ABSTRACT

GoPro is the largest manufacturer of the world’s most versatile cameras, dominating with almost 50% of the market share (Thomas, 2015). Their commitment to innovation, specifically in the area of cameras and drones, has led them to exceed sales of 5 million products annually (Lango, 2017). One of the biggest challenges faced when using cameras and camcorders is determining how the device should be configured to specific activities, climates, etc. GoPro currently has a mobile application that allows users to post photos and videos from their phone to major social media platforms such as Facebook and Twitter. The application also allows users to view categories of photos and a daily “Photo of the Day” which will redirect users to their website. Although GoPro has an existing application in place, it is highly impractical and provides no information regarding the settings required to take a particular photo. The main objective of this project is to design a comprehensive social networking application that will enable GoPro users to utilize the full potential of their cameras allowing them to take the best pictures and videos possible while staying connected with the community.

This report will explain the design and implementation of the GoConnect social networking application as well as the impact that this application can have on a GoPro user.
ACKNOWLEDGMENTS

I would like to express my gratitude to my friends who provided me with valuable concepts and technologies that could be used and especially to my supervisor, Professor Tony White, whose guidance and assistance contributed to the overall design of the project.
# TABLE OF CONTENTS

1. Introduction........................................................................................................6
   1.1 Overview........................................................................................................6
   1.2 Functional Requirements............................................................................7
   1.3 Non-functional Requirements.................................................................7
   1.4 Goals...........................................................................................................8
   1.5 Objectives....................................................................................................9
   1.6 Report Outline.............................................................................................9

2. Background.........................................................................................................10
   2.1 Integrated Development Environments..................................................10
   2.2 Apache Tomcat..........................................................................................11
   2.3 JavaScript Object Notation.......................................................................11
   2.4 Jersey RESTful Service.............................................................................12
   2.5 Hibernate..................................................................................................12

3. Approach............................................................................................................13
   3.1 Assumptions...............................................................................................13
   3.2 Design of Project......................................................................................14
   3.3 Design of User Interface..........................................................................16
   3.4 Interaction Models.....................................................................................22
      3.4.1 Media Upload Model.........................................................................22
      3.4.2 Follower Model................................................................................23
      3.4.3 Account Privacy Model.....................................................................24
   3.5 Implementation..........................................................................................25
      3.5.1 Private Information............................................................................25
      3.5.2 Uploading and Retrieving Media......................................................26
      3.5.3 Cards..................................................................................................28
      3.5.4 Client/Server Communication.........................................................29
   3.6 Problems Solved.........................................................................................30

4. Conclusion..........................................................................................................33
   4.1 Future Work...............................................................................................33

5. References..........................................................................................................36

   Appendix A – Acronyms..................................................................................40
LIST OF FIGURES

1. GoPro Application.................................................................................................................. 14
2. UI vs UX.................................................................................................................................. 17
3. GoConnect Hierarchical Navigation...................................................................................... 18
4. Profile Interface Comparison................................................................................................. 20
5. GoConnect Tab Bar Navigation............................................................................................ 21
6. Mobile Device Comfort Zones.............................................................................................. 22
7. Media Upload Model.............................................................................................................. 23
8. Follower Model...................................................................................................................... 24
9. Account Privacy Model.......................................................................................................... 25
10. List vs Card Views............................................................................................................... 29
11. ORM Model.......................................................................................................................... 30
LIST OF TABLES

Table 1 – Functional Requirements.................................................................7
Table 2 – Non-functional Requirements.......................................................7
Table 3 – Breakdown of Posting Media.......................................................27
1. Introduction

1.1 Overview

Conventionally, GoPro users were only able to find suggestions or recommendations on what settings should be used for their GoPro camera by searching the Internet or YouTube for configuration videos. Although, these methods have the potential to be quite helpful, it can require reading through lengthy blog posts or a sitting through extensive videos, when most users are only searching for a small fraction of the information they will be exposed to. This can be a tedious process and there are still numerous elements that must be taken into account when deciding what settings should be used with your GoPro such as the climate and activity. Having a central solution that captures all of the aspects associated with a particular photo or video would allow GoPro users to save time researching and configuring their devices.

