Test Cases Document

Changelog

4/5	Re-work on total document. Expanded the existing test cases to test each functionality, and added traceability to test cases.
9/5	Added exceptional cases.
10/5	Updated with progress on current cases.
15/5	Updated with more test cases.
18/5	Updated with more progress
26/5	Updated with reports and progress.

Introduction

This document contains the test cases for the software “Timeline Manager”. This document will be divided in 3 parts.

The first part consists of the unit tests and integration tests. This part shows the unit and integration tests, and the steps to perform the tests. These tests are done to make sure if the core of the software is working as intended or not.

The second part contains the result of the system testing. This part contains a report that states which of the requirements that have been met.

The last part contains the results from the delivery testing. In this part we determine if the system is ready for releasing, and if it performs the requested functions properly and exactly as they are designed.

Unit and integration tests

Test case 1

Description: This test is to make sure that our logic and timeline class is working properly. It was made using JUnit, and can be found in the test package as TimelineTest.

Requirement TM-C1	**Create one or more timelines with separate start and end time.**
Test type:	Unit test
Precondition:	Test has opened the testpackage with the testfiles.
Notes:	A timeline object has at least a title, a start time and an end time. A timeline object should not be created when one of these is missing.
Post condition:	Timeline object is created with necessary details.

	Steps to execute	**Expected response**	**Pass/fail**
1.	Run the testfile TimelineTest from the testpackage in eclipse.	Green for all the functions for this class.	**Pass**

Test case 2

Description: This test is to make sure that the timelines we create can be displayed properly in the application or not. It ensures if the visual representation and creation of timelines are correct or not.

Requirement TM-C1	**Create one or more timelines with/without separate start and end time, showing a graphical representation to the user**
Test type:	Integration test
Precondition:	User has opened the application and sees the start screen.
Notes:	A timeline object has at least a title, a start time and an end time. A timeline object should not be created when one of these is missing.
Post condition:	Timeline object is created and user sees a timeline, represented by a bar, where at the beginning and end of the bar are shown the specific dates.

	Steps to execute	**Expected response**	**Pass/fail**
1.	Click new timeline button.	A pop up window opens with fields to enter timeline information	**Pass**
2.	A title, a start date and an end date is entered in the pop up window	Pop up window holds all necessary information	**Pass**
3.	Button ‘Save’ is clicked.	Timeline data is checked for errors, Timeline object is created, Pop-up window closed.	**Pass**

	Exceptional cases / User errors	**Expected response**	**Pass/fail**
1.	Entering no title for the timeline	User gets an error message. User is then returned to the previous screen.	**Pass**
2.	Entering an invalid time.	User gets an error message. User is then returned to the previous screen.	**Pass**
3.	Not selecting a start and/or end time for the timeline.	User gets an error message. User is then returned to the previous screen.	**Pass**

Test case 3

Description:

This test ensures the functionality of the event class and the functions of the class. The test class can be found as EventTest in the test package.

Requirement TM-C3	Add a non-durational event to the timeline.
Test type:	Unit test
Precondition:	Instantiated timeline object
Notes:	An event at least has a title and a date, specifying the occurrence of the event.
Post condition:	An event is created and tied to the timeline. The event should be part of the timeline object and is able to be retrieved from the timeline.

	Steps to execute	**Expected response**	**Pass/fail**
1.	Run the testfile EventTest located in the testpackage.	Green for all the function of this class.	**Pass**

Test case 4

Description:

This test is to ensure that the graphical representation of creating and displaying non-durational events are working correctly.

Requirement TM-C3	Add a non-durational event to the timeline, which is shown correctly in the graphical user interface.
Test type:	Integration test
Precondition:	User has opened the application and sees the start screen, containing one timeline.
Notes:	An event is created and tied to the timeline. The event should be part of the timeline object and is able to be retrieved from the timeline. The event is visible on the timeline in the User Interface.
Post conditions	An event at least has a title and a date, specifying the occurrence of the event.

	Steps to execute	**Expected response**	**Pass/fail**
1.	In the timeline, click button to create new event.	A pop up window opens with fields to enter new event information.	**Pass**
2.	Event details are entered in the pop up window	Pop up window holds all necessary information	**Pass**
3.	Button ‘Save’ is clicked.	Event data is checked for errors, Event object is created and tied to the timeline, Pop-up window closed. Return to main screen where timeline is shown including the just added event.	**Pass**

	Exceptional cases / User errors	**Expected response**	**Pass/fail**
1.	User tries to create an event without title.	User gets an error message. User is then returned to the previous screen.	**Pass**
2.	User tries to create an event without entering the date.	User gets an error message. User is then returned to the previous screen.	**Pass**

3.	User tries to create an event outside of the timespan of the timeline	User gets an error message. User is then returned to the previous screen.	**Pass**

Test case 5

Description: This test is to ensure that the graphical representation of creating and displaying durational events are working correctly.

