

NQ

by

William Page, Argina Horn, and Brandon Ghearing

Submitted to
The Faculty of the School of Information Technology
in Partial Fulfillment of the Requirements for
the Degree of Bachelor of Science
in Information Technology

© Copyright 2018, William Page, Argina Horn, and Brandon Ghearing

The author grants to the School of Information Technology permission to
reproduce and distribute copies of this document in whole or in part.

William Page

Date

Argina Horn

Date

Brandon Ghearing

Date

Abdou Fall

Date

University of Cincinnati
College of Education, Criminal Justice, and Human Services



November 27, 2018

NQ



Prepared By:

William Page, Project Manager
Argina Horn, Security Analyst
Brandon Ghearing, Software Developer

Students of University of Cincinnati
College of Education, Criminal Justice and
Human Services
School of Information Technology
cech.uc.edu

April 2019



TABLE OF CONTENTS

NQ	i
TABLES LIST	iv
FIGURES LIST	v
ABBREVIATIONS & ACRONYMS	vi
ABSTRACT	1
1. PROBLEM STATEMENT	2
1.1 Introduction:	2
1.2 Problem:.....	2
1.3 Solution:	3
1.4 User Profile:.....	4
1.6 Use Case Diagram:.....	7
2. PROJECT MANAGEMENT	8
2.1 Budget	8
2.1.1 Office Startup Costs.....	8
2.1.2 Server Room Expenses	8
2.1.3 Software Development Expenses	10
2.1.4 Staff Expenses	10
2.1.5 Investor Returns & Funds.....	11
2.2 Deliverables.....	13
2.2.1 Fall Semester	13
2.2.2 Spring Semester	14
3. PROJECT SCHEDULE: GANTT CHART & WBS (WORK BREAKDOWN STRUCTURE) 16	
3.1 Gantt Chart.....	16
3.2 Work Breakdown Structure & Category Breakdown	18
3.2.1 Design.....	18
3.2.2 Security.....	18
3.2.3 Hardware.....	19
4. TECHNICAL ELEMENTS	22



4.1	Technical Approach	22
4.1.1	Technical Architecture / Discussion	24
4.1.2	Explanation of Technical Architecture	25
4.2	Visuals	26
4.3	User Interface.....	27
4.3.1	Button Placement.....	27
4.3.2	Easy Swipes	28
4.3.3	Camera Design	29
4.3.4	Vertical / Horizontal Shift.....	29
4.3.5	Problems Encountered.....	29
5.	PROJECT TESTING	32
5.1	Overview	32
5.2	Objective	32
5.3	System Testing	32
5.4	Testing Procedures.....	33
5.5	Testing Scenarios.....	33
5.5.1	Launch Test	33
5.5.2	Stability Test.....	34
5.5.3	User Interface Test	35
5.5.4	Functionality Test	37
5.5.5	Quick Test Scenarios	38
5.5	Pass / Fail Conditions	39
5.6	Testing Timeline	39
5.6.1	Developer Testing Timeline.....	40
5.6.2	Beta Testing Timeline	41
5.7	Lessons Learned	42
5.7.1	Login Process	42
5.7.2	Storing Videos	43
5.7.3	UI Editing	43
5.8	Future Recommendations / Recommendations for Improvement	43
5.8.1	Analysis if we had to do project again	44
5.8.2	What if we had more time?	45



5.8.3 Suggestions from Others.....	45
5.8.4 Future Project Plans	46
6. CONCLUSION	47
6.1 Fall Semester 2018:	47
6.2 Spring Semester 2019:	47
6.2.1 Lessons Learned for Spring Semester / Fall Semester	47
6.2.1.1 Fall Semester Work	47
6.2.1.2 Spring Semester Work.....	48
6.2.2 Abilities/Skills Developed & Enhanced.....	49
6.2.3 Progress Since Fall Semester	50
6.2.4 Lessons Learned from Expo.....	50
7. REFERENCES.....	53
APPENDIX A.....	54
NQ Logo Artwork	54
Poster	55

TABLES LIST

Table 1: User Profile.....	6
Table 2: Office Startup Costs.....	8
Table 3: Server Room Expenses	9
Table 4: Software Development Expenses	10
Table 5: Staff Expenses	11
Table 6: Investor Returns.....	12
Table 7: Gantt Chart.....	17
Table 8: Launch Testing.....	34
Table 9: Stability Testing	35
Table 10: Emulation Interface Testing	36
Table 11: Samsung Galaxy S9 Testing	37
Table 12: Functionality Testing	38
Table 13:Functional Testing Cont.	38
Table 14: UI Testing Cont.	39
Table 15: Compatibility Testing Cont.....	39
Table 16: Staff Testing.....	40



November 27, 2018

Table 17: Beta Testing.....	41
Table 18: Fall Project Resources	49
Table 19: Spring Project Resources (Original).....	50
Table 20: Spring Project Resources (Updated)	50

FIGURES LIST

Figure 1: Use Case Diagram	7
Figure 2: Estimated Profit Margin.....	12
Figure 3: Security Authentication	19
Figure 4: Work Breakdown Structure	21
Figure 5: Technical Diagram.....	24
Figure 6: Notifications.....	26
Figure 7: Button Placement	28
Figure 8: Design Vertical	30
Figure 9: Design Horizontal	30
Figure 10: Error Message	42
Figure 11: Expo Poster	55



November 27, 2018

ABBREVIATIONS & ACRONYMS

Videographers	A person who makes video films
SATA	Serial Advanced Technology Attachment
HDD	Hard Disk Drive
SSD	Solid State Drive
SAS	Statistical Analysis System



November 27, 2018

ABSTRACT

In 2018, disruptiveadvertising.com reported that 3.03 billion people are using some sort of social media. The most popular apps provide something for everyone, but very few actively connect video makers with their audience. NQ is an app designed to connect video makers with knowledgeable critics within their niche. Users will no longer need to worry about using the unhelpful thumbs up, thumbs down rating system to determine what viewers think of their work. All viewers comments are monitored by NQ to ensure high quality feedback. Every feature offered in the app will help artists create their best content every time.



November 27, 2018

1. PROBLEM STATEMENT

1.1 Introduction:

Social media has become the one of the most common way to set apart this generation from the ones before it. There is next to nothing that's can't be attained by looking to social media. While it is an excellent way to attain news, it also is a way to find endless entertainment. Much of this entertainment comes from videos shared by creatives all over the world. However, while there are a multitude of platforms to share and watch, there is not one that actively works to bring the viewer and creator together to create quality content. NQ addresses the challenge of building a video sharing social media app that works for viewers and creators. A feedback rating system designed to help video makers create better content will ultimately be a service to the user. With fewer guesses at how make their content better, video makers are in partnership with viewers in creating a more customizable viewing experience. Combined with a clean and modern UI, NQ is designed to usher in a new era of social media sharing.

1.2 Problem:

An article written for disruptiveadvertising.com reported that "close to half of the world's population (3.03 billion people) are on some type of social media." What this means is, the world is always watching something. However, where there is a lot of websites to promote user-content, there isn't a platform that helps videographers make better videos.

Newgrounds.com for decades has made it so video makers could post their videos while receiving criticism successfully. However, their app for the iOS and Android OS isn't as



November 27, 2018

successful. The application has been known to crash, and not load videos properly. This can hurt aspiring videographers because they don't have a stage where they can not only share their content, but also receive helpful feedback that will inspire them to make the best videos that they can.

YouTube is known as one of the most successful franchises of social media. The problem with YouTube is that YouTube makes it very difficult for a new videographer to share videos for others to see. As backlinko.com (Dean) demonstrates, for a videographer to showcase videos, they need to make proper names, have high video times (around 15 minutes), get a lot of likes, and have a strong fanbase. Since people watching videos is what gets the top spots, videographers are already at a disadvantage.

1.3 Solution:

Some may argue that this is what YouTube does. How then, is NQ, any different? NQ brings critics and videographers together on the same platform as partners. With NQ, you can upload content based on the category you choose and await your ratings and reviews from critics from that category. The higher your video ratings, the more chances you receive to have your video featured at the top of your category. Ratings also translate to points that you can use to improve the quality of your video. Use points to buy features for editing your videos, increase video length, and other fun effects that will make your video more appealing. So, what's the catch?



November 27, 2018

Our idea is to combine the ability to share your own personal videos and receive criticism on them. Some of these videos can also be between co-workers on project specifications as well. You can freely rate these videos and go up in ranks. The more you go up in ranks, the more your votes count toward the popularity of the video. An application like this will continue to use mobile devices with media.

When you rank up as a viewer, your vote counts for more. When you rank up as a video maker, your videos get more premier spots for all to see.

If you continue to get a low rating on one video where after 100 people seen it and it is scored lower than a 1.15, the video will be removed from the app.

You can also send messages to friends and make "rating groups" where you can get fame or respect for however you handle rating videos.

1.4 User Profile:

Social media is designed to entertain and share information with people all over the world. With such a high demand, we have found a hole in the video media that needs to be filled. NQ end users will be anyone who likes to watch videos on their phone and artists who want to receive honest and helpful feedback. Users will be able to create an account and select categories of videos that they want to follow. Every category will be populated with videos



November 27, 2018

tagged for those topics. End users who are using the service to upload their videos will be presented with the ability to upload videos for unlimited lengths of time, use in app design upgrades, and access to a collection of all the feedback provided by the critics. Refer to *Table 1: User Profile* below for a quick breakdown of the User Profile.

