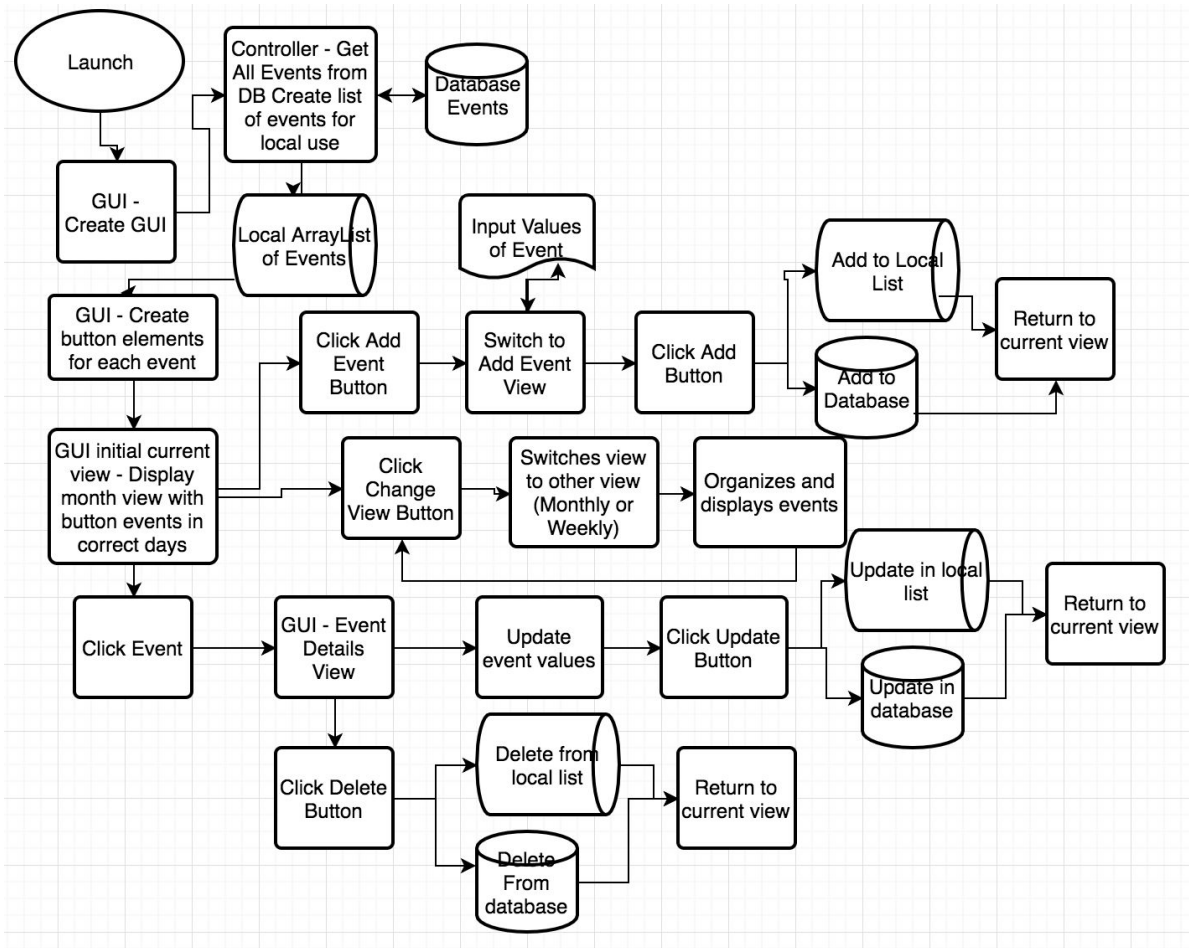


## 1. Classification of Components



a.

