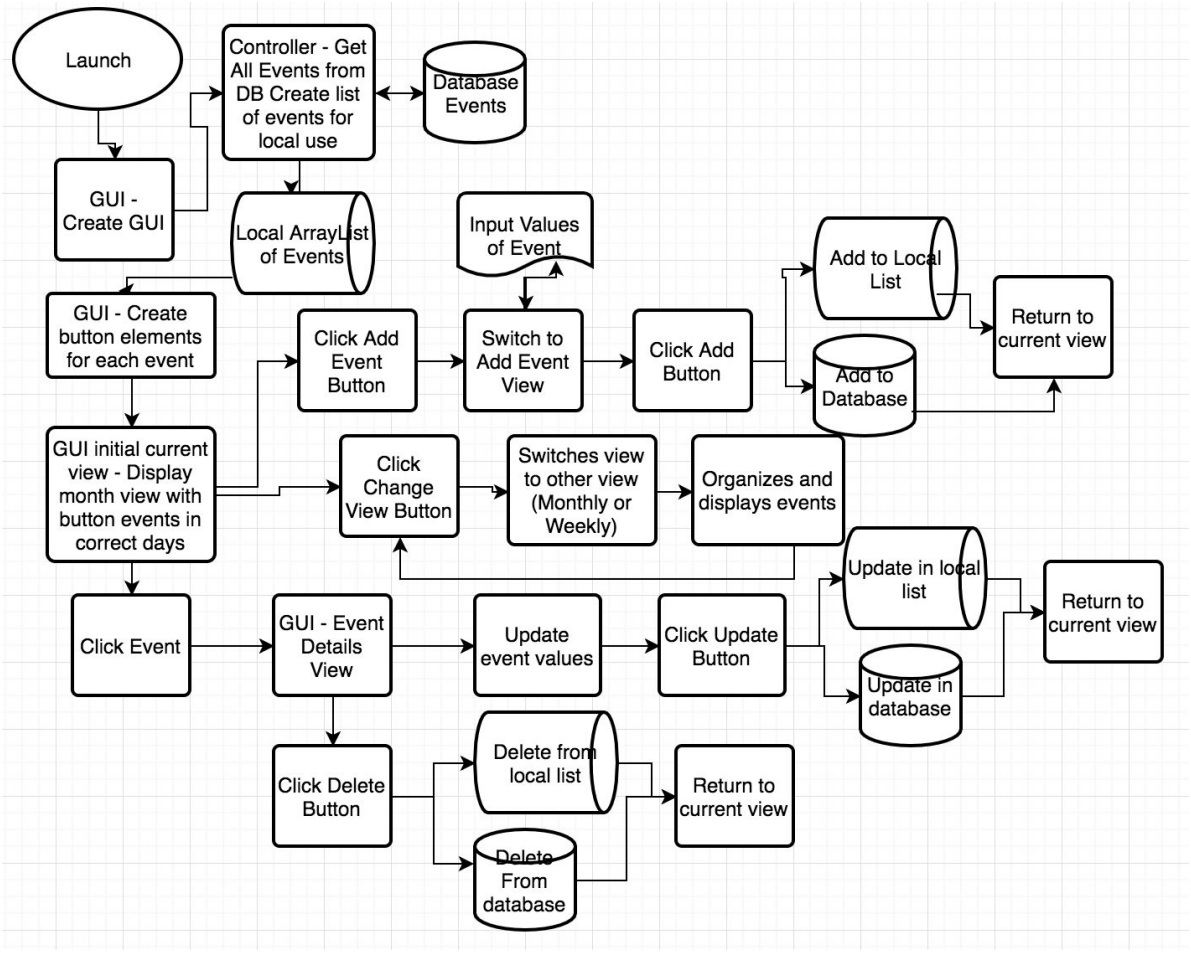
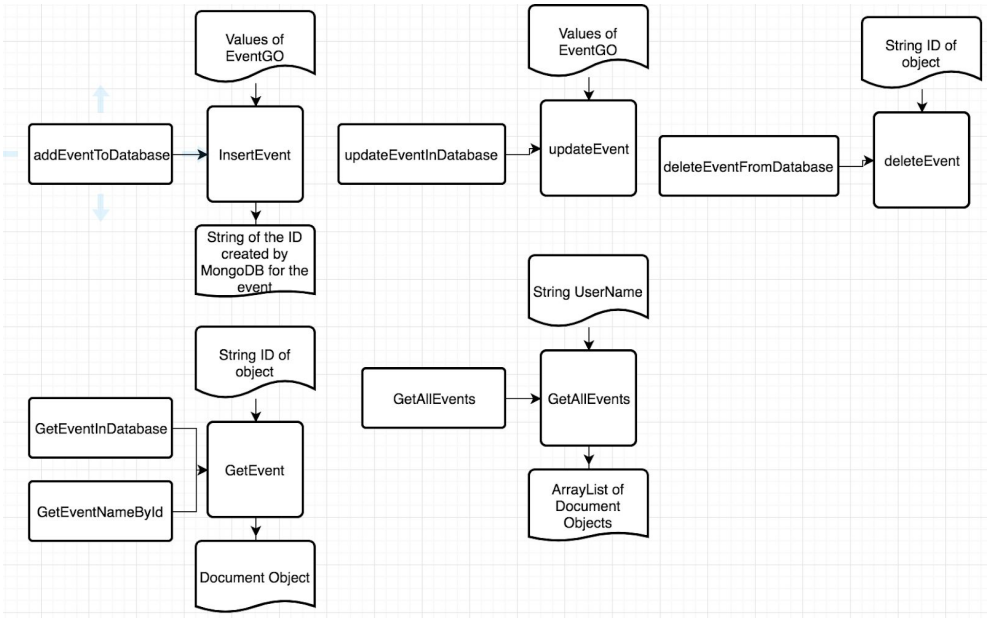
