**Yaşar University**

**Spring, 2022-2023**

**SE2224 - Software System Analysis**

**Final Project Report**

|  |  |
| --- | --- |
| **Student Name:** | **Mert Kahraman** |
| **Student No:** | **20070006020** |
| **Department Name:** | **Software Engineer** |
| **Course Section No:** | **2224.01** |

**Table of Contents**

[**1** **Introduction** 3](file:///C:\Users\Mert\Downloads\SE2224_FinalProjectReportTemplate.docx#_Toc101958757)

[**2** **Requirements Definition** 3](file:///C:\Users\Mert\Downloads\SE2224_FinalProjectReportTemplate.docx#_Toc101958758)

[**2.1** **Functional Requirements** 3](file:///C:\Users\Mert\Downloads\SE2224_FinalProjectReportTemplate.docx#_Toc101958759)-4

[**2.2** **Nonfunctional Requirements** 5](file:///C:\Users\Mert\Downloads\SE2224_FinalProjectReportTemplate.docx#_Toc101958760)

[**3** **Use Case Analysis** 5](file:///C:\Users\Mert\Downloads\SE2224_FinalProjectReportTemplate.docx#_Toc101958761)

[**3.1** **Actors** 5](file:///C:\Users\Mert\Downloads\SE2224_FinalProjectReportTemplate.docx#_Toc101958762)

[**3.2** **Scenarios** 5-6](file:///C:\Users\Mert\Downloads\SE2224_FinalProjectReportTemplate.docx#_Toc101958763)

[**3.3** **Use Cases** 6-9](file:///C:\Users\Mert\Downloads\SE2224_FinalProjectReportTemplate.docx#_Toc101958764)

[**3.4** **Relationships among Actors and Use Cases** 9](file:///C:\Users\Mert\Downloads\SE2224_FinalProjectReportTemplate.docx#_Toc101958765)

[**3.5** **Use Case Diagram** 9-10](file:///C:\Users\Mert\Downloads\SE2224_FinalProjectReportTemplate.docx#_Toc101958766)

[**4** **Behavioral Models** 10](file:///C:\Users\Mert\Downloads\SE2224_FinalProjectReportTemplate.docx#_Toc101958767)

[**4.1** **Sequence Diagram** 10-11](file:///C:\Users\Mert\Downloads\SE2224_FinalProjectReportTemplate.docx#_Toc101958768)

[**5** **Structural Models** 12](file:///C:\Users\Mert\Downloads\SE2224_FinalProjectReportTemplate.docx#_Toc101958769)

[**5.1** **Class Diagram** 12](file:///C:\Users\Mert\Downloads\SE2224_FinalProjectReportTemplate.docx#_Toc101958770)

[**6** **Process Modeling** 12](file:///C:\Users\Mert\Downloads\SE2224_FinalProjectReportTemplate.docx#_Toc101958771)

[**6.1** **Data Flow Diagram (DFD)** 12-13](file:///C:\Users\Mert\Downloads\SE2224_FinalProjectReportTemplate.docx#_Toc101958772)

[**7** **Final Version(s) of the Graphical User Interface(s)** 13-22](file:///C:\Users\Mert\Downloads\SE2224_FinalProjectReportTemplate.docx#_Toc101958773)

[**8** **Conclusion and Future Work** 22-23](file:///C:\Users\Mert\Downloads\SE2224_FinalProjectReportTemplate.docx#_Toc101958774)

# **1 Introduction**

Using the software program Task Planner, users may efficiently and conveniently organize their responsibilities and activities. The user must have a valid username and password to log in to the application. When customers log in to the system first, a notification message will appear, showing the tasks that are 1 day away from the deadline. After that main page will greet the user. In the main screen there are 9 buttons. These are Insert a task, Update a task, Delete a task, Show all tasks, View a task’s image, Reset, Exit, Filter tasks and Sort by priority. There are 5 pieces of information that the user has to fill in. These are task name, description, deadline, priority and reminder image. Moreover, the task planner automatically gives a task's database ID and entry date. User can create a new task by using “Insert a task button”. User can update a task or delete a task instantly by using “Update a task” or “Delete a task” button. When user clicks on the “Show all tasks “ button the system will display the table with the task information. If task have a image you can see the image by using “View a task’s image” button and system have a 2 filter button these are “Filter tasks” and “Sort by priority”. If you want to use the filter tasks button, you must first choose which date range tasks you want to see and you can sort the tasks by using “Sort by priority” button. Task Planner is a useful tool for users to stay organized, manage tasks successfully, and accomplish their goals thanks to its user-friendly layout and wide range of functionality.

# **2 Requirements Definition**

## **2.1 Functional Requirements**

Write all the functional requirements in your project.

1. Login:

• The system should allow registered users to log in by entering their username and password. Upon successful authentication, the user should be granted access to the main application interface.

• Upon successful login, the system should display a notification message to the user if there are any tasks due within one day.

• The system must protect the user's username and password from danger.

2. Insert a Task:

• The system should provide a form or input fields where users can enter the given information for a new task

• The system should automatically generate a unique task ID for each new task created. Additionally, the system should record the date and time when the task was entered into the system.

• After successfully creating the task, the system should display a confirmation message to the user, indicating that the task has been successfully inserted into the database.

• The system should validate the user's input to ensure that all required fields are filled in and that the entered data is of the correct format and within acceptable limits.

3. Edit & Update a Task:

• The system should provide a mechanism for users to select a task from the list of existing tasks that they want to update.

• Once the user has selected a task, the system should display the current details of the selected task, including the task name, description, deadline, priority, and reminder image (if available). Users should be able to modify any or all of these task attributes.