Project: NQ
Potential Users: ➤ Aspiring filmmakers, videographers, and artists ➤ Social media enthusiasts of all ages ➤ Fans of visual media
Software, Interface, and Related Experience: NQ users should be familiar with using desktop and mobile social media applications across several platforms and user interfaces.
Experience with Similar Applications: ➤ Vine ➤ NewGrounds ➤ YouTube * ➤ Vimeo ➤ Dailymotion ➤ Instagram ➤ Snapchat ➤ Facebook *YouTube uses a thumbs up, thumbs down point system; however, NQ's ratings will be based entirely on quality user feedback
Task Experience: Upon logging in for the first time, the user will be greeted with a signup/login screen with a preview of the different video's users can find on the app. After the user has created an account, they will only be presented with their feed that has been created based on who they follow and the videos they have critiqued.



Frequency of Use:

NQ does not have a specific occasion wherein it should or should not be utilized. The frequency in which a content creator uses the app will be based on how often they upload. For content viewers, the frequency may not be scalable.

Key Interface Design Requirements and Profile Suggests:

- **Straight forward, modern UI**
- **Intuitive menu navigation**
- **Clean, readable font paired with hierarchical text formatting**

Table 1: User Profile

1.6 Use Case Diagram:

NQ's Use Case Diagram goes into how we plan on designing the application towards users and videographers. *Figure 1: Use Case Diagram* displays the usability differences between viewers and videographers. Viewers have more control over viewing content and providing ratings. Videographers will have more control over their own videos and who can see them (whether it's private or public).

NQ USE CASE
DIAGRAM

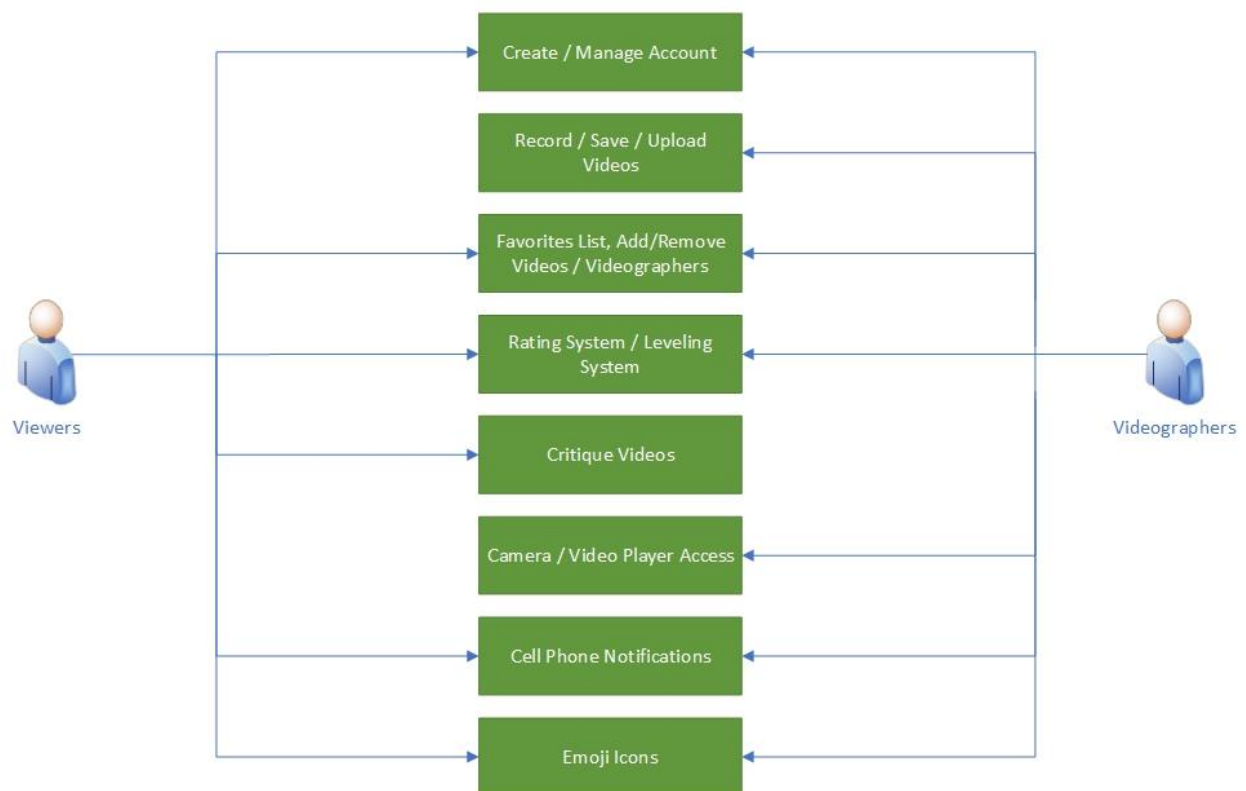


Figure 1: Use Case Diagram



2. PROJECT MANAGEMENT

2.1 Budget

NQ's budget will be broken down into several categories. These include the initial startup costs.

Afterwards, there will be a breakdown of each area in the projected office room.

2.1.1 Office Startup Costs

NQ's budget is split into several categories. The first one is listed as what our pricing is at monthly. *Table 2: Office Startup Costs* displays the minimum costs for a small office building for maintaining NQ servers. This covers basic rent and utilities expected expenditures by monthly, quarterly and annual charges.

Office Charges	Monthly Charges	Quarterly	Annual Total
Rent	\$ 3,000.00	\$ 9,000.00	\$ 36,000.00
Electric	\$ 245.71	\$ 982.84	\$ 2,948.52
Internet	\$ 84.90	\$ 339.60	\$ 1,018.80
Chairs	N/A	N/A	\$ 195.00
Desks	N/A	N/A	\$ 244.89
Grand Total	\$ 3,330.61	\$ 10,322.44	\$ 40,407.21

Table 2: Office Startup Costs

2.1.2 Server Room Expenses

The Server room charges consist of what NQ projects as the starting costs. After the initial price is paid, the maintenance required on each device over an annual total is checked and



November 27, 2018

maintained to insure optimal efficiency. For several years to come, the current maintenance will be needed each year without the addition of new equipment (This will be discussed later in a further budget chart). In *Table 3: Server Room Expenses*, shows the ideal charges for a Server area.

Server Work Area	Charges	Maintenance	First Year Total
Work Computer	\$ 1,724.99	\$ 200.00	\$ 1,924.99
2 Monitors	\$ 219.99	\$ 50.00	\$ 269.99
Server Machine / Cubby	\$ 12,700.04	\$ 300.00	\$ 13,000.04
Networking	\$ 300.00	\$ 130.00	\$ 430.00
Keyboard	\$ 20.00	\$ 10.00	\$ 30.00
Mouse	\$ 25.00	\$ 10.00	\$ 35.00
Firewall	\$ 449.00	\$ 125.00	\$ 574.00
Switches / Routers	\$ 248.99	\$ 150.00	\$ 398.99
Power strip	\$ 38.99	N/A	\$ 38.99
Air Conditioning	\$ 948.50	\$ 350.00	\$ 1,298.50
Grand Total	\$ 16,675.50	\$ 1,325.00	\$ 18,000.50

Table 3: Server Room Expenses



2.1.3 Software Development Expenses

This area requires the bare minimum startup costs for starting up. Work Computers are Pentium i7 7th edition to allow the software to be updated and edited simultaneously. In *Table 4: Software Development Expenses*, the breakdown also includes product licenses startup costs for a small business.

Software Development Area	Charges	Maintenance	Annual Total
Power Strips	\$ 79.54	N/A	\$ 79.54
2 Work Computers	\$ 1,099.98	\$ 150.00	\$ 1,249.98
4 Monitors	\$ 439.96	\$ 100.00	\$ 539.96
Updates / Software	Software Salary	Software Salary	\$ -
Android Studio License	N/A	N/A	\$ -
Product License	\$ 2,500.00	\$ 250.00	\$ 2,750.00
Grand Total	\$ 4,119.48	\$ 500.00	\$ 4,619.48

Table 4: Software Development Expenses

2.1.4 Staff Expenses

The expenses shown are the original startup costs for a three-person team. This includes 2 software programmers and a server maintenance technician. The software programmers will design and update software. Their job is to develop new innovative ways to make the software easier to use. The server technician will be responsible for all the networking. This includes running all wires, minimizes damages, and maximizing connection speeds. *Table 5: Staff*



November 27, 2018

Expenses demonstrates the salary costs for the employees and an annual 2% raise after the first year.

Staff	Salary	Raises	Annual Total
Software Programmer	\$ 60,000.00	2%	\$ 61,200.00
Software Programmer	\$ 60,000.00	2%	\$ 61,200.00
Server Maintenance Tech	\$ 37,324.80	2%	\$ 38,071.30
Grand Total	\$ 157,324.80	\$ 0.06	\$ 160,471.30

Table 5: Staff Expenses

2.1.5 Investor Returns & Funds

What does the investor expect in return for funding the project? In *Table 6: Investor Returns*, shows the investor profits after a time. The numbers projected are at the low end for expected gains assuming the application doesn't take off as expected. As demonstrated in the 6 months column, NQ is expected to be losing money. After a year, NQ is expected to cut into the borrowed funds of the investor. Around the 2-year mark, the projected numbers are expected to jump based on the number of users. There will be a solid return for the investor and the company can consider expanding and adding new options.

