CSE 5324 Software Engineering: Analysis, Design and Testing

Dr. Michael F. Siok, PE, ESEP

To-Do Planner

Iteration #2

- GROUP 7

GROUP MEMBERS:

- MANE, PARIDNYA SANJIV (1001863514)
- KALAPALA, YOGESH (1001879640)
- PARSHVA URMISH SHAH (1001838879)
- PATEL, PARTH BHANUPRASAD (1001720900)
- ABDULBARI SYED (1001995871)

04/06/2022

Agenda

- Project Description
- Requirements
- Overall Design Approach
- Use Cases
- Requirements Use Case Trace Matrix
- Increment Matrix
- Domain Model
- Expanded Use Case Diagrams
- Sequential Diagrams
- Domain Class Diagrams
- Code Snippets
- Summary

- Q & A

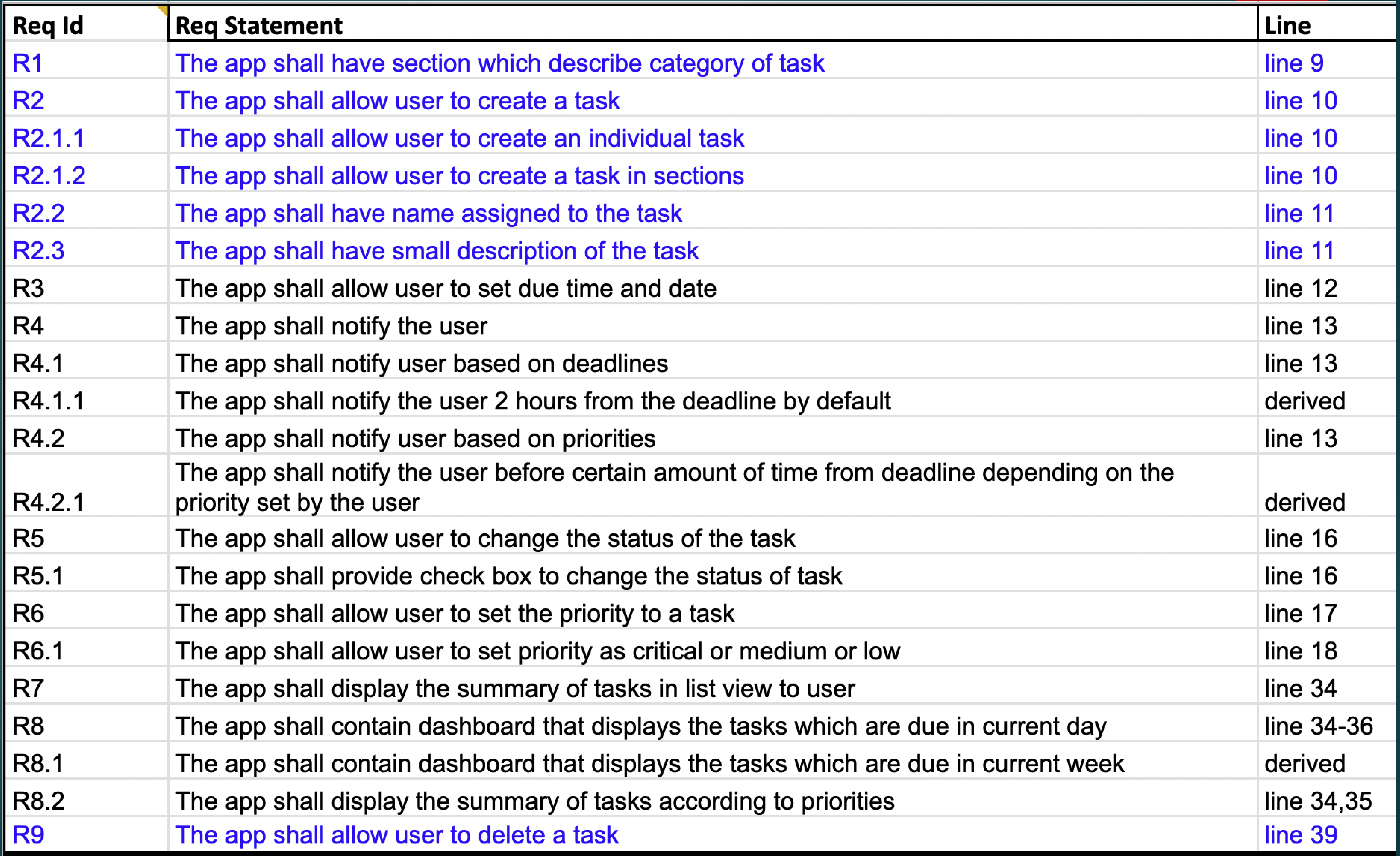
Project Description

- The app is used to schedule and organizes tasks
- This app aims to solve the problem of productivity.
- The app intends to do so by:

- Offering a platform to view summary of pending tasks

- Providing notification certain amount of time before it is due

Requirements



Req Id	Req Statement	Line
R1	The app shall have section which describe category of task	line 9
R2	The app shall allow user to create a task	line 10
R2.1.1	The app shall allow user to create an individual task	line 10
R2.1.2	The app shall allow user to create a task in sections	line 10
R2.2	The app shall have name assigned to the task	line 11
R2.3	The app shall have small description of the task	line 11
R3	The app shall allow user to set due time and date	line 12
R4	The app shall notify the user	line 13
R4.1	The app shall notify user based on deadlines	line 13
R4.1.1	The app shall notify the user 2 hours from the deadline by default	derived
R4.2	The app shall notify user based on priorities	line 13
R4.2.1	The app shall notify the user before certain amount of time from deadline depending on the priority set by the user	derived
R5	The app shall allow user to change the status of the task	line 16
R5.1	The app shall provide check box to change the status of task	line 16
R6	The app shall allow user to set the priority to a task	line 17
R6.1	The app shall allow user to set priority as critical or medium or low	line 18
R7	The app shall display the summary of tasks in list view to user	line 34
R8	The app shall contain dashboard that displays the tasks which are due in current day	line 34-36
R8.1	The app shall contain dashboard that displays the tasks which are due in current week	derived
R8.2	The app shall display the summary of tasks according to priorities	line 34,35
R9	The app shall allow user to delete a task	line 39

