Group2 - Time line manager
Design Document

1 Introduction

This document will present the design decisions of the application time line manager. This document can be used as a complement to the Analysis document which focuses more on the requirements of the applications. In this document, you can find both the static and dynamic models such as high-level design, detailed Class diagram, sequence diagrams and activity diagrams. The base for this design is the use cases that can be found in the Analysis document.

2 Design decisions

The main focus on the software design for this application is to make a well-structured design which separate the back-end and front-end. Therefore, a MVC software architecture has been chosen. This is to make it easier to make changes in either part of the software without affecting another part directly. This architecture will also make it easy to develop the software as a team without stalling each other waiting for different parts to be ready. The database solution will be upgraded iteratively to decrease the risks. That means in the very first iterations there will be no database at all, which will in next step be upgraded to a simple JSON-text file database, which later on if time allows will be upgraded to a SQL server as storage for the timelines and task. And if we have good response on this kind of server we might test to change the model to be server based.

3 High-Level Design

In this chapter, you can see the high-level design. In fig1 in the attachments in chapter 8, you can see the connections between the classes. It is chosen a MVC software architectural pattern. The MainWindow class is an anchor for all view classes. The view classes have a controller each that listen for user inputs. These controllers will make changes to the model classes. The most controllers are connected to the timeline model which contains the timelines and the task and also some settings for the timelines. There will also be a help model with a controller and view that will hold information for how to use this software.

4 Detailed Class Diagrams

The class diagrams show the detail static design. Variables and methods for the classes can be carefully studied and will be used for creation of the classes. The getters and setters is not included in the diagrams. The class diagram for the model can be seen in chapter 8: Attachments, illustrated in fig 2.

5 Class connection diagram

The class connection diagram is showing all connections between the classes, it can be used to get a good insight in what approach one can take if needed to have new communication among classes. The class connection diagram can be seen in fig3 in the attachments in chapter 8.

6 Sequence Diagrams

The sequence diagrams show the interactions that take place in each use case. They can all be seen in “chapter 8: Attachments”, illustrated in fig 4 - 9.

7 Activity Diagrams

The activity diagrams show the flow of each use case. They can all be seen in “chapter 8: Attachments”, illustrated in fig 10 - 20.

8 Attachments

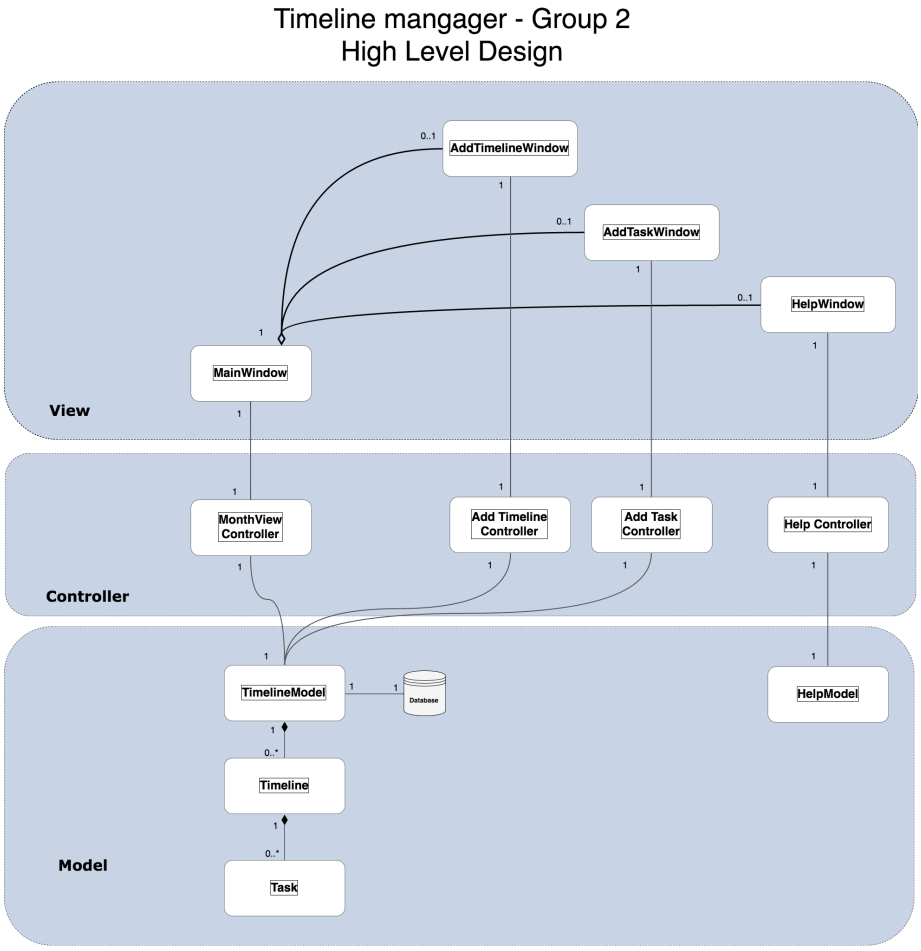

fig1: High level design. A high level design of the timeline manager, it is described in chapter 3.

