Smart Mirror

System Requirements Specifications

Version 1.3

12/12/2017

Version History

Version #	Implemented By	Revision Date	Reason
1.0	Austin Nantkes	10/22/17	Initial Document
1.1	Zena Abulhab	12/4/17	Fixing Comments
1.2	Austin Nantkes	12/5/17	Fixing Comments
1.3	Zena Abulhab	12/12/17	Added use case IDs

Table of Contents

List of Figures

Glossary of Terms

Term	Definition
Rpi	Raspberry Pi mini computer
SMS	Smart Mirror system

1. Introduction

1.1. System Purpose:

The Purpose of the Magic Mirror is to add value to a user's daily routine by enabling them to access information quickly and conveniently.

1.2. System Scope:

The Smart Mirror System will address the customer's desire for an easy and interesting way to get basic information simply by standing in front of a mirror/display. This will be accomplished using computer vision techniques and different Application Programming Interfaces from information providers (such as major online weather, news, and mail outlets).

1.3. References:

N/A

1.4. System Overview:

The system will recognize a user when he or she stands in front of the mirror. The user will then be able to see a display on the mirror with useful information such as the current weather conditions, the time and date, and their email notifications among others. The user will be able to customize the display with their own information.

2. General System Description

2.1. System Context:

The system's context includes an gmail interface for viewing emails, a weather and news-fetching interface, a non-wifi reliant built-in time, date, and calendar interface, and a facial-recognition interface

2.2. System Modes and States (Two defined states):

Idle: Means that the Smart Mirror System is detecting image input at constant intervals and looking to detect a user standing in front of the mirror. In the Idle state the Mirror will show basic information. When a registered user is found, then the system leaves the Idle state and enters the Active state.

Active: The system will display the user's information. The system will continue to scan for a user standing in front of the mirror. If the system fails to find a user during a predetermined number of scan cycles (the current optimal number is 35), then it will return to the Idle state and continue to scan for a user.

2.3. Major System Capabilities (Each top level bullet point represents a major group of capabilities):

Vision

REQ01 The user's face shall be able to be visually acknowledged through processing of images taken at regular intervals by the system.
Images could in the future be stored for bug fixing, maintenance, and improvement of the recognition process, but are not currently required to be stored

Display (all of the following will be implemented by searching for and incorporating APIs from major news, weather, and email information providers)

REQ02 The user shall be able to view the displayed weather

REQ03 The user shall be able to view current time and date

REQ04 The user shall be able to view upcoming events

REQ05 The user shall be able to view unread emails

REQ06 The user shall be able to view important news from major news outlet

2.4. Major System Conditions

REQ07 The user shall be able to be identified reliably by the system when standing in front of it

REQ08 The user shall be able to store news/alarm preferences and apply that information

REQ09 The user shall be able to have their select emails and events displayed if they are are recognized user.

2.5. Major System Constraints

The system must provide at least basic functionality by no later than December 2017.

2.6. User Characteristics

The sole user type of this project is a viewer, someone looking at the monitor.

2.7. Assumptions and Dependencies

N/A

2.8. Operational Scenarios

Use Cases

Use Case ID: UC01

Display Weather

Actor: Viewer

Preconditions: Wifi connection

Basic Flow of Events:

1. A viewer looks at the mirror
2. Mirror scans the user's face and sees if it is Jerry's.

Alternative Flows:

2a. No valid face is detected

2b. The facial recognition system fails

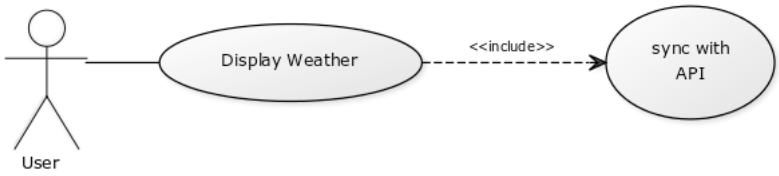

Figure 1. A use case diagram for displaying the weather.

