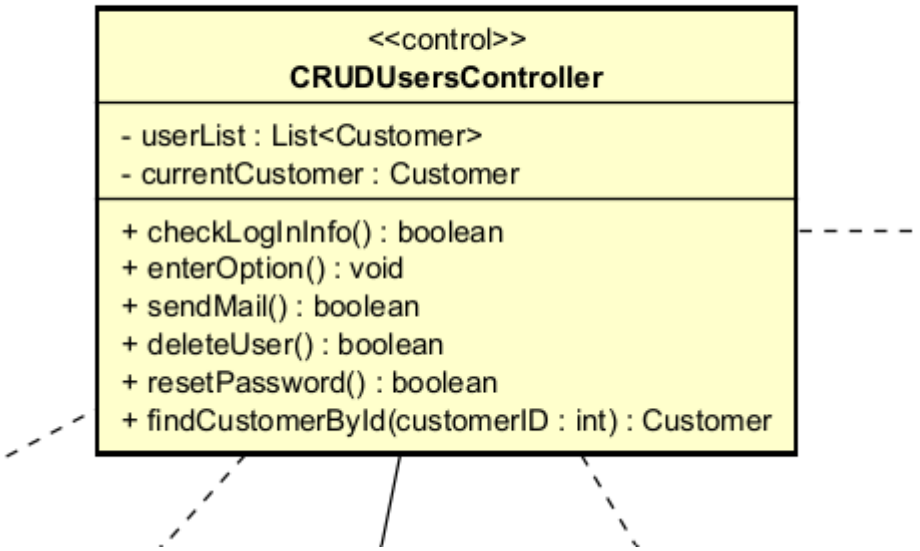
Fullname: Doan Thi Thu Quyen
Student ID: 20226063

1. Design for class “CRUD Users”



- Attribute design

#	Name	Data type	Default value	Description
1	userList	List<Customer>	Null	List of customer for admin to modify/view
2	currentCustomer	Customer	Null	Current customer (when process a request or update)

- Operation design

#	Name	Return type	Description (purpose)
1	checkLogInInfo	boolean	Check whether the login information is valid or not
2	enterOption	void	Allow admin to enter option to modify users and process the chosen one

3	sendMail	boolean	Send mail to notify user, return true if successfully sent, else return fail
4	deleteUser	boolean	Delete user, return true if successfully deleted, else return fail
5	resetPassword	boolean	Reset user's password, return true if successfully reset, else return fail
6	findCustomerByID	Customer	Filter customer by ID

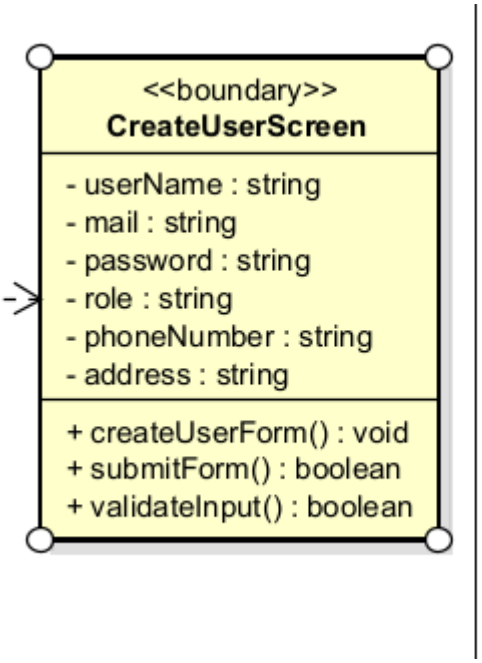
- Parameter

  Null

- Exception
  - UserNotFoundException: When the user is not found
  - InvalidUserDataException: When the user data is invalid
  - EmailSendingException: When sending the email fails
- State
- Method
  - checkLogInInfo() : boolean
    - Purpose: Verifies user login information.
    - Algorithm: Checks credentials against the userList.
    - Related Attributes: Uses userList, currentCustomer.
  - enterOption() : void
    - Purpose: Handles user input for various operations.
    - Algorithm: Prompts the user and calls appropriate methods based on input.
  - sendMail() : boolean
    - Purpose: Sends an email to the current customer.
    - Algorithm: Uses email APIs to send messages.
    - Related Attributes: Uses currentCustomer.
  - deleteUser() : boolean
    - Purpose: Removes a customer from the system.
    - Algorithm: Searches userList and deletes the specified user.
    - Related Attributes: Uses userList.
  - resetPassword() : boolean
    - Purpose: Resets the password for a customer.
    - Algorithm: Validates the user and updates the password.