• After successfully updating the task, the system should display a confirmation message to the user.

4. Delete a Task:

• The system should provide a mechanism for users to select a task from the list of existing tasks that they want to delete.

• The system should delete the selected task from the task database.

5. Show all Tasks:

• When the user clicks on the "Show All Tasks" button, the system should retrieve and display a table or list of all the tasks

• The user interface for displaying tasks should be intuitive and visually organized, making it easy for users to scan and comprehend the task information

6. Filter a task:

• The system should provide options for users to select a date range to filter tasks.

• The system should not display tasks out of range to the user

7. Sort by Priority:

• The system should provide options for users to select a date to sort.

• The system should sort the task list in ascending order based on the selected priority sorting option.

8. View a Task’s Image:

• The system should support a variety of common image formats.

• The system must display the image if the selected task has a image.

## **2.2 Nonfunctional Requirements**

# Performance: The Task Planner software should respond quickly and handle a large number of tasks and users concurrently without severe delays or performance concerns.

# Usability: The user interface of the Task Planner should be intuitive, easy to navigate, and visually appealing. It should require minimal training for users to understand and operate the software effectively.

# Security: The Task Planner should ensure the confidentiality and integrity of user data. It should employ robust authentication mechanisms to prevent unauthorized access and encryption to protect sensitive information.

# Reliability: The software should be highly reliable and available, minimizing the occurrence of system failures, crashes, or data loss. It should have proper error handling mechanisms to gracefully recover from unexpected errors or exceptions.

# Scalability: The Task Planner should be scalable to accommodate a growing number of users and tasks. It should be able to handle increasing loads without degrading performance or requiring significant hardware or infrastructure upgrades.

# **3 Use Case Analysis**

## **3.1 Actors**

**USER**

The user logs into the system using her username and password and can add, edit, sort and view his/her tasks using the facilities available to his/her.

## **3.2 Scenarios**

**Scenario name: Inserting a New Task**

Actor: Mert

Flow of Events:

1. Mert logs into the Task Planner application by using his personal username and password.

2. The system shows the fields that need to be filled and the table if there are any previously saved task(s).

3. Mert fills the required fields, these are task name, description, deadline and reminder image. There is also a priority area according to the user's request.

4. Mert clicks on the “Insert a Task” button to create a new task.

6. The system adds the task to table and database.

5. The system shows the confirmation message.

**Scenario Name: Deleting a New Task**

Actor: Mert

Flow of Events:

1. Mert logs into the Task Planner application by using his personal username and password.

2. Mert clicks on the "Show all tasks" button to be able to select the task you want to delete.

3. Mert selects the task which task he want to delete.

4. Mert clicks on the “Delete a Task” button.

5. System deletes the task from table and database.

6. The system shows the confirmation message.

## 

## **3.3 Use Cases**

**COMPLETE USE CASE FORMS**

**Use Case 1: Insert a Task**

Actor: User

Description: The user can create and insert a new task into the Task Planner program.

Preconditions: User is logged into the Task Planner application.

Postconditions: The new task is successfully added to the task database.

Basic Flow:

1. System prompts the user to enter the task details: task name, description, deadline, priority, and reminder image.

2. User fills in the required information for the task.

3. User confirms the task details.

4. User clicks on the “Insert a task” button.

5. System generates a unique database ID and entry date for the task.

6. System saves the task in the task database.

7. System displays a success message indicating that the task has been inserted.

Alternative Flows:

2A1: If user doesn’t fill the any of the required fields, system display displays an error message.

Use case continues on 1.

2A2: If user doesn’t fill the required fields with valid value, system display displays an error message.

Use case continues on 1.

**Use Case 2: Update a Task**

Actor: User

Description: This use case allows the user to update an existing task in the Task Planner application.

Preconditions: User is logged into the Task Planner application and there is at least one task in the task database.

Postconditions: The selected task is successfully updated with the new information.

Basic Flow:

1. System displays a list of existing tasks.

2. User selects a task from the list to update.

3. System presents the current task details to the user for editing.

4. User modifies the task details (name, description, deadline, priority, or reminder image).

5. User confirms the updated task details.

6. User clicks on the “Update a Task” button.

7. System saves the changes to the task in the task database.

8. System displays a success message indicating that the task has been updated.

Alternative Flows:

1A1: If there is no valid task in the table system can’t show anything for updating.

Use case ends.

6A1: If the user stops the task editing, the system returns to the main screen without making any updating.

Use case ends.

**Use Case 3: Delete a Task**

Actor: User

Description: This use case allows the user to delete an existing task from the Task Planner application.

Preconditions: User is logged into the Task Planner application and there is at least one task in the task database.

Postconditions: The selected task is successfully deleted from the task database.

Basic Flow:

1. System displays a list of existing tasks.

2. User selects a task from the list to delete.

3. System presents the selected task details for confirmation.

4. User confirms the deletion of the task.

5. System removes the task from the task database.

6. System displays a success message indicating that the task has been deleted.

Alternative Flows:

1A1: If there is no valid task in the table system can’t show anything for updating.

The use case ends.

5a. User cancels the task deletion:

The use case ends.

**BASIC USE CASES**

**Use Case: Show all Tasks**

Actor: User

Description: This use case allows the user to view all the tasks in the Task Planner application.

Preconditions: User is logged into the Task Planner application and there is at least one task in the task database.

Postconditions: User can see a table displaying all the tasks' information.

**Use Case: View a Task's Image**

Actor: User

Description: This use case allows the user to view the image associated with a specific task.

Preconditions: User is logged into the Task Planner application and there is at least one task with an associated image.