Overall Design Approach

Iteration # 1

Project Description

Requirements

Use Case Diagram

High Level Use Case

Requirements User Case Traceability Matrix

Iteration Matrix

Domain Model

Iteration # 2

Expanded Use Cases

Sequence Diagrams

Design Class Diagram

Partial Use case Implementation

Iteration # 3

Expanded Use Cases for remaining Use Cases

Sequence Diagram for Use Cases

Design Class Diagram for remaining Use Cases

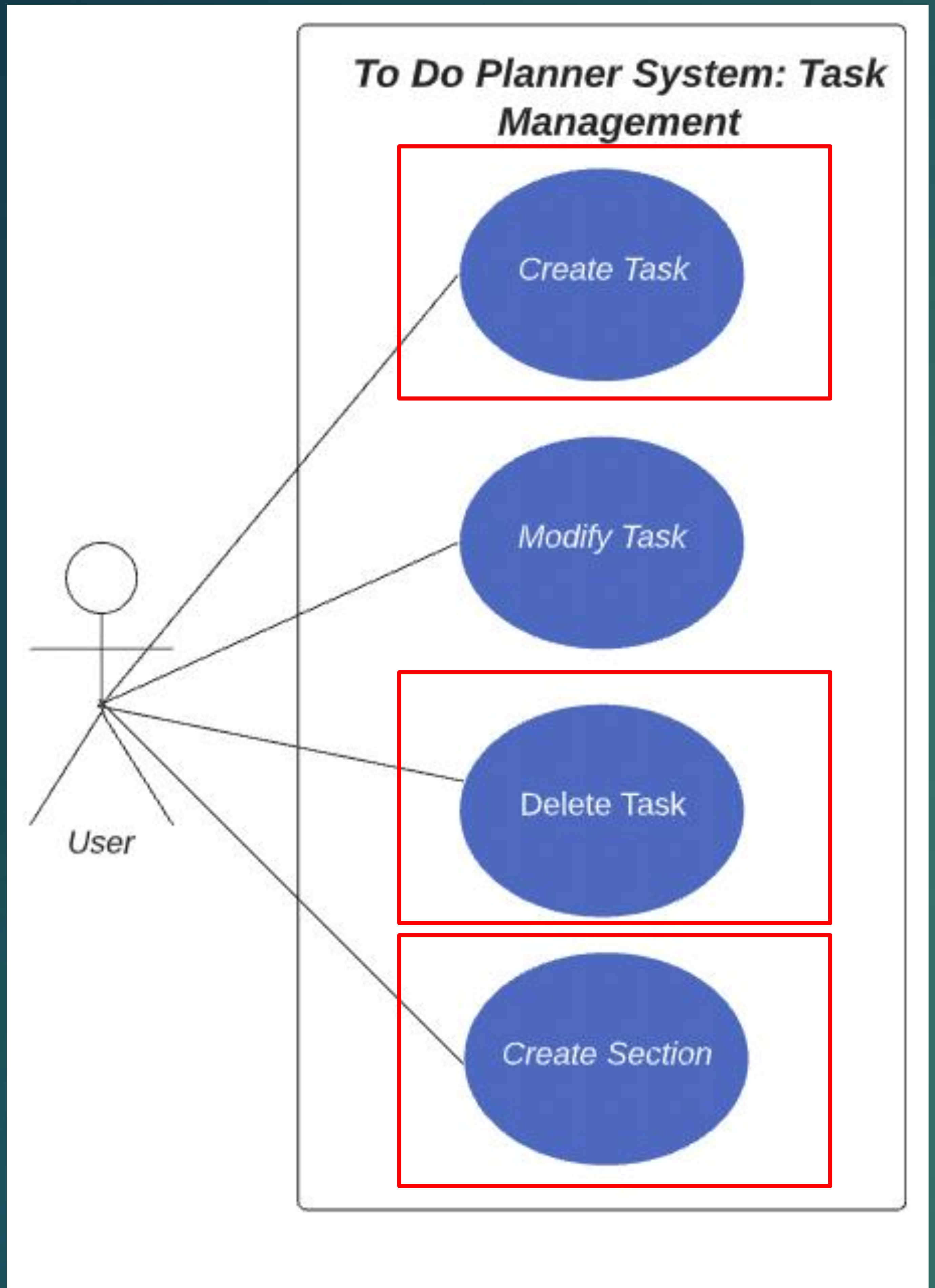
Complete Implementation

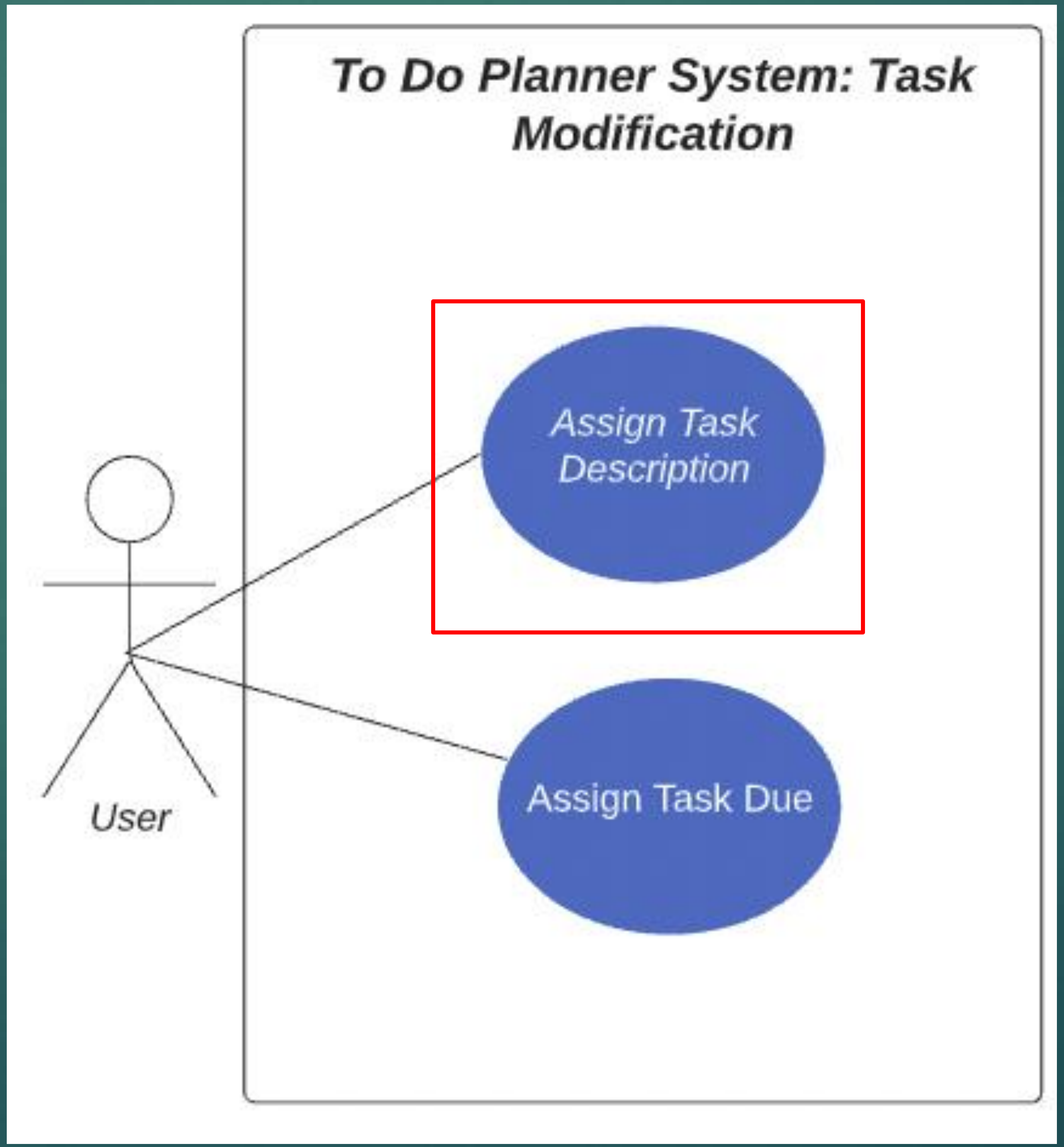
App Demo

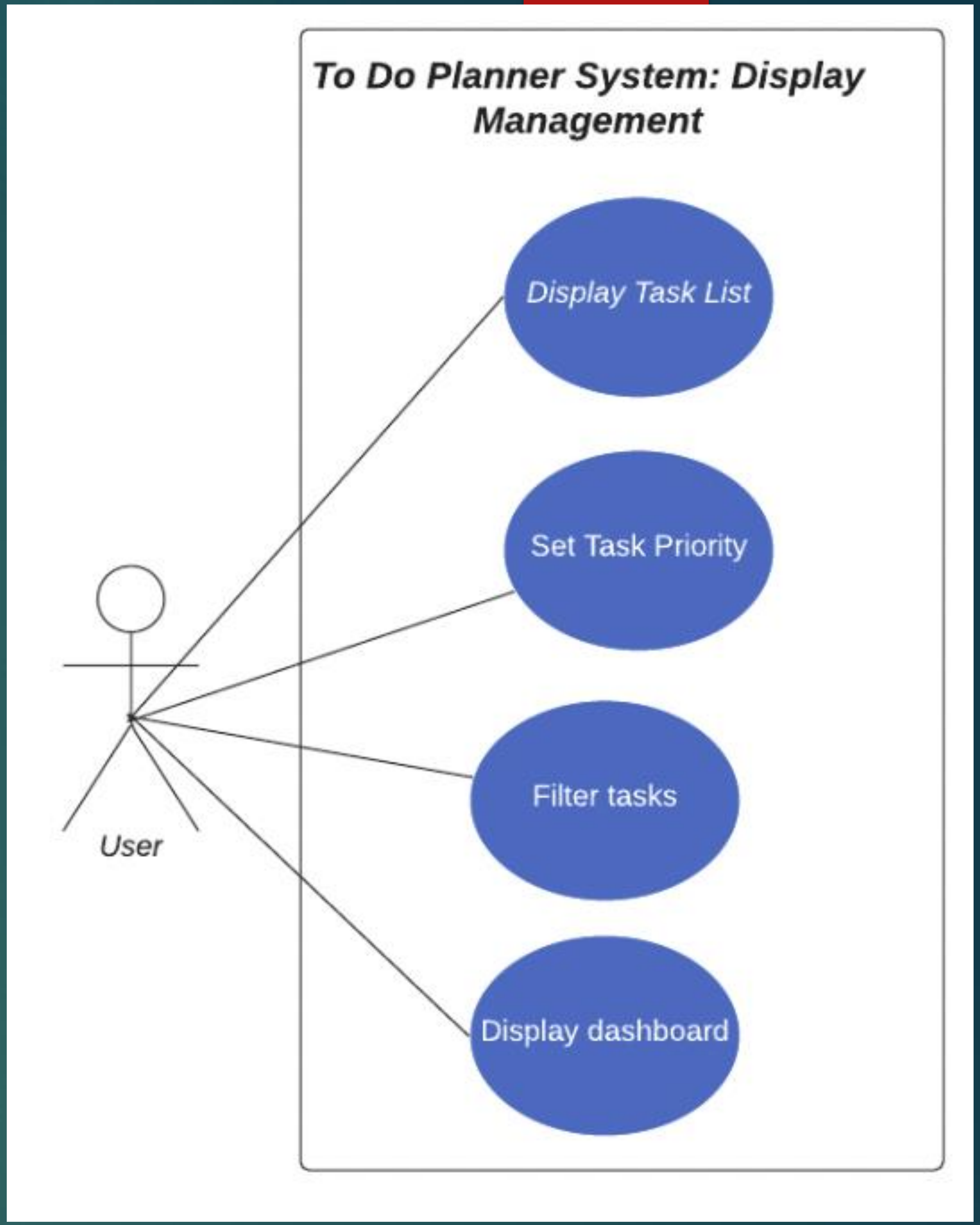
Use Cases

- Use Case 1: Create Task
- Use Case 2: Delete Task
- Use Case 3: Modify Task
- Use Case 4: Create Section
- Use Case 5: Assign Task Description
- Use Case 6: Assign due date
- Use Case 7: Set Task Priority
- Use Case 8: Display Task List
- Use Case 9: Display Dashboard
- Use Case 10: Filter Tasks

Use Case Diagram







High-Level Use Case

Use Case1: Create task

a. TUCBW the user clicking on “+” symbol
b. TUCEW the user sees a new task successfully added to the list of tasks

Use Case2: Delete Task

a. TUCBW the user clicks a delete button in task window
b. TUCEW the user sees task deleted from the list of tasks

Use Case3: Modify Task

a. TUCBW the users click on the edit button of task window
b. TUCEW the user changes the required parameter of the task

Use Case4: Create Section

a. TUCBW the user clicking on button
b. TUCEW the user sees a new section successfully added to the list

Use Case5: Assign Task Description

a. TUCBW the user select text box beside Task Description label
b. TUCEW the user successfully assigns a description to task

Use Case6: Assign Task Due

a. TUCBW the user select date picker beside Due Date label
b. TUCEW the user successfully assigns a deadline to task

Use Case7: Set Task Priority

a. TUCBW the user clicks priority from the drop-down menu
b. TUCEW the priority is set to the particular task

Use Case8: Display Task List

a. TUCBW the user clicks on the task icon at the bottom of the application.
b. TUCEW the application displaying a list view of tasks.

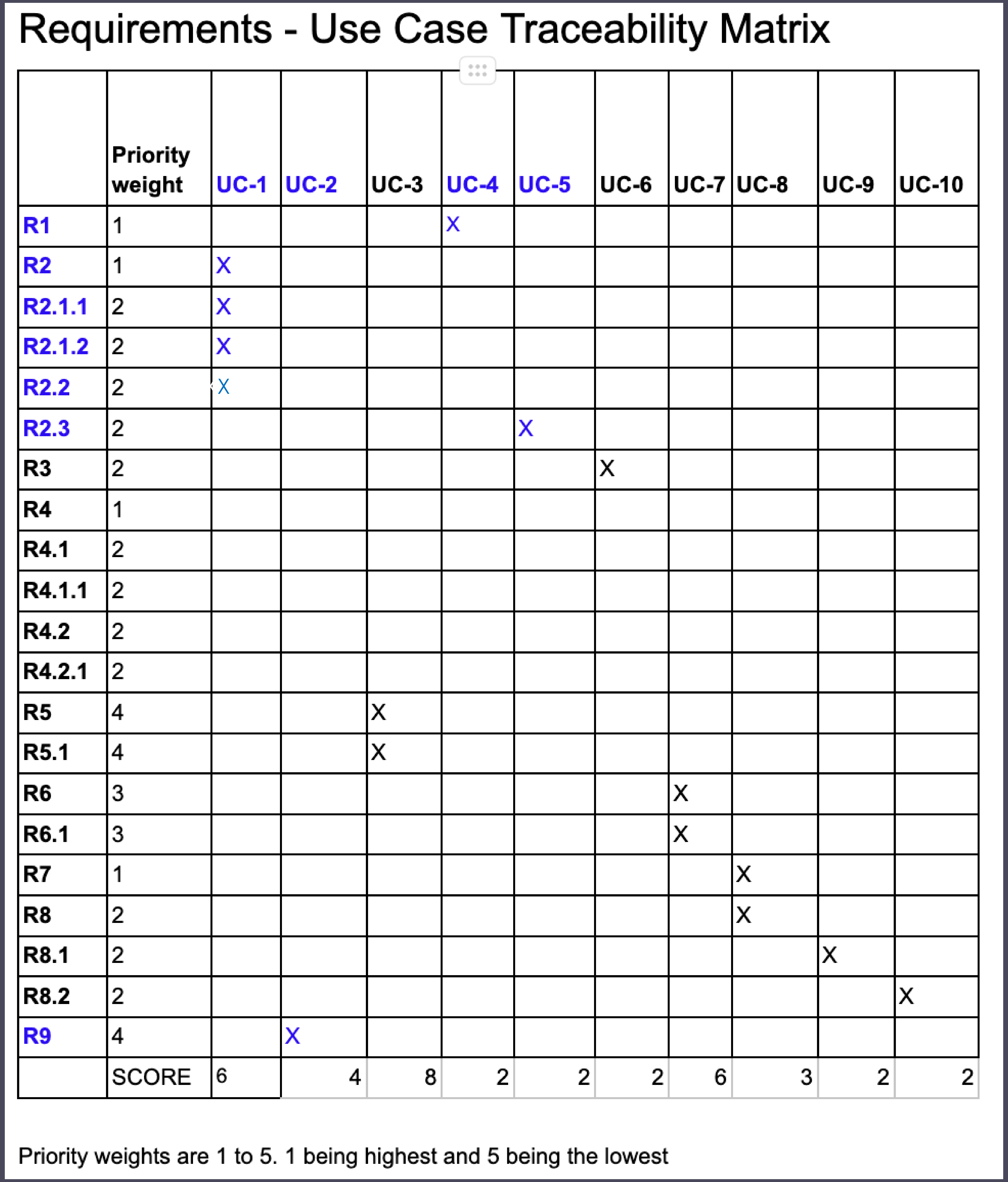
Use Case9: Display Dashboard

a. TUCBW the user clicks on the dashboard icon
b. TUCEW the application displays summary of tasks and tasks by due date are displayed in grid view.