Graphing Chart	6 Months	1 Year	2 Years
Number of Customers	25,000	100,000	600,000
Investor Borrowed Funds	\$ (223,498.49)	\$ (223,498.49)	\$ (223,498.49)
App Purchases	\$ 74,750.00	\$ 299,000.00	\$ 1,794,000.00



November 27, 2018

More Space	\$ 124,750.00	\$ 499,000.00	\$ 944,000.00
Office Expansion	\$ 81,994.76	\$ 163,989.52	\$ 327,979.04
More Employees	\$ 75,000.00	\$ 450,000.00	\$ 900,000.00
Continual Maintenance	\$ 91,044.38	\$ 182,088.76	\$ 364,177.52
Profit Margin	\$ (272,037.63)	\$ (221,576.77)	\$ 922,344.95
Grand Total + Borrowed Funds			\$ 922,344.95

Table 6: Investor Returns

In *Figure 2: Estimated Profit Margin*, the graph shows what the expected gain is going to be. It can be evaluated that the number of users will continue to raise as the app becomes more successful.



Figure 2: Estimated Profit Margin



2.2 Deliverables

In this section, we breakdown the deliverables into two different parts, Fall and Spring Semester. Fall Semester mostly covers the basic startup operations. These include researching the app, designing the UI, writing the code for the camera, making a budget list, and developing an overall direction. The Spring Semester includes testing all the bugs and adding in more features as the application becomes more stable.

2.2.1 Fall Semester

Research Topic & Problem Solving Search Relative Topics Determine Application Flexibility	6 Days	8/27/18 – 9/3/18
Analyzing & Research Determine Application Use / Target Audience Research Features / Coding Languages	16 Days	9/3/18 – 9/24/18
Determine Software & Necessary Programs Research Java Potential Research PHP Check Plugins / Security Software	12 Days	9/18/18 – 10/3/18
Proof of Concept Visual / Android Studio Determine Design Walk-through Research & Check / Edit Ideas	16 Days	9/21/18 – 10/12/18
Develop Prototype Write Code (Java / PHP) Design Application Develop Score & Rating System, Security Features,	22 Days	10/12/18 – 11/12/18



November 27, 2018

Interface Design, Animations,
Logo

Prepare for Prototype Presentation Microsoft PowerPoint Have Phone Application display ready	8 Days	11/12/18 – 11/21/18
Beta Test Prototype Test Features Crash Prevention	6 Days	11/21/18 – 11/28/18
Bug Check Prototype Test each section of the code individually Combine testing and report results	8 Days	11/21/18 – 11/30/18
Deliver Prototype Presentation	1 Day	12/3/18 – 12/3/18

2.2.2 Spring Semester

Determine Additional Features & Design Implement surveys Notifications Saving User Favorite Videos Saving Artists Emoji Icons One-Click Ez-Rate System	101 Days	12/5/18 – 4/24/19
Add-in Discussed Features Implement Surveys Notifications Saving User Favorite Videos Saving Artists Emoji Artists One-Click Ez-Rate System	40 Days	1/7/19 – 3/1/19
Beta Test Features (Round 2) Implement Surveys	11 Days	3/4/19 – 3/18/19



November 27, 2018

Notifications
Saving User Favorite Videos
Saving Artists
Emoji Artists
One-Click Ez-Rate System

Prepare for IT Expo	11 Days	4/1/19 – 4/8/19
Designs implemented		
Graphs checked		
Portfolio easy-read for viewers		

IT Expo	1 Day	4/9/19 – 4/9/19
----------------	-------	-----------------

Team Discussion / Summary	6 Days	4/22/19 – 4/29/19
Discuss how the project went		
What we could have done differently		
Personal feelings, emotions		



3. PROJECT SCHEDULE: GANTT CHART & WBS (WORK BREAKDOWN STRUCTURE)

3.1 Gantt Chart

The Gantt Chart is broken down into several timeline sections. Refer to *Table 7: Gantt Chart* below for a excel version Gantt Chart.

Task Name	Duration	Start	Finish
Project Development	173 days	Mon 8/27/18	Wed 4/24/19
1. Research Topic & Problem Solving	6 days	Mon 8/27/18	Mon 9/3/18
1.1 Search Relative Topics	6 days	Mon 8/27/18	Mon 9/3/18
1.2 Determine Application Flexibility	3 days	Wed 8/29/18	Fri 8/31/18
2. Analyzing & Research	16 days	Mon 9/3/18	Mon 9/24/18
2.1 Determine Application Use / Target Audience	3 days	Mon 9/3/18	Wed 9/5/18
2.2 Research Features	7 days	Thu 9/6/18	Fri 9/14/18
2.3 Research Coding Languages	6 days	Mon 9/17/18	Mon 9/24/18
3. Determine Software & Necessary Programs	12 days	Tue 9/18/18	Wed 10/3/18
3.1 Research Java Potential	3 days	Tue 9/18/18	Thu 9/20/18
3.2 Research PHP	4 days	Fri 9/21/18	Wed 9/26/18
3.3 Check Plugins / Security Software	5 days	Thu 9/27/18	Wed 10/3/18
4. Proof of Concept	16 days	Fri 9/21/18	Fri 10/12/18
4.1 Visual Studio / Android Studio	2 days	Fri 9/21/18	Mon 9/24/18
4.2 Determine Design	9 days	Tue 9/25/18	Fri 10/5/18
4.3 Walk-through Research & Check / Edit Ideas	5 days	Mon 10/8/18	Fri 10/12/18
5. Develop Prototype	22 days	Fri 10/12/18	Mon 11/12/18
5.1 Write Code (Java)	22 days	Fri 10/12/18	Mon 11/12/18
5.1.1 Develop Score System	22 days	Fri 10/12/18	Mon 11/12/18



5.1.2 Code Camera / Video	22 days	Fri 10/12/18	Mon 11/12/18
5.1.3 Develop Rating System	22 days	Fri 10/12/18	Mon 11/12/18
5.2 Application Design	22 days	Fri 10/12/18	Mon 11/12/18
5.2.1 Interface Design	22 days	Fri 10/12/18	Mon 11/12/18
5.2.2 Animations	22 days	Fri 10/12/18	Mon 11/12/18
5.2.3 Logo	22 days	Fri 10/12/18	Mon 11/12/18
6. Beta Test Prototype	6 days	Wed 11/21/18	Wed 11/28/18
6.1 Test Features	4 days	Wed 11/21/18	Mon 11/26/18
6.2 Crash Prevention	3 days	Mon 11/26/18	Wed 11/28/18
6.3 List & Tally all found bugs	6 days	Wed 11/21/18	Wed 11/28/18
7. Bug Check Prototype	8 days	Wed 11/21/18	Fri 11/30/18
7.1 Test each section of the code individually	5 days	Wed 11/21/18	Tue 11/27/18
7.2 Combine testing and report results	4 days	Tue 11/27/18	Fri 11/30/18
8. Determine Additional Features & Design	101 days	Wed 12/5/18	Wed 4/24/19
8.1 Implement Surveys	TBD		
8.2 Notifications	TBD		
8.3 Saving Favorite Videos	TBD		
8.4 Saving Artists	TBD		
8.5 Emoji Icons	TBD		
8.6 One-Click Ez-Rate System			
9. Add-in Discussed Features	40 days	Mon 1/7/19	Fri 3/1/19
10. Beta Test Added Features	11 days	Mon 3/4/19	Mon 3/18/19
11. Team Discussion / Summary	1 day	Mon 4/22/19	Mon 4/22/19
11.1 Discuss how the project went	1 day	Mon 4/22/19	Mon 4/22/19
11.2 What we could have done differently	1 day	Mon 4/22/19	Mon 4/22/19
11.3 Personal Feelings, Emotions	1 day	Mon 4/22/19	Mon 4/22/19

Table 7: Gantt Chart



November 27, 2018

3.2 Work Breakdown Structure & Category Breakdown

NQ's Work Breakdown Structure is broken down into four different categories. These categories are, Design, Security, Hardware, and Features.

3.2.1 Design

The Design covers the interface of the application. The idea around the design is to make the product easy to use. Measuring the interface can be difficult depending on the type of Android device being used. While relative positioning of the application works in some situations, there has been problems on smaller devices (mainly a 4.4-inch screen or less).

3.2.2 Security

The Security on the application is going to use a two-factor authentication. The user will create an account and be sent a verification number to their cell phone. The android device being used is the only one that's trusted to run an NQ account and upon login, the user will automatically be connected to NQ. *Figure 3: Security Authentication* shows a sample of what's intended security behind the two-factor authentication.

