



DALHOUSIE
UNIVERSITY

Master of Applied Computer Science

CSCI 5709

Advanced Web Services

Assignment 1 (Individual)

B00924759

A1.5. User Experience and Task Flow

User Profile Management

Feature Owner – Viraj Joshi (viraj.joshi@dal.ca)

Tasks

- a. User Registration
- b. Existing User Login.
- c. View User Profile
- d. Update User Profile

a. User registration

Scenario

A student needs food at midnight but is too tired to cook a meal. The student does not want to order food from a restaurant due to a tight budget. The student wants to create an account with a website that lists restaurants in the vicinity offering leftovers from the day at no cost.

Persona – Student.

Feature – Create an account.

Need – Find restaurants offering food for free.

Context – Could not afford to order food from a restaurant.

Use Case

1. User opens the browser and types www.lastserve.ca
2. System displays the home page.
3. User clicks on 'Register as a user' button on the home page.
4. System displays the registration page, requesting first name, last name, email address and password.
5. User enters the details.

6. User clicks the 'register' button.

6.1. System displays 'First Name cannot be blank' error message.

6.1.1. User enters the first name.

6.1.2. User clicks the 'register' button.

6.2. System displays 'Email already exists' error message.

6.2.1. User enters a new email address.

6.2.2. User clicks the 'register' button.

6.3. System displays 'Password must be minimum 8 characters long' message.

6.3.1. User enters a password with 8 or more characters.

6.3.2. User clicks the 'register' button.

6.4. System clears password and confirm password field and displays 'Password and confirm password fields do not match' error message.

6.4.1. User enters matching passwords.

6.4.2. User clicks the 'register' button.

7. System registers the user.

8. System redirects the user to the login page.

Task Flow Diagram

Task flow diagram created as seen in **Figure 1** is drawn using draw.io tool [1]

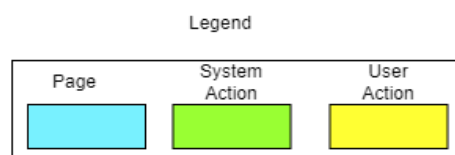
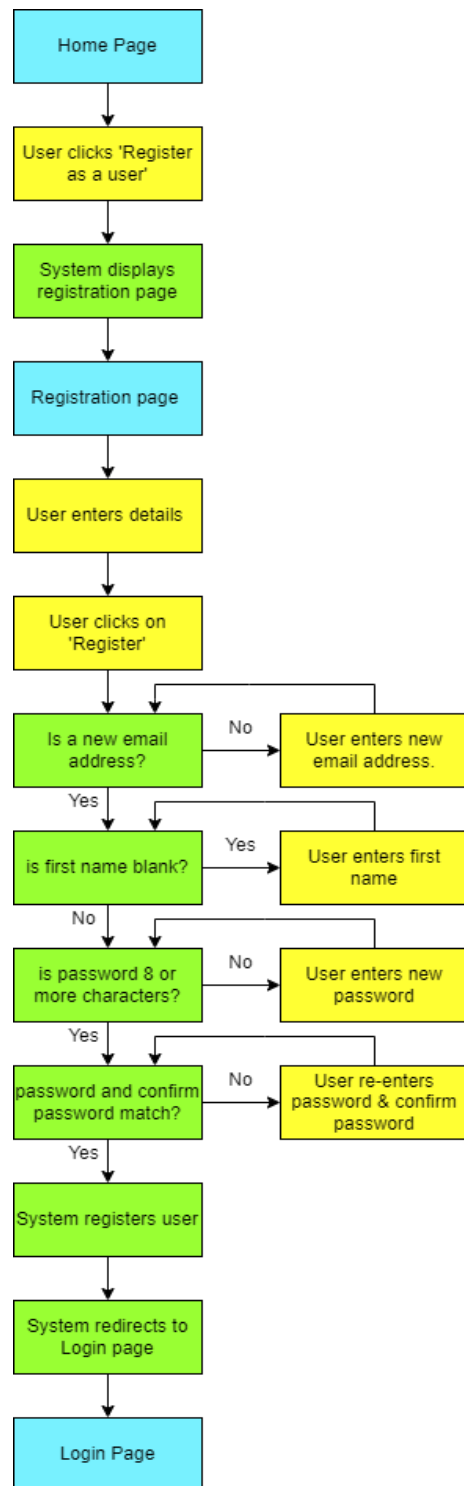


Figure 1: Task Flow diagram to create an account developed by Viraj Joshi.

b. Existing User Login

Scenario

A student is hungry and wants to check restaurants in the vicinity offering free leftover food on the LastServe website.

Persona – Student

Feature – Login account

Need – Find nearby restaurants offering free food

Context – Hungry student looking for free food

Use case

1. User opens the browser and types www.lastserve.ca
2. System displays the home page.
3. User clicks the 'User Login' button.
4. System displays the login page, requesting email and password.
5. User enters the email address and password.
6. Users clicks the 'Login' button.
 - 6.1. System displays an 'Invalid email and/or password' error message.
 - 6.1.1. System requests email and password.
 - 6.1.2. User enters details.
 - 6.1.3. User clicks 'Login' button.
 - 6.2. System displays 'Invalid email and/or password' error message.
 - 6.3. System displays 'Reset password' message.
 - 6.5. User clicks on 'Reset password' link.
 - 6.6. System displays reset password page, requesting the user to enter the registered email address.
 - 6.6.1. User enters email id.
 - 6.6.2. User clicks on 'Submit' button.
 - 6.7. System sends an authentication code to the entered email address.