Use Case10: Filter Tasks

a. TUCBW the user clicks on the filter menu
b. TUCEW application displays the list view of tasks based on a selected category of priorities.

Requirements-Use Case Trace Matrix

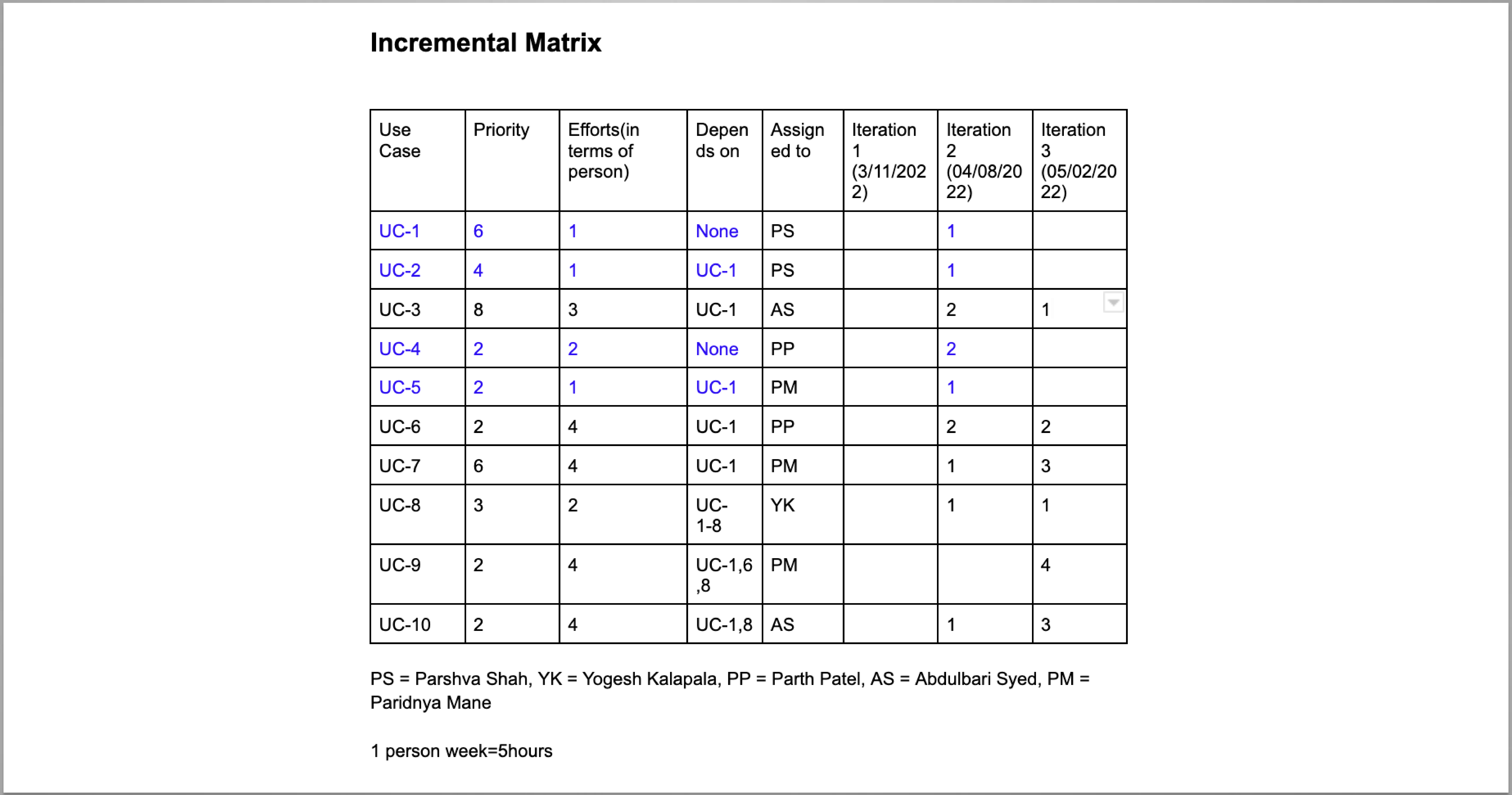


Requirements - Use Case Traceability Matrix

	Priority weight	UC-1	UC-2	UC-3	UC-4	UC-5	UC-6	UC-7	UC-8	UC-9	UC-10
R1	1				X						
R2	1	X									
R2.1.1	2	X									
R2.1.2	2	X									
R2.2	2	X									
R2.3	2					X					
R3	2						X				
R4	1										
R4.1	2										
R4.1.1	2										
R4.2	2										
R4.2.1	2										
R5	4			X							
R5.1	4			X							
R6	3							X			
R6.1	3							X			
R7	1								X		
R8	2								X		
R8.1	2									X	
R8.2	2										X
R9	4		X								
	SCORE	6	4	8	2	2	2	6	3	2	2

Priority weights are 1 to 5. 1 being highest and 5 being the lowest

Incremental Matrix



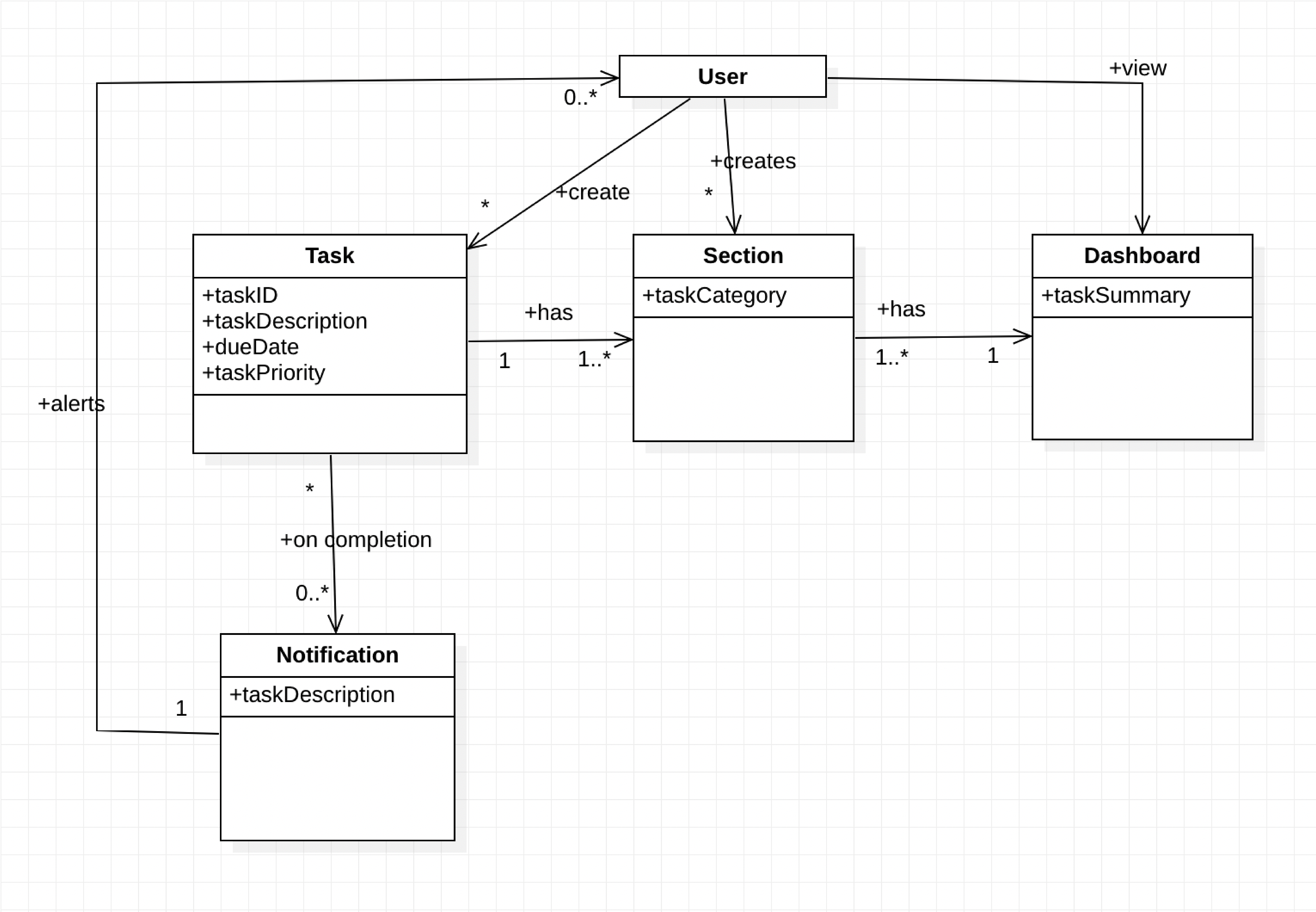
Incremental Matrix

Use Case	Priority	Efforts(in terms of person)	Depends on	Assigned to	Iteration 1 (3/11/2022)	Iteration 2 (04/08/2022)	Iteration 3 (05/02/2022)
UC-1	6	1	None	PS		1	
UC-2	4	1	UC-1	PS		1	
UC-3	8	3	UC-1	AS		2	1
UC-4	2	2	None	PP		2	
UC-5	2	1	UC-1	PM		1	
UC-6	2	4	UC-1	PP		2	2
UC-7	6	4	UC-1	PM		1	3
UC-8	3	2	UC-1-8	YK		1	1
UC-9	2	4	UC-1,6,8	PM			4
UC-10	2	4	UC-1,8	AS		1	3

