Contributors:
Conrad Persowski
Austin Beal
Jordan Miller
Tim Zelikovsky

Scheduler Design Manual

Introduction

Our scheduling app is designed in java and mainly uses object oriented designs to achieve our desired implementation. Since the app needs an interface for user interaction, we decided javafx would be a very important tool to use. We had multiple GUI's, some with minimal functionality such as the welcome screen and others containing more functions such as the main screen. Throughout the project we had about 22 classes each having a specific task and bearing too many burdens. For example each GUI had a different class which made it easier for us to make changes throughout the project. We also utilized the resources folder in order to get different images on our startup screen GUI and different time buttons. After the GUI's were completed, we obviously needed to figure out how to help the user become an overall more productive person. We created a class for tasks, an administrator and a productivity graph. The administrator is able to assign tasks to the user and the user is able to set times in which they hope to complete the task. The average time spent on each task is then displayed on the graph which we are hoping encourages the user to try and not exceed a specific average time. The majority of these classes were integrated and worked well together.

OOD

Almost every class we made used some sort of object oriented design. In our GUI's we had different children our stage such as a text, button, shapes etc. These were all objects and we used different java built- in functions to manipulate them however we wanted. As a result we

were able to create good interfaces with different colored text, different sized shapes which made the interaction with our app simple. There were also different scheduling components that needed to be implemented using objects such as start and end date of specific tasks. The timing objects and other objects related to productivity were all used to help determine how much time was spent on each task which was then displayed in our barchart which is also an object. So as you can see, by using object oriented design we were able to complete a lot of our user stories such as wanting to choose between a user and an admin. Two buttons were created each having different functionality for the user and administrator.

User Stories

1	As a user, I want to choose between logging in as an admin or as a user in Productivity, so that I can make sure certain features are restricted
2	As an admin, I want to be able to enter credentials to access admin interface, so that only valid admins can access the admin interface
3	As a user, I want to be able to choose between 10min, 30min & 60min stop watches, so that I can dedicate that amount of time to a task
4	As a user, I want to be able to reset a timer once its completed, so that I can keep working on a task and remain productive
5	As a user and admin, I want to be able to visualize my productivity chart which displays time spent for each task type, so that I can start working on tasks that take the longest time to complete
6	As a user and admin, I want to be able to input dates/times when a task was being worked on/completed, so that I can visualize the productivity plots
7	As a user and admin, I want to be able to pause/unpause a timer, so that I take a quick break if needed and later return to the task
8	As a user and admin, I want to receive a popup message when a timer is finished, so that I can either move on to the next task or reset the timer and continue working
9	As a user, I want to make sure I am restricted from using admin priveleges within Productivity, so that I can focus on the current tasks at hand
10	
11	**Uncompleted User Stories**
12	As a user and admin, I want to be able to enter a custom category of tasks to work on (i.e **school, fitness/training, meditation etc)**
13	As a user and admin, I want to be able to manually input the date/time spent per task
14	As a user and admin, I want to be able to enter a custom task to work on
15	As a user and admin, I want to visualize my productivity charts according to time spent on custom task entered by user
16	As a system informs user of which task to start first based on time spent across tasks
17	As an admin, I want to be able to assign tasks for users to complete and a date/time to complete them by
18	As a user and admin, I want the Productivity system to inform me of which task to start on first based on time spent

CRC Cards

Class:

- Administrator

Responsibilities:

- Create an administrator with specific credentials
- Assign tasks to users
- Add users

Collaborators:

- User

Class:

- GUIController

Responsibilities:

- Accept the Username and password of administrator
- Change pages depending on user inputs
- Add events and tasks upon request

Collaborators:

- GUIMain

Class:

- GUIMain

Responsibilities:

- Run the GUI

Collaborators:

- GUIController
- GUIView
- GUIModel

Class:

- GUIModel

Responsibilities:

- Create administrator object with credentials

Collaborators:

- Administrator

Class:

- GUIView

Responsibilities:

- Display the GUI
- Place all the tasks on the GUI
- Places the buttons on the GUI and creates functionality for them

Collaborators:

- Task

Class:

- Task

Responsibilities:

- Takes in details for different tasks

Collaborators:

- Administrator

Class:

- User

Responsibilities:

- Creates a user

Collaborators:

- Administrator

Class:

- startUp

Responsibilities:

- Create startup GUI with choice of user or administrator

Collaborators:

- GUIController

Class:

- ManualController

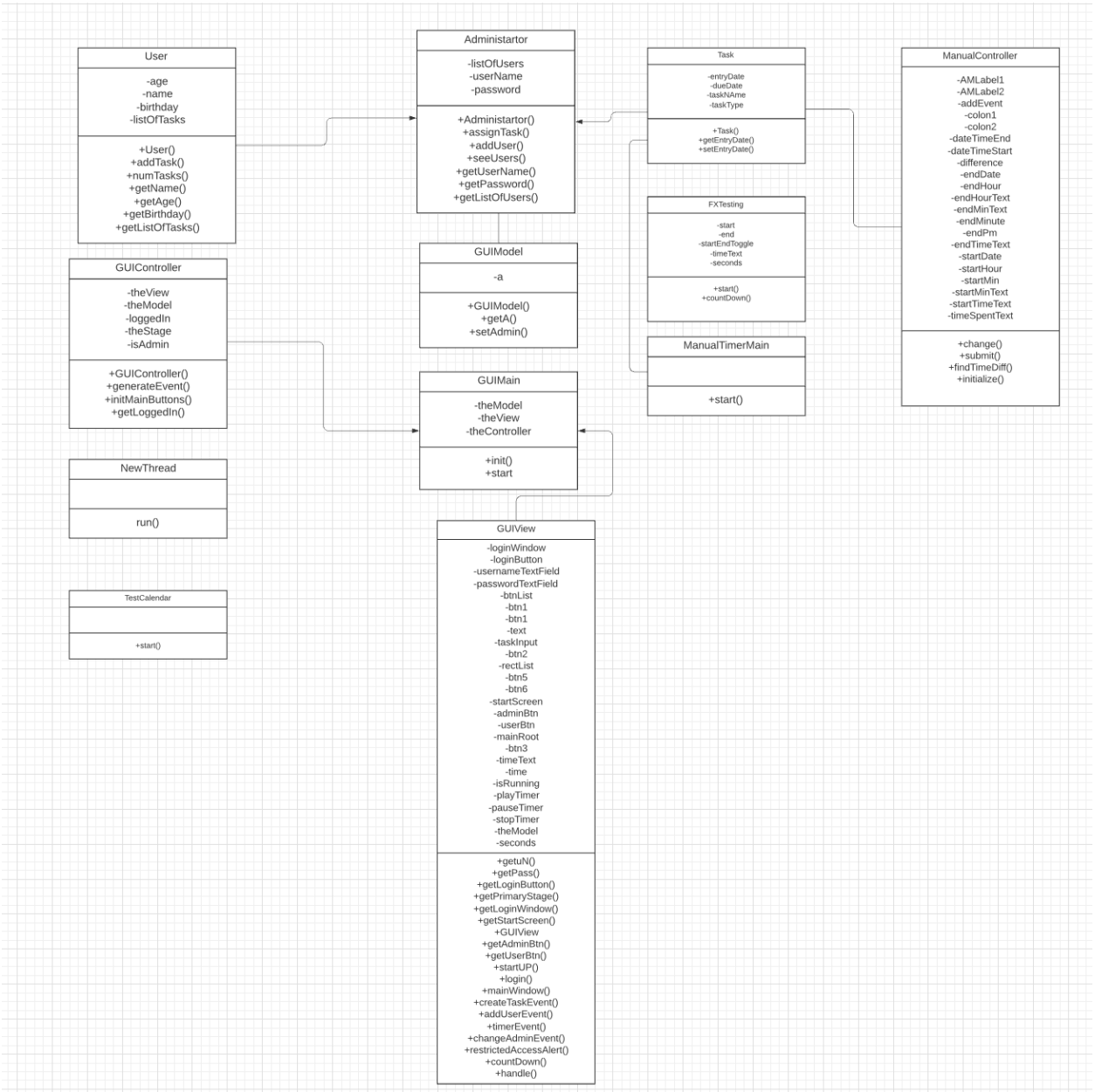
Responsibilities:

