**Android Synchronized Multimedia Integration Language**

**Software Requirements Specification (SRS)**

**Version 0.2**

**Produced for:**

**Dr. Maureen Doyle**

**CSC – 440**

**Software Engineering**

**Produced by:**

**Not User Friendly**

**Brad Barker**

**Alex Gilbert**

**Jacob Ensor**

Executive Overview

The Synchronized Multimedia Integration Language (SMIL) implementation, for the Android platform, seeks to elegantly and seamlessly allow for the creation, distribution, and viewing of SMIL formatted messages. Android SMIL will give the customer an unparalleled ability to create dynamic presentations on a mobile platform and allow them access to their media virtually anywhere via cloud storage.

The **user** objectives for the SMIL are to:

Provide them with powerful mobile tool for creating and viewing presentations.

Enable them to easily add, size, and place media into a presentation.

Allow them to quickly view their presentations.

Allow them to store and access their presentations on the cloud.

Enable them to share their work with other users.

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 01/28/2012 | 0.1 | Initial Draft. Identified use cases.  Defined Functional Requirements.  Defined Non-Functional Requirements. | Brad Barker |
| 03/21/2012 | 0.2 | Revised use case format.  Clarification on various topics. | Brad Barker |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

1 Introduction 7

1.1 Specification Definition 7

1.2 Specification Objectives 7

1.3 References 7

1.4 Specification Overview 7

2 SMIL Overview 8

2.1 Definition 8

2.2 User Business Benefits 8

2.2 External Hardware 8

2.3 SMIL Capabilities 8

2.3 Summary of System Capabilities 9

2.3.1 User Capabilities 9

2.3.1 Other Capabilities 9

3 Functional Requirements 10

3.1 User 10

3.1.1 Use Case: User Login / Home 11

3.1.1.1 Path: Create Account 11

3.1.1.2 Path: Account Already Exists 12

3.1.1.3 Path: User Information Invalid 12

3.2.2 Use Case: User Sends a Message 13

3.2.2.1 Path: User selects recipient and SMIL message 13

3.2.3 Use Case: User Views a Message 14

3.2.3.1 Path: Play the message 14

3.2.3.2 Path: Pause the message 15

3.2.3.3 Path: User FFW / Rewinds the message 15

3.2.3.4 Path: User closes the message 16

3.2.4 Use Case: User Creates a Message 16

3.2.4.1 Path: User creates blank message 16

3.2.4.2 Path: User creates message from a template 17

3.2.5 Use Case: User Edits a Message 17

3.2.5.1 Path: User edits the layout 18

3.2.5.2 Path: User adds/edits a text field 18

3.2.5.3 Path: User adds image 19

3.2.5.4 Path: User adds video 20

3.2.5.5 Path: User adds audio 21

3.2.5.6 Path: User performs player actions 22

4 System Requirements 23

4.1 Timeliness 23

4.2 Installation 23

4.3 Interoperability 23

4.4 Integrity 23

4.5 Security Requirements 25

4.6 Quality 25

Table of Figures

* Figure 1: User Diagram 9

# Introduction

The section introduces the system requirements specification (SRS) for the Synchronized Multimedia Integration Language (SMIL) system to its readers.

## Specification Definition

This specification documents the system-level requirements for the SMIL system.

## Specification Objectives

The objectives of this specification of the SMIL are to:

To formally specify:

System overview

Functional requirements.

Non-functional requirements.

## References

This specification references or complies with the following documents:

**SMIL Specification 3.0 W3C:**

W3C’s specifications for the Synchronized Multimedia Integration Language.

Implements, and adheres to core markup elements

**Agile Scrum Conventions:**

Use Case Modeling Guidelines and methodologies.

## Specification Overview

This specification is organized into the following sections:

*Introduction*, describing SMIL and it’s basic functionality to the reader.

*System Overview*, which provides a brief, high level description of the Android SMIL application and it’s interaction between modules.

*Functional Requirements*, which specifies the functional system requirements in terms of a use case model consisting of each external’s use cases and use case paths.

*Non-Functional Requirements*, which specifies the required system factors, accessibility, quality, and security.

# SMIL Overview

This section outlines the high level overview of the Android SMIL application. Defines the interactions between modules and primary objectives.

## Definition

The SMIL application will be a mobile composer and player of SMIL based presentations that can be seamlessly stored on the cloud. The application will give the user the capability to easily create and share media from virtually anywhere.

## User Business Benefits

SMIL will:

Provide its user with a globally accessible cloud storage for their presentations

Give its users the ability to quickly download and play presentations

Enable the user to quickly create a presentation quickly adding text and various media

Enable the user to create a presentation from a template

Enable the user to save their work to the cloud

Provide the user with the ability to share their work with others.