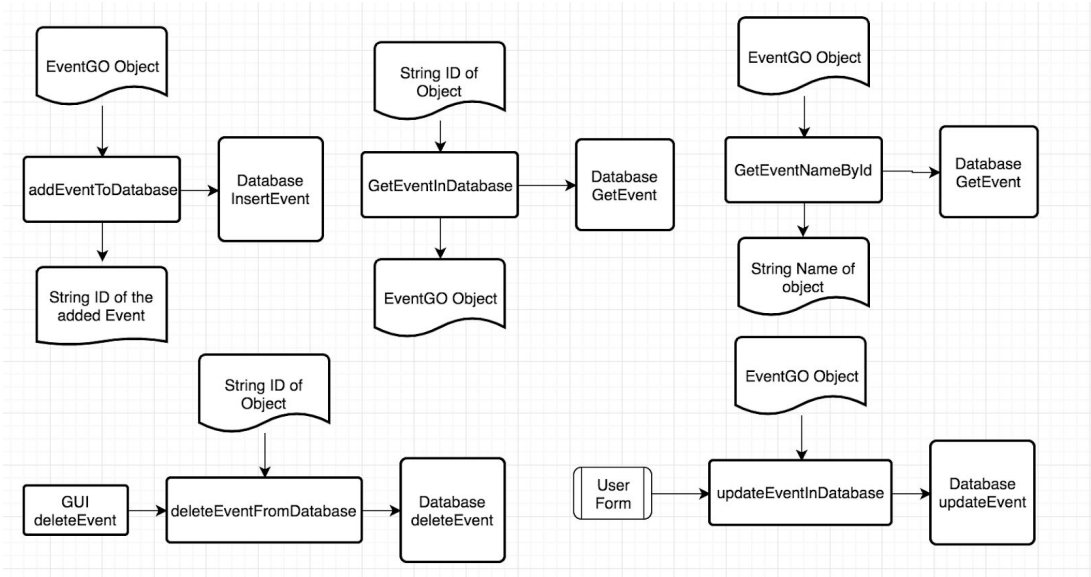
1. Classification of Components