Use Case ID: UC02

Show Date and Current Time

Actor: Viewer

Preconditions: None

Basic Flow of Events:

1. A viewer looks at the mirror
2. The mirror shows the date and time along the top

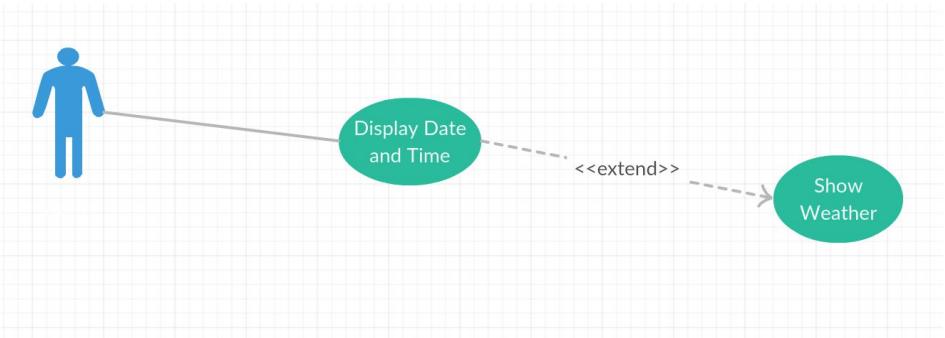

Figure 2. A use case diagram for displaying date and time

Use Case ID: UC03

Viewer Login

Actor: Viewer

Preconditions: None

Basic Flow of Events:

1. A viewer looks at the mirror
2. Mirror detects a face
3. The mirror recognizes whose face it is (known/unknown) through the recognition program, by taking a frame of the video feed of the user's face.

Alternative Flows:

4a. The user is not recognized

4b. The user is not a valid user and cannot log in

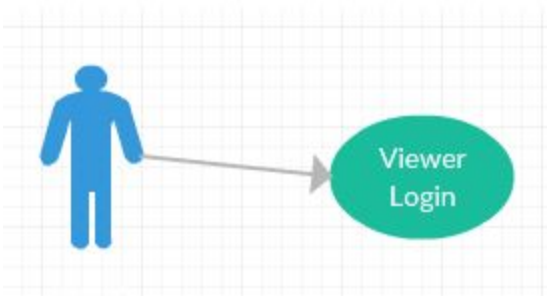

Figure 3. Use case to login to google account (only for Jerry)

Use Case ID: UC04

Show A Viewer's Emails

Actor: Viewer

Preconditions: A network connection to the internet

Basic Flow of Events:

1. A viewer looks at the mirror
2. Mirror detects a face
3. The mirror recognizes whose face it is (known/unknown) through the recognition program, by taking a frame of the video feed of the user's face.
4. The mirror displays the viewer's unread emails from gmail API

Alternative Flows:

4a. The user is not recognized and is prompted to take a better photo

4b. The user is not a valid user and cannot log in

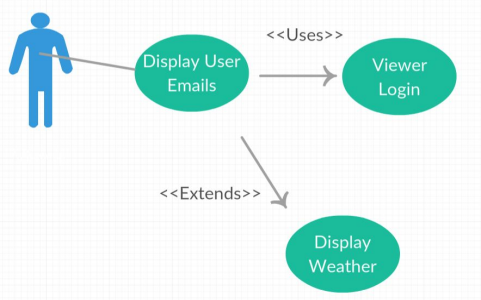

Figure 4. Use case for displaying user emails.

Use Case ID: UC05

Show A Viewer's Events of the Day

Actor: Viewer

Preconditions: A network connection to the internet

Basic Flow of Events:

1. A viewer looks at the mirror
2. Mirror detects a face
3. The mirror recognizes whose face it is (known/unknown) through the recognition program, by taking a frame of the video feed of the user's face.
4. The mirror displays the viewer's events for the day

Alternative Flows:

4a. The user is not recognized and is prompted to take a better photo

4b. The user is not an actual user and cannot log in

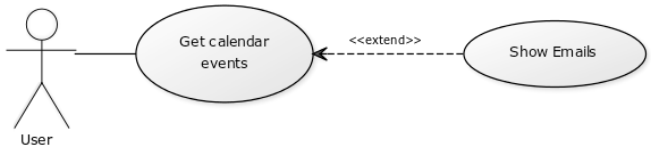

Figure 5. Use case for showing calendar events

Use Case ID: UC06

Use News and Alarm Preferences

Actor: Viewer

Preconditions: A network connection to the internet

Basic Flow of Events:

1. A viewer looks at the mirror
2. Mirror recognizes a face and turns display on
3. The mirror is updated with the latest news preference and alarm settings
4. The mirror shows the first three national news headlines.
5. The system time matches the alarm time alarm and plays the alarm at the set time.