Postconditions: User can view the image related to the selected task.

**Use Case: Reset**

Actor: User

Description: This use case allows the user to reset the occupied places.

Preconditions: User is logged into the Task Planner application.

Postconditions: The application is reset to its initial state.

**Use Case: Exit**

Actor: User

Description: This use case allows the user to exit the Task Planner application.

Preconditions: User is logged into the Task Planner application.

Postconditions: The user exits the application and is no longer logged in.

**Use Case: Filter Tasks**

Actor: User

Description: This use case allows the user to filter the tasks based on a specified date range.

Preconditions: User is logged into the Task Planner application and there is at least one task in the task database.

Postconditions: User can see a filtered list of tasks based on the specified date range.

**Use Case: Sort by Priority**

Actor: User

Description: This use case allows the user to prioritize tasks based on a specific date.

Preconditions: User is logged into the Task Planner application and there is at least one task in the task database.

Postconditions: User can see the tasks sorted by priority.

## **3.4 Relationships among Actors and Use Cases**

## Log in: The user logs into the application using a valid username and password.

## Insert a task: The user can create a new task by providing the necessary information, such as task name, description, deadline and reminder image. There is also priority information. User can fill it if he/she want.

## Update a task: The user can update an existing task by modifying its information.

## Delete a task: The user can delete a task from the system.

## Show all tasks: The user can view all the tasks stored in the system.

## View a task's image: The user can view the image associated with a particular task.

## Filter tasks: The user can filter the tasks based on a specified date range.

## Sort by priority: The user can sort the tasks based on their priority.

## Reset: The user can clear the filled fields with the reset button.

## Exit: The User can log out of the application.

## **3.5 Use Case Diagram**

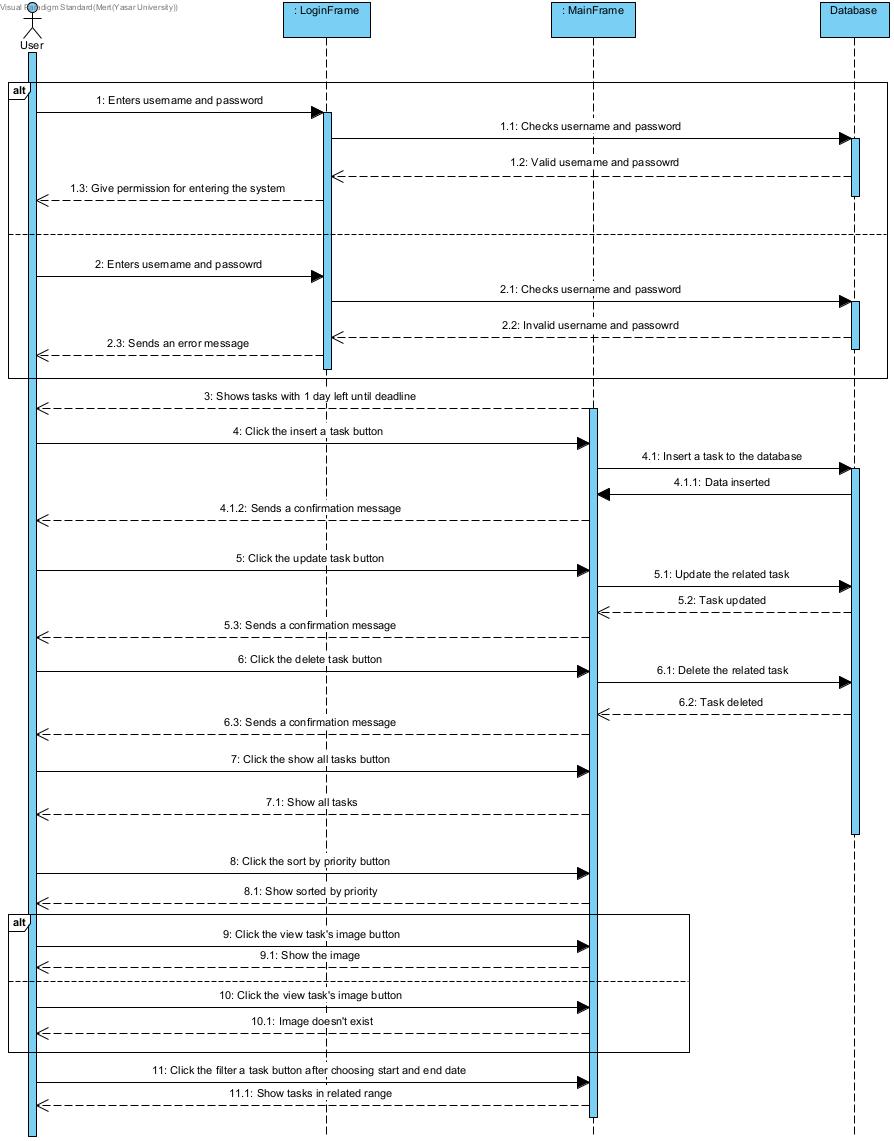
In this use case diagram, user connected to Insert a task, Edit & Update a task, Delete a task, Show all task, Sort task by priority, View a task’s image and Filter a Task but the user must be logged in before to use them. Also there is a notification, if there is a task with 1 day left until the deadline, its notification is displayed.

# 

# **4 Behavioral Models**

## **4.1 Sequence Diagram**

In this sequence diagram, there are two loop. First one is about login. If user enters his/her username or password wrong, system sends a error message. Second one is about view a task’s image. If the task doesn’t have a image. System sends a error message again. The rest of the diagram is Deadline notification, Insert a task, Update a task, Delete a task, Show all task, Sort by priority and Filter a task.



