Governors State University

THREEMIUM

Combined Requirements and Design

Version 3

**THREEMIUM**

**Illinois**

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Rev | Date | Purpose | Originator |
| 0 | 09-22-2019 | Original | Brittany, Dayne, Eric, Jim, Michael |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

[Revision History 1](#_Toc22297358)

[1 Feature Description (DWL) 3](#_Toc22297359)

[1.1 Competitive Information <Jimmy Ellis> 3](#_Toc22297360)

[1.2 Relationship to Other Features 4](#_Toc22297361)

[1.3 Assumptions and Dependencies 4](#_Toc22297362)

[1.4 Future Enhancements (DWL) 4](#_Toc22297363)

[1.5 Definitions and Acronyms 4](#_Toc22297364)

[2 Technical Description 4](#_Toc22297365)

[2.1 Software Architecture (DWL) 4](#_Toc22297366)

[2.2 Software User Information flows 5](#_Toc22297367)

[2.3 Database Architecture <Eric Gomez> 5](#_Toc22297368)

[2.4 Database Information flows 5](#_Toc22297369)

[2.5 Interactions with other Features (if Any) 5](#_Toc22297370)

[2.6 Interactions with other application Elements 5](#_Toc22297371)

[2.7 Capabilities <Michael McRae> 5](#_Toc22297372)

[2.8 Risk Assessment and Management (DWL) 6](#_Toc22297373)

[3 Feature Requirements 6](#_Toc22297374)

[3.1 Identification of Software Application Requirements 6](#_Toc22297375)

[3.2 Identify of Database Requirements (DWL, EG) 18](#_Toc22297376)

[3.3 Operations, Administration, Maintenance and Provisioning (OAM&P) (DWL) 19](#_Toc22297377)

[3.4 Security and Fraud Prevention (DWL) 19](#_Toc22297378)

[3.5 Release and Transition Plan (DWL) 19](#_Toc22297379)

[4 Design Description (DWL) 19](#_Toc22297380)

[4.1 Internal/external Interface Impacts and Specification (DWL) 19](#_Toc22297381)

[4.2 Design Units Impacts (DWL) 19](#_Toc22297382)

[4.3 Design Unit A (e.g. Main Control Functions) (DWL) 21](#_Toc22297383)

[***4.3.1*** ***Functional Overview (DWL)*** 21](#_Toc22297384)

[***4.3.2*** ***Impacts (DWL)*** 21](#_Toc22297385)

[***4.3.3*** ***Requirements*** 21](#_Toc22297386)

[4.4 Design Unit B (e.g. Photo and Quote Update Process) (DWL) 22](#_Toc22297387)

[***4.4.1*** ***Functional Overview*** 22](#_Toc22297388)

[***4.4.1*** ***Impacts*** 22](#_Toc22297389)

[***4.4.1*** ***Requirements*** 22](#_Toc22297390)

[4.5 Design Unit C (e.g. Automated Time Setting Process) (DWL) 22](#_Toc22297391)

[***1.1.2 Functional Overview*** 22](#_Toc22297392)

[***4.5.1*** ***Impacts*** 22](#_Toc22297393)

[***4.5.2*** ***Requirements*** 22](#_Toc22297394)

[5 Open Issues 22](#_Toc22297395)

[6 Acknowledgements 22](#_Toc22297396)

[7 References 22](#_Toc22297397)

[8 Appendices 23](#_Toc22297398)

# Feature Description (Dayne Longtin)

This mobile application will enable users to access a multitude of backgrounds and quotes and create beautiful compositions of images and quotes. These quotes and backgrounds can range from: inspirational quotes, humor quotes, personal quotes, romantic quotes, nature backgrounds, national park backgrounds and personal photos. The application will enable the user to configure said backgrounds and quotes differently. If the user would like the background and quote to display with different settings in a sequential, random or fixed rate, they can do so. With this in mind; the user will be able to create, edit and delete said compositions, and save or publish the compositions at any time.

## Competitive Information (Jimmy Ellis, Brittany Lynch, Dayne Longtin)

*Identify competitor products or services that this feature will directly compete against, and whether your company has the potential to be the first to market this new feature or capability.*

There is a large market for camera applications that allow you to edit photos you took on your smart phone. This market exists because of the large amount of people who have cell phones in the world. Many people in America have cell phones. Due to the size of the market, there are many successful applications that have come into existence before this application that we are working on. Due to how large the mobile market is, it creates a better environment for our application to succeed. In stating this, we believe there is room for new applications in the market to grow and succeed.

Canva Application: Versatile graphic design platform. Choose from various templates and stock photos to create graphics. Edit images and add elements like personal trademarks or text.

Adobe Smart Phone Apps: Adobe has 30+ apps on the market geared towards smartphone users creating content with pictures they took on their cellphones. The apps allow you to take that content and share it on social media or distribute it across the internet. Apps in the adobe suite for smartphones that directly have the features proposed in this app include Adobe photoshop Express, Adobe Spark Video, Adobe Fresco, Adobe Lightroom – Photo Editor, Adobe Illustrator Draw, Adobe Spark Post

## Relationship to Other Features (Dayne Longtin)

In terms of relationships to other programs. The comparison could be made that this application is relative to a social media application. Albeit, it is not. The assumption could be made since the application revolved around different background compositions with quotes. Being that applications like Facebook, Twitter and Instagram all deal with photos, it’s easy to see how someone would relate this application to other apps. This application is not a social media application as it is intended to create different photo and quote compositions.

## Assumptions and Dependencies (Jimmy Ellis, Dayne Longtin)

* Describe any assumptions made (e.g., new functionality or capability that will provided by other in the same time frame as your feature).

1. We assume that once the app is complete that both the android and apple app stores will agree that our creation is up to standard and allow the app on their respective markets.
2. It is assumed that the app will be downloaded by at least one person on its release.
3. It is assumed that users will follow instructions and use the app to put quotes on pictures.

* Identify other features, services, capabilities, and network elements upon which this feature, service, or capability is depend on.