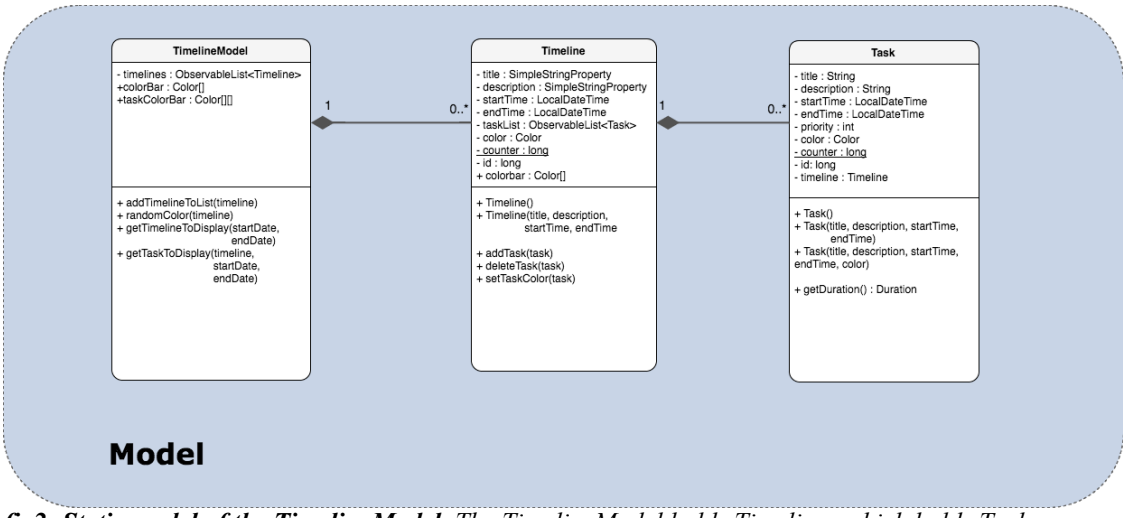

fig2: Static model of the TimelineModel*. The TimelineModel holds Timelines which holds Tasks.*

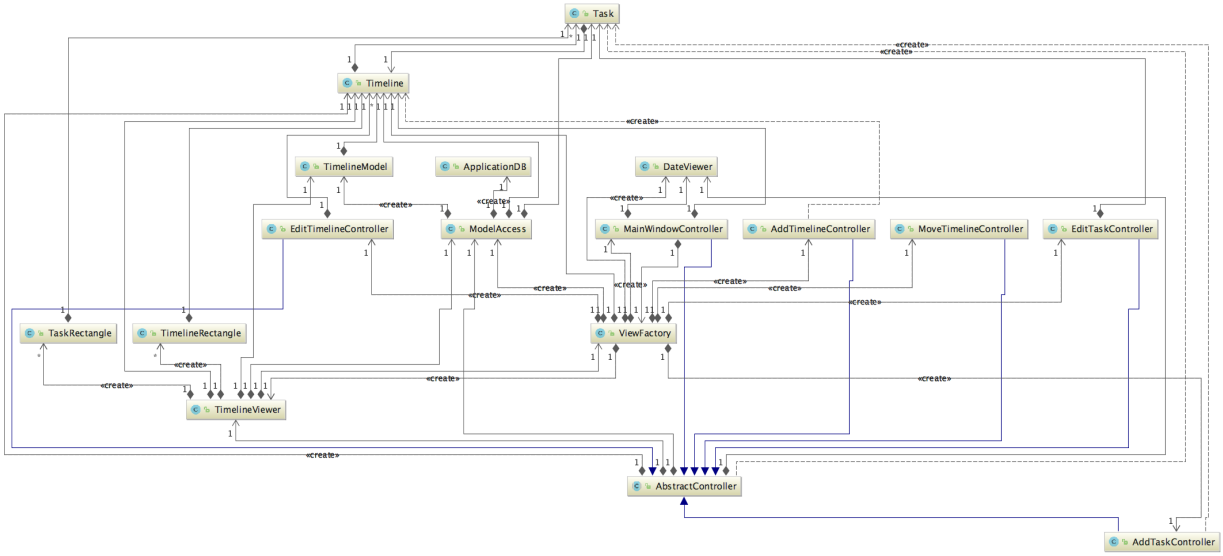

Fig3: Class Connection Diagram. *A static class connection diagram which shows all connections between the classes in the application. By study this you can se that all controllers inherit from the abstract controller which then have communication to model access and timeline viewer which makes it possible for communication among the classes.*