# 

# **5 Structural Models**

## **5.1 Class Diagram**

In this class diagram, there is a login class for opening the LoginFrame class. LoginFrame’s purpose is checking the information entered by the user from the database and providing the user's login. If the user logs in successfully then MainFrame appears. In main frame, There are all of the functions required to operate the Task Planer app with all of its capabilities.

**metin, yazı tipi, çizgi, ekran görüntüsü içeren bir resim

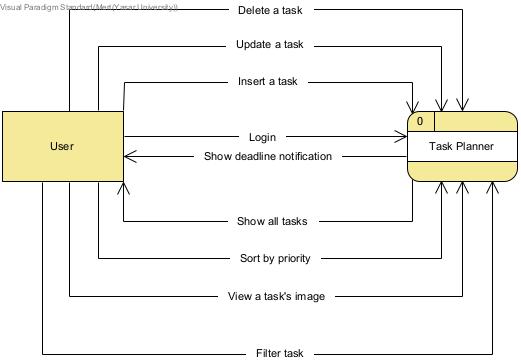
Açıklama otomatik olarak oluşturuldu**

# **6 Process Modeling**

No explanation is needed here. Only complete the subsection.

## **6.1 Data Flow Diagram (DFD)**

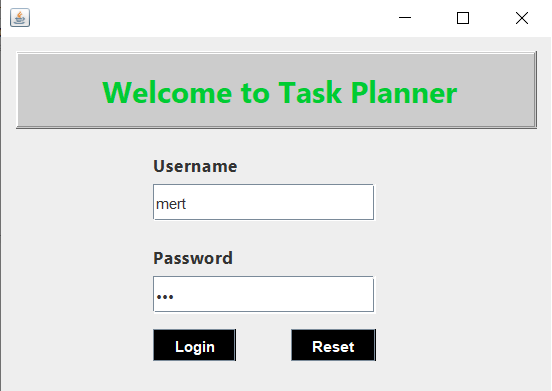
This is the data flow context diagram for the Task Planner. All potential interactions between the task planer app and the user are displayed here.



This is the level 0 of the data flow diagram for the Task Planner. There are 8 processes, 2 databases and 1 entity. Delete a task, Insert a task, Update a task and connected to Information database. Login connected to login database.

# **7 Final Version(s) of the Graphical User Interface(s)**

Put the images of your GUIs, also put the output screenshots of each function that is tested with sample data. Briefly explain the purpose of each GUI. Do not forget to add the form title for each image.



**When we click on the application, the login screen first welcomes us. In this screen user enters his/her username and password and clicks on the Login button. When user clicks on the login button system will check the username and password from database. Reset button is used to reset the information in the spaces.**

metin, ekran görüntüsü, yazılım, bilgisayar simgesi içeren bir resim

Açıklama otomatik olarak oluşturuldu

**If user enters wrong username or password, system shows a error message. User can enter his/her username and password after clicking the “OK” button.**

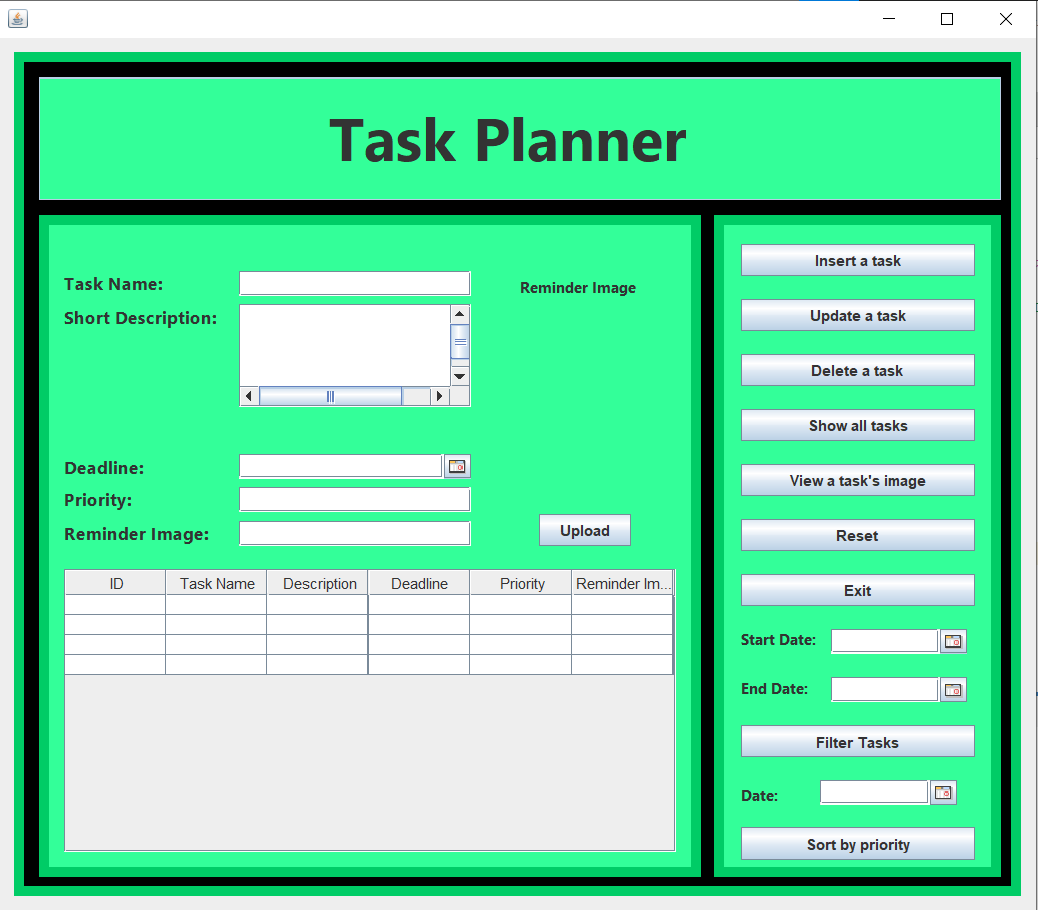
**After the login successfully, system shows a notification about deadline. If any, it reminds the user of the tasks that are 1 day left to the deadline.**

