

**Software Requirements Specification**

**for**

**Live Editing System**

**By**

**Nofar Shaked**

**Table of Contents**

[**1.** **Introduction** **3**](#_Toc352421985)

[**2.** **Overall Description** **3**](#_Toc352421986)

[2.1 Product Perspective 3](#_Toc352421987)

[2.2 User Classes and Characteristics 3](#_Toc352421988)

[2.3 Operating Environment 4](#_Toc352421989)

[2.4 Design and Implementation Constraints 4](#_Toc352421990)

[2.5 User Documentation 4](#_Toc352421991)

[2.6 Assumptions and Dependencies 4](#_Toc352421992)

[**3.** **System Features** **4**](#_Toc352421993)

[3.1 Creating and editing a Project 4](#_Toc352421994)

[3.2 Pre- Record and Recording Session 7](#_Toc352421995)

[**4.** **External Interface Requirements** **9**](#_Toc352421996)

[4.1 User Interfaces 9](#_Toc352421997)

[4.2 Hardware Interfaces 9](#_Toc352421998)

[4.3 Software Interfaces 9](#_Toc352421999)

[4.4 Communications Interfaces 9](#_Toc352422000)

[**5.** **Other Nonfunctional Requirements** **10**](#_Toc352422001)

[5.1 Performance Requirements 10](#_Toc352422002)

[5.2 Safety Requirements 10](#_Toc352422003)

[5.3 Security Requirements 10](#_Toc352422004)

[5.4 Software Quality Attributes 10](#_Toc352422005)

[**Appendix A: Data Dictionary and Data Mode**l **11**](#_Toc352422006)

[**Appendix B: Analysis Models** **13**](#_Toc352422007)

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
|  |  |  |  |
|  |  |  |  |

1. **Introduction**
   1. **Purpose**

This SRS describes the software functional and nonfunctional requirements for the demo of the Live Editing System (LES). This document is intended to be used by Nofar Shaked, who will implement and verify the correct functioning of the system. Unless otherwise noted, all requirements specified here are high priority and committed for the demo.

* 1. **Project Scope and Product Features**

The Live Editing System will allow users to record videos from Intel's depth camera and add effects using hand gestures and voice commands, by so reducing editing time and giving room for creativity.

* 1. **References**

1. **Overall Description**
   1. **Product Perspective**

The Live Editing System is a new system that will save the valuable time needed to edit effects into videos using other programs, like After Effects, by using the recorded person's hand gestures and voice as commands that control predefined effect on screen, all while recording, so that the filmed person can see the effects and himself on screen while recording. The system is expected to allow live broadcasting as well if by connecting to appropriate broadcasting networks, as needed.

* 1. **User Classes and Characteristics**

The Live Editing System is a system designed for personal use in your own computer. Any user can connect his program to their own Gmail/YouTube account and create a personal connection from the system to these outsourced sites, through which the videos created using the system can be posted.

The user can use the program to create projects with unique properties and data source (for effects and inputs for recording). Using these projects the user will record videos with the system's special abilities and the predefined sources and properties. Videos created using the system can be **reviewed** (double checking if everything looks good), **exported** (for advanced editing options, like cropping) and **posted** to Youtube.

* 1. **Operating Environment**

OE-1: The Live Editing System shall operate on Windows using Intel's Perceptual Computing SDK using their depth camera.

OE-2: The system shall connect to Gmail/YouTube accounts and allow posting videos to them.

* 1. **Design and Implementation Constraints**

CO-1: All scripts shall be written in C++ using Visual Studio.

CO-2: The system shall use the latest version of the Intel Perceptual Computing SDK.

* 1. **User Documentation**

UD-1: The system shall include a Help bar, leading to tutorials on how to create projects.

UD-2: On the **Recording Screen** (see FRS for screens descriptions) will appear a help button leading to a graphical tutorials for using the hand gestures and voice commands to start and stop recording, to make effects appear and disappear and how to control them on screen, as well as how to publish the resulting video.