        - Related Attributes: Uses currentCustomer.
    - findCustomerById(customerID : int) : Customer
        - Purpose: Retrieves a customer by their ID.
        - Algorithm: Searches the userList for the matching ID.
        - Related Attributes: Uses userList.

2. Design for class “Create User Screen”



- Attribute design

#	Name	Data type	Default value	Description
1	userName	string	Null	Username
2	mail	string	Null	User’s email
3	password	string	Null	User’s password
4	phoneNumber	string	Null	User’s phone number
5	address	string	Null	User’s address

- Operation design

#	Name	Return type	Description (purpose)
1	createUserFor	void	Create a form to enter input

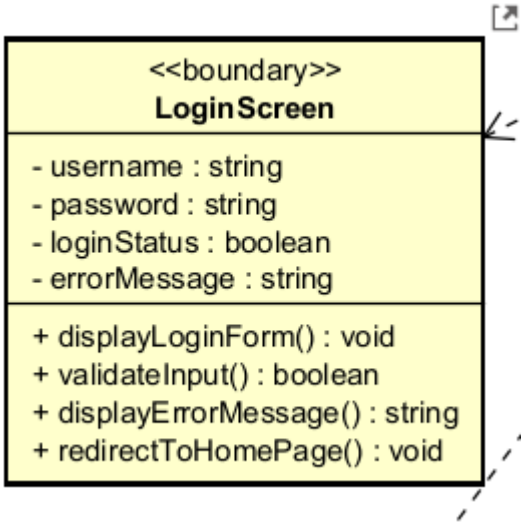
	m		
2	submitForm	boolean	State to know if the form is submitted and create user
3	validateInput	boolean	Check whether the all input is valid

- Parameter

  Null

- Exception
  - InvalidInputException: When the input information is invalid.
  - SubmissionFailedException: When the form submission fails (eg: network issues, server errors, etc.).
- State
- Method
  - createUserForm() : void
    - Purpose: Initializes and displays the user creation form.
    - Algorithm: Sets up input fields for user data (name, email, password, role, phone number, address).
  - submitForm() : boolean
    - Purpose: Submits the form data to the server or database.
    - Algorithm: Validates input and calls backend API for user creation.
    - Related Methods: Uses validateInput() before submission.
  - validateInput() : boolean
    - Purpose: Ensures that all input fields are correctly filled.
    - Algorithm: Checks for non-empty fields, valid email format, and password requirements.
    - Related Attributes: Uses userName, mail, password, role, phoneNumber, address.

3. Design for class “LogInScreen”



- Attribute design

#	Name	Data type	Default value	Description
1	userName	string	Null	Username
2	mail	string	Null	User’s email
3	password	string	Null	User’s password
4	phoneNumber	string	Null	User’s phone number
5	address	string	Null	User’s address

- Operation design

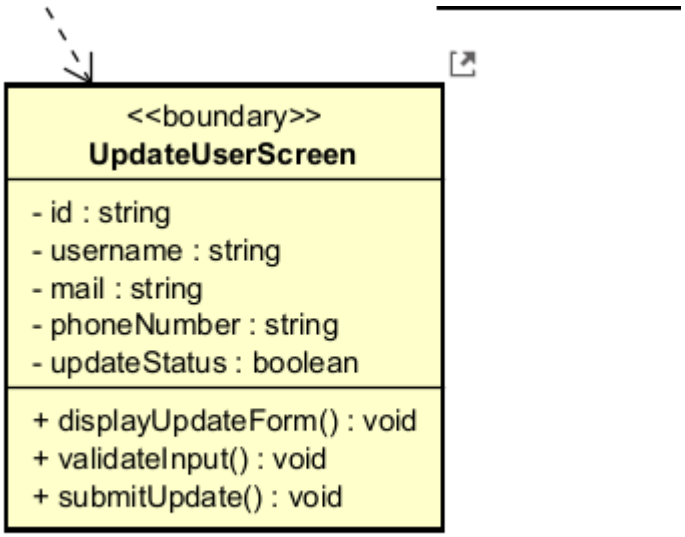
#	Name	Return type	Description (purpose)
1	displayLoginForm	void	Display the login form
2	validateInput	boolean	Check whether all the input is valid for login
3	displayErrorMessage	string	Show error if there are any
4	redirectToHomePage	void	Redirect to Home Screen