a. 

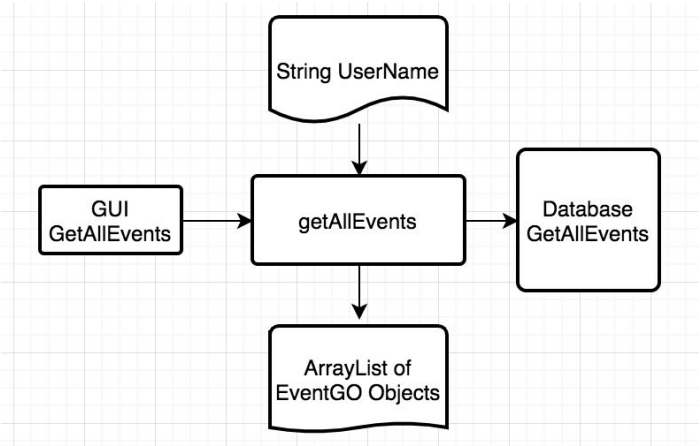
b. EventDBO Components (Components that depend on it depicted on the left of each)

i. 

c. Controller Components (Components that depend on it depicted on the left of each Components that it depends on depicted on the right of each)

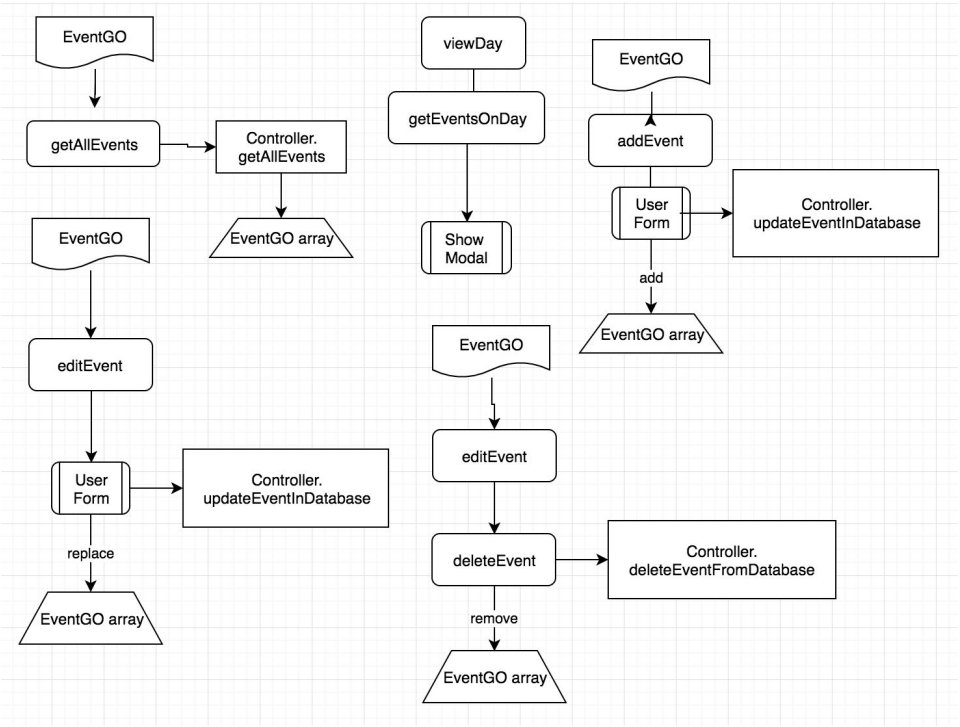
i. 

ii.

