Final Project Report:

TaskEase Management Application

Team Members: Janhavi Kulkarni, Abdu Raziq Hidayathulla, Vyshanavi Gavara

Table of Contents

1. Introduction

This document outlines the functionality, design framework, and the overall progress of the TaskEase software project.TaskEase is a comprehensive task management application designed to simplify task organization and boost productivity for individuals and teams. Whether you're a busy professional juggling multiple projects, a collaborative team member working towards common goals, or a diligent student managing academic assignments, TaskEase provides the essential tools you need to stay organized, focused, and on track. With its intuitive interface and robust features, TaskEase empowers users to create, manage, prioritize, and track tasks effortlessly, streamlining taskmanagement and reducing stress.

1.1 Purpose and Scope

The purpose of this document is to provide a detailed description of the TaskEase software, its requirements, and its design considerations. It also serves as a reference for the development team and the client commissioning the project.

1.2 Target Audience

This document is intended for the development team, project stakeholders, and the client.

2. Project Design

2.1 Design Decisions

Design Methodology:

We opted for an object-oriented approach given the bounded entities within the project. While a functional programming approach was also considered, we chose OOP for its modularity and scalability, making future expansions and complexity additions easier. Java was chosen as the implementation language.

UML Documentation:

The project design was documented using UML. The following UML diagrams were created:

- Use Case Diagram
- Class Diagram
- FlowChart Diagram

2.2 Adjustments to the Design

While the initial design was robust, some adjustments were made after coding began:

- UI/UX Improvements: During implementation, we realized the need for better user interaction. Some changes were made to the UI to enhance user experience.
- Database Management: Initially, we planned to use a local file for data storage. However, during implementation, we decided to switch to a more scalable solution by integrating a database.

TaskEase–Simplify Your Task

By: Abdu Raziq, Janhavi, Vyshnavi

Agenda

Introduction

Problem Statement

Objective

System Architecture

Implementation

Conclusion

INTRODUCTION

- Introducing Task Manager, the free, user-friendly app designed to help you streamline your daily activities and stay organized effortlessly.
- With its simple interface, you can easily add, manage, and track tasks without the need for continuous internet access.
- Whether you're a busy professional or a student, Task Manager is your go-to solution for keeping on top of your to-dos, anytime and anywhere.
- Users can visually track their progress on various tasks or projects, which can boost motivation and provide a sense of accomplishment.

PROBLEM STATEMENT

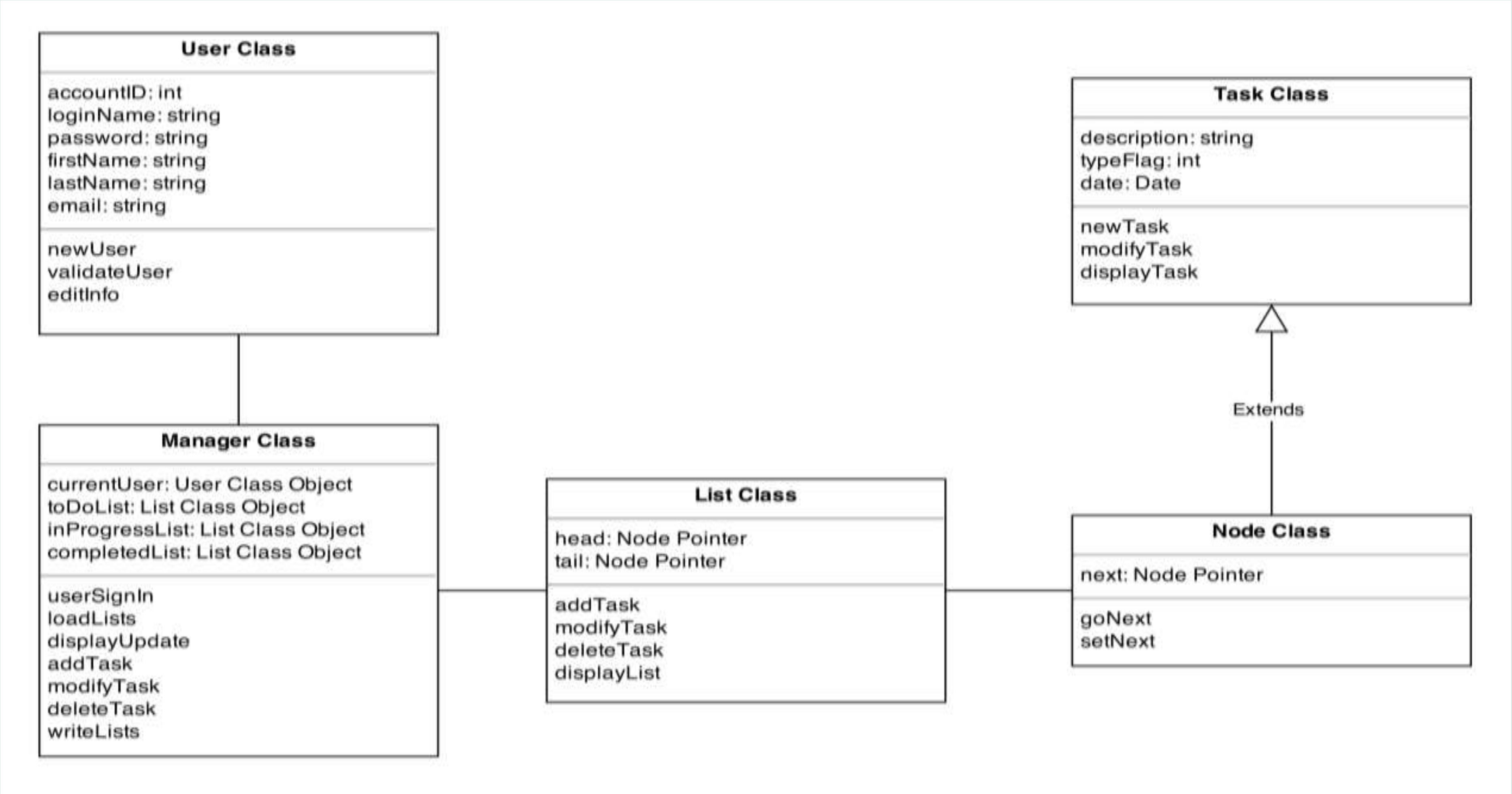
- In a hectic world, staying organized can be a challenge, leading to missed deadlines and increased stress. Existing task management tools often lack simplicity and struggle to meet users' needs effectively. A user-friendly, offline-capable app with intuitive features is needed to streamline task management, boost productivity, and alleviate the burden of daily responsibilitie.

