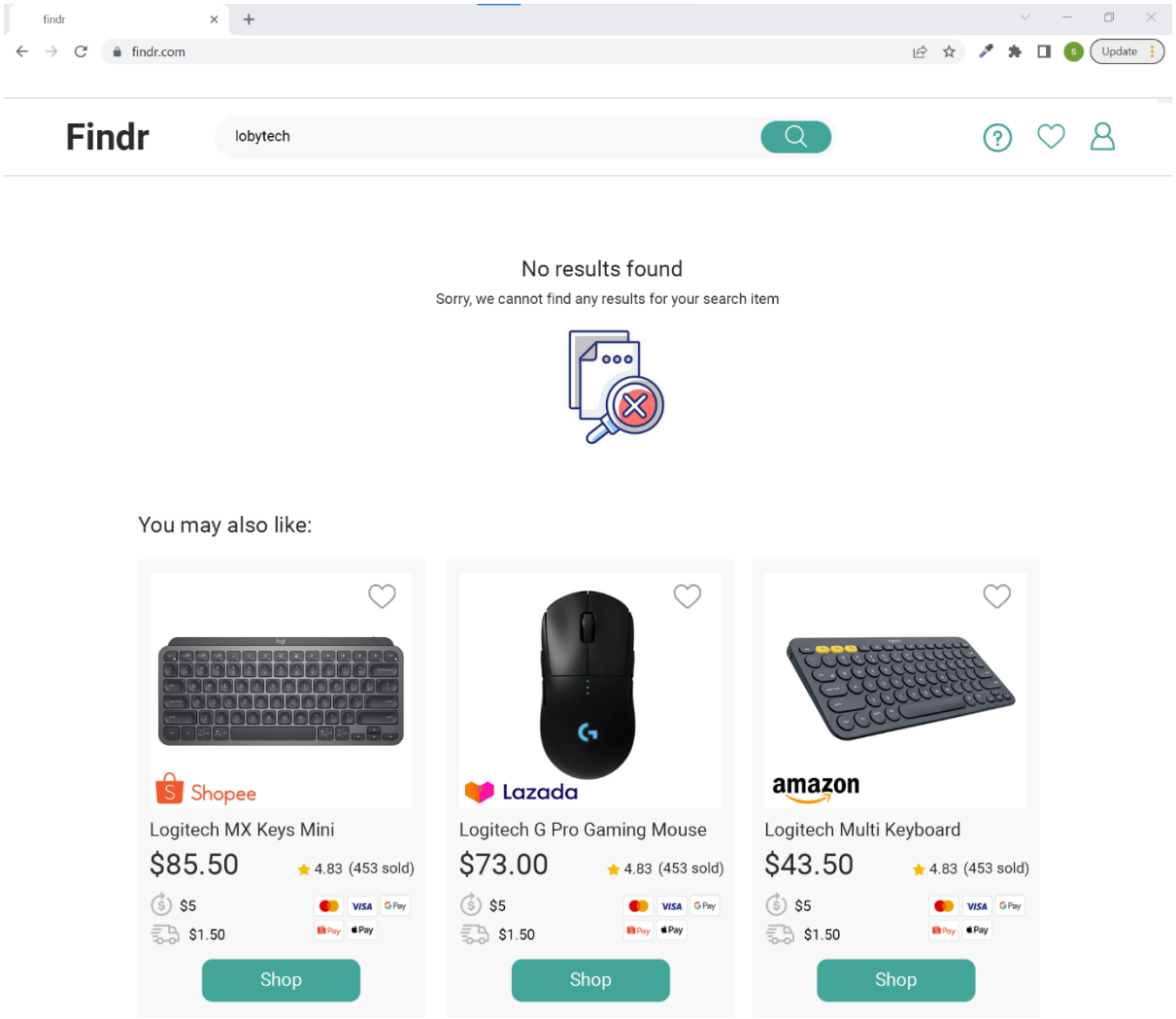
Actor:	App User (Initiating Actor), Database
Description:	The App User can create a wish list of items sold on E-commerce platforms and can view his/her, and his/her friends' wish list using this use case.