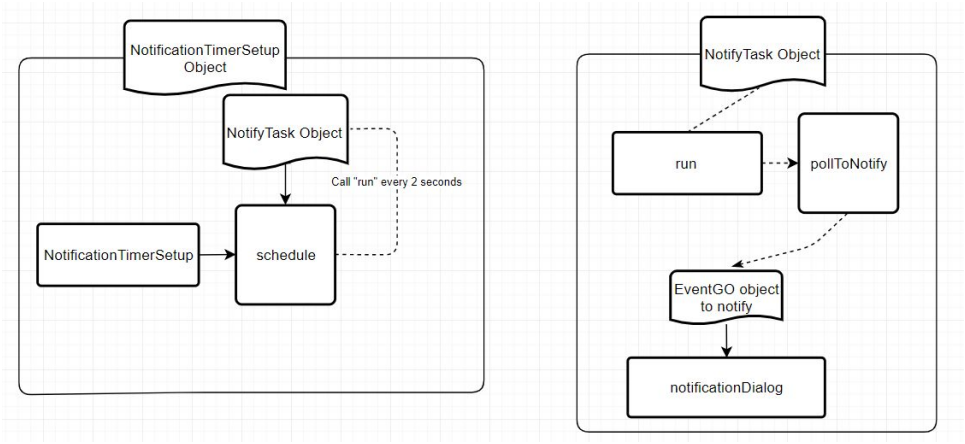


d. GUI Components

i.



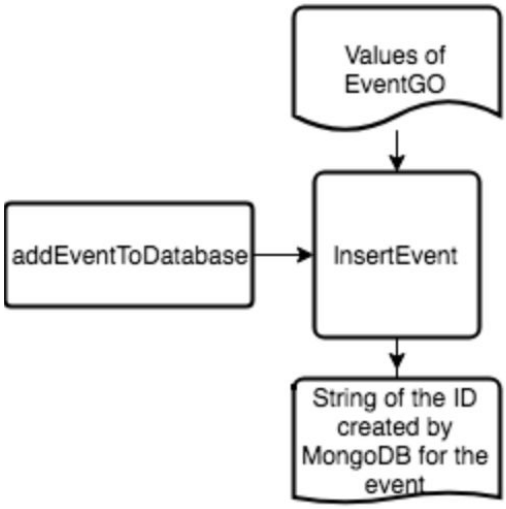
e. Notification Components



f TopDown vs. BottomUp

ii. We decided to do bottom up testing for our project. We had completed the I/O of the database and corresponding controllers but we had not yet completed our UI so it made more sense to do bottom up while UI was still in development.

2. Incremental testing



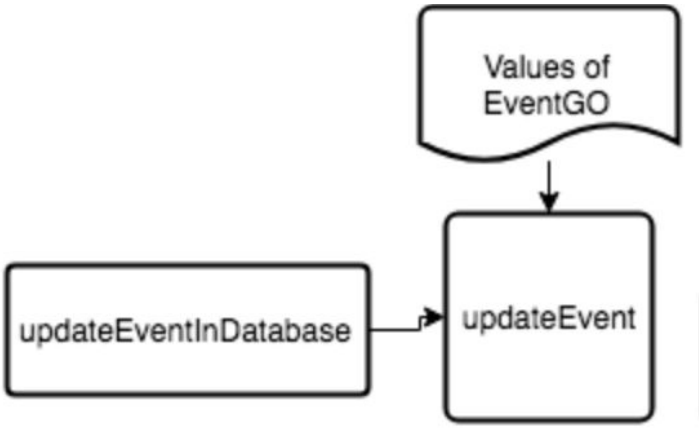
Module	**Database Insert Event**

Incremental Testing

Defect #	Description	Severity	How Corrected
1	Application should not crash when attempting to add null values to database	1	There should be no case where a null value is sent to the database handler. Do not add the event and return an empty string.

Regression Testing

Defect #	Description	Severity	How Corrected
1	Fixing adding null values in update Database causes addEventToDatabase to add an event to the local list that is invalid and not in database	1	When adding to local list check if the value of the id is empty. If it is don't add it.