metin, ekran görüntüsü, yazılım, ekran, görüntüleme içeren bir resim

Açıklama otomatik olarak oluşturuldumetin, ekran görüntüsü, ekran, görüntüleme, yazılım içeren bir resim

Açıklama otomatik olarak oluşturuldu

**This is the main frame. User can see the all of the functionalities and necessary fields in this page.**

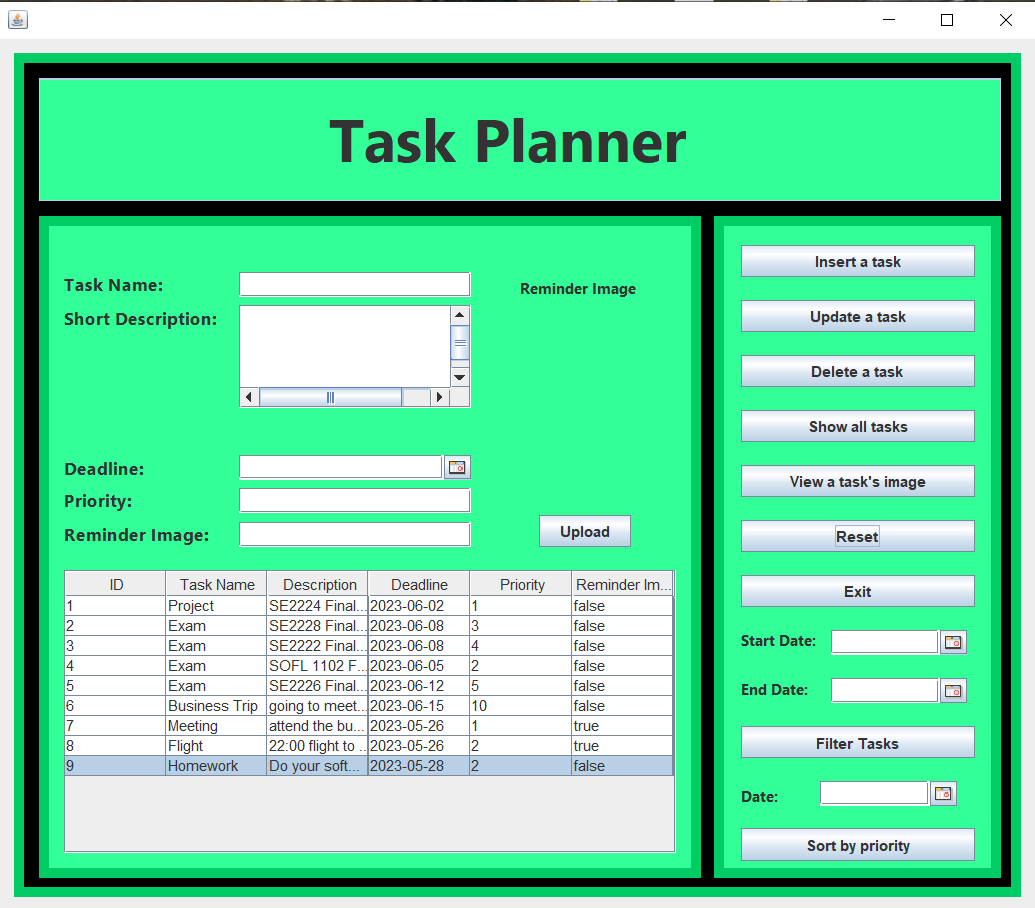


**This is the “Insert a Task” part. When user clicks on the “Insert a Task” button after fill the necessary fields system shows a confirmation message.**

