



ASKGC

Final Report

Authors:

Steve Jean¹

Kelsey Dramis²

Jordan Esty³

1. Email: sjean@loyola.edu
2. Email: kndramis@loyola.edu
3. Email: jresty@loyola.edu

Table of Contents

Requirements Specification

| | |
|--|-------|
| 1.0 Introduction | |
| 1.1 Overall Scope and Goal..... | 6 |
| 1.2 Stakeholders..... | 6 |
| 1.3 Scope..... | 6-8 |
| 2.0 Problem and the Problem Domain | |
| 2.1 Website Functionalities..... | 9 |
| 2.2 Database Functionalities..... | 9 |
| 2.3 High Level Diagrams..... | 10 |
| 2.4 Problem Domain..... | 10 |
| 3.0 Glossary | |
| 3.1 Ask Greycomm Forum Page..... | 10-11 |
| 3.2 Tutorial Page..... | 11 |
| 3.3 Frequently Asked Questions Page..... | 11 |
| 4.0 Requirements | |
| 4.1 Primary Actors..... | 11 |
| 4.2 List of Features..... | 12-13 |
| 4.3 Use Cases..... | 14-19 |
| 5.0 Technology Used | |
| 5.1 Technology Requirements..... | 20 |
| 6.0 Other Requirements | |
| 6.1 Performance..... | 20 |
| 6.2 Used Requirements..... | 20 |
| 6.3 Maintenance..... | 20 |

Plan of Development

| | |
|---|-------|
| 1.0 Introduction | |
| 1.1 Process Development..... | 21 |
| 1.2 Work Breakdown Structure..... | 22 |
| 2.0 Development Iterations | |
| 2.1 Phase 1: User Interface..... | 22-23 |
| 2.2 Phase 2: Database..... | 23 |
| 2.3 Phase 3: Forum..... | 23 |
| 3.0 Estimated Required Effort | |
| 3.1 Requirements Elicitation and Documentation..... | 24 |
| 3.2 Implementation..... | 24 |
| 3.3 Testing..... | 24 |

Software Design Specification

1.0 Introduction

| | |
|--|-------|
| 1.1 Overview of Design..... | 25 |
| 1.2 Major Constraints..... | 25 |
| 2.0 Data Design | |
| 2.1 Database description..... | 26 |
| 2.2 Internal Software Data Structure..... | 27 |
| 3.0 Architectural and Component-Level Design | |
| 3.1 Program Structure..... | 27 |
| 3.1.1 Architecture Diagram..... | 27 |
| 3.1.2 Alternatives..... | 27 |
| 3.2 Description for Component - Website..... | 28 |
| 3.2.1 Functional narrative for component..... | 28 |
| 3.2.2 Component Description..... | 28 |
| 3.3 Description for Component - Database..... | 28 |
| 3.3.1 Functional narrative for component..... | 28 |
| 3.3.2 Component Description..... | 29 |
| 3.4 Description for Component - iOS Application..... | 29 |
| 3.4.1 Functional narrative for component..... | 29 |
| 3.4.2 Component Description..... | 29 |
| 3.5 Software Interface Description..... | 30 |
| 3.5.1 External machine interface..... | 30 |
| 3.5.2 External system interfaces..... | 30 |
| 3.5.3 Human Interface..... | 30 |
| 4.0 User Interface Design | |
| 4.1 Description of the user interface..... | 30-32 |
| 4.1.1 Screen Images..... | 33-40 |
| 4.1.2 Objects and Actions..... | 41 |
| 4.2 Interface Design Rules..... | 41 |
| 5.0 Restrictions, Limitation, and Constraints | |
| 5.1 Overview..... | 41 |
| 6.0 Appendices | |
| 6.1 Requirements Traceability Matrix..... | 42 |
| Software Test Specification | |
| 1.0 Introduction | |
| 1.1 Goals and Objectives..... | 43 |
| 1.2 Test Cases..... | 43-45 |
| 2.0 Test Plan | |
| 2.1 Testing Strategy..... | 45 |
| 2.1.1 Unit Testing..... | 46 |
| 2.1.2 Integration Testing..... | 46 |

| | |
|---|-------|
| 2.1.3 High-order or System Testing..... | 46 |
| 2.2 Testing tools and environment..... | 47 |
| 2.3 Test Schedule..... | 47 |
| 3.0 Test Procedure | |
| 3.1 Unit Test cases - Websites..... | 47 |
| 3.2 Unit Test cases - Database..... | 47-48 |
| 3.3 Unit Test cases - Application..... | 48 |
| 3.4 Integration Testing: Test Cases & Result..... | 48 |
| 3.5 High-order Testing (System Testing)..... | 48 |
| 3.5.1 Performance Testing..... | 48 |
| 3.5.2 Security Testing..... | 49 |
| 3.5.3 Load/Stress Testing..... | 49 |
| 4.0 Testing Specific to web application | |
| 4.1 Content Testing..... | 49 |
| 4.2 Database Testing..... | 49 |
| 4.3 Compatibility Testing..... | 50 |
| 4.4 Usability Testing..... | 50 |
| 4.5 Navigation Testing..... | 50 |
| 4.6 Performance Testing..... | 51 |
| 4.7 Security Testing..... | 51 |
| The Guide for Future Maintenance | |
| 1.0 Introduction | |
| 1.1 Future Work..... | 51 |
| 1.1.1 Live Chat..... | 51-52 |
| 1.1.2 Privileged Users..... | 52 |
| 1.1.3 Validate Post..... | 52-53 |
| 1.1.4 Search bar..... | 53 |
| 1.1.5 FAQ Retrieval..... | 53 |
| 2.0 GitHub Link | |
| 2.1 AskGreycomm Maintenance..... | 54 |
| Reflections | |
| 1.0 Reflections on the Software Development Process | |
| 1.1 Kelsey's Reflection..... | 54-56 |
| 1.2 Steve's Reflection..... | 56-58 |
| 1.3 Jordan's Reflection..... | 59-60 |
| Size of Product | |
| 1.0 Product Metrics | |
| 1.1 Estimation of Size of Product..... | 60-61 |
| Testing Results | |

1.0 Results

| | |
|------------------------------------|-------|
| 1.1 Types of Test and Results..... | 61-63 |
| 1.2 Test Cases..... | 64 |

Table of Figures

| | |
|--|----|
| Figure 1: <i>Navigation/Activity Flow of the User</i> | 10 |
| Figure 2: <i>Work Breakdown Structure</i> | 22 |
| Figure 3: <i>ASKGC Database</i> | 26 |
| Figure 4: <i>Architecture Diagram of the Application</i> | 27 |
| Figure 5: <i>Signup Form</i> | 33 |
| Figure 6: <i>Login Form</i> | 33 |
| Figure 7: <i>Forgot Password</i> | 33 |
| Figure 8: <i>Reset Password</i> | 33 |
| Figure 9: <i>Change Password Form</i> | 34 |
| Figure 10: <i>Activate Account</i> | 34 |
| Figure 11: <i>Homepage</i> | 35 |
| Figure 12: <i>Forum Home Page</i> | 35 |
| Figure 13: <i>Questions in Category</i> | 36 |
| Figure 14: <i>Forum Category</i> | 36 |
| Figure 15: <i>Forum Questions</i> | 37 |
| Figure 16: <i>Post Reply</i> | 37 |
| Figure 17: <i>FAQs</i> | 38 |
| Figure 18: <i>Tutorials</i> | 38 |
| Figure 19: <i>Login Screen</i> | 39 |
| Figure 20: <i>Signup Screen</i> | 39 |
| Figure 21: <i>App Change Password</i> | 39 |
| Figure 22: <i>Homepage within App</i> | 39 |
| Figure 23: <i>Load Testing 1</i> | 62 |
| Figure 22: <i>Load Testing 2</i> | 62 |
| Figure 22: <i>Load Testing 3</i> | 63 |

Requirements Specification

1.0 Introduction

1.1 Overall Scope and Goal

The purpose of this project is to design website that would serve as a forum where the user would be able to find information on specific media and video editing related topics, as well as information related to Greycomm. A user must login to be able to post and answers questions. FAQ page will be created to keep track of popular questions. There will also be a tutorial page which will provide a basic overview of production techniques. We all will be creating an iOS App that will house the appropriate login functionalities and embed the website.

1.2 Stakeholders

- **The Customer:** (GreyComm) plans to use this as a resource to teach users about video production and other media related topics.
- **Loyola University Maryland:** The university owns GreyComm and all their resources.
- **Developers:** hope to provide a piece of quality software than has the ability to achieve all functional requirements and is easy to maintain.
- **Investors:** the people who want to see the service grow and are willing to provide funding or resources to improve the quality.
- **Students/Users:** the feedback from the people who will use the service will help increase usability. They will provide input that can greatly increase usage of the service.

1.3 Scope

Within our scope for this project includes creating secure logins so that the administrator will be able to keep track of the questions being asked. We will be including a database that is used to store the users and the categories and topics of questions being asked. Once a user asks a question, or creates a topic, they will wait for the question to be answered. If another user knows the answer, they will be able to reply. Otherwise, the administrator will reply to the question. The AskGreycomm homepage will **not** be visible to anyone that does not have an account. Once

an account is created, that user will receive a verification email, letting them know that an account under that email has been created.

In Scope

- Add a mobile app
- Creating secure user logins so that the administrator can keep track of all questions being asked
- Validate users email to know they are a real person
- Ability to post questions
- Frequently asked questions (FAQ) page so that user can gain a better understanding of how the forum operates. This would provide the ability to view the page and if the same question is asked enough then it will post to the page
- Having a database so that if a question was already asked the database can retrieve the answer
- Having a general tutorial page to help users start production projects
- A list of online users
- Administrators can ban spam users
- A contact us page/section

Out of Scope

- Unable to create Greycomm certification tests
- Unable to give user rewards for different levels of certification

Details of the Scope:

Login

- For an unsuccessful result, we expect the form to alert the user that the email and password combination is invalid
- For successful result, we expect the form to redirect you to the homepage

Registration

- We expect the system to make sure that both emails entered are the same and both password entered as the same.

- If the emails or passwords do not match, we expect the system to notify you which one or both do not match.
- We also expect the system to check if it is a valid email.
- Checks whether or not the user already exists based on the email provided.

Forgot Password

- First form on the forgot password page checks to see if the email is a valid email and exists in the system.

Reset Password

- We expect the system to check to make sure the new password that is entered is the same in both fields.
- We also expect the system to check the user's security questions and confirm that they are entered correctly.
- If the security questions are not answered correctly, we expect the system to tell the user that one or both of their answers were incorrect, to maintain a sense of security.

Change Password

- We expect the system to check to see if the email is valid and in the system.
- We expect the system to check to see if the current password entered matches what exists within the database.
- And finally checks to see if the two new passwords match.

Post a Question

- We expect the question to be put into the correct Subject and Category that the user has chosen based on the type of question they would like to ask.

Answer Question

- User chooses a subject that they would like to review the questions of. If they think they know the answer, they will be able to post a reply to that question, and that will be posted underneath the question.

2.0 Problem and the Problem Domain

2.1 Website Functionalities

The website we are developing is a question and answer forum related to video production with the ability to allow users to post questions, answers, links, and videos. A user must login to be able to post and answers questions. FAQ page will be created to keep track of popular questions. There will also be a tutorial page which will provide a basic overview of production techniques. In addition to the website, we will have an iOS App that has a clearer login interface, but includes the website embed into the inside of the App.

2.2 Database Functionalities

The database will keep track of all user data and information such as username and password and all questions, answers etc. related to that user. It will also keep track on if the account has successfully been activated. All of the accounts are linked to the forum, that is, if that user posts a question while they are logged in, it will appear under their name and also post that they asked the question.

2.3 High Level Diagrams

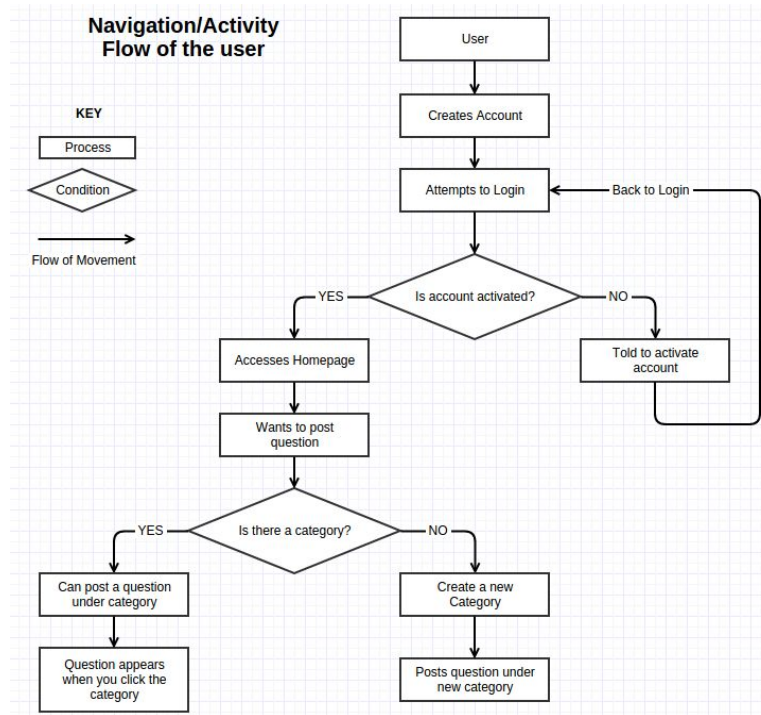


Figure 1: Navigation/Activity Flow of the User

2.4 Problem Domain

The website we are trying to create is supposed to be a school specific forum for the Loyola community with emphasis on Greycomm students. The forum is question and answer based and would allow users to learn more about video production, broadcast technology, and other related topics. Our overall goal for this project is to create a interface that can allow users to become more knowledgeable about media related technology and what GreyComm has to offer.

3.0 Glossary

3.1 Ask Greycomm Forum Page

The most recent post can be viewed on the forum page of the website. All active post will be viewable. A post becomes closed when either the person who asked the questions determines whether not their question has been answered or the Administrator deems the question answered.

If the administrator believes that the question is important enough to add to the FAQ Page, they will be able to do so.

3.2 Tutorial Page

A page with a basic starting tutorial will outline all the fundamental things needed to start learning about creating media.

3.3 Frequently Asked Questions Page

Will provided some useful information about the way the site works. This page will be comprised of all the frequently asked questions about the site. It will also provide the ability to ask new questions.

4.0 Requirements

4.1 Primary Actors

- Users
 - Sign up
 - Secure login
 - Activate account
 - Post questions
 - Post answers
 - View tutorials
- Administrator
 - Update the site
 - Provide tutorials
 - Post answers
 - Ban spam users
 - View all user statistics and history

4.2 List of Features

➤ **HIGH PRIORITY / FUNCTIONAL REQUIREMENTS**

- Add a mobile app with a simple web based interface
- To view the homepage and any pages within the website, the user will be required to login to the site
 - If they have not yet created a account the website will have the ability to allow the user to sign up / create an account (secure login).
 - While selecting a username the system should determine if that username is already in use and if it is they will have to choose another.
 - Check for valid username and password guidelines
 - Send validation email to the user, and not let them login until they have successfully activated their account
 - If they forgot their password will give them the ability to reset
- When the user signs in they have the ability to post and answer questions to the forum.
- Users can post videos and other content within the forum.
- Administrators will also have the ability to maintain the website.
- A frequently asked questions page with overview questions about how the site works.
- The use of tags and metadata to make it easier to find related information

➤ **LOW PRIORITY / NON-FUNCTIONAL REQUIREMENTS**

- A live chat room
 - user will be able to view a list of all online users
- Administrators and Moderators have the ability to ban users
- user can report other users
- a rating system for questions and answers
- Integration with google search results
- Provide information regarding the Greycomm equipment checkout process

- instructional guides on how to use the equipment
- About us page
 - brief overview about what the site was created for
- A contact us section
 - this would allow users to be able to communicate with the administrator

4.3 Use Cases

Use Case #1: User Signs Up for AskGreycomm

| | |
|-----------------------------|---|
| Actor | Loyola Student, new user |
| Description | Student arrive at website, needs to create account |
| Pre-conditions | The email is not in the system |
| Post conditions | The user has created an account, and has successfully activated their account |
| Priority | High - The user needs an account to access forum |
| Normal Course of Events | <ol style="list-style-type: none">1. Student goes to Greycomm website2. Student creates account3. Student activates account4. Student logs on to website5. Student accesses forum and asks question |
| Unsuccessful Vetting result | The person's email is not a valid email, and they are unable to activated their accoun. When creating account, they will see issues if they don't enter all the data, their passwords don't match, or their emails don't match. |
| Notes and issues | In order to access anything within the website, the user MUST have an account. |

Use Case #2: A User is Posting a Question to the Forum

| | |
|-----------------------------|---|
| Actor | A user who has an ASKGC account |
| Description | A user wants to ask a question related to video editing and film production, so wants to go to the forum. |
| Pre-conditions | Must have an account, and there must be a category for them to post their question under. |
| Post conditions | The question appears under the category they have chosen. |
| Priority | High - This is one of the main features of the website |
| Normal Course of Events | <ol style="list-style-type: none">1. Student logs onto the system.2. They access the forum and see there are available Categories to ask questions in.3. They post a question and choose which category to place it in.4. Their question is posted, and they wait for either the Admin or another user to answer their question. |
| Unsuccessful Vetting result | There is no category to post the question under or it did not appear. |
| Notes and issues | None |