GoConnect is a social networking application designed for GoPro users with an iOS device. It provides the GoPro community with a method to interact with others by sharing and commenting on photos, as well as following profiles that they are interested in. GoConnect does not require any third-party applications to transfer or modify images from a GoPro device. GoPro already has the capability for users to automatically save their photos and videos to their mobile devices through a Bluetooth connection. This allows for a seamless integration between GoPro and the GoConnect social networking application.
GoConnect will decrease the amount of time required for a user to configure their device, as well as improve the quality of photos and enhance the overall experience of using a GoPro by viewing photos and videos shared by other application users.

1.2 Functional Requirements

Table 1 – Functional Requirements

<table>
<thead>
<tr>
<th>F-01</th>
<th>Users must be able to select a photo or video from the iPhone photo library</th>
</tr>
</thead>
<tbody>
<tr>
<td>F-02</td>
<td>Users must be able to upload a photo or video</td>
</tr>
<tr>
<td>F-03</td>
<td>Users must be able to upload a profile picture</td>
</tr>
<tr>
<td>F-04</td>
<td>Users must be able to comment on posts</td>
</tr>
<tr>
<td>F-05</td>
<td>Users must be able to define their profile preferences</td>
</tr>
<tr>
<td>F-06</td>
<td>Users must be able to follow/unfollow users and accounts</td>
</tr>
<tr>
<td>F-07</td>
<td>Users must be able to like posts</td>
</tr>
<tr>
<td>F-08</td>
<td>Users must be able to search for photos and videos</td>
</tr>
<tr>
<td>F-09</td>
<td>Users must complete all fields when posting media</td>
</tr>
<tr>
<td>F-10</td>
<td>Users must be able to refresh screens to update the content displayed</td>
</tr>
</tbody>
</table>

1.3 Non-functional Requirements

Table 2 – Non-functional Requirements

<table>
<thead>
<tr>
<th>NF-01</th>
<th>Usability: searching for photos and videos in the application should have a similar approach to GoPro’s playlists</th>
</tr>
</thead>
<tbody>
<tr>
<td>NF-02</td>
<td>Usability: the application should make use of a tab bar to switch screens wherever possible</td>
</tr>
<tr>
<td>NF-03</td>
<td>Reliability: if an error occurs, a descriptive error message and possible solution will be displayed to the user</td>
</tr>
<tr>
<td>NF-04</td>
<td>Reliability: if the server crashes or a user deletes uploaded media from their iOS device, the media will continue to exist in external storage</td>
</tr>
<tr>
<td>NF-05</td>
<td>Supportability: the application appearance should be practical on most iOS device with slight modifications</td>
</tr>
<tr>
<td>NF-06</td>
<td>Implementation: the application should be written in Objective-C</td>
</tr>
<tr>
<td>NF-07</td>
<td>Implementation: the GUI for the application should be designed using Xcode</td>
</tr>
<tr>
<td>NF-08</td>
<td>Interface: the application must be able to upload and download .PNG and .MOV files</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>NF-09</td>
<td>Interface: unauthorized users should not have access to a user’s profile</td>
</tr>
<tr>
<td>NF-10</td>
<td>Interface: the application should present a cleared screen every time a user decides to upload a photo or video</td>
</tr>
<tr>
<td>NF-11</td>
<td>Interface: the application should allow users to autofill text fields based on preset information</td>
</tr>
</tbody>
</table>

### 1.4 Goals

The goal of this project is to create a social networking application that will allow for GoPro users to quickly and effortlessly discover desired camera settings for their device. This application should be simple and easy to navigate, so that configuration times for a device are reduced. Allowing a user to view content with its settings readily visible will ensure collecting information for a device will be enjoyable and not cumbersome. The ambition of GoConnect is to replace the existing GoPro application since GoConnect incorporates all the features of its precursor and more. Thus, GoConnect was designed to have a similar feel to the GoPro website and mobile application, to ease users into a smoother conversion.
1.5 Objectives

The best way to achieve these goals is to create a social networking application that allows the user to feel connected with the camera as opposed to only the picture. The users are able to configure their devices more quickly using the GoConnect application in the following ways:

1. The application should display all aspects pertaining to a specific photo or video within a post
2. The application should allow users to search for photos and videos based on the categories that GoPro currently provides
3. The application should allow users to follow accounts that they are interested in
4. The application should display the most recent posts by accounts that you follow to reduce the number of steps needed to find device settings

1.6 Report Outline

The remainder of this report is organized as follows: Section 2 defines the software and technologies used to complete the GoConnect social networking application. Section 3 outlines the details of the design and implementation of the project. Lastly, Section 4 concludes the report with future work that can be accomplished to increase the practicality of the project.
2. Background

This section provides a fundamental understanding of the terminologies and technologies used in the design and implementation of the GoConnect social networking application.

