

Project Plan - Timeline Manager

1. Introduction

Our task in this project is to present a fully functional timeline manager. The requirements for the application includes that the user shall be able to create any number of timelines with a start, end date and name provided by the user. The user shall be able to add any number of events to the timeline with name, description, start date and by choice of the user also an end date. The events will show name and duration (if there is one) along with showing all the event information when an event is clicked. Events shall be editable and placed at correct position in the timeline. The user shall be able to save timeline and event and load an existing timeline from chosen file.

The Use-Cases for the final product are:

- Add Timeline
- Edit Timeline
- Delete Timeline
- Add Event
- Edit Event
- Delete Event
- Save Timeline and Event
- Load (Open) Timeline and Event

2. Organization

This project will be developed by a team of 7 people.

- Amelie Löwe - Project Manager
- Caroline Nilsson
- Johan Eriksson
- Stefanos Bampovits
- Indré Kvedaraite
- Patel Pranav
- Aya Kathem

3. Practices

The project will have an iterative practice where all requirements are analyzed, designed, implemented and tested. Important artifacts in the project is the Project Plan, Interface Vision, Requirement List, Use-Case Specifications, UML Diagrams, Test Plan and Test-Cases.

4. Milestones and Objectives

In the creation of the Time Manager the developers along with the Project Manager has decided to use Eclipse as the IDE-tool. The project code is available at GitHub - Group7-1DV508 (<https://github.com/Group7-1DV508/1DV508-group7>). Documents under progress will be available to the team in Google Docs and the finished Documents and artifacts will be added to GitHub.

| Week | Objectives | Start | End |
|------|--|----------|-------------------|
| 12 | <ol style="list-style-type: none">1. Select Project Manager2. Setup GitHub Repository3. List Requirements4. Create Interface Vision5. Identify Use-Cases From Requirements6. Create Project Plan | 17-03-21 | 17-03-26 12:00 |
| 13 | <ol style="list-style-type: none">1. Update Documents Depending of Feedback2. Create Use-Case Specifications3. Finalize Interface Vision4. Create UML Diagrams | 17-03-27 | 17-04-06 12:00 |
| 14 | <ol style="list-style-type: none">1. Update Documents Depending of Feedback2. Finalize UML Diagrams | 17-04-07 | 17-04-12 12:00 |
| 15 | <ol style="list-style-type: none">1. Implementation Plan #12. Detailed Design (Use-Cases: Add Timeline, Add Event)3. Implement Add Timeline, Add Event4. Test-Cases5. JUnit implementation6. Test Result | 17-04-13 | 17-04-19 12:00 |
| 16 | <ol style="list-style-type: none">1. Implementation Plan #22. Detailed Design (Use-Cases: Edit Event, Delete Event)3. Implement Edit Event, Delete Event, Main UI4. Test-Cases5. JUnit implementation6. Test Result | 17-04-20 | 17-04-27 12:00 |
| 17 | <ol style="list-style-type: none">1. Implementation Plan #32. Detailed Design (Use-Cases: Delete Timeline, Save Timeline)3. Implement Delete Timeline, Save | 17-04-28 | 17-05-04 12:00 |

| | | | |
|----|--|----------|-------------------|
| | <ul style="list-style-type: none"> Timeline 4. Test-Cases 5. JUnit implementation 6. Test Result | | |
| 18 | <ul style="list-style-type: none"> 1. Implementation Plan #4 2. Detailed Design (Use-Case: Load Timeline) 3. Implement Load Timeline, Update Visuals 4. Test-Cases 5. JUnit implementation 6. Test Result | 17-05-05 | 17-05-11 12:00 |
| 19 | <ul style="list-style-type: none"> 1. Implementation Plan #5 2. Detailed Design (Overview and update) 3. Implementation: Error handling, Help Function 4. Test-Cases (Overview and update) 5. Junit (Overview and update) 6. Test Result (Overview and update) | 17-05-12 | 17-05-18 12:00 |
| 20 | <ul style="list-style-type: none"> 1. Overview documentation 2. Error handling 3. UI Implementation, Help Function 4. Distribution Plan 5. Prepare Presentation. | 17-05-19 | 17-05-23 12:00 |