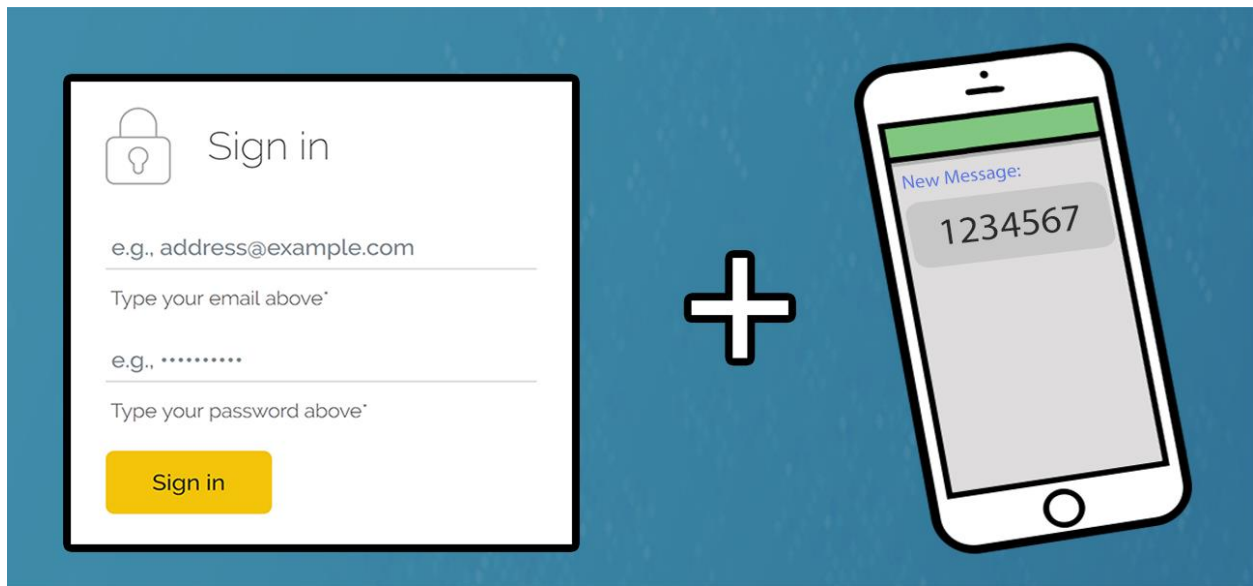


Figure 3: Security Authentication

3.2.3 Hardware

The Hardware is going to consist of a server which all Android devices can connect to. The server itself is a Dell PowerEdge T330 Tower Server. It contains:

- Intel Xeon E3-1270 v6 3.8 GHz, 8M Mache, 4C/8T, turbo (72W)
- Fresh Air Cooling
- 16GB of 2400MT/s DDR4 ECC UDIMM Ram
- Raid 10, H330/H730 for SAS/SATA (purpose for this is it allows 4 minimum HDD, allowing for extra storage)
- 12 TB 7.2K RPM SATA 12Gbps SAS HDD (Highest Available for space)
- 120GB SSD SATA 6Gbps HDD

A server of this power will continue to run for some time with proper cooling for years. It'll need light maintenance with a server technician employed.



November 27, 2018

3.2.4 Features

The features will be designed as the application progresses. The most important features at the start is to get the camera working. This will allow videos to be posted live. Next, will be the rating system to get accurate feedback to the user. In *Figure 4: Work Breakdown Structure* shows a graphical design of NQ's Work Breakdown Structure. NQ's WBS shows the features that are currently under development.



November 27, 2018

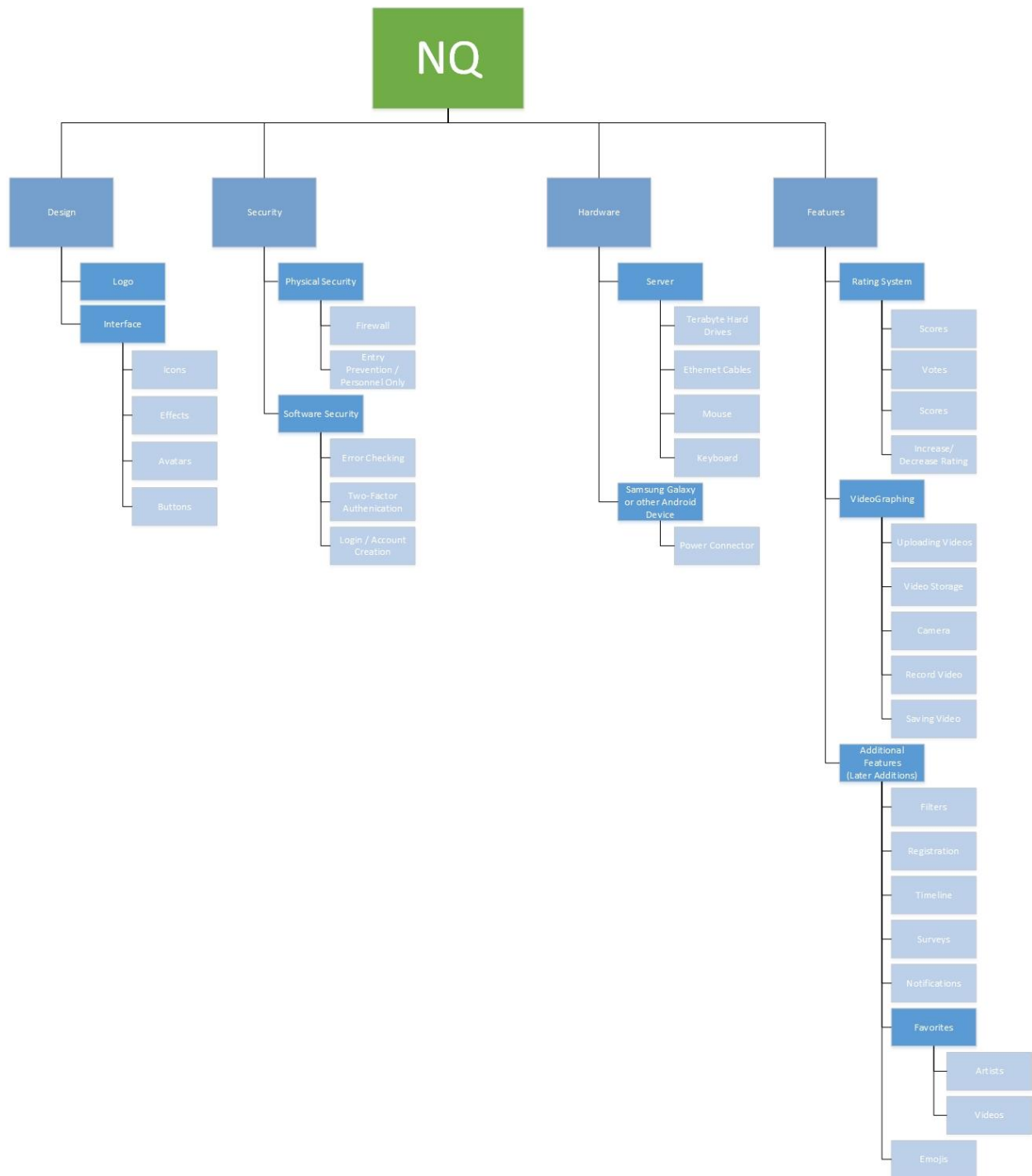


Figure 4: Work Breakdown Structure



4. TECHNICAL ELEMENTS

4.1 Technical Approach

Design Requirements: All design requirements were met using Android Studio Emulator. To show the app in real time, we used a Samsung Galaxy for our full-scale demonstration.

Procedures: Android studio emulator was the main source for how we decided on which testing solutions were needed in order to test our code. All code was catalogued and updated within Android studio. Our code, based entirely in providing an app for Android devices, was successful through our use of Android studio. The tech used in our backend was Firebase. Using Firebase gave us the freedom to decided how we wanted users to log into our app. With Firebase, users are given the choice to log in with their existing Google Gmail accounts or sign up with their email directly to NQ. These options were modeled after the login procedures used by many mobile social media applications.

Procedural Approach: The construction of the NQ Social Media App was done using the following ordered steps:

- 1) Begin coding in Android Studio
- 2) Test all application elements within Android Studio
- 3) Record both positive and negative outcomes during testing



November 27, 2018

- 4) Conduct follow-up tests inside of Android emulator
- 5) Install beta app on actual Android device. (i.e. Samsung Galaxy)
- 6) Test application functions on Android device (Samsung Galaxy)
- 7) Record positive and negative outcomes of tests
- 8) When code is successful in completing all functions, note that the code is working
- 9) Record all notes on features and updates on code in Android studio

4.1.1 Technical Architecture / Discussion

Figure 5: Technical Diagram shows the most basic elements we used when developing NQ. Our code was largely based on Java (for coding), Google, Firebase, and Android Studio. Each component is mandatory for the success of the NQ application.

