Project title	Project Management Application
Author(s)	Beáta Keresztes, Borbála Fazakas
Group	30422

Project Management Application

Overview

Our application is intended to help teams organize their shared projects and tasks, assign responsibilities and track the progress. The functionalities provided by our app are expected to increase the efficiency of teamwork, especially in a virtual environment.

Goals and Functionalities

Based on the main goals presented above, we plan to implement the following functionalities for the app:

1. User Registration and Authentication

A new user can register with a username and a password. These will be required later when the user wants to sign in and access the further functionalities of the application.

2. Team Creation and Management

Any user can create a new team with a specific name, and thus the creator will become the manager of it.

As a manager, the user has access to the following functionalities:

- Delete the team
- Add new team members based on the username
- Remove members from the team
- Change the name of the team
- Regenerate the unique code of the team which is used for joining the team
- Pass the manager status to another member

Furthermore, users other than the manager may

- Join the team using the unique code generated for the team
- Leave the team (the manager can leave the team only after passing the manager status to someone else)

3. Projects

a. Creation and Management

Any member of a team may create a new project which has the following attributes:

- Name
- Deadline

- Assignee: a team member who is responsible for this specific project
- Supervisor: a team member who monitors the progress, evaluates the proposed final version and decided whether it fulfills the requirements
- Description
- Status (to do/in progress/marked as done/finished)
- Contributors: the list of users who have worked on this project

The creator of the project will automatically become the supervisor, and thus will gain access to the following options:

- Change the name, deadline, assignee and description of the project
- The supervisor is the only one who can mark a project as “finished” after any other user marked it as done. If the supervisor decides that the requirements are not fulfilled, they may return the project and set its status back to “in progress”
- The supervisor may also pass the supervisor status to any other user

The Assignee of the project can

- Mark the project as done (set the status to “marked as done”)

Any member of the team in which the project was created can:

- Mark it if they contributed to the project
    - At the moment when the first user marks their contribution, the project’s status will be set from “to do” to “in progress”

b. Progress Tracking

Any user can easily track the progress of

- Projects assigned to them
- Projects supervised by them

To make the monitoring process easier, they can

- View the projects categorized by status
- View the projects ordered by deadline
- View overdue projects

Technologies used

The project will be implemented in form of a desktop application, using only Java language.

In order to create the user interface, we decided to use the **Java Swing** library. With the aid of the components provided by this framework, we can create a user-friendly interface for our application, with windows, forms, dialogs, etc.

Additionally, a database is needed in order to store the data across multiple executions of the program. Having the goal of easily transferring the app in form of a jar file, we needed to choose a lightweight embedded database, so we finally decided to implement our app with **SQLite**. Database access is achieved through the JDB API and through using the SQLite driver.

Furthermore, to make building and importing dependencies easier, we use the **Maven** project management tool.

Design

Regarding the overall design of the implementation, we stick to the MVC pattern in order to achieve a high-level separation of concerns.

Model

The model

- Defines the POJO classes for our main entities (User, Team, Project)
- Implements the business logic (it knows about access levels, rules for progress monitoring, …) in the "Manager" classes
  - UserManager
  - TeamManager
  - ProjectManager
- Provides access to the database
  - UserRepository
  - TeamRepository
  - ProjectRepository
- Due to the Observable pattern, it can notify the View about changes so that the View can be updated correspondingly

View

The view represents the front-end part of the application. It is responsible for the creation of the user interface, displaying the model in a user-friendly way, and allowing the user to interact through it with the components of the model. It only handles the UI and design part of the application, the functionalities are implemented by the other components.

The Views which manage the User Interface are:

- Sign-in and Sign-up Views: allow the user to sign-in or create an account and access the application.
- Main View and the Main Menu: allow the user to view and manage their teams and their account information or leave the application.
- Create Team View and Join Team View: allow the user to create or become the member of a certain team.
- Account Settings View: allow the user to edit their account information, such as the username or password.
- …

Controller

The controller is the component which handles all the actions of the UI and sends the corresponding command to the model.