OBJECTIVE

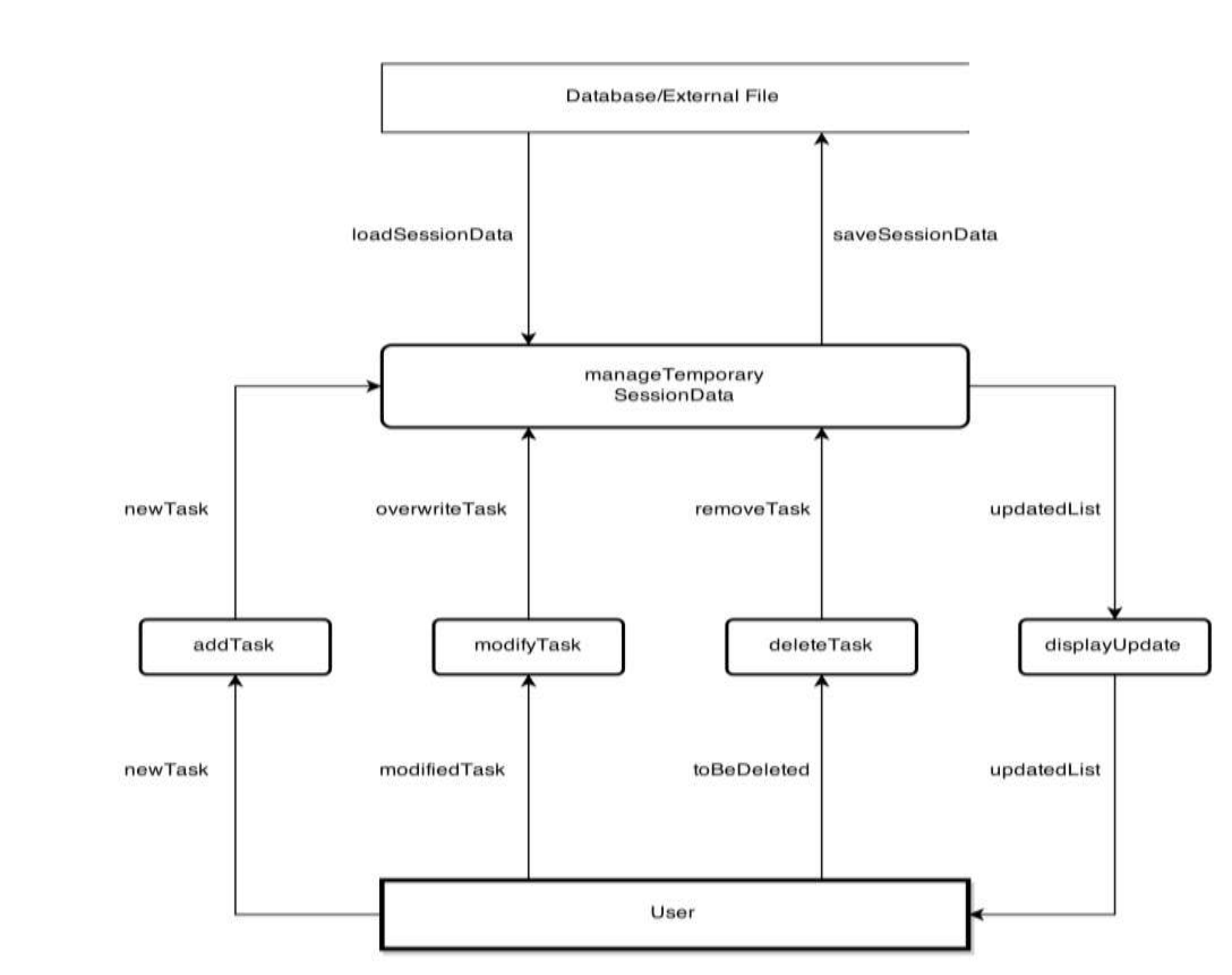
- **Simplify Task Management:** Create an app that makes organizing tasks straightforward and effortless, ensuring that users can easily add, prioritize, and track their daily activities without unnecessary complexity.
- **Enhance Productivity:** Develop intuitive features within the app that empower users to effectively manage their workload, meet deadlines, and reduce stress, ultimately boosting their overall productivity and efficiency.
- **Ensure Offline Accessibility:** Design the app to operate seamlessly offline, allowing users in areas with unreliable internet access or those who prefer offline functionality to manage their tasks without interruption, ensuring accessibility for all users.

