



CSC 2920 Software Engineering Project



Group #2 Members:

- Kareem J • Stephen K • Brianna T
- Chaa L • Elias R



# Presentation Outline

- Project Overview/Statement
- Use Cases
- Functional requirements
- Non-functional requirements
- GUI Design
- Sequence Diagram
- Class Diagram

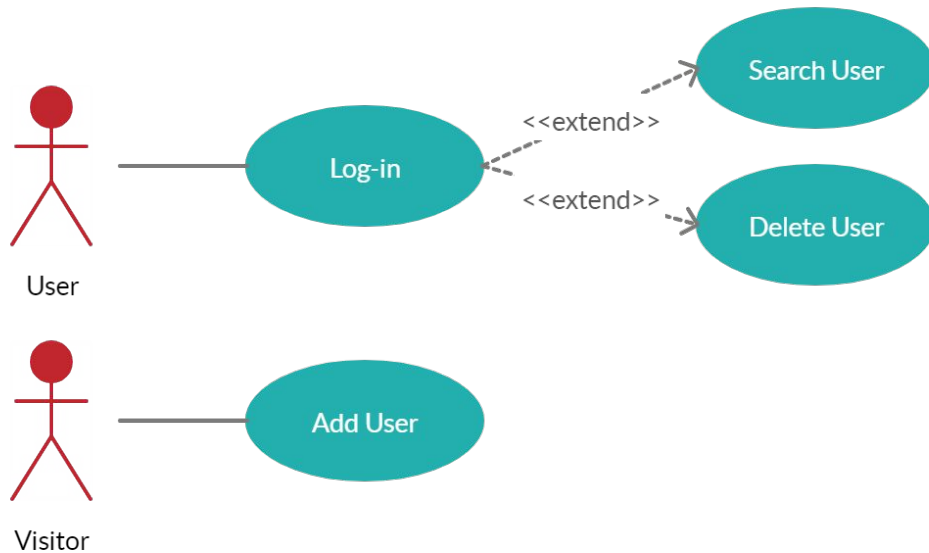


## Project Overview/Statement


We are developing a contact list that validates whether or not a contact has been submitted using the correct format. We will also provide the same parameters concerning email and phone number format.

Contact lists need a way to validate that an email and phone number have been given in the correct format, because the submitted data must be able to work successfully with the client's application.

# Use Cases



# Use Cases

<b>Use case name:</b>	Add User.
<b>Participating actors</b>	Initiated by the user
<b>Flow of events</b>	<ol style="list-style-type: none"><li>1) The user inputs the required information in all text fields.</li><li>2) The system validates the required information is in the correct format.</li><li>3) The system displays text notifying the user that text is an invalid format.</li><li>4) The user's information is added to a.CSV file.</li></ol>
<b>Entry conditions</b>	The user must be logged in and the add button must be clicked.
<b>Exit conditions</b>	The user has been notified by the system that all conditions have been met. 

# Use Cases

<b>Use case name</b>	Delete user
<b>Participating actors</b>	Initiated by the user
<b>Flow of events</b>	<ol style="list-style-type: none"><li>1) The user searches for a user by inserting his/her name.</li><li>2) The system finds the user on a.CSV file.</li><li>3) The user clicks the delete button.</li><li>4) The system deletes the user.</li></ol>
<b>Entry conditions</b>	The user must be logged in, the user must input a valid name, then click the search button, then the delete button.
<b>Exit conditions</b>	The system must notify the admin to know that the user has been successfully deleted.

# Use Cases

<b>Use case name</b>	Search for User
<b>Participating actors</b>	Initiated by the user
<b>Flow of events</b>	<ol style="list-style-type: none"><li>1) The user gives input for the user's first and last name.</li><li>2) The user clicks the search button.</li><li>3) The system finds the user on.CSV file.</li><li>4) The system notifies the user that the user has or has not been found.</li></ol>
<b>Entry conditions</b>	The user must be logged in and the search button must be clicked.
<b>Exit conditions</b>	The system notifies the user that the user has been found.



# Functional requirements

## Create-Contact

**Input:** Details of the new contact information.

**Action:** Displays a form to be filled in by the owner and records contact information like name, address, email id, phone no., and etc.

**Output:** Creates a new contact bar with the given information and adds it to the contact list.





# Functional requirements

## Search-Contact

**Input:** Name of the contact searched for.

**Action:** Displays a list of possible contacts while the user types the search keywords in the search bar.

**Output:** Displays existing details of the selected contact.

# Functional requirements

## Delete-Contact

### Input:

- Name of the contact searched for.
- Details of the searched contact information obtained from the result set.

**Action:** Prompts the user to confirm if he/she wants to permanently delete that contact.

**Output:** Updates the contact bar with the newly updated information.



# Non-functional requirements

## Performance Requirements

- The system shall accommodate 1 terabyte without fault.
- Responses to view information shall take no longer than 5 seconds to appear on the screen.

## Safety Requirements

- System use shall not cause any harm to human users.

# Non-functional requirements

## Security Requirements

- System will validate passwords with boolean method.
- Users should not be able to log in without having or creating an account.

## Error Handling

- The Contact List shall handle all expected and unexpected errors in ways that prevent loss in information and a long downtime period. For example, if the user has issues logging in, creating an account, searching user, deleting from the contact list, or logging off.