Use Case #3: Loyola Student trying to Access Tutorials

| | |
|-----------------------------|---|
| Actor | Loyola Student, any user |
| Description | Student logs onto website, accesses the tutorials tab |
| Pre-conditions | The tutorial is not restricted |
| Post conditions | The user accesses the tutorials tab, seeing all available tutorials, and can learn about a selected topic |
| Priority | Medium - helpful when stuck |
| Normal Course of Events | <ol style="list-style-type: none">1. User gets onto website2. Clicks tutorial tab on main screen3. Brought to all available tutorials4. Can choose a tutorial to watch or read |
| Unsuccessful Vetting result | The video is unavailable or is not loading |
| Notes and issues | If a specific tutorial is unavailable, have option to contact administrators. |

Use Case #4: Create a Category

| | |
|-----------------------------|---|
| Actor | Anyone with ASKGC Credentials |
| Description | Student logs onto website, goes to the Forum homepage, and wants to add a new category of questions. |
| Pre-conditions | User must be logged onto the website to create a Category. |
| Post conditions | The category created must include a description. |
| Priority | Medium - categories help organize the questions being asked |
| Normal Course of Events | <ol style="list-style-type: none">1. User logs on to website2. User accesses the forum and clicks the "Create Category" button.3. User enters category name, and includes the description of the category.4. Category and description appear on the Forum home screen. |
| Unsuccessful Vetting result | Could not create category. |
| Notes and issues | None |

Use Case #5: Replying to a Question

| | |
|-----------------------------|---|
| Actor | Any user with ASKGC credentials |
| Description | A user thinks they can answer a question posted by another user, so they reply to the question they have posted. |
| Pre-conditions | Someone must have posted a question under a specific category. |
| Post conditions | The reply will appear underneath the original question, and it will include the information about the user and the time and date at which they replied. |
| Priority | Medium - People want their questions answered |
| Normal Course of Events | <ol style="list-style-type: none">1. User logs onto website and accesses forum.2. User looks in Category and sees that someone has asked a question.3. The user knows the answer, and replies to the question.4. The reply is posted under the question. |
| Unsuccessful Vetting result | The reply is not posted under the questions. |
| Notes and issues | None |

Use Case #5: Replying to a Question

| | |
|-----------------------------|---|
| Actor | Any user with ASKGC credentials |
| Description | A user thinks they can answer a question posted by another user, so they reply to the question they have posted. |
| Pre-conditions | Someone must have posted a question under a specific category. |
| Post conditions | The reply will appear underneath the original question, and it will include the information about the user and the time and date at which they replied. |
| Priority | Medium - People want their questions answered |
| Normal Course of Events | <ol style="list-style-type: none">1. User logs onto website and accesses forum.2. User looks in Category and sees that someone has asked a question.3. The user knows the answer, and replies to the question.4. The reply is posted under the question. |
| Unsuccessful Vetting result | The reply is not posted under the questions. |
| Notes and issues | None |

5.0 Technology Used

5.1 Technology Requirements

In order to use this software, the user will need a desktop computer or laptop in which they can access the website. When designing the backend of our software project, we will be using various languages. For instance, when designing the website, we will use languages such as PHP and HTML to program it to our liking. When working on the database, we will be using MYSQL. This web based application should be able to run on any type of operating system. In order to access our app, we have been designing the app to be able to run on an iPhone 6. When designing the app, we used XCode and implemented the functionalities using Swift.

6.0 Other Requirements

6.1 Performance

We would like our software to have enough data to cover all of the requirements that the customer wants, but we also want to avoid having too much data that it slows down the system. Additionally, when we need to pull information from the database, we want that to be quick and efficient.

6.2 User Requirements

The user will be presented with a clean and simple interface that is easy to use and quick to learn. The user will not have to spend a lot of time understanding what the website does and how it functions, and be able to become accustomed to it rather quickly. If the user needs to ask a question, that should be clearly labeled on the forum page. All operation and functionalities should be well maintained and labeled.

6.3 Maintenance

Our code should be easy enough to maintained by our customer, who has little experience with websites. We should all follow the same coding style, and always use javadoc comments when commenting our source code.

Plan of Development

1.0 Introduction

1.1 Process Development

Initially, the software development process that we plan on following for our Senior Project will be utilizing an iterative process to control our project design and workflow in order to get all of our tasks done in a reasonable amount of time. In doing so, we will be able to follow a traditional life cycle process which will include planning, designing, coding, and excessively testing our code. This will ensure that we have effectively tested and met all of the requirements our customer needs from us. With the use of user stories and well defined use cases, we will be able to tackle each requirement in a reasonable amount of time. Additionally, since we are a group of 3, we will utilize pair programming, which will enable us to work through problems together and more efficiently.

1.2 Work Breakdown Structure

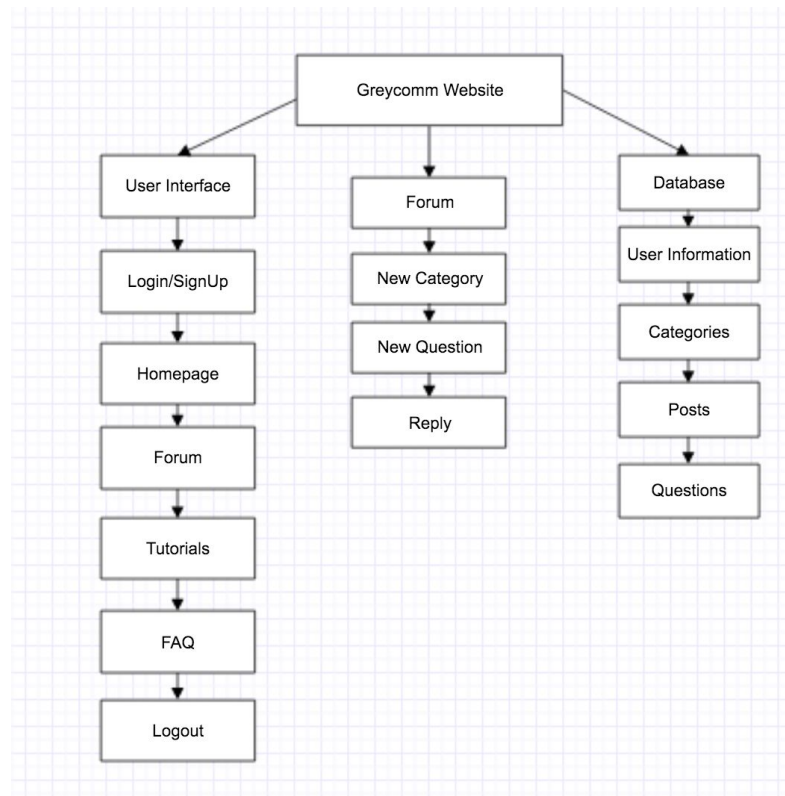


Figure 2: Work Breakdown Structure

2.0 Development Iterations

2.1 Phase 1: User Interface

- Login/Signup: When directed to the website, the user will first be asked to either login or create an account before able to access the Forum. Once the user signs up, they will receive a verification email to activate their account. The user also has the ability to change their password, or reset their password if they have forgotten it.
- Homepage: The user will be redirected to this screen once they have logged into their account. This screen offers you all of the navigation throughout the site, and presents a little information about Greycomm, as well as some contact information

- Forum: Users are able to create new categories, and provide descriptions of these categories they have made, and other users are able to answer other user's questions if they feel they have an understanding.
- Tutorials: The user will have access to tutorials to help them perform certain functions for editing, filmmaking, or general technology.
- FAQs: If a question is posted or asked enough times within the question forum, that question will be added to the FAQs to provide an easy answer to the users on the website.
- Log out: When the user has decided they want to logout of the site, they click the logout button and will be redirected back to the login/signup screen.

2.2 Phase 2: Database

- Holds Users: Provides a list of users that include information such as an id, username, password, first name, last name, security questions and answers, user type, user the date created an account, if their account is activated, and their activation key.
- Metadata Information: Questions and post will have metadata and tags to make information more accessible for the database and will make information retrieval quicker.
- Category: This information includes the category name and the category description, and then will be displayed on the homepage of the forum.
- Questions: This table in the database include the main question topic, then which category the user has specified it should be in, and who posted the question.
- Posts: This table in the database holds the post content that is linked to the question. This also holds the reply information, and who the reply is from.

2.3 Phase 3: Forum

- Post Question: The user is able to post a question to the forum regarding a functionality that Greycomm provides, or some type of technology functionality they have questions on.
- Answer Question: If you are knowledgeable on the topic, the user has the ability to answer a posted question.
- Create a category: If someone needs help on something that has never been asked, they can create a new category and post a question relating to that category.

3.0 Estimated Required Effort

3.1 Requirements Elicitation and Documentation

We all believe that we should work equally on the requirements elicitation because it is an overall very good skill to have. Most programmers think that documentation is not important in regards to Software Projects, but it is the one of the fundamental reasons Software Project's work. Without this skill, software projects may fail. **Estimated Time for Documentation:** 20 hours total.

3.2 Implementation

When working on our software, we will together work on the backend functionalities such as the database and the backend of the website. When it comes to designing the look and feel of the website, that will be split up among members. Since we do not have much previous knowledge on websites and web content management systems, it will be a learning process together.

Estimated Time for total implementation: 10 hours a week and an estimated 100 hours overall. The total estimated **lines of code** would be about 2,000-3,000 between the website and the app.

3.3 Testing

Testing will be done along the way by the individual programmers, but additional JUnit Testing will be completed towards the end of the semester. **Estimated Time for total Testing:** 3 hours a week and an estimated 30 hours overall.

Software Design Specification

1.0 Introduction

1.1 Overview of Design

Our design for the AskGreycomm project includes a variety of platforms it is working with. On the front end, there is a HTML, PHP, Javascript, and CSS documents to create the website. We used HTML to do the look and feel of the interface, PHP to connect to the database, and the CSS documents to design an aesthetically pleasing interface. There are 2 small javascript files that are needed for the search bar on the navigation toolbar. In order for the user to view the website, there is a need to create an account and store that account, which is where our database becomes involved. Our database, called “askgc” will hold the account information of each user, as well as the information stored for the forum. For the app portion of our software, we have designed it to be an iOS app that was designed to work on an iPhone 6. We used swift to implement the app, and have the simulator live on XCode.

1.2 Major Constraints

Since we are acting as the admin while we are creating the application, we do not have many constraints. We are setting up all of the accounts and creation so that they can be modified for future work. Our customer gave us complete freedom on how to design our product, but just advised us to learn more about their work, and to use their color scheme when designing. One additional constraint would be that we would have to obtain a Mobile App License in order to make the app go live, which our customer may be willing to cover, if he finds that the app is crucial and meets his needs.

2.0 Data Design

2.1 Database description

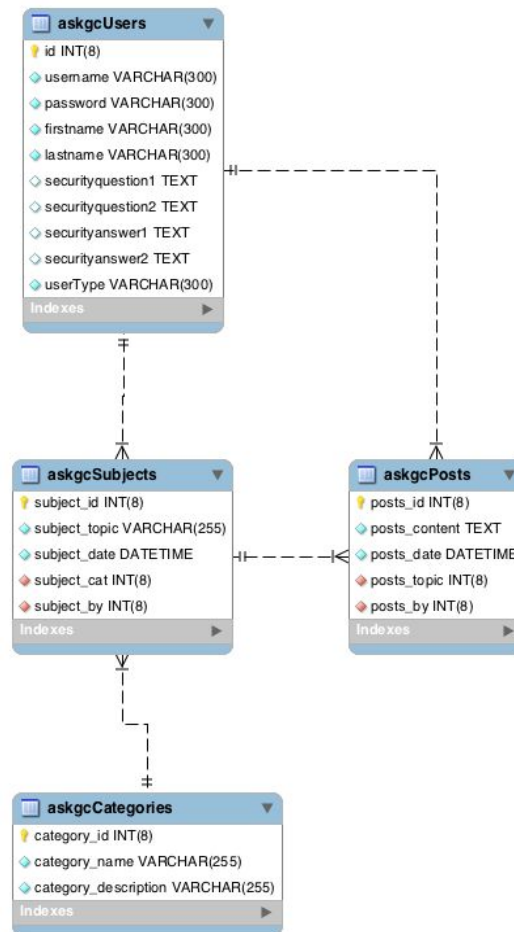


Figure 3: ASKGC Database

Our database has connections between them that are one-to-many in a few cases. The database relationship includes a user who can have many questions(subjects) and many posts. Categories can contain many questions. Questions(subjects) can have many posts related to them.

2.2 Internal Software Data Structure

When our MySQL database returns the results of the query that was sent to it, the results will be passed along through the built in data structures within our ASKGC Database.

3.0 Architectural and Component-Level Design

3.1 Program Structure

Our architecture is surrounded by a database using MySQL. Most of the front end of the website will interact with the database, as well as the iOS app. In order to use the ASKGC website and app, the user will be required to either create an account or login to the site using their credentials. Additionally, the forum questions and answers will be stored within the database, and that information will be retrieved.

3.1.1 Architecture Diagram

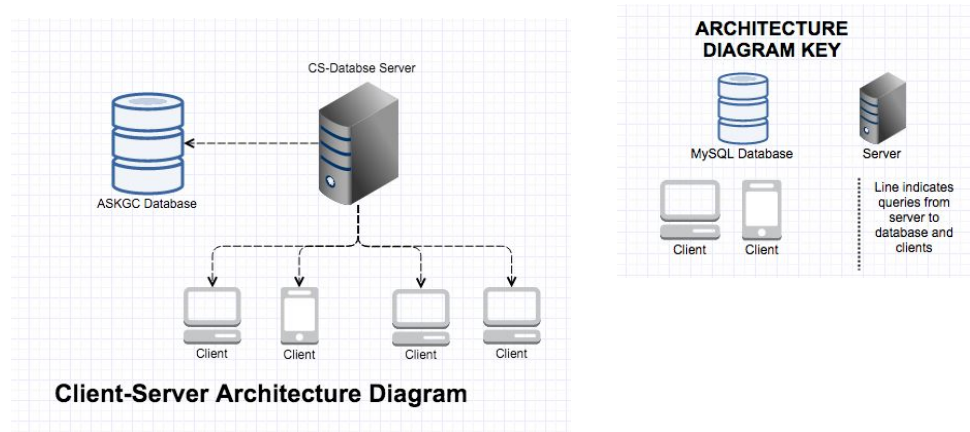


Figure 4: Architecture Diagram of the Application

3.1.2 Alternatives

Originally we had thought we would be doing software design based on component engineering, using a variety of software components rather than just relying on a web service, and our database. We used our web service and database to fully design our software for Greycomm Services, so we are working with a Client Server Architecture.

3.2 Description for Component - Website

3.2.1 Functional narrative for component

The website and app will function as a homepage and forum style application for Greycomm at Loyola University Maryland. They must create an account and log in to the website/app in order to access the homepage and everything else within it. In order to ask and answer questions, they will have to be monitored by the admin of the website, who will be our customer Jay Dunmore.

3.2.2 Component Description

The user will input their information on the login or signup form and be allowed access to the homepage for Ask Greycomm. Once signed up and registered within the database, the user will receive a confirmation email saying that they have registered. Additionally, if they change their password within the system, the user will receive a confirmation email. They will be informed that they are the correct user that is signed onto the system while on the homepage. It will be known when their question posted on the forum is validated.

3.3 Description for Component - Database

3.3.1 Functional narrative for component

Our database is called ASKGC. Within the database, we have four tables holding information that is coming from the website on the front end. One table holds the user's account information, including the name, email, password, and security questions. To keep the information safe, the security questions are a must. The database also includes tables for the topics, categories, and posts for the forum. Each table holds information about the title, information, and id of the called on forum information.

3.3.2 Component Description

Data is entered in the ASKGC database when an account is created. The information can then be altered if the user wants to change their password. You can double check the database and confirm that in fact the password has been changed on the table. Also, when a question is asked on the forum, we are able to check to see if the question was successfully added into the specific table for the forum. You can also perform those same operations for a new topic as well as a new category in the forum.

3.4 Description for Component - iOS Application

3.4.1 Functional narrative for component

The iOS application will function as a more convenient way of being able to access our forum based interface. The site and the application will have most of the same functionalities. Account activation will occur through the web for both the application and the website. The only difference between the two is that you can only access forgot password through the site. The user still has the ability to change password through the application.

3.4.2 Component Description

The user will input their login credentials into the application and be allowed access to the Ask Greycomm homepage if they have entered valid input. The web pages are embedded into the application's interface. We have restricted the ability to freely navigate to any page on the web. The user interface is created in such a way that, the user must remain within the boundaries of the website. We have implemented a back button and a refresh button for convenience.

3.5 Software Interface Description

3.5.1 External machine interfaces

Our website and app will be running with a machine running the MySQL database in the background. Currently, the database is connected to a Linux machine on the cs-database network, and will be housed there until otherwise noted.

3.5.2 External system interfaces

The database server is on cs-database.cs.loyola.edu and can be reached from any IP on the 144.126.12 subnet.

3.5.3 Human Interface

Our software is available to anyone who has internet access and a valid email address.

Details are provided in the next section. Our website will be available to anyone using any type of Internet Browser.

4.0 User Interface Design

4.1 Description of the user interface

When designing our user interface, we have incorporated a large amount design choices based on popular websites. We designed our website with a large amount of learnability in mind, making sure the user who is trying to use this application knows exactly what they are doing, and do not need to be taught to use the website. The website will be accessible to anyone with a valid email address. Also, the website does not bombard you with information, as we have broken up sections into tabs for a better way to focus the user's attention.

Sign Up Form

- Allows the user to enter all of their credentials which then allows them to be stored into the database as needed. They will know exactly what to enter in each required field.

Login Form

- Once the account has been created, they will be able to login to the system using their credentials.

Forgot Password Form

- If the user has forgot their password, they will be asked to enter their email address. If they do not have an account in the system, they will not be redirected to the “Reset Password” Form. If they have an account in the system, they will be directed to the “Reset Password” Form.

Reset Password Form

- Once the user has entered their email address, they are redirected to this screen and told to enter a new password, as well as confirm their security questions to make sure it is completely safe.

Change Password Form

- If the user wishes to change their password, they must enter their current password in order to change it.