* 1. **Assumptions and Dependencies**

DE-1: The system's special abilities depend on the user's ability to use hand gestures and voice commands. Some disabilities may prevent the user from using them properly (such as inability to use hands or voice or cases of missing limbs).

DE-2: The system cannot run without Intel's CREATIVE Depth Camera and its most updated Perceptual Computing SDK.

DE-3: The system needs an internet connection in order to post videos online.

1. **System Features**
   1. **Creating and editing a Project**
      1. **Description and Priority**

User wants to create a new project, add items and effects to the project, set the menu and commands for the recording, manage other projects and connect to his/her Gmail/Youtube accounts. Priority = High.

* + 1. **Stimulus/Response Sequences**

**3.1.2.1 Project File**

Stimulus: User creates new Project.

Response: As the user names the project, the system checks if that name is already used (at least in the same location), If it does, a message will alert the user and ask him if he wishes to overwrite another project with the same name or not. A new project is created in the Projects folder, without any items, effects or presets in it (if another project is already open and its changes weren't saved, a message will appear, asking if the user wants to save the changes, the options will be Save/Don't save/Cancel)..

Stimulus: User saves project.

Response: Name checking, like in project creation. A project, with all its properties, items and recent changes, is saved to the Projects Folder (in its own folder).

Stimulus: User loads existing project.

Response: An existing project opens up on the screen, with all the properties and items in its folder (if another project is already open and its changes weren't saved, a message will appear, asking if the user wants to save the changes, the options will be Save/Don't save/Cancel).

**3.1.2.2 Items and project Options**

Stimulus: User loads an item or effect to the project.

Response: The item is added to the project's folder and to the Items Window on screen.

Stimulus: User selects an item from Item Window.

Response: Item properties open up in the editing section of the screen (see FRS for screen details). In that section, the item can be previewed, its position on the recording menu can be selected and the commands that operate it on the recording screen can be edited.

Stimulus: User edits an item through the Item editing section (add a voice command or adds it to the recording screen window or edit its position on screen).

Response: Item's new properties are saved and will be implemented in the recording session.

Stimulus: User presses on a trash can symbol on the item that is selected in the Items Window.

Response: Item is removed from Items Window and project.

Stimulus: User presses the Start Recording button.

Response: Directed to Recording Screen (see FRS).

Stimulus: User presses the help button.

Response: A help window will appear.

Stimulus: User presses Connect to Gmail Account.

Response: A pop-up window will appear in which the user needs to fill a form with his Gmail username and password. After user fills all details and presses Connect the system tries to connect using the details. If it succeeds, a message appears saying Connection Successful. If the connection fails, a message appears saying Connection Failed and the user will remain in the form window in order to try again with a different username or password.

* + 1. **Functional Requirements**

|  |
| --- |
| NewProject: The system checks if there isn't a project already open. If there is and there have been changes that were not saved, a pop-up will appear, asking the user if he wants to Save the changes and open the new project, not save the changes and open a new project or cancel and not open a new project. The system will follow the user's choice.  SaveProject: The system updates all the temp changes to the project's folder.  LoadProject: The system checks if there isn't a project already open. If there is and there have been changes that were not saved, a pop-up will appear, asking the user if he wants to Save the changes and load the other project, not save the changes and load the other project or cancel and not load the project. The system will follow the user's choice. |
| AddNewItem: Browsing screen appears from which the user selects an item from his computer. The item is added to the project's folder and is shown in the Items Window.  RemoveItem: A pop-up window will appear, asking the user if he is sure he wants to remove the item from the project. The system will follow the user's choice.  EditItemVoiceCommand: The system lets the user record a voice command that will activate the item in the record session. The voice recording can be reheard and recorded again as many times as the user pleases.  EditItemMenuPosition: the user may choose a position on the recording menu in which a certain item will be placed.  EditItemDefaultScreenPosition: The user may choose a position on screen on which the item will appear once user commands it during recording session. |
| SwitchToRecordScreen: The user will be directed to the Record Screen (see FRS). |
| ProjectHelpMenu: A help window will appear with tutorials on how to manage projects. |
| ConnectToGmail: A connection window will appear with a form. The user needs to fill in his Username and Password. After the user presses Connect, The system will try to connect to Gmail using the user's Username and Password. If it fails, it will alert the user and let him try again. If the connection is successful, a message will alert the user and let him continue. The connection will remain as long as the project is open. |

