Design and Testing

Project Name: ResearchTube

Prepared By:

Document Owner(s)	Role
Pardeep Rathore	PM, Developer
Marcus Tam	Technical Lead, Developer
Zhaoyang Zhong	Quality Assurance, Developer
Yue Wang	Client Liaison, Developer
Zhewen Zhang	Integration Lead, Developer

Table of Contents

Software Description:

Our project is to design an aesthetic interface for a video hosting website. “ResearchTube" will be the new web home for academics and researchers. The main concept behind ResearchTube will be the AI algorithm that converts research articles into video. Our main task is to develop the front end and back end to store videos for this web-app. This includes:

Primary Tasks:

- Index page: Where users will be able to register or login themselves, small description about the business model, some demo videos, application features’ description, and client feedback.
- Home page that displays published videos in a categorized fashion.
- Video page, where users will be able to upload videos.
- Contact Us page, a user sends a query or request for more information.
- About Us page, Company’s history, business model, ReseachTube information, and customer feedback.
- Plan page, user will be able to pay for his access level.

Secondary Tasks:

- User registration will be authorized via: email, facebook or gmail account:
- A user can upgrade his/her account from basic to pro or vice versa.

There are other features that will be added as per client’s advice.

User Groups:

1) Administrator
2) Researchers from Universities and Industries

User Scenario

Use Case ID	Use Case Name	Actor(s)
1	Browse Welcome Website	Researcher, Admin , (User)
2	Home Page	Researcher, Admin , (User)
2.1	Search Function	Researcher, Admin, (User)
3	Sign Up	User
3.1	Payment Plan	Researcher, Admin
4	Sign In	Researcher, Admin, (User)
5	Text to Video Converter	Researcher, Admin
6	Account Settings	Researcher, Admin
6.1	Add/Edit/Delete Account Info	Researcher, Admin
7	Watch Video	Researcher, Admin
8	View Profile	Researcher, Admin

9	About Us	Researcher, Admin, (User)
9.1	Company Description	Researcher, Admin, (User)
9.2	Demo Video	Researcher, Admin, (User)
9.3	Feedback	Researcher, Admin, (User)
9.4	Information about the owners	Researcher, Admin, (User)
9.5	ResearchTube Team	Researcher, Admin, (User)
10	Contact Us	Researcher, (User)
10.1	Map Location of Headquarters	Researcher, (User)
10.2	Address & Phone Number info	Researcher, (User)
10.3	Inquiry Form	Researcher, (User)
11	Admin Portal	Admin
12	View member list	Admin
12.1	Upgrade Member to Admin	Admin
12.2	View Member Information	Admin
12.3	Remove User	Admin
12.4	Upgrade Researcher Plan	Admin

13	View Feedback	Admin

Use Case Model Explanation

ID:	1
Name	Browse Welcome Website
Actor(s):	Researcher, Admin , (User)
Flow of Events	NA
Pre-Conditions:	NA
Post-Conditions:	NA
Description:	This is the user welcome page. This page will lead the user to the rest of the website. This portion displays the outline of ResearchTube

ID:	2
Name	Home Page
Actor(s):	Researcher, Admin , (User)
Flow of Events	1. User Navigates to Welcome Page 2. Redirect to Home Page

Pre-Conditions:	NA
Post-Conditions:	NA
Description:	The home page is where all the videos will be displayed. If the user is logged in, the videos will be displayed in a fashion that aligns with the users interests.

ID:	2.1
Name	Search Function
Actor(s):	Researcher, Admin, (User)
Flow of Events	1. User Navigates to Home Page 2. Click search box 3. Enter the thing user wants to search 4. Click search button
Pre-Conditions:	NA
Post-Conditions:	Home page outputs list of videos that correspond to the user query
Description:	Search box can help users to search something on this website.

ID:	3
Name	Sign Up
Actor(s):	User

Flow of Events	Navigate to sign up page
Pre-Conditions:	Do not have an account, or create a new account with another email address
Post-Conditions:	Upload user information to the database, and redirect users to the home page.
Description:	Allow user to create an account

ID:	3.1
Name	Payment Plan
Actor(s):	Researcher, Admin
Flow of Events	1. User Navigates to Welcome Page 2. Sign in 3. Redirect to Plan Page
Pre-Conditions:	Users must sign in
Post-Conditions:	Researcher's plan is now upgraded. Billing of the user will now commence.
Description:	Users can choose their plan and pay for it on this page.