SYSTEM ARCHITECTURE

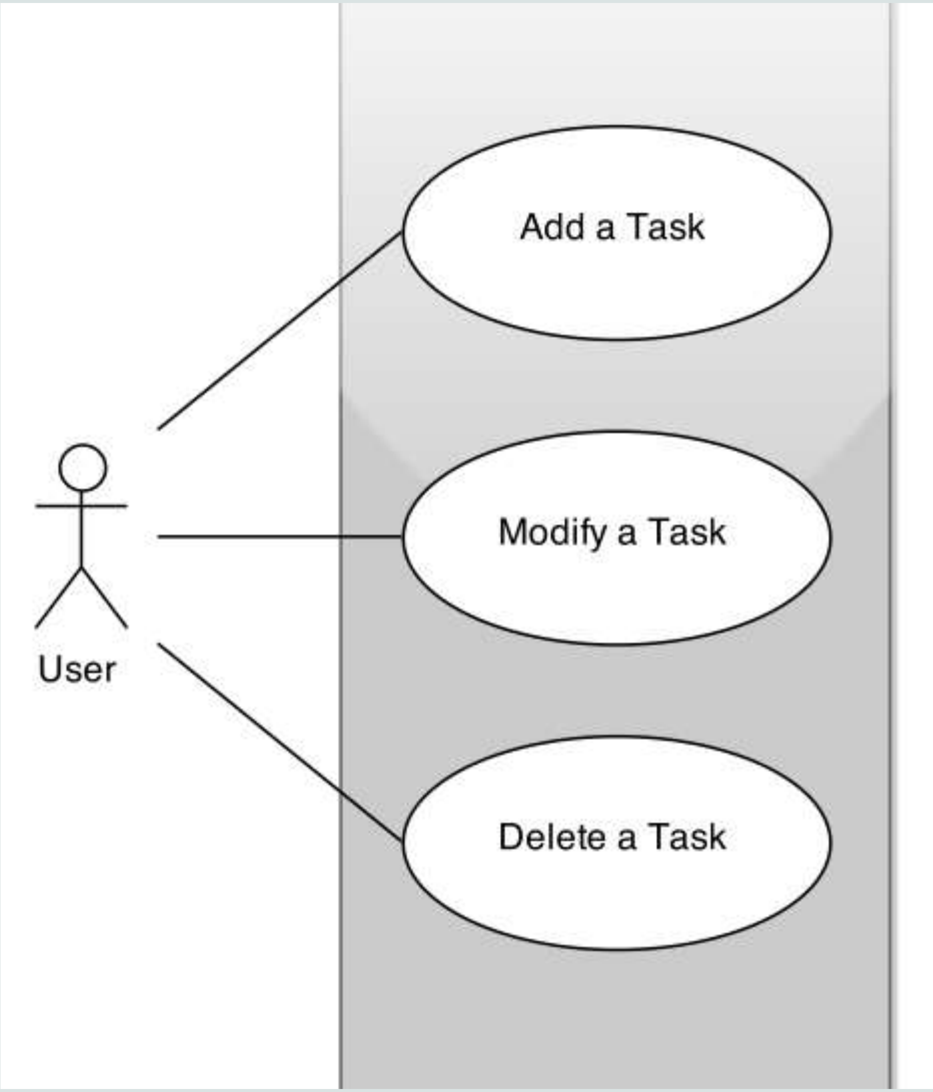
CLASS DIAGRAM



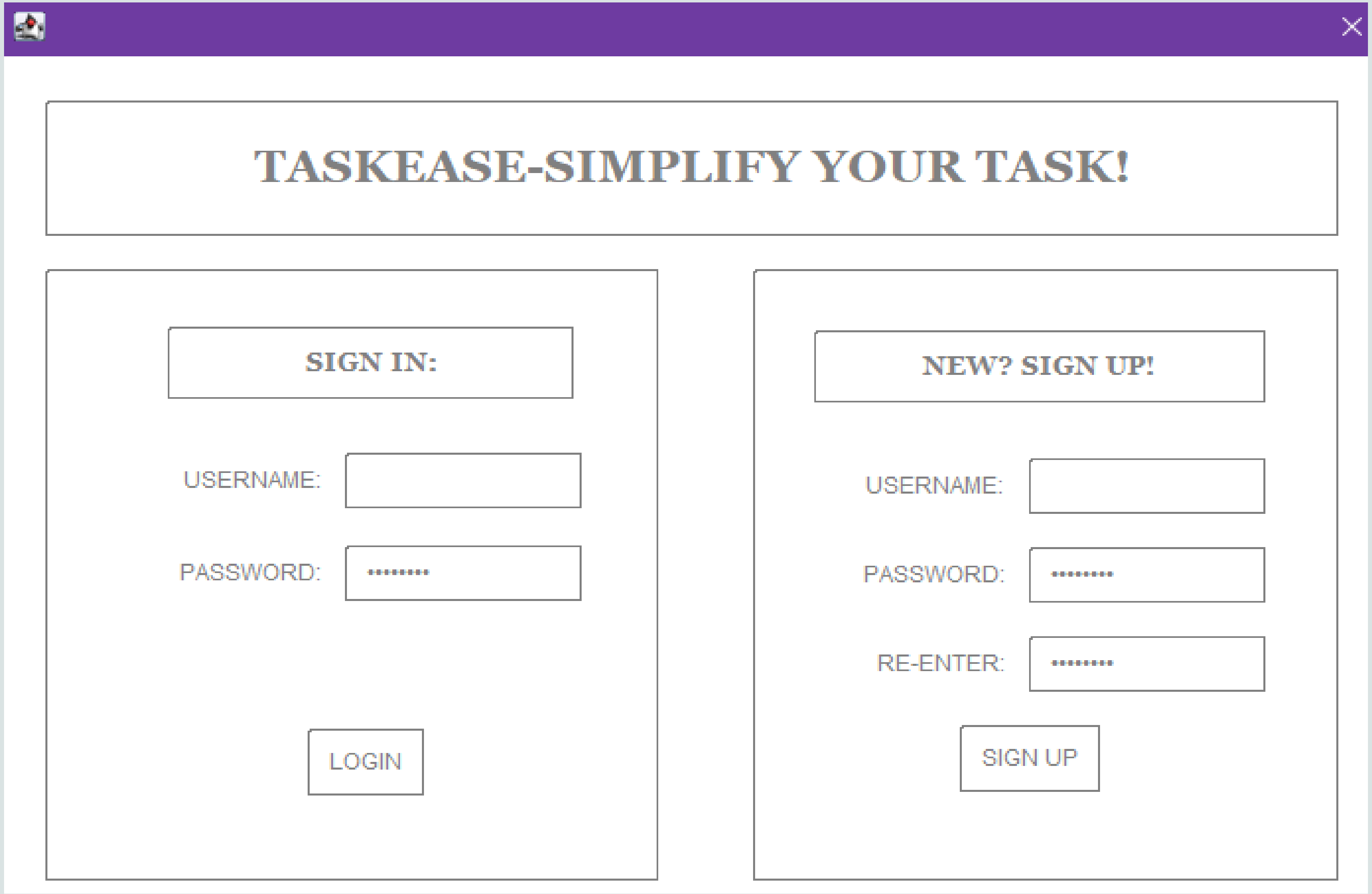
FLOW DIAGRAM



USE CASE DIAGRAM