- Creates manual control for time

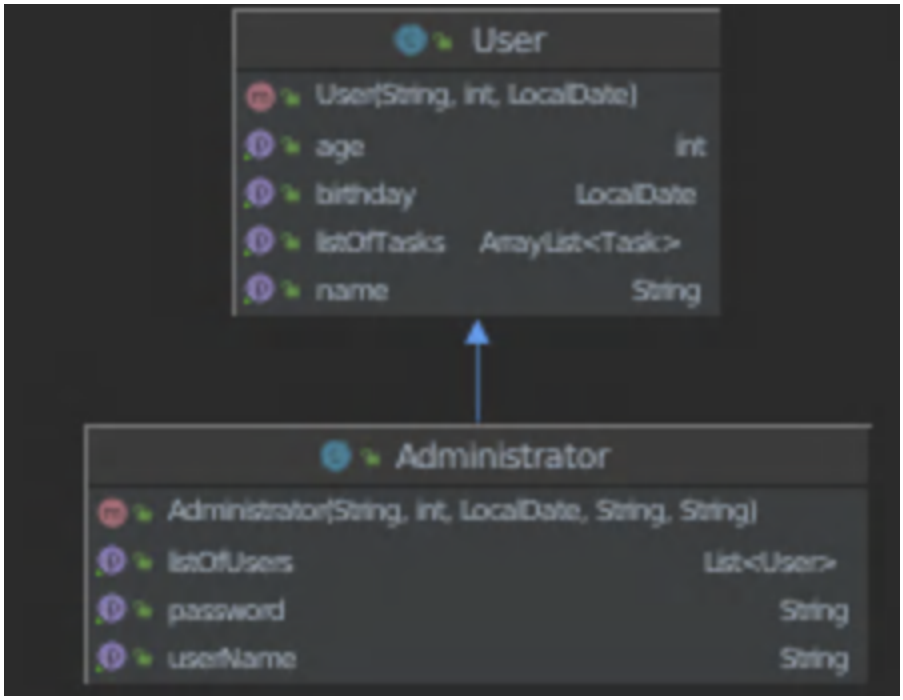
Collaborators:

- GUIControoler
- Task

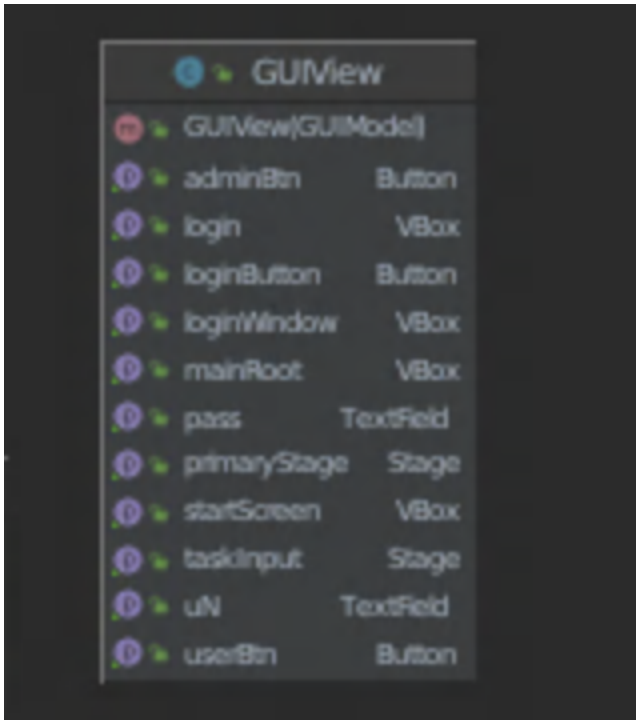
UML Diagram:



Intellij UML Diagram:



This screenshot shows the relationship between the user class and the administrator class. It shows that the User is created by the administrator therefore the user cannot exist without an administrator. The diagram also shows the different private and public members that both classes contain.



This screenshot shows the UML class for the GUIView. There are many private members and public members shown as this class is doing a lot for our program.

Citations

1. Timer Functionality Source: https://www.youtube.com/watch?v=t2Bv6hwELsU
2. JUnit Testing: https://medium.com/aperto-an-ibm-company/unit-testing-in-agile-web-projects-4db5547a733b