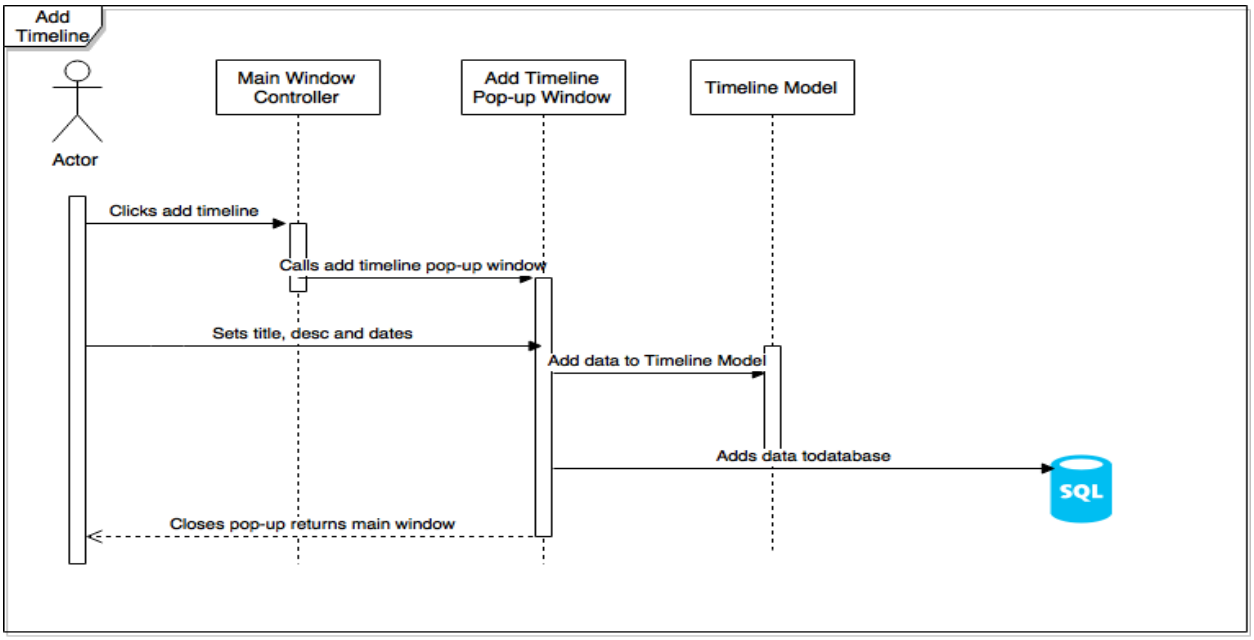

Fig 4: Add Timeline. *When the user clicks the "Add Timeline" button, the command goes to the Main Window Controller, that calls a pop-up window view in which the user inputs the data of the new timeline. After saving, the data is sent to the model class and then back to the Main View Controller to be displayed as a new timeline.*

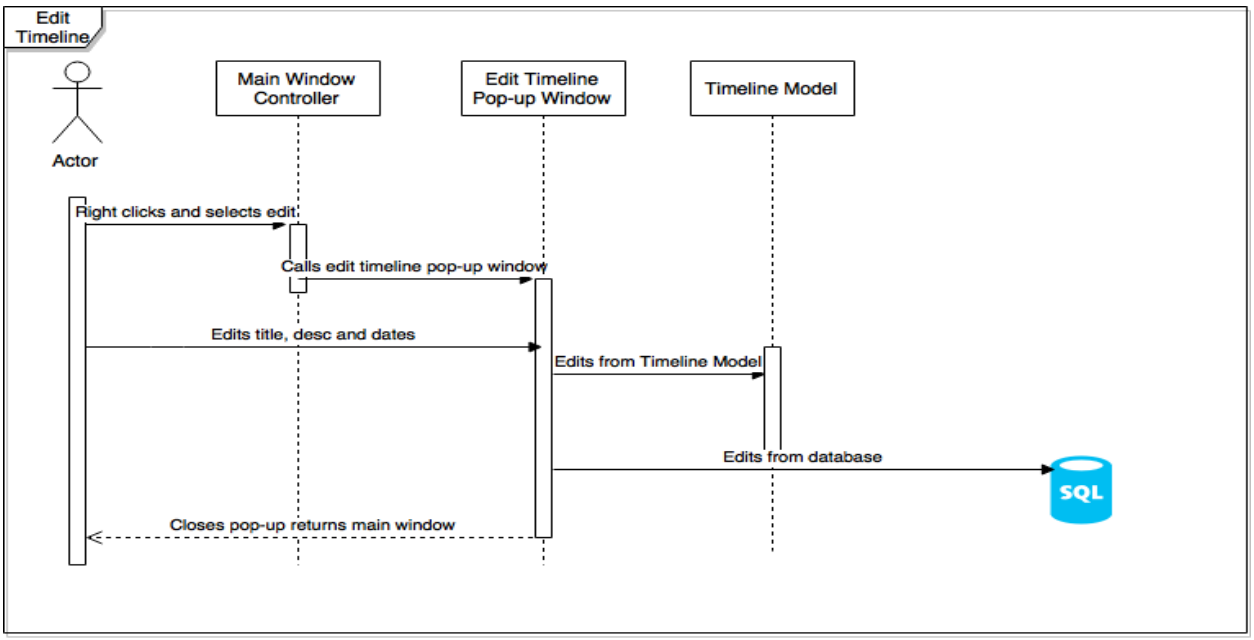

Fig 5. Edit Timeline. *When the user right click a context menu is shown, if user selects edit timeline, a pop-up window is called in which the user modifies the data of the timeline. After saving, the data is sent to the model class as well as the database and then back to the Main View Controller to display the modified timeline.*