ID:	4
Name	Sign In
Actor(s):	Researcher, Admin, (User)

Flow of Events	Navigate to sign page
Pre-Conditions:	Must logout before login
Post-Conditions:	Login successfully if the id and password are both correct.
Description:	Allow user to sign in

ID:	5
Name	Text to Video Converter
Actor(s):	Researcher, Admin
Flow of Events	1. Researcher / Admin logs in 2. Navigate to their respective portal 3. Click Text to Video Converter 4. Upload research article to be converted
Pre-Conditions:	Having a summary article and having a basic/pro account
Post-Conditions:	Having a video generated from their research article
Description:	This function can create the short video from an executive summary by using artificial intelligence.

ID:	6
Name	Account Settings
Actor(s):	Researcher, Admin
Flow of Events	1. User Navigates to Welcome Page

	2. Sign in 3. Redirect to Profile Page
Pre-Conditions:	Users must login
Post-Conditions:	NA
Description:	The page will show the information about users and users can change their profile on this page.

ID:	6.1
Name	Add/Edit/Delete Account Info
Actor(s):	Researcher, Admin
Flow of Events	Navigate to user profile page
Pre-Conditions:	Must login first and have an account
Post-Conditions:	Update information
Description:	This function is used to edit account information.

ID:	7
Name	Watch Video
Actor(s):	Researcher, Admin
Flow of Events	Navigate to video page

Pre-Conditions:	After searching by keywords
Post-Conditions:	The video will be displayed
Description:	This function is used to watch video.

ID:	8
Name	View Profile
Actor(s):	Researcher, Admin
Flow of Events	Navigate to user profile page
Pre-Conditions:	Must login first
Post-Conditions:	All the profile information will be displayed
Description:	User can view profile information through this process

ID:	9
Name	About Us
Actor(s):	Researcher, Admin, (User)
Flow of Events	1. User Navigates to Welcome Page 2. Redirect to About Us Page
Pre-Conditions:	NA

Post-Conditions:	NA
Description:	This page is for users to get familiar with the company and ResearchTube team.

ID:	9.1
Name	Company Description
Actor(s):	Researcher, Admin, (User)
Flow of Events	1. User Navigates to Welcome Page 2. Redirect to About Us Page
Pre-Conditions:	NA
Post-Conditions:	NA
Description:	This part introduces the ResearchTube company.

ID:	9.2
Name	Demo Video
Actor(s):	Researcher, Admin, (User)
Flow of Events	1. User Navigates to Welcome Page 2. Redirect to About Us Page 3. Open the video
Pre-Conditions:	NA

Post-Conditions:	NA
Description:	Demo video is a short video which introduces the ResearchTube.

ID:	9.3
Name	Feedback
Actor(s):	Researcher, Admin, (User)
Flow of Events	1. User Navigates to Welcome Page 2. Redirect to About Us Page
Pre-Conditions:	NA
Post-Conditions:	NA
Description:	Feedback is the comments about the ResearchTube from other users.

ID:	9.4
Name	Information about the owners
Actor(s):	Researcher, Admin, (User)
Flow of Events	1. User Navigates to Welcome Page 2. Redirect to About Us Page
Pre-Conditions:	NA

Post-Conditions:	NA
Description:	This part introduces the owners of the ResearchTube company.

ID:	9.5
Name	ResearchTube Team
Actor(s):	Researcher, Admin, (User)
Flow of Events	1. User Navigates to Welcome Page 2. Redirect to About Us Page
Pre-Conditions:	NA
Post-Conditions:	NA
Description:	This part introduces the whole team of the ResearchTube company.

ID:	10
Name	Contact Us
Actor(s):	Researcher, (User)
Flow of Events	1. User Navigates to Welcome Page 2. Redirect to Contact Us Page
Pre-Conditions:	NA

Post-Conditions:	NA
Description:	The Contact Us page shows the location about the ResearchTube company and users can submit the feedback.