PS = Parshva Shah, YK = Yogesh Kalapala, PP = Parth Patel, AS = Abdulbari Syed, PM = Paridnya Mane

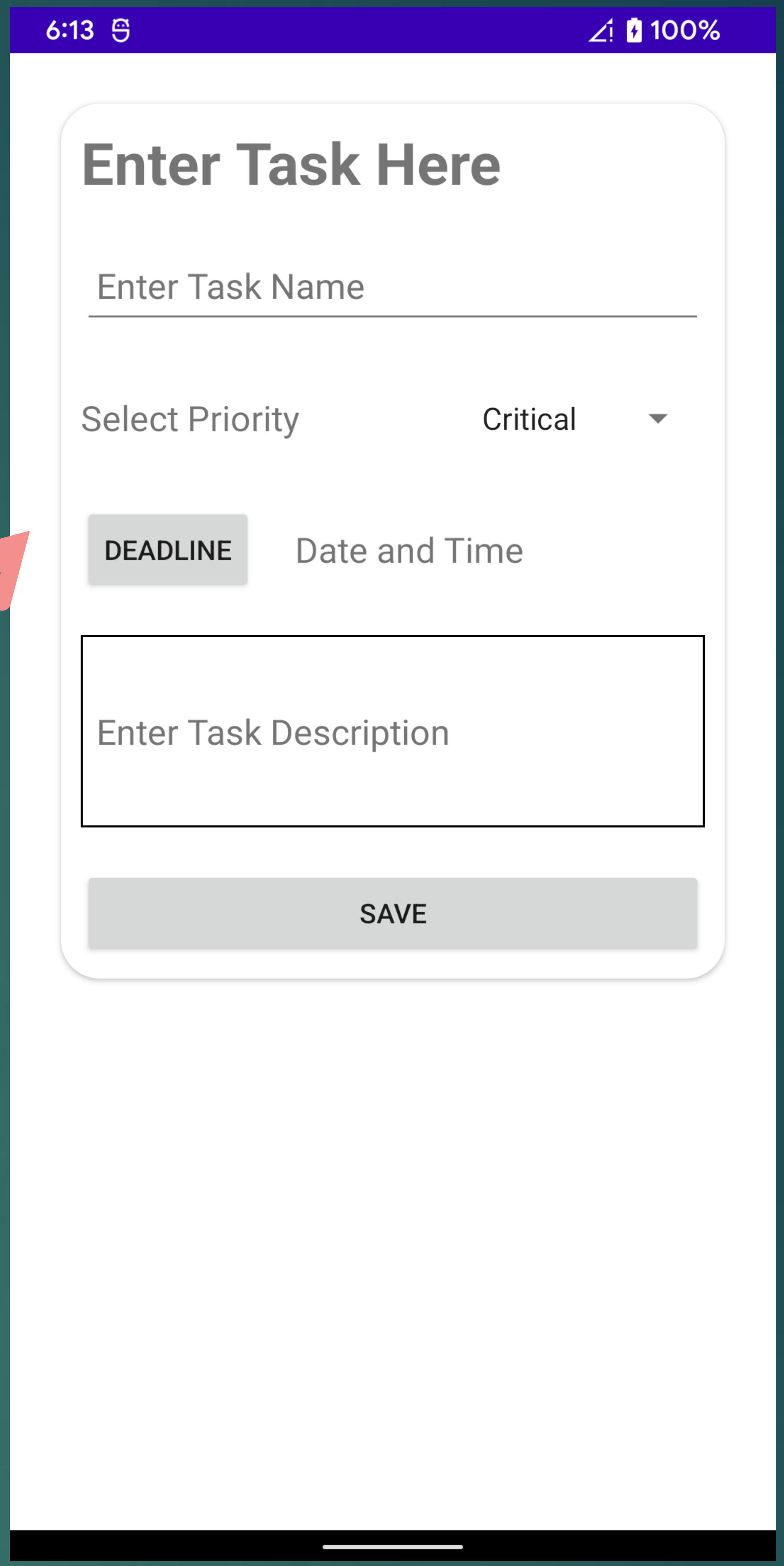
1 person week=5hours

Domain Model


User
+view
0..*
+creates
+create
*
*
Task
+taskID
+taskDescription
+dueDate
+taskPriority
Section
+taskCategory
Dashboard
+taskSummary
+has
1
1..*
+has
1..*
1
+alerts
*
+on completion
0..*
Notification
+taskDescription
1

Expanded Use Case Diagrams

EUC 1: Create Task	
Pre-Condition: This use case assumes that the user is in sections part of application.	
Actor: User	**System: To Do Planner**
	0. App displays section screen.
1. **TUCBW** the user clicking on "+" symbol.	2. App displays Add task screen.
3. The user enters details and clicks save button.	4*. App creates the task.
5. **TUCEW** the user sees a new task successfully added to the list of tasks.	
Postcondition: The created task is added to the task list.	


