Software Design Documentation

## By: Darcy Merilan, Bryan Welton, Conner Wadsworth, Angela Slinker, Andrew Swazye

# Front End:

User Account Management

The system must ensure security by implementing robust authorization, authentication, and access control mechanisms to protect user data and privacy.

**View:** The user interface elements for account registration, login, profile management, and account settings. This includes forms for input, pages for viewing and editing user profiles, and menus for accessing account settings. The View listens for user actions (e.g., button clicks, form submissions) and communicates these actions to the Controller.

**Create**

* **Registration Form**
  + **Description**: Allows new users to create an account by entering their details, such as username, email, and password.
  + **Elements**: Text inputs for username, email, password, and possibly captcha or security questions for bot protection and user verification. A submit button to send the registration request.
  + **Actions**: When the user fills out the form and clicks the submit button, the View sends this information to the Controller to process the account creation.

**Read**

* **Login Form**
  + **Description**: Enables users to access their account by providing their credentials.
  + **Elements**: Text inputs for username/email and password, with a submit button for logging in. Links for "Forgot password" or "Register" for navigation to other views.
  + **Actions**: Submitting the form sends the credentials to the Controller, which verifies them against the Model data.
* **Profile Page**
  + **Description**: Displays the user's profile information, including username, email, and any additional details like bio or profile picture.
  + **Elements**: Text and possibly images showing the user's information. May include an "Edit Profile" button to navigate to the Update view.
  + **Actions**: Viewing the profile does not directly interact with the Controller but may involve fetching updated information from the Model through the Controller.

**Update**

* **Profile Edit Form**
  + **Description**: Allows users to update their profile information, such as email, bio, and profile picture.
  + **Elements**: Pre-filled text inputs and file upload inputs with the user's current information, and a submit button to apply changes.
  + **Actions**: Submitting the updated information sends it to the Controller, which processes the update requests and modifies the Model accordingly.
* **Change Password Form**
  + **Description**: Provides a way for users to change their password.
  + **Elements**: Text inputs for the old password, new password, and new password confirmation. A submit button to apply the password change.
  + **Actions**: On submission, the View sends the password details to the Controller, which handles the password update process.

**Delete**

* **Account Deletion Option**
  + **Description**: Offers users the option to delete their account.
  + **Elements**: A "Delete Account" button, often accompanied by a confirmation step (e.g., entering a password or answering a security question) to prevent accidental deletions.
  + **Actions**: Confirming account deletion sends a request to the Controller to remove the user's account from the Model.

---------------------------------------------------------------------------

Community Posts

Users can create posts to seek or offer rides, promoting transportation assistance within the community. Posts must include attributes like start and stop destinations, description, category, timestamp, rating, and available passenger seats to enable effective communication.

**View:** The components and pages that display community posts, comments, and the interface for creating and editing posts. This includes the post feed, individual post views, and forms for post creation and editing. The View updates in response to user interactions and data changes, informing the Controller of user actions like post submissions or comment additions.

**Create:**

* **Create Post Form** 
  + **Description:** Allows users to create new posts by entering details such as start and stop destinations, description, category, timestamp, rating, and available passenger seats.
  + **Elements:** Input fields, dropdowns, or other UI elements for each post attribute. A submit button to create the post.
  + **Actions:** When the user submits the create post form, the view sends the post details to the controller, which processes the request and adds the new post to the model.

**Read:**

* **Post Feed**
  + **Description:** Displays community posts in a feed format, allowing users to easily browse through available options.
  + **Elements**: A list or grid displaying posts with summarized information. Each post may include start and stop destinations, description, category, timestamp, rating, and available passenger seats.
  + Actions: Viewing the post feed involves fetching and displaying data from the model. Users can click on individual posts to view more details.

**Update:**

* **Edit Post Form**
  + **Description**: Enables post creators to update their posts by modifying details such as start and stop destinations, description, category, timestamp, rating, and available passenger seats.
  + **Elements**: Pre-filled input fields and dropdowns with the current post details. A submit button to apply the changes.
  + **Actions**: Submitting the updated information sends it to the controller, which processes the update requests and modifies the model to reflect the changes.

**Delete:**

* **Delete Post Option**
  + **Description**: Provides users with the option to delete their own posts, preventing accidental deletions through a confirmation step.
  + **Elements**: A "Delete Post" button, accompanied by a confirmation modal or dialog. Users may need to confirm the deletion by entering a password or answering a security question.
  + **Actions**: Confirming post deletion sends a request to the controller to remove the post from the model. The confirmation step helps prevent accidental deletions.

---------------------------------------------------------

**Search Functionality**

The system must provide users with the ability to search for specific posts based on criteria such as destination, category, and timestamp, facilitating efficient discovery of relevant information.

**View:** The search functionality view is responsible for providing users with a user-friendly interface to perform searches. This includes input fields for entering search criteria, search result display, and any additional features that enhance the search experience. The view communicates user input to the controller, which then processes the search query and updates the view with the relevant results.

**Create:**

* **Search Form**
  + **Description**: Allows users to input search criteria such as destination, category, and timestamp.
  + **Elements**: Text inputs, dropdowns, or other input fields for specifying search criteria. A submit button to initiate the search.
  + **Actions**: When the user submits the search form, the view sends the search criteria to the controller, which processes the request and queries the model for matching posts.

**Read:**

* **Search Results**
  + **Description**: Displays the search results based on the specified criteria.
  + **Elements**: A list or grid displaying relevant posts that match the search criteria. Each result should provide a summary of the post.
  + **Actions**: Viewing search results doesn't directly interact with the controller but involves fetching and displaying data from the model based on the search criteria.

**Update:**

* **Refine Search Options**
  + **Description**: Allows users to modify their search criteria without starting a new search.
  + **Elements**: Additional controls or filters to refine the ongoing search, such as adjusting timestamps, changing categories, or adding/removing criteria.
  + **Actions**: When the user modifies the search options, the view sends the updated criteria to the controller, which reprocesses the search and updates the results.

**Delete:**

* **Clear Search Form**
  + **Description**: Enables users to clear the search form and start a new search.
  + **Elements**: A "Clear" or "Reset" button to clear all entered search criteria.
  + **Actions**: Clicking the "Clear" button sends a request to the controller to reset the search criteria, clearing the form and potentially resetting the displayed results.