Login and Signup Page



TASKEASE-SIMPLIFY YOUR TASK!

SIGN IN:

USERNAME:

PASSWORD: •••••••••

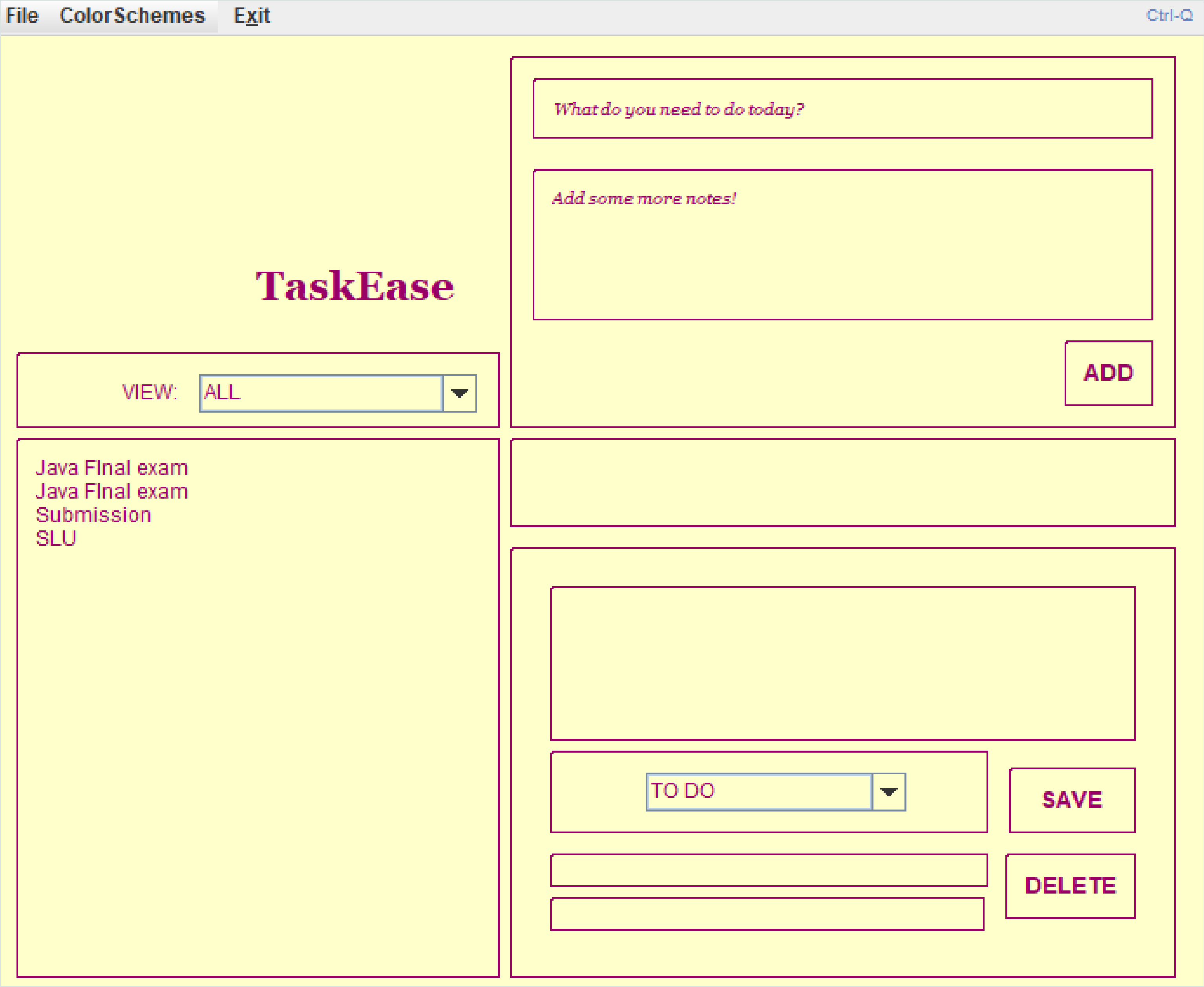
NEW? SIGN UP!