2.1 Integrated Development Environments

Integrated development environments (IDE) allow for programmers to write, test and debug from within a single program and contain many useful tools such as automatic code generation and auto completion (Keeley, Lee, & Albright, 2015). This project made use of the following IDEs:

- Eclipse
  
  Eclipse is the most widely used, open-sourced Java IDE. It supports a vast number of plugins which developers can add to their programs which include features such as utilizing a group of functions created by other developers, supporting new tools in the development environment, etc. (Point, 2017).

- Xcode
  
  Xcode is an IDE that is specifically designed for the development of Apple software. Interface Builder, another Apple application, is integrated within Xcode to allow for developers to construct graphical user interfaces (GUI) for their applications.
• MySQL Workbench

MySQL Workbench is a tool that allows a developer to visualize the architecture of the database they are designing. It contains many features including server configuration, SQL creation and generation, database performance metrics and enhanced entity-relationship (EER) diagram generation (MySQL, 2017).

2.2 Apache Tomcat

Apache Tomcat is an open source web server which allows developers to run Java Servlets and JavaServer Pages (JSP). A Tomcat server can be deployed from an IDE such as Eclipse allowing programmers to quickly create and interact with a Hypertext Transfer Protocol (HTTP) server (Apache Tomcat, 2017).

2.3 JavaScript Object Notation

JavaScript Object Notation (JSON) is a data interchange format that is lightweight, easy to read and can be used for transferring data to web applications. JSON is composed of two data structures: objects and arrays. Objects are composed of a series of key/value pairs contained within a set of curly brackets. Arrays are collections of values that can store numerous formats of data (Squarespace, 2017).
2.4 Jersey RESTful Service

Representational State Transfer (REST) is an architectural style which outlines principles that should be followed when creating Web services. RESTful services make use of the main HTTP methods: Get, Post, Put and Delete. Jersey allow developers to build RESTful web services with Java (IBM, 2015).

2.5 Hibernate

Hibernate is a lightweight, open source object-relational mapping framework (ORM). It enables developers to map Java classes to their relational database, streamlining the creation, manipulation and access of data (Java, 2017).
3. Approach

3.1 Assumptions

Several assumptions were made prior to the development of the GoConnect social networking application. These assumptions allowed the developer to focus solely on the implementation and design of the GoConnect without worrying about external variables of the application.

First, the GoConnect social networking application is only offered for Apple devices. The application was designed using Xcode which is used exclusively for the development of iOS devices. In addition to being iOS dependent, GoConnect is optimized for the iPhone 5s as the developer had access to a physical device allowing for synchronized testing.

Second, GoConnect is not responsible for the transfer of images between a GoPro device and a user’s Apple iPhone. GoConnect anticipates that users of the application will pair their GoPro device to their iPhone via Bluetooth when capturing photos and videos. This will allow for media to become synced automatically between the two devices.

Finally, it is assumed that the server for the GoConnect social networking application will be configured and executed prior to the use of the application. All tables and columns of the GoConnect database schema will be generated within MySQL Workbench allowing all functionality within the application to run smoothly.
3.2 Design of Project

To understand what the GoConnect social networking platform needed to achieve, it was important to identify what GoPro was trying to accomplish with their application. Initially, the intention of the GoPro application was to serve as a wireless controller for a GoPro device. Users would have the ability to configure the settings on their paired GoPro device and capture photos and videos. Next, they expanded into editing, allowing users to modify, enhance and post their media to applications such as Facebook and Twitter. Lastly, the application allowed users to view posts by other users on the GoPro channel. In an attempt to direct more traffic to their website, the application simply redirects users to the GoPro website, providing no actual functionality on the application.

