

Test Plan

Introduction

This document includes an overview of what will be tested and how it will be tested throughout the project. The tests will be held on a weekly basis to find and fix possible faults. This document will be updated regularly, as new test cases and tests will be performed.

Objectives

This Test Plan document supports the following objectives:

- Identify existing project information and the features that should be tested.
- List the recommended test requirements.
- Identify the required resources and provide an estimate of the test efforts.
- List the deliverable elements of the test activities.

Purpose

The main purpose is to describe the various testing strategies and tools used for the whole App testing of this project.

Scope

The app will be tested mainly for correct functionality. The tests will be run on a weekly basis and all faults will be removed before the next iteration starts.

To be tested:

- Add timeline
- Modify timeline
- Remove timeline
- Add event
- Modify event
- Remove event
- Save timeline
- Load timeline

Testing Process

- Creating Test cases for every Use case we have and continually adding Test cases When adding new features and create new Use cases.
- Creating Tests so that the test cases can be tested individually.
- Executing Test Cases when the app is updated and reloaded.

Test Strategies

Unit test

To be tested: add/remove Timeline, add/remove Event

Manual tests

All requirements transformed in Use Cases will be tested to control if everything is working as expected.

Tools

JUnit library for Java.

Test Schedule

Planning:

The Test Plan document has to be finished.

Design Phase:

Test cases need to be written.

Code Complete function:

Unit and manual testing needs to be finished when functionality is implemented.

Feature Complete:

When all features have been implemented, last faults should be found and solved and re-test.