Lab 3 – JavaFX Login System

Students
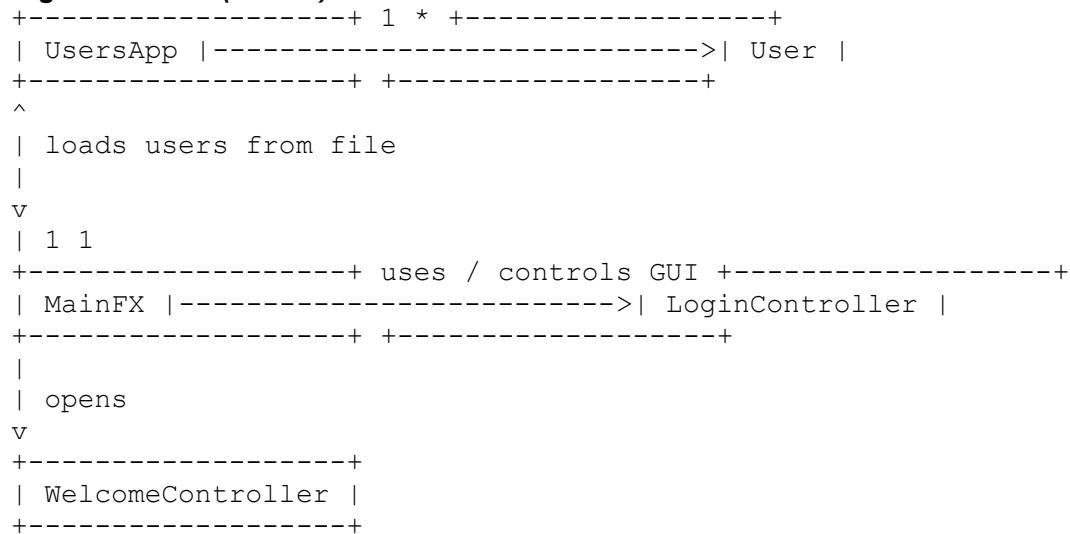• **Alaa Miari** (ID: 326180031)
• **Ofek Horowitz** (ID: 211576129)

Project Overview:
The goal of this lab is to implement a secure login system using JavaFX and FXML. The system
reads users from a text file, validates usernames and passwords according to strict rules, limits the
number of failed login attempts, and temporarily locks the user for *t* seconds after *n* failed attempts. A
successful login opens a separate Welcome window that greets the user by their email (username).

Main features:
• File-based user management (Users.txt.txt).
• Strict validation of email (username) and password according to lab requirements.
• Configurable login policy using parameters n (max attempts) and t (lock seconds).
• Thread-based handling of lock timer and attempts counter to keep GUI responsive.
• Separation between logic classes (User, UsersApp) and GUI classes (MainFX, LoginController, WelcomeController).

High-level UML (textual)
```
+------------------+ 1 * +----------------+
| UsersApp |---------------------------->| User |
+------------------+ +----------------+
^
| loads users from file
|
v
| 1 1
+------------------+ uses / controls GUI +------------------+
| MainFX |------------------------->| LoginController |
+------------------+ +------------------+
|
| opens
v
+------------------+
| WelcomeController |
+------------------+
```


Class-Level Design – Short Notes

1 MainFX (JavaFX Application Entry Point)

•
 Extends JavaFX Application and overrides start(Stage).
• Parses command-line arguments n (max attempts) and t (lock time in seconds).
• Loads login.fxml using FXMLLoader.
• Obtains a reference to LoginController and calls initConfig(n, t) to pass configuration.
• Sets up the primary stage (title, scene) and shows the login window.

2 User (Validation & Comparable User Object)

• Represents a single user with username (email) and password.
• Performs strict validation in the constructor:
• – Email format (must contain '@' and a valid domain).
• – Maximum username length (e.g. 50 characters).
• – Password length between 8 and 12 characters.
• – Password must contain at least: one uppercase, one lowercase, one digit, one special character.
• – Rejects invalid characters in the password.
• Implements Comparable<User> so users can be sorted by email (case-insensitive).
• Static helper isValidCredentials(username, password) reuses constructor logic to validate pairs.

3 UsersApp (Console Tester for User Class)

• Reads the Users.txt.txt file line by line.
• Splits each line into username and password.
• For each line, tries to create a new User.
• If the user is invalid, catches IllegalArgumentException and prints a clear message showing:
• – The original line from the file.
• – The exact error message (reason of failure).
• Collects all valid users in a list, sorts them using compareTo, and prints them.
• Used mainly to test the validation rules from Lab 1 & 2.

4 LoginController (Main GUI Logic)

• Connected to login.fxml via fx:controller="lab1.LoginController".
• Holds FXML fields: TextField usernameField, PasswordField passwordField, Label statusLabel.
• Keeps configuration parameters: maxAttempts (n), lockSeconds (t).
• Loads valid users into a Map<String,String> from Users.txt.txt in initConfig.

- Maintains per-username maps:
- – attempts: number of failed attempts per user.
- – lockUntil: timestamp until which the user is locked.
- Implements handleLogin(ActionEvent) as the event handler for the Login button.
- Starts a background thread to:
- – Check if the user is currently locked (uses lockUntil).
- – Compare input password with the stored one.
- – Validate credentials via User.isValidCredentials.
- – Update attempts and possibly lock the user when the max is reached.
- All GUI updates are executed via Platform.runLater(...) to keep JavaFX thread-safe.
- On successful login, opens the Welcome window by loading welcome.fxml and passing the username.

5 WelcomeController (Second Window Controller)

- Controller for welcome.fxml.
- Contains a Label welcomeLabel and a Button closeButton.
- Method setUsername(String username) updates the label to display: *"Welcome, <username>!"*.
- Method handleClose() closes the current stage (window) when the user presses the Close button.
- Provides a simple confirmation screen after successful login.

6 FXML Files (login.fxml & welcome.fxml)

- **login.fxml**:
- – Defines the login window layout using a VBox and GridPane.
- – Includes fields for username (TextField) and password (PasswordField).
- – Adds a status label for messages and a Login button with onAction="#handleLogin".
- **welcome.fxml**:
- – Simple VBox layout with a big welcome label and a Close button.
- – Connected to WelcomeController to show dynamic username text.

Input / Output Behavior

1 Inputs

• Text file Users.txt.txt in the project root folder (next to pom.xml).
• Each line: <username> <password>, separated by whitespace.
• Username must be a valid email; password must follow all complexity rules.
• Command-line arguments when launching MainFX:
• – n: maximum number of wrong attempts allowed.
• – t: lock time in seconds once the user reaches n failed attempts.

2 Outputs

• Console output from UsersApp showing which Users.txt lines are valid or invalid.
• Status messages in the login GUI:
• – *"Login failed. Attempts left: X"* for wrong credentials.
• – *"You used all n attempts. User is locked for t seconds."* when the limit is reached.
• Error alerts (JavaFX Alert) for missing file, invalid input, or lock messages.
• Welcome window confirming successful login with the user's email.