* 1. **Pre- Record and Recording Session**
     1. **Description and Priority**

The user can open a help menu or record a video. The recording starts a session in which items can appear and be operated using voice commands and hand gestures (captured by Intel's Depth Camera). Recording can start and end using mouse, voice command or hand gesture. After a recording is finished the user can review the video, publish it, delete it or re-record it.

* + 1. **Stimulus/Response Sequences**

**3.2.2.1 Windows redirections**

Stimulus: User Presses the **Help** button.

Response: A tutorial window on how to use the recording and editing system is opened

Stimulus: User Presses the **Return to Project** button.

Response: User is redirected to the **Project Editing** **Screen**.

**3.2.2.2 Recording commands**

Stimulus: User presses **Start Recording** OR uses the **Start Recording Hand Gesture**.

Response: The screen shows a short countdown to the beginning of the recording session, a blinking red dot shows at the corner of the screen and the recording starts. An **End Recording** button now appears. The **Start Recording** and **Help** buttons are removed.

Stimulus: User presses **Stop Recording** button OR uses the voice command OR hand gesture for stopping the recording session.

Response: Recording stops and a new window appears in which user can choose what to do with the recently recorded video

**3.2.2.3 After Recording is done**

Stimulus: user chooses on menu (after recording is over) either to:

* Watch a preview of the recently recorded video OR
* Start another recording (video is saved in the project folder) OR
* Delete the video and start recording a new one OR
* Delete the video and return to Project Screen OR
* Post The video to YouTube and return to Project Screen OR
* Return to Project Screen (video is saved in the project folder)

Response: system follows user's choice.

**3.2.2.4 Items Controls**

Stimulus: User moves hand over an item on the **Items Menu** and closes his hand.

Response: Item appears on screen on its default (or set) position (if is a sound type, it is simply activated).

Stimulus: User closes his hand on an item that appears on the screen, he then drags his closed hand around and finally he opens his hand.

Response: Item is dragged after the hand and stays at the last closed hand position.

Stimulus: User closes both hands close to an item on screen and then move them far from each other or closer.

Response: Item Changes size by distance between hands. It remains at last size when user opens his hands.

Stimulus: User pokes an item that appears on screen with one finger from each hand.

Response: item disappears from screen.

Stimulus: User uses voice command saved for a certain item X.

Response: Item X appears on screen on its default (or set) position (if is a sound type, it is simply activated).

* + 1. **Functional Requirements**

|  |
| --- |
| RecordHelpMenu: A tutorial window appears to help the user learn how to use the different features in the Recording Screen.  SwitchToProjectScreen: System redirects the user back to the Project Screen. |
| StartRecording: System records everything on the screen (camera input and items that appear. Doesn't show the Items Menu. While recording, a red dot will appear on the corner of the screen (to let the user know the system is recording).  StopRecording: System stops recording and saves video to the Project Folder. A menu appears with different options for the user to choose what to do next (see next block of functions). |
| WatchPreviewOfVideo: Recently recorded video is opened from the Project Folder in a video player for the user to preview.  RecordAnotherVideo: Recently recorded video remains saved in the Project Folder and a new recording starts (the new video will be saved to the same Project Folder as well).  DeleteVideoAndRecordAnother: Recently recorded video is deleted from the Project Folder and a new recording starts.  DeleteVideoAndReturnToProjectScreen: Recently recorded video is deleted and the user is redirected back to the Project Screen. |
| QItemAppear: Selected item is generated in its default or set position. Item is now operable for user gestures.  QItemDisappear: Selected item will be removed from view.  QDragItem: Selected item will follow its grabbing hand's path until that hand opens or disappears from sight.  QResizeItem: System captures the user's hands positions (as long as they are both closed) and according to their movement, determines if they get closer or farther from one another and resize the item by that distance. |

1. **External Interface Requirements**
   1. **User Interfaces**

UI-1: The system's **Project Screen** Is operable with the mouse only, except for the voice command recording.

