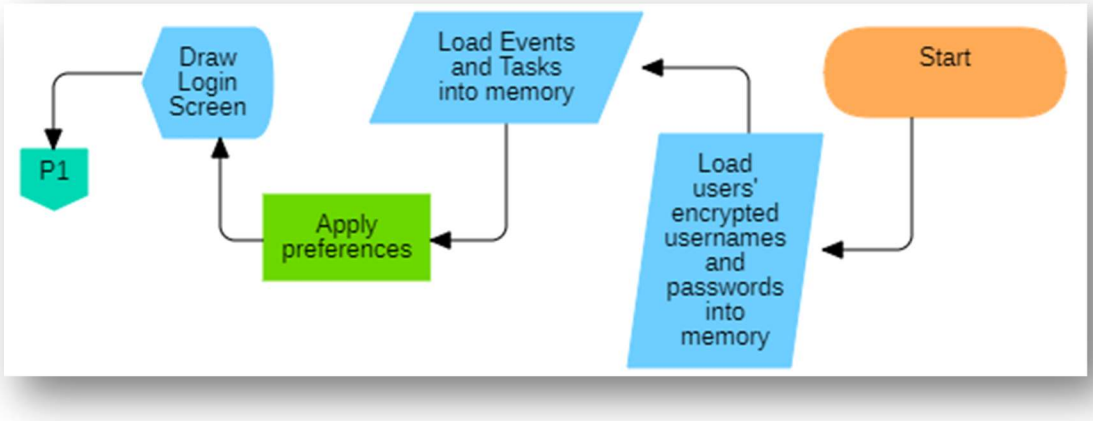
Criterion B:

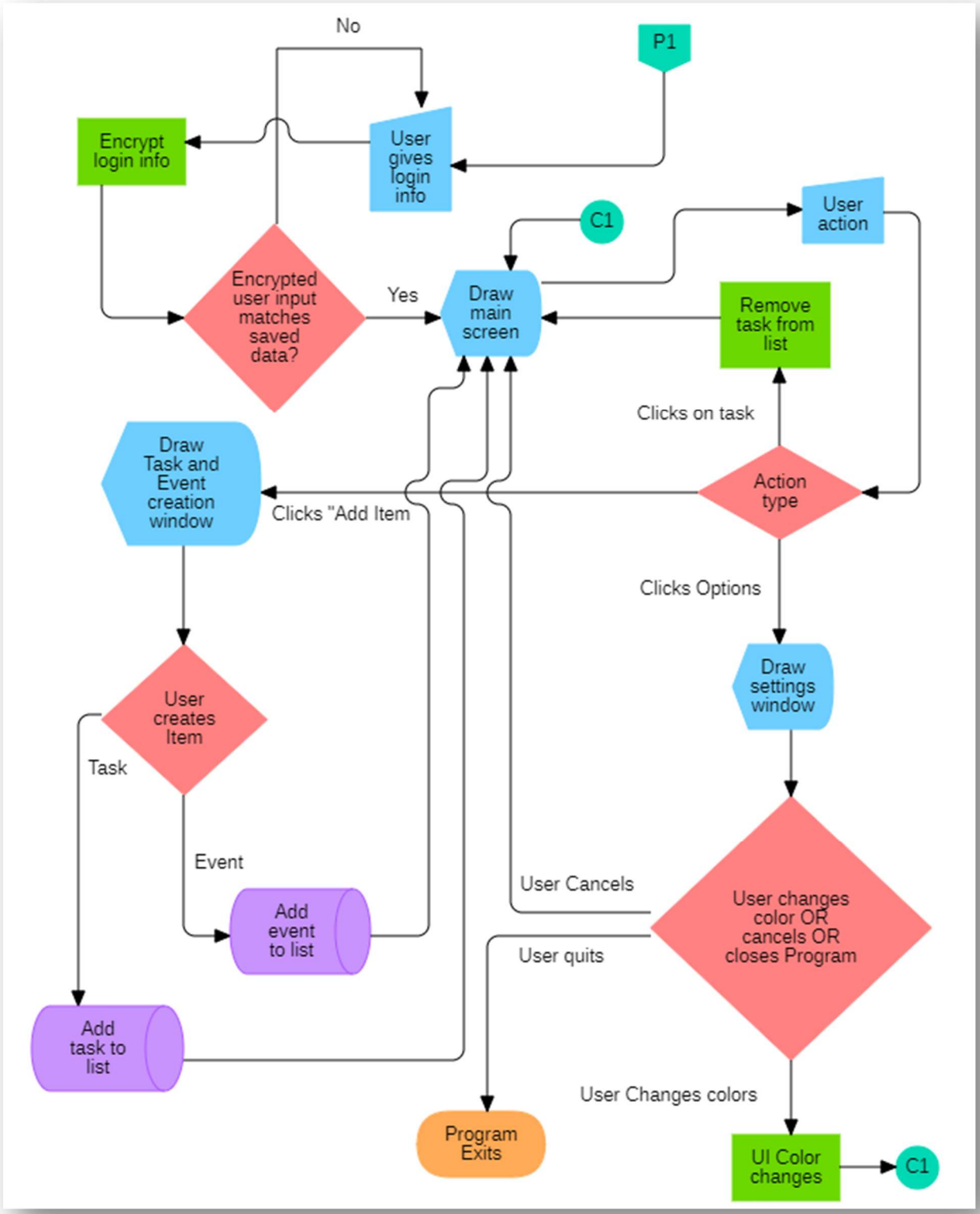
Tests:

Criteria	Test	Result
1. Can the client add and remove tasks and events with ease?	Check that tasks and events can be added with ease.	Pressing the add button brings up a menu where tasks are added to the task list.
2. Can the client add due dates to tasks?	Check that tasks can have due dates added to them.	Clients can be notified when tasks are due.
3. Can the client add reminders to events and tasks?	Check that events and tasks can be given reminders.	In the "Add item" dialog there is a place to add reminders.
4. Can the client add priority levels to their tasks?	Check that tasks can be given priority levels.	In the "Add item" dialog there is a JSlider to choose priority levels.
5. Can the client see completed tasks?	Check that completed tasks are still accessible to the client.	Clients can press a button and be shown completed tasks.
6. Can the client change the proportion of their UI used by the calendar and task list?	Check that the calendar and task lists are resizable.	The JSplitPane has a built in resize bar.
7. Can the client change the colors of their UI?	Check that UI colors are user mutable.	The "Options" dialog has color settings.

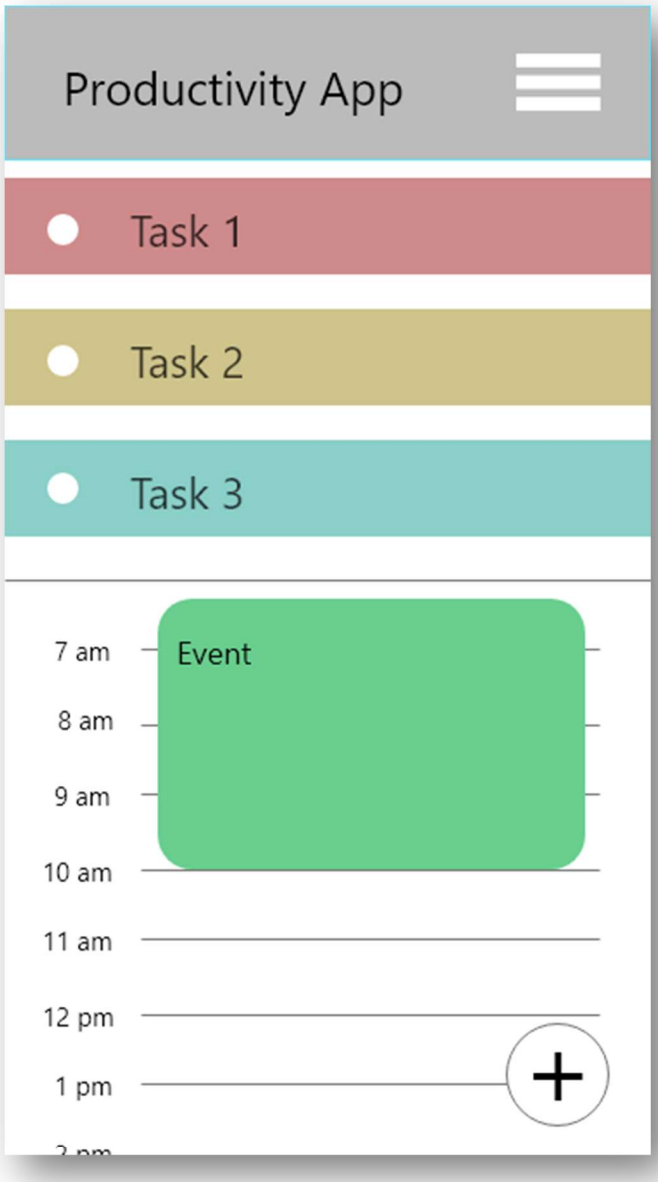
Flow Charts

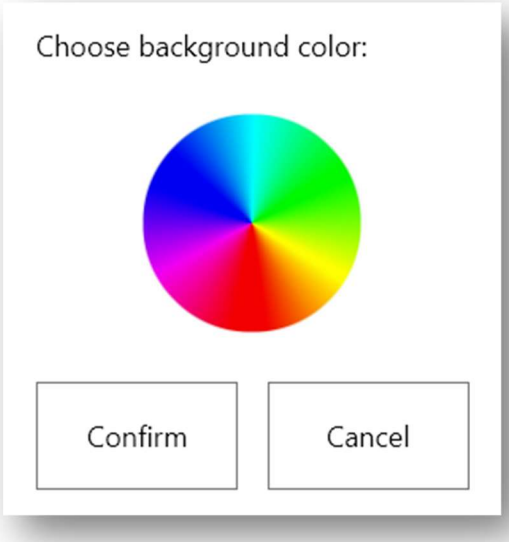


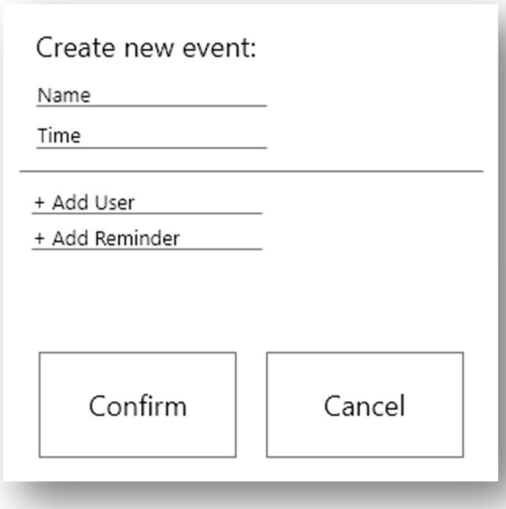
Flow Charts


No
P1
Encrypt login info
User gives login info
C1
User action
Encrypted user input matches saved data?
Yes
Draw main screen
Remove task from list
Clicks on task
Draw Task and Event creation window
Clicks "Add Item
Action type
Clicks Options
Draw settings window
User creates Item
Task
Event
Add event to list
User Cancels
User changes color OR cancels OR closes Program
User quits
Add task to list
User Changes colors
Program Exits
UI Color changes
C1