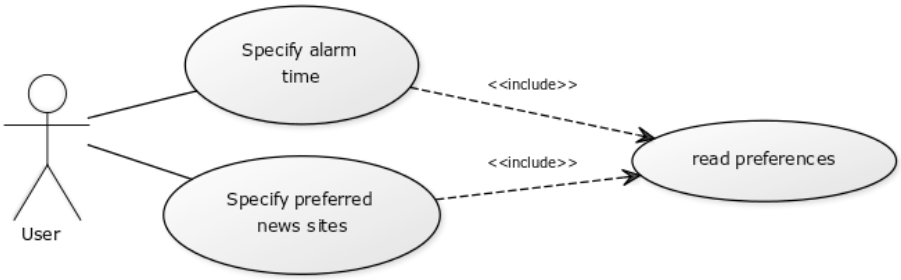

Figure 6. Use case for showing news and playing an alarm based off preferences.

3. System Capabilities, Conditions and Constraints

3.1. Physical

Construction

The user shall be able to install the system anywhere near a power source to power the RPi and computer monitor.

Parts required include a RPi, monitor, camera and speakers.

The RPi can be substituted for any small computer that can attach to the back of the monitor, has 2 USB ports and run Python and Javascript(Node.js)

Durability

The system is composed of a monitor which cannot take much shock damage to the screen. The system should rarely be touched or moved, but the monitor can be moved quite often as long as it's handled with care.

Adaptability

The RPi has enough USB slots for extra user interface devices in the future. The code for the user display can easily be edited through SSH to display new or modified content to the user.

Environmental Conditions

The system needs to be a in a dry, weather-free and room temperature environment.

Motion should be reduced to the system to enable the camera tracking to work properly.

3.2. System Performance Characteristics

The user's position may change, and multiple faces may be present, so the system should consider this when identifying the user's face.

The mirror is intended to be used for any amount of time, but typically, for a few minutes daily, such as after waking up and before going to sleep.

The mirror will take pictures in time intervals and wait to recognize a change in its environment indicating that a face is present

3.3. System Security

Connecting to the RPi from the network requires SSH login access that only the user will have.

Physically connecting to the RPi has very little security, and could be used to do malicious activities. This includes the USB ports and the Network port. The storage device can also be taken out and modified fairly easily, though there will be a cover over the system, it can be removed.

3.4. System Operations

System Human Factors

N/A

System Maintainability

The system will need to have the capability to store image data in the future for maintenance of the Vision component. The image data will also allow for retraining of the internal image classifiers at work in identifying a user, but currently the actual storing mechanism need not be present.

Downtime should ideally not exist, but in the case of downtime, the vision component should continue to operate normally until connection to the server is restored (This maintainability requirement may change in future versions of this document)

System Reliability: (This is most applicable when addressing the user recognition portion of the system, since it will be the most prone portion to present issues with reliability as a result of its use of algorithmic image classification methods.)

The system should correctly identify a user at least 90% of the time.

The system should ideally have a false positive rate of 0, but false positive results (identifying a user when there is no user in front of the mirror) realistically will aim to fall under 5% of all positive identifications.

3.5. Policy and Regulation

N/A

3.6. System Life Cycle Sustainment

N/A

3.7. System Interfaces:

They system will have four basic internal components interfacing with each other and will only interact with one external actor: the user.

Vision: The vision component will interact with the user by detecting them. Then it will interact with the display by determining the state of the system. Idle means no specific information display and Active means that the display is on.

Display: The display will interact with the appropriate APIs to display the information that the user wants to see displayed on the mirror. The display will be the only portion of the system that will provide “interaction” with the user, as it will display the information required by the user.

Settings: The settings will determine which sets of information the user will want to see on the display. They will internally communicate with the API portion which will then be used to display appropriate information on the display.