6:13 100%

Enter Task Here

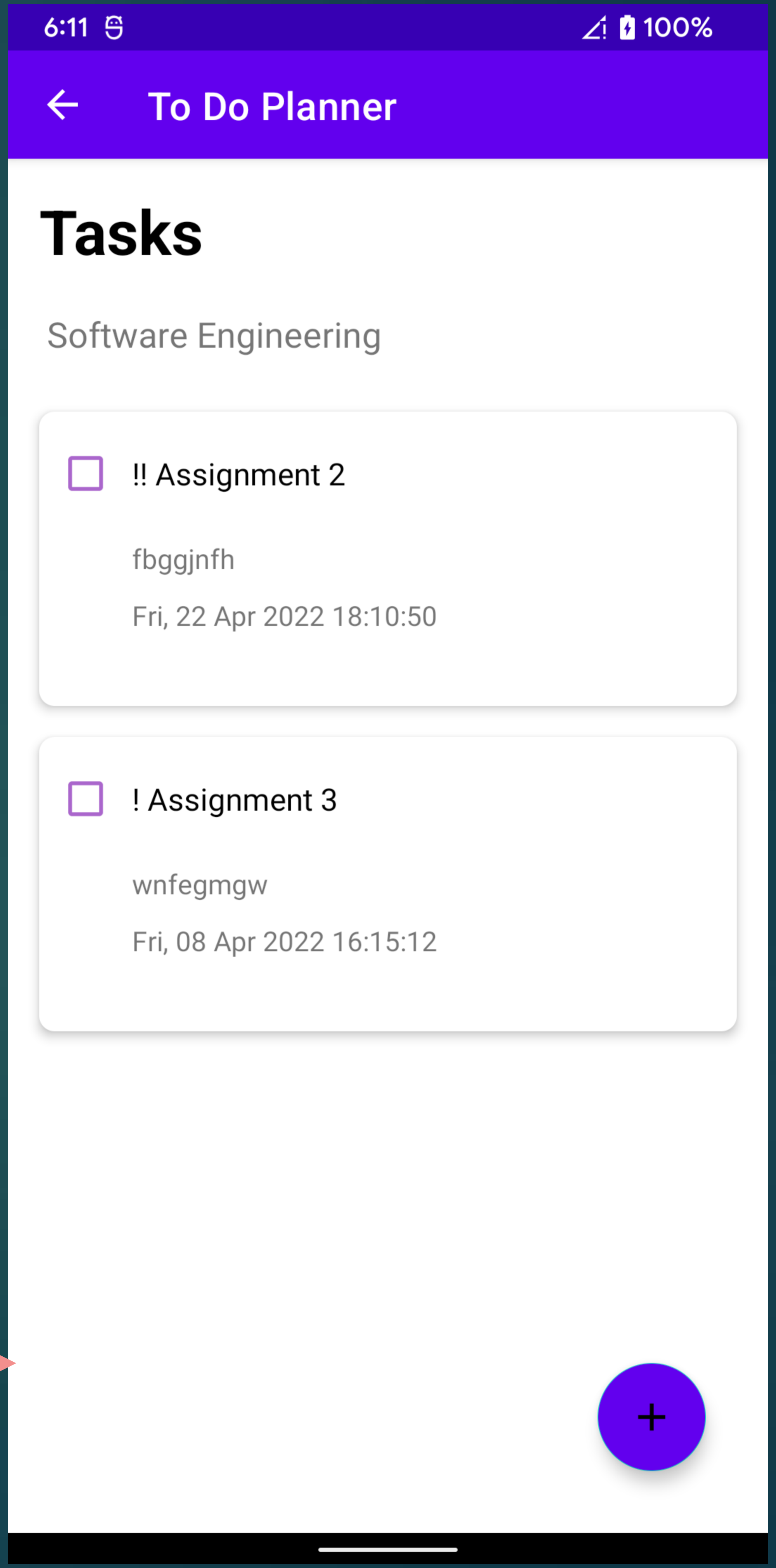
Enter Task Name

Select Priority Critical

DEADLINE Date and Time

Enter Task Description

SAVE


6:11 100%

To Do Planner

Tasks

Software Engineering

□ !! Assignment 2

fbggjnfh

Fri, 22 Apr 2022 18:10:50

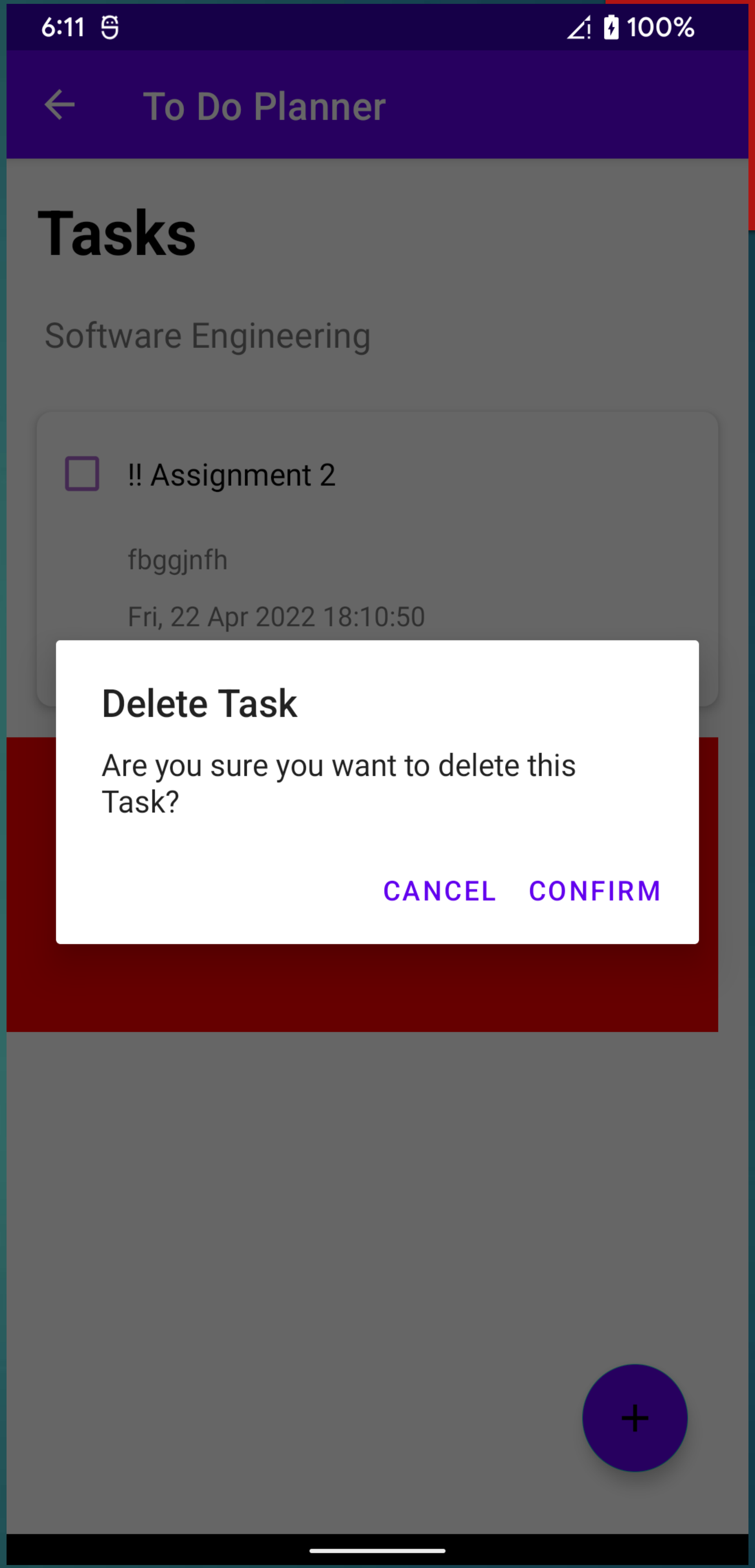
□ ! Assignment 3

wnfegmgw

Fri, 08 Apr 2022 16:15:12

+

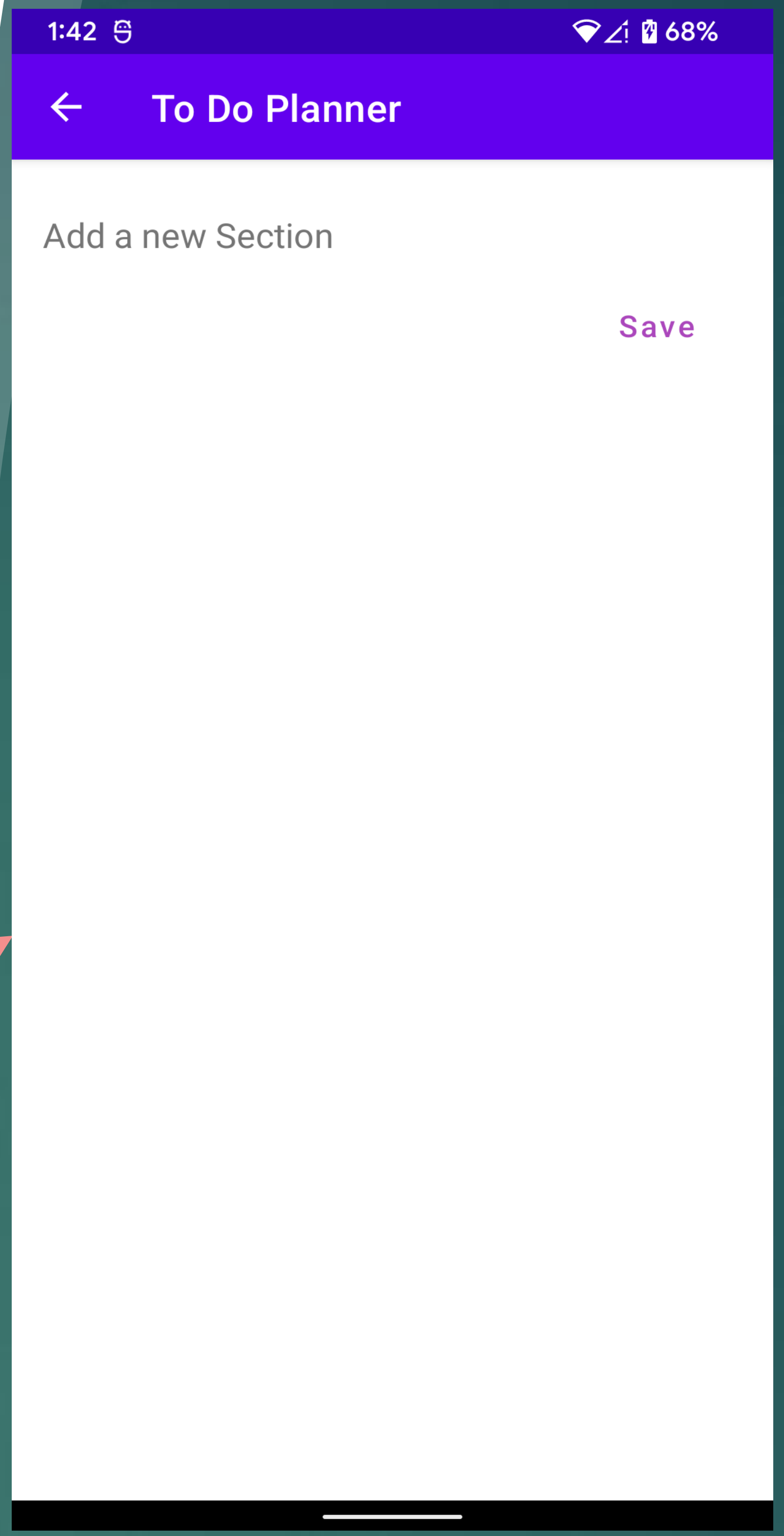
EUC 2: Delete Task	
Pre-Condition: This use case assumes that the user has created a task.	
Actor: User	**System: To Do Planner**
	0. App displays list of tasks on screen.
1. **TUCBW** the user swipes left to delete button in task window.	2. App displays Delete task window.
3. The user clicks confirm button.	4*. App deletes the task.
5. **TUCEW** the user sees task deleted from the list of tasks.	
Postcondition: The task is deleted from the task list.	



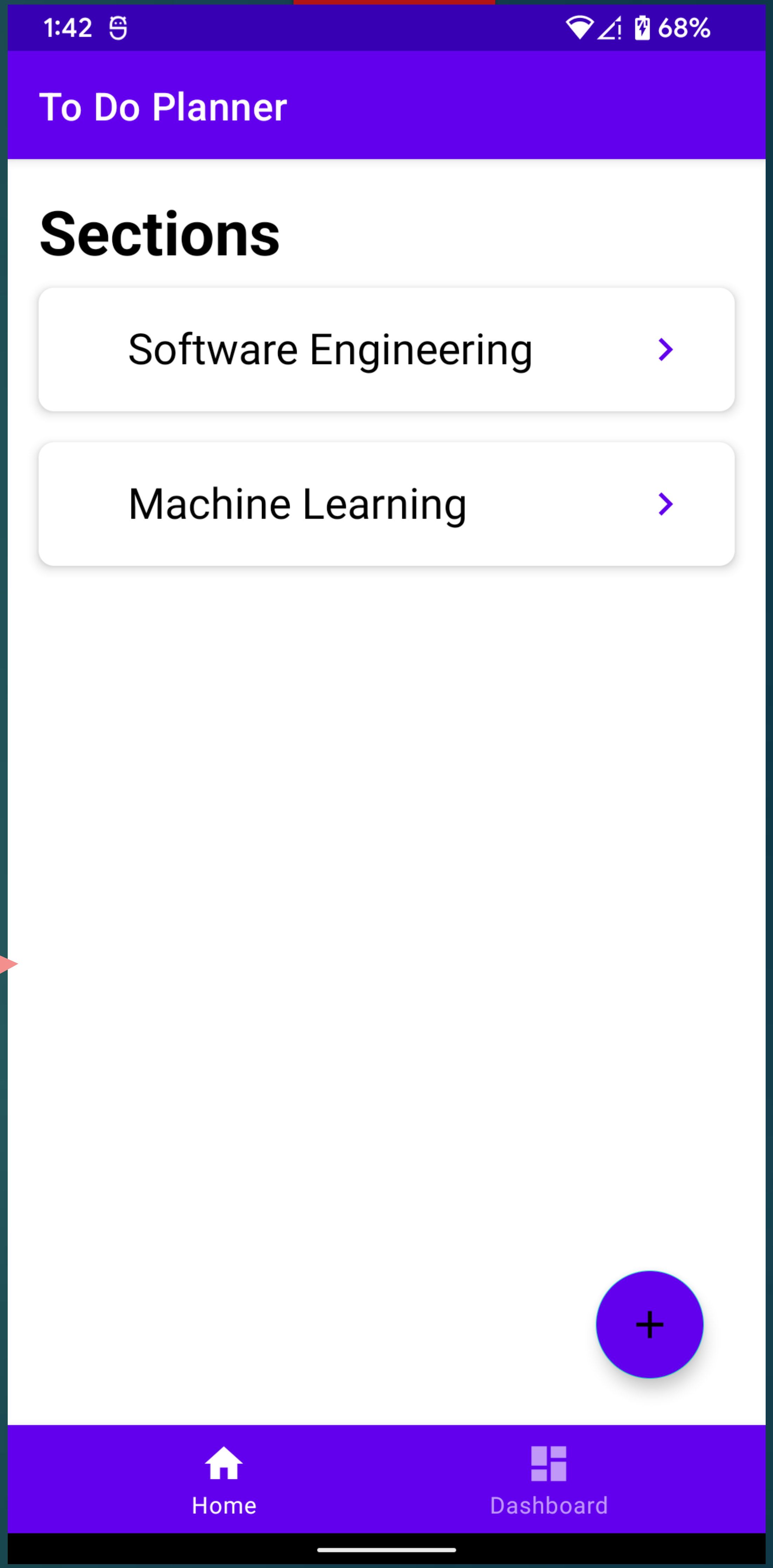
Delete Task

Are you sure you want to delete this Task?

EUC 4: Create Section	
Pre Condition: This use case assumes that the user is in sections part of application	
Actor: User	**System: To Do Planner**
	0. App displays launch screen.
1. **TUCBW** the user clicking on button.	2. App displays Add Section screen .
3. The user enters section name and clicks create button.	4*. App Creates Section.
5. **TUCEW** the user sees a new section successfully added to the list.	
Postcondition: The App creates a new section to list.	