- 6.7.1. System displays a field to enter code.
- 6.7.2. User enters the authentication code.
- 6.7.3. User clicks on 'Submit' button.
- 6.8. System displays the reset password page, requesting a new password and confirm password.
 - 6.8.1. User enters a new password and confirm password.
 - 6.8.2. User clicks on 'Reset Password' button.
- 6.9. System displays "New password must be minimum 8 characters" message.
 - 6.9.1. User enters a new password.
 - 6.9.2. User clicks on 'Reset Password' button.
- 6.10. System displays 'New Password' and confirm password does not match"
 - 6.10.1. User enters a new password and confirm password.
 - 6.10.2. User clicks on 'Reset Password' button.
- 7. System resets the password and displays 'Password reset successful' message.
- 8. System redirects user to login page.
- 9. System authenticates the user.
- 10. System redirects the user to the home page.

Task Flow Diagram

Task flow diagram created as seen in **Figure 2** is drawn using draw.io tool [1]

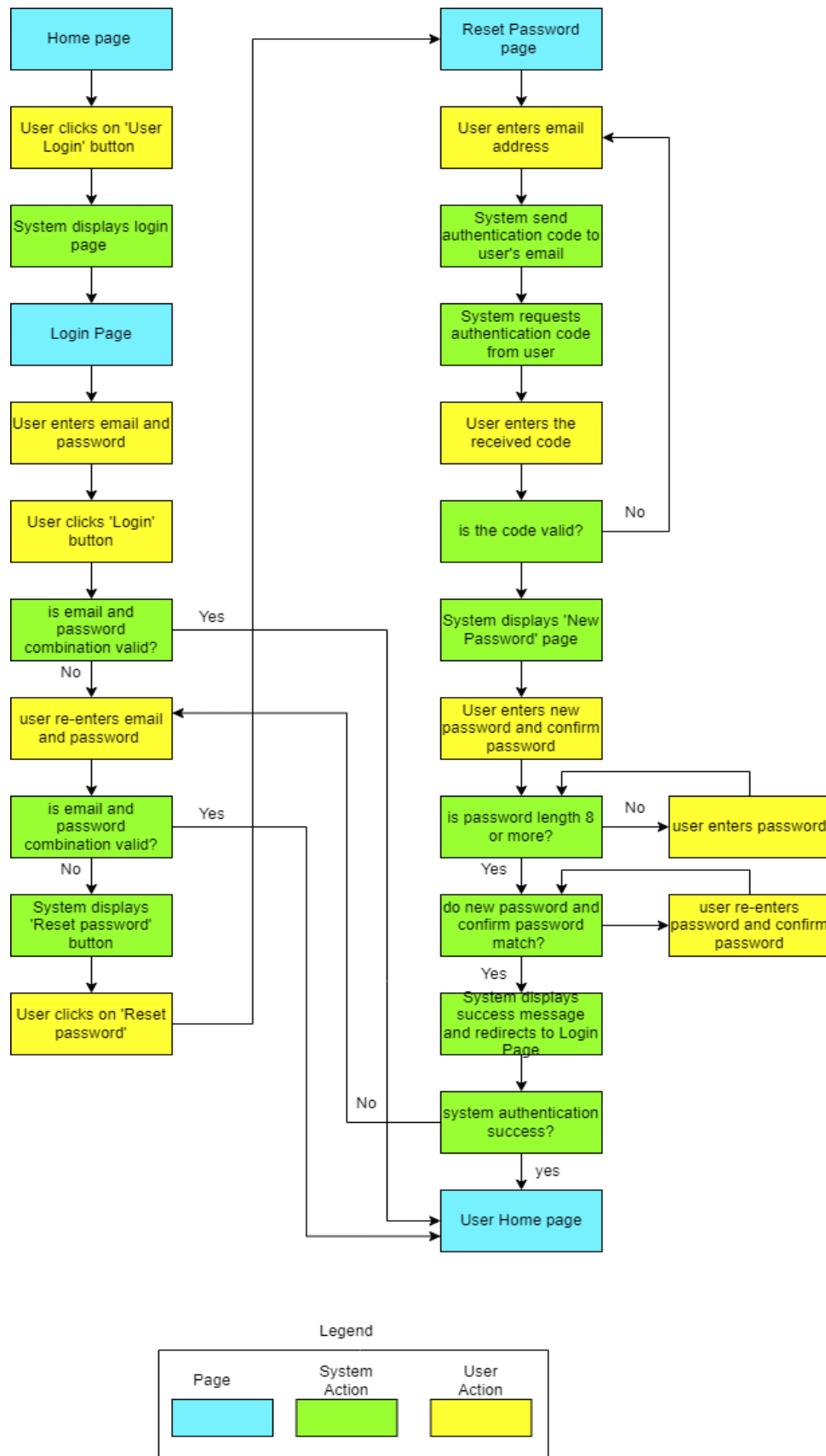


Figure 2: Task Flow Diagram for user login developed by Viraj Joshi

c. View User Profile

Scenario

The student has subscribed to get an email alert when a subscribed creates a post.
The student wants to see a list of subscribed restaurants.

Persona – Student

Feature – View user profile

Need – See all the restaurants for which an email alert has been set.

Context – The user is logged in and wants to view details.

Use case

Assumption – User is logged in.

- 1.System displays the home page.
- 2.Student clicks on the 'View Profile' icon on the home page.
- 3.System displays the view profile details page.
- 4.User scrolls to the 'Subscriptions' section on the view details page.
- 5.System displays a list of subscribed restaurants.
- 6.User clicks on the 'Back' button.
- 7.System redirects user to the home page.