Homepage

- Once logged onto the system, they will enter the homepage where they have access to the various tabs within the website.

Forum: Home

- A user is able to see all of the categories on this page with their descriptions, and they are able to click the category, and see the questions asked within.

Forum: Create Category

- A user is able to create a category, and is usually paired with a small description about the category, and is then posted onto the “Forum Home” link.

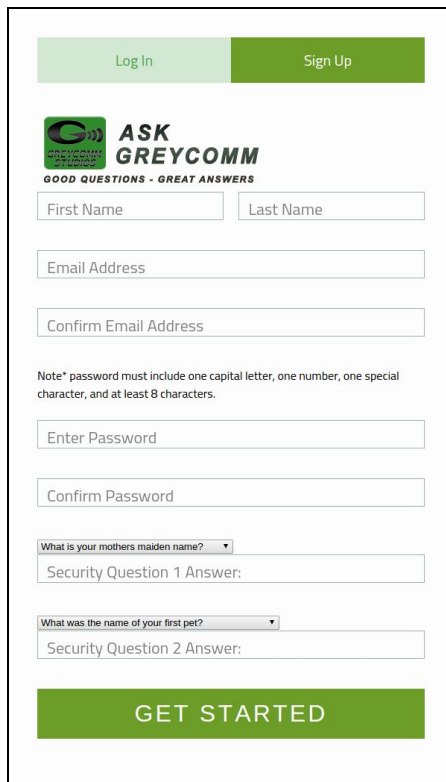
Forum: Post Question

- If a user wants to post a question, they will post the header question, choose the category they want the question to go under, and then will add a description for specifically what they are asking. The question will appear under the Category page. More specifically, the user will be able to see who asked the question, and what time and date it was asked.

Forum: Reply to a Question

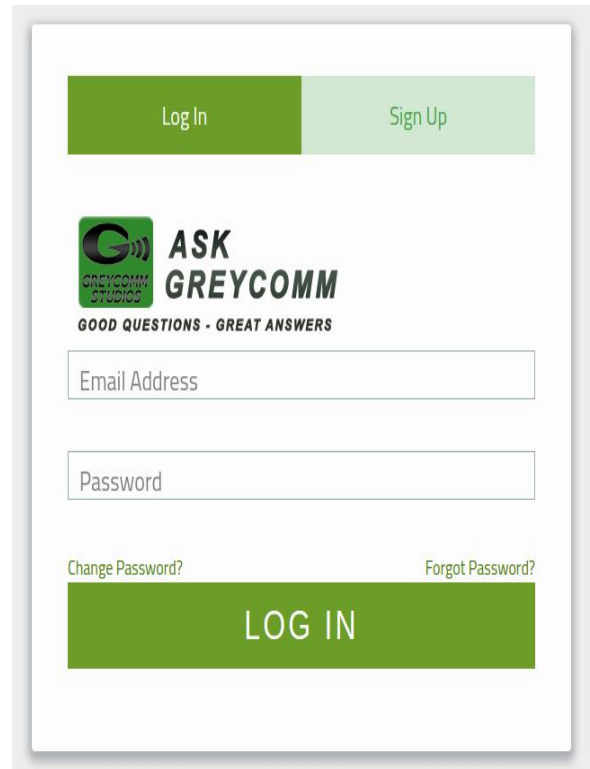
- Once a question has been asked within a category, a user is able to reply to that question if they think they know how to help them.

4.1.1 Screen Images



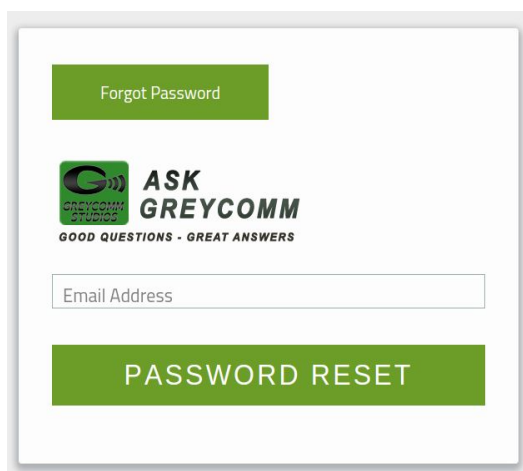
The Signup Form features a header with 'Log In' and 'Sign Up' buttons. Below the 'ASK GREYCOMM' logo and tagline, it includes input fields for 'First Name', 'Last Name', 'Email Address', and 'Confirm Email Address'. A password requirement note is provided, followed by 'Enter Password' and 'Confirm Password' fields. Two security questions are included with dropdown menus for selection and text boxes for answers. A large green 'GET STARTED' button is at the bottom.

Figure 5: Signup Form



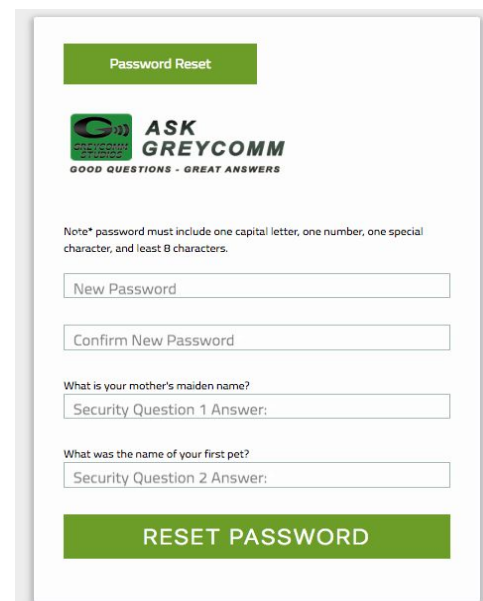
The Login Form has a header with 'Log In' and 'Sign Up' buttons. It features the 'ASK GREYCOMM' logo and tagline. Below are input fields for 'Email Address' and 'Password'. Links for 'Change Password?' and 'Forgot Password?' are positioned above a large green 'LOG IN' button.

Figure 6: Login Form



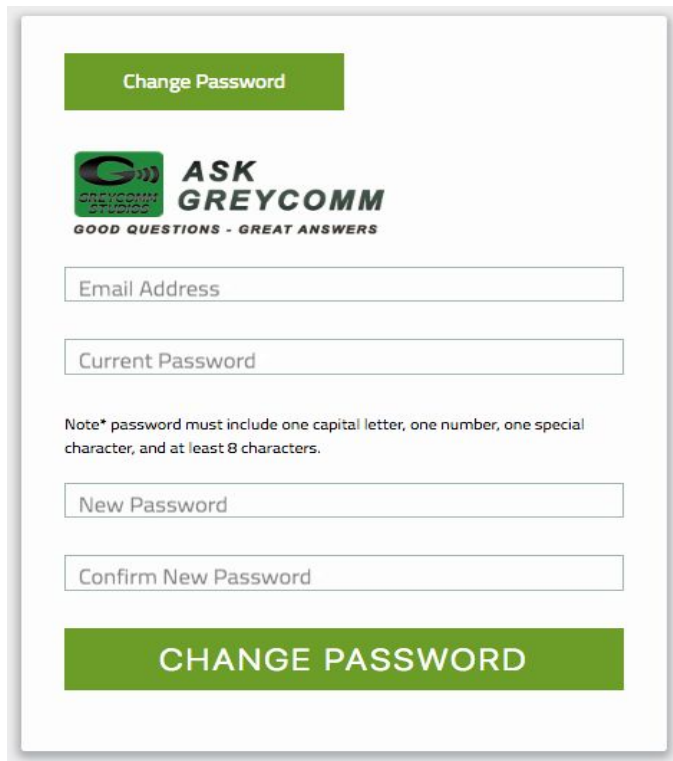
The Forgot Password form includes a 'Forgot Password' button at the top. It displays the 'ASK GREYCOMM' logo and tagline. A single input field for 'Email Address' is present, followed by a large green 'PASSWORD RESET' button.

Figure 7: Forgot Password




The Reset Password form starts with a 'Password Reset' button. It features the 'ASK GREYCOMM' logo and tagline, followed by a password requirement note. Input fields for 'New Password', 'Confirm New Password', and two security questions are included. A large green 'RESET PASSWORD' button is at the bottom.

Figure 8: Reset Password Form



The 'Change Password' form features a green header bar with the text 'Change Password'. Below this is the Ask Greycomm logo, which includes a green square with a white 'G' and the text 'ASK GREYCOMM' and 'GOOD QUESTIONS - GREAT ANSWERS'. The form contains four input fields: 'Email Address', 'Current Password', 'New Password', and 'Confirm New Password'. A note specifies that the password must include one capital letter, one number, one special character, and at least 8 characters. A large green button at the bottom is labeled 'CHANGE PASSWORD'.

Change Password

 **ASK GREYCOMM**
GOOD QUESTIONS - GREAT ANSWERS

Email Address

Current Password

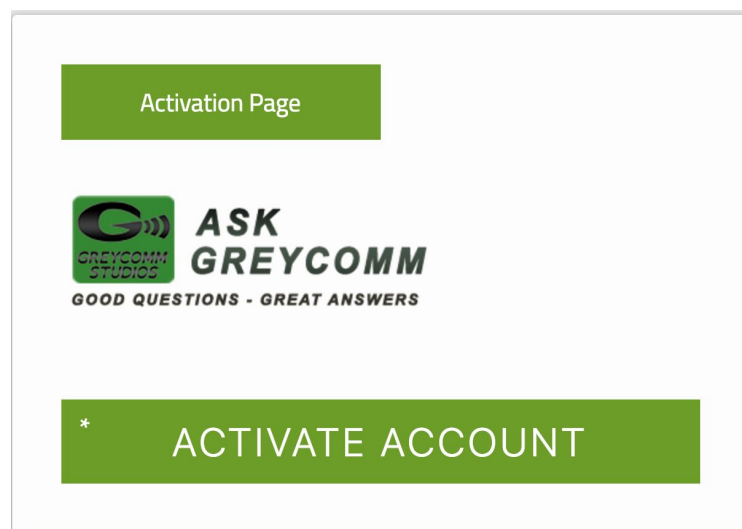
Note* password must include one capital letter, one number, one special character, and at least 8 characters.

New Password

Confirm New Password


CHANGE PASSWORD

Figure 9: Change Password Form



The 'Activation Page' has a green header bar with the text 'Activation Page'. It features the Ask Greycomm logo, consisting of a green square with a white 'G' and the text 'ASK GREYCOMM' and 'GOOD QUESTIONS - GREAT ANSWERS'. A large green button at the bottom is labeled '* ACTIVATE ACCOUNT'.

Activation Page

 **ASK GREYCOMM**
GOOD QUESTIONS - GREAT ANSWERS

* **ACTIVATE ACCOUNT**

Figure 10: Active Account

Website Interface:

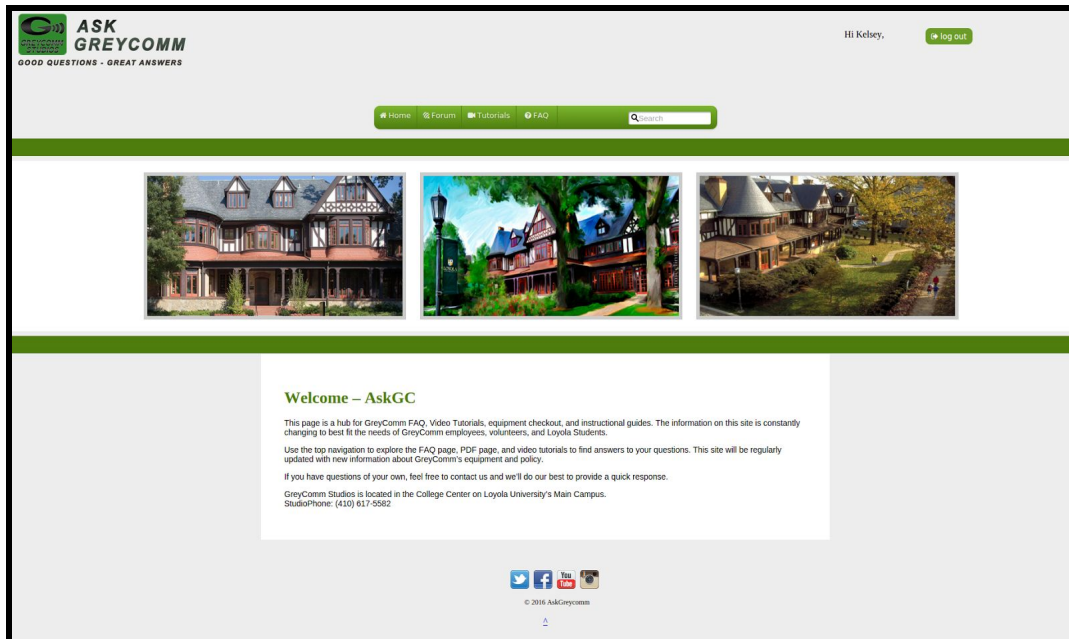


Figure 11: Homepage

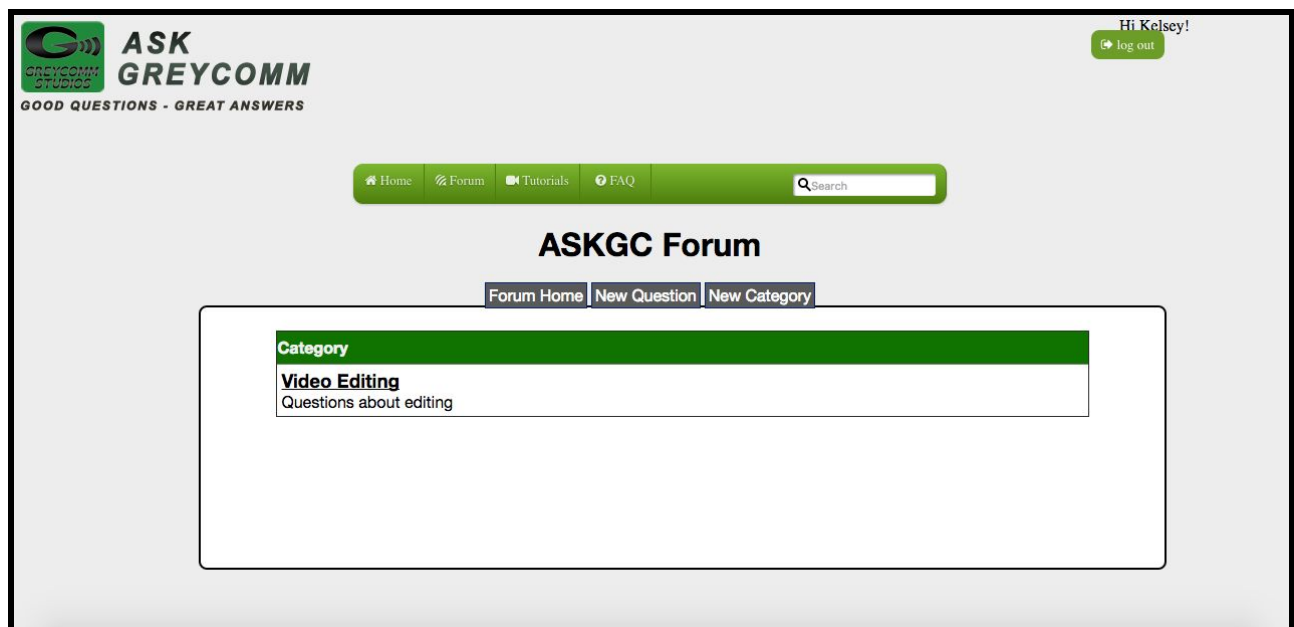


Figure 12: Forum Home Page

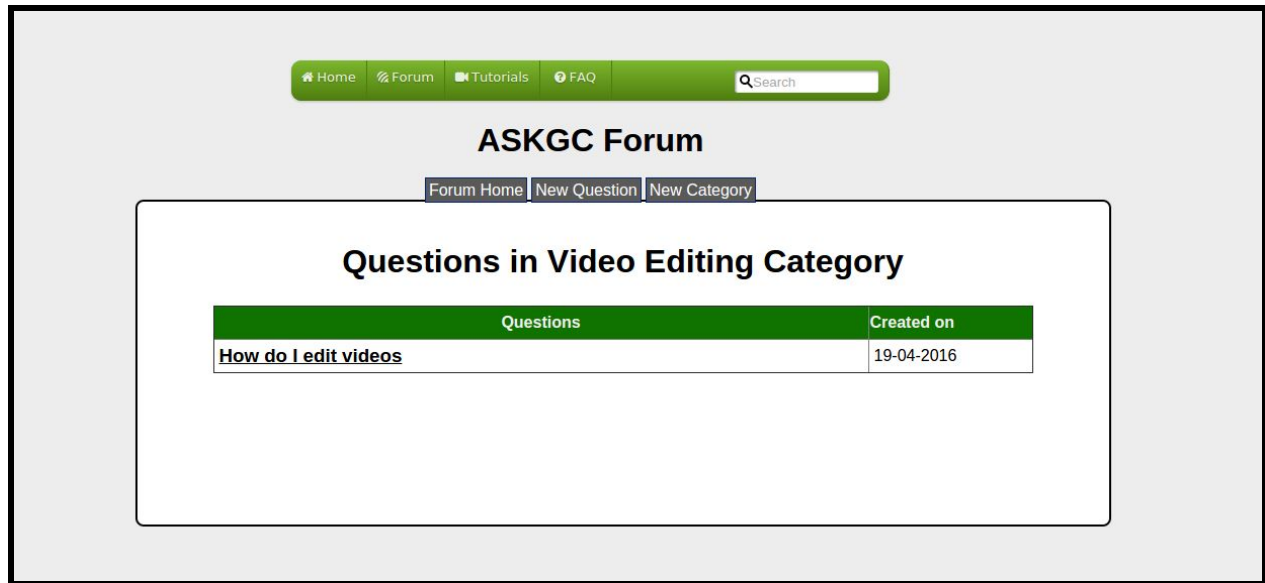


Figure 13: Questions in Category

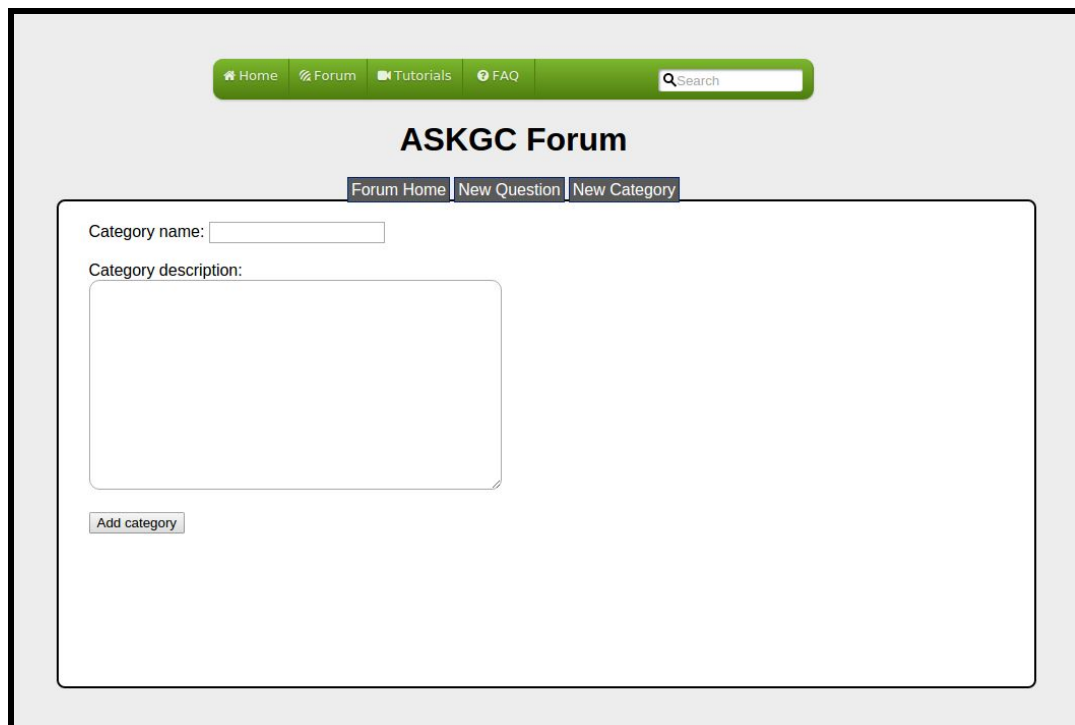


Figure 14: Forum Category

Home Forum Tutorials FAQ Search

ASKGC Forum

Forum Home | New Question | New Category

Question:

Category: Video Editing ▼

Description:

Post Question

Figure 15: Forum Questions

Home Forum Tutorials FAQ Search

ASKGC Forum

Forum Home | New Question | New Category

Posts related to "How do I edit videos"

| Post | Created By |
|-------------|---------------------------------------|
| i dont know | Kelsey Dramis on: 2016-04-26 21:02:33 |

Post a reply

Submit reply

Figure 16: Post Reply

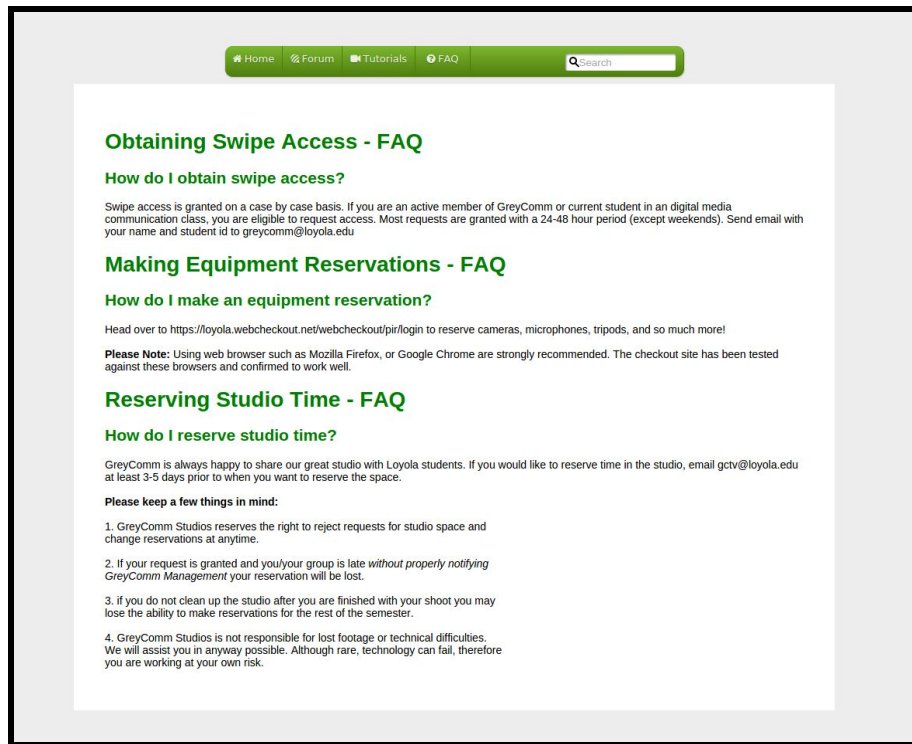


Figure 17: FAQs

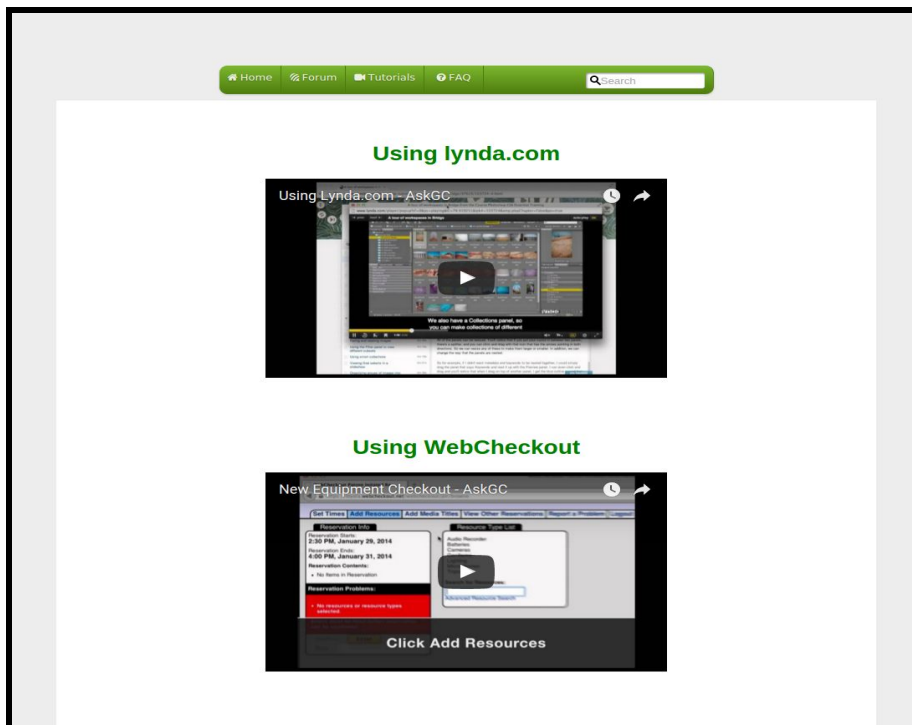
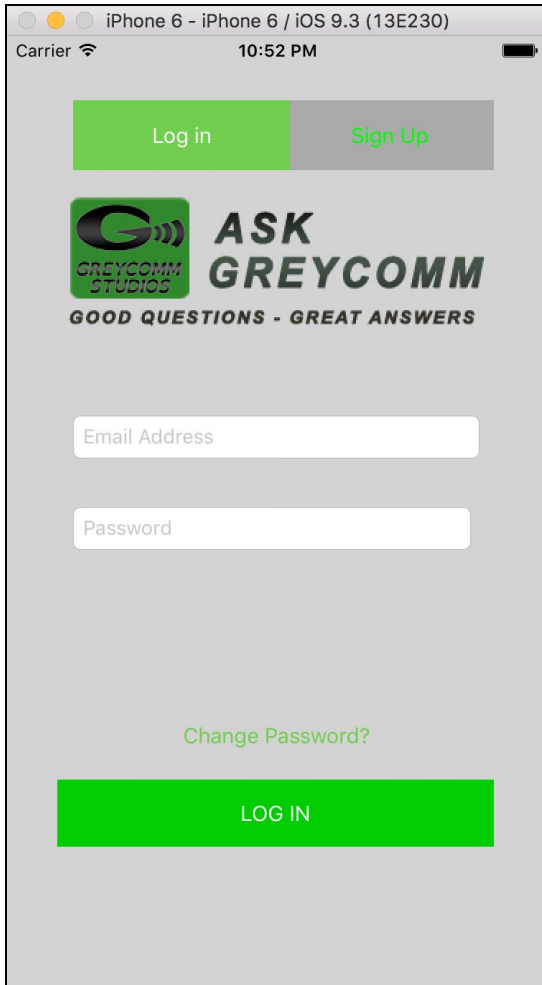


Figure 18: Tutorials

Application Interface



iPhone 6 - iPhone 6 / iOS 9.3 (13E230)
Carrier 10:52 PM

Log in Sign Up

ASK GREYCOMM
GOOD QUESTIONS - GREAT ANSWERS

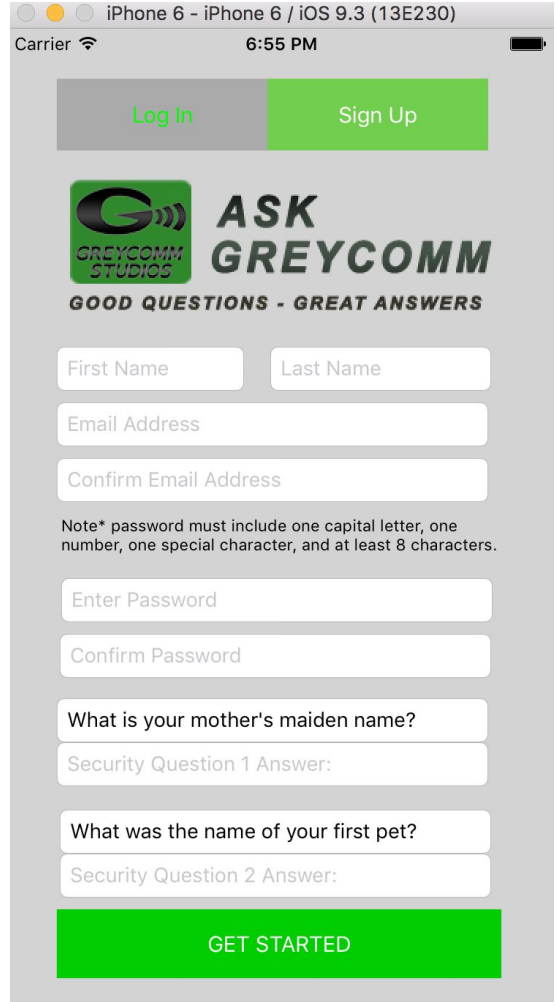
Email Address

Password

Change Password?

LOG IN

Figure 19: Login Screen



iPhone 6 - iPhone 6 / iOS 9.3 (13E230)
Carrier 6:55 PM

Log In Sign Up

ASK GREYCOMM
GOOD QUESTIONS - GREAT ANSWERS

First Name Last Name

Email Address

Confirm Email Address

Note* password must include one capital letter, one number, one special character, and at least 8 characters.

Enter Password

Confirm Password

What is your mother's maiden name?

Security Question 1 Answer:

What was the name of your first pet?

Security Question 2 Answer:

GET STARTED

Figure 20: Signup Screen

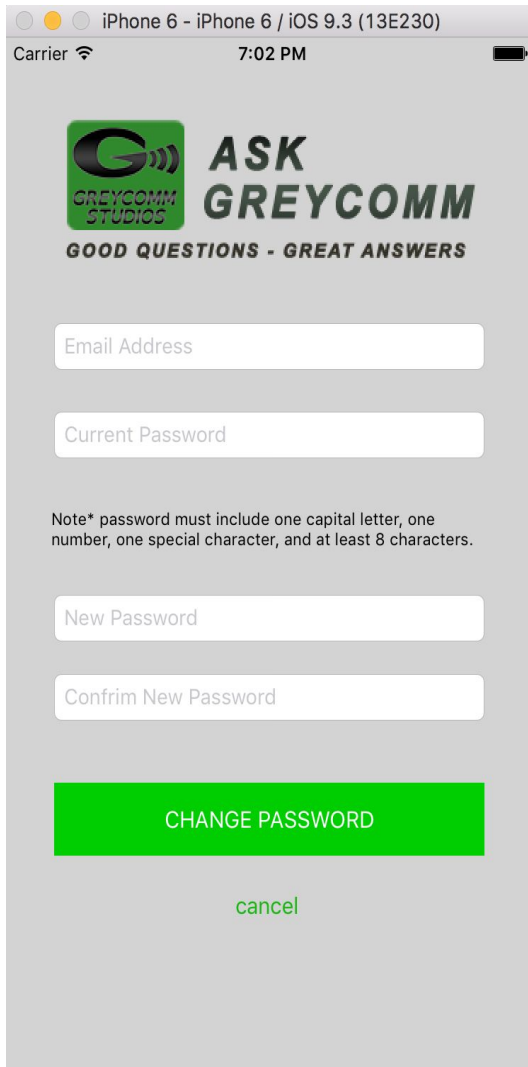


Figure 21: App Change Password



Figure 22: Homepage within App

4.1.2 Objects and Actions

The main objective of all of these pages is to maintain a secure website/app for AskGreycomm. Without the ability to reset your password and have a password protected website, they would not be able to have a trustful forum website. Once the user is logged onto the system, they can access everything from videos, tutorials, forums, and a list of Frequently Asked Questions. With access to the website, they will be able to become very knowledgeable on Video Production.

4.2 Interface Design Rules

We are following the color guidelines based upon our customer's request. His only requests were to follow the color scheme of Loyola University Maryland.

5.0 Restrictions, Limitations, and Constraints

5.1 Overview

Information will be stored in the MySQL Database, since information is being retrieved from the database, the server for database must be up and running to allow access to the website.

6.0 Appendices

6.1 Requirements Traceability Matrix

| Requirement | Functional Requirement | Use Case Name & ID |
|--|--|---|
| Need to have a database in the background to store the information from each question. | The user wants to sign up for ASKGC and access the forum. | #1: User Signs up for AskGreycomm |
| Must have an account to view pages, so the user must be stored in the database. | The user can access the tutorials and any subsequent information on the website once logged on. | #3: Loyola Student trying to Access Tutorials |
| There must be a category available to post the question under. | User wants to post a question to the forum and find out more information about something such as film editing and film production. | #2: A user is posting a question to the forum |
| The user must have an account to view the forum page. | The user wants to ask a question in a specific category, but that category doesn't exist. | #4: Create a category |
| There must be a question that the user wants to reply to. | If a user knows how to answer another student's question, they can reply. | #5: Replying to a question |

Software Test Specification

1.0 Introduction

1.1 Goals and Objectives

For our software, we will be using a variety of testing platforms to make sure everything works as intended. For each of the available functionalities of our website and database, we know what the expected output is for certain instructions given to it. For example, if we are creating an account, the hope is that the account is then added into the database and stored properly. Our hope for testing is ensuring that all of the functionalities are working properly. There is a long list of functionalities that we'll be testing, which is listed below in the statement of scope category. Before we turn in a working piece of software to our customer, we hope to make sure there are no major or minor bugs within the system.

1.2 Test Cases

| Piece of Software to Test | Description | Expected Result |
|----------------------------------|---|---|
| Login functionalities Test #1 | Check to see if the user can successfully login to the website. | <ol style="list-style-type: none">1. Successful - They are presented with the homepage.2. Failure - If they do not enter required information, they will be told "All data is required."3. Failure - If the email/password combination is incorrect, they will be told.4. Failure - If your account is not activated, they will be instructed to activate their account. |
| Registration Test #2 | User enters all of the required information (first name, last name, email, password, and security questions). | <ol style="list-style-type: none">1. Successful - They have successfully created their account.2. Failure - All data is |

| | | |
|----------------------------|---|--|
| | Information is stored in the database, and they are sent a confirmation email to activate their account. | <p>required to create account.</p> <ol style="list-style-type: none"> 3. Failure - Email is already in use. 4. Failure - Email/Password do not match (confirm). 5. Failure - Not a valid email. |
| Forgot Password Test #3 | If the user has forgotten their password and they wish to change it, they can go to the "Forgot Password" form and enter their email. | <ol style="list-style-type: none"> 1. Successful - Their email exists and they are presented with the reset password form. 2. Failure - That email does not exist in the system. 3. Failure - The email is not a real email. 4. Failure - The email is not activated. |
| Reset Password Test #4 | Once their email has been checked, the user must enter new password to change it to, and answer their security questions. | <ol style="list-style-type: none"> 1. Success - All data is correct and the password has been changed successfully and they are brought back to the login page. 2. Failure - If one of their security questions is not correct, they will be notified. 3. Failure - If the passwords do not match, they will be notified. 4. Failure - All data is required. |
| Change Password Test #5 | If the user knows their current password, but want to change it, they will go to this form. | <ol style="list-style-type: none"> 1. Success - They have entered all the correct information and their password has changed, and brought it back to the login page. 2. Failure - You must fill out all the required |

| | | |
|-------------------------------------|--|--|
| | | information. 3. Failure - Checks to make sure the email is valid and in the system. 4. Failure - If the current password is invalid. 5. Failure - Passwords entered do not match. |
| Post a Question to forum Test #6 | If a category exists in the forum, the user can post a question within that category. | 1. Success - Question and description are successfully posted within the Category. 2. Failure - No category exists. 3. Failure - Could not post your question. |
| Reply to a Question Test #7 | If there is a question that another user can reply to, they reply and it is posted beneath the question. | 1. Success - Reply is posted beneath question. 2. Failure - There is no question to reply to. 3. Failure - Could not post your reply. |
| Create a Category Test #8 | If there is no category for something specific the user wants to ask, create one. | 1. Success - Category and description are posted on the forum home page. 2. Failure - Could not post the Category. |

2.0 Test Plan

2.1 Testing Strategy

We will be using a mixture of both black box and white box testing. We are going to know what both the system and the code look like, so we will be looking for the correct output when a certain operation is performed.