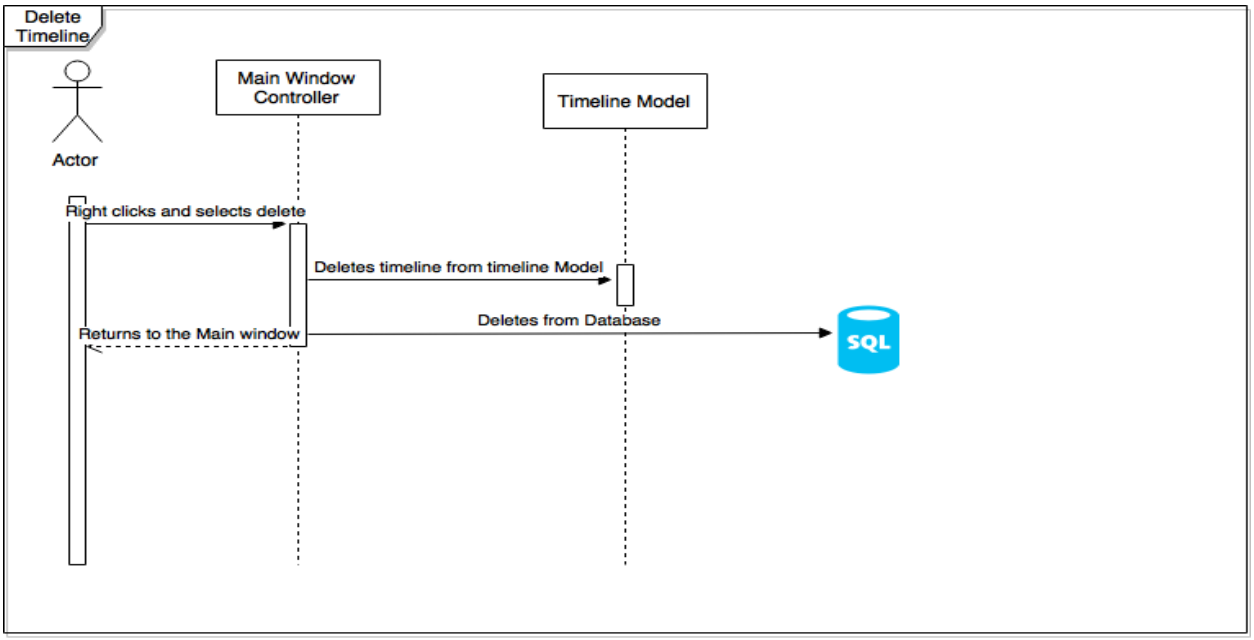

Fig 6: Delete Timeline. *When user right clicks a timeline and chooses delete timeline in the context menu, the timeline with all it's tasks is deleted from the model as well as from the database. Then the user returns to the Main View Controller to display the main window without the deleted timeline.*

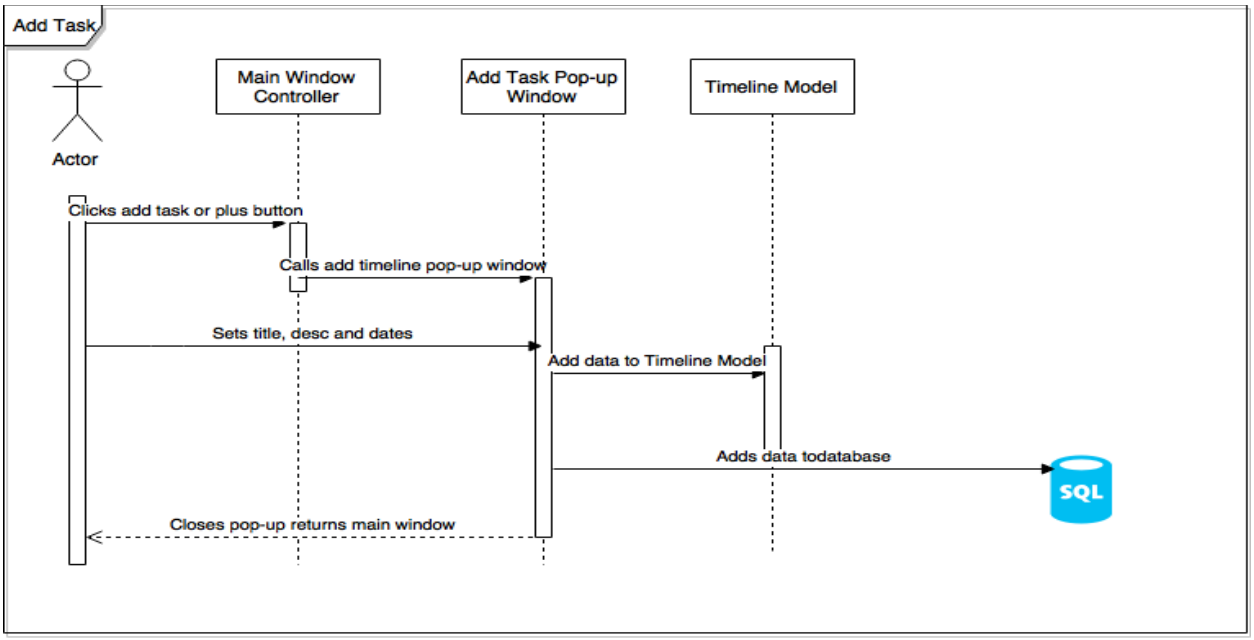

Fig 7: Add Task. *When the user selects a timeline and then clicks the "Add Task" button, the command goes to the Main Window Controller, that calls a pop-up window view in which the user inputs the data of the new task. After saving, the data is sent to the model class and then also updates the database. Then returning to Main View Controller to display the new task for the respective timeline in main window..*