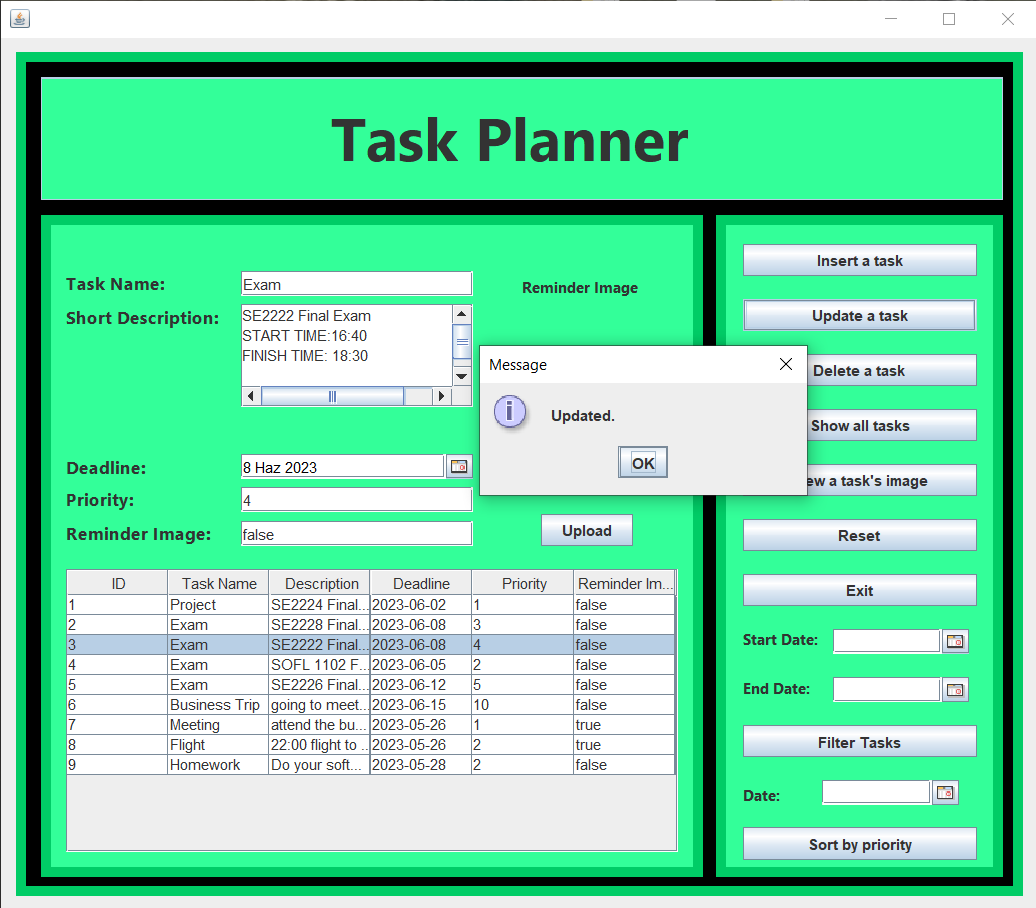
metin, ekran görüntüsü, yazılım, ekran, görüntüleme içeren bir resim

Açıklama otomatik olarak oluşturuldu

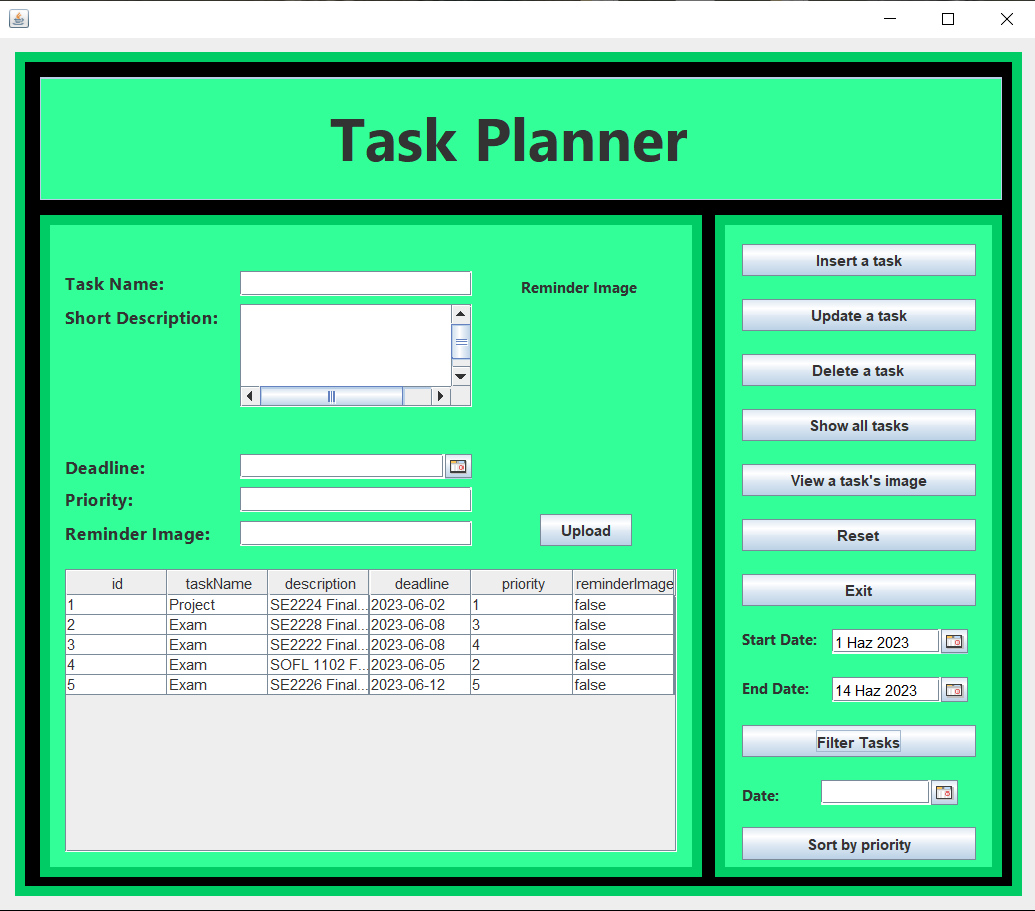
**This is the main page after using the “Insert a task” button. Also user can see the table with using “Show all tasks” button.**