- 
- ```

    usecaseDiagram
        participant UC1 as Values of EventGO
        participant UC2 as Values of EventGO
        participant UC3 as String ID of object
        participant UC4 as String User Name
        participant UC5 as String ID of object

        participant UC6 as addEventToDatabase
        participant UC7 as InsertEvent
        participant UC8 as String of the ID created by MongoDB for the event
        participant UC9 as GetEventInDatabase
        participant UC10 as GetEvent
        participant UC11 as Document Object

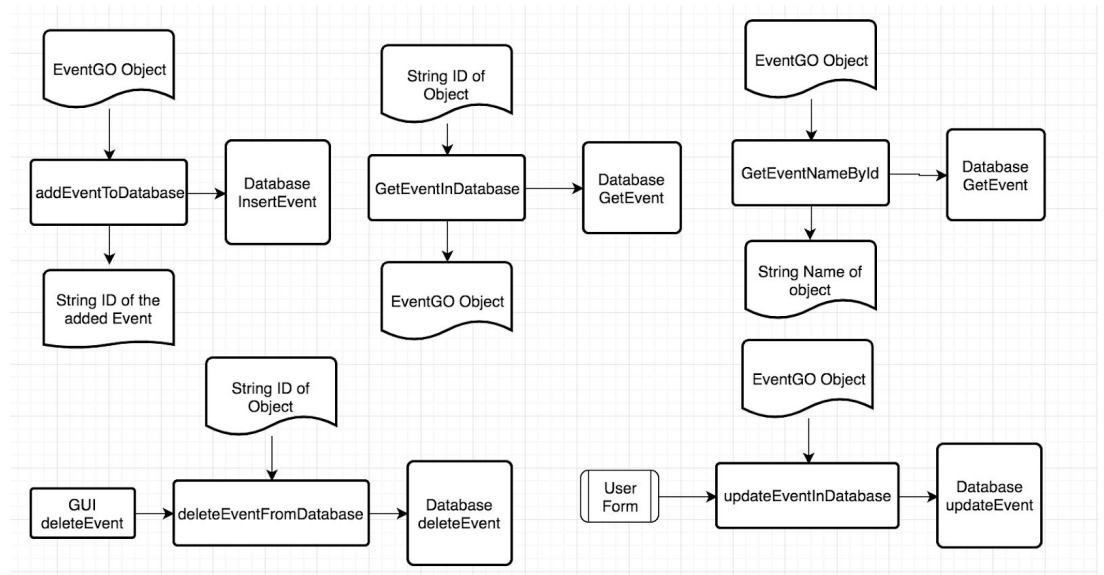
        participant UC12 as updateEventInDatabase
        participant UC13 as updateEvent
        participant UC14 as GetAllEvents
        participant UC15 as GetAllEvents
        participant UC16 as ArrayList of Document Objects

        participant UC17 as deleteEventFromDatabase
        participant UC18 as deleteEvent

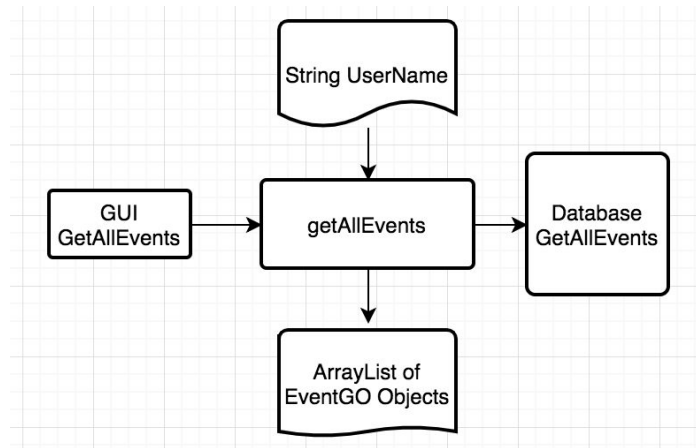
        UC1 --> UC7
        UC2 --> UC13
        UC3 --> UC10
        UC4 --> UC15
        UC5 --> UC18

        UC6 --> UC7
        UC7 --> UC8
        UC9 --> UC10
        UC10 --> UC11
        UC12 --> UC13
        UC14 --> UC15
        UC15 --> UC16
        UC17 --> UC18
    
```
- The diagram illustrates the following use cases and their relationships:
- addEventToDatabase** (Actor) interacts with **InsertEvent** (Boundary).
  - InsertEvent** (Boundary) interacts with **String of the ID created by MongoDB for the event** (Control).
  - String of the ID created by MongoDB for the event** (Control) interacts with **GetEvent** (Boundary).
  - GetEventInDatabase** (Actor) interacts with **GetEvent** (Boundary).
  - GetEventNameById** (Actor) interacts with **GetEvent** (Boundary).
  - GetEvent** (Boundary) interacts with **Document Object** (Control).
  - updateEventInDatabase** (Actor) interacts with **updateEvent** (Boundary).
  - Values of EventGO** (Control) interacts with **updateEvent** (Boundary).
  - String User Name** (Control) interacts with **GetAllEvents** (Boundary).
  - GetAllEvents** (Actor) interacts with **GetAllEvents** (Boundary).
  - GetAllEvents** (Boundary) interacts with **ArrayList of Document Objects** (Control).
  - deleteEventFromDatabase** (Actor) interacts with **deleteEvent** (Boundary).
  - String ID of object** (Control) interacts with **deleteEvent** (Boundary).