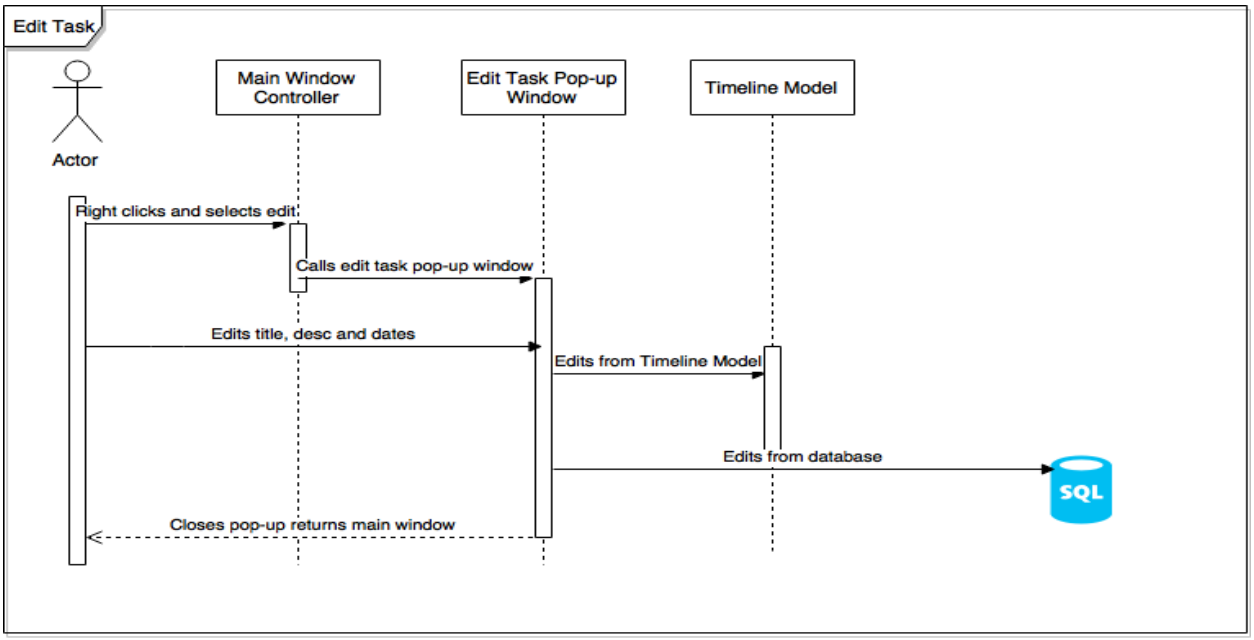

Fig 8: Edit Task. *When the user right click on a task and chooses edit task in the context menu, that calls a pop-up window view in which the user modifies the data of the task. After saving, the data is sent to the model class before sending it to the database and then back to the Main View Controller to display the modified task.*