2.1.1 Unit Testing

Different units for our design include testing the different forms that are used within the website and app. We will be testing each form to make sure the correct functionality occurs, such as sending emails to confirm a change of password, and looking within the database to ensure that a user was properly created. Since many of the operations within the forms could fail, we must check each form individually to make sure they do exactly what they are supposed to do. For example, if we request to change the password, we should see the password changed within the database, and the user should receive the email notifying them of this change.

2.1.2 Integration Testing

We have three separate components for our software: the website, the iOS app, and the database. In order for them all to work as one, we must test them together. Without the database in the backend, the users would not be able to gain access to the ASKGC Homepage and any of its functionalities. Our customer wants the secure login and wants to know who has an account, so the database is crucial for a working system. To test this, we will be creating a multitude of fake accounts to ensure that all of the information is being stored correctly within the database tables. We can additionally test the interaction of the two components by logging on with these different accounts to ensure they work correctly.

2.1.3 High-order or System Testing

Similarly to our integration testing, we will be testing the main functionality of the system by creating accounts, logging in, changing password, reset password, post a question to the forum, reply to a question on the forum, and so on. We will be testing this functionalities hands on, and hoping to see the expected output, on the correct inputs within the database. Our software is very easy to use and easy to maintain. Many of our files are easy to read, so any future work that needs to be completed should have no issues.

2.2 Testing tools and environment

We have chosen to use the web testing tool called “Doit: Simple Web Application Testing.” For our project, we feel that this service will best enhance our ability to test all of the functionalities of the software. It will test our forms, and check if they are being stored into the database file. This testing platform is a scripting tool and language for testing web applications that use forms. It uses a console based web client tool to receive requests and responses.

2.3 Test Schedule

| | |
|----------------------------------|-----------------|
| Unit Tests (testing each forms) | April 25th |
| Integration Testing | April 26th-27th |
| System Testing | April 28th |
| DoIt Testing | April 29th-30th |
| XCode built-in testing framework | April 30-May1 |

3.0 Test Procedure

3.1 Unit Test cases - Website

All of the values entered into the website will be of type string. Most of code was written using HTML and PHP, with some additional MySQL and supporting CSS code and Javascript for our design. We will be testing by checking the strings that have been entered into the system. We expect the system to return different cases for each, depending on if the values were entered as what is expected.

3.2 Unit Test cases - Database

When the information of type string is entered into the website, we are expecting that the information is then stored into the database properly. We have made a handful of tables, which include all of the information about the existing forum as well as the existing users in the system. We expect the database to update when a user changes their password. We also expect the

database to ensure that if someone is not entered into the database, they will not have access to any of the pages within.

3.3 Unit Test cases - Application

We will be testing our application very similarly to the way we have tested our website. We used all of our affiliate PHP and HTML cases, but only made minor changes to them. We developed the App in XCode, using swift and will be using the integrated testing features that XCode provides to us.

3.4 Integration Testing: Test Cases & Result

We plan to write test cases by creating many accounts to ensure that our code for the website is being read in properly, and ensure that the MySQL queries are being read correctly and correctly stored into the database. We will also be testing to ensure that there is a proper creation of Categories by the administrator, and that the existing subjects and posts under the respective category are being stored properly in the subsequent database tables. Additionally, we will be testing the account creation and additional features that are dealing with the accounts to make sure the queries are correct and work as intended.

3.5 High-order Testing (System Testing)

We might consider the use of beta testers so that we can properly determine the system's capabilities. We will have several users register accounts and check to see if they can all register, login, change their password and reset their password. We also want to determine if posts can be asked and answered properly and that they display which user asked and answered each.

3.5.1 Performance Testing

We want to have as many users on the system as we possibly can and then note whether or not the system slows down with all those users interacting with it. Theoretically, we could have everyone in the Linux lab try and register at the same time, or all try to post a reply to a question at the same time and see how it affects the system. We could check to see how long the database takes to return a specific query at that time as well.

3.5.2 Security Testing

Password

The password of each user is suppose to meet minimum set of requirements. There must be at least one capital letter, one number, one special character, and a minimum of eight characters.

3.5.3 Load/Stress Testing

If they don't have an active session they should not be allowed to view the internal pages of the site. We want to try and get as many users as we can, so that we can perform a stress test of the system. This kind of test will let us know how many users can interact with the site without it crashing. It will also allow the test for performance, robustness, security, and load capacity.

4.0 Testing Specific to web application

4.1 Content Testing

<http://validator.w3.org>

We will use the website listed above to test the content.

4.2 Database Testing

The queries we'll use to test will return true or false if the event was successful. We query the database for creating a new user and check to see if that user was successfully added. We check to see if the user can login by verifying they they exist in the system. We can then verify that they can change their password based on their login info in the database. We determine if they can reset your password if they forget by the use of security questions queried from the database. Active sessions are created upon login based on a query as well.

4.3 Compatibility Testing

We will be testing our web application on a variety of different browsers including Google Chrome, Firefox, Safari, and Internet Explorer. We will also be checking to see how the web application behaves throughout this entire process, we have been checking our site on these different browsers periodically to ensure that everything works as intended. We have found a few errors this way, and have been able to fix them quite easily and quickly. Additionally, we have been working to make sure the overall design of our system looks compatible on different browsers as well as different screen sizes. We have found in our design that working in the Linux Lab versus working on our laptops creates some issues on how the appearance looks. Throughout the process, we have also been testing our website on both Linux and OSX platforms. We will also be testing on Windows to ensure compatibility.

4.4 Usability Testing

To conduct usability testing would be extremely beneficial for our project. To do so, we will be conducting something similar to a user study for our application. We will ask a variety of users to test out our application by asking them to do specific tasks on our website such as create an account, login, post a question, or reply to a question. If they find that the website is easy to use and easy to understand, we will be led to believe that it is user friendly.

4.5 Navigation Testing

If the user is not logged in, they will not be able to view the homepage, forum, tutorials, and FAQ page. Once a user is logged on, they will be redirected to the homepage. If the user attempts to directly access the homepage and the link, they will be redirected to the login screen. The register form will also redirect you to the homepage upon successful registration. From the login page, if the user clicks the “Forgot Password” form, they will be redirected to the password reset page. From there, they will be redirected again to the reset password page. If the user tries to access this, without going through the forgot password page they will be redirected to the login page.

4.6 Performance Testing

We will test performance by determining whether or not the server response time degrades based on the number of users. Then we will figure a way to time the response of each user request to see how long it took for each action to be performed and note if there is a significant difference in time based on number of users. After that we will then calculate the average time of each type of request.

4.7 Security Testing

The security of the website is based on sessions and if you are not logged in then you do not currently have an active session. If your session is not active then you will be redirected to the login page. All the pages within the site such as the homepage, the forum, faqs and tutorials are only accessible if the user is logged in.

The Guide for Future Maintenance

1.0 Introduction

1.1 Future Work

Over the course of developing this software we realized that certain features would not be implemented due to the interest of time. Our main focus in development is to make sure that our software can be easily picked up and maintained. We focused our efforts on ensuring that the main functionality of the software was achieved. Our client Askgc requested that we produce a web based application that would allow user to be able to login and post and answer questions. There were many restraints to the development. User posts had to be pre-approved if they were a new user. Below are the key functions we were not able to implement.

1.1.1 Live Chat

The live chat was not a part of the original scope collected from our customer, it was an additional requirement added on for complexity. It was suppose to be a feature that would allow a user the ability to communicate with an administrator about any system, or other

issue related to the site or questions in general. We have created a simple interface that would allow for someone to add on this feature if need be. The live chat implementation is going to use some kind of socket connection that will allow user to write to it and be able to process the information and display the results in the chat interface.

1.1.2 Privileged Users

A key point in our software was to implement the privileged user ability, meaning that a user that had at least 10 meaningful post would be allowed to freely post to the website. Normal users would be required to have their post reviewed before it would be allowed to be posted to the forum. A valid post or question is defined as any post that is related to video production that does not contain any inappropriate language or profanity. We have already added the normal user attribute to our askgcUsers table in our database. In order to implement the privileged user functionality we would have to count the number of meaningful post every time the admin or moderator has determined that they have contributed a meaningful post. Once they have reached the tenth valid post their user status would be updated to privileged user.

1.1.3 Validate Post

When a “Normal” user submits a question or post to the forum the question is supposed to be sent to an administrator or a moderator. The question or post would then be reviewed for quality control purposes. The admin or moderator would then approve or deny that question based on the questions relevance. If the question is inappropriate or the question contains any profanity the question will be denied and the user will be notified. If the question is approved it will be posted to the forum and the user will be notified of this action. If we were to implement this feature would have a temporary questions and temporary post table and then we would create a form that would only be accessible to admin or moderator that would be able to approve or deny the questions. If approved or denied they would be removed from the temporary tables. If approved then they would be added to the post or questions table depending on which type it was and

the user would receive an email. If denied the entry would be deleted and the user would be notified.

1.1.4 Search bar

The search bar implementation was meant to be a way to search for a particular question in the database or search for content related to the site. If we had implement this search we would have queried the database and selected all questions that contained the keywords from the search. The result of this query would then be displayed on a search results page where the user could click on a links to that would direct them to content relevant to their search. We would have also made it so the most relevant items would appear first. This could have worked as a form submission that would function to submit data to a page that would process the user's request and then render a page of results and if nothing was found the page would inform the user.

1.1.5 FAQ Retrieval

This feature was not a part of the original scope, it was added on for more complexity. The purpose of this feature is to store all questions related to the useage of the site and if the same question was asked at least 5 times then it would be retrieved from the database and then posted to the FAQ page. If we were to implement this feature we would have created a form that would take in the question and parse the data for keyword and then increment a counter. Once the count reached 5 then we would know to display this question.

2.0 GitHub Link

2.1 AskGreycomm Maintenance

In order to go about extending, adding features, or bug-fixing the application, you will need access to the code. You will be able to find all of the necessary files in our GitHub account on the link below:

<https://github.com/askgcSeniorProject>

In order to access the live website and test out its functionalities, please visit the link below:

<http://www.cs.loyola.edu/~sjean/askgc/login.php>

Currently, the iOS App is not live, but we will have a direct simulation available when asked, and those files are also available on the GitHub account. In order to make the App go live, we would need an App Developer License, which is \$100 to acquire. If our customer believes that the App would be crucial for his needs, he may acquire the license.

Reflections

1.0 Reflections on the Software Development Process

1.1 Kelsey's Reflection

Throughout this year, I have learned so much about the software development lifecycle and the importance of each section within it. Starting out at the beginning with meeting with our customer to find out information about what requirements he wanted was crucial for us to deliver a good, full functioning piece of software. In our original consultation meeting, our customer was explaining functionalities he wanted, and it was difficult for us at that time to tell him if we did not know how to do that, or let him know that we thought we might not have enough time to complete all of the requirements he wanted.

As young software developers, this was new to us. I know that I never wanted to tell him that we would not be able complete some of his requirements, because that would make us look bad. That is something I had to learn as the process went on, being able to voice what we would be able to do, versus something that we would not be able to do. Of course, in any situation,

dealing with a customer who does not understand coding functionalities is difficult, but the goal is always to please them. The additional stages of the software development process such as coding, testing, and documentation were all also very important, but I found the most important to be the requirements elicitation, and the documentation. Our customer wants to be able to take this piece of software with him, so documentation was important.

In the beginning, we had a rough estimate of 3-4K LOC, which was actually about where we ended up in the final stages of the project. We had originally estimated that certain parts would take longer, such as building the database and connecting to it, but that was actually a pretty simple task to take on. One of the harder tasks that we had underestimated was figuring out how to validate an email address and have the user activate their account. Having a secure website was important to us and our customer, so we wanted to ensure all those functionalities worked as intended.

Some requirements had to change over time, such as the different leveled users that would be in the system. We have set up our database tables in a way such that if someone would like to implement that requirement, it is possible, but we simply did not have enough time. Our customer was understanding about these changes, and felt that they weren't necessary to have.

Learning more about the Software Design process was important to me. I am going on to work as a real-life Software Engineer, and having these skills before I enter the workforce is important. Additionally, working in a group setting is preparing me for the real world, since in many cases, my team members will be relying on me to finish a specific portion of the project.

For this project, I learned almost all new languages and I feel confident in the fact that I now know them. Previously, I had only worked briefly with html and php, but now I can say that I have a full understanding of how to combine them together to get a fully working product. I also had little experience with MySQL, so finally getting to understand MySQL and get a fully working project that interacts with a database was extremely rewarding to see in the end. I am very happy that I gained these experiences, especially because it was a type of software that I was able to test along the way. We also had some exposure to XCode and using Swift for building an App. Although daunting at first, the way in which XCode sets up a storyboard was a pivotal element in the way in which we were able to design an iOS App.

To ensure the quality of our software, we had to perform a lot of testing and a lot of validation on our website. Coming up with test cases was crucial to ensuring that our software met all of the required specifications. We used a variety of different testing tools to ensure the quality of our project, such as online validators, the integrated XCode tests, and using some load/stress testing. These tools were decently easy to use for our project.

Finally, to make our project easily maintainable, we had to use similar coding styles throughout the files. Additionally, pieces of code that need to repeat in multiple files, we combined in one file, and then would include those files in the header of the file. This would help because if you need to make one simple change to the navigation bar, for example, it would be easy to change it in just one place. We also had to add header comments to our files to ensure if someone were to pick up our code, they would understand what was going on in that specific project.

Watching this project from beginning to end was extremely rewarding, and I think we are very proud of the piece of code we have developed. Although happy the semester is over, I hope that I take all that I have learned from this semester and continue to practice all I learned in the workforce after graduation.

1.2 Steve's Reflection

Over the course of this semester I learned a lot about requirements elicitation. For our senior project we had to go out and find a client that had a need for software. We meet with our customer (Jay Dunmore) several times to make sure that we fully understood every aspect of what he wanted us to implement. I learned a lot about use cases, design, and testing. Use cases help better understand how to implement the functionality you are trying to achieve. The design is a good way to ensure you know the direction of your project before you write a line of code. Testing is the best way to determine that your software achieved the function you're trying to implement. The most important part in this process is the requirements elicitation. If you are not fully aware of what the customer wants, you may create something that is completely the opposite or more complex than what they actually wanted.

We initially estimated that it would take roughly 3,000 lines of code to complete this project and in the end we basically achieved that number. It took most of the semester to complete the site portion because most of the initial effort went towards researching the different languages.

Over time the requirements remained about the same, but what we could achieve in the given timeframe changed. We spent the first month of the project doing research and then within that month we also had to plan out what we wanted the interface to look like. Then we decided what path a user would take when accessing the site. Initially when you signed up and it was successful the user was redirected to the homepage; to make the site more secure, we decided that their account had to be active first before they were allowed to access anything. Originally the initial login through sign up was allowed but the next time you used the site the account had to be active. We changed this so that no matter what, it had to be active to ensure better security. This change was more of a design change on our part but it helped with the requirement that logins had to be secure.

I learned that with Software Design you need to make sure that the software is easy to use, easy to maintain, and robust. The software has to be designed in such a way that any new user can easily understand how everything functions. It also has to be written so that any future programmer can easily make adjustments or add new functionalities. We use several different design notations that helped make the overall project scope easy to understand. Making the software robust means that the program has to be able to handle invalid input and not cause a system failure. We used a database entity relationship diagram, data flow diagram, and an architecture diagram. The use of use cases also helped better understand the project and how to design it.

For this project I had to learn PHP, JavaScript, HTML, CSS, Swift, and MYSQL. I had some basic knowledge of HTML and MYSQL, but the rest I had to learn from scratch. I spent a lot of time watching tutorials looking up functions and syntax. It was a challenge to learn all these. In the end it may not seem like we accomplished much but it was a lot, if you consider that we never worked with any of these before. They were all pretty easy to pick up. The hard part of it

all was just trying to understand what we wanted to do so could find the proper tools that could assist in accomplishing that task.

Quality assurance is a very important aspect of software development. We ensured that our software met the requirements by having different test cases and seeing if we could create a condition where that case would be met. For example when you try to login there are several different cases. One, if no data was entered. Two, if one or both values are blank. Three, if your account is not active. Lastly, if the username or password combination is not valid it will also print an error message. We have taken several steps to ensure that our software can perform well under such conditions.

The main goal in our testing is to stress test the system to see if there are any changes when multiple users try to access any given functionality. We plan on using a PHP validator.

We took several steps to ensure that our project was easily maintained. We added useful comments; we created conditional statements that are simple and easy to read. We have well documented the process of creating our software, and we condensed as much of the code as possible to make it easy to change things that function across several pages.

1.3 Jordan's Reflection

Upon reflection, I have learned more about the software development lifecycle through participating in the project. Many of the principles and topics I had learned about in the previous semester, I was able to apply and understand more. Getting the requirements down was the most important especially as it provides the framework for doing anything related to the project afterwards. There was no difficulty in working our customer, Jay Dunmore from GreyComm Studios. The communication did not lack in any way and we were given some leeway with what could and could not be accomplished. Although it was not a completely professional interaction, it did provide a better glimpse as to how these interactions might take place in the real world. The design aspect of the project allowed for the creation a framework for the project. It also provides some assurance that the requirements are understood thoroughly.