Figure 1. This image illustrates the home screen for the GoPro application (Uptodown, 2016).
Before the development of the GoConnect application, it was also important to understand the features and functionality of major social media sites. Being a social networking application, the following social aspects were taken into consideration while designing the project:

- Handling public and private information
- Internal messaging system
- Profile pages
- User’s social following
- Ability to ‘like’ posts

While social media sites such as Facebook and Instagram have amazing features, some of which were implemented in the design of GoConnect, they also contain many complicated and unnecessary features that would take away from the simplicity and attractiveness of the project. For example, Instagram allows users to edit and transform their pictures before they are posted. Although this feature could have been added very easily, filters and retouching photos can give users a false impression about the quality of the image. It is important that unedited content is posted to GoConnect as it provides users with a better understanding of just how important the settings are and how they affect the quality of the shot. The GoPro application made it appear as though there was more functionality then what was actually being offered, as options such as “Photo of the Day” and “GoPro Channel” were simply links that would open the GoPro website in another browser.
The objective of GoConnect was to create an application that was highly functional and easy to use. To use GoConnect, a user simply needs to create an account to get started. Unlike other social media sites, who bombard new users with questions in order to assist them with tailoring user specific advertisements, GoConnect requires no additional information. This allows users to instantly begin using the app to search for photos and videos they are interested in. Similarly, GoConnect will not ask you to import contacts or friends from other platforms. This ensure that users will have the same opportunity at generating a social gathering based on the quality of their content rather than their personality. Finally, users have the option to be as involved with the application as they wish. If they decide on posting media, they can customize their profile to identify their typical or preferred setup allowing them to post pictures with ease.

3.3 Design of User Interface

GoConnect is an application designed at educating and assisting users about the GoPro settings users can configure to get the best possible shot. There are many features to the application other than posting images, browsing images and a messaging system. These features needed to be organized within the application which meant that the developer of the project needed to find a relationship between the user interface (UI) and the user experience (UX). UI refers to the layout and overall appearance of the product, whereas UX involves how users interact with the product (Lanoue, 2016). The following image distinguishes between the two terms.
Figure 2. This image depicts the differences between user experience and user interaction (Burges, 2016).

The UI was designed while adhering to Apple’s iOS Human Interface Guidelines. Apple provides an extensive list of standards and suggestions that should be followed to create an attractive interface and provide an engaging experience for users (Apple Inc., 2017). I also chose to follow the agile development pattern by completing tasks in short iterations while following a timeframe to complete features and functions. The first iteration of the UI was designed to test a few of the basic functions of the application such as F-05 and F-06. I initially decided to place all effort on the UI and neglect the UX as it was essential that key features of the application were in place. First, view controllers, instances of a GUI within the application, were created that served only a single purpose. For example, view controllers were created for views such as the login, comment, profile page, etc. The benefit of using this approach is the developer is able isolate each screen
within the application allowing for quicker and more accurate testing since Xcode allows the developer to specify a view controller to display on application launch. This means that each page can be tested independently from one another to ensure that they are functioning as intended.

Figure 3. Initial design of GoConnect using Apple’s Hierarchical Navigation structure (Tzogas, 2017).

After features were functioning properly, navigation between screens needed to be configured. This is where some UX planning came into consideration as it was important that the transition between screens in the application was smooth and logical. I followed Apple’s Hierarchical Navigation structure, where each decision made on a screen, via button press, determined the next screen to be displayed.
Although this was a simple solution, it was very impractical as it depended on a user following specific paths within the application in order to reach a particular screen.

The objective of the second iteration for the user interface was to organize and simplify the screens of the application to improve the user’s experience. The main reason that applications are downloaded are because they assist users in solving a particular problem. Having an application that is not overly complicated will more likely encourage them to download and reuse your application (Chapman, 2013). Apple recommends minimizing the use of modality so that your application can remain simple. It is important that people remain concentrated on a specific task in the application and not become distracted by other features or wording. Simplifying the application involved tasks such as removing features from screens, rewording titles to make them more appropriate at describing their purpose and replacing instructions and text with images or buttons to reduce the number of elements on the screen. For example, the settings button originally contained the word “Settings”, indicating what a user should expect to see if it were pressed. Replacing the text with a universal image such as the gear icon, improved the arrangement of features within the screen without taking away from its functionality.
Figure 4. This image compares the user interfaces for the profile screen between the first and final iteration of the project (Tzogas, 2017).

In the final iteration of the user interface, displayed in Figure 4, I opted to switch from Apple’s Hierarchical Navigation to tab bar navigation. Many major organizations such as banks and social media companies have adopted this design making it a standard practice.
Figure 5. This image illustrates the final iteration of the user which includes tab bar navigation, allowing users to quickly access the most important features within GoConnect (Tzogas, 2017).