1. This app is dependent on the social circumstance that everyone has a cellphone.
2. This new cellphone eco system where every cellphone is a smart phone with a camera is environment this app was created in.
3. This app is dependent on wifi and 4G networks existing so that the online features exist.
4. This app is dependent on the android and iphone markets to release the app to the public too.

* Also identify the required development and/or changes in customer operational procedures needed to support this feature

**Customer Operational Procedures Needed to Support Features**

1. The first line of support for customers would be the help documentation that would teach the user how to use the features

* A FAQ document of frequently asked questions to help users with common problems.
* A forum on the applications website filled with people asking questions about the app so that common questions, issues, and concerns, can be publicly viewed and used to help users' sort through their issues.

## Future Enhancements (DWL)

*Discuss any planned or possible evolution of the feature. Some project will need to be delivered in phases; this will be the place to document your plan.*

In terms of this project, we are currently determining a multitude of options with the DB, user specifications, etc. With that in mind, in future renditions of the project we would like to clarify everything that will go into DB and how it will function as a whole. We are in determination of user options in terms of how it will work with the database. Anything database related; we would like to clarify in future renditions. Since this is a competition for a contract, we will also be taking future enhancements into consideration before making the changes needed.

## Definitions and Acronyms <Brittany Lynch>

**GSU** – Governors State University

**API** – Application Programming Interface a software interface that allows two applications to talk to each other

**App** – Software Application

**DB** - Database

**FAQ –** Frequently asked questions

**IDE –** Integrated Development Environment is a software application that provides comprehensive tools for programming and software development.

**OOP-** Object Oriented Programming

**OS -** Operating System

**UI-**  User Interface

# Technical Description <Jimmy Ellis>

*Give a brief overview of feature design and describe the technical make-up of the feature. Remember, the main reason for writing this document is to provide sufficient details to the developers who will implement this feature. In this case, it will be your own team.*

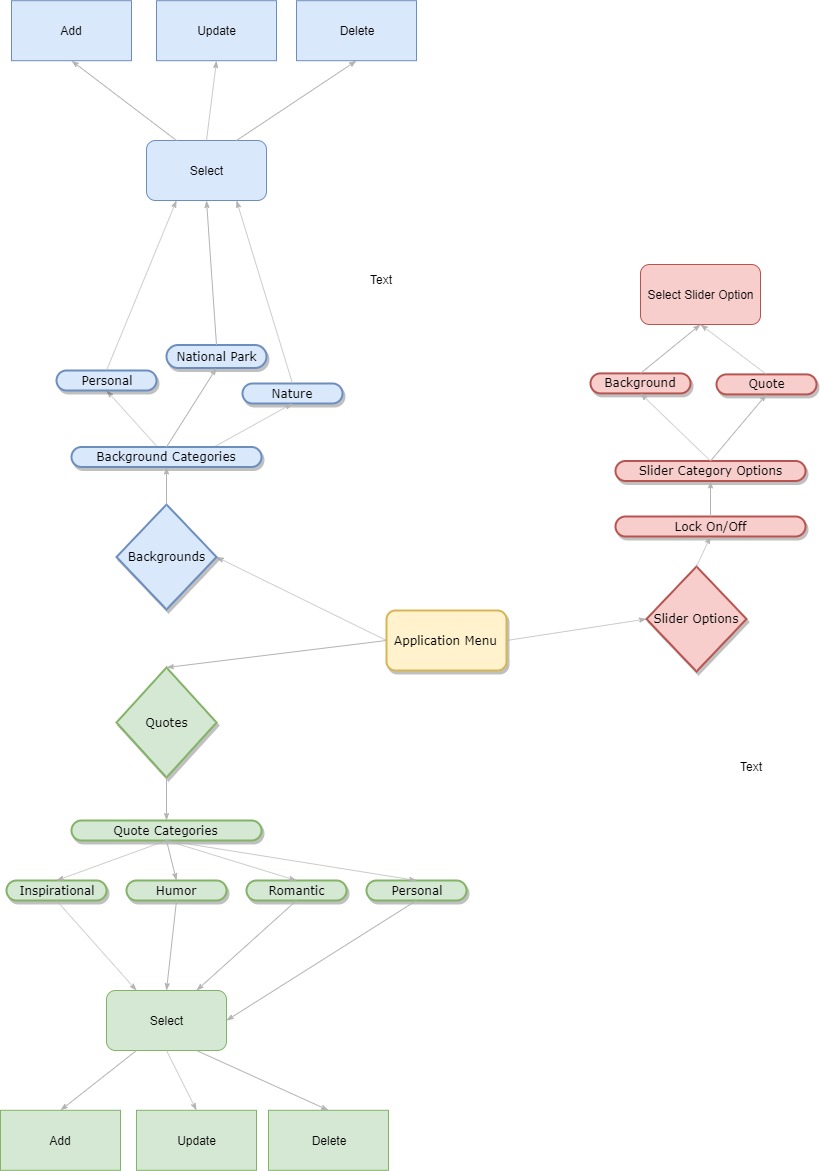
*You should focus on user interface, data model, data information flow, these interfaces can be used as starting point, but what are those data and what happens to those data as your users move from one state to another state and where are the data stored. It will be easier to fill in other subsection once this is done. Example: Need to check and select different “photo data”, "People Quotes data" ......  in the databases that will be used by the application.   e.g., when a customer visits the App and clicks the different service icon: a query is sent and retrieve associated part of the database and pass the info to the application for display ... etc. Need to consider all different kind of usage, thus a different combination of customer data query, security check/blocking, data update, data delete, insert new data, remove old data, .............etc.*

Java Programming Language is being used to create an APK for the computer program to be uploaded to the app store. The photos are stored in a database. The information that is held in the application from the user consists of pictures and text quotes. This information is all stored in a local database. The text information are all “string” data in code and the information stored. The pictures are all in jpeg format stored locally on the application.

## Software Architecture (Dayne Longtin)

This application will be developed using both Flutter and Realm as IDEs. The interface will have a menu with multiple attributes, being options for accessing the slider for background and quote rotations, along with creating new compositions. The database will be tied to the Android Studio program in order to enable the user to access backgrounds and quotes.