The original estimate did slightly change, only in the way of making minor additions. That is to be expected, however and through my courses I have learned that usually estimates

made in the beginning are not very reliable. Honestly, I am not sure how I could have done anything differently to get a better estimation on the required effort of this project. The only conclusion that can be made is that it is much better to make an overestimation as opposed to an underestimation.

Throughout this project I had to utilize to new languages: PHP and Swift. Understanding how each of these languages worked provided great difficulty; however I was able to learn things that contributed to the project. Since Swift's updates cause many older functions to become obsolete, finding an abundance of resources with regard to some newer functionalities provided a level of difficulty. I also had to understand, to a degree, how to use Xcode's testing framework, as well as PHPUnit, though that stopped working mid test. I had ran into many difficulties understanding the components, especially as majority, if not all the project files needed to make a connection to the database.

As seen with many other projects, ensuring the quality of the software was of great importance. Each file is commented with a description as to what it does, aiding with maintainability. Most files lack much complication, too. Through interacting with the software in various ways, it was ensured that the software met the requirements specifications as best as they could.

Overall, this project is arguably the biggest that I have had a stake in and definitely provided great difficulty; along with the difficulty came a great deal of learning. It has shown me slightly more of the intricacies of the software development process. Although I can say the most important thing that it has shown me is that coding is not my forte, this experience was not one that was negative.

Size of Product

1.0 Product Metrics

1.1 Estimation of Size of Product

When estimating the size of our product, we could not use additional tools because we did not create our project in Eclipse, instead we build a website and iOS Application. From that, we

were able to simply see how many lines of code our project has. In total, we came to approximately 4,162 lines of code, give or take a few lines. This is including some copies of files that are needed for the iOS Application that have about a 10 line difference. Without including copies, it came to a total of 3,134 lines of code. This estimation is much bigger than what we had thought it would be in the beginning, but all of these functions and lines of code are necessary for our project.

Testing Results

1.0 Results

1.1 Types of Testing and Results

Load testing: LoadImpact.com

Loadimpact.com is a website that provides many different forms of testing. Load testing was the focus as opposed to stress testing. # of test were carried out to see how the site could handle 50 users performing repetitive actions. Each test took place over a period of five minutes and had 50 virtual users repeatedly interact with the system. Due to monetary limitations, testing with more than 50 consistently active virtual users was not possible. However, it is not expected that more than that amount of users will be actively be using our interface simultaneously. Unfortunately with this said, each graph is flat meaning that the system was not loaded enough.

There was difficulty in providing a various number of user with different registered information that were also validated. Email validation would allow a user to access the complete site. Although manipulation of the php script handling activation would most likely allow for the overcoming of that obstacle, this was not something that was able to work completely and therefore not included.

Test 1: User information selected randomly from a pool of 45 different sets of information. Whether or not information was repeated was not the concern. The goal was to see how the system handled having that number of user interaction.

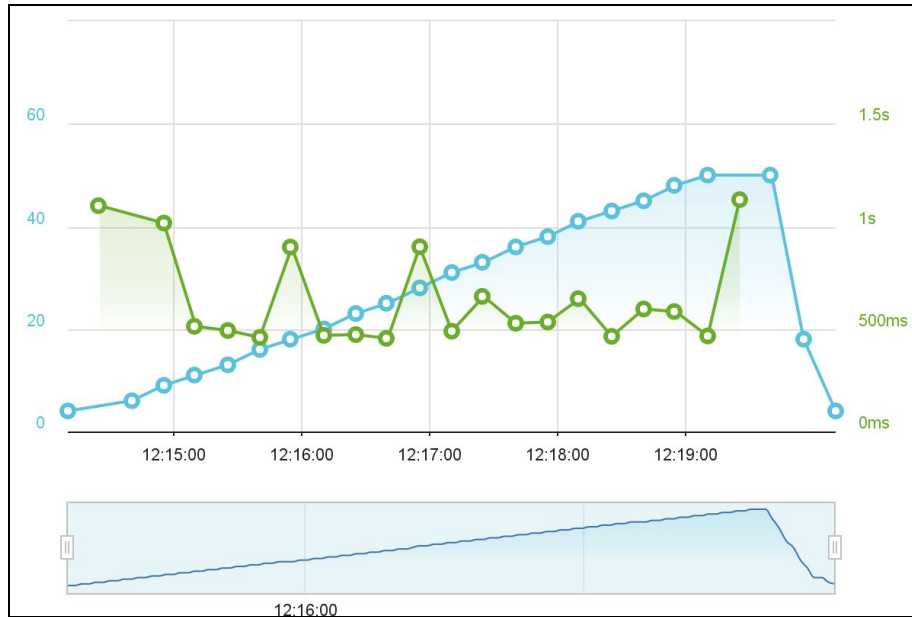


Figure 23: Load Testing 1

Results

Test2: Multiple virtual users logging in with the same credentials and posting a new question to a particular forum.

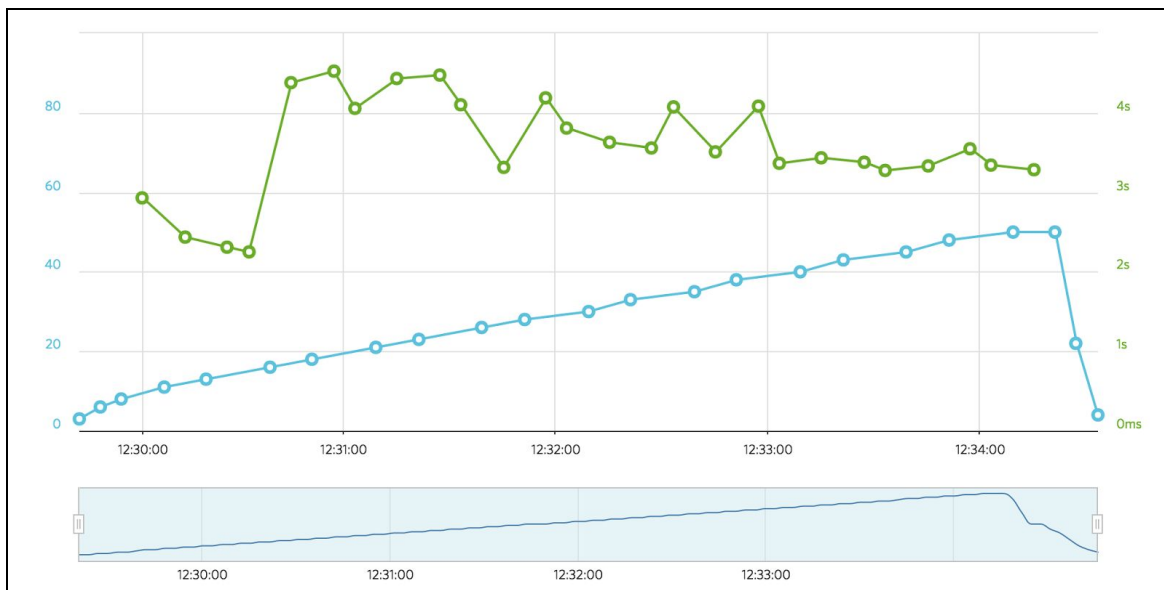


Figure 24: Load Testing 2

[Results](#)

Test3: Multiple virtual users logging in with the same credentials and interacting with pages in the interface, including: posting a reply as well as browsing through the Tutorials tab, and the FAQ tab, before logging out.

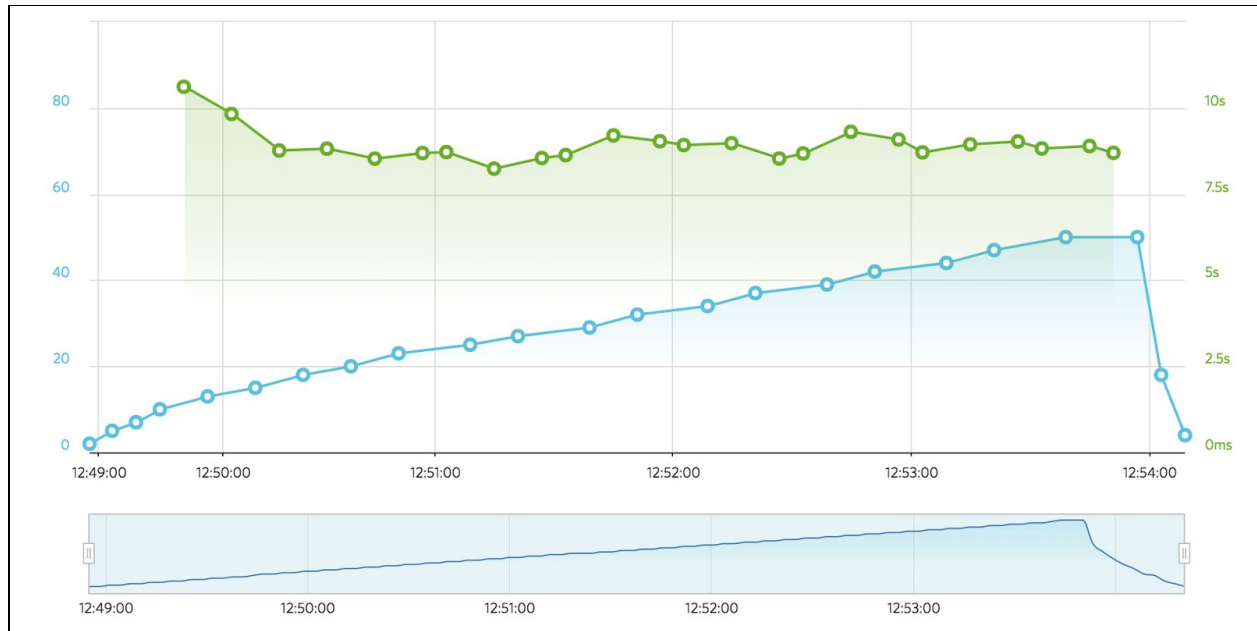


Figure 25: Load Testing 3

[Results](#)

Initially frameworks such as PHPUnit, JMeter, SimpleTest had been attempted. However, there was the realization that components needed to run, for example PHPUnit, either were not installed on the Linux machines or refused to work on my own personal computer. We additionally used an HTML Validator to test and see how our overall website functioned. It returned us with 6 small errors, so overall good results.

1.2 Test Cases

| Test Case Number | Success |
|--|--|
| Test Case #1: Login functionalities | Login successfully works after activation ✓ |
| Test Case #2: Registration | Emails sent and registration works ✓ |
| Test Case #3: Forgot Password | Checks email exists ✓ |
| Test Case #4: Reset Password | Creates new password, ensures password security, checks security questions ✓ |
| Test Case #5: Change Password | Checks current password, ensures password security on new password ✓ |
| Test Case #6: Post a Question to Forum | Question is posted within Category ✓ |
| Test Case #7: Reply to a Question | Reply is posted beneath question ✓ |
| Test Case #8: Create a Category | Category is posted on forum home ✓ |

Integration Testing/System Testing

| What we Tested | Results |
|--|--|
| Account Creation on both Website and App | User information was properly stored in the database. ✓ |
| Logging in using test accounts | We notice that the accounts can be properly accessed and the user signed is clearly noted. |

Additional

| | |
|-------------------------------------|--|
| Security Testing | Password is secure (complex) ✓ |
| Database Queries | Store all correct information, user information, active account ✓ |
| Compatibility Testing | Chrome, Firefox, Safari, Internet Explorer ✓ On iPhone, Logout takes time X |
| Navigation Testing | Only access homepage once logged in ✓ |
| DoIt Simple Web Application Testing | Could not access source code for DoIt X |
| Black Box and White Box Testing | All output we expected was a success ✓ |

User Manual Table of Contents

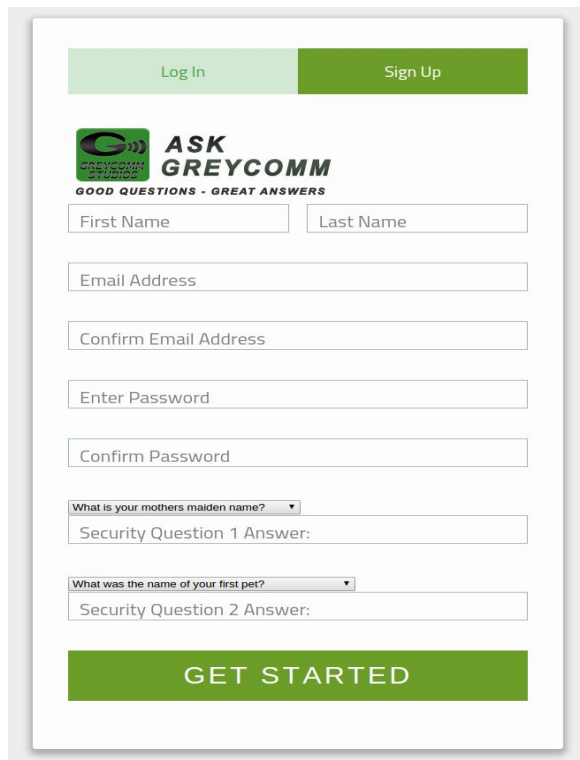
| | |
|--|----|
| Step 1: <i>Signup/Registration</i> | 65 |
| Step 2: <i>Check Email Confirmation for Activation</i> | 66 |
| Step 3: <i>Login</i> | 68 |
| Step 4: <i>Homepage Redirect</i> | 69 |
| Step 5: <i>Navigation to Forum</i> | 70 |
| Step 6: <i>Navigation to Tutorials</i> | 73 |
| Step 7: <i>Navigation to FAQ</i> | 74 |
| Step 8: <i>Logout</i> | 75 |
| Step 9: <i>Change Password</i> | 75 |
| Step 10: <i>Forgot Password</i> | 76 |

User Manual Table of Figures

| | |
|--|----|
| Figure 1: <i>Signup Form Website</i> | 65 |
| Figure 2: <i>Signup Form App</i> | 65 |
| Figure 3: <i>Registration Successful Email</i> | 66 |
| Figure 4: <i>Activation Account Link</i> | 66 |
| Figure 5: <i>Confirmation Email that you have activated your account</i> | 67 |
| Figure 6: <i>Login Form Website</i> | 68 |
| Figure 7: <i>Login Form App</i> | 68 |
| Figure 8: <i>Homepage on website</i> | 69 |
| Figure 9: <i>Homepage on App</i> | 69 |
| Figure 10: <i>Forum Home Page</i> | 70 |
| Figure 11: <i>Question View under Category</i> | 70 |
| Figure 12: <i>Posting a reply</i> | 71 |
| Figure 13: <i>Creating a new Category</i> | 71 |
| Figure 14: <i>Create a new Question under a Category</i> | 72 |
| Figure 15: <i>Tutorials Tab</i> | 73 |
| Figure 16: <i>Frequently Asked Questions Tab</i> | 74 |
| Figure 17: <i>Change Password Website</i> | 75 |
| Figure 18: <i>Change Password App</i> | 75 |
| Figure 19: <i>Confirmation Password Changed Email</i> | 75 |
| Figure 20: <i>Password Reset Email Confirmation</i> | 76 |
| Figure 21: <i>Reset Password with security Questions</i> | 76 |
| Figure 22: <i>Reset Password Confirmation Email</i> | 77 |

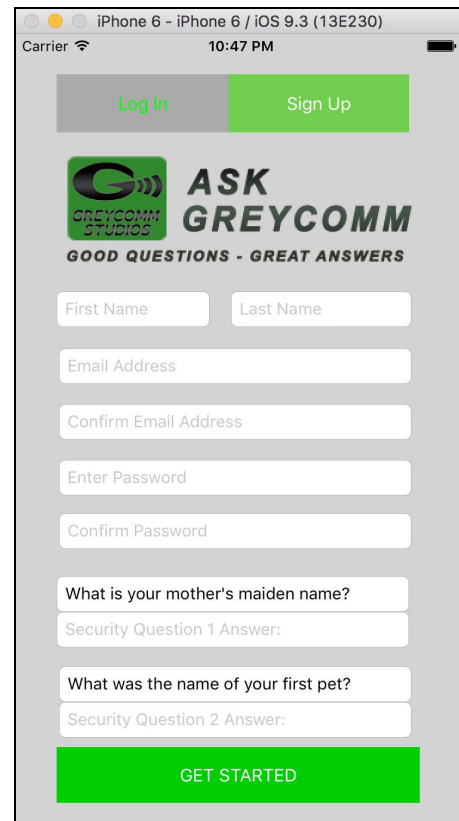
User Manual

1. Go to <http://www.cs.loyola.edu/~sjean/askgc/signup.php> or view the Sign Up Form on the App. (Left: website, Right: App)
 - a. Fill out the form and make sure that all data is entered



The screenshot shows the web version of the sign-up form. At the top, there are two green buttons: "Log In" and "Sign Up". Below them is the "ASK GREYCOMM" logo with the tagline "GOOD QUESTIONS - GREAT ANSWERS". The form consists of several input fields: "First Name", "Last Name", "Email Address", "Confirm Email Address", "Enter Password", "Confirm Password", "What is your mothers maiden name?" (with a dropdown arrow), "Security Question 1 Answer:", "What was the name of your first pet?" (with a dropdown arrow), and "Security Question 2 Answer:". At the bottom is a large green "GET STARTED" button.

Figure 1: Signup Form Website



The screenshot shows the mobile app version of the sign-up form. The status bar at the top indicates "iPhone 6 - iPhone 6 / iOS 9.3 (13E230)" and "10:47 PM". The app has a grey background. At the top, there are two green buttons: "Log In" and "Sign Up". Below them is the "ASK GREYCOMM" logo with the tagline "GOOD QUESTIONS - GREAT ANSWERS". The form consists of several input fields: "First Name", "Last Name", "Email Address", "Confirm Email Address", "Enter Password", "Confirm Password", "What is your mother's maiden name?", "Security Question 1 Answer:", "What was the name of your first pet?", and "Security Question 2 Answer:". At the bottom is a large green "GET STARTED" button.