ID:	10.1
Name	Map Location of Headquarters
Actor(s):	Researcher, (User)
Flow of Events	1. User Navigates to Welcome Page 2. Redirect to Contact Us Page
Pre-Conditions:	NA
Post-Conditions:	NA
Description:	This part will show the exact location of the ResearchTube company on the map.

ID:	10.2
Name	Address & Phone Number info
Actor(s):	Researcher, (User)
Flow of Events	1. User Navigates to Welcome Page 2. Redirect to Contact Us Page
Pre-Conditions:	NA

Post-Conditions:	NA
Description:	This part shows the exact address and the phone number of the ResearchTube company.

ID:	10.3
Name	Inquiry Form
Actor(s):	Researcher, (User)
Flow of Events	Navigate to the Inquiry Form page
Pre-Conditions:	Must login as an user or researcher
Post-Conditions:	Upload the form to the database and store it.
Description:	This form is used to collect information

ID:	11
Name	Admin Portal
Actor(s):	Admin
Flow of Events	Navigate to admin login page
Pre-Conditions:	Must logout before this process
Post-Conditions:	NA
Description:	This function is used to give admins a convenient gate to manager and fix the website

ID:	12
Name	View member list
Actor(s):	Admin
Flow of Events	Navigate to the admin manager page
Pre-Conditions:	Must login with an admin account
Post-Conditions:	The member are displayed as a list
Description:	This process is used to check how many members the website have

ID:	12.1
Name	Upgrade Member to Admin
Actor(s):	Admin
Flow of Events	Navigate to admin management page
Pre-Conditions:	Must login with an admin account
Post-Conditions:	Upgrade normal researcher to administrator
Description:	This process is used to upgrade a researcher to an administrator position.

ID:	12.2

Name	View Member Information
Actor(s):	Admin
Flow of Events	Navigate to the admin manager page
Pre-Conditions:	Must login with an admin account
Post-Conditions:	The member information are displayed as a list
Description:	This process can check how many members the website have and the corresponding users information

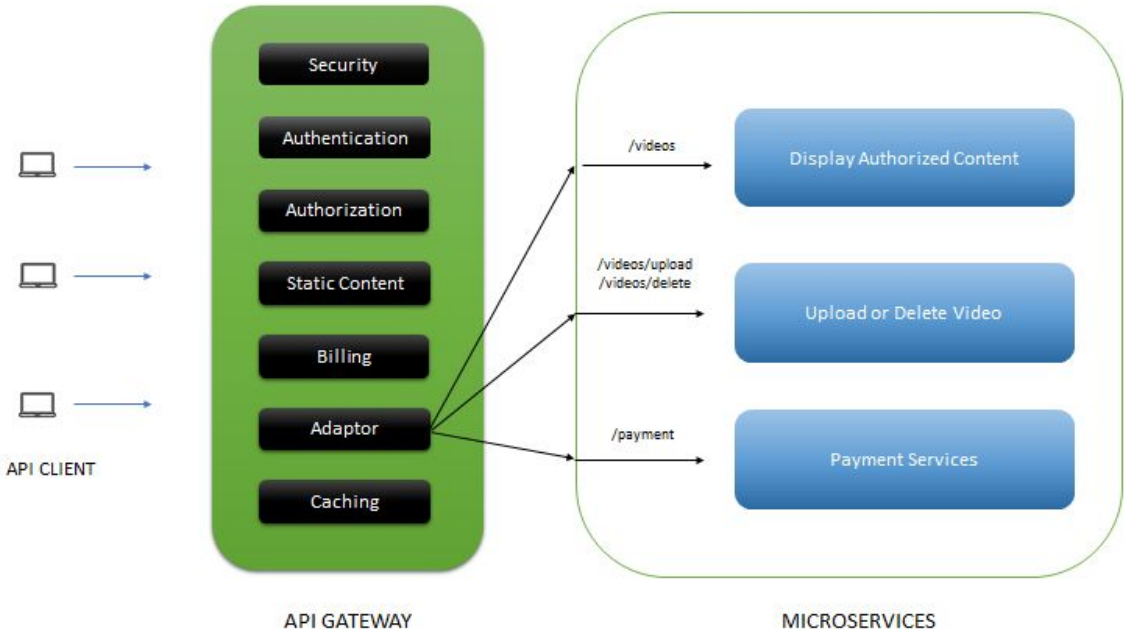
ID:	12.3
Name	Remove User
Actor(s):	Admin
Flow of Events	Navigate into the admin control page
Pre-Conditions:	Must login with an admin account
Post-Conditions:	The users related information will be deleted
Description:	This process is used to remove user information from registered users.

