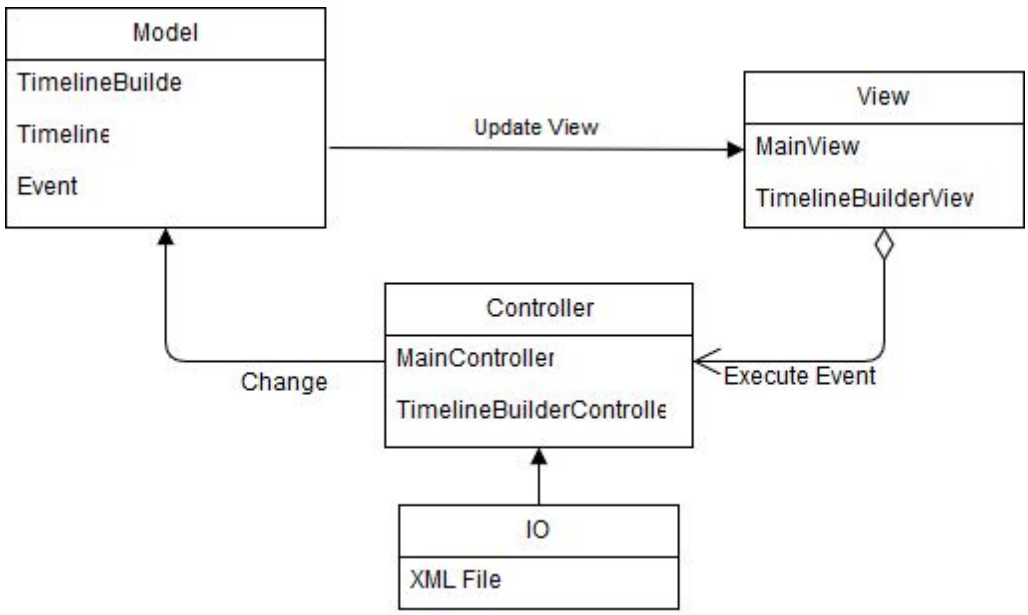
High-level Design

Project in computer science 1DV508

Design Overview

The Timeline Manager will follow an architectural pattern similar to Model-View-Controller (MVC), in which the application is divided into three components: The *Model*, which handles the program's internal state and representation of data, the *View*, which is the graphical interface that the user interacts with, as well as the *Controller*, which controls what happens when the user interacts with the view. This design for the Timeline Manager will also have an IO component, which will manage saving and loading of the timelines.

The main reason for choosing MVC is that it will allow the team to work more independently on different parts of the code, since MVC promotes low coupling between components.



Model

The model manages the loaded timelines and their events. It updates the timeline's state when the controller asks it to and pushes changes to the view, which displays the updated information.

Subcomponents

- **TimelineBuilder** – Responsible for creating new timelines and events
- Objects representing timelines and events

View

The view receives information about the timelines and events contained in the model and displays it to the user. It also displays the interface that the user interacts with (toolbars, buttons, etc.) but does not control what happens when the user interacts with them.

Subcomponents

- **MainView** – Displays the interface as well as the timelines and events
- **TimelineBuilderView** – The popups displayed when the user adds or edits a timeline or adds or edits an event.

Controller

The controller controls what happens when the user interacts with the view. It does this by listening for events from the view, for example a button press, and then performing the steps required to complete the requested action. Each view will have a corresponding controller.

Subcomponents

- **MainController** – Handles user interactions that occur in MainView
- **TimelineBuilderController** – Handles user interactions when adding or editing a timeline/event. Send the information to be added/edited to the model which performs the requested action.

IO

Handles the logic for saving and loading timelines.