### External Hardware

SMIL interacts, either directly or indirectly, with the following significant external hardware:

**Mobile Device**:

**Android Phone**, for running the application.

**SD Storage,** for locally saving/caching presentations.

**Networking**:

**Internet**, which is the global network used for communication between the cloud and SMIL.

**Google Cloud,** providing WAN storage to facilitate global access and enable the users to easily share information with each other.

## SMIL Capabilities

This subsection provides a high-level overview of major capabilities of the GPM. Note that this subsection provides useful information for understanding the following requirements, but does not contain specific testable requirements.

### Summary of System Capabilities

#### User Capabilities

SMIL will provide the following capabilities to our users:

**Editors**. The SMIL will enable users to:

Create a blank presentation

Create a presentation from a template

Edit layout settings

Add, move, and resize text

Add and modify images, audio, and video

Adjust element timing

Save and share presentations

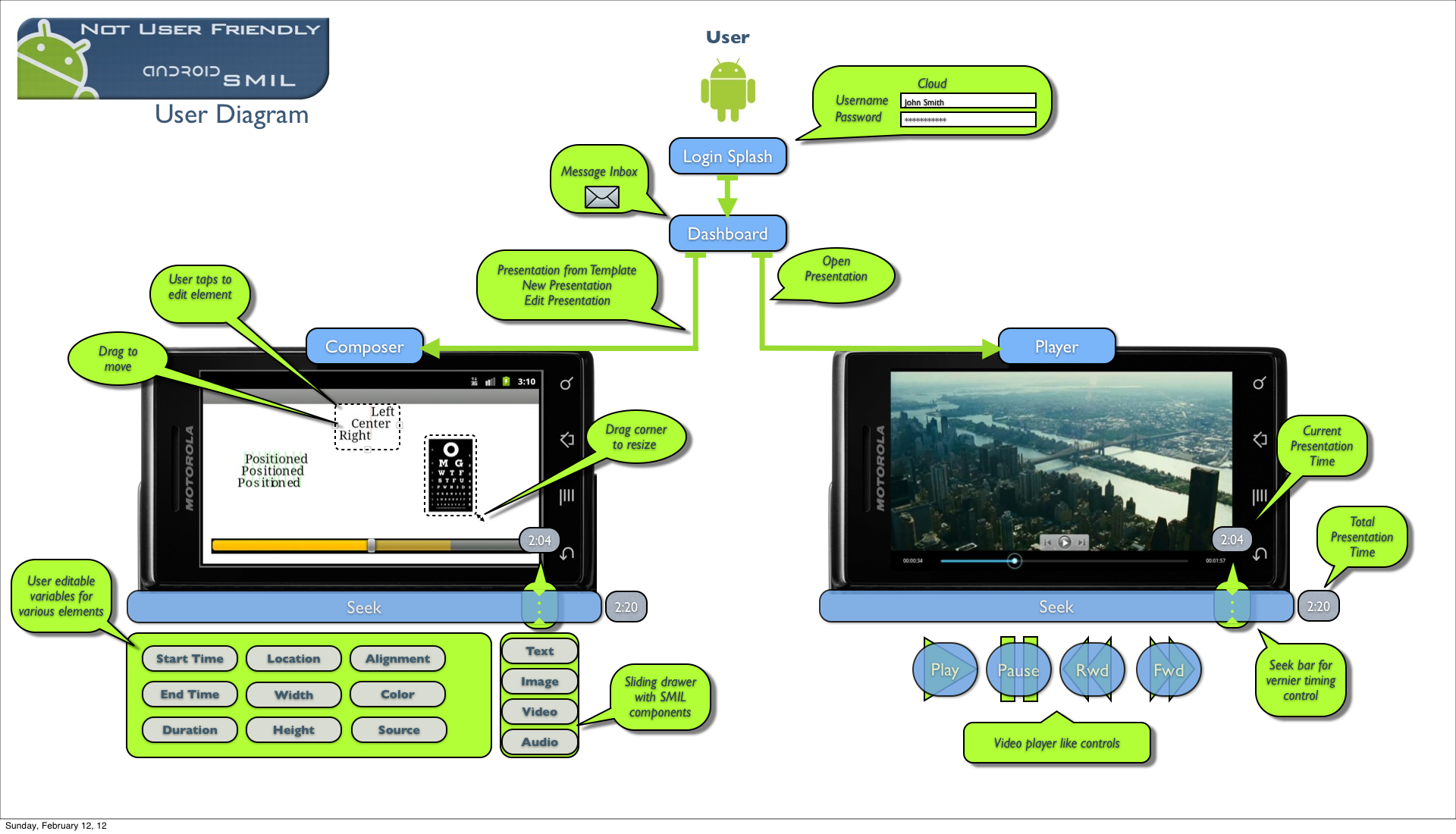
**Viewers**. SMIL presentations will be able to:

Open presentations

Play, pause, and rewind presentations

#### Other Capabilities

**Google Cloud**. SMIL will make use of Google’s Cloud technology to:

Store and retrieve presentations

* Figure : User Diagram

# Functional Requirements

The section of the SRS specifies the functional requirements of SMIL in terms of use cases and their associated use case paths.

### User

The subsection specifies the functional requirements primarily associated with users.

Definition

A *user* is the role played by a person who uses SMIL.

Required Capabilities

A user needs the following required technical expertise, experience, and training to effectively interact with SMIL:

Use of an Android phone or table device

Experience sending and receiving email to use send messages

Familiarity with a presentation creator program such as PowerPoint

States

A user can be in the following states:

Login / Home:

Viewing Message

Sending Message

Composing:

Adding Components

Editing Components

Playing:

Playing

Paused

Rewinding / Fast Forwarding

Use Cases

User Login

User sends SMIL message

User views SMIL message

User creates SMIL message

User edits SMIL message

User saves SMIL message

#### Use Case: User Login / Home

Use Case Requirement

The user will login to their account before being taken to the home screen. This will allow the user to enter their Google account credentials that will give them access to their information on the cloud.

Business Justification

Separate user accounts provide accountability and privacy.

Use Case Paths

Create Account

User Login

User information invalid

##### Path: Create Account

Path Requirement

The user will be give the option to create a new user account

Actors

Google Cloud

User

Preconditions

The initial activity will provide the user with a prompt to add an account as well as a link for new user registration

User has a Google Gmail account

Username or email is not already registered

Interactions

* 1. The user selects their Gmail account from the list provided
* 2. Alternatively the user clicks a link for “new user registration”
  + a. The link will open a form for the user to register their Gmail account.
  + b. Once authenticated the user will be returned to step 1.

Post conditions

The users now has access to the Google cloud interface

The user will be forwarded to the SMIL home screen

Exceptions

The username and/or password is incorrect, the user will need to resolve this issue before continuing. (This should be a non-issue as any android user will have a Gmail account registered on their device)

##### 3.1.1.2 Path: Account Already Exists

Path Requirement

The user information is valid and has permissions for the cloud.

Actors

Google Cloud

System

Preconditions

User has a Google Gmail account

Username or email is already registered

Interactions

1. The system will automatically send saved credentials to the server in an attempt to authenticate.

Post conditions

The user will then be taken to the SMIL home screen

Exceptions

The user has changed their account information such as password, the user will then be taken to use case 3.1.1.1

##### 3.1.1.3 Path: User Information Invalid

Path Requirement

The users login email address and/or password are incorrect.

Actors

Google Cloud

User

Preconditions

The user has supplied invalid credentials

Interactions

1. Google will fail to authenticate.
   1. This can be repeated indefinitely, although the Gmail account will be locked after a number of attempts.

Post conditions

The user will remain at the login screen

A message will be displayed to notify the user of the error

#### Use Case: User Sends a Message

Use Case Requirement

At the home screen the user will be given their inbox with the option to edit/view/send an existing message or to create a new one. The user will be able to share their message with other SMIL users.

Business Justification

Allows sharing of information while maintaining privacy through user account.

Use Case Paths

User selects recipient and SMIL message

Recipient does not exist

##### 3.2.2.1 Path: User selects recipient and SMIL message

Path Requirement

The intended recipient has a registered account

Actors

Google Cloud

User

Preconditions

The user in fact has a SMIL message in their inbox.

Interactions

1. The user will select a message from their inbox and click on a mail button
2. The user will enter the intended recipients email address
3. The user will then click send
   1. The cloud will verify the recipients account and will give their account access to the message
4. Dialog message will be displayed to verify receipt

Post conditions

The user will be taken back to the inbox

Exceptions

The intended recipient does not exist, the user will be given an error and returned to step 2.

#### Use Case: User Views a Message

Use Case Requirement

At the home screen the user will be given their inbox with the option to edit/view/send an existing message or to create a new one. The user will be able to open a SMIL message, which will then cause the player activity to open and render their presentation.

Business Justification

Provides the user with the ability to play their presentations.