ID:	12.4
Name	Upgrade Researcher Plan
Actor(s):	Admin

Flow of Events	Navigate to admin management page
Pre-Conditions:	Must login with an admin account
Post-Conditions:	Upgrade registered user to a better plan
Description:	This function is to manually upgrade a user to a better plan. This could be an upgrade to either basic or pro plan.

ID:	13
Name	View Feedback
Actor(s):	Admin
Flow of Events	Navigate the feedback page
Pre-Conditions:	Must login with admin's account.
Post-Conditions:	The feedback are displayed as a list
Description:	This function is used to help admins to collect errors and bugs. It also can be used to collect ideas from users, which is helpful for updating the website's user interface and functions.

System Architecture



Api Client: A client will be sending a request to the application.

Api Gateway: All the requests will be handled by api gateway (entry point).

Security: SSL certificate will be added to make sure that all the requests are secure(https).

Authentication: Users will be able to browse the whole web-app. They can see the Home page, Videos page, contact us, and about page. But to watch the videos, users must be logged in.

Authorization: Users must be authorized before they will upload a video or update/delete their content.

Static Content: Whenever there is a request for static content(html, js, and css) Api gateway will handle that request. Microservies will be invoked.

Billing: This microservice will be taking care of all the payment related requests. It also takes care of upgrades(if a user upgrades his/her account from basic to pro).

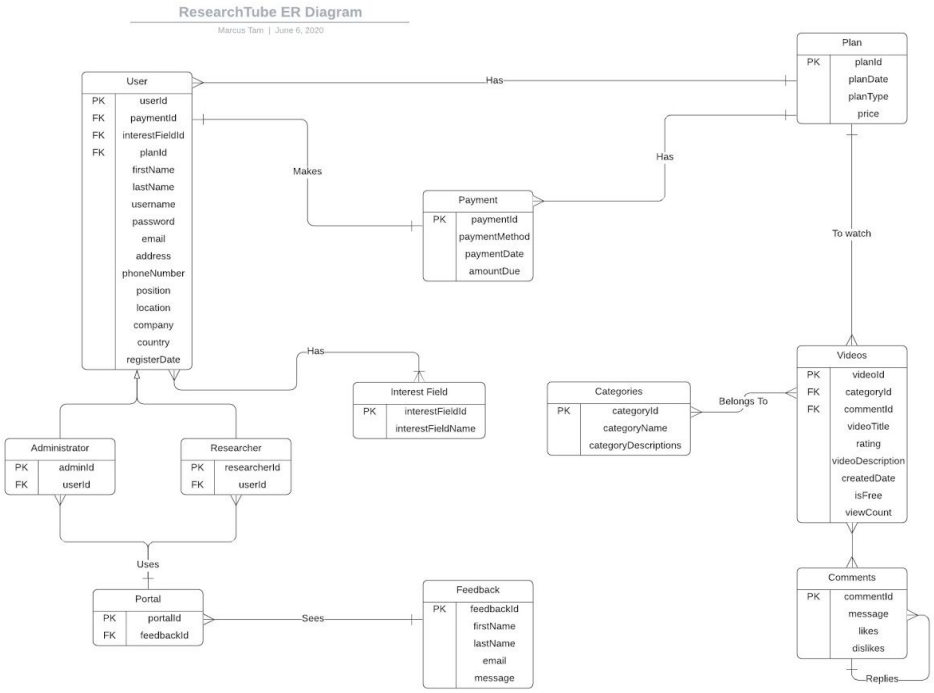
Adaptor: The adaptor will be used as a main controller. It routes all the api calls to the related services.