USERNAME:

PASSWORD: •••••••••

RE-ENTER: •••••••••

TaskEase Application



What do you need to do today?

Add some more notes!

VIEW: ALL

Java FInal exam
Java FInal exam
Submission
SLU

TO DO

IMPLEMENTATION DEMO

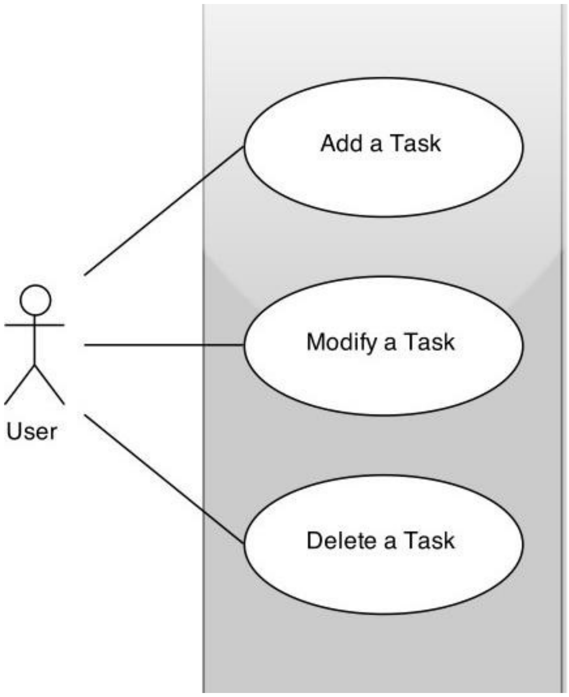
CONCLUSION

- **User-friendly interface** and intuitive features simplify task management.
- **Offline accessibility** ensures uninterrupted usage regardless of internet connectivity.
- The app **enhances productivity** and reduces stress by empowering users to meet deadlines efficiently.
- Overall, Task Manager aims to **streamline task management**, improve productivity, and enhance users' well-being.

