"Motiv 8" : An Inspirational Media Interface

Gabriel Rodriguez
Montclair State University

Follow this and additional works at: https://digitalcommons.montclair.edu/etd
Part of the Computer Sciences Commons

Recommended Citation
https://digitalcommons.montclair.edu/etd/7

This Thesis is brought to you for free and open access by Montclair State University Digital Commons. It has been accepted for inclusion in Theses, Dissertations and Culminating Projects by an authorized administrator of Montclair State University Digital Commons. For more information, please contact digitalcommons@montclair.edu.
Abstract

We are molded by our environment and our surroundings. In a society immersed in social media, our environment is the content we consume. Social media has various positive benefits such as networking, e-commerce, connecting past acquaintances, and a source for entertainment. However, there are also various unfavorable characteristics that are found in most social media networks, which are tackled in this project. These characteristics include the relationship between content creator and content consumer, and the content that is displayed to the user. Whether miniscule or aggressive, the content a user consumes will have an impact on them. Therefore, content consumers are at the mercy of the content creators. As it is clear to see, content on social media is not always appropriate, desired, or valuable. The objective of this study is to find a solution to the negative facets of social media by creating a sub-branch/form of social media, “Inspirational Media,” to which “Motiv8” will fall under.

This thesis has two major steps: to study the effects of social media on content consumers and to develop a social media application, "Motiv8," that can avoid negative aspects of traditional social media. By studying the characteristics of current social media applications, it was discovered that the majority of content a user views lacks their interest and rarely contains thought provoking material. Motiv8 increases the likelihood that users will be interested in the content they consume by displaying content on topics, selected by the user, rather than content created by "followed" users. Furthermore, Motiv8 filters negative content by limiting content to topics that promote positivity, personal growth, and education. Since, the application is user-based, a vast majority of users are needed in order to analyze the results of the application properly
MONTCLAIR STATE UNIVERSITY

"Motiv8" an Inspirational Media Interface

by

Gabriel Rodriguez

A Master's Thesis Submitted to the Faculty of

Montclair State University

In Partial Fulfillment of the Requirements

For the Degree of

Master of Science

Computer Science

January 2017

College/School

Computer Science

(e.g. College of Science and Mathematics)

Department

Computer Science

(e.g. Geoscience)

Thesis Committee:

[Signatures]

(Thesis sponsor signs online; type name under signature)

Thesis sponsor

[Signatures]

(Committee member signs online; type name under signature)

Committee Member

[Signatures]

(Committee member signs online; type name under signature)

Committee Member
“Motiv8” AN INSPIRATIONAL MEDIA INTERFACE

A THESIS

Submitted in partial fulfillment of the requirements
For the degree of
Master of Science
Computer Science

by

GABRIEL RODRIGUEZ
Montclair State University
Montclair, NJ
2016
# Table Of Contents

Table Of Contents........................................................................................................i-iii

Figures List.......................................................................................................................iv-v

1 Introduction...................................................................................................................1-4
  1.1 What is Motiv8? .................................................................................................2-3
  1.2 Thesis Overview .................................................................................................4

2 Development.................................................................................................................4-34
  2.1 Android Studio ....................................................................................................5
  2.2 Firebase ...............................................................................................................5-8
    2.2.1 Real-Time Database ......................................................................................6-7
    2.2.2 Storage .........................................................................................................7
    2.2.3 Authentication..............................................................................................7-8
  2.3 Class Objects ........................................................................................................8-13
    2.3.1 Topic...........................................................................................................8
    2.3.2 Category.......................................................................................................8-9
    2.3.3 User ............................................................................................................9-10
    2.3.4 Post ..............................................................................................................10
    2.3.5 Comment.....................................................................................................11
    2.3.6 Advice.........................................................................................................11
    2.3.7 Conversation..............................................................................................11-12
    2.3.8 Message......................................................................................................12
    2.3.9 ReportedItem ............................................................................................12-13
  2.4 Views......................................................................................................................13-35
    2.4.1 Login...........................................................................................................13-14
3.1 Phase 2: UI Testing (With Backend)...............................................................35-36

3.3 Phase 3: Testing Post-Production.................................................................36-37

4 Results/Feedback........................................................................................................37-38

5 Future Work/Conclusion.........................................................................................38-39

6 References..................................................................................................................40

7 Additional Documentation.........................................................................................41-44
List Of Figures

Figure 1.1 Login View ..................................................................................................13
Figure 1.2 Lean More Prompt ......................................................................................13
Figure 1.3 Forgot Password Prompt .............................................................................13
Figure 2 Registration View ..........................................................................................15
Figure 3 Home View ....................................................................................................16
Figure 4.1 Filter Topics View (All Topics) .................................................................18
Figure 4.2 Filter Topics View (My Topics) .................................................................18
Figure 5 Select Topic View ..........................................................................................19
Figure 6 Suggest Topics View .....................................................................................20
Figure 7 New Post View ...............................................................................................21
Figure 8 Media Options View .......................................................................................22
Figure 9 Comments View .............................................................................................23
Figure 10.1 Get Advice View .....................................................................................24
Figure 10.2 New Advice View .....................................................................................24
Figure 11 Give Advice View .........................................................................................25
Figure 12 Advice To Give View ...................................................................................26
Figure 13 Messaging View ............................................................................................27
Figure 14 My Motivators View ......................................................................................28
Figure 15 Profile View ..................................................................................................29
Figure 16 Settings View ...............................................................................................30
Figure 17 Suggested Topics View ................................................................................31
Figure 18 Posts (Admin) View .....................................................................................32
Figure 19 Reported Conversations View .......................................................................33
Figure 20  Reported Users View .................................................................34
Figure 22  Firebase Crash Reporting Chart .............................................36
Figure 23  Firebase Analysis .................................................................37
1 Introduction

In an era where internet access is easily accessible, the use of social media has become a daily ritual for many connected users. Social media has various positive values such as networking, entertainment, growth of businesses, and it is a powerful platform to voice one’s opinion, and to display one’s talents and skills. However, social media also has a negative impact on society. The main unfavorable facets that are found in most social media tools, which are tackled in this project, are the relationship between content creator and content consumer, and the content that is displayed to the user. All content has an effect on consumers; miniscule or aggressive. Therefore, content consumers are at the mercy of the content creators. Content on social media is not always appropriate, desired, or valuable. In the cases where users are using social media to connect with others, or as a form of entertainment, content may not be a problem. However, there are users viewing social media as a source of information and advice. In this case, the context of the content is crucial. When this is the case, content has a larger effect on users and the wrong content can have adverse repercussions. This is especially true with younger content consumers, and with users in the process of self-development, for they are highly susceptible to the content they view on social media. Motiv8 was developed to tackle the negative aspects of traditional social media.