## Software User Information flows (Dayne Longtin)



## Database Architecture <Eric Gomez>

1-Tier Architecture:

This architecture is also referred to as a local data base system. A single tier architecture is centralized, meaning that all the data is stored in one location. This data is directly available to the user to store and read data. The reason for using this architecture is that we don’t need a network connection to perform the actions on the database. This will also rid free of needing to setup any servers which will relief cost of maintenance and implement network security since we are not sending data in and out of application and having to sustain client data.

## Database Information flows

*Provide the steps of the usage for this feature. (See above comments)*

## Interactions with other Features (if Any)

## Interactions with other application Elements

*List all possible interaction with outside vendor’s database, and what are needed to address these interactions.*

## Capabilities <Michael McRae>

System security: Regarding users, Active Directory will verify credentials upon logging into the application. Also, when not within the confines of the LAN a VPN client will be required to access the company network. Regarding internal network security, an end point security system (Vipre) will be in place as well as an email monitoring service (Proofpoint Essentials), A firewall will also be utilized, and all users will be assigned credentials based on their level of access required.

Data integrity audit: Data integrity will be ensured by the use of RAID 5 configuration with a hot spare on the file server. This will ensure that if a drive fails it will be seamlessly replaced with the hot spare. Also, upon drive failure or data corruption only a none of the data will be lost/corrupted due to parity across drives. This will be cheaper and more efficient than RAID 0 (mirroring) which would require twice the amount of drives needed.

Routine Backup/Fault Recovery: Synchronized incremental backups backups will be performed two times per week to a dedicated backup server. This will only backup files that have been changed since the previous backup which will increase the speed of the backup process. Fault recovery will be from this backup server. Regarding e-mail, Office 365 will remove the ability for maintaining an exchange server on-premise. Proofpoint Essentials, which we will be utilizing for email security, includes an emergency inbox/outbox that is to be used in the event that Office 365 is temporarily unavailable.

System Updates: Application updates will be performed per the frequency established within the proposal and/or customer contract. System updates regarding network software will be determined by the software developers that we are utilizing (I.e. Microsoft).

## Risk Assessment and Management (DWL)

*This section is used to identify as early as possible some of the risks that are associated with the introduction of this feature. It should also contain recommendations to eliminate or minimize these risks.*

The mobile application could have a multitude of risks associated with it. If we’re not specific, there’s a chance that the slider being used to change the settings of all the rotations the backgrounds and quotes could have bugs and cause slider settings to go in places the user hasn’t intended. Another issue could be in terms of not being specific enough about personal photos. Is there only going to be a select few images from the DB that the user can choose from that they uploaded from their gallery. Will the user be able to just pull from their gallery? Or will the DB just contain their whole gallery? If not more specific, this could have major implications and problems in the future.

# Feature Requirements

## Identification of Software Application Requirements

This section provides a brief explanation of the use of named and enumerated requirements to identify and number requirements. Requirements need to be enumerated, but different tools can be used to accomplish this. The following format is an example:

**<THREEMIUM – Enter Custom Quote - 1 v1> (Michael McRae)**

Users shall have the ability to generate a custom quote that is not featured within the main database.   A method will be created in Flutter to allow the generation of a button that will enable this function.

Implementation: Mandatory.

**<THREEMIUM – Select Stock Image - 2 v1> (Michael McRae)**

Users shall have the ability to browse a database of stock images and chose which ones fit their quote best.     A method will be created in Flutter to allow the generation of a button that will enable this function.

Implementation: Mandatory.

**<THREEMIUM – Font Modification Tool - 3 v1> (Michael McRae)**

Regarding quotes, the following characteristics shall be customizable at a minimum: style, size, color, bold, and italic.   A method will be created in Flutter to allow the generation of a button that will enable this function.

Implantation: Mandatory.

**<THREEMIUM – Help Button - 4 v1> (Michael McRae)**

A help menu and User Manual link (PDF) will be available within the settings menu.  This will help users troubleshoot common issues that may be experienced.  A method will be created in Flutter to allow the generation of a button that will enable this function.

Implementation: Mandatory.

**<THREEMIUM – Save to Camera Roll - 5 v1> (Michael McRae)**

Users shall have the ability to select one or more photos and save/export them to a personal device of their choice. A method will be created in Flutter to allow the generation of a button that will enable this function.

Implementation:  Optional.

**<THREEMIUM – Languages - 6 v1> (Michael McRae)**

Applications will be written in high level languages such as SQLite, Java, etc. in the IDE Android Studio.

Implementation: Mandatory.

**<THREEMIUM – User Manual - 7 v1> (Michael McRae)**

Description: Every application should have a published User Manual that will guide a user through the functionality of the application and serve as a resource that will provide guidance where applicable.  The user manual shall be updated to reflect changes made during revisions to the application itself.

Implementation: Mandatory.

**<THREEMIUM – Minimum Password Req’s - 8 v1> (Michael McRae)**

REGEX, Passwords shall meet the following criteria:

* Be a minimum of eight (8) characters long
* Minimum (1) one uppercase letter (A-Z)
* Minimum (1) one lowercase letter (a-z)
* Minimum (1) one digit (0-9)
* Minimum (1) one special character (~`!@#$%^&\*())

Implementation: Mandatory.

**<THREEMIUM – User Folder Size Restriction - 9 v1> (Michael McRae)**

Description: Each user will be given 2Gb of storage capacity at any given time.  Users will have the ability to / responsibility of maintaining a “clean” folder.

Implementation: Mandatory.

**<THREEMIUM –Device Hardware Permissions- 10 v3> (Brittany Lynch)**

Description:  Device hardware and software capabilities that support app functionality.

Alarm Manager API on Android to determine when scheduled notifications should be deployed

Camera roll and Photo Gallery: uploading and downloading photos and images

Operating system bootloader: For the application to be notified when the phone has been rebooted

Implementation: mandatory

**<THREEMIUM –Push Notifications- 11 v3> (Brittany Lynch)**

Description: This is required to support the motivation message feature as scheduled notifications. It will be implemented with Flutter local notifications plugin. The plugin will populate the scheduled messages within the Android Alarm API. The application will request notification every time the device is rebooted in order to repopulate the scheduled notifications in the Alarm Manager API on the device.