UI-2: The **Project Screen** includes a **Help** button in its menu bar which leads to a full tutorial on how to use the Project Screen.

UI-3: The Record Screen is operable with mouse, hand gestures or voice commands. The parts on this screen that are operable are the **Items Menu**, items on screen, **Start Recording** and **Stop Recording** buttons and the **Help** button.

UI4: The **Recording Screen** includes a **Help** Button which leads to a full tutorial on how to use the **Recording Screen**’s many features and creating a video to post online.

* 1. **Hardware Interfaces**

Intel's CREATIVE Depth Camera has a big part in the recording process and because using it involves viewing the screen, it should be located on top of the screen or close to it, so when a user is recording himself, he can see himself on the screen and operate the features with ease.

* 1. **Software Interfaces**

SI-1: The system shall connect and post videos online using the user's details.

SI-2: Any editing of a project done by the user (adding items and editing them) shall be implemented in the **Recording Screen** using a Project **Data File**, containing all data on the project.

* 1. **Communications Interfaces**

CI-1: The system shall use the user's Username and Password to connect to his Gmail account. If connection is successful, the user's details are saved for future connections.

CI-2: Once a user approves posting a video online, the system uses the user's connection details (from previous connection, if available or asks the user for it and tries to make the connection, as in CI-1) to connect to the user's account and upload the video there.

1. **Other Nonfunctional Requirements**
   1. **Performance Requirements**

PE-1: The Recording Session should run smoothly and all input from Intel's Depth Camera should be clear (especially visual image using the RGB channel) since the finished video is designed for broadcasting and should be of high quality.

PE-2: During Recording Session, the system runs a lot of test, checking for any user commands. These tests should not take too much memory and processing so that the video output can run smoothly, maintain a high frame rate and be recorded at the same quality.

PE-3: Uploading a video to YouTube takes a lot of time and should be done directly from the Youtube Web site. The system should send the file location on the computer to the YouTube upload page (and open the page) for the user to complete the upload form.

PE-4: Any item added to a project is added to the Project Folder for easy access in the Recording Session.

* 1. **Safety Requirements**

No safety requirements have been identified.

* 1. **Security Requirements**

SE-1: User cannot upload a video to YouTube without filling his Username and Password.

SE-2: No video created using the system will be uploaded without the user's permission.

* 1. **Software Quality Attributes**

Availability-1: The user will download the installation file from the product's website. The user will install the program on his machine. The installation will include all tools and SDKs needed for the system to run (Including the Perceptual Computing SDK).

* 1. **Software Installation Requirements**

SI-1:

# Appendix A: Data Dictionary and Data Model

User details = username

+ password

Item = path (location on computer or project)

+ type [image, video, sound, visual effect]

+ is active (activated by user)

+ scale (default = 1.0)

+ position on screen (if type of item is image, video or effect)

+ position on Item Menu (-1 if none)

+ voice command for activation (-1 if none)

+ Transparency Alpha Level (0-255)

Item menu = number of items

+ items list

Voice command= Assigned to an item. Once recognized, activates item

Hand gesture = For recording control and item control; includes finger data

Hand position = For each hand

+ Right/Left X position

+ Right/Left y position

+ Right/Left z position

Creating

m

1

containing

items

User

1

1

m

1

starting

Exports

Recording

Project

1

1

Using and controlling

Video

m

m

Figure 1

Partial data model for the demo of the Live Editing System

# Appendix B: Analysis Models