“…social media poses a grave threat to the humanities because it lacks the depth, nuance and permanence that make genuine, meaningful interactions about the human condition possible.” [6]

In addition to the negative content on social media sites, some content lacks depth and insight. Not only has social media caused a decrease in attention span, but it also
decreases critical thinking skills. This is caused by the “rapid-fire attention shifting” [2] content of current social media networks. Content is consumed in quick glances and immediately redirected to other content. With the topics selected in Motiv8, the content creator is forced to create content with depth and with meaning. In return, content consumers will gain positive growth and insight through content that requires more critical thinking an analysis.

"Emotional states can be transferred to others via emotional contagion, leading people to experience the same emotions without their awareness." [1]

The overall purpose of Motiv8 is to spread positivity through the concept of emotional contagion. The concept of emotional contagion states that negative, or positive, emotions can be spread from person to person. In researches, such as the one done in "Experimental evidence of massive-scale emotional contagion through social networks," [5] studies have been made to show the massive effects of this concept in social media networks. The desired outcome of Motiv8 is for it to become a highly accessed platform in order for its content to dilute the negative stream of data in standard social media networks.

1.1 What is Motiv8?

Motiv8 is a social/inspirational media application that allows users to create and view content on select topics of interest. The topics of the application is geared towards positivity and personal growth. Current social media tools, such as Facebook, display feeds based on the people/pages the user "follows." Rather, Motiv8 displays content on specific topics selected by the user. The process of displaying content on traditional social media causes a user’s feeds to include unwanted content. The contents on Motiv8
is linked to the topic selected by the content creator. For instance, a story on what motivated a user to become a teacher will be categorized under the topic area “Fields Of Study>Education.” A user that chooses to receive content on the topic will receive feeds, from various content providers, only on the topics chosen by the user. This filters out unwanted content. Motiv8 promotes growth through, inspiration, motivation, education, and enlightenment. In addition, users will be able to request personalized advice on a certain topic and respond to other advice request.

A unique characteristic of Motiv8 is its content filtration and content categories. Although tools, such as Pinterest, allow users to filter feeds on topics as well, its content type is limited to images, purchases, links and articles. Another aspect that is unique to Motiv8 is its content publication authorization. Some of the topics a user may choose to receive feeds from may be sensitive in nature. For example, “Dealing with Depression” or “Health>Mental.” It is crucial that the messages being portrayed in these topics promote the correct information. Therefore, rather than using an algorithm to search for flaggable content, each post is screened by an admin user.

Motiv8 is intended for anyone in the process of personal growth, in need of proper guidance on life events, or anyone desiring to share their positive wisdom on specific topics. In a society clouded by negativity, it is difficult to find a source for constantly-positive content. The desired users for Motiv8 will focus on helping fellow members of the Motiv8 community in the area of personal growth. This can be done by sharing positive experiences and valued information through the topic feeds.
1.2 Thesis Overview

The following sections of this thesis discusses the development of Motiv8, successes and failures, future development plans, and surveys from current users. The development section contains information on tools used to develop the application, design details, functionality, and classes with focus on the backend system and the interactions with the user interface. Throughout the development has been various successes and failures that propelled, or hindered, the success of the application. The current version of Motiv8, in production, and plans for future development of the iOS version is discussed.

2 Development

The initial plan was to develop a cross platform application, with a single code set, through one of the following platforms: Microsoft Xamarin with Microsoft Azure backend, Phonegap, and Appcelerator. Xamarin is a relatively new environment and was not suitable for the beginning stages of development. Reviews for Phonegap suggested it does not function well with media-rich content. Appcelerator reviews report numerous bugs that are preferred to be avoided. Due to these factors and time constraints, it has been decided to develop the initial application with the best native development environment for Android, Android Studio. Limiting the application to Android devices allowed for time to be dedicated to business logic and handling backend transactions. In addition, 80% of all mobile devices in use are Android based. In addition to Android Studio, Google’s Firebase Cloud Service is used due to its close connection to Android development and being a feature of Android Studio itself.
2.1 Android Studio

Android Studio is an Integrated Development Environment (IDE) developed specifically for Android platform development. Released in 2013, it has become one of the most widely used IDE's for mobile app development. It is developed in such a way for developers of all levels to be able to function, learn, and maneuver. With code written in Java and XML, a developer can easily develop applications suitable to their specifications. The intellisense feature allows for code correction and code completion. A graphical user interface layout editor is implemented for the developer to drag and drop controls to the views rather than only developing the view in XML. Another feature provided by Android Studios is the use of emulators. After a developer creates an application, Android Studio allows for testing of apps in emulators. An emulator refers to a hardware or software that is able to mimic another hardware or software. Android Studio contains emulators for, devices such as the Nexus 5, Nexus 6, Pixel C, tablets, and Android Wear devices. Emulators were used at the beginning phases of the Motiv8 development. Although the emulators are extremely useful, they have a slight disadvantage for applications that implement Google Services. For the emulators available on Android Studio the latest version of Google Services in inaccessible. Therefore, physical, devices were used for testing in later stages of the Motiv8 application development.

2.2 Firebase

Google Firebase is a cloud computing platform that allows developers to implement a backend environment for mobile and web applications. Firebase is furnished with a variety of features that include the following: a real-time database, a file storage
system, application analytics, web hosting, test labs (mobile application testing), and crash reporting. A combination of these features can be used for the requirements of the application in development. Applications such as Shazam, Asus, Booking.com, DailyMail.com, SnapChat, and Pocket Casts are currently implementing the Firebase platform. In the development of Motiv8, the real-time database, file storage system, analytics, crash reporting, test lab, and authentication features are used.

Cloud computing is the use of a network of off-site servers, that are accessible through the internet, to complete tasks such as data management, storage, and hosting. Cloud computing benefits start-ups and small businesses that do not have the means of having on-site servers. With the increase in mobile apps and mobile websites, the need for cloud computing has rapidly risen. With the use of mobile devices, certain needs are presented for a backend system such as scalability, speed, and flexibility. In today's era of computing, scalability and speed plays a tremendous roll.