Preconditions:	The App User must be logged in to an account.
Postconditions:	The App User successfully adds an item to his/her wish list. *Or* The system displays the App User's, or his/her friend's wish list.
Priority:	
Frequency of Use:	
Flow of Events:	1. The App User searches for an item using the search function provided in the system. 2. If the item is available on at least one e-commerce platform, the system displays the items available to the App User. 3. The App User selects the item to be added to his/her wish list. 4. The App User can view his/her wish list underneath his/her profile. 5. The App User may click on the Friends navigation panel. 6. The App User can view his/her friends' wish list by clicking on the Profile icon next to each entry on the friends' list.
Alternative Flows:	<u>AF-1: If the item is sold out on the supported e-commerce platforms</u> 1. The item is displayed with a "Sold Out" message. 2. When the App User adds the item to his/her wish list, the App User is prompted with a warning message that says "Warning! The added item is currently sold out and may not be available anytime soon!" <u>AF-2: If the App User has not yet created a wish list</u> 1. When the App User enters his/her Profile page, the App User is prompted with a message to create his/her wish list if he/she chooses to. 2. The App User may respond with "Sure!" or "Maybe later". 3. The former will create an empty wish list for the App User to add items later. The latter will cancel the prompted message.
Exceptions:	<u>EX-1: If the item is not sold on the supported e-commerce platforms</u> 1. The system displays the message "No results found." 2. The system recommends at least one similar item to the user. 3. If the App User selects the recommended item, the system will add the item to his/her wish list. <u>EX-2: If the App User's friend has not yet created a wish list</u> 1. When the App User enters his/her friend's Profile