- Parameter

  Null

- Method
  - displayLoginForm() : void
    - Purpose: Renders the login form on the screen.
    - Algorithm: Displays input fields for username and password, along with a submit button.
  - validateInput() : boolean
    - Purpose: Verifies the correctness of the entered username and password.
    - Algorithm: Checks for non-empty fields and valid username format.
    - Related Attributes: Uses username and password.
  - displayErrorMessage() : string
    - Purpose: Shows an error message when login fails.
    - Algorithm: Sets errorMessage attribute and displays it on the screen.
  - redirectToHomePage() : void
    - Purpose: Redirects the user to the home page after a successful login.
    - Algorithm: Updates loginStatus and triggers a navigation action.
- Exception
  - InvalidInputException: When the input information is invalid.
  - LoginFailedException: When login fails.
- State
  - “loginStatus” changes from false to true if login successfully

4. Design for class “UpdateUserScreen”



- Attribute design

#	Name	Data type	Default value	Description
1	id	string	Null	User’s ID
2	mail	string	Null	User’s email
3	password	string	Null	User’s password
4	phoneNumber	string	Null	User’s phone number
5	username	string	Null	Username
6	address	string	Null	User’s address

- Operation design

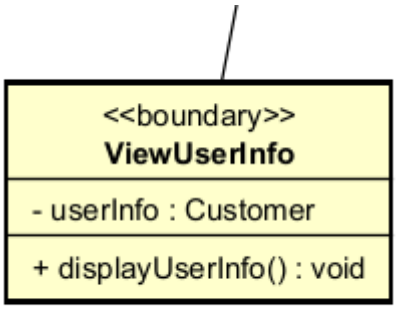
#	Name	Return type	Description (purpose)
1	displayUpdate Form	void	Display the update form
2	validateInput	boolean	Check whether all the input is valid for login
3	submitUpdate	string	Confirm and save update

- Parameter

  Null
- Method
  - displayUpdateForm() : void
    - Purpose: Renders the update form with current user information.
    - Algorithm: Displays input fields pre-filled with existing data (username, mail, phoneNumber).
  - validateInput() : void
    - Purpose: Validates user input to ensure correctness and completeness.
    - Algorithm: Checks for non-empty fields and valid formats (e.g., email, phone number).
    - Related Attributes: Uses username, mail, phoneNumber.
  - submitUpdate() : void
    - Purpose: Submits the updated information to the system.
    - Algorithm: Sends the updated data to the server and updates updateStatus based on the response.

- Exception
    - InvalidInputException: When the input information is invalid.
    - UpdateFailedException: When the update operation fails due to a system error or network issue.
- State
    - “updateStatus” changes from false to true if update successfully

5. Design for class “ViewUserInfo”



- Attribute design

#	Name	Data type	Default value	Description
1	userInfo	Customer	Null	

- Operation design

#	Name	Return type	Description (purpose)
1	displayUserInfo	void	Display the user’s info

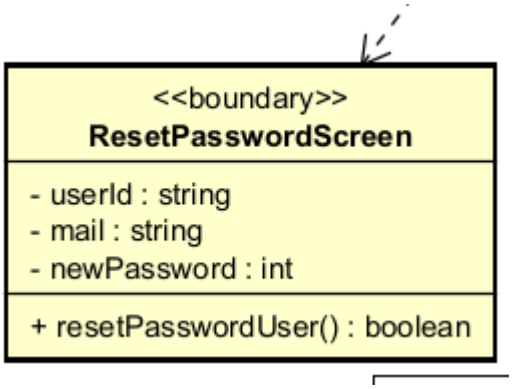
- Parameter

  Null

- Exception
    - DataRetrievalException: When there is any error for getting data from the system.
- State
- Method
    - displayUserInfo() : void
        - Purpose: Displays the user's information on the screen.

- Algorithm: Retrieves data from the userInfo attribute and presents it in a readable format.
- Related Attributes: Uses userInfo to access customer details.

6. Design for class “ResetPasswordScreen”



- Attribute design

#	Name	Data type	Default value	Description
1	userId	string	Null	
2	mail	string	Null	
3	newPassword	string	Null	New password from admin

- Operation design

#	Name	Return type	Description (purpose)
1	resetPassword	boolean	Reset to new password, return true if success, else return fail

- Parameter

  Null

- Exception
    - PasswordUpdateException: When there is any error for resetting the password
- State
- Method
    - resetPasswordUser() : boolean
        - Purpose: Resets the user's password.
        - Algorithm:
            - Validate the userId and mail input.
            - Generate or update the password with newPassword.
            - Save changes and return true if successful, false otherwise.
        - Related Attributes: Uses userId, mail, and newPassword.