2.2.1 Real-Time Database

The Firebase real-time database is a NoSQL database. NoSQL, also known as Not Only SQL, databases have non tabular formats and allow for the insertion, deletion, and updates of data without knowing the data schema of the database. This method of data storage allows for more control of scalability, the data itself, and allows for a simpler design. NoSQL databases consists of a key-value pair structure. The object is converted to JSON which can then be easily accessible by a variety of platforms. This process of storing data is known as the "data first, schema later/never" approach and allows for flexibility. The data is known as Flexible Schema Data. In applications where an object's data structure changes frequently, Flexible Schema Data is beneficial. The Firebase Real-
time Database is synchronized in real time across all connected users. The Firebase database can be used with iOS, Android, and JavaScript. All users, across all three potential platforms, share a single database. Every change in the database is received by any connected user within milliseconds. This is done, through Firebase, by utilizing synchronization rather than an HTTP request.

2.2.2 Storage

Firebase utilizes the Google Cloud Storage platform to implement storage features. Firebase storage also implements Google's security protocols to protect file uploads and downloads. In Motiv8 a security rule is implemented such that only authorized users are able to read or write to storage. Using Firebase for storage has three fundamental benefits: scalability, security, robustness. Since Firebase Storage is built on the Google Cloud Storage Platform, it is extremely scalable. It is scalable to petabytes of data. In some cases, a user may lose connection while uploading files to a server. With Firebase, downloads and uploads are performed regardless of the device's network quality. When connection is lost, or connection quality changes, Firebase will restart where the process ended. Firebase also allows for custom authentication rules to set security parameters based on content type, filename, authentication status, and other metadata.

2.2.3 Authentication

Firebase also provides authentication services based on the OAuth standard. The Firebase authentication process uses tokenization to allow user account information to be accessed by other services without exposing the user's credentials. This process allows users to create database connections using third party tokens such as Facebook, Gmail,
Twitter, GitHub and others. It increases the user's experiences by not demanding him/her to create new credentials for every application. Currently, Motiv8 only implements Email credentials. In future development, the ability to sign in with Facebook, Gmail, and Twitter will be implemented.

2.3 Class Objects

2.3.1 Topic

As previously mentioned, part of what sets Motiv8 apart from other social media platforms is its topics. Topics are meticulously chosen from areas that may promote personal growth or enlightenment. From topics such as positivity, entrepreneur, fitness, and success, there is an attempt to filter inappropriate/undesirable content. As of the data of this thesis, December 6th 2016, there are seventy topics. Users are able to suggest topics and administration is able to add topics. Topics are used to filter content feeds on the user's home screen. When a user creates a post, the user selects a topic to which the post refers to. For instance, if a user creates a post suggesting an exercise regimen, the post will be in the topic "Fitness." The topic class has two properties: Name and Category. In the Firebase real-time database, the topic objects are in the "topics" node with keys generated by Firebase based on the time it was added.

2.3.2 Category

Categories are used to easily traverse through topics. Each topic, with the exception of non-categorized topics, are grouped within categories. Non-categorized topics are topics that appear without having to traverse a category. These include "Simply Positivity" and "Success." Non-categorized topics are meant to be presented to the user, before other topics, when filtering through topics. A category can contain a list of
categories and/or a list of topics. Some categories also contain a parent category. The parent category, child categories, and topics are referenced by their key values.

2.3.3 User

An application user is represented by the class object "User." The User object contains the following information: username, profile image, gender, date of birth, date created, if the user is anonymous, if the user is an administrator, if the user profile has been locked by admin, a list of selected topics, a list of posts/comments that motivate the user, a list of posts/comments that the user reported, a list of advice that the user reported, a list of users that motivate the user, number of posts created, and number of advice given. During the registration phase, the user creates a unique username. Profile image, gender, and date of birth are optional properties. The profile image property contains the name of the image file that references a file in storage. If the user decides to set their profile to anonymous, their posts will still be received in other user feeds. However, their profile will not display their collection of posts to any users. A user has a list of selected topics to view in their feed. A user may also be an admin user. Admin users have access to a variety of views that are later discussed in this thesis. Currently, there are no actions taken when a user is locked. To comprehend the best repercussions for users who violate the terms of the application, more evaluation must be done. As the application's traffic grows, the purpose of having the "postCount" and "adviceCount" parameters is to award users who create a large amount of valid content. These awards can be through physical gift cards or future application perks. The non-admin user has a variety of actions that include the following: select topics, suggest topics, create posts/comments/advice, delete own posts/comments/advice, give advice, report advice/conversations, add/remove users
to/from motivate list, and report users. An admin user can perform all actions of a non-admin user. In addition, an admin user can approve/deny posts, approve/deny advice, approve/deny conversations, view reported conversations, add topics, add categories, and lock users.

2.3.4 Post

Posts are the main focus of the application. Users create Posts that other users receive in their feeds based on the receiving user's selected topics. A Post has a specific Topic to which it is grouped in. Posts are displayed on the Home view of all users with the corresponding Topic selected. Posts are also displayed in a user's Profile view. A Post can contain, text, images, and a video. The Post may contain a maximum of five images and one video. Other parameters of the Post class object are as follows: Title, User ID, if the post is locked, if the post has been reviewed by an admin, motivation count, reported count, and post date. In addition to text, a user can enter a title to their Post. The user id references the user the created the post. If a Post contains undesirable/ inappropriate content, an admin user has the option to lock a Post. A Post that is locked will not be displayed in any users' home screen. However, the Post can still be viewed in the post-creator's Profile view. A locked Post is marked by a red flag. Also, the Post creator can request a re-evaluation of the locked Post if he/she deems it appropriate. The current process of ensuring appropriate content is to review each post. Therefore, the "isReviewed" parameter determines if the Post has been reviewed by an admin user. Each time a user selects that a Post motivates them, the integer value of "MotivatedCount" will be incremented. In the same manner, the "ReportedCount" will be incremented if a user reports the post.
2.3.5 Comment

The Comment class object is a subclass of the Post object. The Comment object also contains the key value for the Post to which it pertains to. A user may comment any available Post. Such as the Post object, a user can mark a comment as "Motivates Me" or report it.