The App Store will manage and deploy application update notifications.

Implementation: optional

**<THREEMIUM –Motivational Message- 12 v3> (Brittany Lynch)**

Description: This feature gives the user the ability to schedule a motivational message. The user can enter choose a quote from the database categories, enter their own text or select random. The user will select a day of the week Monday-Friday, and time of day to be notified that a motivational message is available to view as text. User can toggle the scheduled notification on and off, as well as daily or weekly.

The user will be allowed up to 12 motivational messages to be scheduled and saved in a bank at any one time.

Implementation: optional

**<THREEMIUM –Terms of Conditions- 13 v3> (Brittany Lynch)**

Description: Terms and conditions agreement for users.

Implementation: mandatory.

**<THREEMIUM – Main Controller- 14 v3> (Eric)**

Description: This is what manages the entire application. From accessing screens, to waiting for events, and running functions.

Implementation: mandatory.

**<THREEMIUM – Settings Page / Window- 15 v3> (Eric)**

Description: This page holds settings that can affect the way the slides are played during Play Mode. Settings can be accessed in two ways, through the menu icon in Play Mode or using the TOOL icon on the bottom navigation bar outside of Play Mode.

The settings menu helps update values in the user's unique settings values from Settings Table. The options available are:

Quote Speed: Can be adjusted with slider.

See Quote Speed - **01100**

Picture Speed: Can be adjusted with slider.

See Picture Speed - **01200**

Synchronized Speed: Can be toggled on or off by tapping the lock icon.

See Synchronized Speeds - **01300**

Shuffle: Can be toggled on or off by tapping the shuffle icon.

See Shuffle State- **01800**

All updated values will be queried back to Settings table in Database after the window or page has been exited. And being used by Play Mode should be stored in an Object using OOP. The class should contain the variables listed and functions that set / get / and query back DB.

Implementation: Mandatory

**<THREEMIUM – Quote Speed - 16 v3> (Eric)**

Description:  The speed that the app utilizes play the quote. It can be adjusted during Play Mode by bringing up the Settings Menu or by hitting the TOOL icon on the bottom navigation bar outside of Play Mode. The speed then can be adjusted by sliding the quote speed slider.

The quote speed is an integer value that is under the *Settings* table and can be accessed and updated with a user's unique *settingsID* foreign key from *Users* table. The initial quote speed value is given a default value of 3 when the settings values for user are first created.

In the Settings window, the slider will limit the user of speeds in range of 3s to 15s, for best experience. The value can only be updated to database after closing window. The quote speed value is utilized in Play Mode in a timer/interval function that sets itself off every second passed through it while it's in Play Mode.

Implementation: Mandatory

**<THREEMIUM – Picture Speed - 17 v3> (Eric)**

Description:  The speed that the app utilizes play the picture. It can be adjusted during Play Mode by bringing up the Settings Menu or by hitting the TOOL icon on the bottom navigation bar outside of Play Mode. The speed then can be adjusted by sliding the picture speed slider.

The picture speed is an integer value that is under the *Settings* table and can be accessed and updated with a user's unique *settingsID* foreign key from *Users* table. The initial quote speed value is given a default value of 3 when the settings for user is first created.

In the Settings window, the slider will limit the user of speeds in range of 3s to 15s, for best experience. The value can only be updated to database after closing window. The picture speed value is utilized in Play Mode in a timer/interval function that sets itself off every second passed through it while it's in Play Mode.

Implementation: Mandatory

**<THREEMIUM – Synchronized Speeds- 18 v3> (Eric)**

Description:  This helps keep both picture and quote speed sync up to same value. By default, when activated, the quote speed will sync up to equal the same speed as the picture speed. If any slider is moved, the value to the other slider will be updated to equal the same. The value will be updated with a user's unique *settingsID* foreign key from *Users* table to database after closing window so that when that when the Settings Window is open again, it will be reminded of its current value to visually represent that to user.

Implementation: Mandatory

**<THREEMIUM – Play Mode View - 19 v3> (Eric)**

Description:  This screen is where slides are played. Screen orientation can be adjusted to vertical or horizontal, all depending on the how the device is rotated. The interface is using trusted interface principles that make it an easy experience for the user and techniques used commonly on current major phones, as of 2019.

When entering Play Mode, the slides will play according to the adjusted settings saved under the user's unique settings values stored from the Settings table.

Play mode can be exited my hitting back on your device’s OS assigned back button or by tapping the settings menu icon from the Play Mode Navigation Window and hitting the Home button.

To bring up the Play Mode Navigation Window, you do a single tap on the current slide. The window will appear and is set to go away right after an approximate 3s if Play Mode state is not set to pause, an ideal time frame normally used for best experience.

See: Play Mode Navigation Window, Settings Menu

Implementation: Mandatory

**<THREEMIUM – Play Mode Navigation Window - 20 v3> (Eric)**

Description: To bring up Play Mode Navigation Window you do a single tap on the playing slide. The pane will go away right after an approximate 3s if Play Mode state is not set to pause.

In Navigation Window:

Tapping Home icon takes you back to home page.

Tapping Square Screen icon fixes display horizontally.

Tapping Menu icon opens menu screen.

Tapping Play icon plays the slides.

Tapping Pause icon pauses the slides.

Tapping Forward or Reverse icon reverses to previous slide or moves on to the next slide in slide show.

See: Play States **01600**

This window element is layered above the slide show, but underneath the Play Mode Settings Window. It stylized to stay hidden until the slide is tapped. A Timer Event will be activated which that will make element visibility back to hidden after the seconds (3s) have passed.

All icons will have Trigger Event Listener looking for “click” event, which will run their own individual functions. [all functions still pending]

See: Play States **01600**

Implementation: Mandatory

**<THREEMIUM – Play Mode Controller – 21 v3> (Eric)**

Description: Play Mode is the controller that keeps track of how the list plays.