page, the system displays a message "Sorry! The user has not yet created any wish list!" underneath the profile.
Includes:	
Special Requirements:	

Assumptions:	
Notes and Issues:	

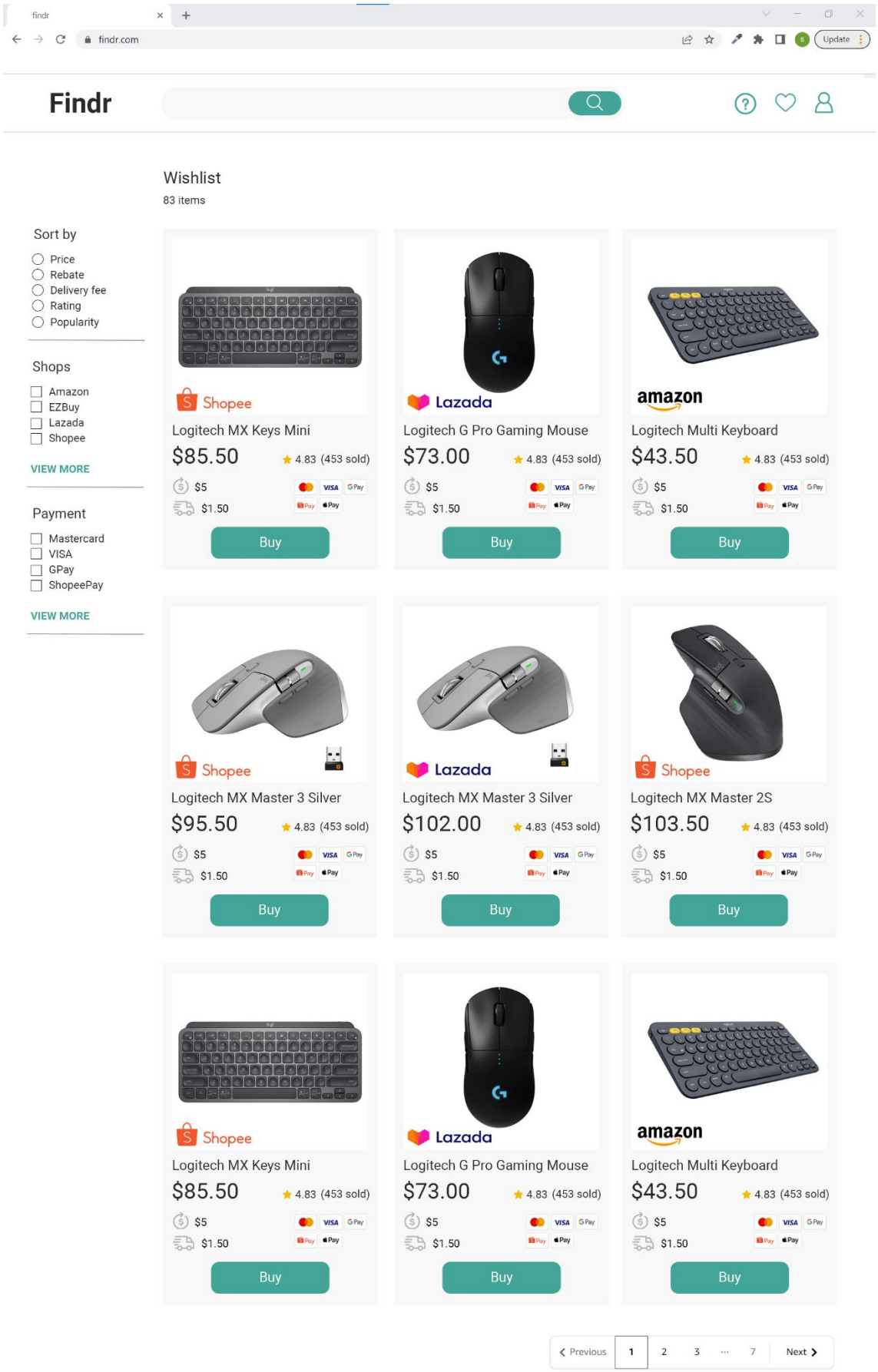
Search Item Page



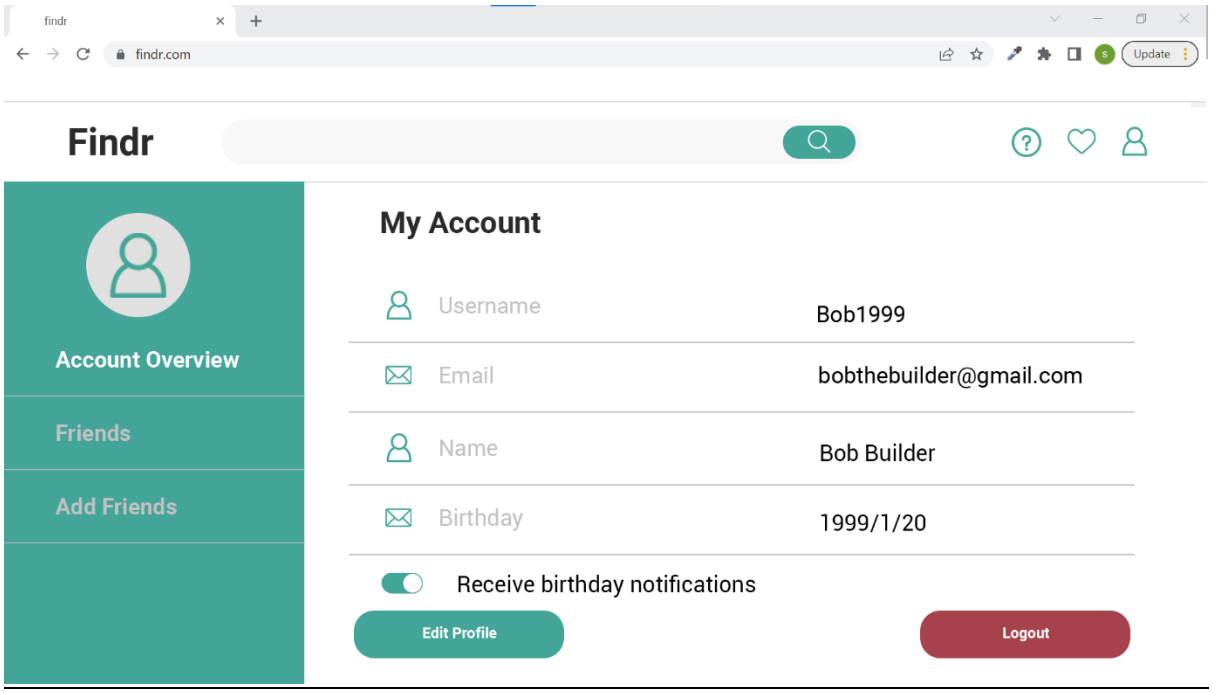
Search Item Not Found Page



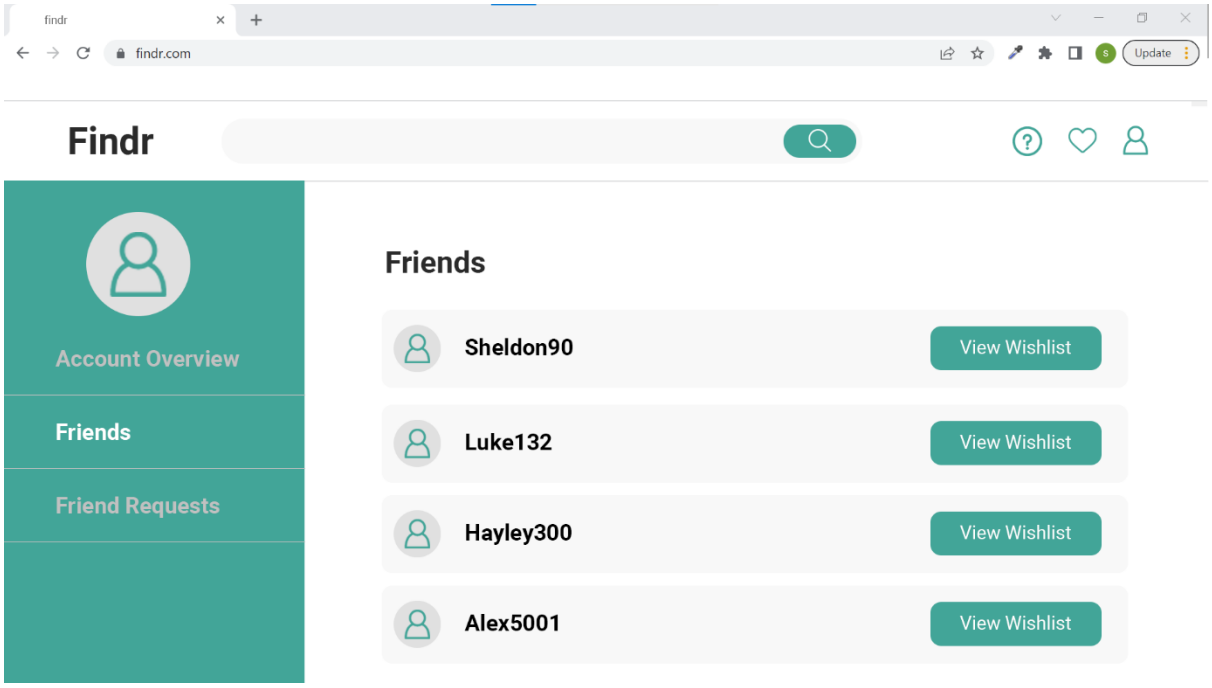
Wishlist Page