Testing: TM-C4	Add a durational event to the timeline.
Test type:	Integration test
Precondition:	User has opened the application and sees the start screen, containing one timeline.
Notes:	An event at least has a title and a date, specifying the occurrence of the event.
Post condition:	An event is created and tied to the timeline. The event should be a part of the timeline object, and should be able to be retrieved from the timeline.

	Steps to execute	**Expected response**	**Pass/fail**
1.	In the timeline, click button to create new event.	A pop up window opens with fields to enter new event information.	**Pass**
2.	Event details are entered in the pop up window	Pop up window holds all necessary information is shown.	**Pass**
3.	Button ‘Save’ is clicked.	Event data is checked for errors, Event object is created and tied to the timeline, Pop-up window closed.	**Pass**

	Exceptional cases / User errors	**Expected response**	**Pass/fail**
1.	User tries to create an event without title.	User gets an error message. User is then returned to the previous screen.	**Pass**
2.	User tries to create an event without entering the date.	User gets an error message. User is then returned to the previous screen.	**Pass**
3.	User tries to create an event outside of the timespan of the timeline	User gets an error message. User is then returned to the previous screen.	**Pass**

Test case 6

Description: This test is to ensure that a timeline object can hold any numbers of events. This test can be found in EventTest in the test package.

Testing: TM-C2	Add any number of events to the timeline.
Test type:	Unit test
Precondition:	Tester has eclipse running.
Notes:	An event at least has a title and their dates, ignoring whether an event is durational or not.
Post condition:	Events are created with their details. (User sees the created events tied to the timeline)

	Steps to execute	**Expected response**	**Pass/fail**
1.	Run the testfile EventTest located in the test package.	Green for all the function of this class.	**Pass**

Test case 7

Description: This test is designed to test the delete function for timelines. This test is made to guarantee that timelines that we create can also be deleted.

Testing: TM-C6	Delete a timeline
Test type:	Integration test
Precondition:	User has opened the application and sees the start screen, containing one timeline.
Notes:	A timeline with start and end date is defined.
Post condition:	The selected timeline has been removed.

	Steps to execute	**Expected response**	**Pass/fail**
1.	User selects a timeline and presses “delete timeline”	Timeline is deleted and removed from view.	**Pass**

Test case 8

Description: This test is designed to test the delete function for events. This test is made to guarantee that events that we create and link to timelines can also be deleted.

Testing: TM-C6	Delete an event.
Test type:	Integration test
Precondition:	User has opened the application and has a timeline open containing an already defined event.
Notes:	A timeline with start and end date is defined. An event at least has a title and their dates, ignoring whether an event is durational or not.
Post condition:	The selected event has been removed.

	Steps to execute	**Expected response**	**Pass/fail**
1.	User right-clicks on an event.	User gets a menu with the “delete” option.	**Pass**
2.	User presses “delete”	User gets a confirmation window.	**Pass**
3.	User presses “Yes”	Event is deleted and removed from the timeline.	**Pass**

Test case 9

Description: This test is to ensure that the save function of timelines work. The goal is to create a functional file that can store all the dates and events to be loaded later on.

Testing: TM-GR1	Save timeline
Test type:	Integration test
Precondition:	User has opened the application and sees the start screen, containing one timeline.
Notes:	A timeline with start and end date is defined.
Post condition:	The selected timeline with attached events are saved to a file.

	Steps to execute	**Expected response**	**Pass/fail**
1.	User presses “Export to file”	User gets to select the file location and names the file.	**Pass**
2.	User presses “Save”	The timeline is saved to a file.	**Pass**

Test case 10

Description: This test ensures that timelines that have been exported, can be loaded and displayed correctly with all the events linked to them.

Testing: TM-GR1	Load timeline
Test type:	Integration test
Precondition:	User has opened the application and sees the start screen or main window.
Notes:	There will be no difference even if there already is a timeline loaded.
Post condition:	The selected timeline with attached events are loaded and displayed.

	Steps to execute	**Expected response**	**Pass/fail**
1.	User presses “Load timeline”	User gets to select the file location and selects the file to load.	**Pass**
2.	User presses “Load”	The timeline is loaded and displayed.	**Pass**