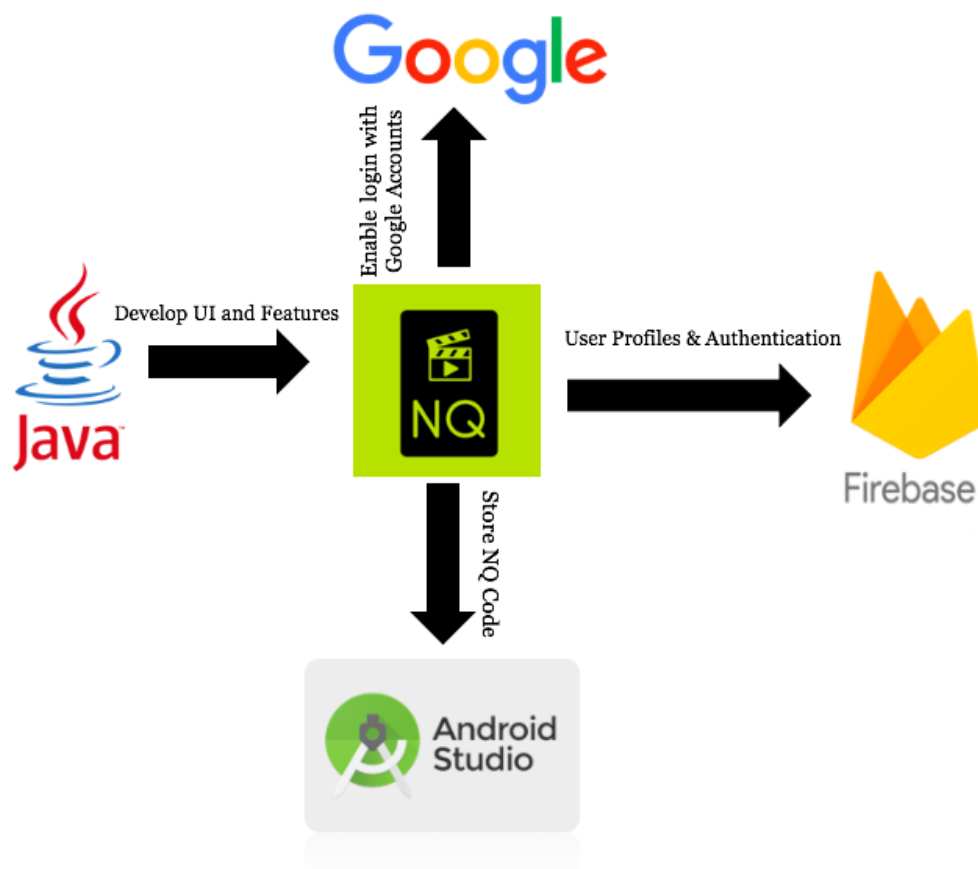


Figure 5: Technical Diagram



November 27, 2018

4.1.2 Explanation of Technical Architecture

To develop NQ, our team used the agile method of software construction. Due to the level of experience we had in developing a mobile app, we turned our focus away from the rigidity of the waterfall method so that we could make room to adjust our testing strategies and overall goals for the functionality of the app. We felt this style worked for us in the following ways:

- 1) Testing was able to be the focus of our process despite having completed the initial plan for development of NQ
- 2) Features and functions were able to be reconstructed, removed, and added without completely undoing the goals for the outcome of the app
- 3) Communication was open and always a big part of the development process
- 4) Coding of the app was done in sections which allowed us more control over the quality of each feature



November 27, 2018

4.2 Visuals

NQ reached a major standpoint when developing the camera and that was making notifications work. *Figure 6: Notifications* shows an example of NQ making progress by allowing the android device to access photos, media and files on the device.

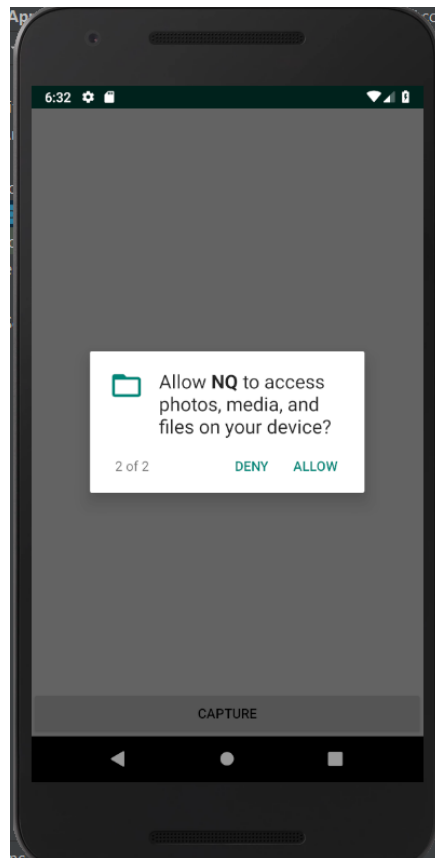


Figure 6: Notifications



November 27, 2018

4.3 User Interface

The Interface is in development. Right now, we are currently working on how to design and make the interface easy to use. The Interface breaks down into these specific categories:

- Button Placement (Touch buttons)
- Easy Swipes
- Camera Design
- Vertical / Horizontal Shift

4.3.1 Button Placement

The button placement comes down to the phone being used and the positioning of the fingers. If the buttons are too small, they become hard to click. If the buttons are too big, they clutter up the UI, cutting into the design and feel of the program. Working inside Android Studio, buttons can be put into relative position, which adjusts the buttons to the screen size. However, the buttons when flipping the screen from vertical / horizontal results in larger buttons which is what NQ is trying to avoid while maintaining the feel. Refer to *Figure 7: Button Placement* as to the feel of the button placement.



November 27, 2018

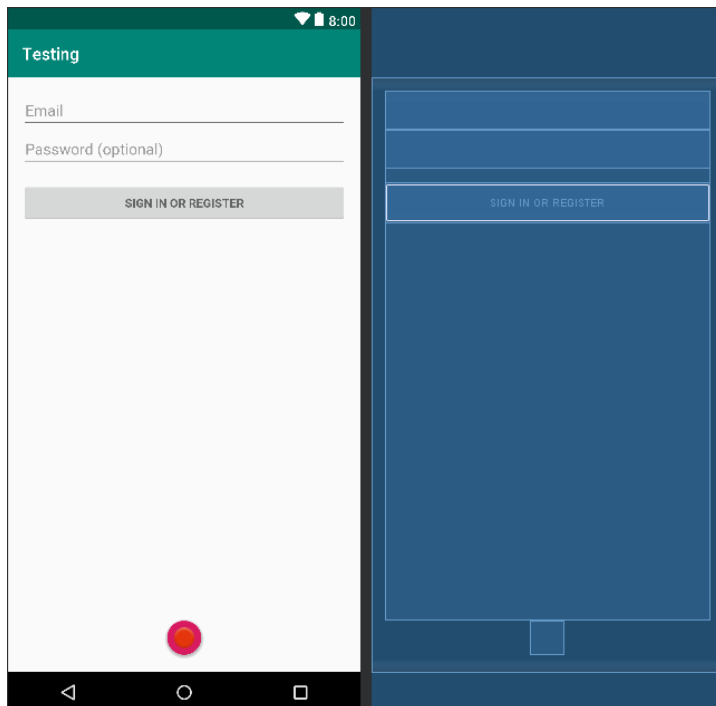


Figure 7: Button Placement

4.3.2 Easy Swipes

Swiping on the application holds one primary function. Swiping from left to right allows you to view videos without having to reload favorites. This is beneficial for viewing an artist who has several parts to his video. As one video loads, the next one will already be loading to minimize the downtime in between. There will be a slight delay from swiping left or swiping right so the user doesn't accidentally go to a different part of the video that they didn't want to see yet (watching part 1 of a video to watching part 4).

Another benefit of swiping will work when looking at video lists. Swiping up and swiping down shows the videos. Videos will be shown in thumbnails so there will be little chance of accidentally clicking a video and swiping to view videos further down.



November 27, 2018

4.3.3 Camera Design

The Camera Design adds another dimension to social media. Users of the application will be able to take photos of themselves or record videos for others to see. The goal behind the photos will be like the android camera so it's convenient and easy to use for the user. NQ plans to add in camera features that will help with videos.

4.3.4 Vertical / Horizontal Shift

NQ is going to attempt to get the same feel regardless of the way the phone is positioned. Relative positioning will work when looking at the device vertically. When flipping the phone to horizontal, the design still needs to hold the original feel, so it can be used both ways.

4.3.5 Problems Encountered

The major problem NQ has had with the application is making each piece work. Most of the designs right now are just blueprints for the code as we are still deciding what would look the best based on the research NQ has done. When the screen flips, it doesn't hold the position that we desire even in relative position. The application flips, but the program leaves out the buttons that we placed. We still are continuing to work on the issue and hope to have it resolved in the next few weeks. Refer to *Figure 8: Design Vertical* and *Figure 9: Design Horizontal* to see the issues we are running into.



