

YouTube Theater

by

Joan Wulff-Vanderpuije, Jonathan Davis and Patrick Marfo

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 2017, Joan Wulff-Vanderpuije, Jonathan Davis and Patrick Marfo

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

Joan Wulff-Vanderpuije Jonathan Davis Patrick Marfo

April 17, 2017

Joan Wulff-Vanderpuije, Jonathan Davis, Patrick Marfo

Date

Russell E Mc Mahon

Russell McMahon

April 17, 2017

Date

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

April 2017

Table of Contents

Abstract	1
Problem Statement	2
Introduction	2
Project Description	2
Problem.....	3
User Profile.....	4
Use Case Diagram	5
Project Management	6
Budget.....	6
Deliverables	6
Project Schedule	7
Methodology.....	9
Testing.....	10
Testing Plan	11
Problems Encountered.....	13
Conclusion	15
References.....	16
Appendix.....	17
Appendix A - Application Architecture Diagram	17
Appendix B - Code Snippets	18
Appendix C - Tech Expo 2017 Poster	18

List of Illustrations

User Profile	4
Use Case Diagram.....	5
Budget.....	6
Project Schedule.....	7
Team Member Testing.....	12
Test Results.....	14

Abstract

According to the Pew Research Center organization, "for today's teens, friendships can start digitally: 57% of teens have met a new friend online...42% of all teens spend time with friends on messaging apps such as Kik and WhatsApp... 52% of all teens spend time with friends playing video games; 13% play with friends daily". Many applications address the flexibility of communication and the exchange of media without speaking to issues like watching a video in synchrony. Therefore, YouTube Theatre provides a web application that allows multiple users to watch YouTube videos virtually with one another in real-time. The application simulates the experience of watching a YouTube video in person with someone by displaying a live video-feed of the other connected party. Instant messenger functionality, a search and playlist feature is available for users while watching videos. This application can be not only used for leisure movie time with friends but also, group studying and work related meeting that require watching a video from YouTube.

Problem Statement

Introduction

According to YouTube, “Launched in May 2005, YouTube allows billions of people to discover, watch and share originally-created videos. YouTube provides a forum for people to connect, inform, and inspire others across the globe and acts as a distribution platform for original content creators and advertisers large and small. YouTube is a Google company”.

Project Description

YouTube Theater is a web application that allows multiple users to watch YouTube videos virtually over the Internet with one another in real-time. The web application simulates the experience of watching a YouTube video in person with someone by displaying a live video-feed of the other connected party. Instant messenger functionality is available so users can exchange messages while watching videos. The web app has a search and playlist feature to allow users to find videos on YouTube and then add them to a playlist for sequential playback. Users can take screenshots of their current screens and draw on top of the image and send it to one another or even upload the image to various social networks.

Problem

Ever want to have a movie night in the comfort of your own home? Ever want to watch a video with someone who is across the world from you? Ever need to study with a friend

we is miles away? Currently, YouTube doesn't allow multiple users to watch a video in synchrony. It only allows users to search for a video, create a play list and share the link of video with others. When it comes to watching a video with someone you have to physically be there with them, which is very time consuming and may limit flexibility and comfortability. YouTube Theater is a solution that solves all of those problems and more. It allows multiple users to watch YouTube videos virtually over the Internet with one another in real-time. The web application simulates the experience of watching a YouTube video in person with someone by displaying a live video-feed of the other connected party.

User Profile

The potential users for the YouTube Theater are according to an article published by Tubular Lab on November 20, 2014 mid-to-late Millennial's, males and females between the ages of 18 and 34 who are watching various content on YouTube. Users will be able to discover videos and simulate the experience of watching a YouTube video with another person as if they are in the same room, by displaying a live video-stream of the connected parties. The web app provides an instant messaging functionality, which allows users to send messages while watching videos. The web app has a search and playlist feature to allow users to find videos on YouTube and then add them to a list for sequential playback. Users can take screenshots of their screens and draw on the image and send it to one another or even upload the image to various social networks. The user interface will be simple yet creative, intuitive, have easily accessible tabs, and an interactive web page. See **Table 1** below.