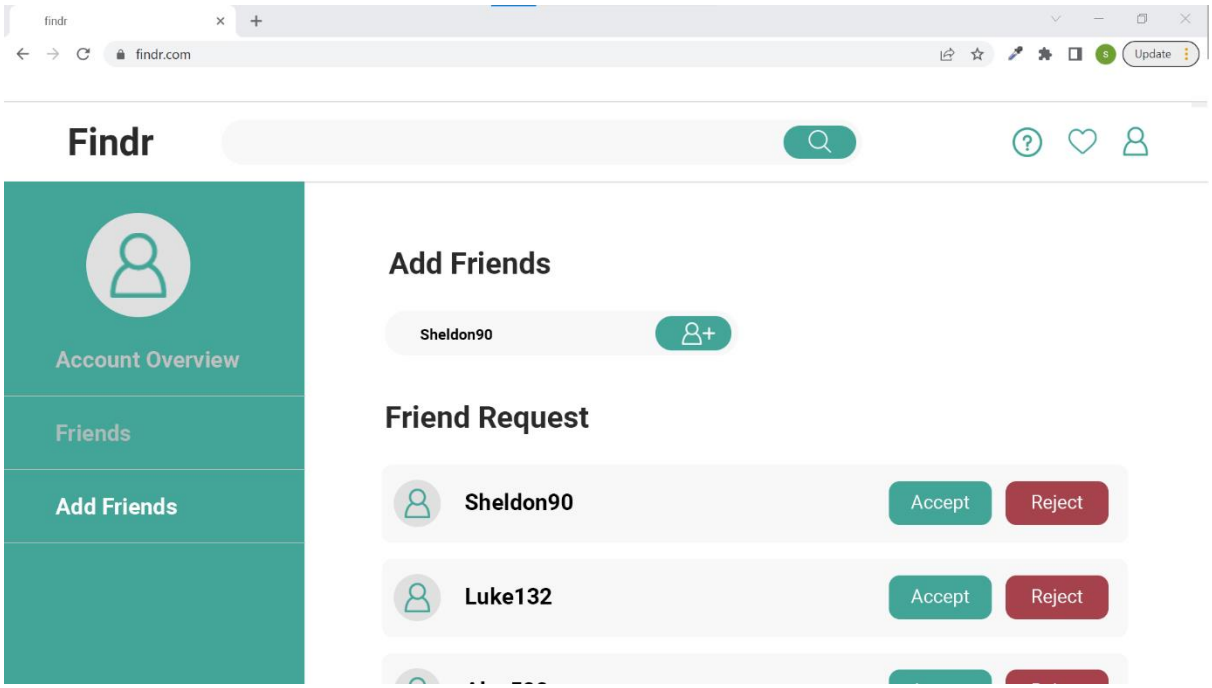
My Account Page



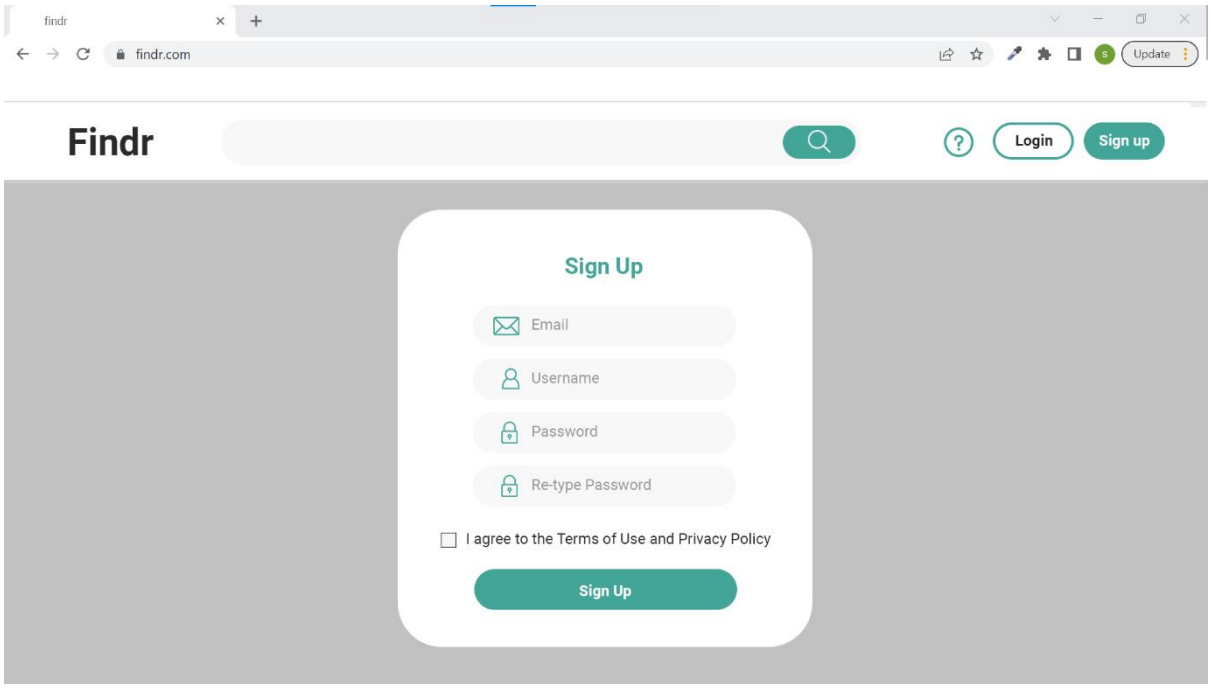
Friends Page



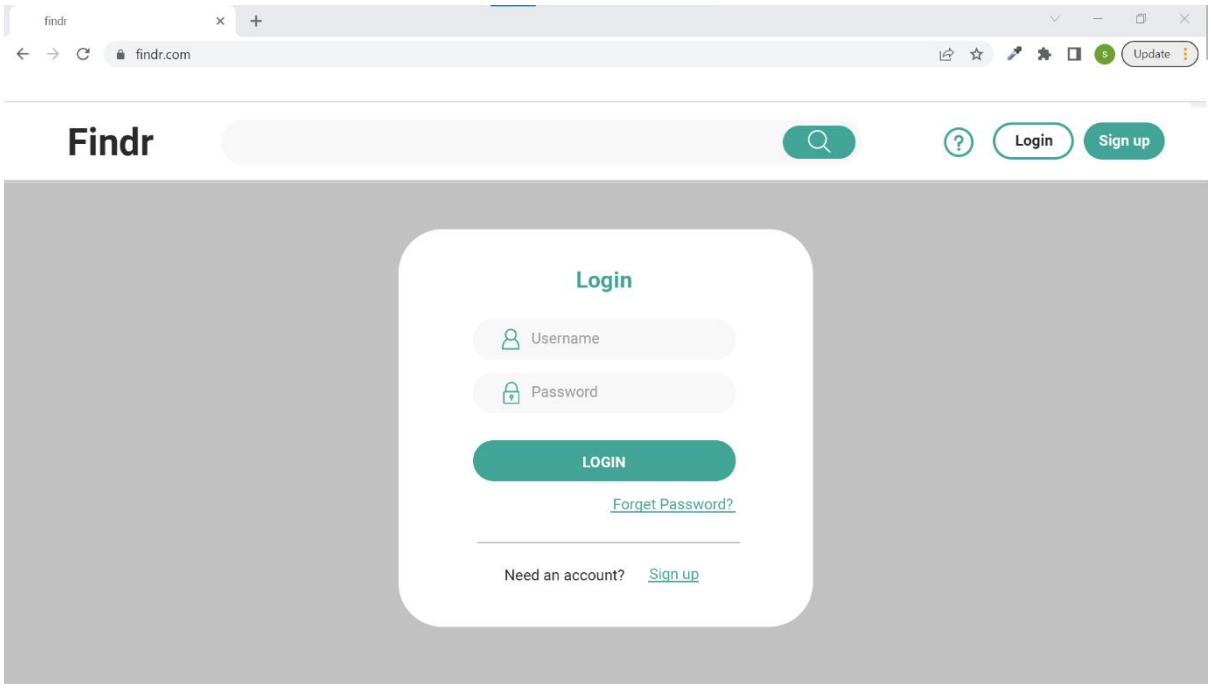
Add Friends / Friend Request Page



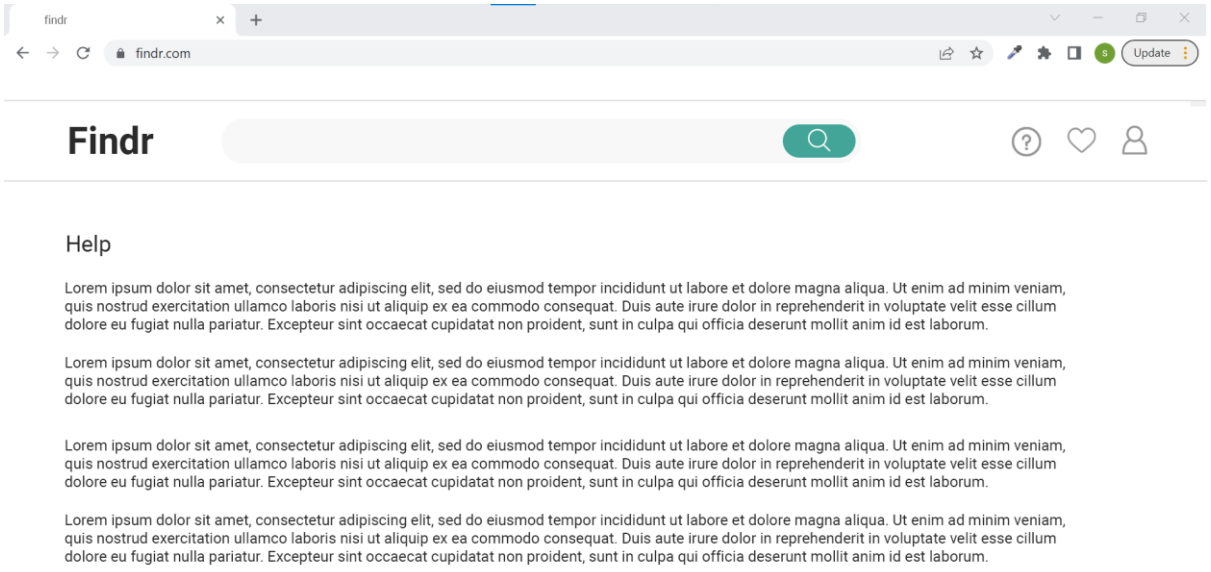
Sign Up Page



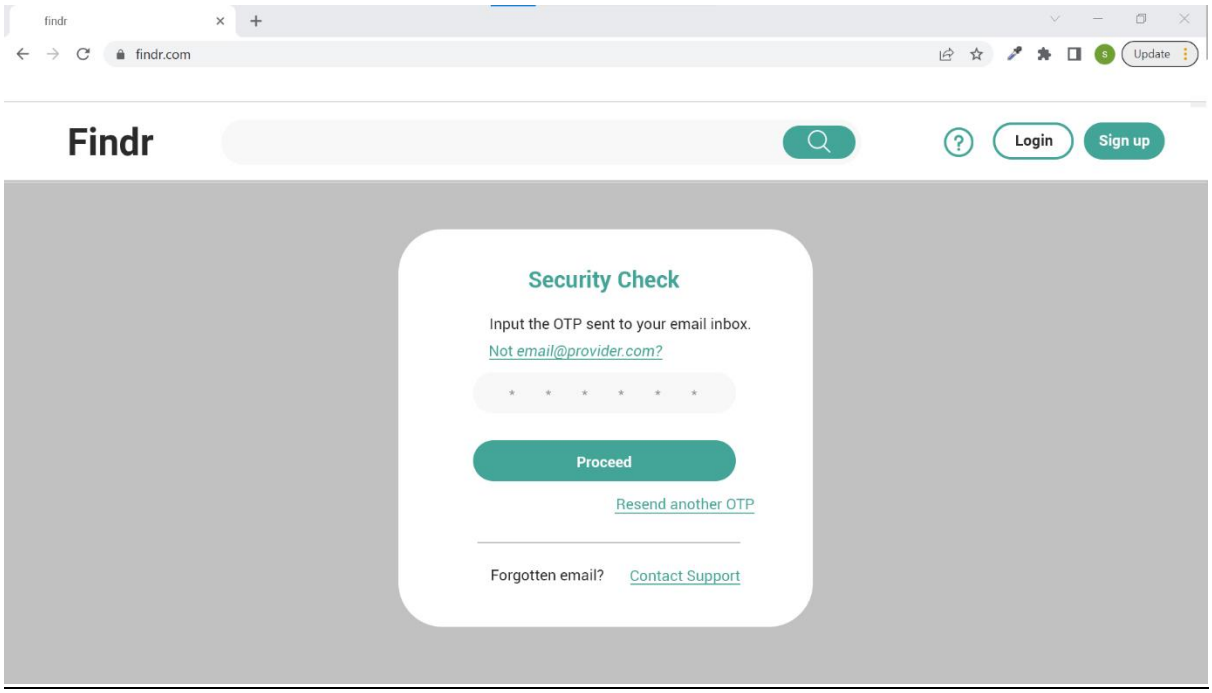
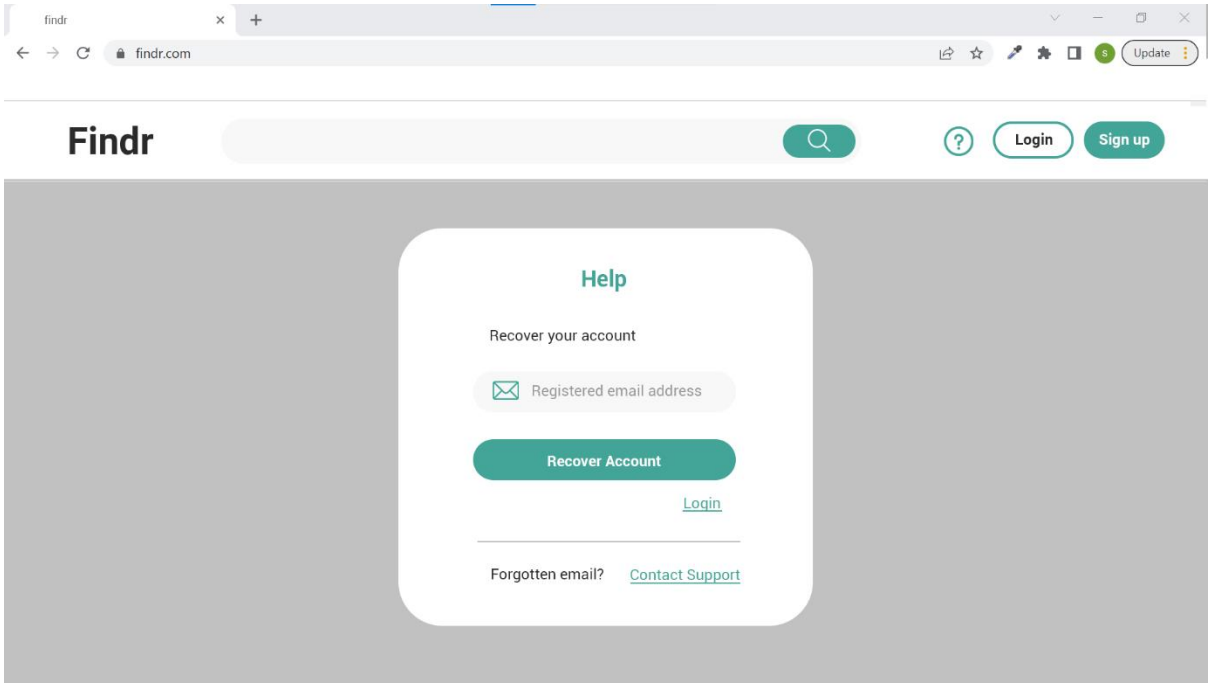
Login Page



FAQ Page



Security Check Page



Forgot Password Page