Figure 2: Signup Form App

2. If signup is successful, check your email for something similar to the email shown below.
 - a. Click the activation link
 - b. Click on the activate account button
 - c. You should receive an email letting you know your account has been activated
 - d. The activation process is the same for both the website and the application.

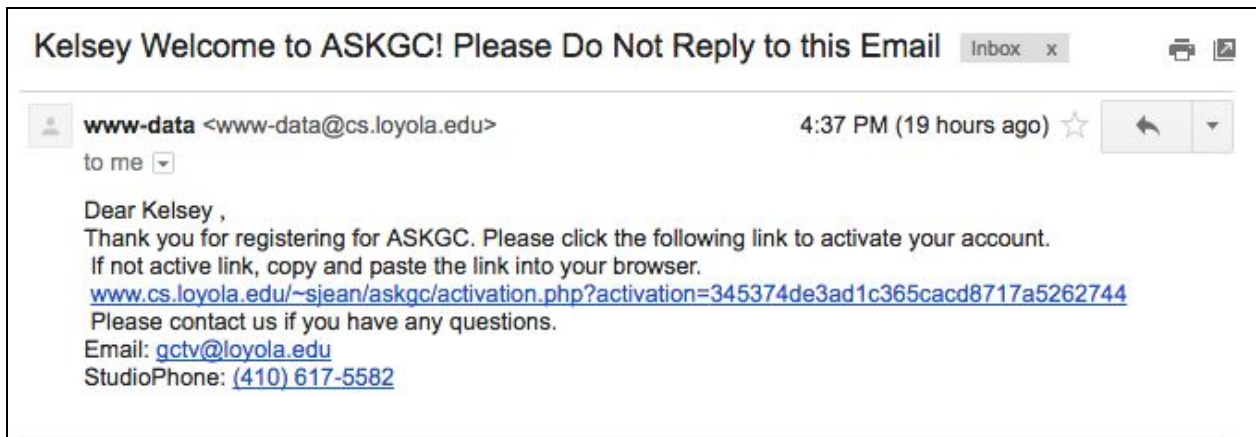


Figure 3: Registration Successful Email

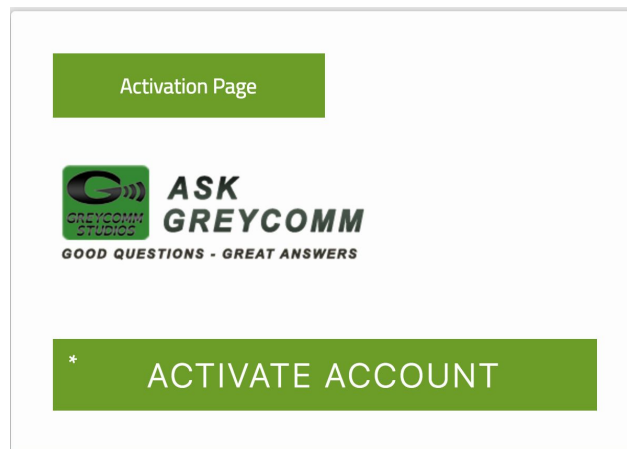


Figure 4: Activation Account Link

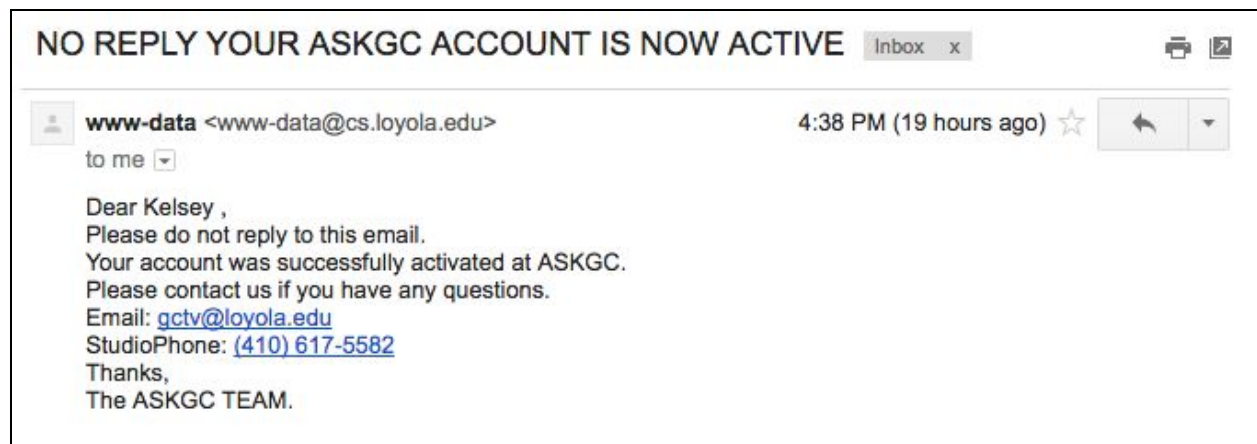
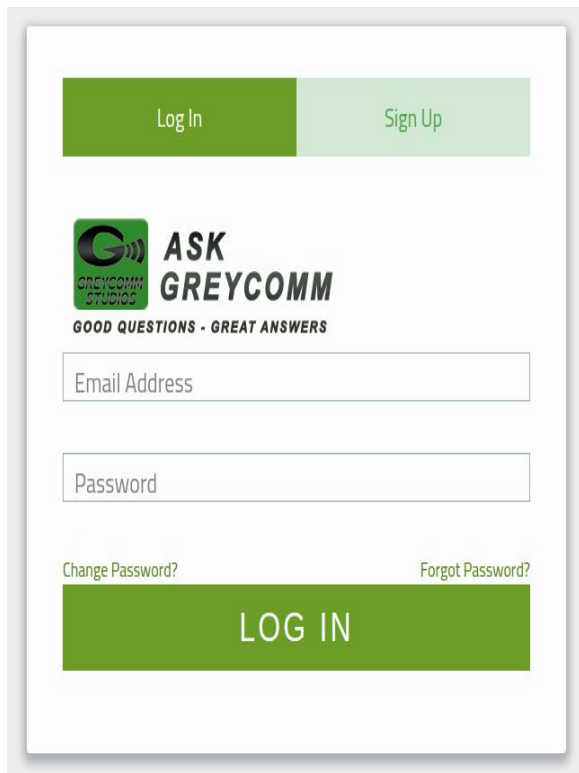


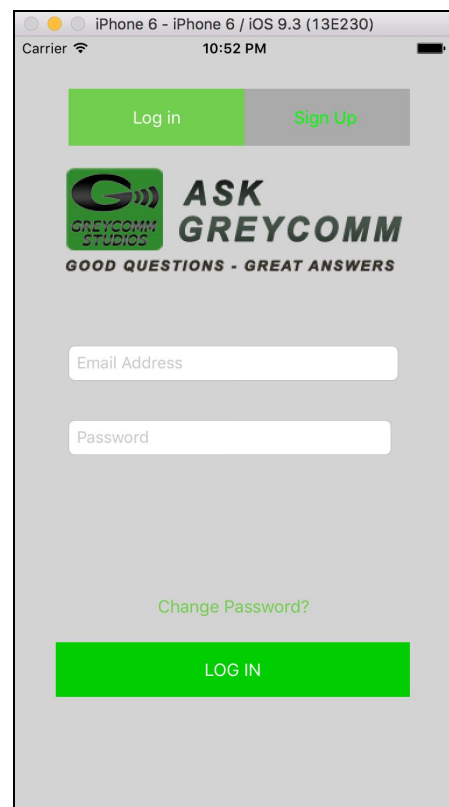
Figure 5: Confirmation Email that you have activated your account

3. You should have been redirected to the login screen, located at <http://www.cs.loyola.edu/~sjean/askgc/login.php> or redirected to the login screen on the app, which are both show below.
- If you are properly registered and are still receiving an error, that means that you have not activated your account and you will receive an additional email telling you to activate it before login.
 - Otherwise, enter your credentials and login to the AskGreycomm website.



The web login form for AskGreycomm features a white background with a green header bar. The header bar contains two buttons: "Log In" in white text on a green background and "Sign Up" in green text on a light green background. Below the header is the AskGreycomm logo, which consists of a green square with a white "G" and the text "ASK GREYCOMM" in bold, with "GOOD QUESTIONS - GREAT ANSWERS" in smaller text below it. The form includes two input fields: "Email Address" and "Password", both with placeholder text. Below the password field are two links: "Change Password?" and "Forgot Password?". At the bottom is a large green button with the text "LOG IN" in white.

Figure 6: Login Form Website



The mobile app login form for AskGreycomm is displayed on an iPhone 6 screen. The status bar at the top shows "Carrier", signal strength, and the time "10:52 PM". The app interface has a grey background. At the top are two buttons: "Log in" in white text on a green background and "Sign Up" in green text on a grey background. Below these is the AskGreycomm logo, featuring a green square with a white "G" and the text "ASK GREYCOMM" in bold, with "GOOD QUESTIONS - GREAT ANSWERS" in smaller text below it. The form includes two input fields: "Email Address" and "Password", both with placeholder text. Below the password field is a link: "Change Password?". At the bottom is a large green button with the text "LOG IN" in white.

Figure 7: Login Form App

4. Once logged in, you will be redirected to the homepage, as seen below.
 - a. The app and website interface for the internal web pages are the same.

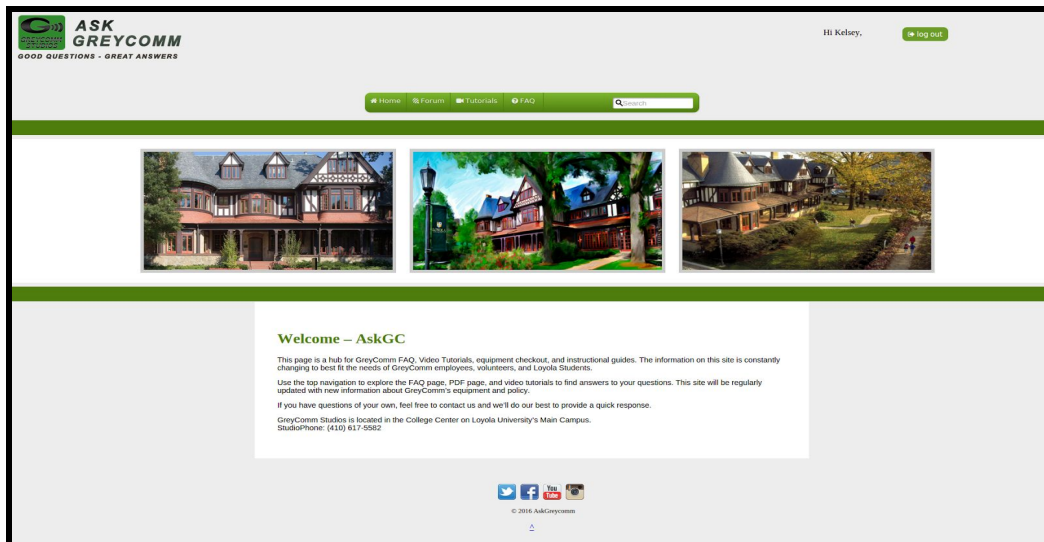


Figure 8: Homepage on Website

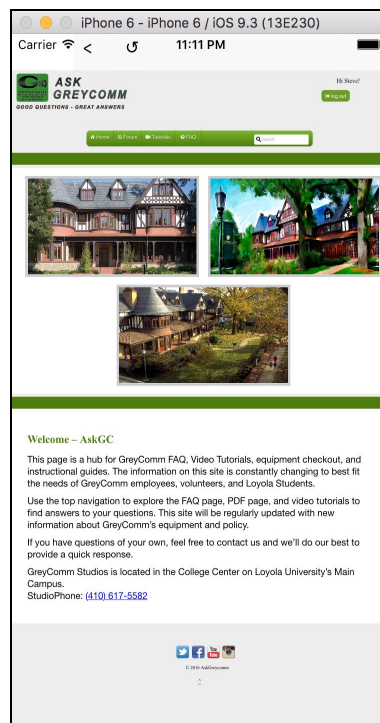


Figure 9: Homepage on App

5. Navigating to the Forum

- a. The next tab over is the Forum. On the forum home, you will see a list of current Categories and their descriptions.

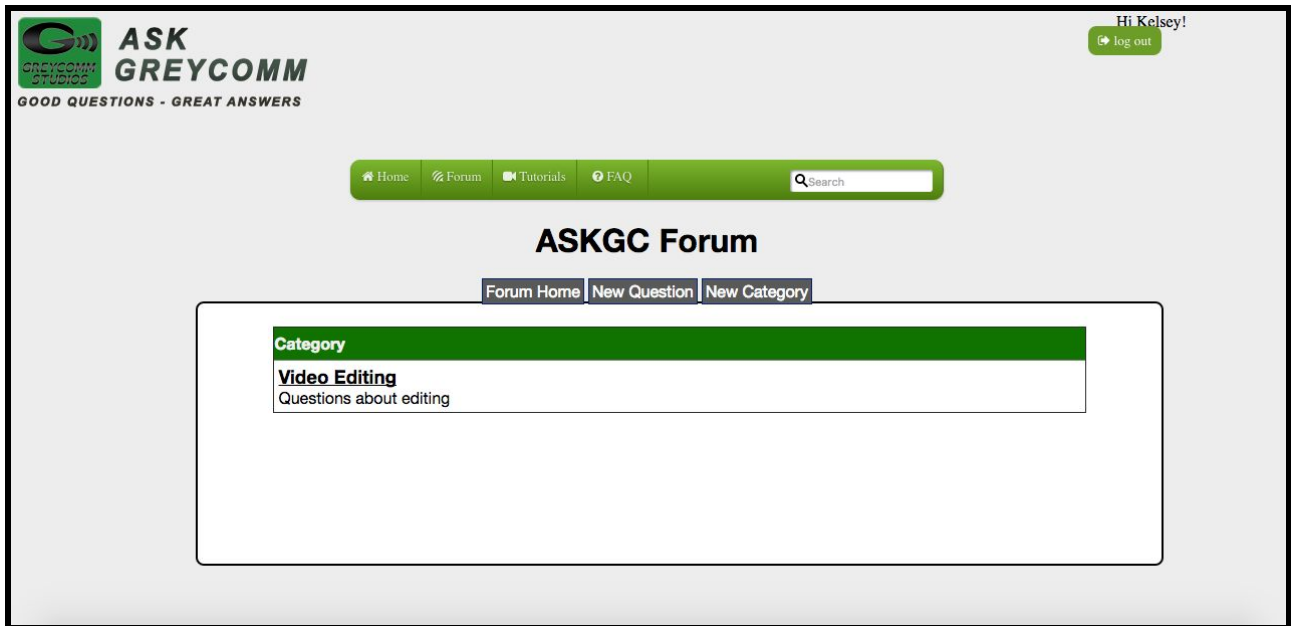


Figure 10: Forum Home Page

- b. If you would like to see questions available within the Category, click on “Video Editing”

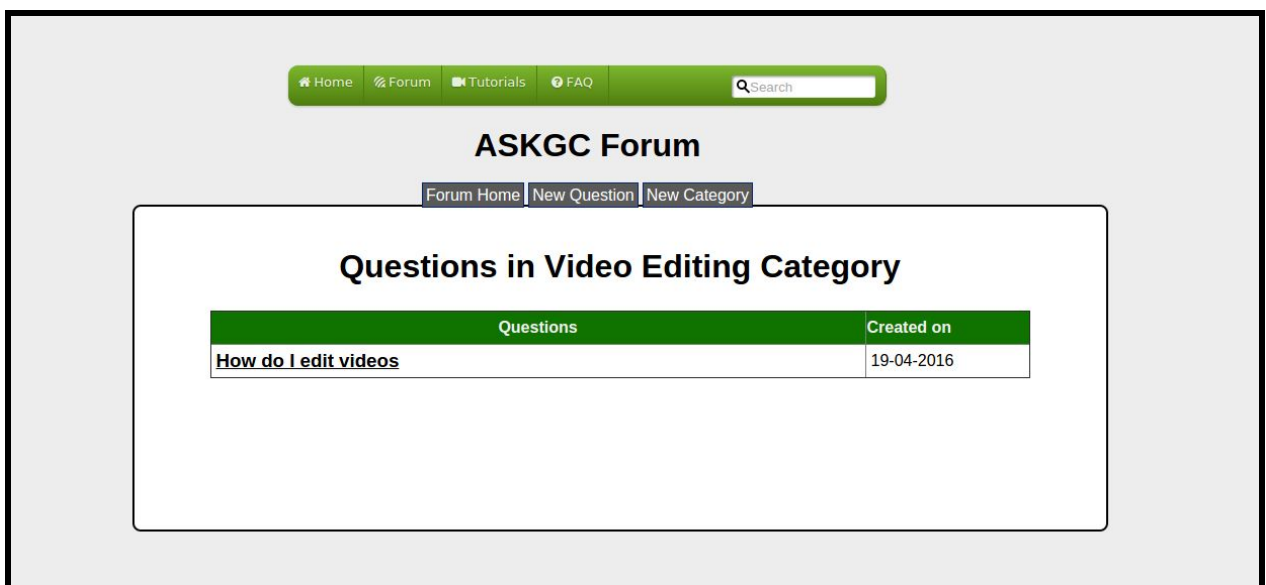


Figure 11: Question View under Category

- c. If you click on the “How do I edit videos” link, it will take you to a user’s question, and another user has the ability to reply to that question. The reply will be posted below the initial question.