2.3.6 Advice

Another key component to Motiv8 is its advice section. Most social media applications allow users to directly message other users. However, in Motiv8, a user may only message a user that has requested advice on a topic. The purpose of the advice section is to create one on-one-conversations between users only by the request of the receiving user. When a user requests Advice, they are creating a Post that is stored in the "advice" node of the database. Like a Post, an Advice has a Topic to which it is grouped by. If an Advice request is reported by another user, an admin user receives the report and is able to lock the Advice. An Advice that has been locked may not receive additional responses from other users.

2.3.7 Conversation

After a user posts an advice request, other users may choose to respond to the request. A conversation refers to the communication between one user and another. An advice request can contain one, or more, conversations. The Conversation class object contains the following properties: AdviserID, AdviseeID, AdviceID, AdviserDeleted, AdviseeDeleted, reported, and isLocked. The user who requests the advice is the advisee, The AdviseeID string property contains the key value for the advisee. With the same respect, the AdviserID string property contains the key value for the advisor. If either the
advisor or advisee choose to delete the conversation, the other user still has access to the conversation. Therefore, the boolean properties AdviserDeleted and AdviseeDeleted are utilized. A user may also report a Conversation. A reported Conversation may be reviewed by an admin user. If the admin locks the Conversation, the advisee and the adviser still have access to the messages. However, they will not be able to create new messages in said Conversation. A Conversation is created when the adviser sends the first message.

2.3.8 Message

A Conversation contains a collection of key values of instances of the Message class object. The Message class object represents a single message transferred between users. The Message class contains the following properties: UserID, ConversationID, Text, MessageDate, imagePath, videoPath, and read. The UserID string property is the key value for the user who created the message. The ConversationID string property is the key value for the Conversation to which the Message pertains to. A Message may contain a single image or a single video. The boolean property "read," refers to if the receiving user has read the message.

2.3.9 Reported Items

Throughout the application, a user has the option to report inappropriate activities in conversations, advice requests, and user actions. To store the reported data, the ReportedItem class object is used. It is also used to store Topic suggestions. The ReportedItem class object has the following properties: Key, Reasons, ReviewedBy, ReviewComment, ReviewDate. The Key references the key value of the item being reported. For Topic suggestions, the Key value is an empty string. When an item is
reported, the application prompts the user to enter a reason for reporting the item. The reason is stored in the list parameter "Reasons." If an item is reported more than once, a new instance of the ReportedItem class is not created. Rather, a new string is added to the Reasons list. After an item has been reported, an admin user receives the detail and takes appropriate actions. When an admin user resolves the reported item, the admin user has the option to leave a comment for other admin users to view. The object then stores the review date and the username of the admin who reviewed it.

2.4 Views

2.4.1 Login

The login screen is the first screen displayed, given the user is not signed-in. The login screen has seven controls. The logo is displayed at the top of the screen and covers one third of the view. The logo is displayed in an image view that animates through a collection of images. The animation displays an oak tree growing from sapling to full

![Figure 1.1: Login View](image1)

![Figure 1.2: Learn More Prompt](image2)

![Figure 1.3: Forgot Password Prompt](image3)

- **Figure 1.1: Login View**
  - Shows a description of the application.
- **Figure 1.2: Learn More Prompt**
  - User enters an email to receive a password reset email
maturity. The reason for having a growing tree as a logo is that it embodies the meaning of the application: the promotion of personal growth. The growing tree, with the rich green style of the application, promotes a sense of positivity.

If the user already has an account, the user will enter his/her email address in the email input and password in the password input. After the email and password sections are complete, the user will click the "Log In" button. The "Log In" button runs a function to check the user's credentials against the Firebase Authentication server. If the credentials are invalid, the user will be prompted that his/her email and/or password is incorrect. Otherwise, the application will open to the landing screen.

If a user forgets his/her password, they will click on the "Forgot Password?" Control. A prompt appears for the user to enter their email address. If the user has already entered an email address in the email input of the Login view, the email will be transferred to the "Forgot Password" input. Firebase Authentication tools handles requests for password resets. The user will receive a link, generated from Firebase, to enter a new password. When the process is complete, the user will be returned to the Login screen to attempt to log in again.

On the lower right of the Login view is the "Learn More" label. Clicking the label will display a prompt that provides the user with more information regarding the application.

Lastly, if a user is new to Motiv8, he/she will register a new account by clicking the "Sign Up" button. The button will open the Registration view.
2.4.2 Registration

A new user must register a new account with Motiv8. This process is handled in the Registration view. The registration view prompts the user to enter their email address, a unique username, a password, a confirmation of the password, a date of birth, a gender, an anonymity option, and to select a profile image. The date of birth, gender, and profile image are optional parameters. Since Motiv8 does not focus on the user, it focuses on the content itself, Motiv8 does not require any identifying information. Users are able to share their thoughts in an anonymous-like manner. Every user has a unique username. After the user enters a complete desired username, the database is queried to assure that the username does not already exist. Also, when the user clicks the "Continue" button, the Firebase authentication tools check if the email is already in use. If either the username or email address is not unique, the user will be prompted to make changes. The email input control checks for valid emails by using Java's pattern matching function to
match for an email address. Firebase Authentication requires passwords of over five characters. Therefore, The password input implements a text change listener and displays an error when the entered password is less than six characters. Clicking "Continue" will create a new instance of the "User" object and store it in the "users" node of the real-time database. After the user is created, the Home screen will be displayed.

2.4.3 Home

![Home View](image)

*Figure 3: Home View*

The landing view for a non-admin user.

If a user has already signed in, and is not an admin, the application will start at the Home view. The Home view contains a collection of the most recent posts available by to the user. The displayed posts are determined by user preference. The Home view is a fragment within an Android viewpager activity. As the development of Motiv8 progresses. Additional fragments will be added to the viewpager. The Home fragment consists of a recycler view, and two floating action buttons. The recycler view consists of cards that represent a post. The post card view contains information about the post including the username of the post creator, the profile image of the creator, topic, post
text, possible post title, possible images, a possible video thumbnail, motivated count and date the post was created. In addition, the post card view contains three buttons: "Motivates Me," "Comments," and "Report Post." If the "postsMotivatesUser" list property of the user object contains the post's key value, the "Motivates Me" button will be displayed with green text. Likewise, if the "postsReportedByUser" list property contains the post's key value, the "Report Post" button will be displayed in red text. Clicking the username will open the Profile view for the creator. If the post references a video in storage, the post will display a video thumbnail with a play image. When the play image is clicked, a new view with a video player will open. If the post contains images, the images will appear in a horizontal scroll view. Clicking on the image scroll view will open an activity that displays the images in full screen. If the post contains images, the images will appear in a horizontal scroll view. Clicking on the image scroll view will open an activity that displays the images in full screen. If the post text does not contain a video, and the text contains a link to a YouTube video, a YouTube thumbnail view will be displayed. Clicking on the YouTube thumbnail will open a stand-alone youtube player for the video.