Use Case Paths

Play the message

Pause the message

User fast forward or rewinds the message

Closes the message

##### 3.2.3.1 Path: Play the message

Path Requirement

The player has successfully opened and loaded the presentation.

Actors

User

Preconditions

The player has loaded

The message is paused

Interactions

1. The user pressed the play button
   1. The play button icon will change to a pause icon
   2. The timer will begin and the presentation will play

Post conditions

The message is playing and the user can perform paths 3.2.3.2 and 3.2.3.4

##### 3.2.3.2 Path: Pause the message

Path Requirement

The player has successfully opened and loaded the presentation.

Actors

User

Preconditions

The player has loaded

The message is playing

Interactions

1. The user presses the pause button
   1. The pause button icon will change to a play icon
   2. The timer will pause and the presentation will stop

Post conditions

The message is paused and the user can perform paths 3.2.3.1, 3.2.3.3, and 3.2.3.4

##### 3.2.3.3 Path: User FFW / Rewinds the message

Path Requirement

The player has successfully opened and loaded the presentation.

Actors

User

Preconditions

The player has loaded

Interactions

1. The user holds the rewind button
   1. The timer will decrement until the button is released or the timer reaches zero
2. The user holds the fast forward button
   1. The timer will increment until the button is released or the end is reached
3. The user moves the slider/seek bar
   1. The timer will be adjust to the point in which the slider is release

Post conditions

The message will display the information at the given time.

##### 3.2.3.4 Path: User closes the message

Path Requirement

The player has successfully opened and loaded the presentation.

Actors

User

Preconditions

The player has loaded

Interactions

1. The user pressed the close button
   1. The presentation state will be saved in the cache
   2. The player will be destroyed

Post conditions

The user will be taken back to the home screen

#### Use Case: User Creates a Message

Use Case Requirement

At the home screen the user will be given their inbox with the option to edit/view/send an existing message or to create a new one. The user will be able to create a new SMIL message from either a template or from scratch. These actions will invoke either an empty composer activity or one containing default elements in a template.

Business Justification

Provides the user with the ability to create presentations.

Use Case Paths

User creates blank message

User creates message from template

##### 3.2.4.1 Path: User creates blank message

Path Requirement

The user has successfully logged into the application.

Preconditions

The home screen is loaded

Interactions

1. The user selects the “New” button
   1. The user selects “new blank presentation” from a list
2. A prompt will allow the user to name the presentation

Post conditions

The user can edit the new presentation

Exception

The user doesn’t enter a valid name.

An error message will be displayed and the user will remain on step 2.

##### 3.2.4.2 Path: User creates message from a template

Path Requirement

The user has successfully logged into the application.

Preconditions

The home screen is loaded

Interactions

1. The user selects the “New” button
   1. The user selects a template from a list
2. A prompt will allow the user to name the presentation

Post conditions

The user will be able to edit the pre-populated elements.

Exception

The user doesn’t enter a valid name.

An error message will be displayed and the user will remain on step 2.

#### Use Case: User Edits a Message

Use Case Requirement

At the home screen the user will be given their inbox with the option to edit/view/send an existing message or to create a new one. Once a message has been selected the user will be taken to the composer where the presentation will be loaded for editing

Business Justification

Provides the user with the ability to modify a presentation.

Use Case Paths

User edits the layout

User adds/edits a text field

User adds image

User adds video

User adds audio

User performs player actions

##### 3.2.5.1 Path: User edits the layout

Path Requirement

The user has opened a presentation for editing.

Actors

User

Preconditions

The composer has loaded the presentation

Interactions

1. The user will open the elements dropdown list.
   1. Select the layout item
2. The layout form will open with fields to edit
   1. Background color
   2. Width
   3. Height
3. The user closes the layout form
   1. The composer will re-render the presentation based on the new layout

Post conditions

The canvas will reflect the new layout options

##### 3.2.5.2 Path: User adds/edits a text field

Path Requirement

The user has opened a presentation for editing.

Actors

User

Preconditions

The composer has loaded the presentation

Interactions

1. The user will advance the presentation to the time to add the text
2. The user will open the elements dropdown list.
   1. Select the text item
3. The text form will open with fields to edit
   1. Value
   2. Width
   3. Height
   4. Color
   5. Z-index (This is the visual depth of the element)
   6. Size
   7. Font
   8. Start time
   9. End time
4. The user closes the text form
   1. The composer will add the text field
5. The user can then tap and drag the text field to the desired location
6. Tapping will allow editing of the text field
7. Double tapping will reopen the text form for manual editing

Post conditions

The user can now see the new text field on the canvas

Exception

The user fails to populate all of the data correctly.

