Name: Abdul Ghiasy
Title: MyPlanner

Project Description: A time management web application that helps better manage scheduling and increase productivity of users by allowing users to add and organize notes and deadlines, and be able to set reminders. Once completed, this application will allow you to keep track of all responsibilities at any moment, lets you get used to focusing on important activities, and give you a feeling of extreme satisfaction at the end of every week.

Features Implemented:

ID	Description	Topic Area	Priority
UR-01	As a ***user***, I want to be able to create an account	Authentication	Critical
UR-02	As a ***user***, I want to be able to log in	Authentication	Critical
UR-03	As a ***user***, I want to be able to add notes	Notetaking System	High
UR-04	As a ***user***, I want to be able to view the notes I have created	Notetaking System	Medium
UR-05	As a ***user***, I want to be able to add deadlines for assignments and projects	Notification	High
UR-06	As a ***user***, I want to be able to set reminders for exam dates and other notable	Notification	High

Features NOT Implemented:

ID	Description	Topic Area	Priority
UR-07	As a ***user,*** I want to receive alerts or notifications for upcoming deadlines	Notification	High
UR-08	As a ***user***, I want to be able to rank my notes in terms of how important they are and priority-wise	Notetaking System	Low

Final Class Diagram:

Design Patterns:

Reflection:

In regard to the process of analysis and design, I've learned that developing a web app and code development in general requires you to detail as specifically as possible your project goals/objectives, plans on implementation, and then requires you to design several UML diagrams that are extremely crucial because they act as a sort of blueprint and logic when you actually begin coding your application. I've also learned that object-oriented programming is a vital way of programming because it allows programmers to set a sort of foundation before implementation and enables you to think like you are working with applicable entities and objects. In addition, I've also learned about the importance of several different concepts such as encapsulation (the method for wrapping data and code acting on that data together as a single unit), inheritance (when one class gains the properties of another), abstraction (giving users only functional methods, but hiding implementation details), etc. Overall, I feel more confident in tackling on the process of creating, designing, and implementing a system.