Table 1 User Profile

*After doing research we believe that users can begin using this application as early as 13 years old

<p>Potential Users:</p> <ul style="list-style-type: none">• YouTube Theater Customer (16 to 34 years) (Daphne 2014)• Anyone who knows how to use YouTube
<p>Software and Interface Experience:</p> <ul style="list-style-type: none">• The YouTube Theater Customer has moderate to high experience using YouTube.• The user will have general knowledge of finding videos in YouTube and experience using web applications.
<p>Task Experience:</p> <ul style="list-style-type: none">• The Customers will be comfortable with and be able to find videos, send link to friend to watch video simultaneously, create a playlist, use screenshot and chat functionality.
<p>Frequency of Use:</p> <ul style="list-style-type: none">• YouTube Theater Customers will heavily use this application during normal business hours (12:00PM – 11:00PM)
<p>Key Interface Design Requirements that the Profile Suggests:</p> <ul style="list-style-type: none">• Simple and intuitive UI• Easily accessible navigation tabs/buttons• Interactive page

Table 1 User Profile

Use Case Diagram

Figure 2 Shows Use Case Diagram YouTube Theater

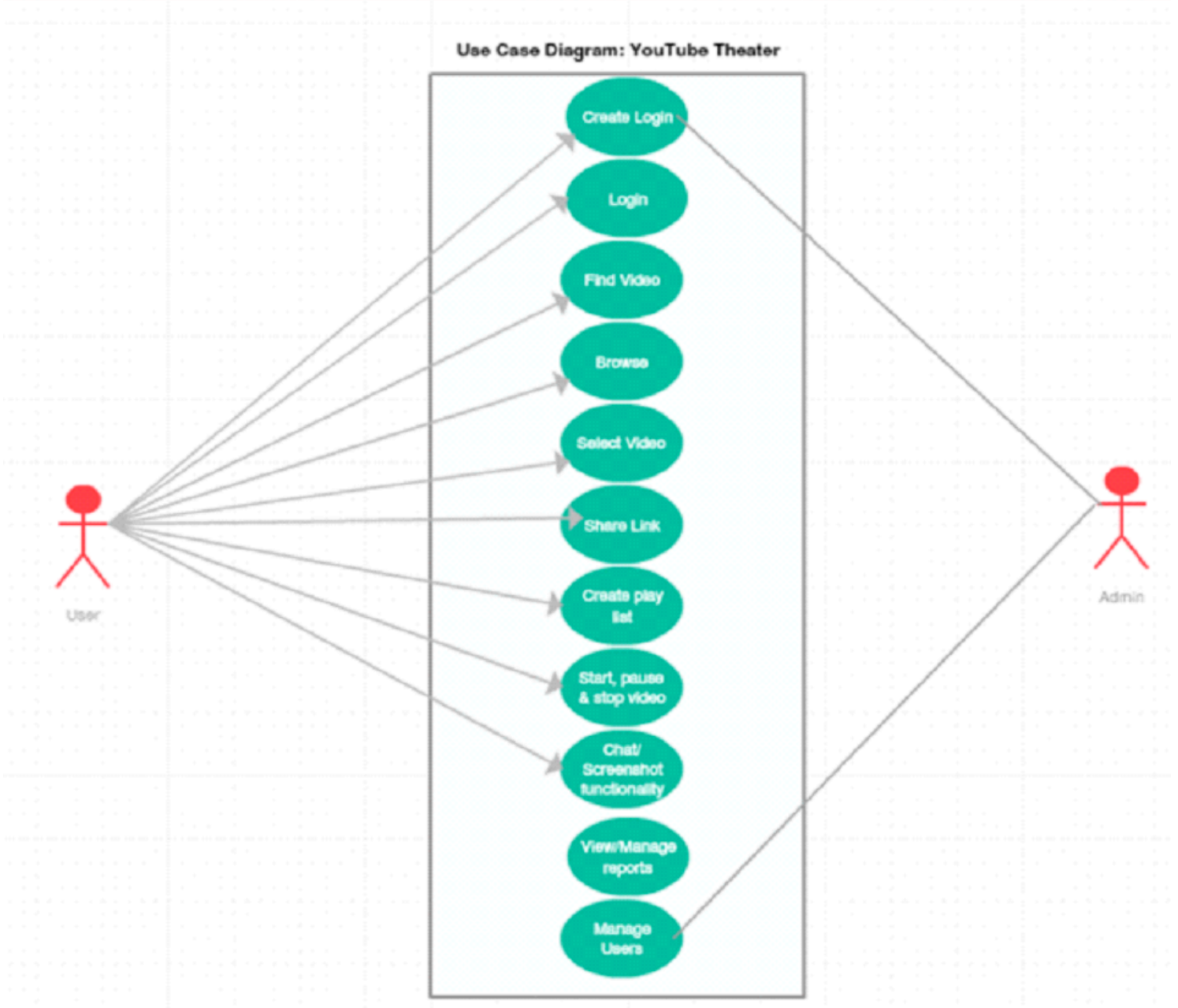


Figure 2 Use Case Diagram

The remainder of this final report outlines in detail how the project was completed. The report includes the following sections: project management, testing, and conclusion.

Project Management

Budget