##### 3.2.5.3 Path: User adds image

Path Requirement

The user has opened a presentation for editing.

Actors

User

Preconditions

The composer has loaded the presentation

Interactions

1. The user will advance the presentation to the time to add the image
2. The user will open the elements dropdown list.
   1. Select the image item
3. The image form will open with fields to edit
   1. Source
      1. User will be given a list of available images
   2. Width
   3. Height
   4. Z-index
   5. Start time
   6. End time
4. The user closes the image form
   1. The composer will add the image to the current frame
5. The user can then tap and drag the image to the desired location
6. Touching and dragging in the lower right will allow resizing of the image
7. Double tapping will reopen the image form for manual editing

Post conditions

The user will now see the image displayed on the canvas.

Exception

The user fails to populate all of the data correctly.

The image file doesn’t exist or cannot be found.

##### 3.2.5.4 Path: User adds video

Path Requirement

The user has opened a presentation for editing.

Actors

User

Preconditions

The composer has loaded the presentation

Interactions

1. The user will advance (seek) the presentation to the time to add the video
2. The user will open the elements dropdown list.
   1. Select the video item
3. The video form will open with fields to edit
   1. Source
      1. User will be given a list of available videos
   2. Width
   3. Height
   4. Z-index
   5. Start time
4. The user closes the video form
   1. The composer will add the video field
5. The user can then tap a drag the video to the desired location
6. Touching and dragging in the lower right will allow resizing of the video
7. Double tapping will reopen the video form for manual editing

Post conditions

The video is displayed on the canvas

Exception

The user fails to populate all of the data correctly.

The video file doesn’t exist or cannot be found.

##### 3.2.5.5 Path: User adds audio

Path Requirement

The user has opened a presentation for editing.

Actors

User

Preconditions

The composer has loaded the presentation

Interactions

1. The user will advance (seek) the presentation to the time to add the audio
2. The user will open the elements dropdown list.
   1. Select the audio item
3. The audio form will open with fields to edit
   1. Source
      1. User will be given a list of available sounds
   2. Start time
   3. End time
4. The user closes the audio form
   1. The composer will add the audio field
5. Opening the elements menu will show audio components

Post conditions

The audio file will then be added to the presentation.

Exception

The user fails to populate all of the data correctly.

The audio file doesn’t exist or cannot be found.

##### 3.2.5.6 Path: User performs player actions

Path Requirement

The user has opened a presentation for editing. Redundant from use cases in 3.2.3, intended to show the composer is similar in functionality as the player.

Preconditions

The composer has loaded the presentation

Interactions

1. The user will have the same user paths as in case 3.2.3.
   1. The composer is an extension of the player

Post conditions

The composer will render the frame based on the timer allowing editing of the displayed elements

# System Requirements

This section specifies the required system quality factors that are not related to the specific functional requirements documented in the use case model.

### Timeliness

This subsection specifies the following requirements concerning the degree to which the system must ensure that its persistent information is current (i.e., up-to-date):

SMIL presentations and their associated media files will be stored in Google’s cloud service but upon initial access they will be copied locally to speed up repeat access. *See Integrity for information on persistence.*

SMIL presentations will also be stored in serialized form to further reduce access time.

## Installation

This subsection specifies the following usability requirements associated with the ease with which the system can be installed.

The SMIL application can easily be installed by adding the APK package to an android device.

The application will additionally require a Google account with privileges to the app engine. (This is only performed once)

## Interoperability

This subsection specifies the following requirements associated with the ease with which the system can be integrated with other system (e.g., browsers, legacy applications, and required databases).

SMIL will be designed to work with Android versions 2.3.3 and later.

## Integrity

Data integrity between the cloud and the cache must be maintained to minimize loss of information. Cached information will be stored for a limited period of time and the cache will represent a combination of the most recent and most frequently accessed presentations.

* Cache storage
  + Created upon opening a file (if none existent)
  + Contains last 3 presentations to be opened
  + Contains 3 most viewed presentations
  + Contains object serial files for quick restart
  + Synced with cloud when application is closed
* Cloud storage
  + Contains all user files
  + Contains no serialized files
  + Supersedes local files in case of data mismatch
  + Updated on application closed

## Security Requirements

The SMIL application connects to Google’s cloud and authenticates using the uses Google account information. The application will not store or transmit any credentials unencrypted. The application will keep each user’s files in separate buckets and ensure their information remains private from other users.

## Quality

The SMIL implementation will be tested using a combination of automated unit testing and usability testing. The non-user interface components will be tested using JUnit tests. The user interface will be tested through usability testing and evaluation. For thorough testing information refer to the SMIL test plan.