November 27, 2018

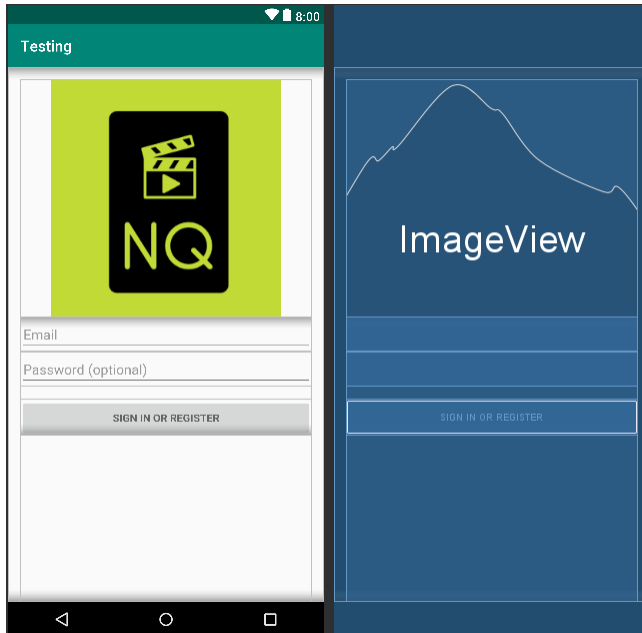


Figure 8: Design Vertical

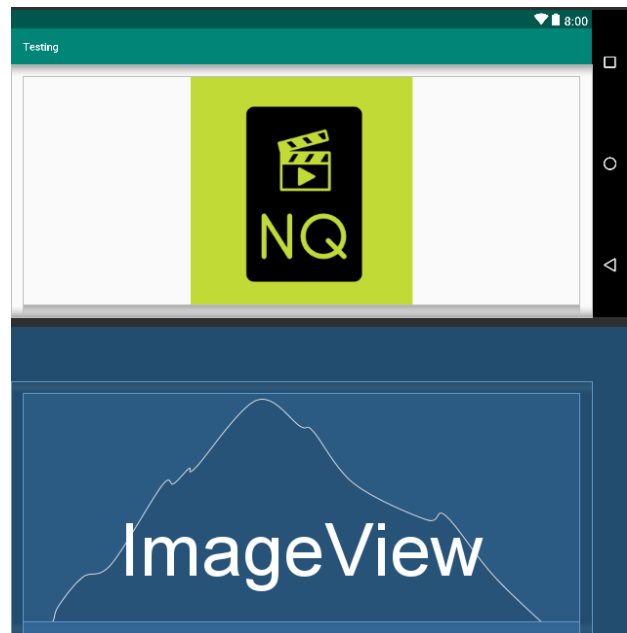


Figure 9: Design Horizontal

The camera is continuing to give us problems. Our programmer has been working steady on it for the last month and still is having difficulty making it work within the application. We figured the code would be simple to write because it's already part of the android operating system, but that's not what we've run into. While some of the components we can use, we struggle to find a good balance. As of right now, NQ's lead programmer has quoted the use of 300 lines of code, and the camera is still not functioning as it should.

UI design continues with its own share of unique problems. The button placement is the easy part but making the buttons so they're easily clickable has been a struggle. With the buttons being too big, the UI became cluttered and the design shifted just a bit. When the buttons were too small, they were harder to click. NQ has managed to work through on this problem for one



November 27, 2018

design part and that's while the phone is in vertical positioning (up and down). The Horizontal view is still being worked on. We are considering writing a multi-code loadout at the start of the application. What this will do is split the code into two different parts, one containing the code for the vertical while the other contains the code for the horizontal. The issue we are running into is because of the code loadout at the start, each time the user goes from vertical to horizontal, it reloads everything. For example, if the user has a video loaded for vertical positioning and switches to horizontal, it will reload that video again. We are still brainstorming on how to fix this.

Network is our biggest problem because of lack of funding. For this application to work, we would need a network setup that a user can send registration information too. Users will also need to be able to save their videos. NQ doesn't have anyone pursuing a network degree, so we are working on ways to accomplish this based on the knowledge we have. It looks good on paper, but with no funding and lack of space, we can't fully test out the idea behind it. We have come up with a work around for registration issues so users can register, but it'll never go out to the internet for others to see until we find another way to accomplish this.



5. PROJECT TESTING

5.1 Overview

Reference this section for information regarding the processes and methods used to test the functionality of the NQ Mobile App. NQ can currently be run on Android devices. The following groups will be able to use this section as a guide:

- Developers
- Beta Users
- UI Designers
- Team Members

5.2 Objective

The immediate goal of testing is to determine which elements of the application source code are running as predicted and which ones are working incorrectly. In the testing phase, each section of the app is tested individually for optimal testing.

5.3 System Testing

Testing of NQ will be conducted on the entire app so that any issues that are found can be reconciled before it is rolled out to users. By testing the application as one unit, the quality of the app will be determined from the point of view of a potential user.



5.4 Testing Procedures

The following is required to begin application testing:

- Establish pass/fail conditions
- Create a table to record all test attempts and results

Tests that will be performed:

- 1) Launch Test: Results will determine if NQ can be started and stopped on an Android phone.
- 2) Stability Test: Determines how long the application runs without performance errors.
- 3) User Interface Test: This test will involve interaction with the app from the point of view of a user.
- 4) Functionality Test: Each element of the app will be tested to make sure they work as they are designed to.

5.5 Testing Scenarios

The Testing Scenarios display what NQ has done to check the application. Each test listed in Testing Procedures will be more detailed here.

5.5.1 Launch Test

The launch test is conducted using both the Android Emulator and the Samsung Galaxy S9 phone. The purpose of this test was to keep launching the application over and over to determine if any issues occur. Each program will be tested 25 times for accuracy. The questions asked for this scenario are:



1.) Will the application open after several launches back to back?

2.) Did any crashes occur?

In *Table 8: Launch Testing*, the results from the above questions are answered by going with the overall times tested and if they were successful. Any errors were recorded in the results section.

LAUNCH TEST	TIMES TESTED	TIMES SUCCESSFUL	RESULTS
Android Emulator	25	25	No errors
Samsung Galaxy S9	25	25	No errors

Table 8: Launch Testing

The results from this scenario were positive. There aren't any displayed issues with running the application.

5.5.2 Stability Test

The Stability Test is designed to see how stable the code is. The code will be tested by letting the program run for set period of hours to see how it does. The objective is to answer the following questions:

- 1.) Is there any slowdown when running the program for 1/2/3 hours at a time?
- 2.) Where there any crashes involved and if there was, what was the error code?
- 3.) While navigating the menu, is there any immediate slowdown from changing activities?

In *Table 9: Stability Testing*, the questions are answered by using the following criteria:

- **Hours Tested**
 - This includes both an Android device and Emulator from Android Studio
- **Crashes**



- Record how many times it crashed if any. None will be in the box if no issue occurred.
- **Slowdown**
 - Checking resources in Windows 10 to determine how the application runs. This is given a green checkmark or a red x.
- **Pass / Fail Conditions**
 - Green Light or Red Light

STABILITY TEST	HOURS TESTED	CRASHES	SLOWDOWN	PASS / FAIL CONDITIONS
Emulation Test	1	None	✓	Application ran smoothly
Emulation Test (2)	2	None	✓	No foreseen errors
Emulation Test (3)	3	None	✗	Slight slowdown on application
Samsung Galaxy S9	1	None	✓	Ran perfectly
Samsung Galaxy S9 (2)	2	None	✗	Slight slowdown
Samsung Galaxy S9 (3)	3	None	✗	Roughly Same slowdown as the 2-hour test.

Table 9: Stability Testing

The test concluded with good results. This data will be used to make improvements to the application.

The slowdown can be minimized as we continue to evolve the code.

5.5.3 User Interface Test

This User Interface Test checks to see how the application is running from a user prospective. This test includes the menus and how easy it is to navigate through them. Performance will be checked to make sure the speed stays consistent for viewing performance. Each menu option is checked 10 times on both Android Studio Emulation and Samsung Galaxy S9. The questions answered in this scenario include:

- 1.) Are the menus working and properly linked to each other?
- 2.) Are there any performance issues when navigating menus?



3.) How long did it take to navigate from menu to menu?

- a. Anytime under 2 seconds is considered acceptable parameters. If over 2 seconds, it'll be displayed in red.

5.5.3.1 Emulation Test

In *Table 10: Emulation Interface Testing*, Tests are conducted on an Android Studio Emulator first. If the menus didn't yield errors and traveled fluently from one menu to another, that was recorded within the navigation click time. Performance issues occur only when there's a delay or crash.

USER INTERFACE TEST	MENUS LINKED	PERFORMANCE ISSUES	NAVIGATION CLICK TIME
Emulation Menu Check 1	✓	None	0.714 Seconds
Emulation Menu Check 2	✓	None	0.712 Seconds
Emulation Menu Check 3	✓	Yes	9.454 Seconds
Emulation Menu Check 4	✓	None	0.882 Seconds
Emulation Menu Check 5	✓	None	1.154 Seconds
Emulation Menu Check 6	✓	None	0.900 Seconds
Emulation Menu Check 7	✓	Yes	3.072 Seconds
Emulation Menu Check 8	✓	None	0.817 Seconds
Emulation Menu Check 9	✓	None	0.902 Seconds
Emulation Menu Check 10	✓	None	0.882 Seconds

Table 10: Emulation Interface Testing

The results concluded that on the Emulation test, there were minimal issues. There could have been other factors contributing to the delays. These factors will be investigated and added to our testing notes.



5.5.3.2 Samsung Galaxy S9 Test

The *Table 11: Samsung Galaxy S9*, displays the test results on a Galaxy Samsung S9 phone. If there were any delays, they were documented for further research. Similar to the test conducted with the Emulation Test, the Samsung Galaxy S9 will use the exact same template.

USER INTERFACE TEST	MENUS LINKED	PERFORMANCE ISSUES	NAVIGATION CLICK TIME
Samsung Galaxy S9 Check 1	✓	None	1.294 Seconds
Samsung Galaxy S9 Check 2	✓	None	1.891 Seconds
Samsung Galaxy S9 Check 3	✓	None	1.509 Seconds
Samsung Galaxy S9 Check 4	✗	Yes (Crash)	37.411 Seconds
Samsung Galaxy S9 Check 5	✓	Yes	2.266 Seconds
Samsung Galaxy S9 Check 6	✓	None	0.681 Seconds
Samsung Galaxy S9 Check 7	✓	None	1.162 Seconds
Samsung Galaxy S9 Check 8	✓	Yes	4.280 Seconds
Samsung Galaxy S9 Check 9	✓	None	1.998 Seconds
Samsung Galaxy S9 Check 10	✓	None	0.612 Seconds

Table 11: Samsung Galaxy S9 Testing

The results from the Samsung Galaxy S9 phone were optimal. The one questionable issue was the crash, but it never occurred again. The response times for the phone were greater than the emulation which was slightly odd. This scenario still contained good results.