Task Flow Diagram

Task flow diagram created as seen in **Figure 3** is drawn using draw.io tool [1]

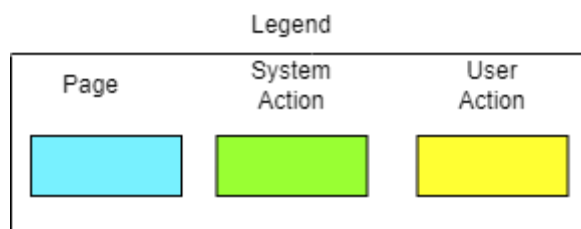
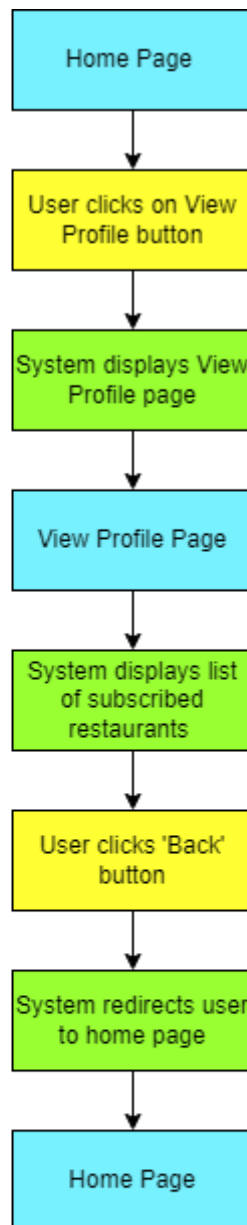


Figure 3: Task Flow Diagram for View Profile developed by Viraj Joshi

d. Edit User Profile

Scenario

While creating an account, the student mis-spelled his name. Also, the student did not enter the last name as it was an optional field. The last time he went to a restaurant, he was not allowed to pick up food as the booking details had a different first name than his ID. The student wants to change his first name and add last name to avoid any misunderstandings in the future.

Persona – Student.

Feature – Edit user profile.

Need – Update personal details.

Context – Not allowed to pick up food as names did not match.

Use case

Assumption – User is logged in.

1. System displays the home page.
2. User clicks on the 'Edit Profile' button on the home page.
3. System displays the edit profile page with first name and last name.
 - 3.1.1. User clicks on the 'Back' button.
 - 3.1.2. System redirects user to the home page.
4. User enters first name and last name.
 - 4.1. User clicks 'back' button.
 - 4.2. System displays 'No data updated' message.
5. User clicks on 'Update' button.
 - 5.1. System displays 'First Name cannot be blank' error message.
 - 5.1.1. User enters first name.
 - 5.1.2. System displays 'First Name updated' message.
6. System updates details.
7. System redirects user to the home page.

Task Flow Diagram

Task flow diagram created as seen in **Figure 4** is drawn using draw.io tool [1]