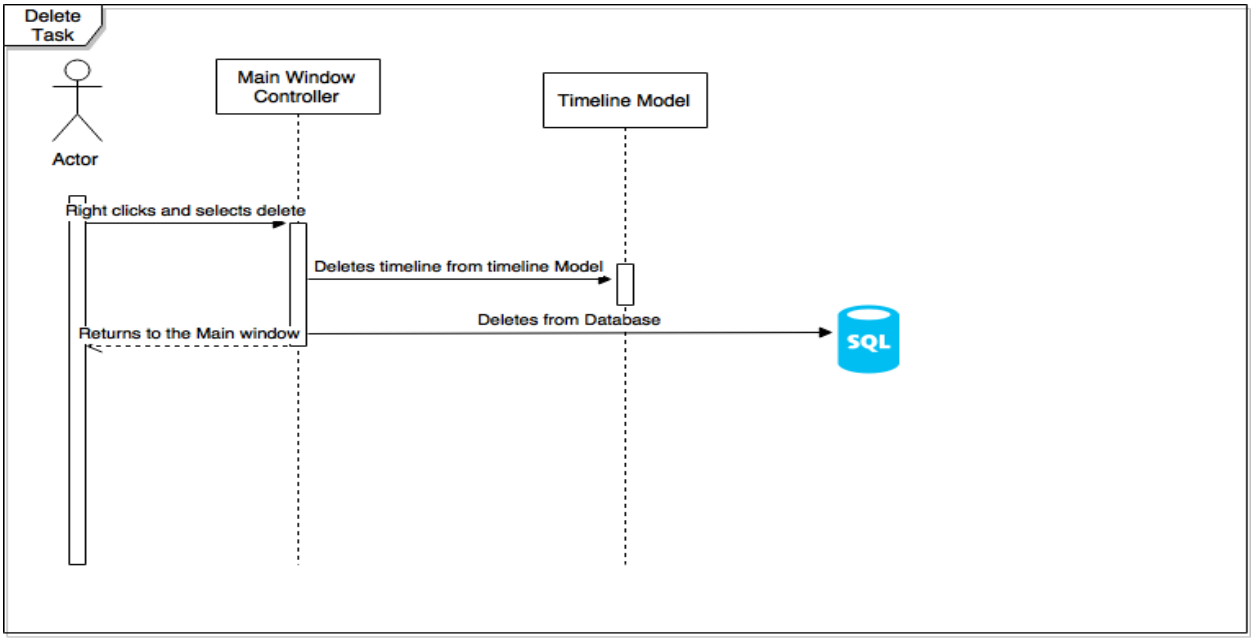

Fig 9: Delete Task. *When the user right clicks on a task and selects the delete option, that send a function to delete the task from the model before deleting it from the database as well. The new model is sent to Main View Controller to display the respective timeline without the deleted task.*

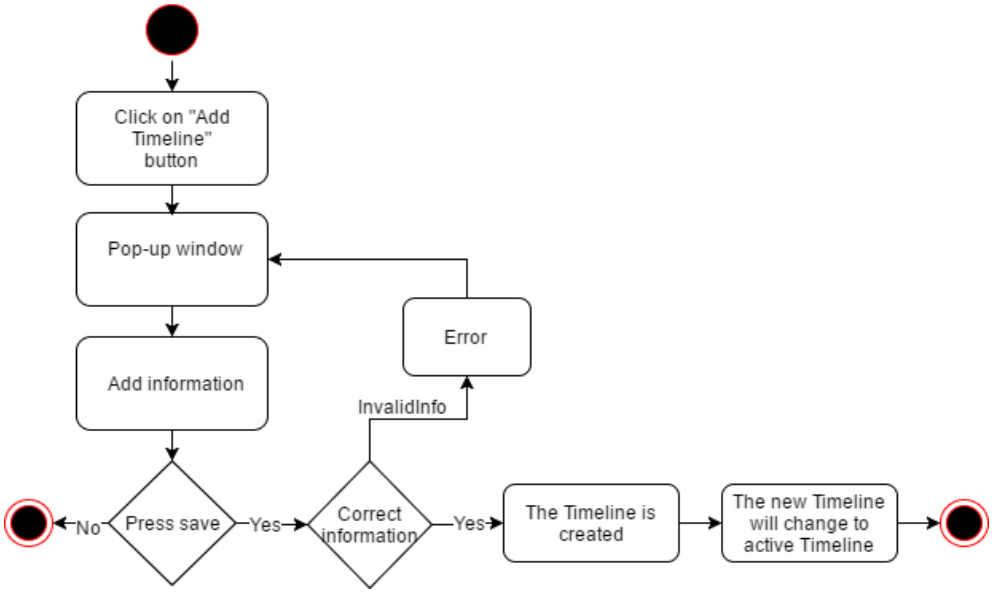

Fig 10: Add new Timeline activity diagram. *A user clicks the "add timeline button" and has to fill in the fields. When user clicks save the timeline, if all fields are correct the timeline will be created in the model and the window will be updated showing the new timeline.*

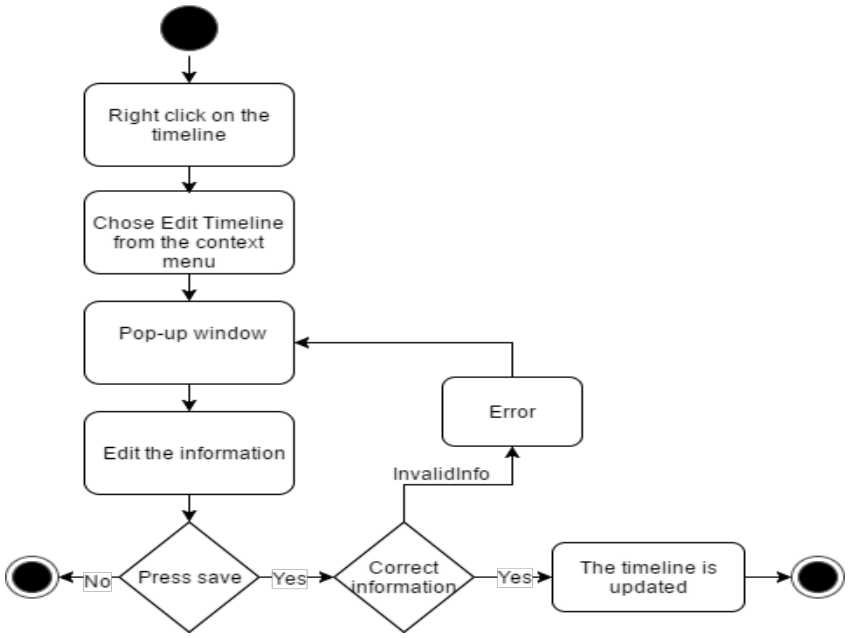

Fig 11: Edit timeline. *If the user double click on a timeline a pop-up window is shown with all info filled in the fields. If the user changes the info and presses save button the information will be checked. If information is valid the timeline will be updated in the database and the view will then be updated as well.*