Table 3 represents the project budget. It includes the real-world costs of approx. \$14.6 Million to create YouTube (Lawler 2010) and how much it cost us to create YouTube Theater. Our cost was \$0 as this is our senior project and we will use YouTube public API which is free.

No.	Item	Line Item Total
Infrastructure		
1	Labor	\$ 8 Million
Hosting Expenses (for 3 months not including data center cost)		
2	Labor	\$ 3 Million
Operating Expenses (first 18 months)		
3	Employee Compensation etc.	\$ 3.6 Million
	Subtotal	\$ 14.6 Million
Youtube Theater		
4	Entire Application	\$0

Table 3 Budget

Deliverables

First Semester

- Research WebRTC
- Research UI
- Research SQL injection
- Build sample WebRTC
- Build Sample using App using YouTube API
- Prototype the UI/UX design
- Research more Features
- Begin Signaling server
- Build and integrate database
- Finalize UI/UX design

Second Semester

- Implement Features
- Video Queue functionality
- Implement Live Video Chat
- Implement Instant Messaging
- Implement screenshot functionality
- Image edit/drawing
- Integrate Social Network sharing
- Testing
- Final Version for IT Expo

Project Schedule

Figure 4 Shown below shows original timeline estimates

First Semester	Week 1-5	<ul style="list-style-type: none"> • Research WebRTC • Research UI
	Week 5-9	<ul style="list-style-type: none"> • Research SQL injection • Build sample WebRTC • Build sample app using utilizing YouTube's API • Prototype the UI/UX design
	Week 9-15	<ul style="list-style-type: none"> • Research more features • Finalize UI/UX design