Tab bar applications play an important role in enhancing user experience as the layout of the application presents all of the important screens and allows for a seamless transition between features. Another aspect regarding user experience, is the physical comfort level of a user when interacting with an application. Research has indicated that almost 50% of people rely on their thumb alone to interact with applications on their phone (Babich, March 20, 2016). Tab bar applications allow easy access to all features providing the highest level of comfortability and can be used regardless of what hand the phone is in (Sharavara, 2016).
3.4 Interaction Models

3.4.1 Media Upload Model

GoConnect revolves around users uploading photos and videos to the application for others to browse. Unlike most social media platforms, where uploads do not require users to enter information such as captions, GoConnect does. GoConnect requires all information fields, such as camera version and resolution to be completed before the upload is accepted. The information is essential to the photo and is one of the key aspects of the post that users will be looking for. It provides valuable information pertaining to the setup and configuration of the camera allowing users to gain a better understanding of what settings they should be using. To simplify the data entry when uploading media, GoConnect enables users to set
their preferred camera preferences in their settings. With the simple click of a button, all preferred information will be uploaded to the post and the users will only need to select a category appropriate for post and a caption.

![Diagram](image)

Figure 7. This image outlines the process users must follow to upload media to the GoConnect application (Tzogas, 2017).

3.4.2 Follower Model

The ability to follow profiles is a very important feature in many social media applications. Studies have shown that the number of followers amassed by individuals can directly affect their self-worth and social anxiety (Edwards, 2017). Following users is a very simple task and can be accomplished many different ways. Users have the ability to search through multiple categories of photos on the application. The posts in these categories come from a variety of different users with public accounts, allowing all users to view their content. Pressing on the username in a post will redirect the user to that account’s profile where they will have the ability to view more content or follow them. Users can view a list of accounts that follow them on their profile. Similarly, by selecting an account in the
list, the user will be redirected to the account’s profile where they have the ability to follow them.

![Diagram](image)

Figure 8. This image identifies the options that users have to follow other users (Tzogas, 2017).

3.4.3 Account Privacy Model

Anonymity and privacy is a growing concern and a recent study conducted by Ponemon Institute has proven that privacy sensitivity amongst users has increased from 55% to 61% since 2010 (Clay, 2015). Similar to many applications, GoConnect allows users to set their profiles as either public or private. Public profiles grant all users the ability to view content on the account’s profile. Private profiles provide users with a sense of security as it prevents accounts without consent from viewing their profiles. Content on private profiles can only be viewed by users in two ways. First, if the user attempting to view the account is followed by the account, then access to the content will be made visible. Second, if two users follow each other, then access to the account will be made visible.
3.5 Implementation

3.5.1 Private Information

The extent to which private information is divulged is a constant debate when considering social media applications. The amount of information required to register an account or the visibility of personal data is a constant worry for users. It is important that users feel safe and comfortable when using an application. The first way GoConnect succeeds at meeting these requirements is the minimal amount of information required to create an account. GoConnect accounts only require a username and password; no email, phone number or security questions are required. Likewise, profile settings only contain a user's camera preferences and profile visibility which is defaulted to private, hiding all posts from users. Many social media platforms, such as Facebook and Twitter, have a complicated following system where there are different rules about when a user can view another user's profile or posts. Twitter for example, has an option where a user can approve follow requests to allow others to gain access to viewing their profile. GoConnect has a
similar concept, if two users follow each other, they are able to see each other’s profile and posts regardless of their privacy settings.

3.5.2 Uploading and Retrieving Media

The premise of GoConnect relies on users posting photos and videos captured on their GoPro devices to the application. Uploading a post can be broken down into three parts: selecting media, entering information and posting. However, as I began to design the UI, it became apparent that there were many decisions that a user could make and are identified in Table 3. For example, when a user decides to upload a video it is important that they can watch the video first. However, images and videos cannot be loaded into the same view as they have different properties, requiring a minimum of two views. Thus, photos would be loaded into one view and videos into the other. For example, if a user decided to upload a video, then changed their mind and uploaded a photo, both selections were displayed to the user. To fix this problem, I enabled a refresh button, which resets the screen to its original state, removing any loaded media and clearing all the text fields. Future iterations of this applications could lead to a better way to solve this particular problem.
<table>
<thead>
<tr>
<th>Action</th>
<th>Requirement</th>
<th>User Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>Select Photo</td>
<td>F-01</td>
<td>Load image into image view</td>
</tr>
<tr>
<td>Select Video</td>
<td>F-01</td>
<td>Load video into view</td>
</tr>
<tr>
<td>First Post Ever</td>
<td>F-09</td>
<td>Enter information into the required text fields</td>
</tr>
<tr>
<td>Multiple Posts Made</td>
<td>NF-11</td>
<td>Select a user preferences button</td>
</tr>
<tr>
<td>Upload Different Media</td>
<td>NF-10</td>
<td>Refresh or clear button</td>
</tr>
<tr>
<td>Upload</td>
<td>F-03</td>
<td>Post button</td>
</tr>
</tbody>
</table>