It has a list of all the indexes for each playlist in either a shuffled or fixed sort. When running the Play Mode Controller, a list is created according to the settings set from the *Settings* table. For example: If shuffled is toggled to true (on), then the list will be created and have its indexes shuffled.

It will also keep track of the index of both the quote and the image to know the position they are on the list and takes notice of event clicks on screen to know when user is clicking on specific icon on the screen and detects phone orientation.

This comes useful when knowing when to clear any timed interval functions or when to start them. A good example for clearing a time interval would be when entering setting or hitting pause and a good way to start a time interval function would be when clicking play or when exiting a settings menu.

The time interval function would oversee displaying the next quote/picture and incrementing the index of its position. Each quote and picture would have their individual timer interval function running so that they can play at their own rate, unless they are set to be played in a synced speed.

**See:** Shuffle State **003700**

Implementation: Mandatory

**<THREEMIUM – Play States - 22 v3> (Eric)**

Description: These are the States that Play mode can be in.

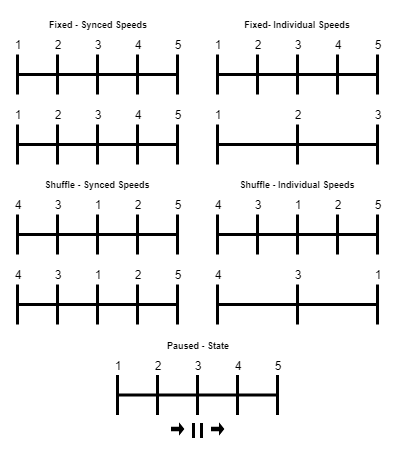
*Play list shuffled – with – synced speeds*

*Play list shuffled – with – individual speeds*

*Play list fixed – with – synced speeds*

*Play list fixed – with – individual speeds*

*Paused list fixed/shuffled – with – speeds not required*



Implementation: Mandatory

**<THREEMIUM – Fixed State - 23 v3> (Eric)**

Description: When playing a slideshow, fixed state plays quotes and images in the order given by user. In a fixed state the speed that the image and quotes play can either play in sync or the can have their own different individual speed.

Implementation:

**<THREEMIUM – Shuffle State - 24 v3> (Eric)**

Description: When playing a slideshow, shuffle state plays quotes and images in a random order. In a shuffle state, the speed that the image and quotes play can either play in sync or the can have their own different individual speed.

The algorithm will be shuffled using the Fisher–Yates Shuffle (Durst enfeld's Version).  This algorithm helps because it already takes a created list and shuffles it in place. This is a huge benefit for the memory and performance when it’s a large list. Its time complexity is O(n), meaning that it will take on the order of *n* operations to perform a shuffle of an item. This will be considered a true shuffle.

To shuffle a list of ***n*** elements:

for **index** from 0 to ***n***−1 do

**rand** ← random integer that is from 0 to (list size - 1) - **index**

exchange list[**index**] and list[**rand**]

Implementation: Mandatory

**<THREEMIUM – Paused State - 25 v3> (Eric)**

Description: In this state both the quote and the image are both prevented from going display the next set of content. It would work by clearing the timer interval function and re-running it when set back to play. When going from a paused to play state, the timer interval functions are reset, and the image/quote will change after the set time has passed.

Implementation: Mandatory

**<THREEMIUM – Playlist View – 26 v3> (Eric)**

Description: This view displays all playlist from Playlist table. Associated with Playlist Controller. Contains a playlist container where all queried playlist will be added. Each container holds the playlist title and a button container with a 'Play', ‘Edit’, and 'Add Slide' button. At the very the top of the view there is a container that holds an ‘Add Playlist’ button.

Implementation: Mandatory

**<THREEMIUM – Playlist Add Playlist View– 27 v3> (Eric)**

Description: This view contains a form to create a new Playlist. Contains a title input. Contains a button container with 'Submit’ button.

Implementation: Mandatory

**<THREEMIUM – Playlist Add Slide View– 28 v3> (Eric)**

Description: This view contains a Multiview layout to select quote and image slide to slide table.

User will be first prompted to select image category; it will then display images associated with category. Once image is selected will prompt for quote category, it will then display quotes associated with category.

Implementation: Mandatory

**<THREEMIUM – Playlist Edit View– 29 v3> (Eric)**

Description: This view contains a Multiview layout to edit playlist name, sorting slides, editing slides, and deleting playlist options.

Multiview :

* + Name edit view contains form with input filled with current name. Contains submit button.
  + Sorting slides view contains slides. Displayed according to slide order. With Cancel and Submit button. In this view user will tap on slides and in order that they would like playlist to play in.
  + Editing slides view contains all slides associated with playlist id. They can edit slide selected and quote. They can also choose to delete from playlist.
  + Deleting playlist will contain a disclaimer before user deletes asking if they are sure to delete with a Cancel and Delete button.

Implementation: Mandatory

**<THREEMIUM – Playlist Controller– 30 v3> (Eric)**

Description: Handles Playlist view, Playlist New view, Playlist Edit, and Playlist Add.

**When Handling:**

**Playlist Add Playlist View**

When submit button is pressed, checks that form has been filled, if not filled will return to this view, if input fields are all filled will insert values title, and auto increment id value to playlist table.

If ‘Play’ button is pressed, it will take playlist id value and go to Play Mode View.

**Playlist Add Slide View**

For image category prompt, it will query image category table for titles and display titles into individual categories containers into categories container.

For image prompt, it will query images table for image blobs, that are also associated with previously selected image category id; decode and display images into individual image containers into images container.

For quote category prompt, it will query quote category table for titles and display titles into individual categories containers into categories container.

For quote prompt, it will query quotes table for titles that are also associated with previously selected quote category id and display quotes into individual quote containers into quotes container.

**Playlist Add Edit View**

When editing name, the controller will query for playlist name and fill input with title as place holder. If user cancels will go to beginning of Multiview, if submits, it will update table with new title using playlist id.

For sorting slides, controller with query for all slides associated with playlist id.

Will display them into their containers. Will wait for user input on individual slide and will start keeping track of each slide that is pressed and label it with associated current index. If submit is pressed will check to see if all slides are indexed and update individual slide in slide table with new order number, the reset will be updated in their current order starting from last index. If cancel is pressed, it will simply go back to beginning of Multiview.

