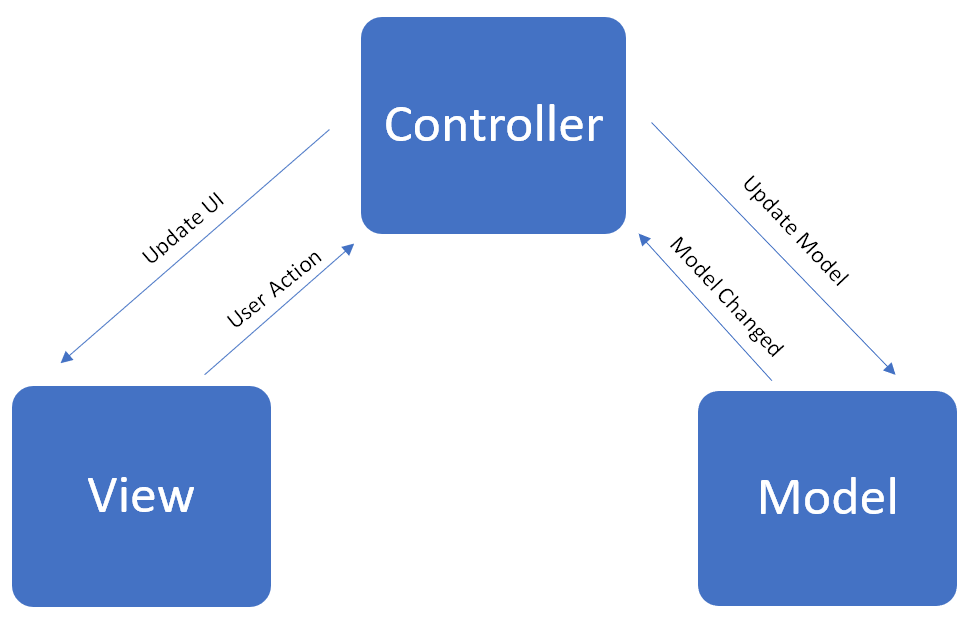
# System design:

## System Architect:

The system uses the MVC architectural pattern, which is divides the application into three interconnected layers these are Model, view, and controller.



## Technologies

Defining the technology to use is vital, many factors must be considered

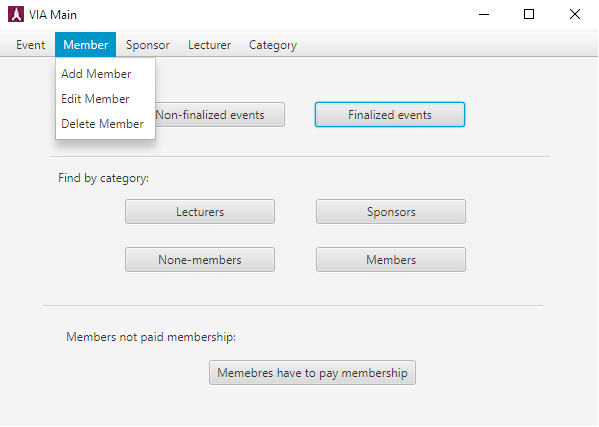
1. The GUI or lack of a GUI: application will have GUI so Java it can be used.
2. The application's functionalities: all the required functions can be done using Java.
3. Deployment of the application: the application must meet the following deployments requirements, and Java supports them
   * Release control
   * Installation
   * Deactivation
   * Uninstallation
   * Update
   * Version tracking
4. Supporting of OOP design patterns: Java support OOP design patterns, so it can be used.

Conclusion:

Based on the previous factors the Java programming language is used.

