Software Requirements Specification

For

Smart Mirror

December 2, 2019

Team 5

Prepared by:

Salzani Diego

Revision History

Version	Date	Name	Description
1	12/02/2019	Diego Salzani	Initial Document

Introduction

Overview

The Smart Mirror project was born triyng to solve every day life probrems and to make easier some common actions.

It shows informations on a mirror, combining visual harmony with technology.

The goal is to deliver information about everyday news, university schedule and weather thought an aesthetic furniture, mirrors.

Goals and Objectives

The goal is to deliver information about everyday news, university schedule and weather thought an aesthetic furniture, mirrors.

Definitions

Smart Mirror – The finished product made by wood, an lcd screen and one raspberry pi

Widgets – Micro programs that are digitally placed within the mirror area range.

Weather Widget – Widget showing live weather conditions using texts and images.

News Healine Widget – Widget that give us updated news from major information magazines and websites.

Univerity Schedule Widget – Shows a table of daily school subjects.

Clock Widget – Showing current time in Hours, minutes and seconds. Also showing current date (day/month/year).

General Design Constraint

Application Enviroment

Smart Mirror software runs with Linux OS on a micro computer, Raspberry PI.

The programs itself is written with python using the graphic interface TKinter.

User Charateristics

The user can't directly interact with the widgets, they only give informations. User can see his reflection on the mirror and in the same time read the info.

Mandated Constraints

Having an internet connection to get and update the data displayed and an electrical source to power the mirror.

Nonfunctional Requirements

Operational Requirements

The system itself has no user configuration required, the product comes already set up with all the information needed by the user.

Secutiry Requirements

For any changing to the software the user can contact the email support the will remotely set the product with new information sources.

Documentation

The instruction manual can be downloaded from the link provided in the box

External Interface

User Interface

The user interface is designed to be easy to read but also beautiful to see.

Software Interface

The software interface is created with Python .

Functional Requirements

Required Features

Use Case: 1

Description: Smart Mirror startup

Actors: University student

Basic Path

1. User plug in the power cable of the mirror
2. System starts up and show a "Welcome to smart mirror" banner
3. After 5 second all the info are showed on the mirror
4. Every time the data is changed on the server the info displayed on the mirror will automatically change.
5. User press the power off button and shuts down the poduct.

Use Case: 2

Description: Changing Smart Mirror informations sources

Actors: University student

Basic Path

1. User changes his residence
2. System is not showing the correct wather
3. User contat the support e-mail to update the weather sources
4. User type in the email his new address
5. Smart Mirror support via remote changes the software with the right sources
6. User reboot the Mirror