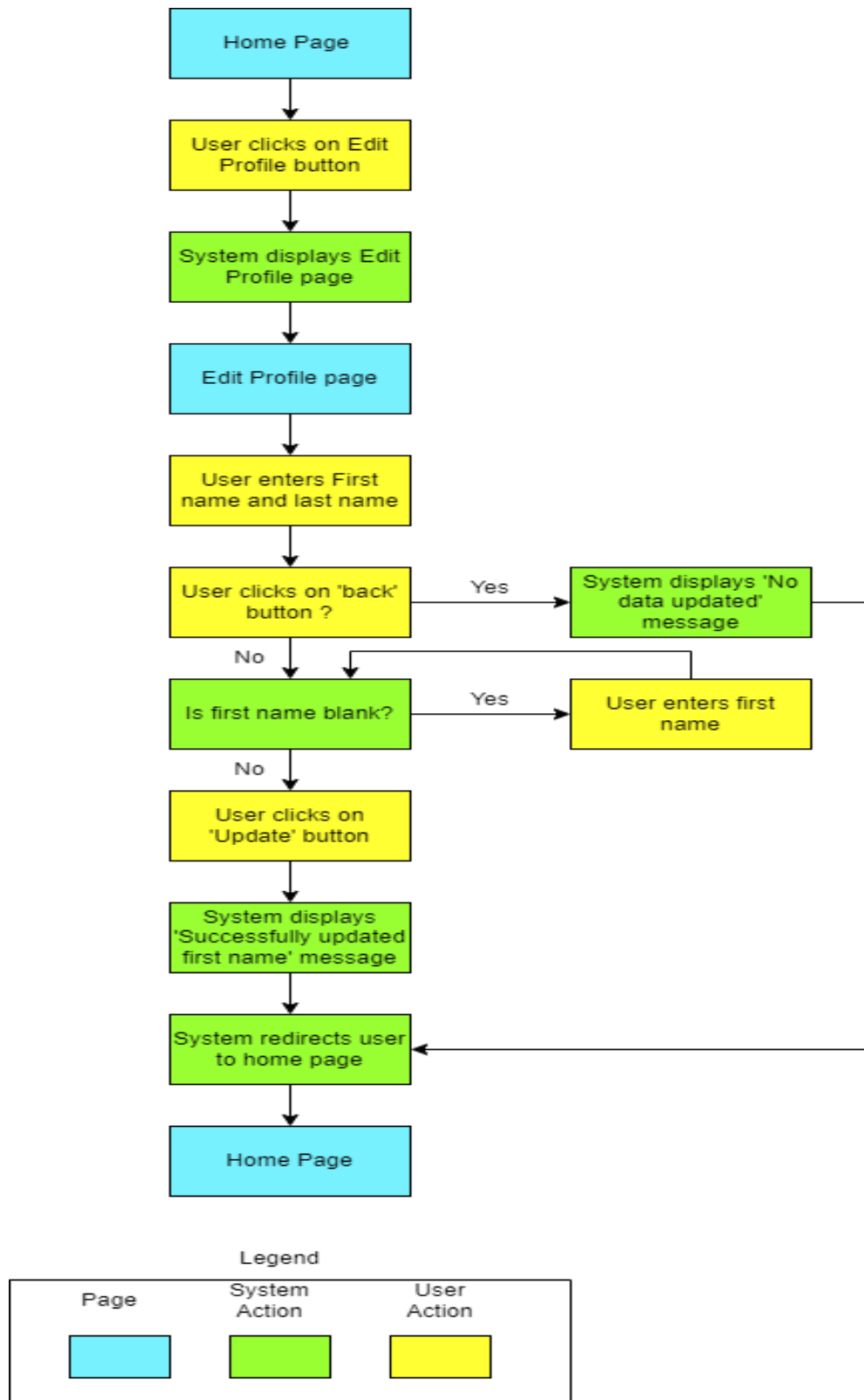
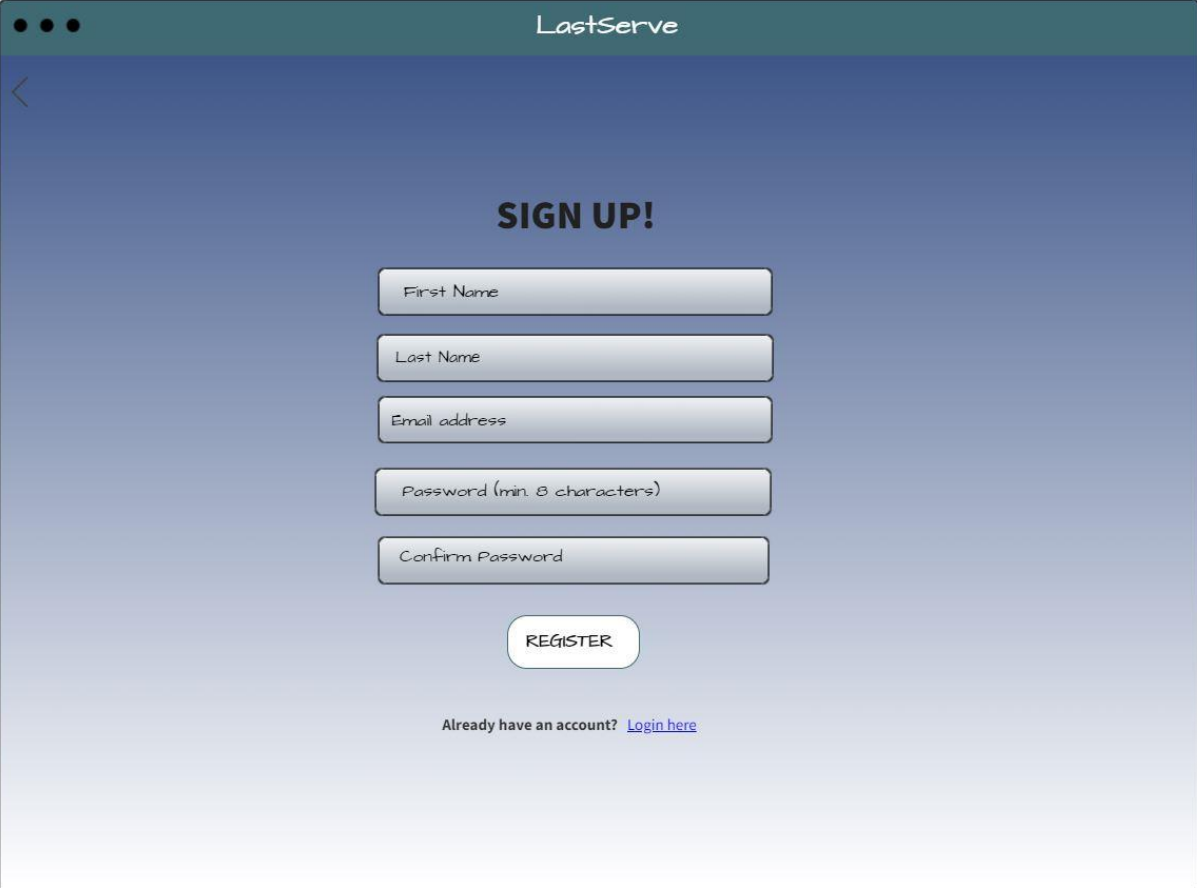


Figure 4: Task flow diagram to edit profile developed by Viraj Joshi

A1.6. Lo-Fidelity Prototype

Low-fidelity prototype of user task to 'Create an account' listed in A1.5 are as shown in **Figure 5** and **Figure 6**. Wireframes are made using online wireframing tool Mockflow [2].

Wireframes for User Registration task



The wireframe shows a web browser window with the title 'LastServe'. The page has a blue gradient background. At the top left is a back arrow icon. The main heading is 'SIGN UP!' in bold. Below it are five input fields: 'First Name', 'Last Name', 'Email address', 'Password (min. 8 characters)', and 'Confirm Password'. A 'REGISTER' button is centered below the fields. At the bottom, it says 'Already have an account? [Login here](#)'.

Figure 5: Wireframe of sign-up page developed by Viraj Joshi.