THANK YOU

3. System Overview

3.1 Use Case Diagram



4. System Architecture

4.1 Manage Temporary List Data

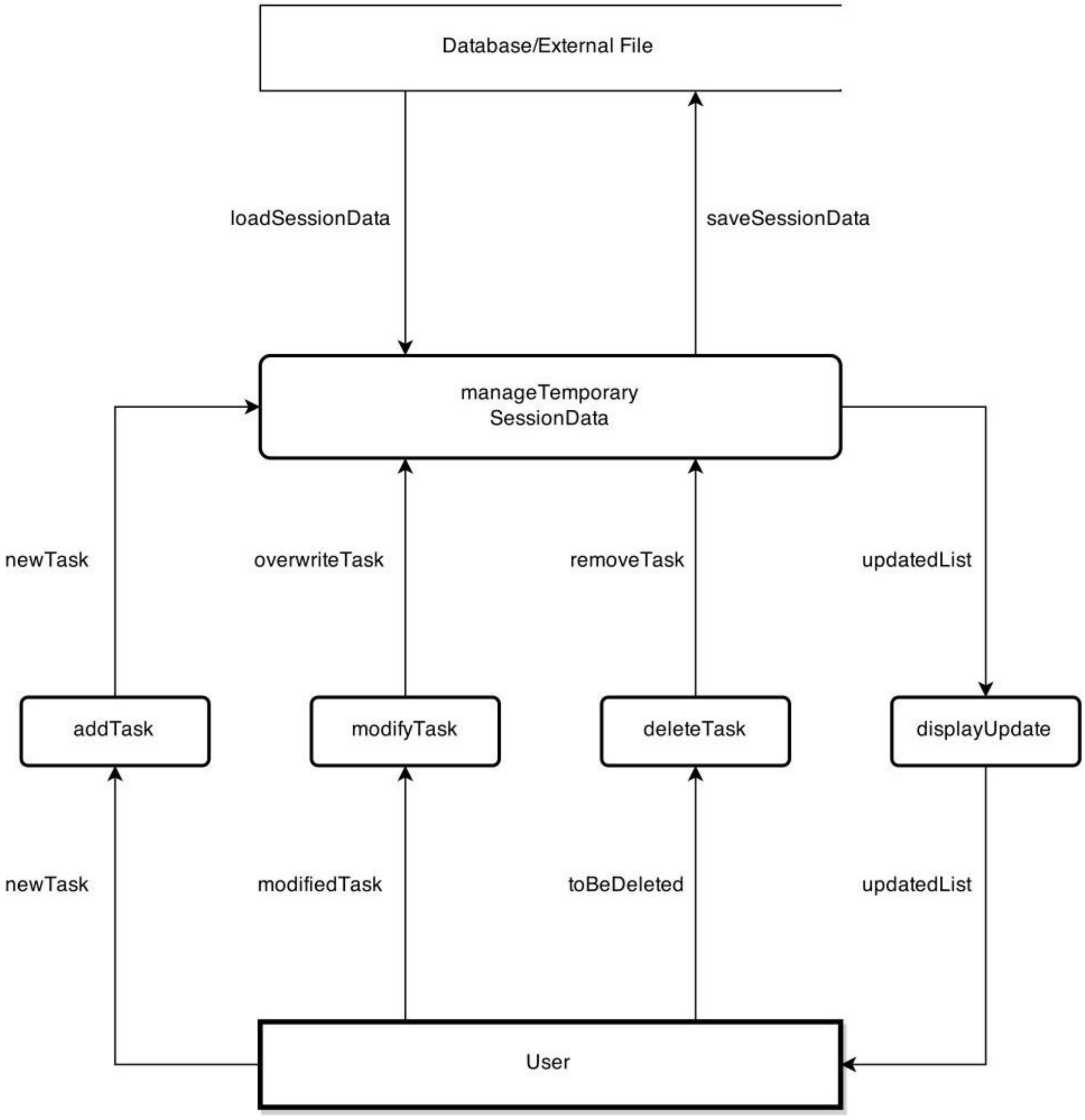


Fig: FlowChart Diagram

5. Detailed System Design

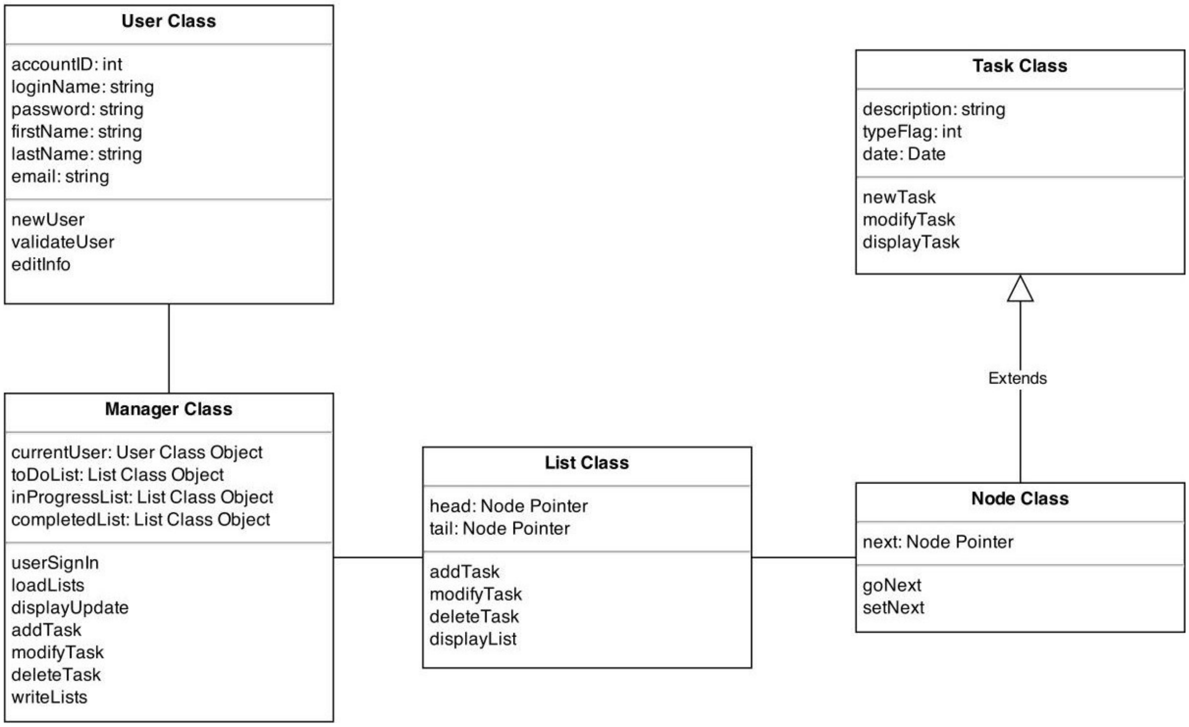


Fig: Class Diagram

5.1 Manager Class

This class is responsible for managing the flow of data within the software. It will first verify the user's credentials, then load the lists from the external database, execute the user's commands and save the changes to the external file. When the program opens and the external data is loaded, the Manager Class will generate three lists: To Do, In Progress and Completed. The three lists will be modified throughout each program session and will be written to the external file as the user logs out or closes the program. Manager Class will also contain several helper functions, such as greeting and farewell messages, and functions to capture and validate user input.