Caching: Whenever there are multiple requests for a particular service, the caching will be used to avoid calling the server every time. The result will be cached for an hour.

Microserviese:

Display Authorized Content: A user can see the videos as per his/her access level.
Upload/Delete Video: Upload videos to the server.
Payment Service: Handle payment and upgrade access level.

ER Diagram



User

A user here is a super class to both researchers and administrators. To be able to create an account, the user will have to provide some generic information (most information can be extrapolated from a google or facebook alternative sign in / up), some additional information required would be the users highest academic degree, place of work, location of where they work, etc. Once the user submits their account information, they will have to confirm by email. Only then will we enter their information into our database. Upon a successful sign up, an unique ID will be assigned to the account.

Attribute	**Description**
userId: INT {PK}	The user's (Researcher or Admin) unique ID assigned to the account upon account creation
paymentId: INT {FK}	The user's (Researcher or Admin) unique payment ID assigned to the account upon payment registration
interestFieldId: INT {FK}	The user's (Researcher or Admin) unique interest field ID associated to their interests
planId: INT {FK}	The user's (Researcher or Admin) unique plan ID assigned to account upon account creation
firstName: VARCHAR(50)	The user's (Researcher or Admin) first name
lastName: VARCHAR(50)	The user's (Researcher or Admin) last name
username : VARCHAR(50)	The user's (Researcher or Admin) username
password: VARCHAR(50)	The user's (Researcher or Admin) password
email : VARCHAR(50)	The user's (Researcher or Admin) email
address : VARCHAR(50)	The user's (Researcher or Admin) address
phoneNumber : VARCHAR(50)	The user's (Researcher or Admin) phone number

position : VARCHAR(50)	The user's (Researcher or Admin) highest achieved academic degree
location : VARCHAR(50)	The user's (Researcher or Admin) location of which they work
company : VARCHAR(50)	The user's (Researcher or Admin) workplace name
country: VARCHAR(50)	The user's (Researcher or Admin) country of residence
registerDate: DATE	The day the user created their account. Auto-generated

Interest Field

This interest field table is associated with the user. A user will input all their academic interests in this table. This will be used to personalize the home screen for the particular user. It will only display videos that are aligned with their interests.

ATTRIBUTE	**DESCRIPTION**
interestFieldId: INT {PK}	The user's (Researcher or Admin) unique ID assigned to the account upon account creation
interestFieldName: VARCHAR(50)	The user's interests in the academic field. The user can have multiple interests.

Administrator

This table is mainly to distinguish between the researchers and administrators. Here each administrator will be given a unique ID while using their user identification as reference.

ATTRIBUTE	**DESCRIPTION**
adminId: INT {PK}	The user's (Researcher or Admin) unique admin ID assigned to the account upon account creation

userId: VARCHAR(50)	The user's (Researcher or Admin) unique user ID assigned to the account upon account creation

Researcher

This table is mainly to distinguish between the researchers and administrators. Here each researcher will be given a unique ID while using their user identification as reference.

ATTRIBUTE	**DESCRIPTION**
researcherId: INT {PK}	The user's (Researcher or Admin) unique researcher ID assigned to the account upon account creation
userId: VARCHAR(50)	The user's (Researcher or Admin) unique user ID assigned to the account upon account creation

Portal

Each portal will be personalized for each user. The portal will be used to do a variety of things. This is mainly to distinguish from an admin portal and researcher portal as admin portals will have more functionality.

ATTRIBUTE	**DESCRIPTION**
portalId: INT {PK}	The portal's unique identifier. This is to make sure we output the right portal to its respective user.
feedbackId: INT{FK}	The feedback's unique identifier. This is to be displayed on the admins portal.

Feedback

The feedback is given from either a random user (somebody not logged into ResearchTube) or a researcher. Somebody will be able to give ResearchTube either a positive or negative rating. This will prompt the user for some generic information (name and email), and lastly the user will leave a message for the admin to read.