Module	**Database Update Event**

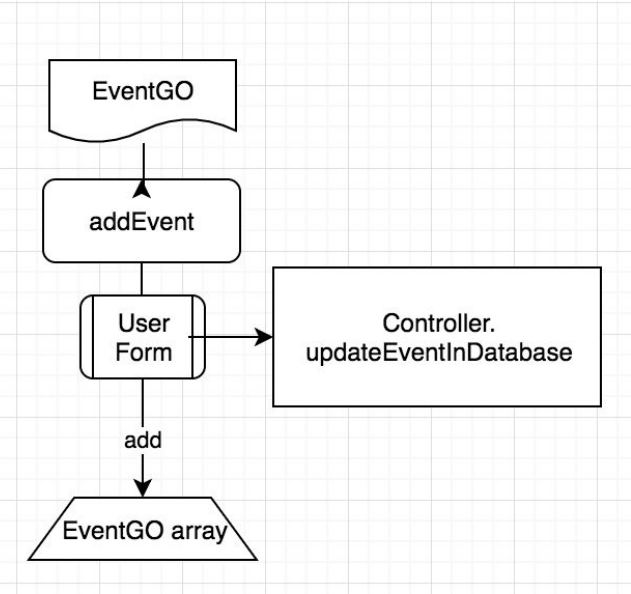
Incremental Testing

Defect #	Description	Severity	How Corrected
1	Application should not crash when attempting to add null values to database	1	There should be no case where a null value is sent to the database handler. Do not add the event and return an empty string.

Regression Testing

Defect #	Description	Severity	How Corrected
1	If the component above passes in the incorrect data for the update event component nothing	1	The functionality is all contained in the component listed there is no functionality based on the return

	should happen		

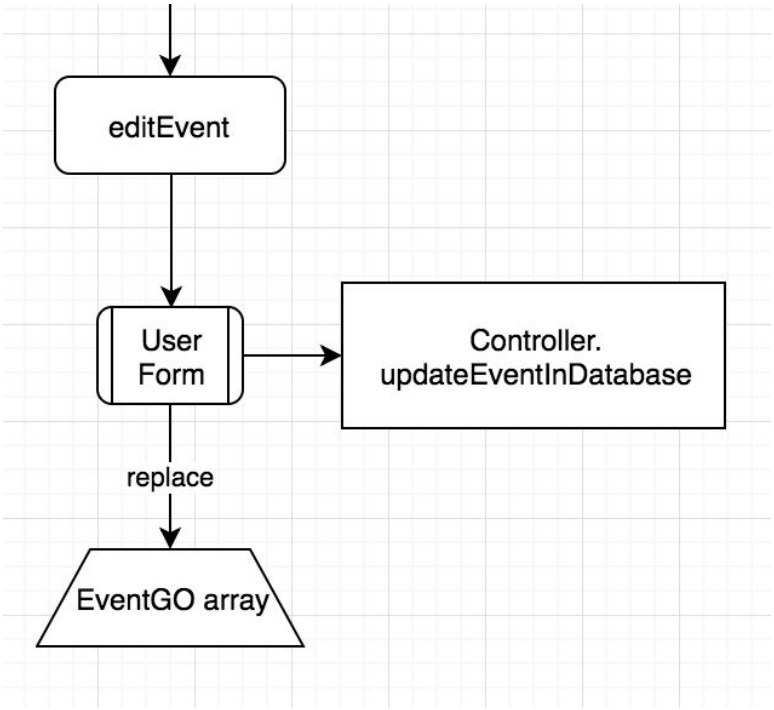


Module	**GUI Component - User Form**

Incremental Testing

After multiple events have been read from the database, the User Form is expected to display all events within the current view (Month or Week) correctly on their appropriate day. On week view, the events should also be in their correct time slot. User Form is also expected to display buttons to navigate to next and previous month/week, a button to add an event, and a button to change between Month View and Week View

Defect #	Description	Severity	How Corrected
1	On weekly view, GUI displays any repeating events multiple times, with each copy being displayed in earlier time slots in the day.	2	Because of time, we decided to fix this in Sprint 2



i.