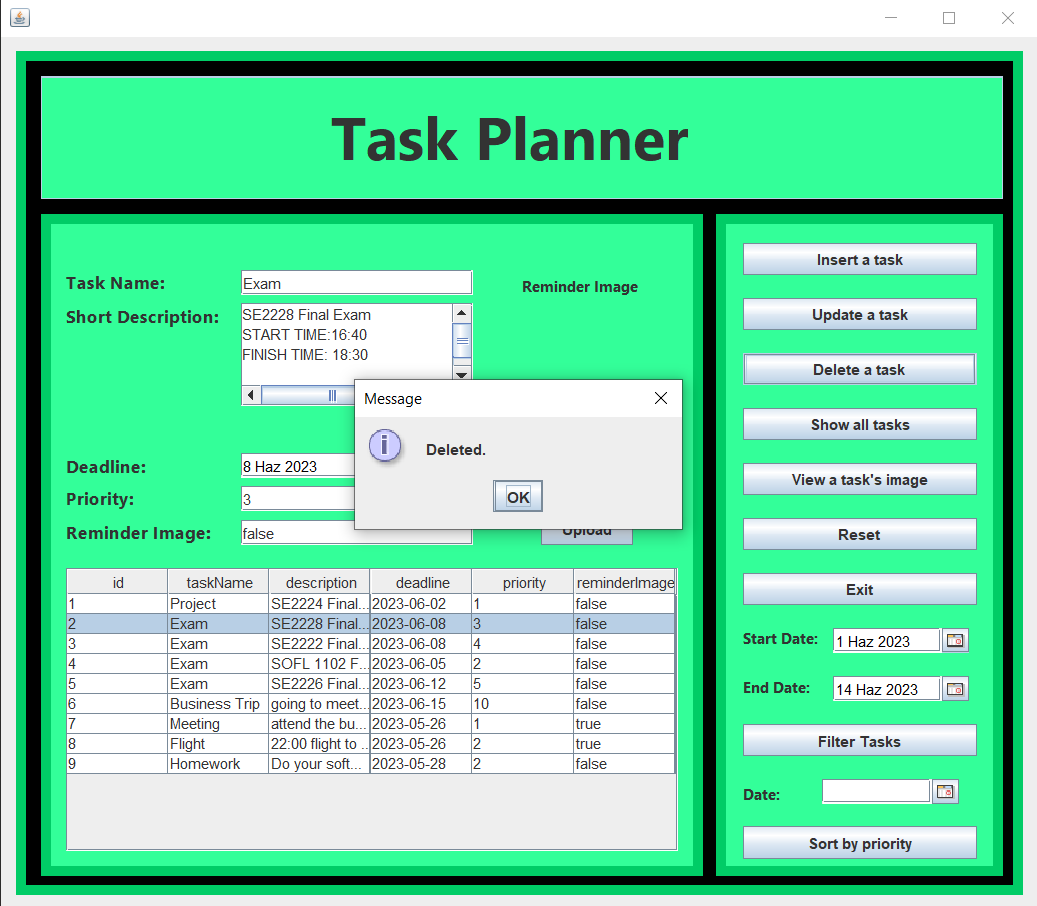
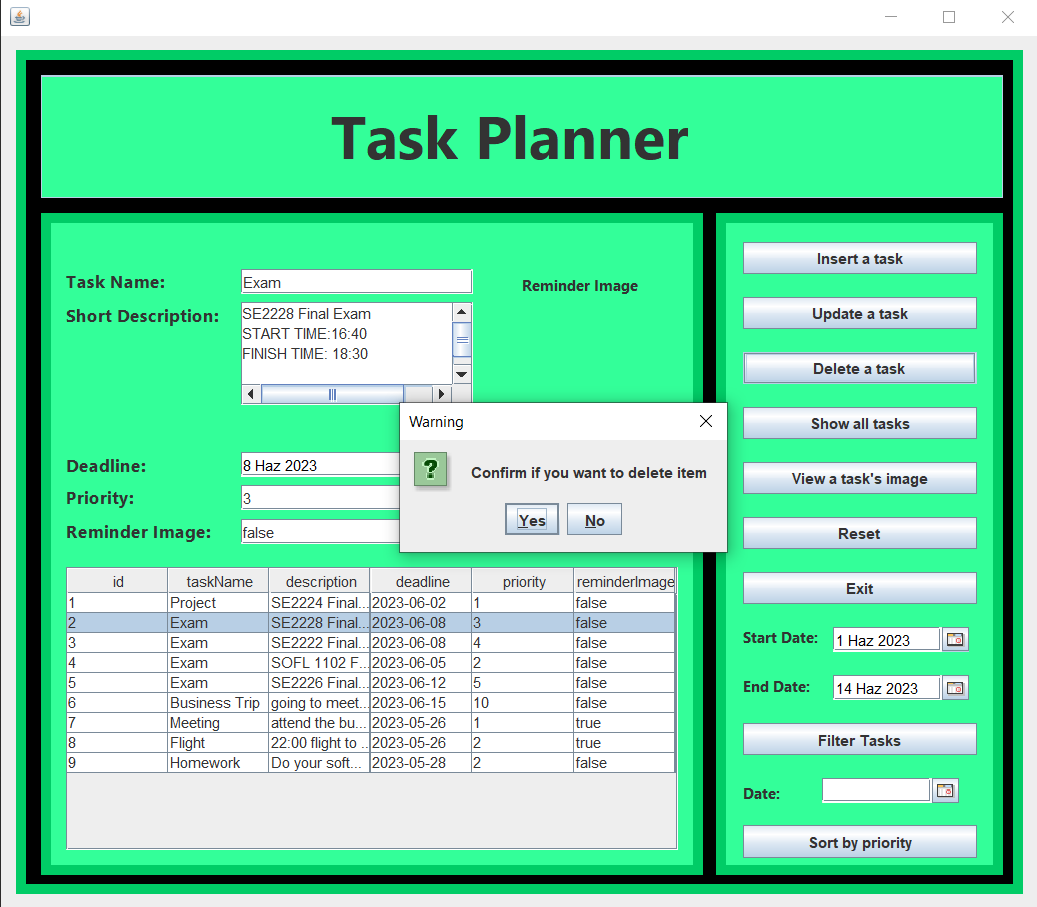
**This is the “Update a Task” part. Firstly user must choose a task from the table and user should edit the desired information. After editing when user clicks on the “Update a task” button task will be arranged and system shows a confirmation message.**