When the Home view starts, it gathers data from the database on the user. On successful retrieval of the user data, the recycler view adapter gathers post information from the database based on the "userTopics" list property of the user object. For each topic key in the list, five of the latest posts with said topic will be displayed. And sorted by the key value of the post. Since Firebase generates key values based on timestamps, sorting posts by key value sorts the posts in chronological order. The recycler view is within a Swipe Refresher Layout. When the user swipes downwards, new posts are displayed. When a user swipes downwards, five new posts for each user-chosen-topic is added to the recycler view.
The user can add, or remove, topics from their chosen topics list by clicking the "Filter Topics" image button; which is represented by a funnel. Clicking the button will open the Filter Topics dialogue view. If the user would like to create a new topic, the user will click the "New Post" image button; which is represented by a plus sign. Clicking the button will start the New Post activity.

2.4.4 Filter Topics

![Figure 4.1: Filter Topics View (All Topics)](image1)

Displays all topics, and categories, to filter from.

![Figure 4.2: Filter Topics View (My Topics)](image2)

Displays topics the user has chosen.

The Filter Topics view utilizes the Filter Topics Activity to allow users to choose what topics to receive feeds on. The view contains two fragments, "All Topics" and "My Topics." The All Topics fragment displays a list of all categories and topics. The All Topics fragment contains a collection of fragments that are created and removed during runtime. Topics and categories without a parent category are displayed on the first fragment of the All Topics fragment. Topics are displayed with checkboxes. If the user's
topics list already contains the topic's key value, the checkbox will be checked. Otherwise it will not be. When a user clicks of a topic, it will add or remove the topic from the list. When the user clicks on a category, a new fragment is added, to the All Topics fragment pager, that will display the category's child topics and categories. The "My Topics" fragment contains a list of all topics chosen by the user. The topics are also displayed with checkboxes. When a topic is unchecked, it is removed from the fragment. With each click of a topic, the list is updated in the user object in the database. When the user returns to the Home view, it will add or remove posts based on the changes made in the Filter Topics view.

2.4.5 Select Topic

![Select A Topic View](image)

**Figure 5: Select Topic View**

View for selecting topics when creating a post, advice request, or searching for advice to give.

Similar to the Filter Topics view, Select Topic uses the Filter Topics Activity and displays the "All Topics" and "My Topics" fragments. Rather than displaying the topics
in checkboxes, the topic is displayed in a standard text view. When clicking on a topic, an action is performed. The Select Topic fragment is displayed in various actions of the app. These locations include creating a new post, requesting advice, and searching for advice to give. In each instance, the Filter Topics Activity will return the key value to the selected topic.

2.4.6 Suggest Topic

![Figure 6: Suggest Topics View](image)

User enters a suggestion for a topic.

Both the Select Topics and Filter Topics views are fragments within the Filter Topics Activity view. In addition to a view pager for the fragments, the activity contains an options button on the upper right corner of the dialogue. Selecting this button will open the Suggest Topics view, which utilizes the Report Activity, to prompt the user to enter their suggestion. When the user clicks the send button, a new instance of the ReportedItem class is added to the "suggested_topics" node of the database. The
suggested topic is then displayed in the admin user’s Suggested Topics view for the admin to take appropriate actions.

2.4.7 New Post

![New Post View](image)

Figure 7: New Post View
User creates a new post.

New posts are created in the New Post view that utilizes the Add Post Activity. The Add Post Activity is started when clicking the "New Post" button in the home view. Before the user is able to see the New Post view, the Select Topic view is displayed. When the user successfully selects a topic, the New Post view is displayed and the name of the topic is displayed in the top right corner. The New Post view has five additional elements: the input for the title, the input for the text, a "media" button, and a "send" button. The title of the post is an optional parameter. Below the title, the user may enter text to the post. The user may also add images, or a video, by clicking the "media" button. There is a limit of one video and five images for each post. The images and video thumbnail are displayed in a horizontal scroll view above the post text input. Each element contains a floating action button that will allow them to remove the element from the post. When a video element is clicked, the video will play in a full screen player view. When an image element is clicked, the image will be displayed in full screen. A new post may not be submitted if the post text is empty and a video or images do not exist. When
the "send" button is clicked, and the post contains content, the Add Post Activity creates a post object and uploads the post to the "posts" node of the database. Images and videos are stored in the Firebase Storage server, and placed in a new directory for the post. When the all items of the post have been successfully uploaded, the user will be notified with a Toast.

2.4.8 Media Options

When creating a new post, advice request, comment, or message, the user has the option to add images or a video. Images and videos are added through the Media Options views. The view contains three options: Image Gallery, New Image, Video Gallery, and New Video. The image gallery option allows a user to select images from their gallery. The application proprietary to their device will handle the image selection. In future development of Motiv8, the application will implement its own image selection options within the application. This will remove the variable that is using external applications.
The gallery allows multiple images to be selected for posts, comments, and advice requests. However, only single images are allowed for messages. The New Image option allows the user to take a new image using their device's camera application. As with using an external application to select images, a self-containing camera view will be implemented in Motiv8 in future development. The Video Gallery option and the New Video option works similar to the Image Gallery and New Image options. The activity of the Media Options activity returns a list of URIs for the media elements chosen and details on if the element is an image or video.

2.4.9 Comments

![Image]

*Figure 9: Comment View*

Displays a post, a list of comments, and input options to create a new comment.

When the "Comments" button of the Post card view is clicked, the Comments view is displayed. The Comment view contains a dropdown for the original post to be displayed. The original post is displayed in a quarter of the entire Comments view in a scroll view. The scroll view for the post can be hidden. Similar to the Home screen, the
Comments view contains a Recycler View that displays a collection of comments pertaining to the post. In the bottom portion of the Comments view, a user is able to add new comments. The same rules as creating a new post apply to creating new comments. When the send button is clicked, and the comment contains content, the comment will be stored in the real-time database under the "comments" node and under the post key's value node within the comments node.