Module	**GUI Component - Edit Event**

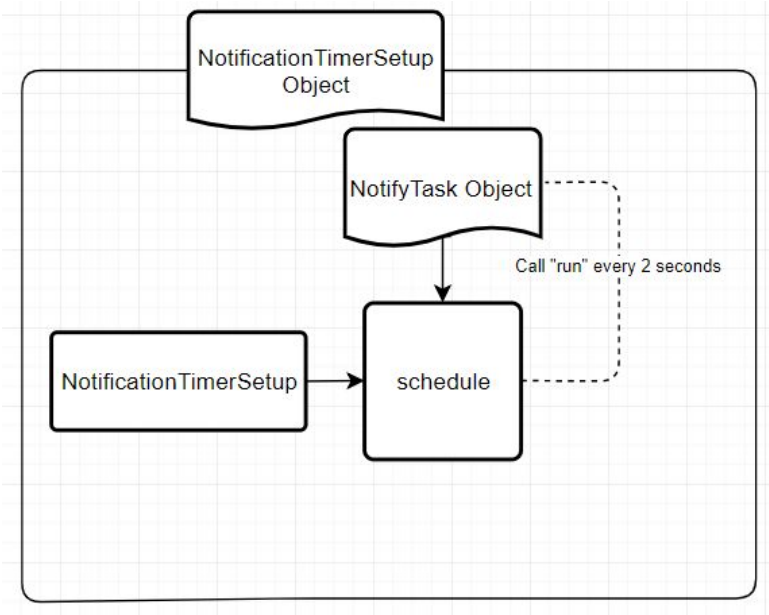
Incremental Testing

Multiple events that were already added to the database were edited using the GUI. We changed their fields, (Name, Date, Duration, etc.) and ensured changes were recorded by database, as well as correctly displayed on the GUI.

Defect #	Description	Severity	How Corrected
1	When editing an event, the Duration field of the dialog box should be set to the numerical value in minutes. It instead displays a different format: “PT50M” instead of “50”	2	Event duration value that is loaded into Duration field is converted into minutes before being converted into String.

Regression Testing

After changes have been made, we tested that editing an event displayed a dialog with the correct Duration display format. When closing the dialog using “Confirm Changes” button, the correct duration was saved. Also, other functionality of the program seem to be working correctly.



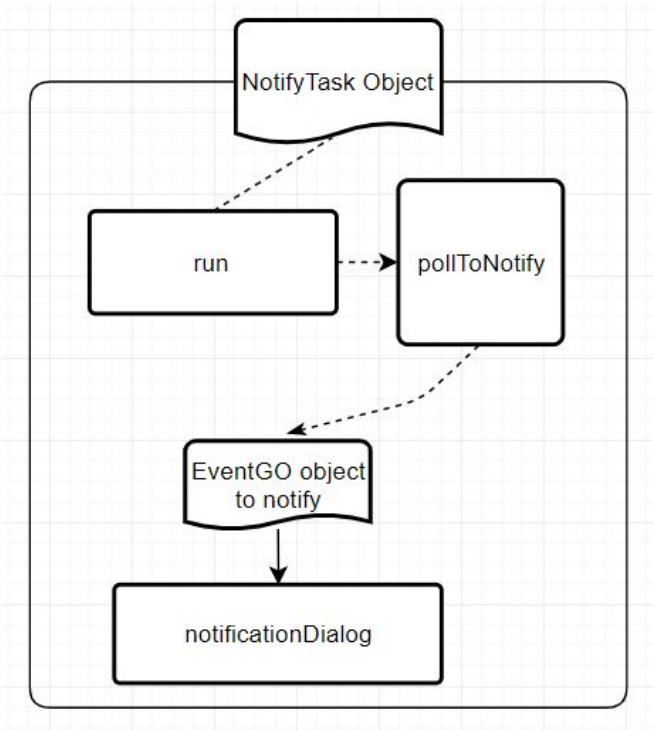
Module	**NotificationTimerSetup Object**

Incremental Testing

Defect #	Description	Severity	How Corrected
1	Notification thread should close when main User Form closes.	3	Ensured timer object closes before main thread stops

Regression Testing

Defect #	Description	Severity	How Corrected
1		1	



Module	**NotifyTask Object**

Incremental Testing

Defect #	Description	Severity	How Corrected
1	Notification “pollToNotify” should be able to properly handle notifying user if events are a day beyond the offset time.	1	Combine “LocalTime” and “LocalDate” objects into a “LocalDateTime” object when polling to handle day wrap around

Regression Testing

Defect #	Description	Severity	How Corrected
1		1	