5.2 User Class

This class is used to connect an individual user to their data using their unique ID number. It also stores details about the user including the login name and password needed to verify the user's credentials.

5.3 List Class

Each Manager Class has three objects of the List Class: To Do, In Progress and Completed. Lists will be implemented as singly linked lists sorted by date. The List Class is responsible for the insertion, deletion, modification and display of objects in the list.

5.4 Task Class

Task Class objects hold the meat of our data. Each object of this class is an item on the to-do list which consists of a field for the description of the task. This class could be easily extended and articulated in greater complexity with more fields for the user to fill in, such as links, location, collaborators, due date, urgency flag, photos and attachments, etc.

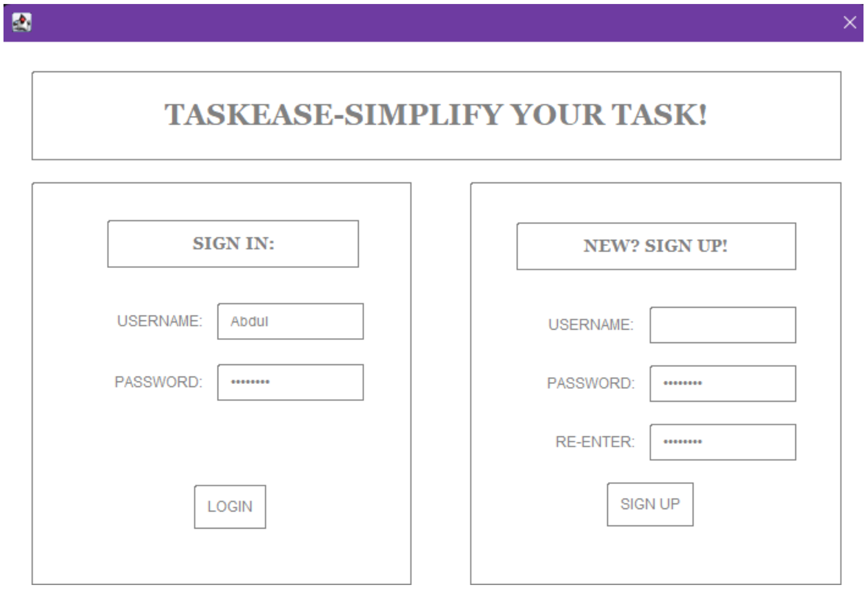
5.5 Node Class

Node Class is a derived class of Task Class, containing all attributes of the Task Class with the addition of a "next" pointer to allow it to link up with other nodes in the list. Due to encapsulation, Node Class also contains only two methods used to get and set its sole data member, the "next" pointer.

6. Snapshots of the Project

6.1 Login and Signup Page:

- The login and signup page serves as the entry point for users to access the Task Manager app.
- Users can either log in with their existing credentials or sign up for a new account.
- The page features user-friendly input fields for email/ username and password, with options for password recovery or account creation.



USERNAME: Abdul

PASSWORD: ••••••••

USERNAME:

PASSWORD: ••••••••

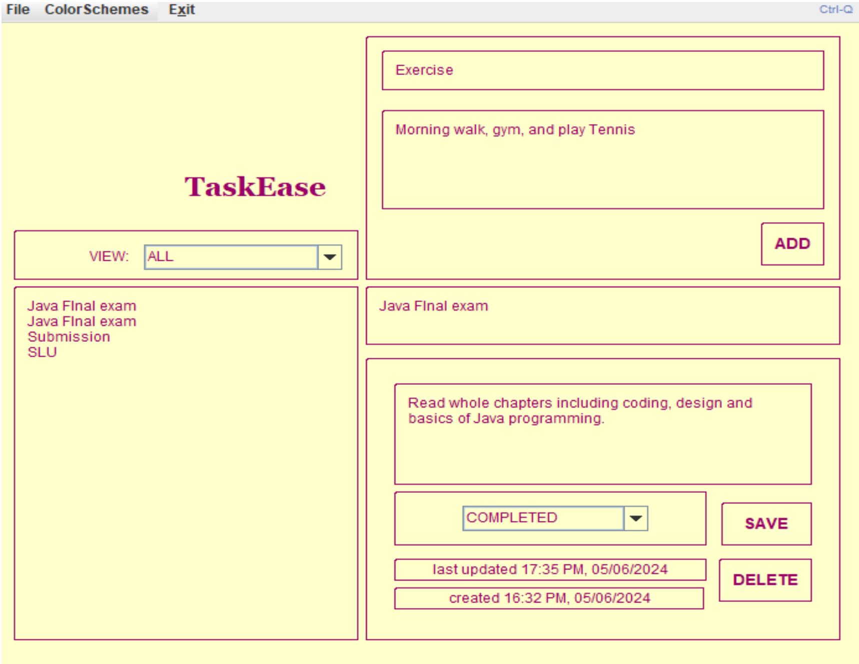
RE-ENTER: ••••••••

Fig: Login and Signup Page

6.2 Home Page for TaskEase Management:

- Upon successful login, users are directed to the home page, which serves as the central hub for managing tasks.