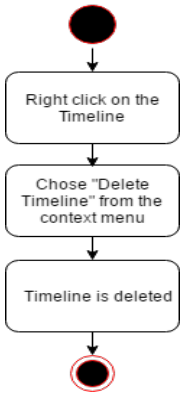

Fig 12: Delete timeline. *A user right clicks on an existing timeline and choosing delete in the context menu. The timeline with all it's task is deleted.*

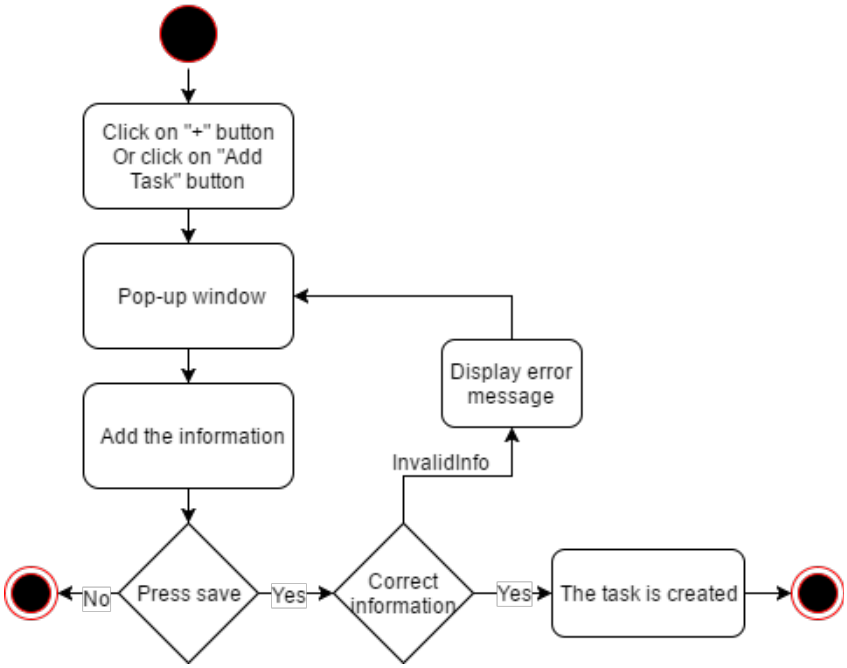

Fig 13: Add task. *The user clicks the "+" button or the "Add Task" button which is only active if there is an active timeline. Then a pop-up window appears with fields to type in the information about the task. If user presses save the information will be checked, if valid it will be saved as a new task in the active timeline.*

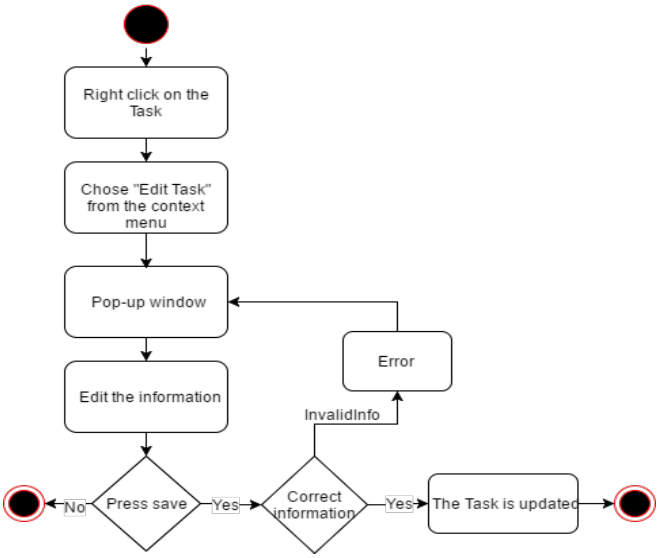

***Fig 14: Edit task**. A user right clicks the task and chooses "Edit Task" in the context menu. A pop-up window with all info filled into the fields is shown. The user can edit these fields and if presses save and the fields are correctly filled in the information will be saved to the task and the view window will be updated.*

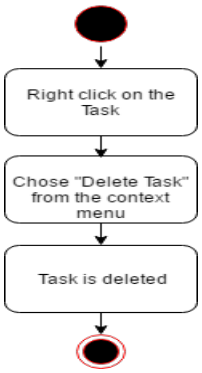

***Fig 15: Delete task**. A user right clicks a task and the chooses delete in the context menu. The task is removed.*

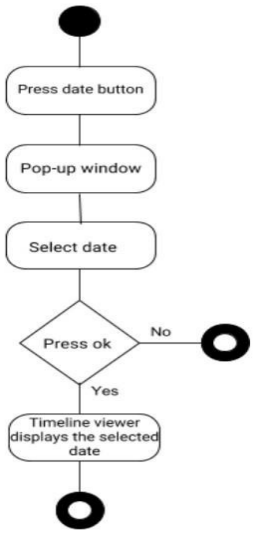

Fig 16: Go to specific date. *The user presses the date button to change date from the pop-up window, if user presses ok the view will change to view the chosen date.*