### 2.4.10 Get Advice

#### Figure 10.1: Get Advice View

Displays a list of advice requests and conversations for the current user.

#### Figure 10.2: New Advice View

User creates a new advice post.

The Get Advice view and the Give Advice view Advice Page Activity. The Advice Page Activity contains a view pager that contains the two fragments. The Get Advice view contains all advice requests and conversations in an expandable list. The group view in the expandable list is the advice post. The child views of each group view represents a conversation for the requested advice. Clicking on the advice post will display the conversations list for the advice request. Similarly, to the standard post card
view, the advice view contains images, text, and a possible video. The click events for the images and videos perform the same tasks as in the post card view. A user can choose to remove the advice request by clicking the options button on the top right corner. The conversation views contain the username of the adviser, the adviser's profile image, and the text of the last message in the conversation. If the advisee has not read the latest message in an advice, both the group view and the child view will be shaded green. Clicking a conversation view will display the Messaging view. The Get Advice view also contains a floating action button to create a new advice request. Creating a new advice request is the same process of creating a new post and uses the same view. However, rather than uploading the data to the "posts" node of the database, it uploads the data to the topic key value node in the "advice" node. When a new advice request is successfully uploaded, it will be displayed in the advice list.

2.4.11 Give Advice

![Figure 11: Give Advice View](image)

*Figure 11: Give Advice View*
Displays a list of advice posts the current user is advising on.
The Give Advice view is the second fragment in the Advice Page view pager. Here, users will see a list of advice requests to which they are advising. The list displays an advice similarly to the group view of the Get Advice view. When clicking on the advice item, the Messaging view is displayed. The Give advice view contains a floating action button for the user to search for new advice requests to advise on. Clicking the button will open the Advice To Give view.

2.4.12 Advice To Give

![Advice To Give View](image)

*Figure 12: Advice To Give View*
Displays a list of advice requests available for the user to advise on.

The Advice to Give view displays a list of advice requests, for the given topic, to which the user may advise on. Here, the user is able to end conversations by clicking the options button of the list item. Prior to viewing the list, the Select Topic view is displayed for the user to select the topic to advise on. Advice that the user is already replying to, or advice that the user has reported, will not be displayed in the list. Clicking on an advice item will open the Messaging activity to begin a new conversation.
2.4.13 Messaging

The Messaging view contains a scrolling list that displays all messages for a given conversation. The messages from the receiving user is display on the right side of the view in a gray rectangle. The messages from the sending user is displayed on the left of the view in a green rectangle. For messages, a user is able to send a single image, or a single video, per message. New messages will appear at the bottom of the list view. A conversation is started when an adviser sends the initial message. In the Messaging view, users are able to delete the conversation or report it.
2.4.14 My Motivators

The All People Activity contains two fragment views: My Motivators and I Motivate. The All People Activity can be accessed by clicking the "group" icon on the action bar of the main screen. The My Motivators view contains a scroll list view that displays all users that motivate the current user. The list of motivators is in the User object. The My Motivators list uses pagination to retrieve ten users at a time. New items are displayed as the user scrolls down. The view for each motivator contains the user image, the username, and a "Add User" button. Clicking the item will open the profile view for the selected user. Clicking the "Add User" button, while it is shaded green, will remove the user from the list.

2.4.15 I Motivate

Similar to the My Motivators view, I Motivate displays a list of users. The list of users represents a list of all users that the current user motivates. The list is stored in the "user_motivatees" node of the database.
2.4.16 Profile

Figure 15: Profile View
Displays a list of posts for a given user.

The Profile view contains all posts from a user, regardless if it is locked. Similar to the Home view, posts are displayed in a recycler view, in chronological order, using card view. Given that the profile is not of the current user, a user can report the owner of the profile by selecting the options button in the top corner. Reported users are stored in the "reported_users" node of the database. If the profile belongs to the user, the user can change, or remove, their profile image by clicking on their profile image. Also, they will be able to request re-evaluations of their locked posts by clicking on the red flags of the posts in the Profile view. A user can view their own profile by clicking the username area of their posts. They can also access their profile by clicking the "person" icon in the action bar of the All People activity. If a user's profile is set to anonymous, their posts will not be displayed, regardless if the viewing user motivates, or is motivated by, the profile user.
2.4.17 Settings.

The Settings view displays various statistics of the current user and tasks. The statistics include the date the user created the Motiv8 account, the total amount of posts created by the user, number of advice given, number of users motivates, and number of users that motivate the user. The tasks in the Settings view include logging out, inviting other potential users, and setting anonymity. Clicking "Log Out" will remove the user from the local Firebase Authentication instance and start the Login view. Clicking "Invite A Friend" will open an email prompt to send the details to the Motiv8 application to various contacts. With the Firebase Tools comes Firebase Invites sets the views for inviting new users. The Invite A Friend view generates an email with information on the application and a link to the Google Play store. Future development of the application will include compatibilities for iOS invites. If the user is an admin user, he/she will be able to switch from/to admin view in the Settings view.
2.4.18 Suggested Topics

Figure 17: Suggest Topics View
Landing view for admin users.
Displays a list of requested topics.

If a user is an admin user, the landing screen view pager contains views, only accessible to admins, as fragments. The first of these views being the Suggested Topics view. After a user suggests a topic in the Filter Topics activity, the data is stored in the "suggested_topics" node of the database. The Suggested Topics view then displays the list of suggested topics. The view has to settings that can be selected from a drop down list: "To Process" and "Processed". The "To Process" option lists all Suggested Topics that has not been reviewed. The "Processed" option shows all suggested topics that have been reviewed. The items in the Processed option displays the username of the admin user who added the topic/category. Each Suggested Topic is displayed in a view with the suggestion and two buttons: "Approve" and "Deny." Clicking the Approve button prompts the admin user to select a category to group the suggestion under. Clicking the Deny process will set the suggested topic as reviewed and not prompt for a category. The Suggested Topics view also contains two buttons, "Add Category" and "Add Topic," at
the bottom of the view. Both buttons prompt to enter the name to be added and to select a category.

2.4.19 Posts (Admin)

![Posts View](image)

*Figure 18: Posts (Admin) View*

Displays a list of posts for the admin user to review.