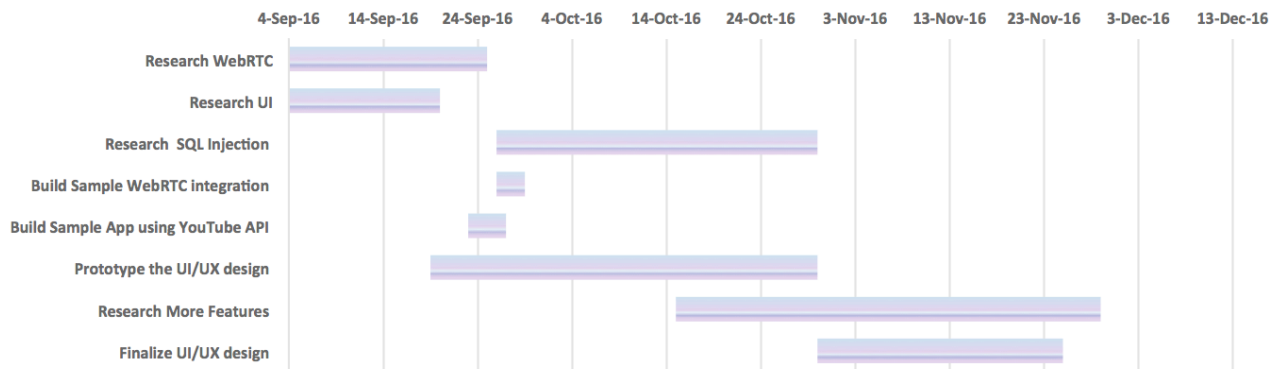


Figure 4 First Semester Timeline

Figure 5 below shows second semester timeline

Second Semester	Week 1-5	<ul style="list-style-type: none"> • Build Signaling Server • Build and integrate database • Implement UI/UX design • Implement Features <ul style="list-style-type: none"> ○ Video Queue functionality ○ Implement Live Video Chat ○ Implement Instant Messaging ○ Implement screenshot functionality ○ Image edit/drawing ○ Integrate Social Network sharing
	Week 5-8	<ul style="list-style-type: none"> • Testing <ul style="list-style-type: none"> ○ Bug fixes ○ Sql Injection ○ UI
	Week 8-15	<ul style="list-style-type: none"> • Final Version For IT Expo <ul style="list-style-type: none"> ○ Making sure we have all we need for expo ○ More testing ○ Presentation and prep ○ Finalize research paper ○ Oral Presentation ○ Final Report

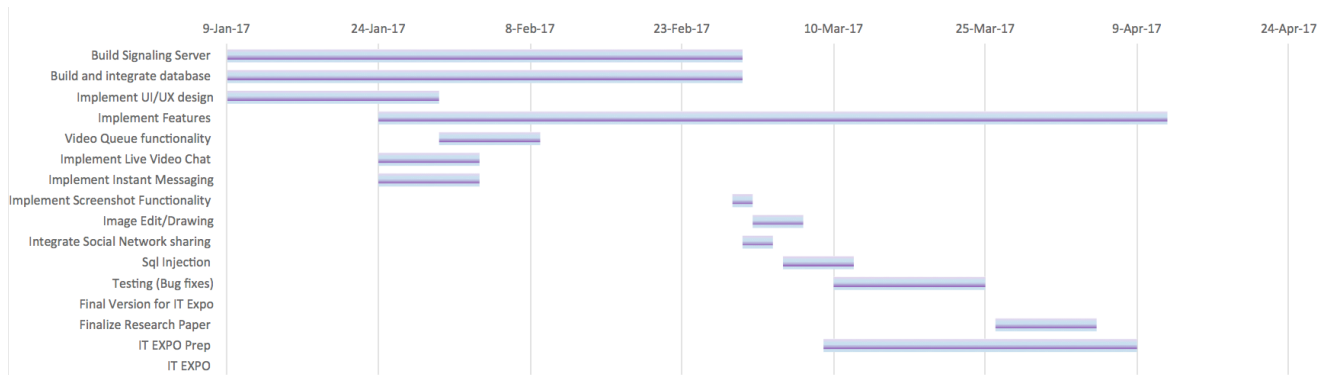


Figure 5 Second Semester timeline

Methodology

Originally, we wanted to create a web application that allows video synchronization for Netflix. With Netflix not having a public program interface; we resulted in creating one for YouTube. We used the traditional approach in developing this application because we knew the project scope and goals will remain consistent till project completion. Our goals were to create a secure web application that allowed multiple people to watch videos in synchrony from any device. During the planning initiation phase we brainstormed and discussed what we wanted the purpose of the application to be. This allowed us to come up with a UI design that could act as a proof of concept for our final version. After creating this proof of concept we used it as a foundation and began to improve on it while progressing through the planning and design stage. Towards the end of spring semester, we added more content, features, and abilities and eventually had a functional working application. Our main goal was to make the user feel as if they were using YouTube but with more features. This ranged from inner workings of the application and making it more efficient to making a streamlined and polished graphical user interface. Testing was done in between the execution stage, and constantly throughout the preparation for the IT Expo in the spring.