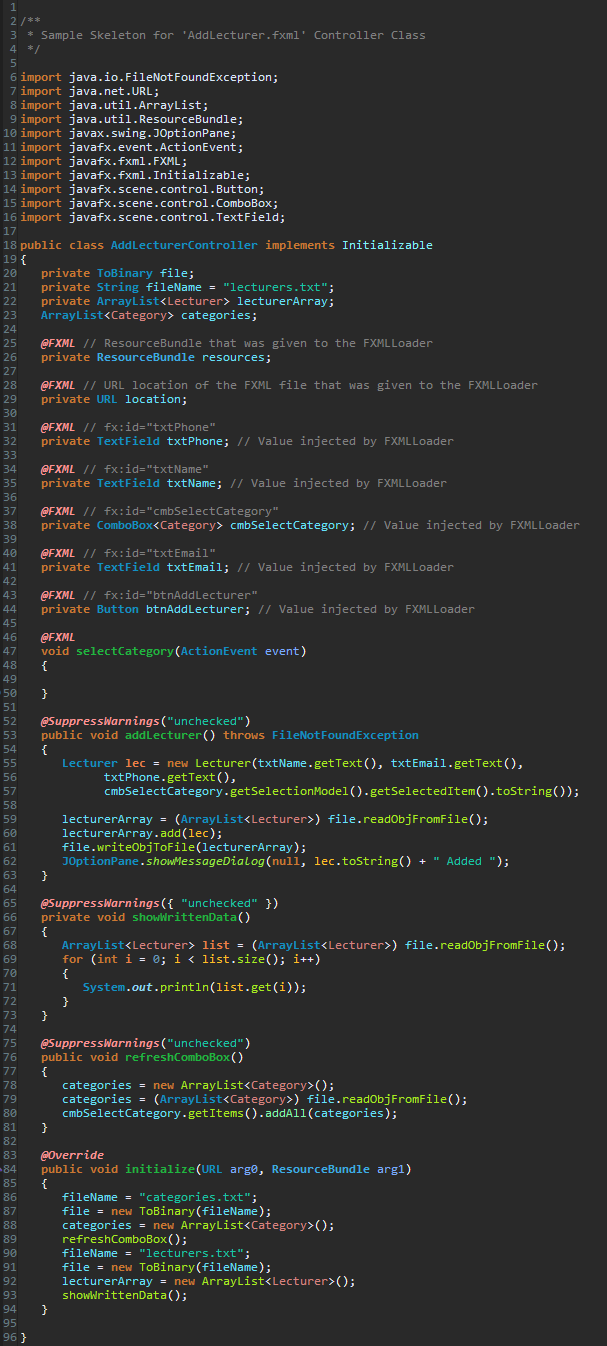
## GUI

The GUI is which technically classes fall in View layer, it has very vital role which is functioning as a mediator between user and other parts of the system, it is consists of menu bar and multiple buttons for the main widow, and for sub windows; the combo box and the table view they have been used significantly.



## Controller classes:

Controller classes are those classes interconnecting the model classes to view classes, so they play a mediator role between GUI and actual models, the class below is an AddLecturerController, it is using a lecturer class as a filed with one-to-many relationship, so array list has been used.

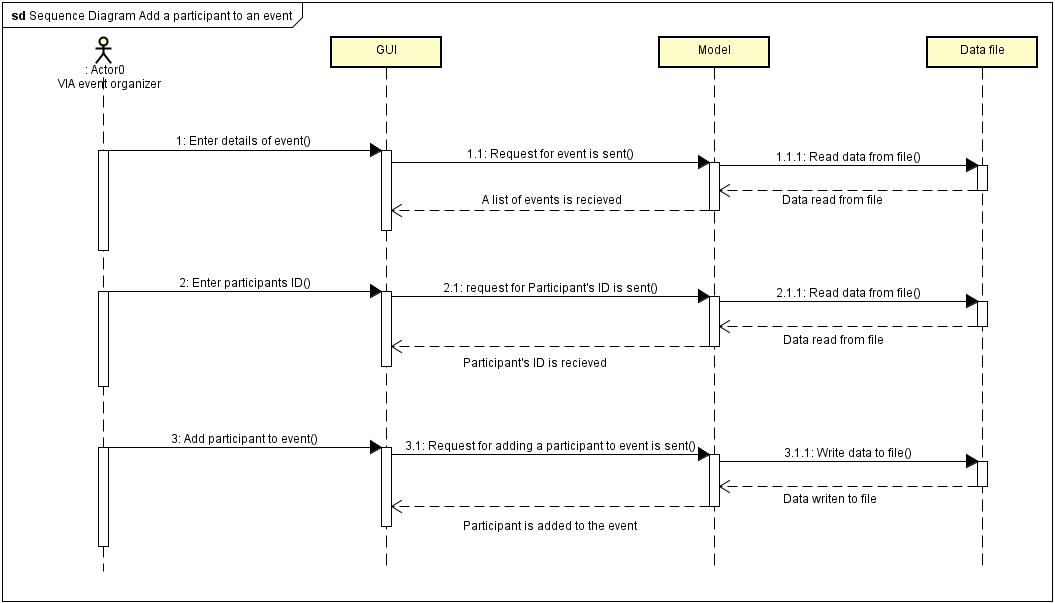


## Model Classes:

Model classes those are representing the blueprints of objects, the one class is a single entity which mean single object has its own fields and methods, the class below is a Lecturer class, so Lecturer object can be instantiated based on it.



## Sequence Diagram:

The purpose of a sequence diagram is to show event in sequential layout that result to get desired outcome, here we are focusing on massages in which order occur, the sequence diagram will communicate what messages are sent between a system's objects as well as the order in which they occur. Below is a sequence diagram for add participant to event, to do this user will find an event to 

## Program