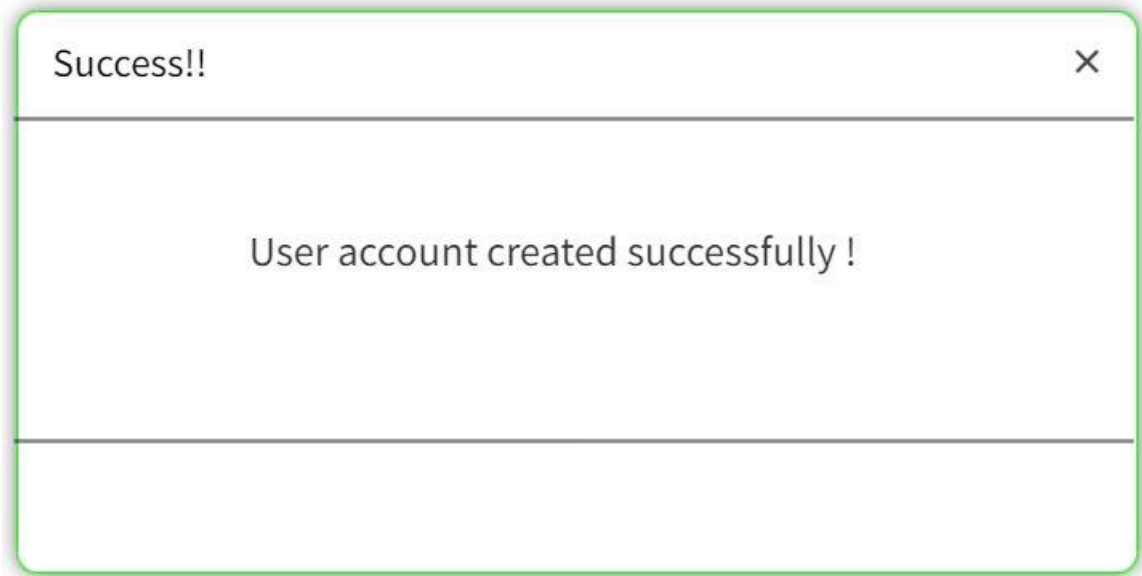


Figure 6: Wireframe for successful account creation developed by Viraj Joshi

A1.7. Semi-Functional Prototype

Semi Functional Prototype

Semi functional prototype of the sign-up page and corresponding modal is deployed on Netlify [3] at URL <https://main--merry-sorbet-809a05.netlify.app/> . Various validation messages displayed by the system and success modal can be seen from **Figure 7 – 17**

Back About Us Food Banks Volunteering

SIGN UP!

First Name*

Last Name

Email*

Password*

Confirm Password*

* Mandatory Fields

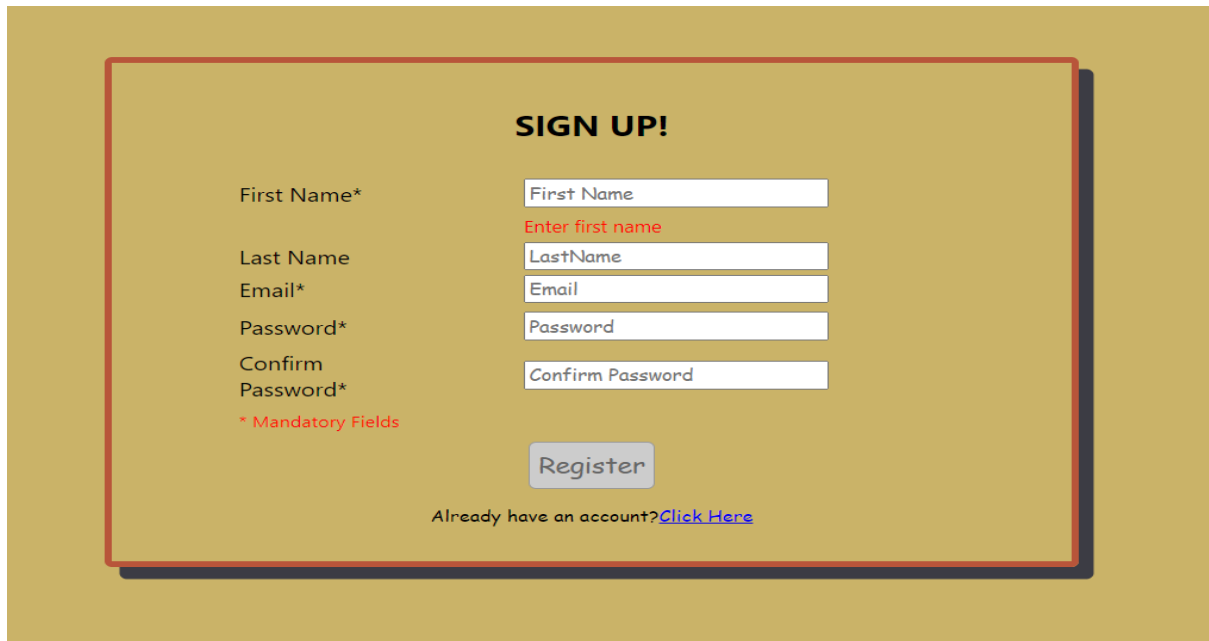
[Register](#)

Already have an account? [Click Here](#)