Table 3 – Breakdown of Posting Media

Considering the fact that people hold onto their images indefinitely, a storage system had to be considered as storing these media files within a database consumes a lot of memory and doing such would increase the size and time it takes to complete a query within the system. Currently, the media is stored in an external folder, created by the application. All of the images and videos obey a naming convention and therefore, guarantee that they will never overwrite one another. In the future, it might be beneficial to use third-party software or an external server, where users can have individual folders for their photos and videos. This could increase the safety of the posted media, by allowing moderators to quickly delete an entire account from the server instead of a collection of posts.

Photos and videos can be selected from your iOS device and uploaded to the application. All media is saved to the database using the following naming convention, the user’s account name and caption. Since every user must have a unique name, it ensures that duplicate photos and videos will not be created, potentially overwriting others. I decided to save videos as MOV files as it was
originally developed by Apple, (Apple Inc., 2017) and images as Portable Network Graphics file (PNG) since Xcode optimizes this file format, which is an important factor when determining the quality of a picture (Gordon, 2011).

Retrieval of photos and videos from the database are completed through queries for the name of the media file. A string indicating the exact location of the image file is converted to an NSData format, ensuring that the item is converted to a file format (Apple Inc., NSData, 2017). The image is then loaded into the appropriate UIImageView in a card. Classes from the Apple AVFoundation framework were used to load and play videos within the application (Apple Inc., AVFoundation Programming Guide, 2017). AVPlayerItem was used to convert strings containing the location of the video files, similar to what was required for images, into a readable data format. AVPlayer was used to trigger playback for the video file. Finally, AVPlayerLayer was used to modify the orientation and display of the video within the post.

3.5.3 Cards

One of the objectives of the GoConnect was to have a visually appealing and easy to navigate interface. Applications such as Facebook and Instagram do an amazing job at presenting information to users and I wanted my interface to reflect some of their best practices. Applications have moved away from presenting web pages in favour of displaying blocks of content. Cards are highly interactive rectangles that containing text and images that can be organized into multiple sections (Babich,
Android allows developers to extend their layouts to use the CardView class, which allows them to display information inside large cells. Xcode however, does not have a similar feature and to replicate it, I had to create my own class that identified all the elements and their orientation within the card. Next, I created a table view, where the cells within the view were styled with my newly created class.

Figure 10. This figure displays the difference between a list view and card view (Material Design for Android, 2017)

3.5.4 Client/Server Communication

Another important aspect of GoConnect was the notion of data being sent and received between the client and server. There are numerous methods and languages to accomplish this task but I decided to use JSON as its primary purpose is to transmit data. JSON is composed of a mapping of key and value pairs, which
can exist within objects and have potential to hold arrays of data (Squarespace, 2017). This was the perfect solution as it was versatile enough to fit all applications of GoConnect. For example, loading in the main feed of the application requires a user to follow at least one account. Thus, a simple query of the database will yield a list of accounts the user follows. An object for each post an account has made is created, with all of the relevant information that needs to be passed to the GUI being stored as key value pairs. The object is then added to an array and sent via a GET request by the client. The array is then divided into smaller arrays containing the information required for a single element of a post such as, username, date, number of likes, etc. Finally, the individual arrays are parsed and the data is loaded into its corresponding card.

Figure 11. This image illustrates the process and technologies required for Hibernate’s object relational mapping (Stackoverflow, What is Object/relational mapping(ORM) in relation to Hibernate and JDBC?, 2016).

3.6 Problems Solved

The main problem I was tasked with happened to be the very foundation of the application, saving and loading images and videos. There were many things that needed to be taken into consideration such as the location and the data type that the media would be stored as. To simplify the process, I thought it would be easiest
to upload and retrieve images directly to MySQL. Although it is not the most efficient
method and would not scale in a practical application, it would prove the concept
that media could be stored and retrieved.