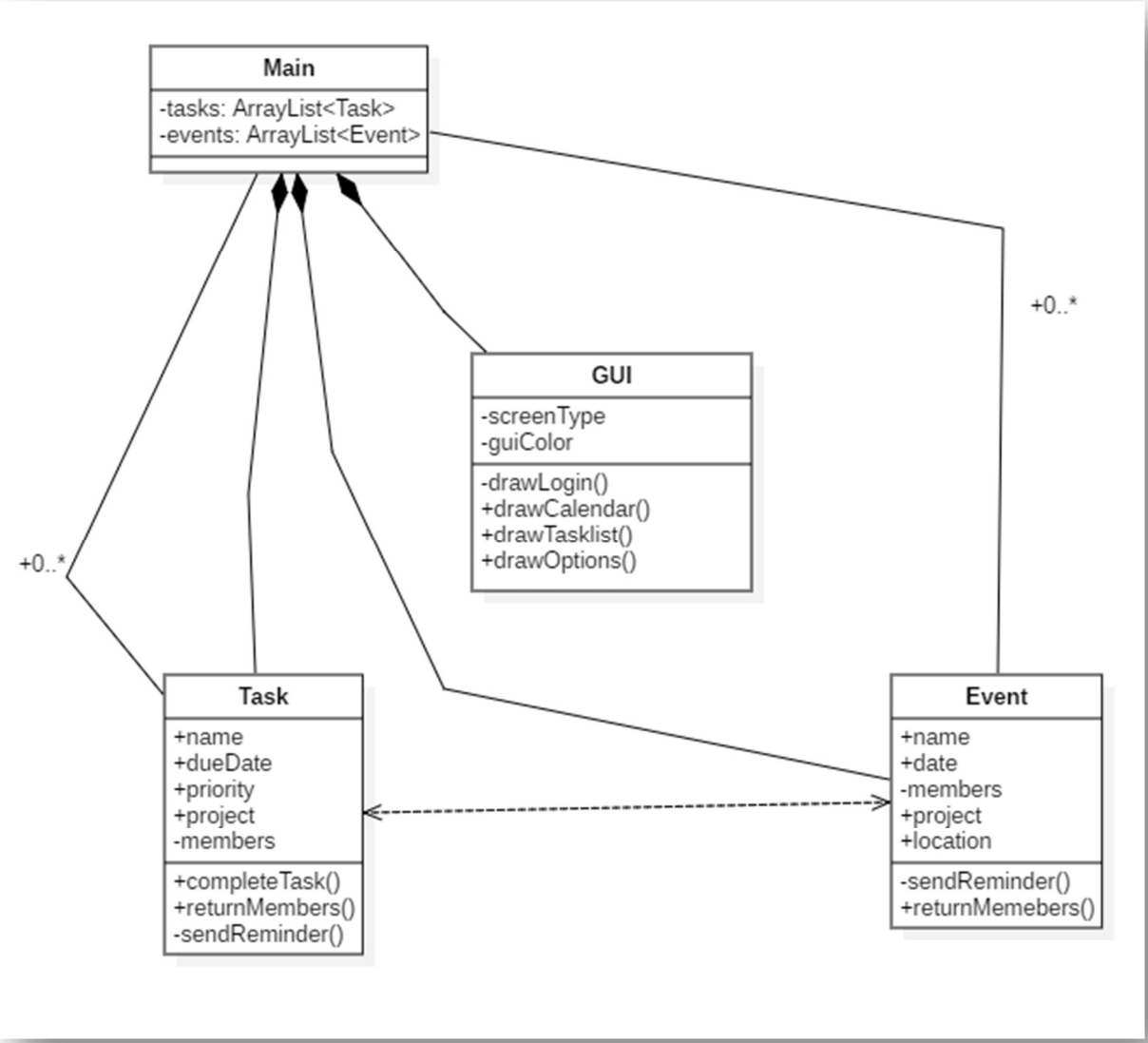
Planned Graphics

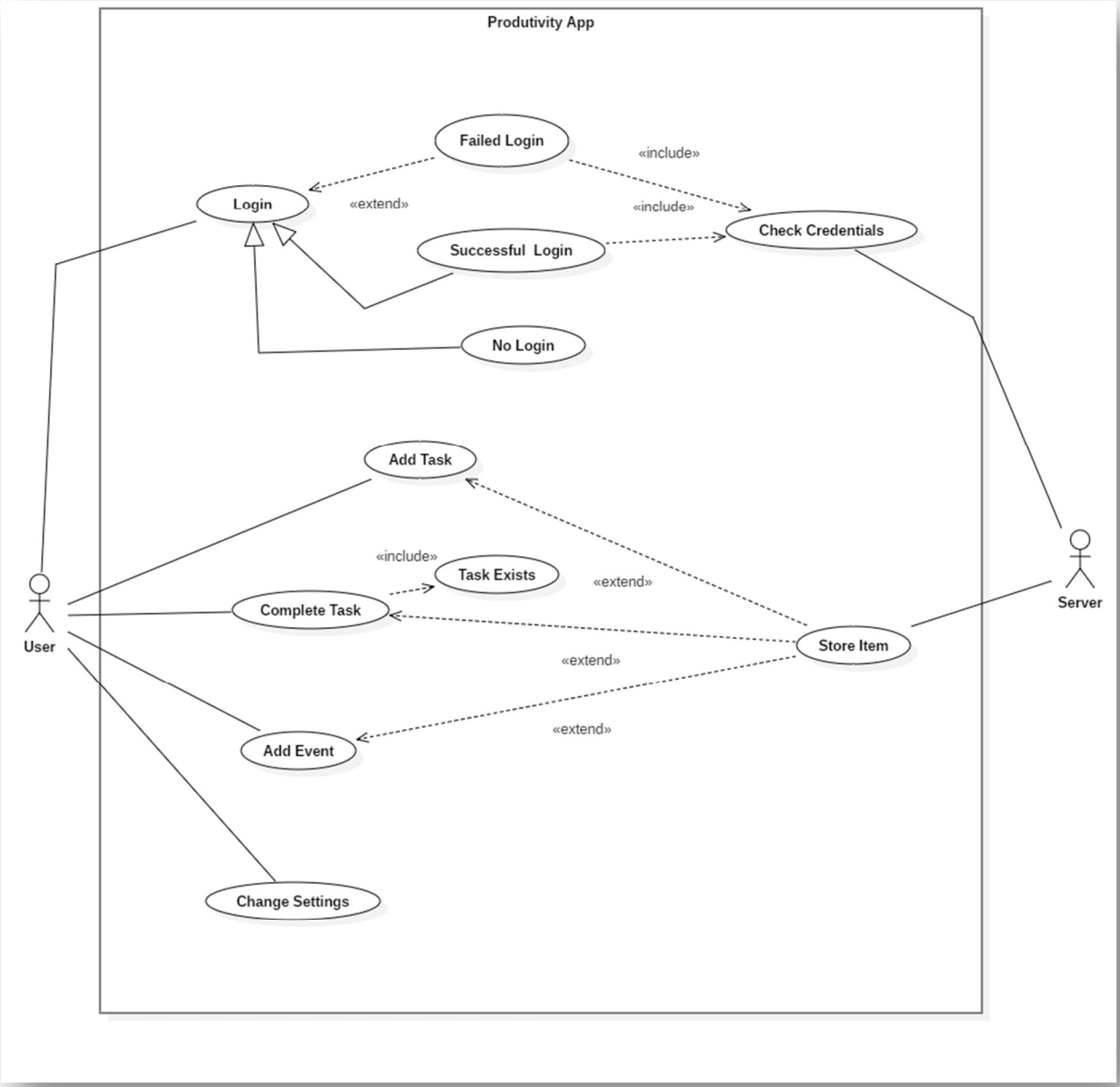






Planning Documents



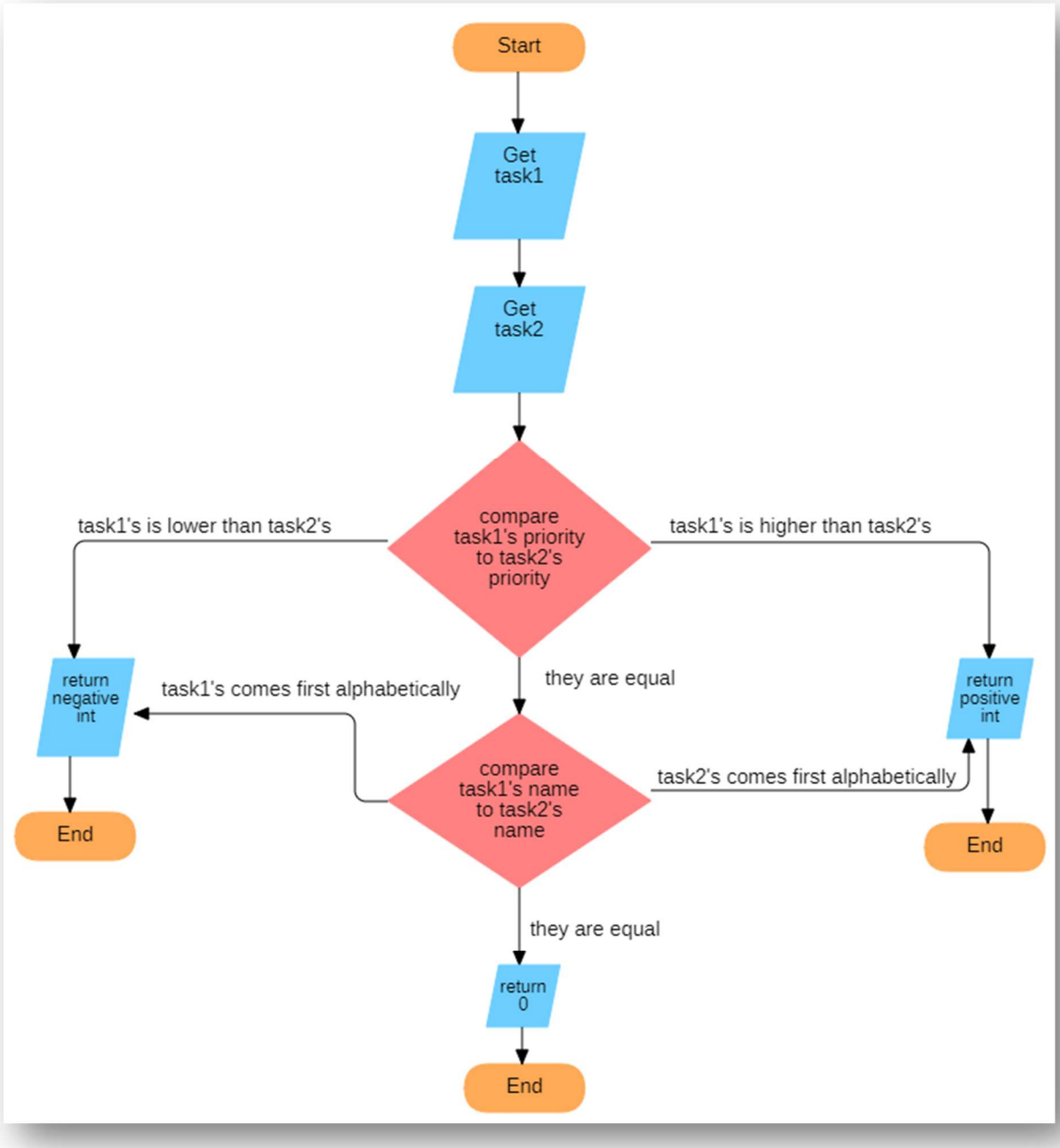


Key Algorithms

TaskComparator

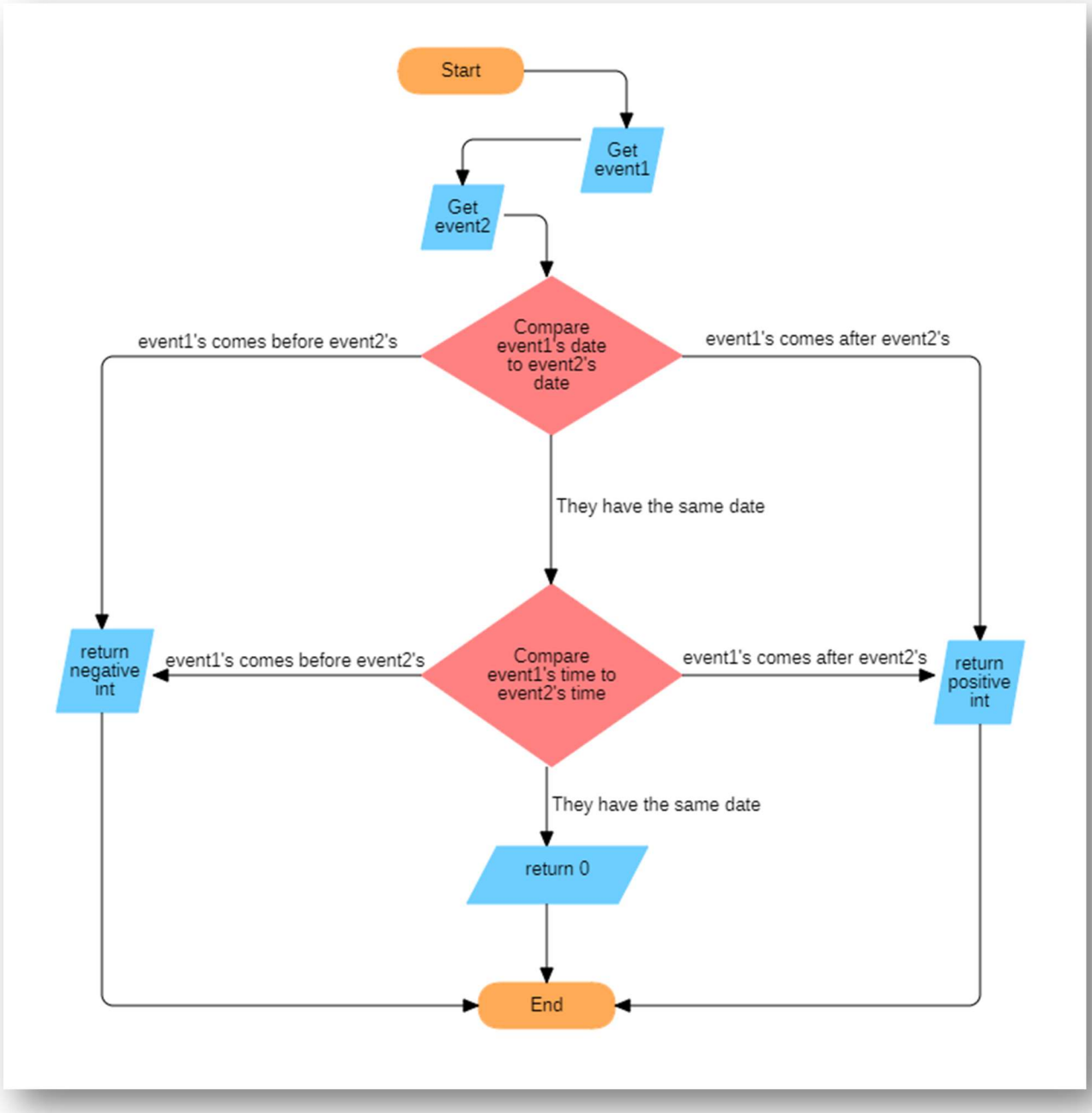
I had to implement a comparator for my custom “Task” class so I could order my tasks by priority in Java Swing. This comparator compares task priorities and then, as a last resort, to settle differences, task names.

TaskComparator



EventComparator

I had to implement a comparator for my custom “Event” class so I could order my events by date in Java Swing. This comparator compares event dates and then, as a last resort, to settle differences, event times. This ensures the proper ordering of events.



TaskCellRenderer

In order to draw tasks in Java swing, I had to create a cell renderer. This extends JLabel so I can use the rendering methods and properties that exist in the JLabel class. This renderer essentially just takes the Task's name and displays it in a JLabel. However, it does have an important functionality where, when the cell is selected, that Task is removed from the JList model (this removes it from the list of drawn tasks, but not from the task list or drive.