Add a new Section



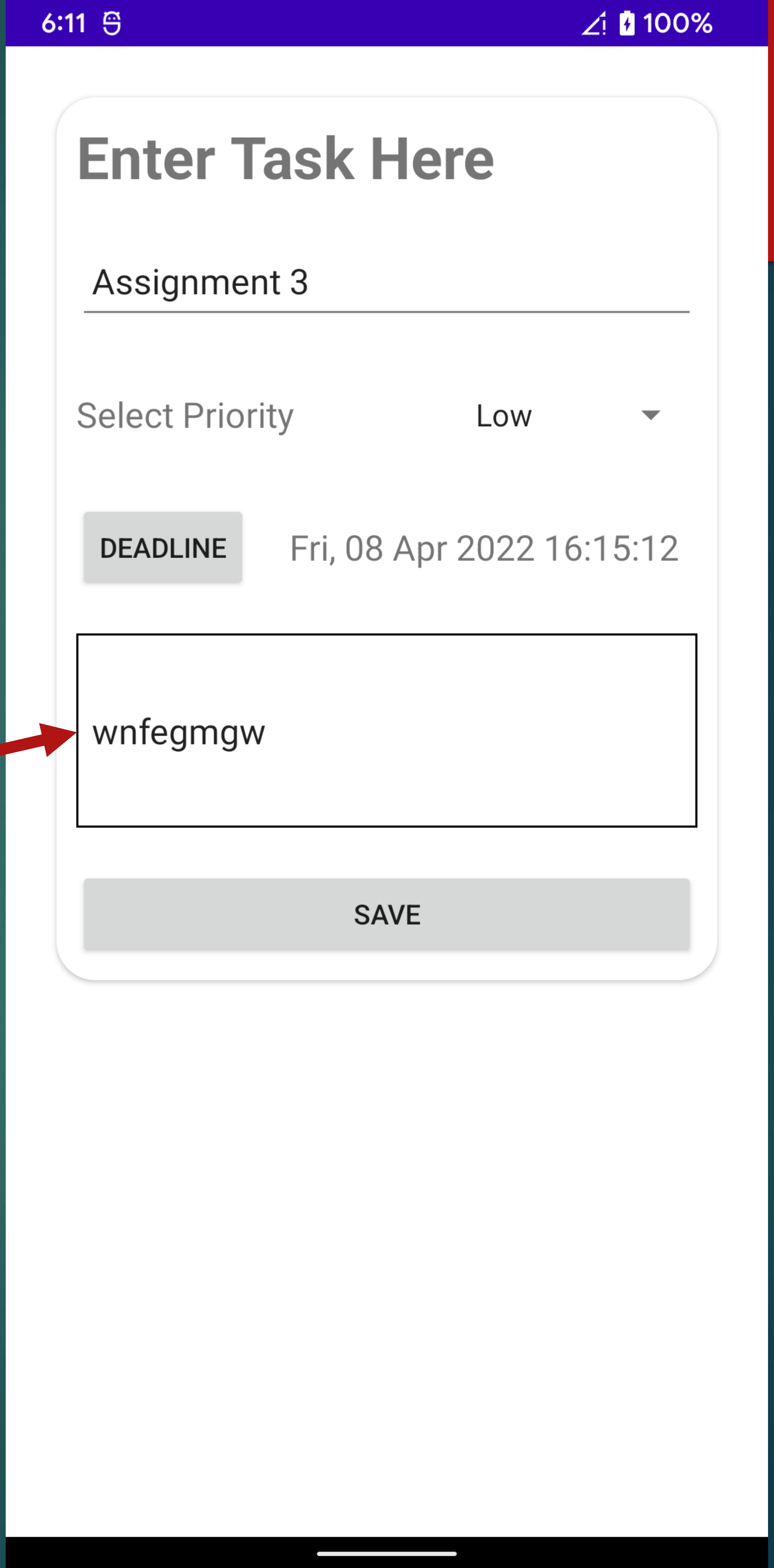
Sections

Software Engineering

Machine Learning

EUC 5: Assign Task Description(trivial)

Pre Condition: This use case assumes that the user is in create task window.	
Actor: User	System: To Do Planner
	0. App displays Add task screen.
1. **TUCBW** the user select text box beside Task Description label.	2. App displays task description parameters window.
3. The user enters task description and clicks create button.	4. App adds text in the task description.
5. **TUCEW** the user successfully assigns a description to task.	
Postcondition: The description is added to task.	



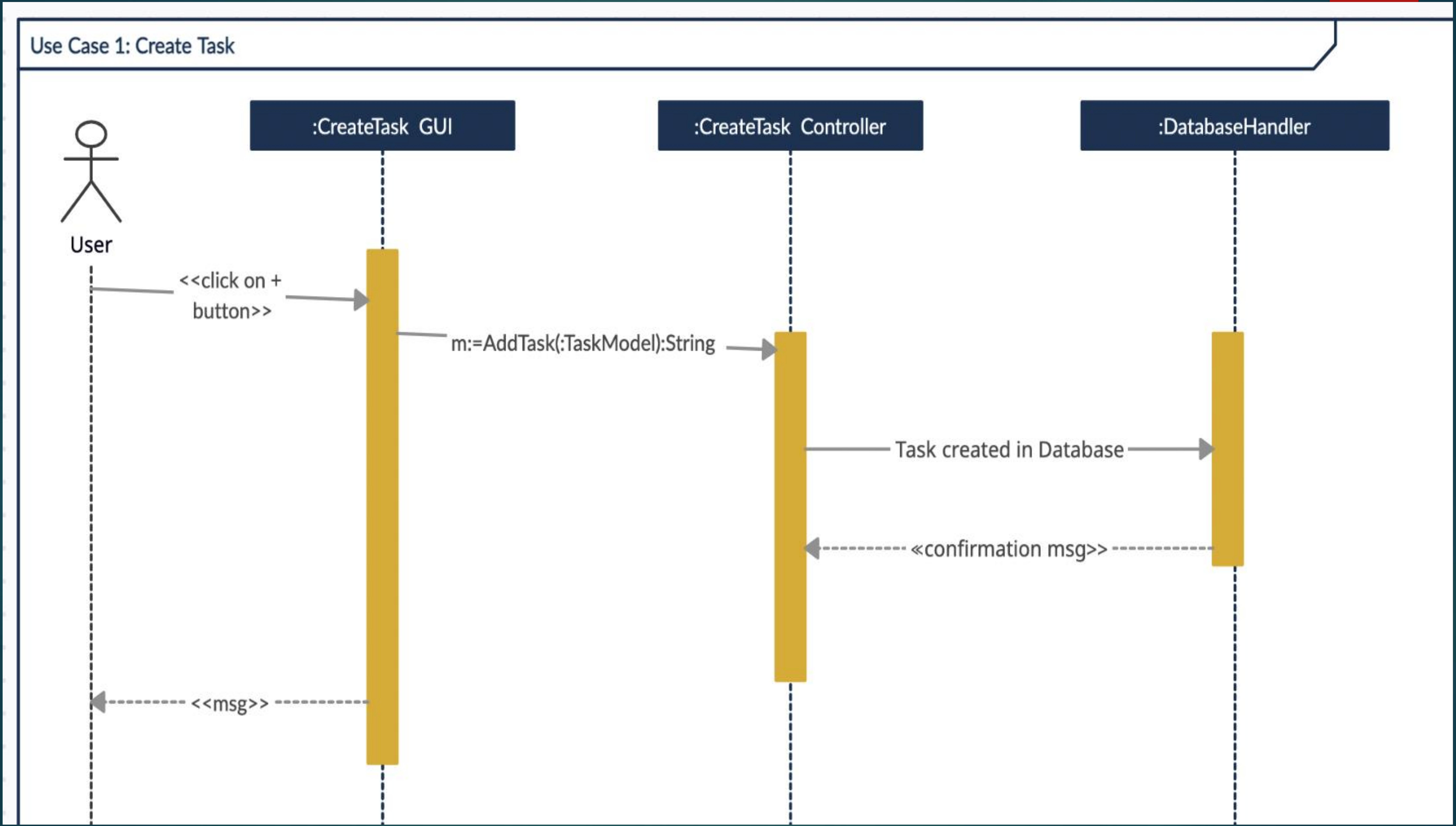
Assignment 3

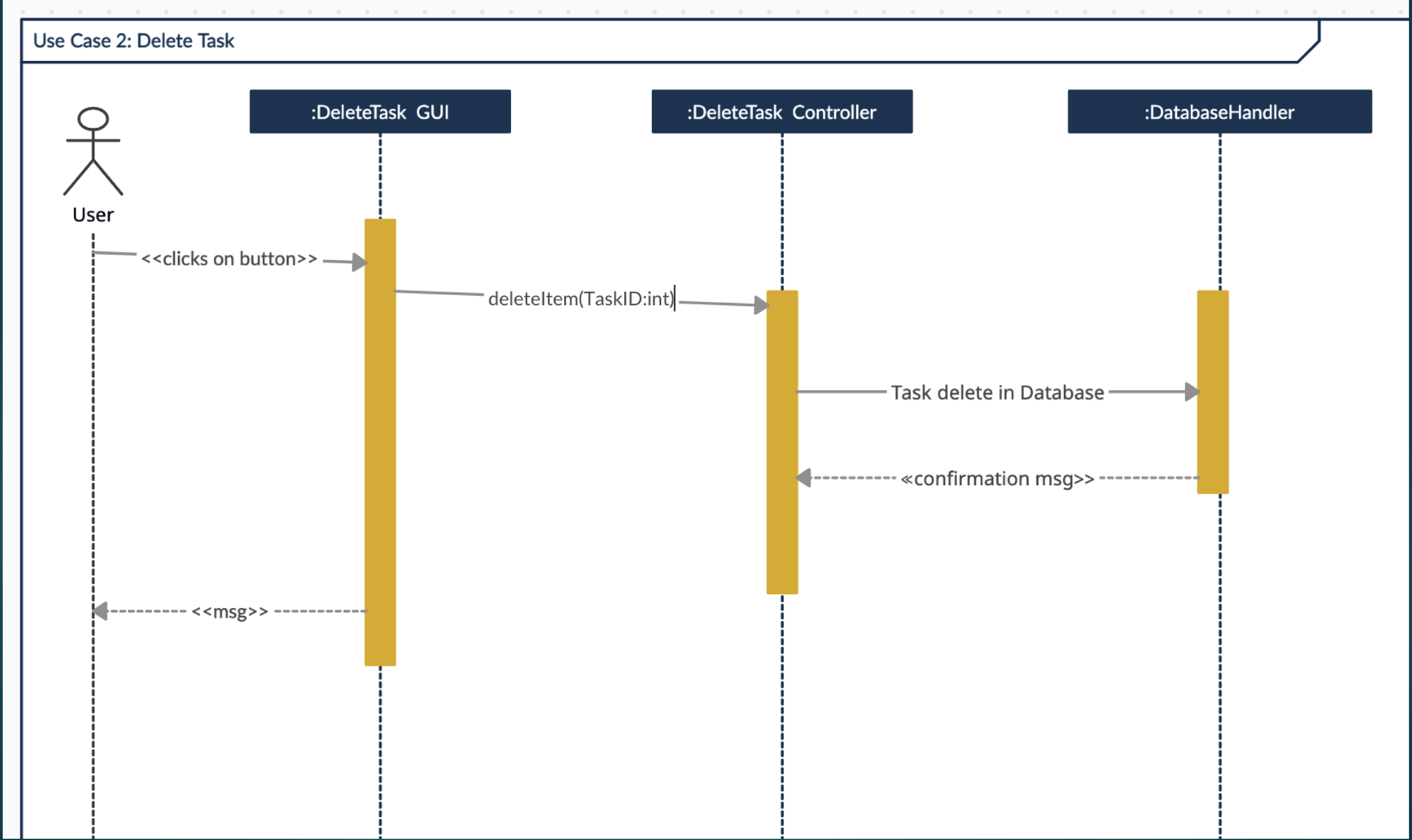
Select Priority Low

Fri, 08 Apr 2022 16:15:12

wnfegmgw

Sequence Diagrams


Use Case 1: Create Task
User
:CreateTask GUI
:CreateTask Controller
:DatabaseHandler
<<click on + button>>
m:=AddTask(:TaskModel):String
Task created in Database
«confirmation msg>>
<<msg>>