Testing

This section explains the testing methodology for the YouTube Theater Web Application and should be used as a guide. The following individuals should use this section:

- Developers
- Project Managers
- Application Testers

Scope

The scope of this plan is to test the operation, ease of use, security and satisfaction of the YouTube Theater application. The test was be organized based on specifications and requirements of the application. This allow developers to know the success or failure of the application.

Objective

The objective of testing is to verify that a particular module of the source code is working as expected. These test are designed to test components in isolation or as a unit. The developers and security testers will run the tests for the components they are working on. Each member will list out a set of goals in order to measure the success of failure of testing.

Test Plan

We used the Agile Methodology to test our application. This testing method allowed us to test the development of our application periodically. We were able to see how easily and efficient the application navigates to satisfy user's needs. One thing we have noticed is that many people get frustrated easily when they do not have the ease of use with their applications. Therefore, testing throughout development showed us what we needed to fix. Along with that, we dedicated the last few weeks of the second semester to focus on testing. First we tested the application ourselves then asked our family members and friends to test and give us feedback.

*We performed security testing against our code to make sure there were no vulnerabilities.

Testing Procedure

The follow Push to Test were used for YouTube Theater web application:

1. Functional Testing (Regression, Integration, Smoke Testing)
2. Browser Compatibility Testing.
3. Performance Testing.
4. Security Testing.
5. Production Monitoring.
6. Usability Testing.

Pass/Fail Conditions

It is expected for the YouTube Theater web application to pass all tests with a 100% success rate. If it does not pass, the tester will document the issue, find mitigation and retest it till it is successful.

Schedule of Team Member Testing

Table 6 below is the schedule for the team member testing:

Team Member	Timeline to be completed	How Often
Developer	3/1/2017	Weekly
UI tester	3/2/2017	Weekly
Project Manager/Security	3/3/2017	Weekly

Table 6 Team Testing

Entry and Exit Goals

Entry Goals:

- ~ Build complete
- ~ Self-testing complete
- ~ Test environment is setup

Exit Goals:

- ~ All tests are run
- ~ Bugs/security vulnerabilities are documented and fixed
- ~ Re-testing complete

Risks

The following may impact the testing process:

- Internet/ Network availability
- Delay in bug fixes
- Browser issues
- Web site performance

Logging Test and Reporting

If a bug, security vulnerability or breach were to found during testing then, the individual will document it. The developers would then met after the test and decide if the reported issue is actually a problem or a false positive.

Once it has been identified as an actual problem, developers along with security analyst would have develop a mitigation plan and act upon it.

System Testing

YouTube Theater Application was tested as a complete application as if it were ready for production. This helped make sure that there aren't any holes in development and all members of development have completed their goals. This allowed testing to affect the whole application rather than some parts.

Problems Encountered

Our team initially had a lot of problems with the login in page but, we were able to get it resolved. Along with that, we noticed a few seconds delay in video synchronization but later found that once a message is sent in chat box is send within parties in session everything went back to normal. Initially, we wanted to use amazon web services to host our application but, after further thought, we decided to host it on our own server. This gave us the flexibility of turning application on an off as we pleased. We also found it easier to troubleshoot with it being on our own server. Ultimately, we know that this solution is only useful for our senior design but we wouldn't be able to host in the future if it goes to production.