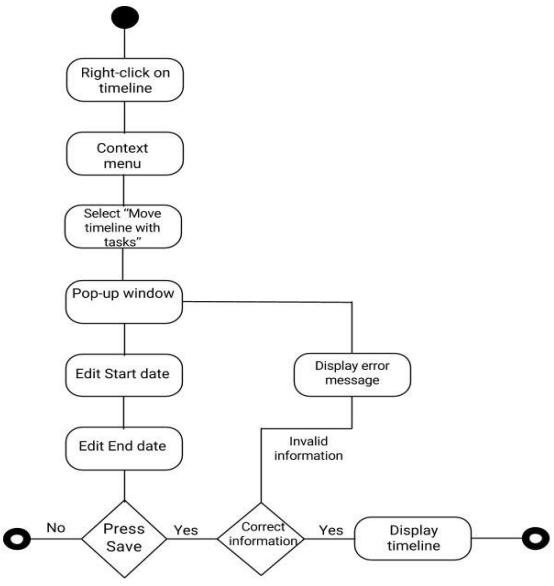

Fig 17: Move timeline with tasks*. When user right click on a timeline and selects Move timeline with tasks a pop-up window appears where user can change either the start date or the end date. The not edited one will update automatically to keep same duration as previously, when user click save the timeline will be moved to the new start and end date and the task will be moved with the same number of days as the timeline.*

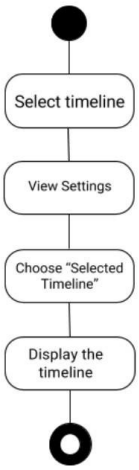

Fig 18: View Selected Timeline. *The user clicks the radio-button "Selected Timeline". Only the currently selected timeline will now be visible.*

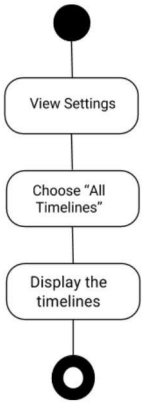

Fig 19: View All Timelines. *The user clicks the radio-button "All Timelines". All timelines within the current viewing will now be visible.*

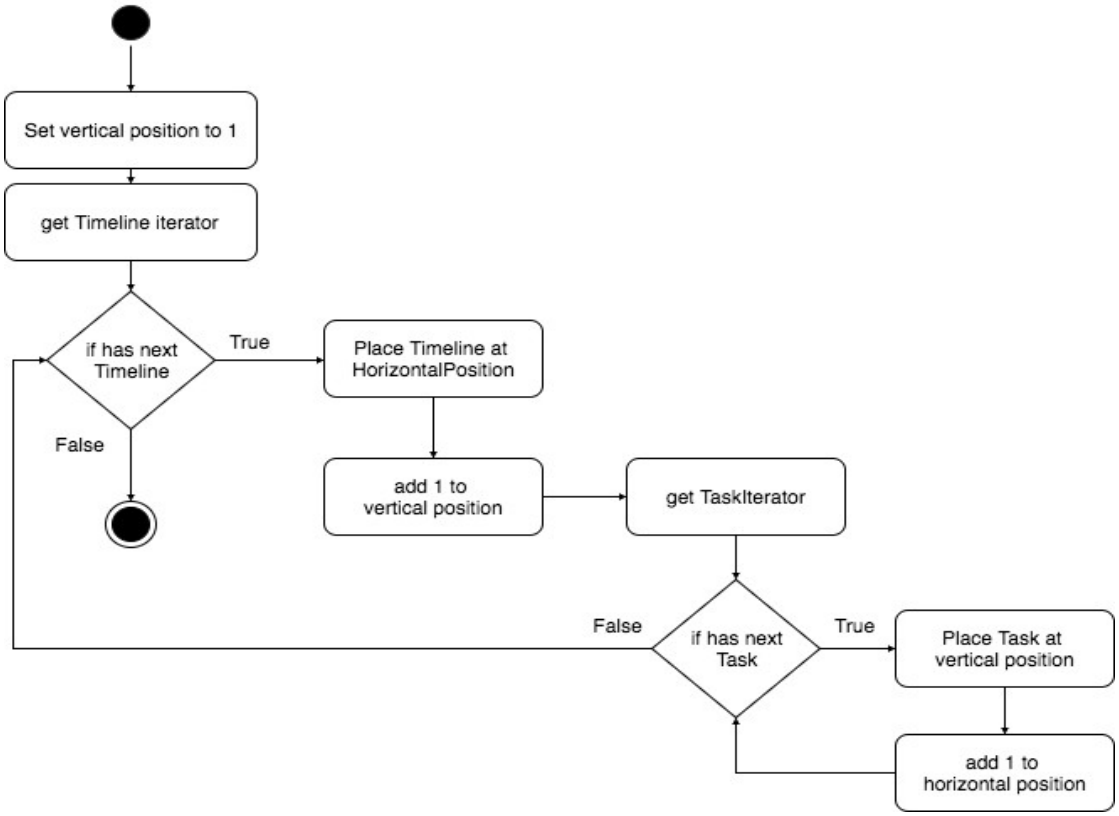

Fig 20: Timeline/Tasks to GUI*. This is the logics used to decide on what row in the Grid Pane the Rectangle visualizing the timeline and the Rectangle visualizing the Task will be put. From the iterator the Timeline / Task will come ordered by the end date primarily and start date secondarily. The horizontal position will be set on simply how many days from the left date in view to timeline start, and the size will be the duration in days from start date to end date multiplied with a fixed pixel size for the width of day in the date-viewer.*