# GUI Design



Username

Password

Log In

## Create Account

Username

Password

Verify Password

Create

## Features Available



Save New Contacts



Add Email Address



Easy Access



# GUI Design



+ Create Contact

✕ Delete Contact

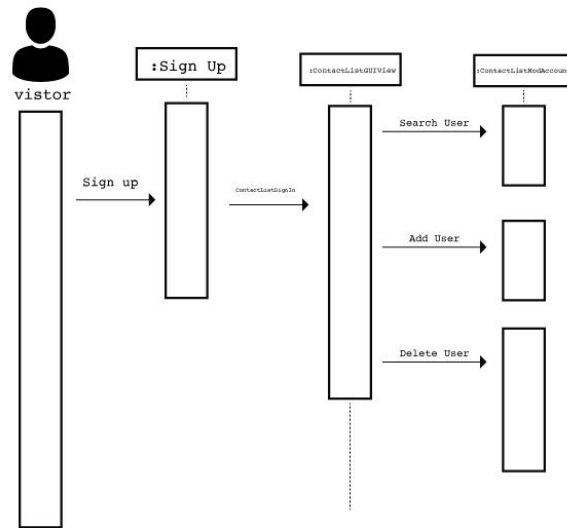
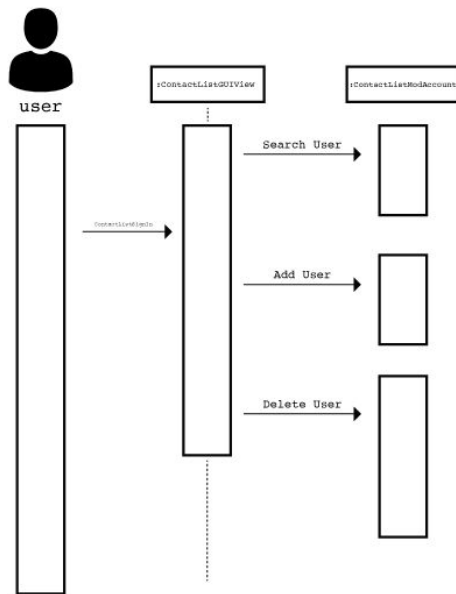
Name

Brave Mail

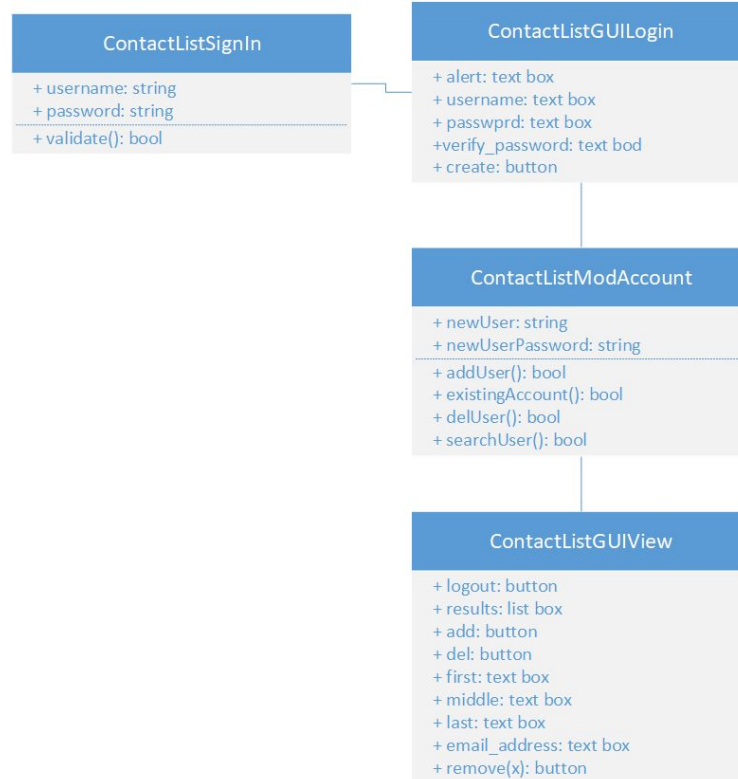
Search

Log off

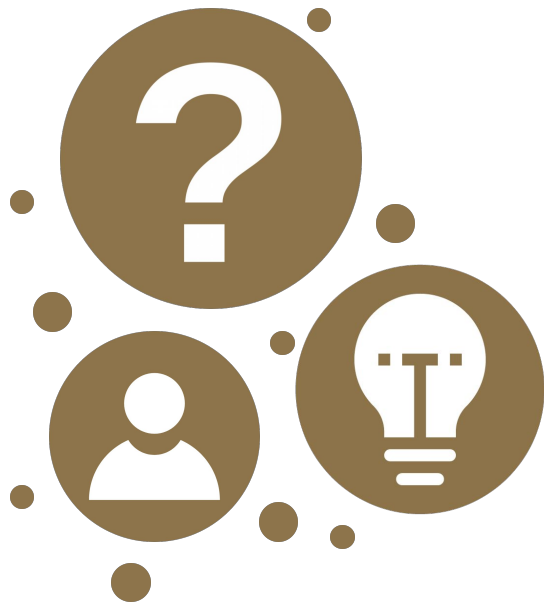
# Sequence Diagram



# Class Diagram







Questions?  
Comments?  
Concerns?