Testing Document and Specification

Test Plan

Smart Mirror Team 5

Chafiki Nabil Salzani Diego Corrizzato Leonardo

Introduction

This document outlines the test plan for the Smart Mirror project. As outlined in the project requirements document, this system should provide a simple approach to the tools a user needs to get started with the necessary information for the day. This system will also provide a mechanism for discussion between users and any requests and developers. The testing activities discussed in this document will verify that the software meets the customer's needs, verifying that the requirements of this system, as outlined in the Requirements Document, are met.

This exercise essentially discovers faults and failures in the software. Faults lead to failures and testing is necessary to eliminate these errors in the software development process. The following document outlines the testing procedures for the Smart Mirror project.

Terminology

Throughout this document the terms user, developer, widget, system, site will be used frequently therefore, formal definitions will be given.

System - the database and bulletin board that supports the backend of the website

User - people who will exclusively use only the front-end of the website

Developer - people who develop the software and with whom users can make requests

Widget - utility that helps the user in the graphical interface that allows to display the necessary information

Site - the front-end website (main page)

Items Tested

Items that will be tested during the testing phase as laid out by the Macro Project Plan will be but are not limited to:

- Ability to display the correct geographical position of the user
- Ability to display the required time correctly
- Ability to display the weather correctly
- Ability to display the widgets correctly in their position on the grid
- Allow the user to interface with the graphical part of the project

Items Not Tested

There are features that will not be included in the current testing procedure. This does not mean that these features will not be implemented, but that they have not been implemented and are not available for testing. Those features include but are not limited to:

- Locating the user's position automatically
- Querying the Database

Approach

The general method of this test procedure is manual testing of the system. Each test case created will have a direct link to the requirements as indicated in the requirements document. Test cases that include similar Feature methods will be tested together. Examples of these features include creating a user in a specific geographic location, accessing an electronic log with a test user to verify correct data reception, news that actually happened and that the weather corresponds to reality.

Test cases such as these test the adaptability characteristics of the system along with the ability to send information to the user. Each test case will test the security feature with invalid and valid data (weather and school time) to ensure that valid user requirements for these features are met.

Each widget intended for time, weather, news and school hours will be tested together to see the compliance of the project, but in cases of separate tests. The ability of the system to produce results that an individual can visually validate will be determined by the correct display of the information received by the code in the back-end.

Item Pass/Fail Criteria

The minimum requirements for this software system were set out in the requirements document and the Project Plan outlined what the creators of the software considered the success of the project.

Implemented features that meet the requirements set by the client, i.e. the functionality does what the user wants it to do with very little difficulty, passes the test procedure. The difficulty, as used here, is determined by the user's understanding and ability to use the functionality with little or no training.

Features that contain important defects and fail the testing procedure will be recorded through an incident report and passed to the developer for investigation and review.

Test Deliverables

In addition to the Test Plan, other test deliverables include the Test Specification which outlines the specific test cases and expected results of each test, and Test Reports which are made up of incidents, defects and changes in the implementation of the various widgets or interface that present themselves to the user.

Testing Tasks

The following list the testing deliverables and the activities required to produce the deliverable.

Deliverables	**Activities**
Test Plan	Analyze Requirements for System Features
Test Specifications	Analyze Requirements
Test reports	Implement Test Cases as Outlined by the Test Specifications

- Determine Testable/Non-Testable Features
- Develop Approach/Method for testing
- Develop Schedule for Testing
- Define Test Cases for Testable Features as Outlined by the Test Plan
- Determine Severity of Incidents and Defects
- Determine Changes that need to be made to system
- Document and send the change request to the developer according to your needs