5.5.4 Functionality Test

The Functionality Test focuses mostly on NQ's features. These include the camera, storing and saving videos, and the login system. Both systems were tested on an Android Emulator and a Galaxy Samsung S9 phone. The testing on these is different than the other charts because they are still being worked on. The questions asked in this scenario include:



- 1.) Can the user login?
- 2.) From the camera, are the user's able to save and store their videos?
- 3.) Does the camera work with good speed times?

Table 12: *Functionality Testing*, displays our current testing scenarios. The sections titled UD (under development) are still being tested in an adaptive process as each new line gets added to the section. If successful, there will be a green checkmark.

FUNCTIONALITY TEST	LOGIN	SAVE / STORE VIDEOS	CAMERA FUNCTIONS
Android Emulator Test	UD	UD (Testing Failed)	✓
Samsung Galaxy S9 Test	UD	UD (Testing Failed)	✓

Table 12: *Functionality Testing*

The results so far have been expected. This test will keep running as we move forward. The functionality test will be changed as more results continue to be added in.

5.5.5 Quick Test Scenarios

This section is detailed toward more testing scenarios that were much quicker than the original testing done in the previous sections. In the tables below, each one shows additional texts that NQ has run to ensure that the application is running successfully.

Functional Testing	Description	Pass / Fail Conditional
Text Fields	Testing the user input fields to determine functions	✓
Page Scrolling	Testing if the Application will crash if the page scrolls	✓
Navigation	Making sure every button is tied to an intent	✓
Application Start-up	Will the application start up every time without application errors? Emulation errors are expected.	✓
Application Idle	Will the application work without error if another application is pulled up and used before going back to the NQ application?	✓

Table 13: *Functional Testing Cont.*



UI Testing	Description	Pass / Fail Conditional
Image Sizes	Are the image sizes correctly being displayed in the application?	✓
Spell-Check	Testing to make sure all words are spelled correctly.	✓
Text Cut-off	Are any text being cut-off by format or another source?	✓
RecyclerView	Will the RecyclerView work while videos are playing?	✓
Video Check	Do All Videos Play?	✗
Video Check No.2	After fixing the play issue, Do the videos play?	✓
Likes Add / Remove	Testing to see if likes can be removed and added without creating any complications	✓
Video Display	Do Videos work as intended even if the phone is recording in Horizontal or Vertical mode?	✗

Table 14: UI Testing Cont.

Compatibility Testing	Description	Pass / Fail Conditional
Incoming Text Messages	Testing to see if text messages will crash the application	✓
Incoming Voice Calls	Testing to see if phone calls will crash the application.	✓
Storage	Testing to see if the storage can be saved to the phone device as well as Firebase.	✓

Table 15: Compatibility Testing Cont.

5.5 Pass / Fail Conditions

All three tests outlined in the testing procedures list must be passed in order to classify the NQ application as bug-free. If at any point a test is failed, it will be recorded and retested.

5.6 Testing Timeline

The timeline will show just how NQ manages responsibilities as to bug checking the code. This is a project that usually begins with the Lead Developer, Project Manager, and Analyst. This minimizes chances of major bugs being found after the code has been released.



5.6.1 Developer Testing Timeline

Table 16: Staff Testing, demonstrates what NQ has done to test. Every time a code is created, it is continuously tested to make sure it works. Not only is this done by the Developer, but also the Project Manager. Later, the Analyst triple checks to make sure the original idea is working and tries a variety of different ways, as an attempt to break the code. If unsuccessful, the code considered ready for beta. If not, more work needs to be done to ensure the usability.

TEAM MEMBER	TITLE	TIMELINE TESTING	FREQUENCY
William Page	Project Manager	10/8/2018 – 4/9/2019	Weekly
Brandon Ghearing	Lead Programmer	10/8/2018 – 4/9/2019	Weekly
Argina Horn	Analyst	10/8/2018 – 4/9/2019	Monthly

Table 16: Staff Testing



5.6.2 Beta Testing Timeline

In *Table 17: Beta Testing*, the Developers and the Family Members will start testing the application. The reason for this is because Family Members will be more familiar with the program because they have watched the NQ developers work on it. Video Developers will start at the same time so traffic can be determined how they save/upload/download videos on the net to check server stability.

The last group will be Young Adults, who aren't familiar with this type of system. Seeing as they are the future generation, we wish to make it easy for them to use without any confusion.

GROUPS	TIMELINE TESTING	FREQUENCY
Family Members	4/16/2019 – 7/19/2019	Weekly
Video Developers	4/16/2019 – 7/19/2019	Weekly
Young Adults	7/19/2019 – 9/23/2019	Weekly

Table 17: Beta Testing



5.7 Lessons Learned

During Testing, the biggest issues we've encountered so far is the login process, storing videos, and editing the UI to make it more user friendly. A lot of the elements we are learning while coding is making the challenge more difficult than simply writing a code that we have done before in previous Java classes. Referring to *Figure 10: Error Message*, this has been the issue we've ran into with no error codes. If there was, we could isolate the problem and fix.

Something went wrong

 Something went wrong. Please try again.

CLOSE

Figure 10: Error Message

The lack of Android experience has been really showing as we continue to develop the application. However, this has not deterred us, only made us change our method of thinking with each approach to a new complication.

5.7.1 Login Process

Our original idea was to use Google services to allow google accounts to login. Unfortunately, every test that we have conducted so far has ended in failure. There's a few more we intend to try before tossing the idea and opting to make our own login / register. We are saving this for the last resort because of the amount of time used trying to make google login possible. If we



November 27, 2018

opted to release this on different platforms, Google login might pose an issue with iOS platforms, but it's not impossible.

5.7.2 Storing Videos

Storing videos has been a challenge on its own. The code shows that everything works, but when the code is running, it doesn't seem to work at all. Movies go through the editor and say they save, but no video is saved. Currently the challenge is fixing the code so we can save our videos to the phone's external photo album.

5.7.3 UI Editing

The UI has been our biggest concern. We have constantly tested it to make sure that it's ready for the user experience. There have been a few questions regarding whether the application is really geared towards social media, or developers. NQ has been trying to implement tools for developers to use. The problem with this is the memory. Each time we ran a few demonstrations, the tools would often cause too much data on the phone (or emulator) and ultimately crash. In order to fix this, we have removed a lot of the tools relating to editing videos, while leaving the ability to see and arrange videos. We figure the coding elements for this wouldn't be too much of a change on an iOS device. If made for the PC, the application would require some editing.

5.8 Future Recommendations / Recommendations for Improvement

When it comes to future recommendations on the project, NQ is finished after this semester. We have received some ideas of directions we could go with the project. However, we feel there is still so much



November 27, 2018

to learn before creating a Social Media Application. There was no way for us to determine the amount of work needed at the start of the semester, not to mention what we had to code. While we knew what we needed to do, it was hard to perform it as we had several complications to overcome.

5.8.1 Analysis if we had to do project again

If NQ had to do the project all over again, we need to pick a focus or try to have more students on the project. When it comes to picking a focus, we needed to pick the greatest need on the application and develop that fully before moving on to other little things. The problem with a Social Media App is everyone is already expecting certain features to work that's expected in a Social Media Application. This is where having more students working on the project is helpful. Certain students could handle the features that is expected in an everyday Social Media Application. These features include:

- Subscribing
- Friends Lists
- Play Lists
- Favorites
- Communication

Because the application required a lot of resources, NQ had to purchase additional services so the expo could be successful. This is where more analytics would come into play as to the realistic costs needed for development. All of this would be charted out thoroughly if we restarted the project.



November 27, 2018

5.8.2 What if we had more time?

If NQ had more time to develop the project, we could approach each section and fully bug check that before moving onto another section. The application would look nicer and the demonstration would be fully developed for everyone to use during the Expo. NQ had quite a few issues with time constraints and completing sections.

5.8.3 Suggestions from Others

The comments from students and professors were pretty useful. During our first presentation, a few students didn't like the use of colors. This was mainly due to the magenta color displayed during the text. NQ took that advice to heart and completely switched it to match the rest of the colors on the interface. The next suggestion was to possibly add the use of buttons so it didn't look as though the users would be clicking text. NQ implemented this feature as well.

The other complaint was the application wouldn't show videos. It turns out, we had all the code right for it, it's just firebase was full, and it didn't properly tell us this until the end of the month summary. If we knew that earlier, the videos would have been displayed at the presentation.

NQ did receive some valuable feedback from a professor who said that this application could also be used for school purposes. The core design of our application is that users can display videos, receive likes and comments, and have an easy to use interface. NQ has all of these features. So how could they be used for school purposes? Students could post their problems with a video to demonstrate exactly what they're doing and what's going wrong. The professor could then respond to the student with their own video or answer them in the comments. Because NQ sends out notifications on feedback, the



November 27, 2018

professor or student will get a live update when their question has been answered. This is something that we never considered.