Use Case 2: Delete Task

User

:DeleteTask GUI

:DeleteTask Controller

:DatabaseHandler

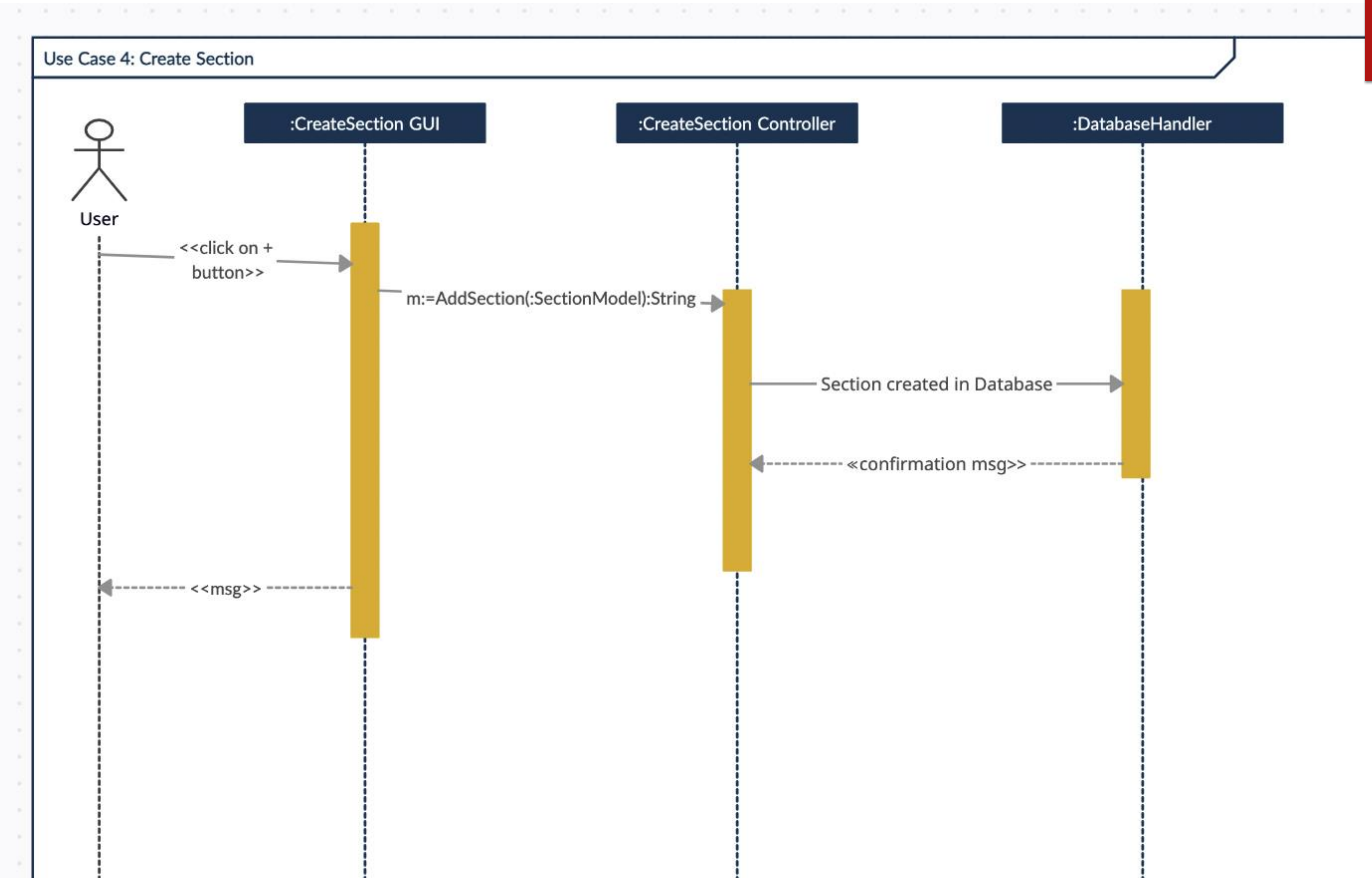
<<clicks on button>>

deleteItem(TaskID:int)

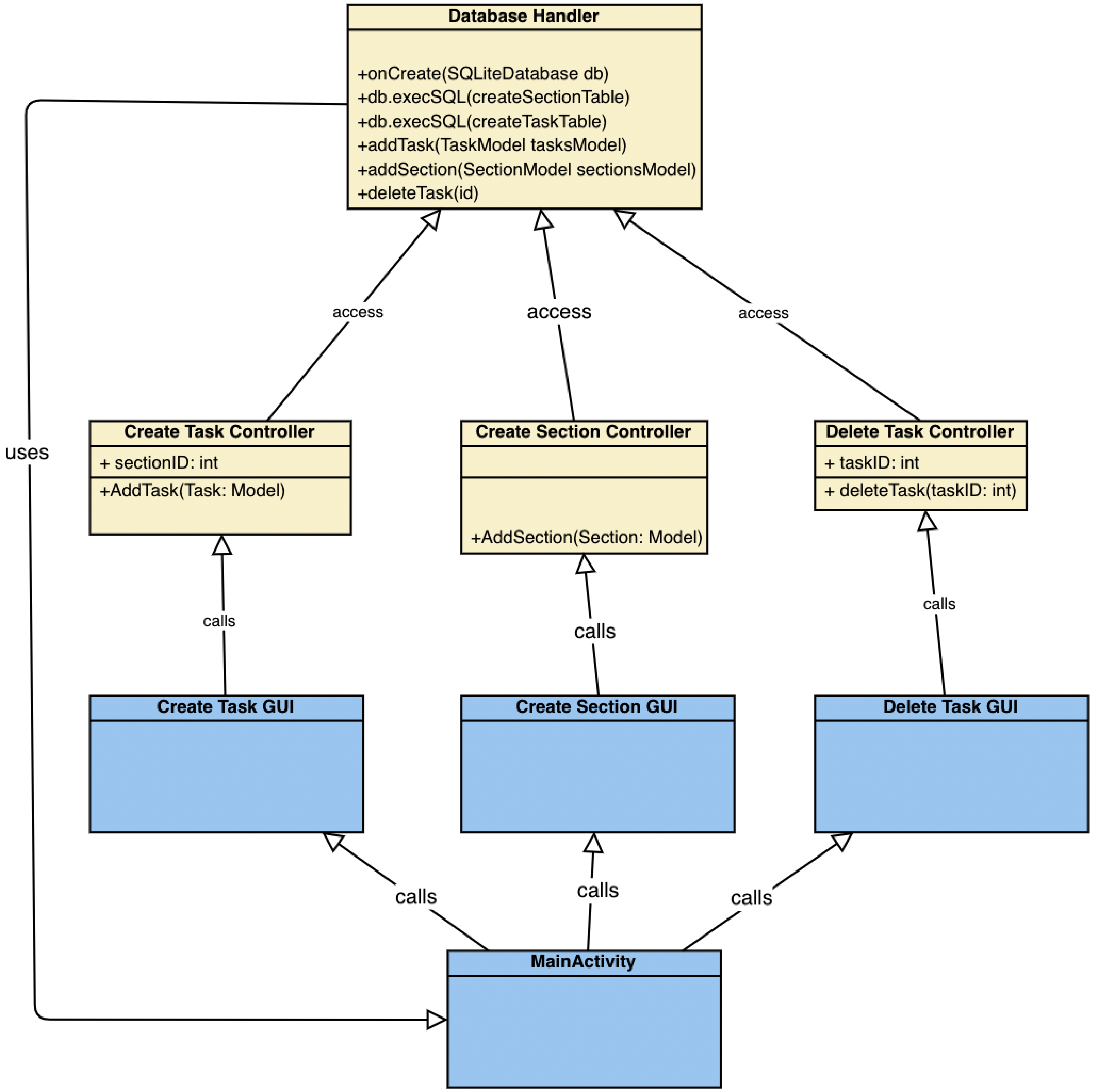
Task delete in Database

«confirmation msg>>

<<msg>>


Use Case 4: Create Section
User
:CreateSection GUI
:CreateSection Controller
:DatabaseHandler
<<click on + button>>
m:=AddSection(:SectionModel):String
Section created in Database
«confirmation msg>>
<<msg>>

Design Class Diagram



Database Handler

+onCreate(SQLiteDatabase db)
+db.execSQL(createSectionTable)
+db.execSQL(createTaskTable)
+addTask(TaskModel tasksModel)
+addSection(SectionModel sectionsModel)
+deleteTask(id)

Create Task Controller

+ sectionID: int

+AddTask(Task: Model)

Create Section Controller

+AddSection(Section: Model)

Delete Task Controller

+ taskID: int

+ deleteTask(taskID: int)