The second fragment in the admin's landing screen view pager is the Posts view. The Posts view displays posts from all topics. Similar to the Suggested Topics view, the Posts view contains four options in a dropdown: "Not Reviewed," "Reviewed (Locked)," "To Re-Evaluate," and "Reviewed (Approved)." All new posts that have not been reviewed will be displayed when the "Not Reviewed" option is selected. The posts are displayed in a standard post card view. However, the "Comments" button is removed, the "Motivates Me," text is replaced with "Approve," and the "Report Post" text is replaced with "Lock." Clicking either will remove the post from the "Not Reviewed" list. Clicking "Lock" will lock the post. As mentioned earlier, locked posts are displayed with a red flag. Clicking "Approve" will approve the post. The "Reviewed (Locked)" option will
display all posts that have been locked by admin users. A user's whose post has been
locked can request a re-evaluation of their post by clicking the red flag. Posts to be re-
evaluated are displayed when the admin selects the "To Re-Evaluate" option. Lastly, all
approved posts are displayed by selecting the "Reviewed (Approved)" option.

2.4.20 Reported Advice

The admin's landing view contains three action icons in the action bar. The first is
the reported advice icon displayed by a "discussion" image. Clicking the discussion icon
starts the Advice activity. The Advice activity view pager, for an admin user, contains
two fragment views: "Reported Advice" and "Reported Conversation." The Reported
advice displays all Advice requests that have been reported by users. The options button
of the advice views displays a list of all reasons why the advice was reported. The admin
user has the option to block or approve the advice. Blocking an advice request prevents
new advisors from advising on the post.

2.4.21 Reported Conversation

![Figure 19: Reported Conversations View]

Displays a list of conversations for
the admin user to review.

The Reported Conversations view displays all conversations that has been
reported. Clicking on a conversation item will open the Messaging view for the admin to
review the messages in the conversation. The Messaging view for the admin removes all controls to send a message. An in the Reported Conversations view, an admin user has the option to lock a conversation. Locking a conversation prevents the adviser and advisee from replying to the conversation.

2.4.22 Reported Users

![Reported Users View]

*Figure 20: Reported Users View*
Displays a list of reported users for an admin to review.

The reported Users view lists all users that have been reported. Currently there are no repercussions for users that have been locked. To fully understand the best way to lock a user, additional testing with a larger user count must be done.

3 Testing

There were various tools utilized for testing the various versions of Motiv8 including the use of Firebase Test Labs, Firebase Crash Reporting, Android Studio Emulators, and physical devices. Various tools were used in various stages of the development phase. Part of the test cases or displayed on Page.

3.1 Phase 1: UI Testing (No Backend)

Phase one of testing including testing the activity flow with autonomously generated dummy data during, and after, the development of the views and activities. The views were developed in the order a user would traverse the application (i.e. First the
Login, then the Registration view, the Main activity, etc.). During, and at the completion of each view, dummy data was entered to fill the views with content. For instance, a loop was created to generate a list of posts with dummy data including text and “Motivates” counts. The layouts were tested using various Android emulators provided in Android Studio. These emulators included the Nexus 5, Nexus 6, and Nexus 10. Various test cases were tested that were ran to assure proper user experience. Test cases for the Login and Registration can be viewed on page 41.

3.2 Phase 2: UI Testing (With Backend)

After successful testing of the initial front-end, the Firebase Cloud Services connections were implemented. For each view, there is a collection of test cases for Firebase requests and transactions. For instance, in the main activity, there is a process for determining if an admin view should be displayed, or a non-admin view. The main activity achieves this by collecting the user data from the database during the “onStart()” of the application. If the user object that is returned has a value of “True” in its “isAdmin” property, the admin view is displayed. It also checks the settings to determine if the admin is temporarily choosing to view a non-admin view. To test the results, a non-admin user signed in, an admin user signed with the option to view as admin, and an admin user signed in with the option to view as a non-admin. Each test resulted in the correct results.

The majority of testing the UI with the Firebase backend resulted in expected results. However, there are various areas that did not result as expected and still require additional testing. These areas include the asynchronous results for retrieving images from Firebase Storage within bounded items in RecyclerViews. For instance, a post that
contains a collection of images required the asynchronous “getDownloadableURI” method from the Firebase Storage class. Since the result of the method occurs after the view has been bounded, the images displayed in the incorrect post view. The latest version of Motiv8 adds a tag value to the post view and compares the tag value within the asynchronous method to avoid adding images to the incorrect view. The solution is currently functioning. However, testing with higher traffic must be performed to assure it is a robust solution.

In order for Firebase to function on an Android device. The device must have the latest version of Google Play Services. The emulators used by Android Studio utilizes machine images that are updated when Android Studio pushes updates. The emulators are not updated with the same consistency as Google Play Services updates their tools. Therefore, testing Motiv8 with the backend implantation is performed on physical devices. This limited the availability to test on different device configurations such as screen size and speed. In addition, Firebase provides a tool names “Test Labs.” Test Labs runs the application through various “behind-the-scenes” tests to assure the application will run on a number of user-selected devices and versions.

3.3 Phase 3: Testing Post-Production

![Firebase Crash Reporting Chart](image)

*Figure 22: Firebase Crash Reporting Chart*
Firebase Crash Reporting chart displaying bugs from production and testing.
In addition to Test Labs, Firebase provides a tool called “Crash Reporting.” Crash Reporting displays statistics and data on crashes that occur across all instances of the mobile application. It provides information equivalent to debugging reports when testing. In addition, the report includes information on the device including Android version, device model, service provider, and network state. This tool provides for a potentially larger range of devices to receive testing information on. However, it is not an ideal form of test. It is preferred that bugs are caught by the tester, not the customer.

4 Results/Feedback

![Firebase Analytics](image)

Figure 23: Firebase Analytics
Firebase Analytics statistics for the first 3 weeks on the Android Market.

Motiv8, like any other social media application, is a user-centric platform. Therefore, to accurately complete studies and tests, a large amount of users is needed. A large amount of users will produce more posts, advice, and traffic. As of the date of this paper, there are a total of nine users and ten posts. Using Firebase's analytics tools analysis was performed for the first month of deployment with 11 users. The average session count of 4.5 daily sessions per user. Due to the lack of content, the average
session duration is one minute and 44 seconds. An increase in traffic will increase the
sessions statistics.