Figure 7: Sign Up page prototype developed by Viraj Joshi

Validations

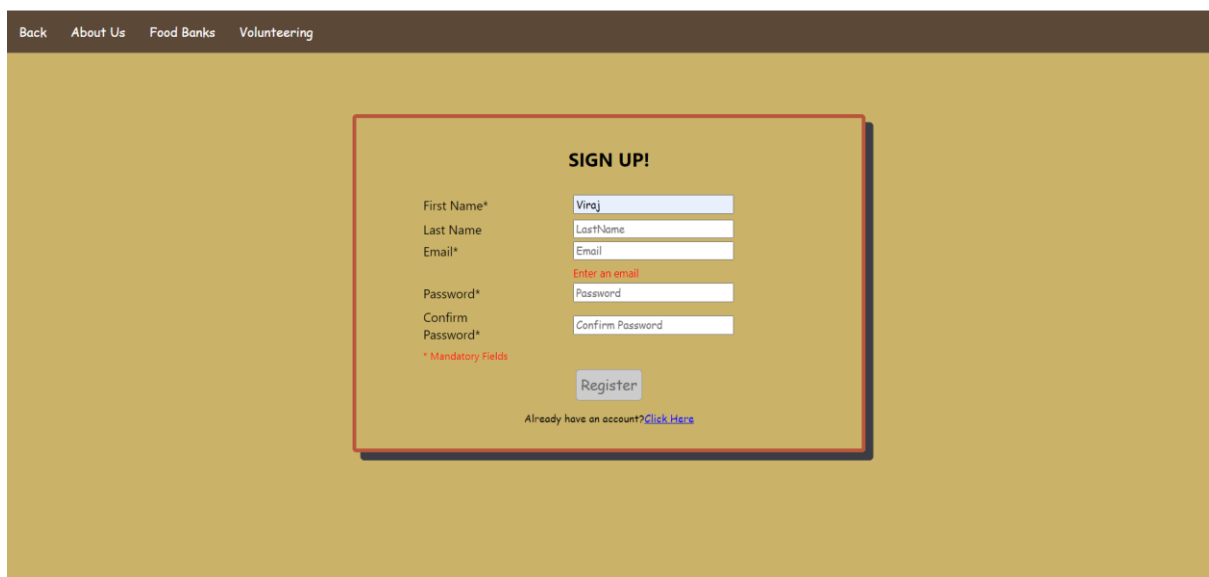
1.If a user does not enter first name, system displays a 'Enter first name' error message. The system disables the register button unless the error is resolved.



The screenshot shows a registration form titled "SIGN UP!" on a gold background. The form contains five input fields: "First Name*", "Last Name", "Email*", "Password*", and "Confirm Password*". The "First Name*" field is empty, and a red error message "Enter first name" is displayed below it. The "Register" button is disabled (grayed out). Below the button, there is a link "Already have an account? [Click Here](#)". A red asterisk legend indicates "* Mandatory Fields".

Figure 8: First Name error and disabled register button developed by Viraj Joshi

2. If a user does not enter an email address, the system displays a 'Enter an email' error message. System disables the register button.



The screenshot shows the same registration form as Figure 8, but with the "First Name" field filled with "Viraj". The "Email*" field is empty, and a red error message "Enter an email" is displayed below it. The "Register" button remains disabled. The "Last Name" field is filled with "LastName". The "Password*" and "Confirm Password*" fields are also filled. The "Already have an account? [Click Here](#)" link is present at the bottom.

Figure 9: Blank email address validation developed by Viraj Joshi

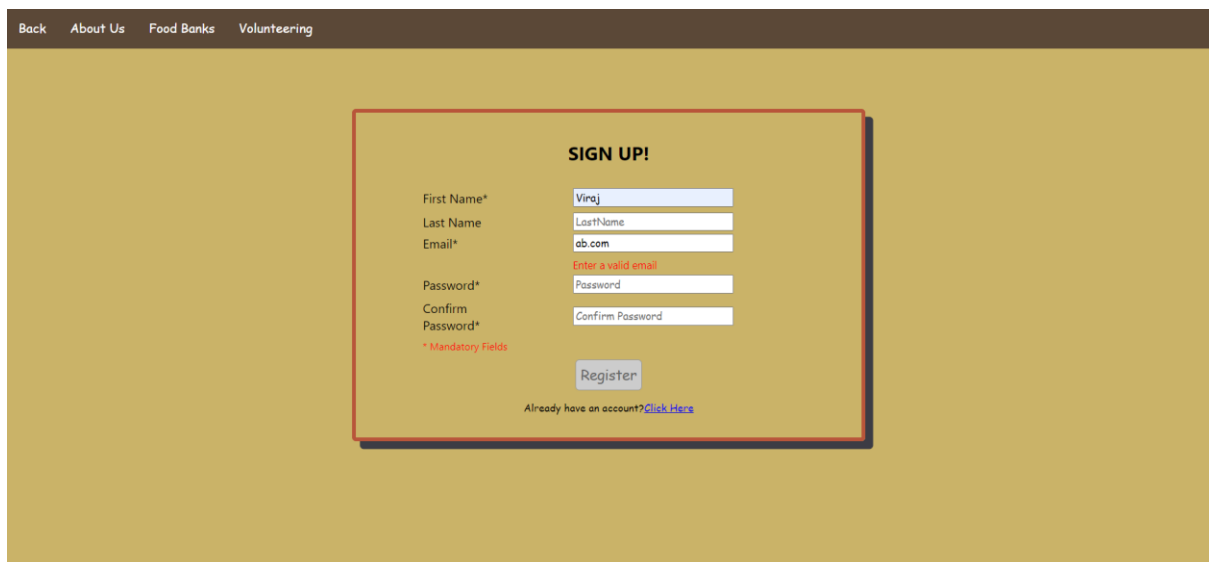
3.If a user enters an existing email address, system gives an ‘Email already exists’ error message and disables the register button. For the semi-functional prototype, a hard-coded value ‘abc@gmail.com’ is used to generate this error.