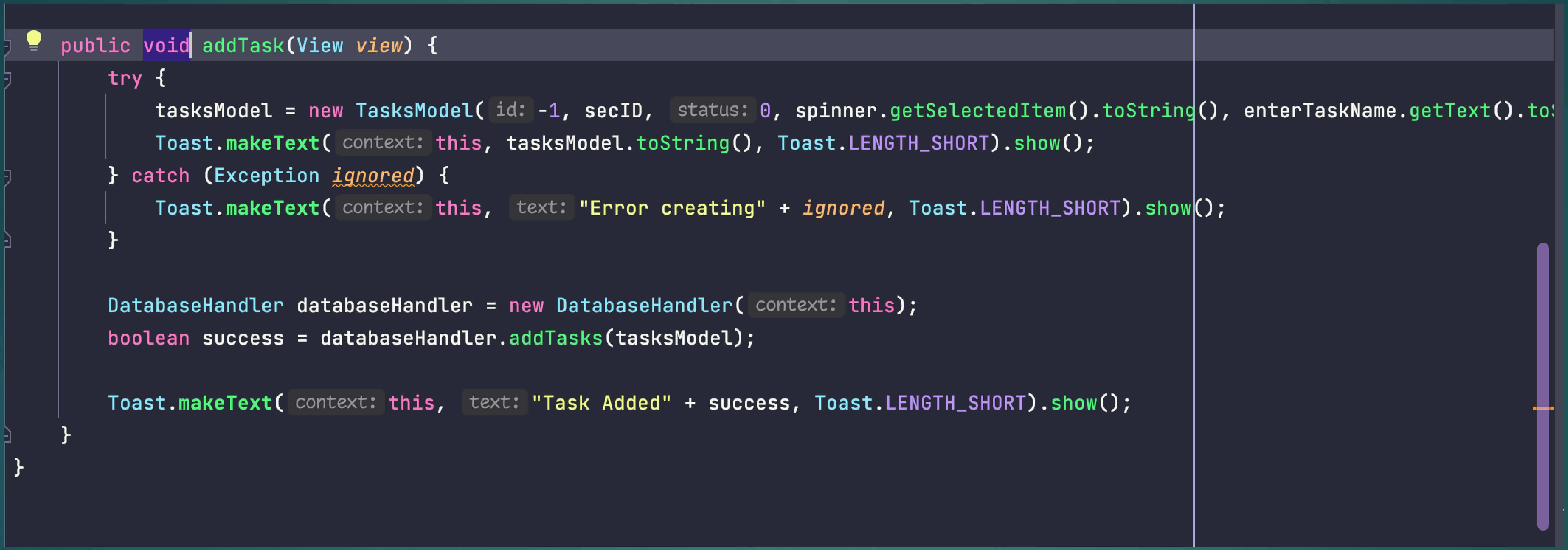
Create Task GUI

Create Section GUI

Delete Task GUI

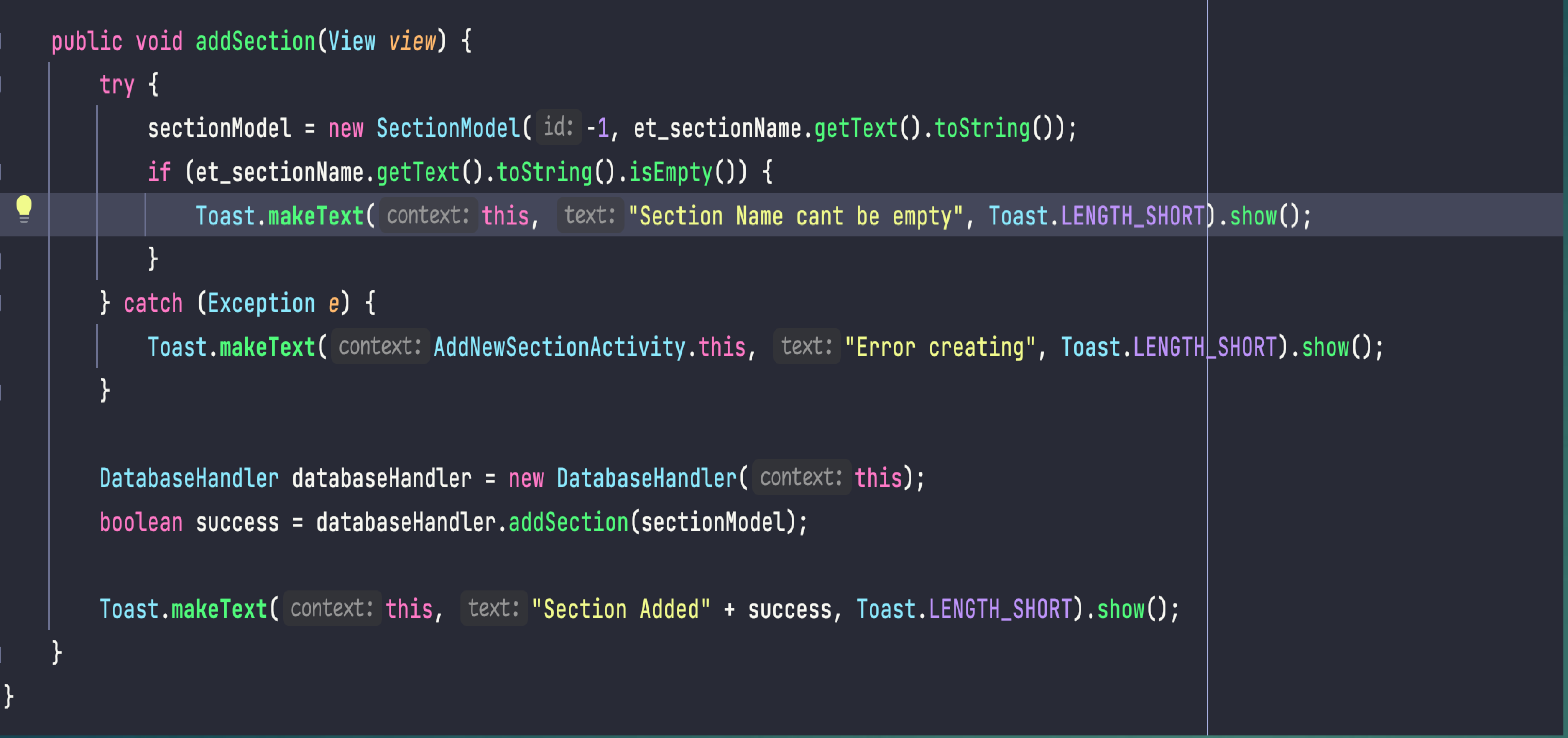
MainActivity

Code Snaps



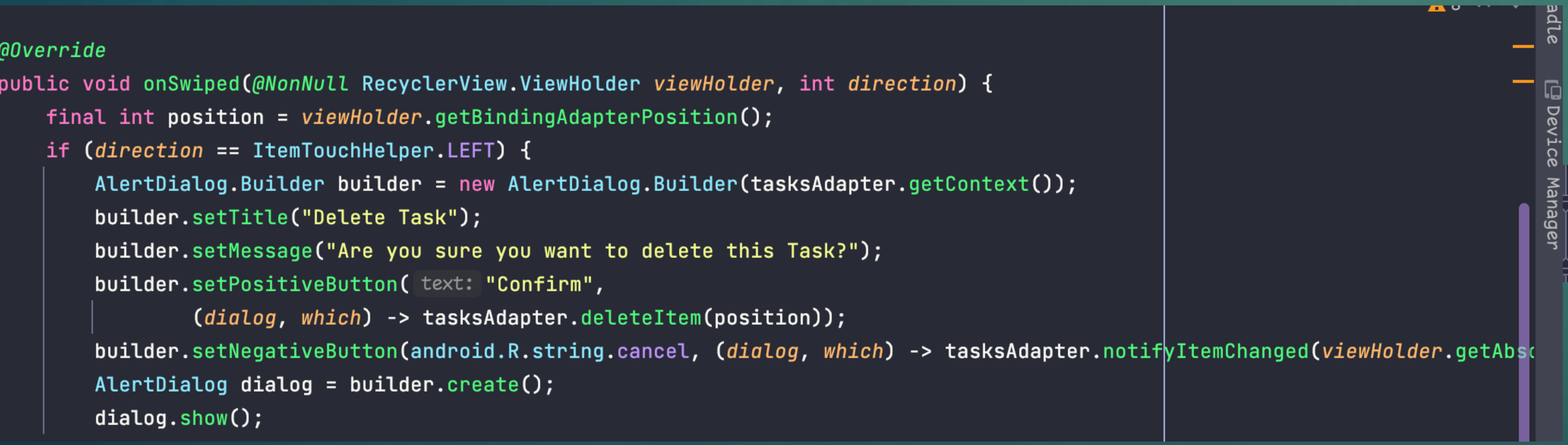
```
public void addTask(View view) {
    try {
        tasksModel = new TasksModel( id: -1, secID,  status: 0, spinner.getSelectedItem().toString(), enterTaskName.getText().to
        Toast.makeText( context: this, tasksModel.toString(), Toast.LENGTH_SHORT).show();
    } catch (Exception ignored) {
        Toast.makeText( context: this,  text: "Error creating" + ignored, Toast.LENGTH_SHORT).show();
    }

    DatabaseHandler databaseHandler = new DatabaseHandler( context: this);
    boolean success = databaseHandler.addTasks(tasksModel);

    Toast.makeText( context: this,  text: "Task Added" + success, Toast.LENGTH_SHORT).show();
  }
}
```

Use Case 1: Create Task

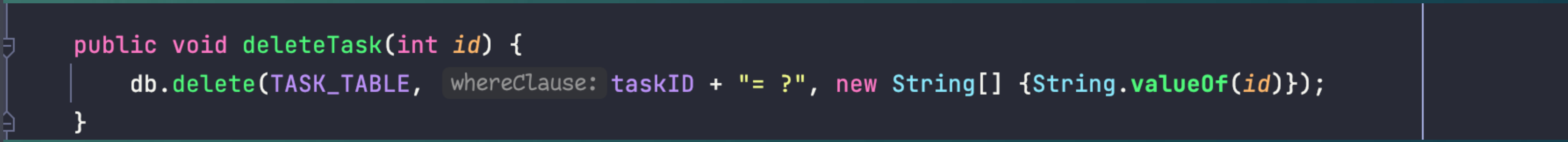


```
    public void addSection(View view) {
        try {
            sectionModel = new SectionModel( id: -1, et_sectionName.getText().toString());
            if (et_sectionName.getText().toString().isEmpty()) {
                Toast.makeText( context: this,  text: "Section Name cant be empty", Toast.LENGTH_SHORT).show();
            }
        } catch (Exception e) {
            Toast.makeText( context: AddNewSectionActivity.this,  text: "Error creating", Toast.LENGTH_SHORT).show();
        }

        DatabaseHandler databaseHandler = new DatabaseHandler( context: this);
        boolean success = databaseHandler.addSection(sectionModel);

        Toast.makeText( context: this,  text: "Section Added" + success, Toast.LENGTH_SHORT).show();
    }
}
```

Use Case 4: Create Section



```
@Override
public void onSwiped(@NonNull RecyclerView.ViewHolder viewHolder, int direction) {
    final int position = viewHolder.getBindingAdapterPosition();
    if (direction == ItemTouchHelper.LEFT) {
        AlertDialog.Builder builder = new AlertDialog.Builder(tasksAdapter.getContext());
        builder.setTitle("Delete Task");
        builder.setMessage("Are you sure you want to delete this Task?");
        builder.setPositiveButton( text: "Confirm",
                (dialog, which) -> tasksAdapter.deleteItem(position));
        builder.setNegativeButton(android.R.string.cancel, (dialog, which) -> tasksAdapter.notifyItemChanged(viewHolder.getAbs
        AlertDialog dialog = builder.create();
        dialog.show();
```



```
public void deleteTask(int id) {
    db.delete(TASK_TABLE,  whereClause: taskID + "= ?", new String[] {String.valueOf(id)});
}
```

Use Case 2: delete task

Summary

Topics that we have discussed so far:

- To sum up, we had walked you through the use cases that we intended to complete in current Iteration.
- The requirements that correspond to the use cases we highlighted.
- Updated High level use cases, RUCTM, Iteration Matrix and Domain Model.
- We covered Expanded Use cases, App screenshots, Sequence Diagram and Design Class Diagram of the tasks we have done so far in our project.
- Going forward, we intend to complete the remaining part of our project with implementing remaining use cases.

Q & A Time

- Shoot some questions but do not kill us!

THANK YOU!