ATTRIBUTE	**DESCRIPTION**
feedbackId: int [PK]	The unique identifier of the feedback given
firstName: VARCHAR(10)	The first name of the user who gave feedback
lastName: VARCHAR(50)	The last name of the user who gave feedback
email: VARCHAR(50)	The email address of the user who gave feedback
message: VARCHAR(255)	The feedback message itself

Plan

A user will have 3 different types of access level as per the plan he/she purchases. The plan table will have a planId as a primary key. There will be other fields in the table to store information related to plans as follows: planType, planDate, and planPrice.

ATTRIBUTE	**DESCRIPTION**
planId: INT [PK]	The plan’s unique id
videoType: VARCHAR(10)	This field will identify the type of plan(Basic, Standard, and Premium)
plantype: VARCHAR(50)	Description of plan type(free, basic, or pro)
planDate: DateTime	Date of the plan is purchased
planPrice: decimal	Price of the plan as per the plantype

Videos

A video will have 3 different labels as per the plan type: Basic, Standard, adn Pro. The video table will store all information related to a particular video. For instance, what is the video category, type, tile, rating and description. Furthermore, there will be a commentId to link with the comment table.

ATTRIBUTE	**DESCRIPTION**
videoId: INT [PK]	The video's unique id
planId: INT [FK]	Plan id will be added to link with the plan table.
videoType: VARCHAR(10)	This field will identify the type of plan(Basic, Standard, and Premium)
categoryId: INT [FK]	Category id will be foreing key
commentId: INT [FK]	Comment id will be foreign key
videoTitle: decimal	Description of video title
rating: INT	Rating will help to identify the popularity of the video
videoDescription: VARCHAR(300)	User will be able to write the description for a video
createdDate: DateTime	Stores the video's creation date
viewCount: INT	This will store the number of view counts for a particular video

Category

A category table will belong to the video table. This table will store the video categoryid, name and description. As per the category information, the videos will be sorted and displayed on the screen.

ATTRIBUTE	**DESCRIPTION**
categoryId: INT {PK}	The category''s unique id

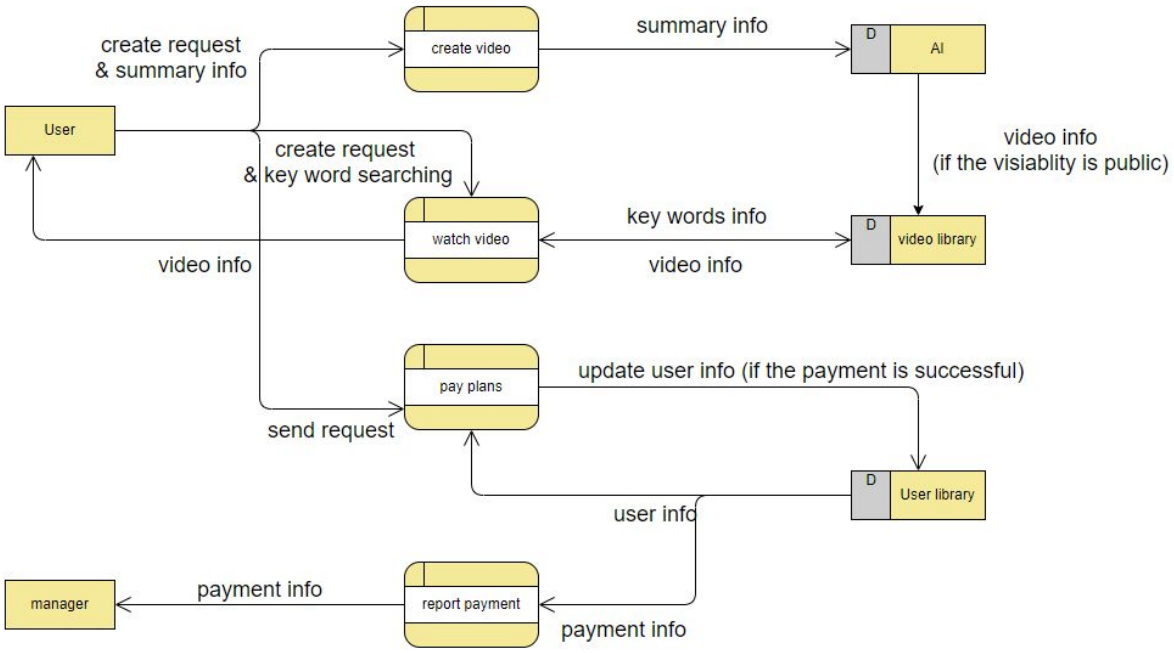
categoryName: VARCHAR(100)	Category name will be stored in this field
categoryDescription: VARCHAR(250)	Description can added and will be limited to 250 characters