- The home page displays a personalized dashboard where users can view their upcoming tasks, flagged items, and completed tasks.
- It includes intuitive controls for adding new tasks, setting deadlines, assigning priorities, and marking tasks as completed.
- Users can also edit or delete tasks directly from the home page, ensuring seamless task management.



Exercise

Morning walk, gym, and play Tennis

VIEW: ALL

Java FInal exam
Java FInal exam
Submission
SLU

Java FInal exam

Read whole chapters including coding, design and basics of Java programming.

COMPLETED

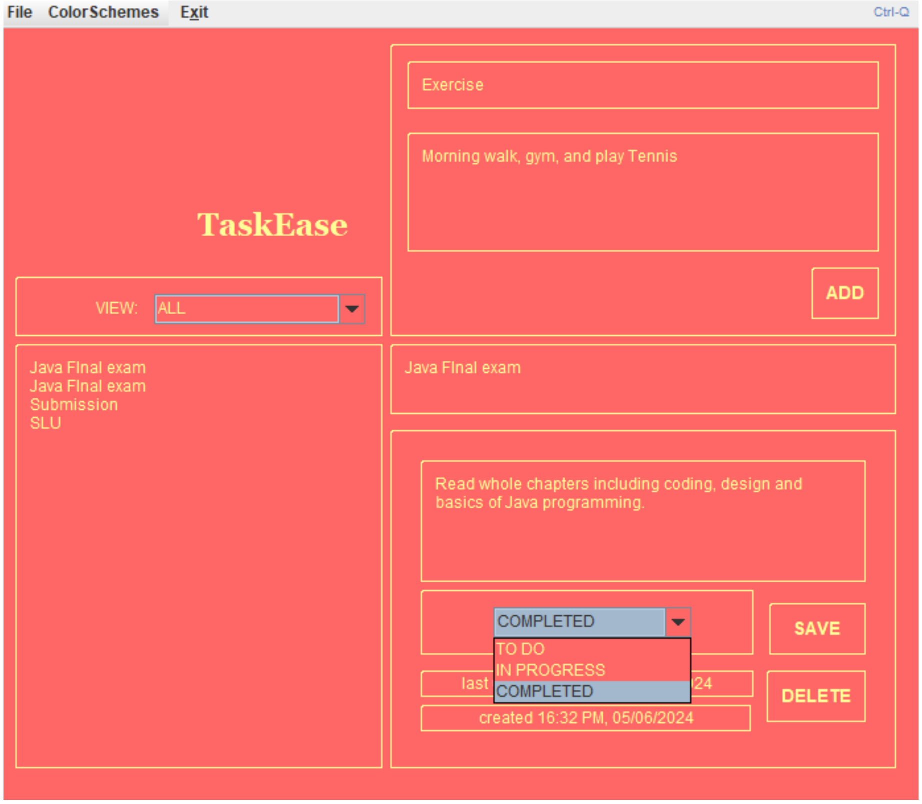
last updated 17:35 PM, 05/06/2024

created 16:32 PM, 05/06/2024

Fig: Home page

6.3 Theme-changing Feature:

- To enhance user experience, the Task Manager app offers a theme-changing feature that allows users to personalize the app's appearance according to their preferences.
- Users can choose from a selection of predefined themes or customize their own color scheme.
- The theme-changing feature is accessible from the app settings or directly from the home page, providing convenience and flexibility.
- By allowing users to customize the app's theme, Task Manager enhances user engagement and satisfaction, catering to individual preferences and aesthetics.



Exercise

Morning walk, gym, and play Tennis

Java FInal exam
Java FInal exam
Submission
SLU

Java FInal exam

Read whole chapters including coding, design and basics of Java programming.

COMPLETED

last 24

created 16:32 PM, 05/06/2024

Fig: Theme change Feature

7. Accomplishments and Reflection

7.1 What was accomplished?

- Successful Implementation: We successfully implemented all planned features.
- User-Friendly Interface: The UI/UX design was well-received by users.
- Scalability: The project architecture allows for easy scalability.

7.2 What was not completed?

- Comprehensive Testing: While testing was done, we could have benefited from more comprehensive test cases.
- Online Account Support: Although initially planned, the feature to sign into TaskEase accounts from public computers and access information using the web version of the software was not implemented in this version.

7.3 Discussions and Changes

- Database Integration: We initially planned to use a local file for data storage but later decided to integrate a database for better scalability.
- UI/UX Improvements: Some changes were made to the UI during implementation to enhance user experience.
- Future Development: Discussions were held regarding future development, including the implementation of online account support and additional user-specified categories.

7.4 Lessons learned for future projects

- Importance of Scalability: Considering scalability from the beginning is crucial for long-term project success.
- Thorough Planning: While our planning was good, more detailed planning upfront could have saved time during implementation.

8. References

- https://www.javatpoint.com/java-swing
- https://docs.oracle.com/javase%2F7%2Fdocs%2Fapi%2F%2F/javax/swing/package-summary.html
- https://www.geeksforgeeks.org/introduction-to-java-swing/