For the majority of invited users, the common problem was the iOS compatibility. As previously mentioned, the application is currently only available for the Android operating system. Other invited users had a lower Android version than what was required to run the app. Of those users invited with the correct device configurations, six completed a feedback. The feedback was overall positive. Users found the application easy to maneuver and understood the concept of the application. With earlier versions of the application, the primary request was the addition of video features. The video upload and playback features were added in version 1.5.0. Additional requests included adding the number of available advice requests in the Select Topics view in the Give Advice process. The feedback was positive with no reported any fatal exceptions.

5 Future Work/ Conclusion

For the Android version of the application, there are various enhancements to be made. Firebase tools provides a Messaging service for notifications. Notifications can be implemented in the advice section of the application. Especially with a low user count, it may take time for a user to receive a response to their advice request. Rather than continuously checking the Get Advice view, Motiv8 will use notifications to notify the user that a new conversation has started. An additional Firebase tool to enhance upon is the Authentication tools. Currently, users can only register for the application by creating a new set of credentials. As previously mentioned, Firebase authentication uses the OAuth standard and allows users to create accounts by tokenization. Future development of Motiv8 will allow users to sign in and register using Facebook or Twitter credentials.
Currently, storage and playback of videos uses a method that is not of best practices. When a user uploads a video, it is directly stored into the Firebase storage as-is. In upcoming versions of the application, Motiv8 will use transcoding to enable adaptive streaming with HLS and MPEG-DASH. Using this method will allow seamless playback of videos with optimal quality. To avoid the use of third party applications, Motiv8 will also include an embedded camera application and gallery media chooser.

As mentioned, the majority of invited users happened to own iOS. Therefore, before marketing the application to increase traffic, the iOS version will be developed. Given the configuration allows Firebase services, the first option for creating the iOS version is to implement Microsoft Xamarin as initially desired. Xamarin will allow for a single project producing an application for both Android and iOS.

Overall, the results of the application are inconclusive. In order to fully analyze the effects of surrounding a user with positive and effective content, a large collection of user data is needed. The purpose of the application is to promote personal growth. For an application such as this to make an impact on society, a high traffic rate is crucial. When the application reaches said heights, further analytics can be performed. Along with a change in the type of content users will consume, Motiv8 will have other benefits. For instance, the analytics can yield information on human desires. Analysis can be performed to determine correlations between topics that users need motivation on, gender, and age. The desired outcome of Motiv8 is to begin adding positive and thought provoking content back to social media. With the correct marketing approaches, this outcome is possible.
6 References


## 7 Additional Documentation

### Test Cases (Login and Registration)

<table>
<thead>
<tr>
<th>View</th>
<th>Scenario</th>
<th>Test Steps</th>
<th>Expected Result</th>
<th>Outcome</th>
</tr>
</thead>
<tbody>
<tr>
<td>Login</td>
<td>Verify that user enters correct credentials when signing in.</td>
<td>Do not enter an email address. Click “Log In.”</td>
<td>Prompt displays to notify the user to enter an email address and password.</td>
<td>Expected results.</td>
</tr>
<tr>
<td>Login</td>
<td>Verify that user enters correct credentials when signing in.</td>
<td>Enter an email address. Do not enter a password. Click “Log In.”</td>
<td>Prompt displays to notify the user to enter an email address and password.</td>
<td>Expected results.</td>
</tr>
<tr>
<td>Login</td>
<td>Verify that user enters correct credentials when signing in.</td>
<td>Enter the correct test credentials.</td>
<td>The main activity is displayed.</td>
<td>Expected results.</td>
</tr>
<tr>
<td>Forgot Password</td>
<td>Verify that the user enters a valid email address.</td>
<td>Do not enter an email address.</td>
<td>Display a Toast prompting to enter a valid email address.</td>
<td>Expected results.</td>
</tr>
<tr>
<td>Forgot Password</td>
<td>Verify that the user enters a valid email address.</td>
<td>Enter an email address that does not match the test email address.</td>
<td>Display a Toast prompting that the email address does not exist.</td>
<td>Expected results.</td>
</tr>
<tr>
<td>Forgot Password</td>
<td>Verify that the user enters a valid email address.</td>
<td>Enter the correct test email address.</td>
<td>Display a Toast prompting a password reset email was sent.</td>
<td>Expected results.</td>
</tr>
<tr>
<td>Sign Up</td>
<td>Verify that the user enters a valid email address.</td>
<td>Enter all data but the email address.</td>
<td>Display an alert prompting the user to enter an email address. Display a message below the email input.</td>
<td>Expected results.</td>
</tr>
<tr>
<td>Sign Up</td>
<td>Verify that the user enters a valid email address.</td>
<td>Enter the test email address.</td>
<td>Display an alert prompting the user that the email address is already in use.</td>
<td>Expected results.</td>
</tr>
<tr>
<td>Sign Up</td>
<td>Verify that the user enters a valid email address.</td>
<td>Enter all data with an email address</td>
<td>Display a toast stating that the account is</td>
<td>Expected results.</td>
</tr>
</tbody>
</table>


<table>
<thead>
<tr>
<th>Sign Up</th>
<th><strong>valid email address.</strong></th>
<th><strong>different than the test email address.</strong></th>
<th><strong>created. Open the main activity.</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Verify that the user enters a password 6 characters in length or greater.</strong></td>
<td><strong>Enter a password less than 6 characters in length,</strong></td>
<td><strong>Display a message below the password input stating the error. Disable the “Continue” button,</strong></td>
<td><strong>Expected results.</strong></td>
</tr>
<tr>
<td><strong>Verify that the user enters a password 6 characters in length or greater.</strong></td>
<td><strong>Enter a password greater than 5 characters in length,</strong></td>
<td><strong>Enable the “Continue” button. Enable the “Confirm Password” input,</strong></td>
<td><strong>Expected results.</strong></td>
</tr>
<tr>
<td><strong>Verify that the user enters a matching password.</strong></td>
<td><strong>Enter a password, in the “Confirm Password” input, that does not match the password in the “Password” input.</strong></td>
<td><strong>Display a message below the confirm password input stating the error. Disable the “Continue” button,</strong></td>
<td><strong>Expected results.</strong></td>
</tr>
<tr>
<td><strong>Verify that the user enters a matching password.</strong></td>
<td><strong>Enter a password, in the “Confirm Password” input, that matches the password in the “Password” input.</strong></td>
<td><strong>Enable the “Continue” button,</strong></td>
<td><strong>Expected results.</strong></td>
</tr>
</tbody>
</table>