For editing slides, the controller will query for all slides associated with playlist id and display them in container. From there it will allow user to select slide and go to slide edit view, with associated slide id. When in edit slide view user will be allowed to change slide and quote associated with that slide id. And update new values into table. If they select to delete table, they will be prompted to delete. If user selects delete, controller will delete all slides associated with that table and then delete the playlist value from playlist table.

Implementation: Mandatory

**<THREEMIUM – Quotes View – 31 v3> (Eric)**

Description: This view contains a Multiview layout to add, edit, delete quotes.

First user will be prompted to select quote category, then will be presented with a gallery of quotes from selected category. There the user can select a quote to edit or press add quote button.

If user selects a quote, they will be taken to an edit quote view will the form will be auto filled with values from selected quote.

If user selects add quote, they will be taken to an add quote view there will be a form to fill title and quote.

There is also be a hidden log for error inputs in both views to notify user if form has not been completed.

If user selects to delete quote, user will be prompted to cancel or delete before sending back to quote gallery.

Implementation: Mandatory

**<THREEMIUM – Quotes Controller– 32 v3> (Eric)**

Description: Handles Quote view. Before user is prompted to select a quote category, controller will query for all quote category titles and place them to select from. When a category is selected, it will take associated id to a quote gallery where all quotes associated with quote category id are placed in individual containers.

If user selects a quote, the controller will take to Multiview to edit quote with associated quote id. Table will be updated according to filled form. If form not filled, view will stay same view until complete or selecting cancel. Else if filled will update table with new values and go back to quote gallery.

If user selects add quote, will take to Multiview to add quote. Table will be insert values according to filled form. If form not filled, view will stay same view until complete or selecting cancel. Else if filled will insert into table with new values and go back to quote gallery.

Inputs: title, quote

Auto Inputs: quoteid(auto), categoryid

Implementation: Mandatory

**<THREEMIUM – Image View – 33 v3> (Eric)**

Description: This view contains a Multiview layout to add, edit, delete images.

First user will be prompted to select image category, then will be presented with a gallery of images from selected category. There the user can select an image to edit or press add image button.

If user selects an image, they will be taken to an edit image view form where it will be auto filled with values from selected image.

If user selects add image, they will be taken to an add quote view there will be a form to fill title and select image.

There is also be a hidden log for error inputs in both views to notify user if form has not been completed.

If user selects to delete image, user will be prompted to cancel or delete before sending back to quote gallery.

Implementation: Mandatory

**<THREEMIUM – Image Controller– 34 v3> (Eric)**

Description: Handles Image view. Before user is prompted to select an image category, controller will query for all image category titles and place them to select from. When a category is selected, it will take associated id to an image gallery where all images associated with image category id are placed in individual containers.

If user selects an image, the controller will take to Multiview to edit image with associated image id. Table will be updated according to filled form. If form not filled, view will stay same view until complete or selecting cancel. Else if filled will update table with new values and go back to image gallery.

If user selects add image, will take to Multiview to add image. Table will be insert values according to filled form. If form not filled, view will stay same view until complete or selecting cancel. Else if filled will insert into table with new values and go back to image gallery.

**Inputs:** title, imageblob

Imageblob will be encoded into table according to selected image from user phone image gallery using imagepicker widget.

**Auto Inputs:** imageid(auto), categoryid

Implementation: Mandatory

**<THREEMIUM – Database Tables - 35 v3> (Eric)**

Description: The tables available for the app. These tables keep records of users, categories, settings, images, and quotes.

**Users** (userid)

userid and *settingsid* have unique auto increment values

**Image Categories** (categoryid, title, *userid*)

categoryid has a unique auto increment value and userid is related to Users table

title – string with image category title.

**Quote Categories** (categoryid, title, *userid*)

categoryid has a unique auto increment value and userid is related to Users table

title – string with quote category title.

**Settings** (settingsid, shuffle, quotespeed, picturespeed, *userid*)

settingsid has unique auto increment values

**Images** (imageid, title, imageblob, *categoryid*)

imageid has unique auto increment values

title, string value for title

imageblob, encoded image

*catagoryid foreign key for Image categories*

**Quotes** (quoteid, title, quote, *categoryid*)

quoteid has unique auto increment values

title, string value for title

quote, string value for quote

*catagoryid foreign key for Quote categories*

**Playlist (**playlistid, title, date)

quoteid has unique auto increment values

title, string value for title

date stores date value from day that is last played

**Slide** (slideid, *playlistid quoteid*, *imageid*)

quoteid has unique auto increment values

*playlistid foreign key for Playlist*

*quoteid foreign key for Quote*

*imageid foreign key for Image*

**Motivational Message** (messageid, message)

messageid has unique auto increment values

Implementation: Mandatory

**<THREEMIUM – Path Mapping User Experience - 36 v3> (Eric)**

Description: This is the list of screens available and paths to get to each screen.

Implementation:  Mandatory

**<THREEMIUM-Application Menu- 37 v3> (Dayne Longtin)**

This part of the application is where the user will have basic access to different elements of the application. Different functions being the tutorial the first time they use the application, creating a new project, accessing saved projects, along with different elements of the application. This user interface will be crucial in allowing users to have a fundamental understanding of the application, along with knowing the basic navigation of the app.

Implementation: Mandatory

**<THREEMIUM-Quote Bank- 38 v3> (Dayne Longtin)**

Section of application where user can look for inspirational quotes from a set bank of them.

Implementation: Mandatory

**<THREEMIUM-Daily Notifications- 39 v3>(Dayne Longtin)**

This portion of the application will send notifications to the user’s phone on a daily basis in order to remind them that they have a project that they have not yet completed, but also there to notify them when they aren’t using the application to pique the user’s interest. Turning off notifications can be accessed in settings and turning off notifications will also turn off daily/weekly encouragement messages.

Implementation: Mandatory

**<THREEMIUM-Complete Function- 40 v3>(Dayne Longtin)**

This function will enable the user to finish the background & quote composition they have been working on. This function will automatically save the project to the user’s phone gallery so they may keep the image. An icon will appear showing that the image has been saved to their phone as soon as they click complete.