Test Results

YouTube Theater Test Results		
	Type of Test	Pass/Fail
	Searching for Video	Pass
	Selecting Video	Pass
	Adding Playback Queue	Pass
	Create/Join Session	Pass
	Application Navigation	Pass
	Sql injection	Pass
	Taking a photo and editing	Pass
	Video Sync	Pass

Table 7 Test Results

Conclusion

Throughout this process we learned about programming for a web application, specifically: allowing two or more users to watch a YouTube simultaneously in real-time. It was a challenge that we conquered and feel proud of. Throughout the hiccups it was a great experience and definitely learned a lot. One of the main things we learned was how to manage time; time flies by quickly. Fall 2016, it seemed as if we had a lot of time to get everything done; but it wasn't so. Spring 2017, we found ourselves working harder to achieve our deliverables on time due to us not managing our time correctly. Overall, we learned something out of it and we won First place in the application development category; which is awesome! To be honest, successfully finishing and passing senior design was our main goal but, with our hard work and dedication, winning was “icing on the cake”.

With great feedback and need from the expo we are now thinking of how we can sell our product to YouTube and possible advance our product. With that said, we are very excited for what the future holds. In conclusion, we just want to express our upmost gratitude and appreciation to our family, friends, professors, the school of Information Technology and anyone who supported and encouraged through this project.

References

Lawler, R. (2010, March 19). How Much Did It Cost to Build YouTube? Retrieved April 17, 2017, from <https://gigaom.com/2010/03/19/how-much-did-it-cost-to-start-youtube/>

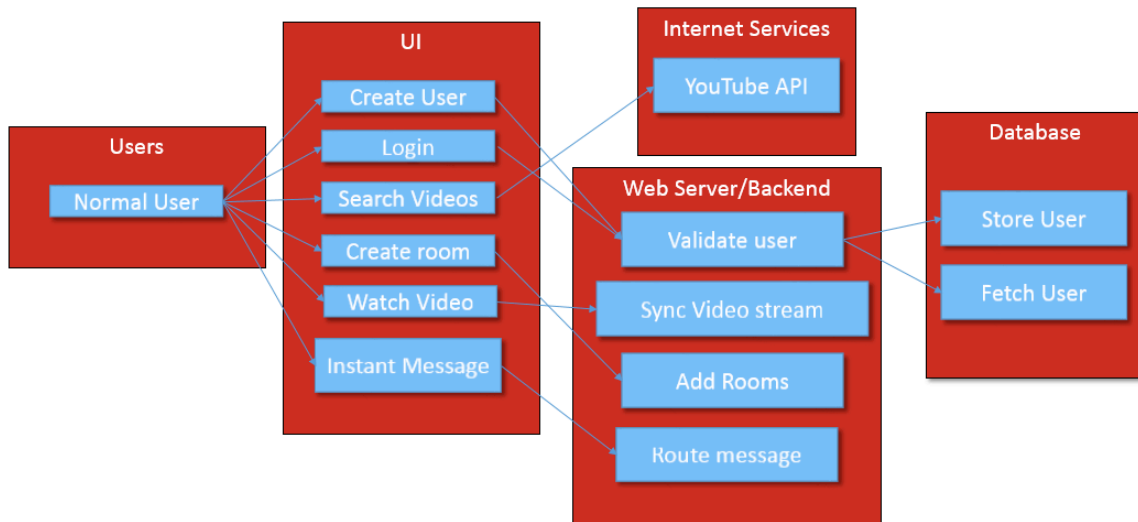
Lenhart, Amanda. "Teens, Technology and Friendships." Pew Research Center Internet Science Tech RSS. August 06, 2015. Accessed November 02, 2016. <http://www.pewinternet.org/2015/08/06/teens-technology-and-friendships/>.

Miller, Daphne. "Demographics on YouTube: Males and Females 18-34." Tubular Labs. November 20, 2014. Accessed October 10, 2016. <https://tubularlabs.com/blog/demographics-on-youtube-males-and-females-18-34/>.