After many painstaking hours of research, the preferred method of media storage
was base64, an ASCII string representing the binary data of the image (Galambos,
2012). The benefit of this method is that it guarantees that all strings will be unique
so you do not need to worry about uploads overwriting each other. Google provides
some open source code that contains methods to transform data into base64
strings. Converting the image and storing the string in the database was simple,
however, decoding the base64 strings did not work and I was not able to retrieve
the image information from the database. The second method that I tried was to
save the image directly from the iOS device to an external location. Images on an
iOS are given a name automatically in increasing numerical order. This meant that I
was able to obtain the image name and save it to the database. However, the idea
revolved around a user never removing the photo from their library, which is highly
unlikely and not scalable. Finally, using knowledge from the previous attempts I
found a method that would provide a scalable and functional solution for uploading
posts. There are two attributes that make a post unique: the user and the post
caption. I decided to create a string combining the two attributes associated with a
post ensuring that there would be no redundancy between posts. Next, I discovered
NSFileManager, an object that allows for the creation and modification of files. This
meant that files could exist externally allowing GoConnect to be scalable. Finally,
the media could be saved to the database with the filename and retrieved by finding
the file with the corresponding filename.

Another major problem I had was loading the comments for posts into a new view.
Prior to this project, I had only used table views for very basic tasks and so I was
not aware about how complex they could be made. In terms of comments, every
post has a button that allows a user to view the comments for that post. The button
needs to know what post it was triggered from and to retrieve the comments from
the database. Determining what cell was interacted with was trivial and I was able to
trace the information from the database, however, the new view controller was
unable to display the content. It turned out the view controller was being called
twice, first with incorrect data and the second with correct data, however, the
second call was not able to override the first. The comments were not being
displayed in the correct order due to how the information was being sent via the
performWithSegue method which is responsible for sending the data to a new view
controller. However, the viewDidLoad method was initializing the view controller and
was overwriting all data from the previous view. I found a new method
prepareForSegue which notifies the new view controller about the data being
transferred, allowing the view to be loaded correctly.
4. Conclusion

Social media applications are more than just a means of staying connected, they allow companies to promote and introduce products to provide users with an additional method of obtaining information, such as instructional videos (Rendler-Kaplan, 2017). GoConnect is a social networking application that allows users to quickly and easily browse photos and videos, allowing users to customize their GoPro devices according to their personal preference. The paper first begins by outlining the struggles of the current GoPro application and how it diverged from the convenience that users were hoping for. GoConnect, a solution to the problem, was introduced, addressing the key features and functionality that the GoPro application was lacking itself. Section 2 clarifies the background information and terminology that is used throughout the paper. The implementation and design for the user interface, as well as many important features of the application are described in detail in Section 3. The following section outlines the future of application and the enhancements that can be made to transform GoConnect into a social media giant.

4.1 Future Work

While GoConnect incorporates all the functional and non-functional requirements, outlined in Table 1 and Table 2 respectively, there are enhancements that can be made to increase the usability and practicality of the application.
First, GoConnect could improve the way that users are able to search for media that they are interested in. The developer modeled his approach based on the current method that GoPro provides, which is allowing users to search through a predefined set of categories for photos and pictures. Although this is great for activities such as sports, it does not capture topics that users might be interested in. It would be beneficial to replace the current page of categories with a search bar that would allow users to find profiles, categories, etc. much more easily. This would allow the user to have the freedom to search for whatever content they want as well as more specific content.

Second, it would be useful if users were able to specify the types of accessories they used when taking a photo or video. Camera settings are one aspect of taking great photos, but so are important attachments such as a camera filter and stabilizers. Having an option for a user to select GoPro accessories and add them to their post would allow great advertisement and possibly generate potential sales. A user would be able to click on a specific accessory attached to the post and it would redirect the user to GoPro’s website where they could purchase that item.

Finally, after redesigning the search page and implementing an option to include accessories used in the taking of a picture, it would be helpful to have a page that recommends media and products to users. Currently, Instagram has implemented something called the “Explore Tab” which displays a selection of posts to users based on things they like or people they follow. It would be beneficial to split this tab
into two sections: photos and videos that you might be interested in and GoPro accessories that you might be interested in. This would require the implementation of a recommender system.
5. References


Appendix A - Acronyms

EER  Enhanced Entity-Relationship
GUI  Graphical User Interface
HTTP Hypertext Transfer Protocol
IDE  Integrated Development Environment
JSON JavaScript Object Notation
JSP  JavaServer Pages
ORM  Object-Relational Mapping
PNG  Portable Network Graphics
REST Representational State Transfer
UI   User Interface
UX   User Experience