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)

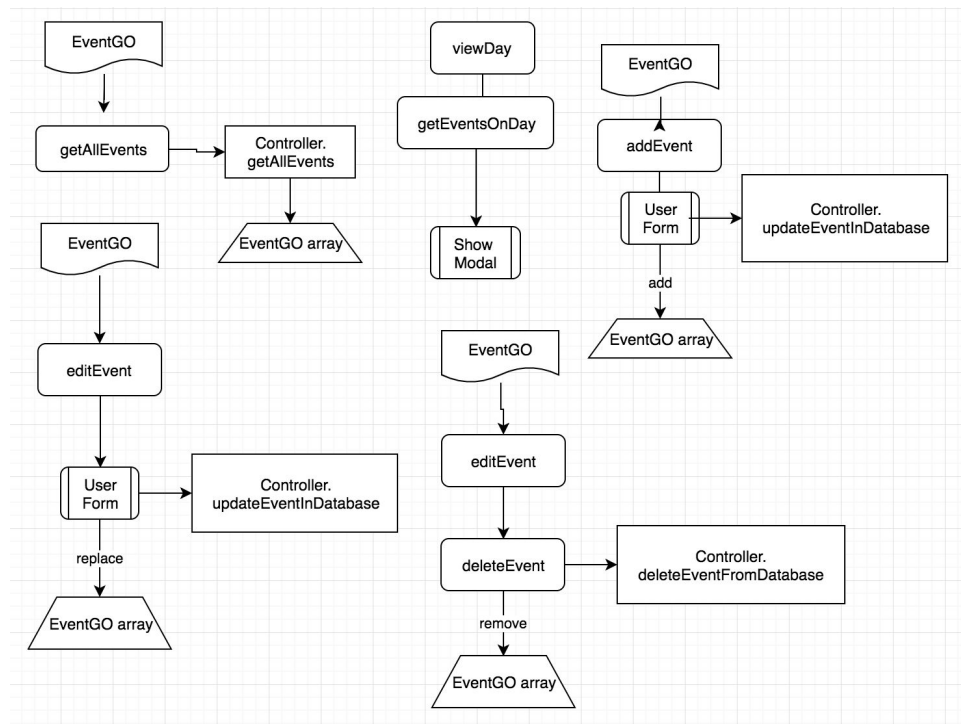


i.



ii.

#### d. GUI Components

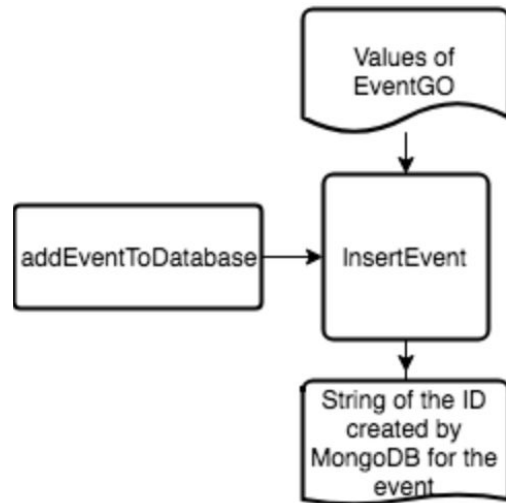


i.

#### e. TopDown vs. BottomUp

- i. 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



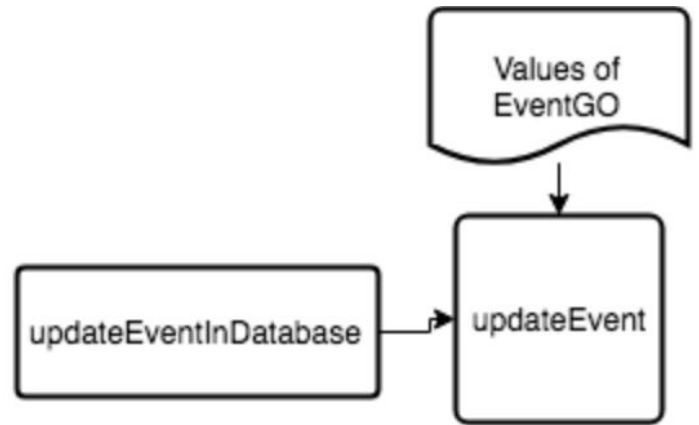
|               |                              |
|---------------|------------------------------|
| <b>Module</b> | <b>Database Insert Event</b> |
|---------------|------------------------------|

### 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. |



|               |                              |
|---------------|------------------------------|
| <b>Module</b> | <b>Database Update Event</b> |
|---------------|------------------------------|

### 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 should happen | 1        | The functionality is all contained in the component listed there is no functionality based on the return |