**Final Project Report**

Student Name: Daler Kim (김달레르)

Student ID: 2022315765

**1. Briefly describe the project purpose:**

The purpose of the project is to develop an interactive desktop application that allows users to view real-time weather information for any city globally. The application provides details such as temperature, humidity, wind speed, and weather conditions, enhancing user experience with features like saving favorite cities and ensuring data persistence across sessions.

**2. Draw the logic flow of the program (with flowchart):**

**A screenshot of a computer

Description automatically generated**

* Start
* Initialize UI
* User logs in or registers
* Main window opens
* User inputs a city name and requests weather data
* Application fetches weather data from API
* Weather data is displayed to the user
* User can add city to favorites
* User can log out (saving is automatically done)
* End

**3. Provide screenshots for each screen with brief description:**

1. Program starts with asking a user to login or register in system.
2. If the registration was successful user can now login and close the registration window.

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

1. User can now search weather for any city in search box under location label:

A screenshot of a computer screen

Description automatically generated

A screenshot of a weather forecast

Description automatically generated

1. If the user clicks the star button, current city will be saved to favorite list and now can be invoked at any time.
2. A screenshot of a computer

   Description automatically generatedA screenshot of a computer

   Description automatically generatedUser can log out with favorite list automatically being saved.

**4. Explain the code of the main functionalities**

* “AccountManager” handle user login and registration. loginUser checks if the provided credentials match a user in the system. registerUser adds a new user if the username isn't already taken.
* The “LoginFrame” class is responsible for rendering the login interface where users can either log in with existing credentials or register for a new account. This class uses AccountManager to authenticate users and register new accounts. It also provides feedback to the user in case of successful or failed authentication.
* The “UI” class handles all user interactions related to weather data display. It updates the UI components based on user input and selections, such as fetching weather for a chosen city or updating favorite cities.
* “WeatherFetcher” SwingWorker class is responsible for making an asynchronous call to the weather API. It uses HTTP to request weather data, processes the response, and updates the UI components without freezing the application.
* The “LogoutHandler” class oversees the logout process, ensuring that the user's session ends properly, and any desired data is saved.

**5. Explain what is included in your project and why it is used (Polymorphism, Inheritance, File I/O, etc)**

* **Inheritance**: Used in extending classes from Java Swing components to create a user interface.
* **Polymorphism**: Utilized in event handling to respond to user actions.
* **File I/O**: Implemented to save and load user data and favorite cities.
* **Multithreading** (SwingWorker): Used for making API requests in the background to avoid freezing the UI.
* **Exception Handling**: To manage potential errors during File I/O or API requests.
* **Swing GUI**: The entire user interface is built using Swing, it provides a range of components like buttons, labels, text fields, and panels that I have customized and used throughout my application.
* **Event-Driven Programming**: This application follows an event-driven programming model, common in GUI applications. This model allows the program to respond to various user interactions, ensuring a dynamic and interactive user experience.
* **Data Structures**: The use of collections like lists and maps facilitates efficient data management and retrieval, particularly when handling the list of favorite cities.
* **JSON Parsing**: By parsing JSON responses from the weather API, my application can extract and display weather data, showcasing the ability to integrate with external services.