5.8.4 Future Project Plans

As of right now, NQ has no future plans for the project. However, there's a chance that we could develop this application for UC at the idea given by the professor. It just comes down to the idea of the professors at UC feel this is something they could use.



6. CONCLUSION

6.1 Fall Semester 2018:

A working prototype of the NQ mobile app was presented during Fall 2018 semester presentations. The prototype presented shows a working camera that will be used to record videos in-app without the use of expensive equipment. The team was also able to fluently discuss the software and hardware used to build up the rating system and design the UI.

6.2 Spring Semester 2019:

The final product will be presented during the second half of Senior design. Going forward, the NQ plans to spend more time working on creating more features that will enhance the user and video makers experience.

6.2.1 Lessons Learned for Spring Semester / Fall Semester

NQ's learned a lot over the course of the two semesters. In this section, we break down the Fall and Spring semesters.

6.2.1.1 Fall Semester Work

During the Fall Semester, we learned just how long it takes to develop an easy-to-use UI. While writing the XML wasn't too difficult, writing the Java was a completely different story while using Android Studio. We hit every error possible in Android Studio. Despite all that, we were still able to develop a UI before the start of presentation.



November 27, 2018

6.2.1.2 Spring Semester Work

The Spring Semester was a nightmare for us. We had to develop so many things at once to make the application work. We had database, video, reference, and messaging errors.

Database Errors were occurring mostly because of our lack of coding skills. Once we started to fully understand how databases work, writing the code from Android Studio to Firebase was a lot easier. However, it took us quite a bit to learn this. We also made an option to be able to login to the application using Google. This took a bit of time because the guides online were all outdated. As a matter of fact, the Google Developers Guide for syncing Google services to the application stated specifically to NOT INTEGRATE FIREBASE BEFORE THESE STEPS. It turns out, Firebase was the one that was right and fixed our ability to login with google. Because of those inaccurate directions, it added to our lessons learned.

Video errors mostly came from “Null Pointer Exceptions.” This basically meant that we weren’t pinpointing any data yet based on our code, we were. NQ solved the problem that was caused by the field because it was empty. If the field is empty, it draws a “Null Pointer Exception” even if it exists. Once we figured out how to fix this by adding a few control statements, most of our Null Pointer Exceptions went away. There was occasionally a few of the errors left, but this was due to the fact that we exceeded our data on Firebase for the month.

Reference errors was a big problem. These were nearly impossible to diagnose because Android Studio doesn’t post where the error occurred. We have to assume just where the error might be and go back to fix. We fixed this with the debug option. While it takes a lot of hours to diagnose, the debug eventually leads us down the right path to fix the error.



November 27, 2018

Messaging errors were the lowest of our concern because it was mostly due to Firebase having issues with how much data was being sent back and forth because of the videos. Once the subscription for firebase was paid, this issue became non-existent.

6.2.2 Abilities/Skills Developed & Enhanced

As the Project Manager for NQ, my skills developed in the coding field as well as being able to diagnose broken code. When we started to fall behind in the project, duties were changed and updated accordingly as to where most of our focus became coding the project. Refer to *Table 18: Fall Project Resources* for the work done.

Project Duties Fall Semester	Percentage Allocated
Research	20%
Project Report	40%
Design	20%
Coding	20%

Table 18: Fall Project Resources

For the Spring Semester, this had to be changed completely. Originally the plan was to allocate more of our time to coding the assignment which our Programmer was in charge of. In *Table 19, Spring Project Resources (Original)*, demonstrates what our idea was to keep the project going and up-to-date.

Unfortunately, this became too much for NQ as we were notified late of what problems the Programmer was having. NQ wasn't defeated yet and changed our resource allocated and time needed to complete the assignment. In *Table 20: Spring Project Resources (Updated)*, we changed our focus and upped our time into Coding. This was working much better, but our Analyst had to move more towards class assignments.



Project Duties Spring	Percentage Allocated
Research	0%
Project Report	30%
Design	30%
Coding	30%
Expo Preparation	10%

Table 19: Spring Project Resources (Original)

Project Duties Spring	Percentage Allocated
Research	0%
Project Report	10%
Design	5%
Coding	70%
Expo Preparation	15%

Table 20: Spring Project Resources (Updated)

6.2.3 Progress Since Fall Semester

At the end of the Fall Semester, NQ only had a working UI and the camera code for the Application and that wasn't fully functional. It was important for us to figure out how to set up the application so we can add code later. During the Spring Semester, NQ managed to double their efforts and code in the logic behind the camera, add a ListView, add a RecyclerView, develop the database, storage, authentication, and record features. The ListView was later scrapped as NQ was able to get the RecyclerView working. In the end, it can be summarized as the Fall Semester pertaining to the research, and time to think. The Spring Semester can be summarized as the time to code, and time to act. This turned out to be a functional logic for the group.

6.2.4 Lessons Learned from Expo

When it came to the Expo, our application was very popular for the younger generation and class mates. However, It was not popular among the judges. This was expected because based on the information



November 27, 2018

given to us about the judges, there can be some predictability as to which way they would judge and what elements of a project they would favor. There were a few questions asked to us by the judges which included:

- 1.) Why would videographers use this application?
- 2.) Does the application include editor tools?

This had to be explained to the judges as this isn't a "Videographer Editing Tool." As our poster demonstrated, this was "Uniting Videographers with their Audience." Because of this, there wouldn't be any editor tools. As for why they would use this application, you have to think about why YouTube has problems. The reason videographers have trouble with YouTube is because it comes down to how much you pay and use key words in your videos that determine how high you get ranked. Video Gamers who make "Let's Play" videos found their income waning over the years as the previous top spots available to them were given to others. NQ is a way where everyone can see your videos regardless.

Unfortunately, NQ wished the judges would have offered some feedback as to areas of improvement or ways we could consider developing the application in another way. This is where the students and the younger generation came in.

We had several students from high schools and other UC students enjoying the application. There was a Sophomore at UC who stayed at our booth with a small group of students for over a half-an-hour. They liked the simplicity of the application and how easy it was to use. This was NQ's purpose, so it was refreshing to see the younger crowd proceed as we projected.

What we didn't expect was a professor who came to our booth and talked to us for over 20 minutes. The professor gave us ideas of directions we could go with this application and even recommended that it would be perfect for UC. This is further detailed inside Section 5.8.3 *Suggestions from Others*.



November 27, 2018

We learned a lot from this semester. NQ developed a solid product and we couldn't be happier. We are very thankful to the staff of UC for helping guidance with needed and making sure we didn't give up. We hope our next big project comes with similar rewards.



November 27, 2018

7. REFERENCES

Dean, Brian. "We Analyzed 1.3 Million YouTube Videos. Here's What We Learned About YouTube SEO." *Backlinko*, 28 Feb. 2017, backlinko.com/youtube-ranking-factors.

"Be in the Know: 2018 Social Media Statistics You Should Know | Disruptive Advertising." *Google*, Google, www.google.com/amp/s/www.disruptiveadvertising.com/social-media/be-in-the-know-2018-social-media-statistics-you-should-know/amp/.

"How to Build and Use a Business Budget That's Useful All Year Long." *Small Business Administration*, www.sba.gov/blogs/how-build-and-use-business-budget-thats-useful-all-year-long.

"PowerEdge T330 Tower Server." *Dell United States*, www.dell.com/en-us/work/shop/cty/pdp/spd/poweredge-t330/

Driver, Saige. "Social Media for Business: A Marketer's Guide." *Business News Daily*, 15 Oct. 2018, www.businessnewsdaily.com/7832-social-media-for-business.html

Hatch, C. (2018, March 13). Be in the Know: 2018 Social Media Statistics You Should Know. Retrieved from <https://www.disruptiveadvertising.com/social-media/be-in-the-know-2018-social-media-statistics-you-should-know/>

Start Integrating Google Sign-In into Your Android App | Google Sign-In for Android | Google Developers. (n.d.). Retrieved from <https://developers.google.com/identity/sign-in/android/start-integrating>



November 27, 2018

APPENDIX A

NQ Logo Artwork





November 27, 2018



Poster



NQ - Uniting Videographers With Their Audience

William Page, Brandon Ghearing, & Argina Horn
College of Education, Criminal Justice, and Human Services – School of Information Technology

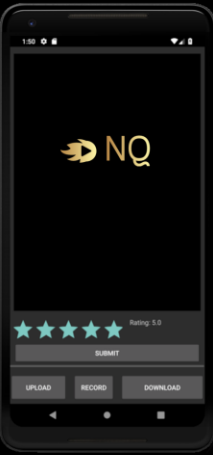
NQ is a video sharing platform that partners videographers with their viewers. In a sea of social media apps, NQ stands out because of its ability to offer videographers constructive feedback designed to help them make better content.

WHY (The Problem)

There is a wide selection of social media applications that offer users the chance to view entertainment. Very few of these options offer a customizable viewing experience for the viewer while constructively critiquing videographers.

TECH USED





HOW (The Solution)

Upload content based on the category it fits in and receive ratings. Higher ratings lead to content upgrades to use on your future videos. Getting featured makes places your video in the top spot in your category.

POTENTIAL USERS

- ❖ Aspiring filmmakers, videographers, and artists
- ❖ Social media of all ages
- ❖ Fans of visual media enthusiasts

ACKNOWLEDGEMENTS: Special thanks to Advisor Abdou Fall and Professor Jim Scott

Figure 11: Expo Poster