Implementation: Mandatory

**<THREEMIUM-Share Function- 41 v3>(Dayne Longtin)**

After the user has completed the background and quote composition, an option will appear below the complete bar and opt for the user to share it on social media. When the user clicks on it, it will open a list of options from which the user will choose the social media platform they would like to share it on. This requirement will serve as a segue for users that want the social experience, but don’t want to save it to their phone.

Implementation: Optional

**<THREEMIUM-Edit Function- 42 v3>(Dayne Longtin)**

This function will enable the user to access any element of the project they are working on being the background or the inspirational quote at any time and edit it. This function will be accessed one of two ways, the editor, which will highlight all elements of the project and the element touched will be the element to be edited. The other way being through hovering and holding the element the user wishes to edit.

Implementation: Mandatory

**<THREEMIUM-Delete Function- 43 v3>(Dayne Longtin)**

Using this function will allow the user to access any element of the project they are working on being the background or the inspiration quote at any time and delete it. This function will be accessed one of two ways, through the editor, which will highlight all elements of the project and the element touched will be the element to be deleted. The other way being through hovering and holding the element the user wishes to delete. The delete function can also be used to delete entire projects by holding and pressing the delete function in the editor and selecting delete all.

Implementation: Mandatory

**<THREEMIUM-Add Function- 44 v3>(Dayne Longtin)**

This part of the application will be accessed through the editor where the user will be able to access different backgrounds and inspirational quotes. When the user clicks on the add function it the application will ask quote or background, the user will select. After the user selects the option the application will then ask if they want to choose from presets from them, upload a background, or use an image/quote from the internet. If the user chooses internet, a google search will appear where they can search and implement the backgrounds/quotes of their interest.

Implementation: Mandatory

**<THREEMIUM-Editor- 45 v3>(Dayne Longtin)**

This element of the application is where the user is immediately sent to after selecting create new project. This editor is directly next to where box where the background and inspirational quote will be displayed. This editor is where the user will, add elements, edit elements, and delete elements/the entire project. The editor will be in landscape mode by default.

Implementation: Mandatory

**<THREEMIUM-Settings- 46 v3>(Dayne Longtin)**

Feature of application that will be directly in the top, right hand corner in the menu of the application. The setting will contain options such as whether or not the user wants notifications, login settings, personal information changes, the tutorial, along with terms and conditions of the application.

Implementation: Mandatory

**<THREEMIUM-Motivational Message- 47 v3>(Dayne Longtin, Michael McRae)**

This function will be a notification that will be sent to user’s phone. Upon opening, a quote will appear for the user to read. This function can be changed in settings where a slide bar will appear that the user can change. Available options will be the daily, weekly, and off. Selecting off is separate from turning all notifications off and will only turn the encouragement messages off.

In order to allow permissions to implement notifications within flutter we must add the following to the source code of our flutter project which will run upon booting the host:

*<uses-permission android: name=android.permission.RECEIVE\_BOOT-COMPLETED”/>*

The following must also be added to the source code within Flutter to enable push notifications:

*<receiver android:name=”com.dexterous.flutterlocal notification.ScheduledNotificationB*

*<intent-filters>*

*<action android:name=”android.intent.action.BOOT\_COMPLETED”></actions>*

*</intent-filters>*

*</receiver>*

We must then add the following dependency to the dependency’s file:

*flutter\_local\_notifications: ^0.6.0*

*path\_provided: ^0.5.0+1*

Within the library folder we will be required to create a local notification stateful widget. Next, a plugin is to be initialized that is to be used to create the notifications by adding the following to the top of the source code:

*Import’ package:flutter\_local\_notifications/flutter\_local\_notifications/.dart’;*

At this time a method must be created for the notifications. This method will initialize the settings from the host device. A button will also be created that will begin running this method. Within this method will be a “Title” and “Body” field that will show up in the notification banner on the host.

In order to get the notifications to display outside of the background only, we will need to set the Boolean ongoing field to true and the Boolean autoCancel field to false within our method. This will allow a ribbon to be displayed on the top of the screen which will display the notification.

Two additional methods are to be created to enhance the user’s experience. These will be a method which allows the toggling of silent mode for the notifications as well as a method for turning off all notifications.

Implementation: Mandotory

**<THREEMIUM-Offline Mode- 48 v3>(Dayne Longtin)**

This part of the application will allow the user to turn the application to offline mode so the user doesn’t have to consume data while accessing the application. Using this functionality will relieve functionality of the search functions in the editor along with share function.

Implementation: Optional

**<THREEMIUM-Upload Background- 49 v3>(Dayne Longtin)**

This will be an option when the user selects between wanting to use a preset background, a search on the internet or uploading background. This option will enable consumers to pick any given image off of their phone and be able to implement it to their project.

Implementation: Mandatory

**<THREEMIUM-Editor Layout- 50 v3>(Dayne Longtin)**

This option in the application will enable the user to change the layout of the editor. This will automatically be an option in the editor itself, simply by selecting layout. The user will be able to change the layout to be over the top, on the bottom, or as a drop-down menu when the user swipes the top of their phone. This option can also be accessed in settings, if the user decides to change it before they access the editor.

Implementation: Optional

**<THREEMIUM –Invite- 51 v3> (Brittany Lynch)**

Description: The user can send an invitation to join the app entering an e-mail address.

Implementation:  Optional

## Identify of Database Requirements (DWL, EG)

Depend on your design, the feature can be supported by one or multiple database. The following is just a suggestion, it is up to you how you would like to organize your database.

**Users** (userid)

userid and *settingsid* have unique auto increment values

**Image Categories** (categoryid, title, *userid*)

categoryid has a unique auto increment value and userid is related to Users table

title – string with image category title.

**Quote Categories** (categoryid, title, *userid*)

categoryid has a unique auto increment value and userid is related to Users table

title – string with quote category title.

