Unit Test Cases

Calendar Test Case:

```
Calendar{
 setTime(int hour, int min)
 getTime()
}

Int known_hour = 1
Int known_min = 30
String known_time = known_hour+”:”+known_min
Calendar TestTime = new Calendar()
TestTime.setTime(known_hour,known_min)
assertEqual(TestTime.getTime(),known_time)
```

setTime(int time) method

Calendar class is used to keep track of the user’s personal calendar

In this test case we are testing setTime() to ensure the user inputs the correct values for time. Some valid times would be ints from 1-12 for hours and int from 0-59 mins).

Create Sticky Note Test Case:

```
String knownMessage = “Random String”
 Int knownX = 500
 Int knownY = 500
 Int badMessage = “bad message”
 Int badX = -30
 Int badY = -30
 createStickyNote testSticky = (knownMessage, knownX,
knownY)
 assertEqual(testSticky.message,knownMessage)
 assertEqual(testSticky.X, knownX)
 assertEqual(testSticky.Y, knownY)
 assertNotEqual(testSticky.message,badMessage)
 assertNotEqual(testSticky.X, badX)
 assertNotEqual(testSticky.Y, badY)
 assertException()
```

Test that checks if the creation of a sticky note is done properly and without error.

Tests if a sticky note is at the correct X and Y that the user selected, tests if the text is equal to inputted text. Checks the oppisite of these things, as well as if the length of the message is too long. Returns error if true

Class BulletinBoard:
Test Case: Add Valid Component

```
BulletinBoard{
    addComponent(Component component)
          containsComponent(Component  component):
boolean
}

Component knowncomponent = new StickyNote();
BulletinBoard testBoard = new BulletinBoard();
testBoard.addComponent(knownComponent);
assertEqual(testBoard.containsComponent(knownComp
onent), true);
```

Add valid Component test case ensures that when we add a valid component such as a ‘Sticky Note” it appears on the bulletin board. For that, we create a new StickyNote instance, which represents the component we want to add. Then we instantiate our bulletin board, we call “addComponent” with our sticky note, Finally we assert that “containsComponent” returns true for our sticky note, which confirms that it has been added successfully.

BulletinBoard
Test Case: Add Valid Component

```
Test Case: Add Null Component
testBoard.addComponent(null);
assertException();

Test case: Add Duplicate Component
testBoard.addComponent(knownComponent);   (place   where
component already added)
assertFalse(testBoard.containsComponent(knownComponent))
;

Test Case: Add component Off-screen
Component offScreenComponent = new StickyNote();
testBoard.addComponent(offScreenComponent);
assertException();
```

We also look at what happens if there's a mistake. Like if we try to add a sticky note that doesn't exist, or if we try to add the same one more than once, or if we put a sticky note somewhere off the edge of the screen. In each of these cases, our program should catch the error and let us know. This will prevent any issues while using the board. For example: adding null component, duplicate component, off-screen component would throw an exception

Task Test Case:

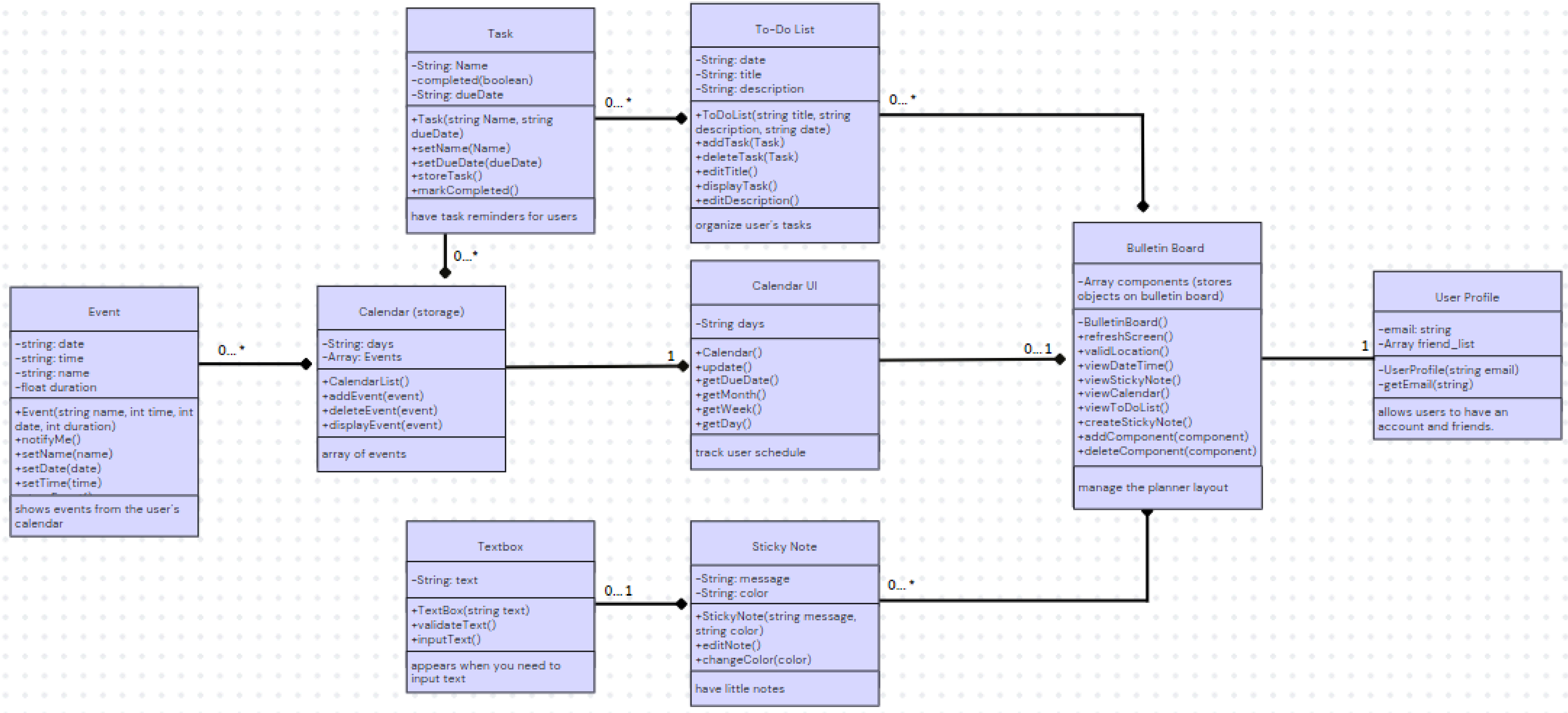
```
Task{
 setName(string)
 getName():string
}

String known_string = “Task”
Task testTask = new Task()
testTask.setName(known_string)
assertEqualString(testTask.getName(),known_string)
```

setName(String name)

Method to set the name of a task. Also testing getName() at the same time. Anything that is a string will work for this method. If known_string was not a string, we would assertFalse() instead.

Updated UML Diagram


Task
-String: Name
-completed(boolean)
-String: dueDate
+Task(string Name, string dueDate)
+setName(Name)
+setDueDate(dueDate)
+storeTask()
+markCompleted()
have task reminders for users
0... *
To-Do List
-String: date
-String: title
-String: description
+ToDoList(string title, string description, string date)
+addTask(Task)
+deleteTask(Task)
+editTitle()
+displayTask()
+editDescription()
organize user's tasks
0... *
0...*
Event
-string: date
-string: time
-string: name
-float duration
+Event(string name, int time, int date, int duration)
+notifyMe()
+setName(name)
+setDate(date)
+setTime(time)
shows events from the user's calendar
0... *
Calendar (storage)
-String: days
-Array: Events
+CalendarList()
+addEvent(event)
+deleteEvent(event)
+displayEvent(event)
array of events
1
Calendar UI
-String days
+Calendar()
+update()
+getDueDate()
+getMonth()
+getWeek()
+getDay()
track user schedule
0... 1
Bulletin Board
-Array components (stores objects on bulletin board)
-BulletinBoard()
+refreshScreen()
+validLocation()
+viewDateTime()
+viewStickyNote()
+viewCalendar()
+viewToDoList()
+createStickyNote()
+addComponent(component)
+deleteComponent(component)
manage the planner layout
1
User Profile
-email: string
-Array friend_list
-UserProfile(string email)
-getEmail(string)
allows users to have an account and friends.
Textbox
-String: text
+TextBox(string text)
+validateText()
+inputText()
appears when you need to input text
0... 1
Sticky Note
-String: message
-String: color
+StickyNote(string message, string color)
+editNote()
+changeColor(color)
have little notes
0... *