The screenshot shows a web application with a dark brown header containing navigation links: 'Back', 'About Us', 'Food Banks', and 'Volunteering'. The main content area has a light olive green background. Centered on the page is a white registration form titled 'SIGN UP!' in bold black text. The form contains the following fields and labels: 'First Name*' with the value 'Viraj', 'Last Name' with the value 'LastName', 'Email*' with the value 'abc@gmail.com', 'Password*' with the value 'Password', and 'Confirm Password*' with the value 'Confirm Password'. A red error message 'Email already exists' is displayed below the email field. A red asterisk with the text '* Mandatory Fields' is located below the password fields. A grey 'Register' button is positioned below the form fields. At the bottom of the form, there is a link: 'Already have an account? [Click Here](#)'.

Figure 10: Email already exists validation developed by Viraj Joshi

4. If a user enters an invalid email address, the system gives an ‘Enter a valid email’ error message and disables the register button.



The screenshot shows the same web application as Figure 10. The registration form titled 'SIGN UP!' is centered on the page. The fields and labels are identical to Figure 10. However, the email field now contains the value 'ab.com'. A red error message 'Enter a valid email' is displayed below the email field. The 'Register' button is still present and appears to be disabled. The link 'Already have an account? [Click Here](#)' is also visible at the bottom of the form.

Figure 11: Invalid email validation developed by Viraj Joshi

5. If a user does not enter a password, the system gives a 'Enter a password' error message and disables the register button.

The screenshot shows a web application with a navigation bar containing 'Back', 'About Us', 'Food Banks', and 'Volunteering'. The main content area has a light brown background. A white box with a red border contains the 'SIGN UP!' form. The form has the following fields: 'First Name*' (filled with 'Viraj'), 'Last Name' (filled with 'LastName'), 'Email*' (filled with 'viraj@cs.dal.ca'), 'Password*' (empty), and 'Confirm Password*' (filled with 'Confirm Password'). A red error message 'Enter password' is displayed below the 'Password*' field. A red asterisk with the text '* Mandatory Fields' is located below the 'Email*' field. A 'Register' button is at the bottom of the form. Below the button, there is a link: 'Already have an account? [Click Here](#)'.

Figure 12: Blank password validation developed by Viraj Joshi

6. If a user enters a password of length less than 8 characters, the system gives a "Password must be min. 8 characters" error message and disables the register button.

The screenshot shows the same web application as Figure 12. In this instance, the 'Password*' field contains four dots '....'. A red error message 'Password should be min. 8 characters' is displayed below the 'Password*' field. The 'Confirm Password*' field is still filled with 'Confirm Password'. The 'Register' button is now disabled (greyed out). The rest of the form and the navigation bar are identical to Figure 12.

Figure 13: Password length validation developed by Viraj Joshi

7. If a user does not enter the confirm password, system gives 'Enters a confirm password' error message and disables the register button.

The screenshot shows a web application with a navigation bar containing 'Back', 'About Us', 'Food Banks', and 'Volunteering'. The main content area has a light beige background. A central white box with a red border contains the 'SIGN UP!' form. The form fields are: 'First Name*' (filled with 'Viraj'), 'Last Name' (filled with 'LastName'), 'Email*' (filled with 'viraj@cs.dal.ca'), 'Password*' (filled with '*****'), and 'Confirm Password*' (empty). A red error message 'Enter confirm password' is displayed below the 'Confirm Password*' field. A red asterisk and the text '* Mandatory Fields' are shown below the 'Email*' field. A 'Register' button is disabled and located below the error message. At the bottom of the form box, there is a link: 'Already have an account? [Click Here](#)'.

Figure 14: Blank confirm password validation developed by Viraj Joshi

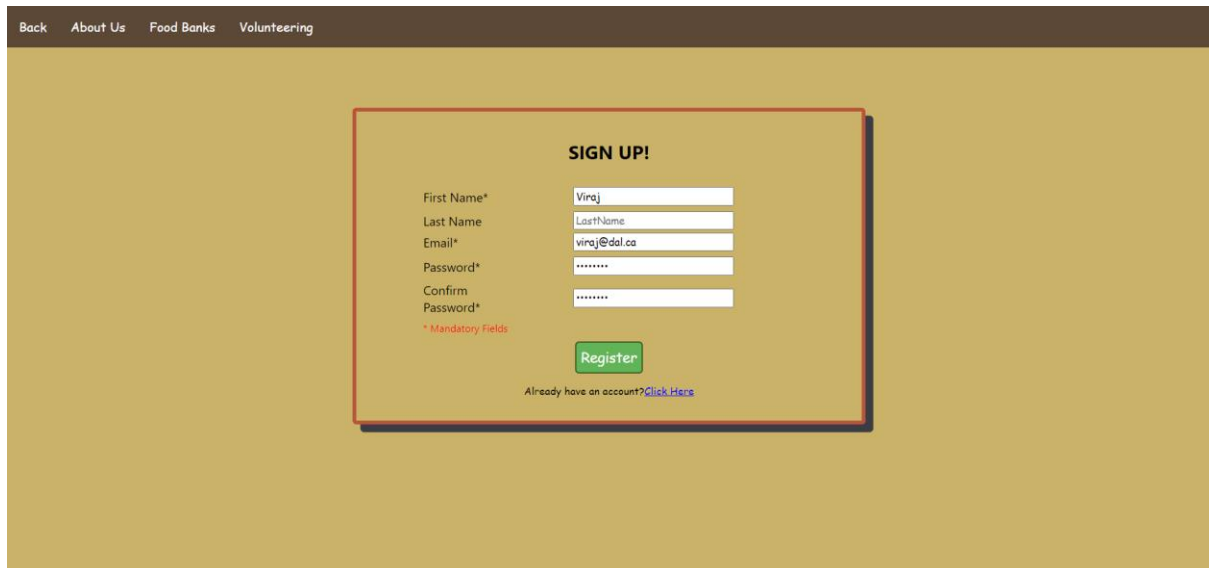
8. If the password and confirm passwords, do not match system gives a 'Password and confirm password do not match' error message and clears the password and confirm password fields for security purpose.