**Settings** (settingsid, shuffle, quotespeed, picturespeed, *userid*)

settingsid has unique auto increment values

[pending - storing indexes for image and quote]

**Images** (imageid, title, imageblob, *categoryid*)

imageid has unique auto increment values

title, string value for title

imageblob, encoded image

*catagoryid foreign key for Image categories*

**Quotes** (quoteid, title, quote, *categoryid*)

quoteid has unique auto increment values

title, string value for title

quote, string value for quote

*catagoryid foreign key for Quote categories*

**Playlist (**playlistid, title)

quoteid has unique auto increment values

title, string value for title

**Slide** (slideid, order, *playlistid quoteid*, *imageid*)

quoteid has unique auto increment values

order int value

*playlistid foreign key for Playlist*

*quoteid foreign key for Quote*

*imageid foreign key for Image*

## Operations, Administration, Maintenance and Provisioning (OAM&P) (DWL)

Due to the constraints that the team will be on in terms of the competition, maintenance will be contingent upon the contract. If no contract is obtained by winning this competition, there is absolutely no point in keeping routine maintenance of the application. Once the contract is obtained, we as the team can negotiate how routine maintenance and updates will be determined.

## Security and Fraud Prevention (DWL)

This application will be secured by having a login page for the user to input information to prevent fraudulent behavior. Actual security will be implemented contingent upon a contract obtained at the end of this competition.

## Release and Transition Plan (DWL)

If team 3 wins the competition, we will immediately release the application on a much wider scale not too far after the competition has been completed. Following that, we as a group will continue to update and further improve the application as we see necessary.

# Design Description (DWL)

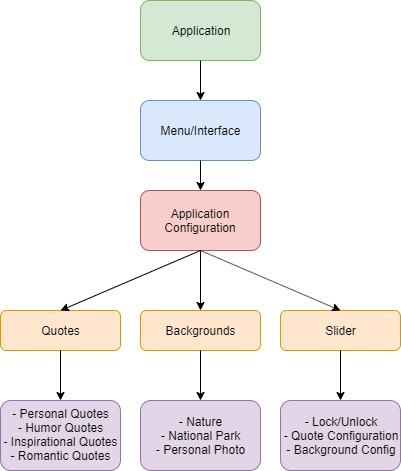
This mobile application will have an intuitive UI to allow users to be able to access a plethora of quotes and backgrounds. When the user selects the options from the UI, a database containing all the quotes and backgrounds will allow the user to implement said backgrounds and quotes to their desire. The UI will also have a slider that the user can lock or unlock to simultaneously or individually change the quote and background rates. Diagrams included further down in section 4.

## Internal/external Interface Impacts and Specification (DWL)

The user interface will contain multiple bars of which they can choose from, being that of quotes, backgrounds and the slider option, after that the user will be able to select options based on that criteria. The feature in of itself has set data, being that of 5 backgrounds and quotes each. From the categories, romantic quotes, humor quotes, inspirational quotes, personal quotes, personal images, national park images and nature images. This will be set by what is already in the application and the personal photos/quotes the user has in the system.

## Design Units Impacts (DWL)

This is a general design diagram and flowchart representing what the application will do in order to function properly.



## Design Unit A (e.g. Main Control Functions) (DWL)

The application will open with the user having access to a multitude of options. The user will be brought the application menu, where the user will have the option to access the slider to change the rotations of backgrounds and quotes. The menu will also give the user access to quotes and backgrounds. From there the user can choose between the types of quotes and backgrounds they would like. After they select, they have the option to add, update, or delete it based on their preference.

### ***Functional Overview (DWL)***

In regards to functionality, when the application is in use when the user is in the editor the user will have the option between all variants of backgrounds being national park, nature, wildlife, landscape and personal photos. The user will also have the option between all variants of quotes being, inspirational, humor, romance and personal quotes. With this in mind, the user will opt to select which of each categories of both the quotes and backgrounds to choose from. Once the user chooses their options, the user will pull both the background and quotes from an offline database. Once the options are pulled from the database, there will be a slider with the option to have the composition to be used at a random, sequential, or fixed rate.

### ***Impacts (DWL)***

There are no known impacts on this subsystem at the moment, as the program is in its infancy and will need further corrections and changes in the near future.

### ***Requirements***

Replace this section with a list of feature requirements covered by this subsystem (from the set of total feature requirements). Requirements are allowed to be implemented by more than one design unit, in which case the requirement is noted here as being partially implemented by this design unit. However in these cases, one design unit must be assigned responsibility for delivery of the **complete** requirement, i.e. ensuring that the "pieces" of the requirement implementation all come together and the entire requirement is met. If there are no impacts to this subsystem, this section can be omitted.

All requirements covered by the design must be listed with the requirements tag number. It is not necessary to copy the text associated with a requirement number as that will require you to update the design document anytime there is a change in the text.

## Design Unit B (e.g. Photo and Quote Update Process) (DWL)

### ***Functional Overview***

The user will be able to access an assortment of both backgrounds and quotes from the DB. With that in mind, when they pull info from the DB, the user will have the option to update said image or quote. When this happens this image or quote will delete the old background/quote and act the new image or background is an addition to the composition.

### ***Impacts***

This will allow the user to easily modify compositions without having to go through excessive settings to change it.

### ***Requirements***

Ensure that the update function acts as a delete then update function to the process.

## Design Unit C (e.g. Automated Time Setting Process) (DWL)

### ***1.1.2 Functional Overview***

This functionality will be implemented by use of a slider, this slider will allow the user to either lock or unlock both the quotes and the backgrounds in terms of their rotations, being fixed, sequential or randomly.

### ***Impacts***

This slider will allow the user to be able to configure the time setting process in a simple, yet effective manner.

### ***Requirements***

Ensure that the feature contains a function enabling the application to configure both the quote and the background time settings at the same time.

# Open Issues

This section should be part of the document only when the document is in a draft form

# Acknowledgements

Authors:

Dayne Longtin

Eric Gomez

Michael McCrae

Brittany Lynch

Jimmy Ellis

# References

All references should include, author, the title of the document, doc ID# and issue date. Please use the following free web site to help generate references/citations from other resources. <http://www.harvardgenerator.com/>

# Appendices

List all appendixes here.