###ImageUploadUI

####Functions:

- `ImageUploadUI()`
- `initializeUI()`
- `uploadAction()`
- `getNextImageId()`
- `saveImageInfo()`
- `getFileExtension()`
- `saveBioAction()`
- `createHeaderPanel()`
- `readUsername()`
- `createNavigationPanel()`
- `createIconButton()`
- `openProfileUI`
- `notificationsUI()`
- `openHomeUI`
- `exploreUI()`

####Functionalities of the Functions:

- `ImageUploadUI Constructor:` Initializes the UI for uploading images.
  - Refactor and Improvement: Extract repetitive layout initialization code into separate methods to improve readability and maintainability.
- `initializeUI():` Sets up the main content panel with image preview, bio text area, and buttons.
  - Refactor and Improvement: Break down the method into smaller, more focused methods for setting up each component, improving readability and allowing for easier modification of individual components.
- `uploadAction(ActionEvent event):` Handles the action of uploading an image, including file selection, saving, and displaying the image preview.
  - Refactor and Improvement: Separate file handling logic into helper methods for better organization and potential reuse. Use try-with-resources for file operations to ensure resources are properly closed.
- `getNextImageId(String username):` Retrieves the next available image ID for a given username.
  - Refactor and Improvement: Extract the logic for finding the next image ID into a separate method to improve readability and maintainability. Ensure proper error handling for file operations.
- `saveImageInfo(String imageId, String username, String bio):` Saves image information (ID, username, bio) to a text file.

  - Refactor and Improvement: Break down the method into smaller, more focused methods for file handling and timestamp generation. Use try-with-resources for file operations.

- `getFileExtension(File file):` Retrieves the extension of a given file.
  - Refactor and Improvement: Simplify the method by using java.nio.file.Path instead of manipulating file names directly.

- `saveBioAction(ActionEvent event):` Handles the action of saving the bio text.
  - Refactor and Improvement: Consider storing bio text in a more permanent storage solution such as a file or database instead of showing a message dialog.

- `createHeaderPanel():` Creates and returns the header panel for the UI.
  - Refactor and Improvement: Consider making the header panel more generic by allowing customization of the header text and icon. Extract icon creation and button styling into helper methods.

- `readUsername():` Reads the username from a file.
  - Refactor and Improvement: Ensure proper error handling for file operations. Consider using a more robust data storage solution like a database for user information.

- `createNavigationPanel():` Creates and returns the navigation panel with icons for home, explore, add, notification, and profile.
  - Refactor and Improvement: Extract icon creation and button styling into helper methods. Consider making the method more flexible by allowing customization of icons and actions.

- `createIconButton(String iconPath, String buttonType):` Creates and returns a button with the specified icon and action.
  - Refactor and Improvement: Utilize enums for button types instead of strings to improve readability and avoid potential typos. Consider extracting action listeners into separate methods for better organization.

- `openProfileUI(), notificationsUI(), openHomeUI(), exploreUI():` Methods to open different UI frames.
  - Refactor and Improvement: Consider consolidating UI navigation logic into a single method for better control and consistency. Use enums or constants to represent different UI frames instead of hardcoding strings.