The screenshot shows the same web application as Figure 14. In this instance, the 'Password*' field is filled with 'Password' and the 'Confirm Password*' field is filled with 'Confirm Password'. A red error message 'Password & confirm password do not match' is displayed below the 'Confirm Password*' field. The 'Register' button remains disabled. The other fields and the bottom link are the same as in Figure 14.

Figure 15: Password and Confirm password do not match validation developed by Viraj Joshi

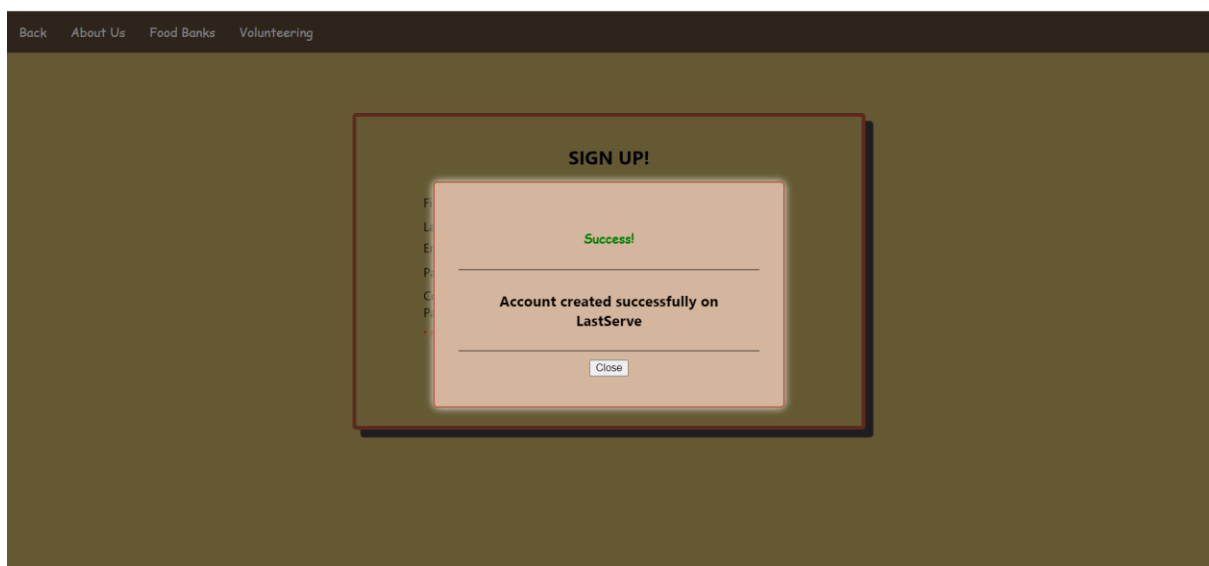
9. When the register button is disabled, along with the change in colour to give a disabled button look, the system changes the cursor to a 'Not allowed' pointer. Once the registration button is enabled, the cursor on the register button turns into to a pointer.

10. Once, all the mandatory fields are added, user can click on the register button and the system displays a success message. Once the success message is displayed, the system clears all the entered fields as a security measure. The background is darkened when the modal opens to focus on the modal contents.



The screenshot shows a web application with a dark brown header containing navigation links: 'Back', 'About Us', 'Food Banks', and 'Volunteering'. The main content area has a light tan background. A modal window titled 'SIGN UP!' is centered, featuring a red border and a drop shadow. Inside the modal, there are input fields for 'First Name*' (containing 'Viraj'), 'Last Name' (containing 'LastName'), 'Email*' (containing 'viraj@dal.ca'), 'Password*' (containing '*****'), and 'Confirm Password*' (containing '*****'). A red asterisk and the text '* Mandatory Fields' are positioned below the password fields. A green 'Register' button is located at the bottom center of the form. Below the button, there is a link: 'Already have an account? [Click Here](#)'.

Figure 16: Enabled register button when no errors



The screenshot shows the same web application header and background as Figure 16. A modal window titled 'SIGN UP!' is centered, but it now displays a success message. The modal has a light orange background and a red border. The message reads: 'Success!' in green, followed by 'Account created successfully on LastServe' in black. A 'Close' button is at the bottom center of the modal. The background of the page is darkened to focus on the modal.

Figure 17: Success message modal

On clicking close, user would be re-directed to the login page to login using the created credentials.

References

- [1] “Draw.io - free flowchart maker and diagrams online,” *Flowchart Maker & Online Diagram Software*. [Online]. Available: <https://app.diagrams.net/>. [Accessed: 06-Feb-2023].
- [2] “Wireframe tools, prototyping tools, UI mockups, UX suite, remote designing,” *MockFlow*. [Online]. Available: <https://wireframepro.mockflow.com/>. [Accessed: 06-Feb-2023].
- [3] “Develop and deploy websites and apps in record time,” *Netlify*. [Online]. Available: <https://www.netlify.com/>. [Accessed: 06-Feb-2023].