The screenshot shows the ASKGC Forum interface. At the top, there is a navigation bar with links for Home, Forum, Tutorials, and FAQ, along with a search bar. Below the navigation bar, the forum title "ASKGC Forum" is displayed. Underneath, there are tabs for "Forum Home", "New Question", and "New Category". The main content area is titled "Posts related to 'How do I edit videos'". It contains a table with two columns: "Post" and "Created By". The first row shows a post with the text "i dont know" and the user "Kelsey Dramis on: 2016-04-26 21:02:33". Below the table, there is a section titled "Post a reply" with a large text input area and a "Submit reply" button.

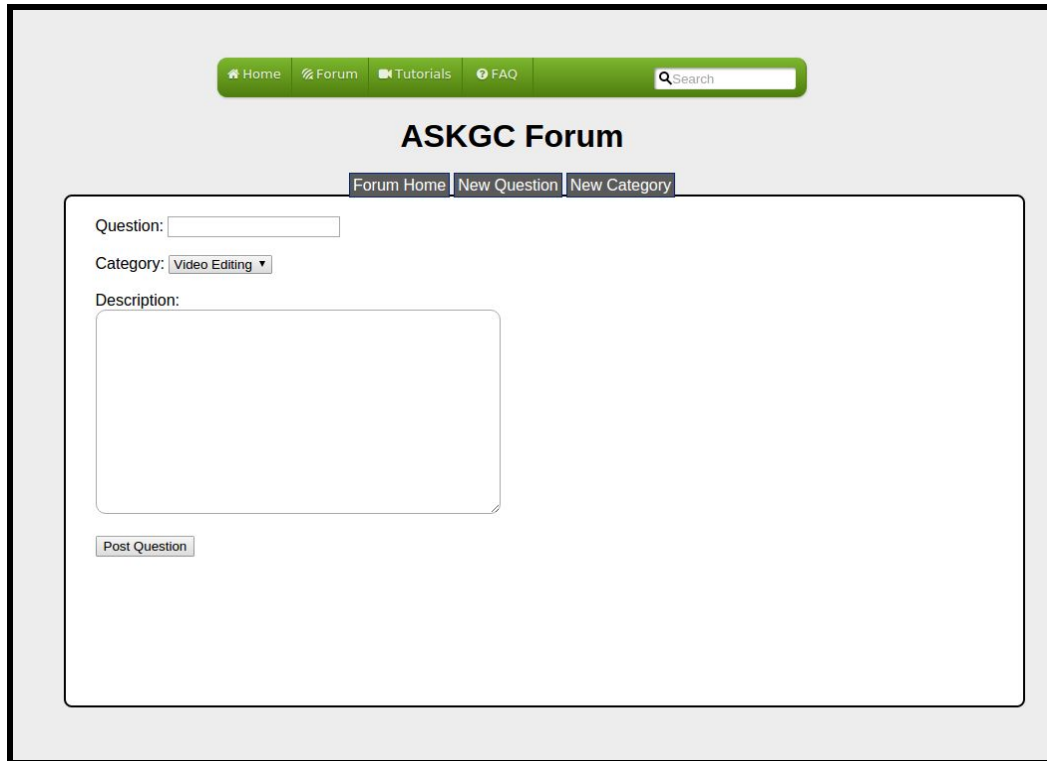
Figure 12: Posting a reply

- d. If you would like to create a new category, click on the “New Category” tab, where you will enter the Name and Description of that Category. The category will be added under the “Forum Home” tab.

The screenshot shows the ASKGC Forum interface with the "New Category" tab selected. The form contains two input fields: "Category name:" and "Category description:". Below the "Category description:" field is a large text input area. At the bottom of the form, there is an "Add category" button.

*Figure 13:
Creating a new
Category*

- e. If you would like to ask a new question, click on the “New Question” tab. You will enter the Question, select which Category you would like it to go under, and then post a description about your question. Once your question is posted, it will be available under the Category you have selected.



The screenshot shows the ASKGC Forum interface. At the top, there is a green navigation bar with links for Home, Forum, Tutorials, and FAQ, along with a search bar. Below this, the title "ASKGC Forum" is centered. Under the title, there are three tabs: "Forum Home", "New Question" (which is highlighted), and "New Category". The "New Question" tab is active, displaying a form with the following fields: "Question:" with a text input, "Category:" with a dropdown menu showing "Video Editing", and "Description:" with a large text area. A "Post Question" button is located at the bottom left of the form.

Figure 14: Create a new Question under a Category

6. Navigating to the Tutorials Tab

- a. After the Forum, there is the tutorials tab, which provides the user with a slew of videos to look at to assist them in common film production questions.

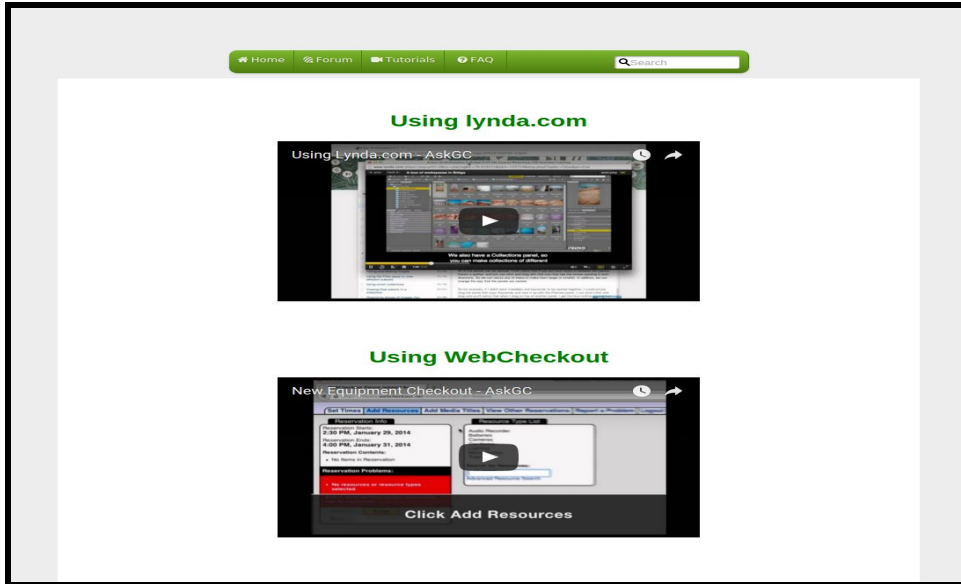


Figure 15: Tutorials Tab

7. Navigating to the FAQ Tab

- a. The FAQ tab holds the major frequently asked questions within AskGreycomm's website.

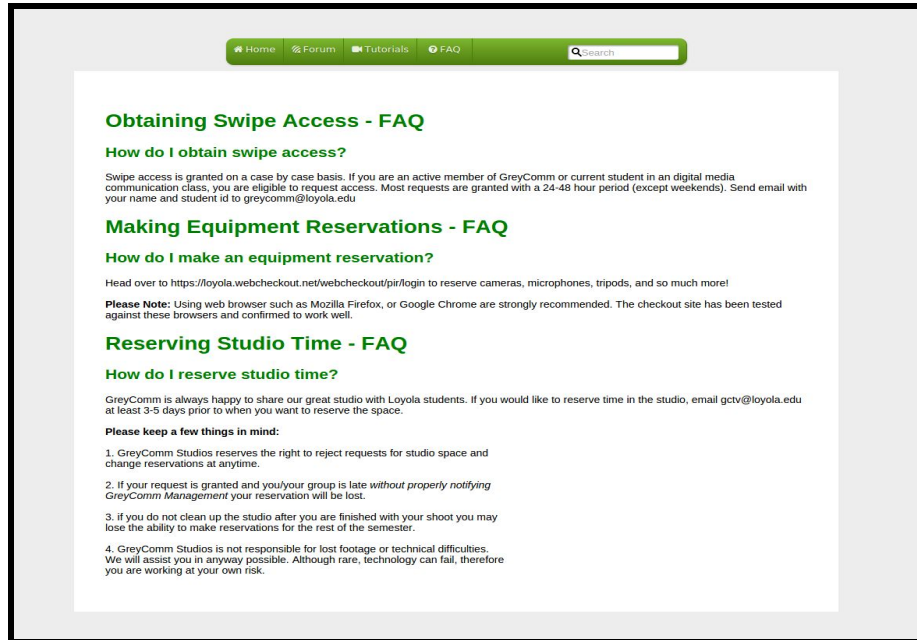
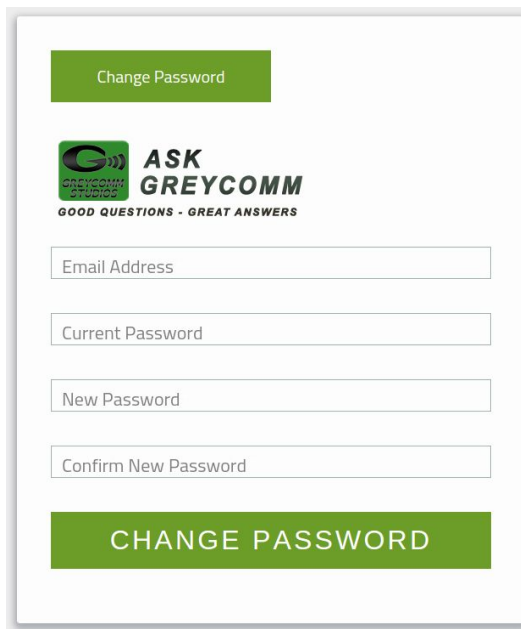


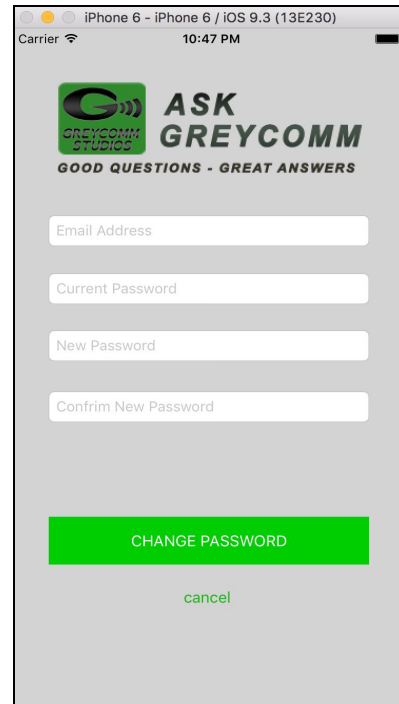
Figure 16: Frequently Asked Questions Tab

8. If you log out of the website, you will be redirected back to the login screen.
9. If you would like to change your password and you remember what it is, click the “Change Password” link, located on the login screen. You will be brought to the image below.



A screenshot of the ASK GREYCOMM website's 'Change Password' form. At the top, there is a green button labeled 'Change Password'. Below it is the ASK GREYCOMM logo with the tagline 'GOOD QUESTIONS - GREAT ANSWERS'. The form contains four input fields: 'Email Address', 'Current Password', 'New Password', and 'Confirm New Password'. At the bottom of the form is a large green button labeled 'CHANGE PASSWORD'.

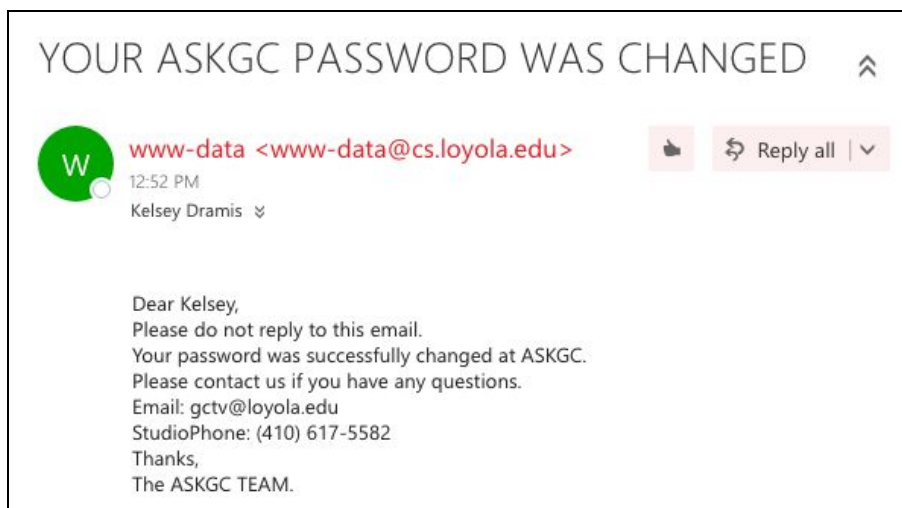
Figure 17: Change Password Website



A screenshot of the ASK GREYCOMM app's 'Change Password' screen. The status bar at the top shows 'Carrier', signal strength, and the time '10:47 PM'. The app header includes the ASK GREYCOMM logo and the tagline 'GOOD QUESTIONS - GREAT ANSWERS'. The screen features four input fields: 'Email Address', 'Current Password', 'New Password', and 'Confirm New Password'. At the bottom, there is a large green button labeled 'CHANGE PASSWORD' and a smaller green link labeled 'cancel'.

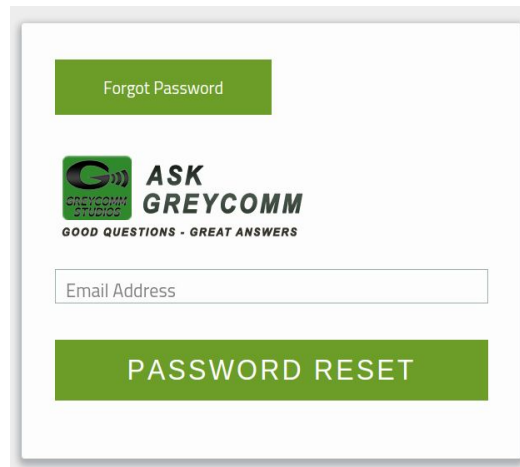
Figure 18: Change Password App

- a. Enter all of the appropriate information, and you should receive a confirmation email letting you know your password has been changed.



*Figure 19:
Confirmation
Password Changed
Email*

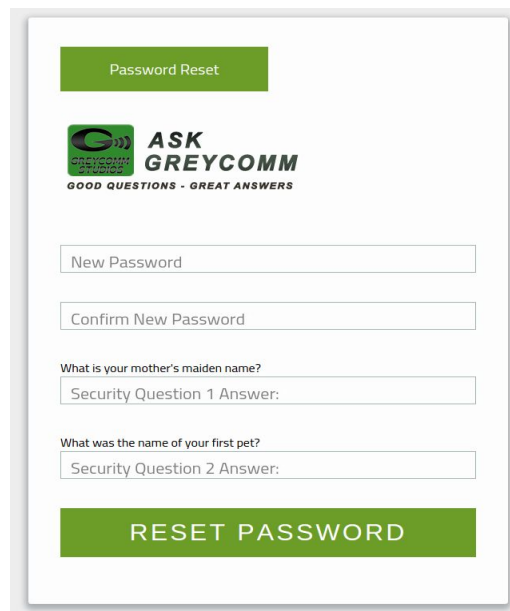
10. If you have forgotten your password and wish to change it, you will click the “Forgot Password” link (*Note: the application does not contain this functionality*)
- a. Initially, you will be brought to a page where you are required to enter your email.



The screenshot shows a web form titled "Forgot Password" in a green header bar. Below the header is the AskGreycomm logo, which includes a green square with a white 'G' and the text "ASK GREYCOMM" and "GOOD QUESTIONS - GREAT ANSWERS". Under the logo is a text input field labeled "Email Address". At the bottom of the form is a large green button with the text "PASSWORD RESET" in white capital letters.

Figure 20: Password Reset Email Confirmation

- b. Once you enter your email and the email is correct, you will be brought to the password reset form, and will be required to enter your security question answers.



The screenshot shows a web form titled "Password Reset" in a green header bar. Below the header is the AskGreycomm logo, which includes a green square with a white 'G' and the text "ASK GREYCOMM" and "GOOD QUESTIONS - GREAT ANSWERS". Under the logo are two text input fields: "New Password" and "Confirm New Password". Below these are two security question prompts: "What is your mother's maiden name?" followed by a text input field labeled "Security Question 1 Answer:", and "What was the name of your first pet?" followed by a text input field labeled "Security Question 2 Answer:". At the bottom of the form is a large green button with the text "RESET PASSWORD" in white capital letters.

Figure 21: Reset Password with Security Questions

- c. You should receive a confirmation email letting you know your password was reset.

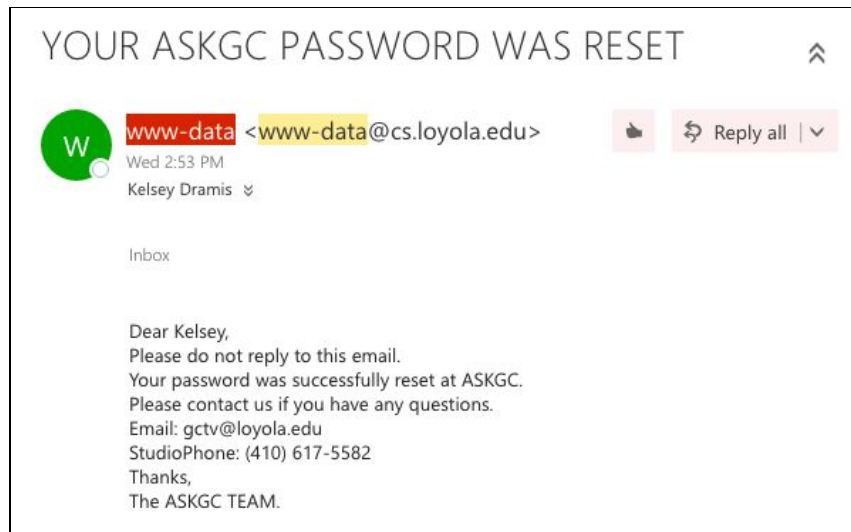


Figure 22: Reset Password Confirmation Email