Employee Tasks Application

1. Setup guide

- Download the repositorium from Github by clicking the green “Code” button and selecting “download ZIP” from the dropdown menu: https://github.com/bezimeniludak/EmployeeTaskApplication
- Extract the file with WinRar or 7Zip
- Open the folder “EmployeeTaskApplication-main”
- Open the file “EmployeeTaskApplication.sln” with Visual Studio
- Start the application with F5 or by clicking the “EmployeeTaskApplication” button with the green arrow in the toolbar

2. Overview

The application is a .NET Windows Forms Application and uses the Model-View-Controller (MVC) architectural pattern.

The Model layer consists of three classes: employee, task and project. The employee class contains details about employees, such as their name, email address, phone number, date of birth and monthly salary. The task class is responsible for holding information about individual tasks, including their title, description, due date and assignee name. The project class holds data about each project, including its title, description, start and end dates, budget as well as a list of tasks.

To interact with this data, a set of forms has been created that act as both the views and controllers. Each form is responsible for displaying and manipulating data related to a specific component of the model. All forms are displayed in a MDI container, which makes the whole application run in a single window. The main menu form serves as the main hub for the application, providing buttons to access all other forms.

The employee, tasks and project forms all have a list box that displays all instances of that class. When an instance is clicked on all the textboxes and datetime pickers next to the list box get updated with the data of the instance allowing the user to easily modify it. The user can then choose to update that instance or create a new one. They can also remove the selected instance or go back to the main menu. If the user deletes a project instance all the tasks that were in the project get deleted as well. The application uses text files to store the data for tasks, employees, and projects. When the application starts up, it reads the data from the text files and creates instances of the corresponding classes. When the user creates, updates, or deletes an instance, the application writes the changes to those files.

The statistics form displays 4 buttons which when pressed display the data in a message box. The first one shows the list of top 5 employees with the most tasks in the past month. The second one displays the list of the top 5 employees with the most tasks in the current month. The third button shows the top 5 employees with the biggest monthly salaries. The fourth button orders the list of projects by the number of tasks they have and shows the top 5.

3. Additional functionalities

The additional functionalities include the project class and the additional projects statistic. I chose to expand the application with the project class because it provides a way to organize tasks and group them into projects. This allows for easier management of tasks and enables users to view all the tasks associated with a specific project in one place. It also helps to track progress on a project and determine if it is on schedule.

The new statistic could provide valuable insights into the workload of different projects and could help with resource allocation and scheduling. This addition should make the application more useful and informative for users.