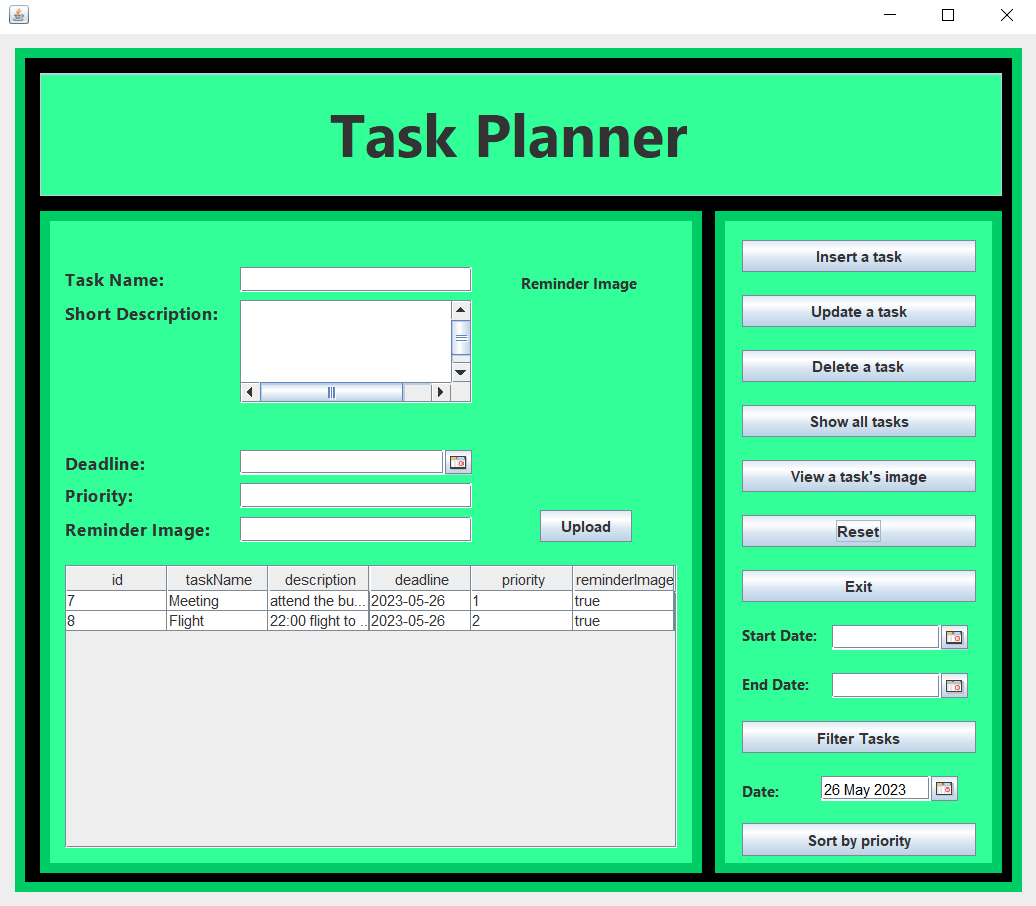
**This is the “Filter a Task” part. In this part user can filter the task by given dates. Firstly user must choose start and end date. If dates are exist then system will show the tasks in that range after clicking the “Filter Tasks” button.**



**This is the “Delete a Task” part. In this part user can delete a task from table. Firstly user must choose a task he/she want to delete from table. Then user must click the “Update a task” button. After clicking system shows a message with yes no option. If user click the “Yes” button, System deletes the task from table and shows a confirmation message.**



**This is the “Sort by priority” part. In this part user can sort the tasks by priority. Before clicking the “Sort by priority” user must choose a exist date. After clicking system will show the table sorted by priority.**



**This is “View a task’s image” part. In this part if you want you can add reminder image to your task and you can see the image by using “View a task’s image” button.**

# **metin, ekran görüntüsü, ekran, görüntüleme, yazılım içeren bir resim Açıklama otomatik olarak oluşturuldu**

**User can upload an image by using “Upload” button and the user can see the uploaded image on the home screen.**

# 

# **8 Conclusion and Future Work**

In conclusion, the Task Planner software program offers users a convenient and efficient way to organize and manage their responsibilities and activities. With a user-friendly interface and a comprehensive set of features, users can easily create, update, and delete tasks, view task details and images, filter and sort tasks, and stay informed about upcoming deadlines through reminder notifications. The system also maintains a task database with unique IDs and entry dates, providing a reliable record of task management.

**Future Features:**

**Sign Up:** A button can be added to enable the user who logs into the application for the first time to register to the system.

**Task Reminders and Notifications:** Provide several notification channels, such as email, smartphone push notifications, or SMS alerts, to improve the reminder system. Users may set their own settings and receive timely notifications for upcoming deadlines or task modifications.

**Task Categories and Tags:** Implement the ability to categorize tasks using custom tags or predefined categories. This would provide users with a flexible way to organize and filter tasks based on specific criteria, such as project, department, or priority.

**Subtasks and Task Hierarchy:** Providing the ability to break down tasks into smaller subtasks and establishing a hierarchical structure, allowing for better organization and tracking of complex projects.

**Completed Task:** When user complete a task, select the task from the table and click the completed button, the task will be transferred to the completed tasks table and the user will be able to easily see the previously completed tasks from the task history.