Youtube, Y. (n.d.). About YouTube. Retrieved April 17, 2017, from <https://www.youtube.com/yt/about/en-GB/>

Appendix

Appendix A - Application Architecture Diagram



Appendix B - Code Snippets

```
(function () {
    'use strict';

    var app = angular.module('youTubeTheaterApp', [
        // Angular modules
        'ui.router',
        'jtt_youtube'
        // Custom modules
        // 3rd Party Modules
    ]);

    app.config(['$stateProvider', '$urlRouterProvider', '$locationProvider',
        function ($stateProvider, $urlRouterProvider, $locationProvider) {

            //=====Routing using UI Router=====

            $urlRouterProvider.otherwise('Login');

            $urlRouterProvider.when('/', 'Login');
            // use a state provider for routing
            $stateProvider
                .state('Login', {
                    url: '/Login',
                    templateUrl: 'views/login.html',
                    controller: "LoginController"
                })
                .state('Register', {
                    url: '/Register',
                    templateUrl: 'views/register.html',
                    controller: "RegisterController"
                })
                .state('Forgot', {
                    url: '/Forgot',
                    templateUrl: 'views/forgot.html',
                    controller: "Forgotcontroller"
                })
                .state('Home', {
                    url: '/Home',
                    templateUrl: 'views/Home.html',
                    controller: "HomeController"
                })
        })
    });
});
```

```

<!DOCTYPE html>
<html ng-app="YouTubeTheaterApp">
<head>
<meta charset="UTF-8">
<title>YouTube Theater</title>
<meta name="viewport" content="initial-scale=1.0, width=device-width" />
<!-- Style Sheets -->
<link rel="stylesheet" type="text/css" href="css/bootstrap.min.css">
<!-- Font Awesome -->
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.5.0/css/font-awesome.min.css">
<!-- Ionicons -->
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/ionicons/2.0.1/css/ionicons.min.css">
<!-- Font Icons -->
<link rel="stylesheet" href="css/hover-min.css" />
<link rel="stylesheet" href="css/animate-min.css" />
<link rel="stylesheet" href="css/owl.carousel.css" />
<link rel="stylesheet" type="text/css" href="css/styles.css">
<link rel="stylesheet" type="text/css" href="css/responsive.css">
<link href="css/jquery.mCustomScrollbar.css" rel="stylesheet" />
<link href="scripts/themes/smoothness/jquery-ui.min.css" rel="stylesheet" />
</head>
<body ng-controller="MainController">
<div id="wrapper">
<div id="main-content">
<!-- Load HTML templates-->
<div ui-view></div>
</div>
</div>
<!-- Scripts -->
<script src="js/wow.min.js"></script>
<script src="js/jquery-1.12.3.min.js"></script>
<script src="js/jquery.mCustomScrollbar.concat.min.js"></script>
<script src="js/bootstrap.min.js"></script>
<script src="js/owl.carousel.min.js"></script>
</script>
</body>
</html>

```

Appendix C - Tech Expo 2017 Poster

The following is the Tech expo poster for the YouTube Theater web application.



YouTube Theater

Joan Wulff-Vanderpuije Patrick Marfo & Jonathan Davis
 Advisor: Prof. Russell McMahon
 College of Education, Criminal Justice, & Human Services
 School of Information Technology
 University of Cincinnati



ABSTRACT

Web application that allows multiple users to watch YouTube videos virtually with one another in real-time. According to the Pew Research Center organization, "for today's teens, friendships can start digitally; 57% of teens have met a new friend online...42% of all teens spend time with friends on messaging apps such as Kik and WhatsApp... 52% of all teens spend time with friends playing video games; 13% play with friends daily".

"Anywhere; Any Device"



Application Architecture Diagram

TECHNICAL ELEMENTS



POTENTIAL USERS

The potential users for the YouTube Theater are according to an article published by Tubular Lab on November 20, 2014 mid-to-late Millennials, males and females between the ages of 18 and 34 who are watching very different types of content on YouTube.

Multiple locations around the world where web application can be accessed



PROBLEM/SOLUTION

Many applications address the flexibility of communication and the exchange of media without speaking to issues like watching a video in synchrony. YT simulates the experience of watching a YouTube video in person with someone by displaying a live video-feed of the other connected party.

Acknowledgments

A special thanks to Professors Pat Kumpf, James Scott and Russell McMahon and our friends and family.