Comment

This table will store all the information related to comment, likes and dislikes with respect to a video. A user will be able to comment that will be limited to 300 words.

ATTRIBUTE	**DESCRIPTION**
commentId: INT [PK]	The comment's unique id
message: VARCHAR(300)	A comment of 300 characters will be stored in this field
likes: INT	Number of likes will be stored
dislikes: INT	Number of dislikes will be stored

Data-flow Diagram



Entities:

1) User: users are the customers. When they browse this website, they can do whatever they want.
2) Manager: is the owner of this website.

Processes:

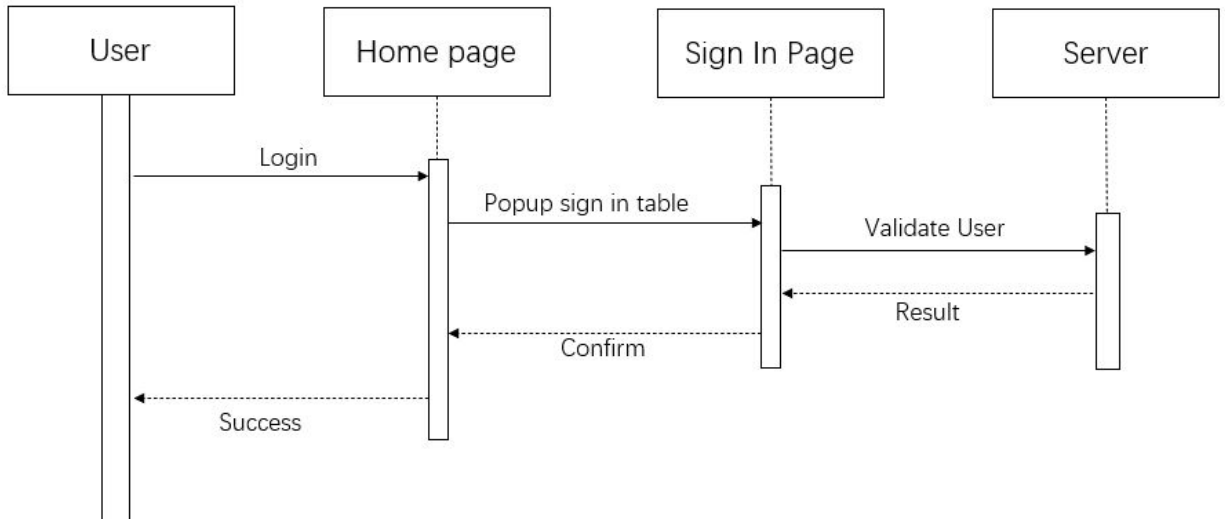
1) Create video: this process is the function used to create video after receiving the request from a user. It allows the user to upload a summary file and choose a preferring style.
2) Watch video: this process is the function used to search and watch relevant videos by key words. It receives the request and key words from the user and sends them to the video library to list corresponding videos, then receive them from the library and send the video information back to the user. Thus, users can watch videos.

3) Pay plans: this process provides users an opportunity to experience more function in this website by purchasing a periodic plan. A user sends requests to this process and the user database provides the user's information during this process. Once the payment is successful, the process will send the updated user information back to the user database.
4) Report payment: after updating the user's information in the user database, the user database will send the payment information to the website manager during this process.

Data stores:

1) AI: It connects to the database about how to create a video with known requirements. After AI receives the summary process, it will use its own algorithm to analyze and create the corresponding video and send it back to the user.
2) Video library: it is the database to store created videos on this website.
3) User library: it is the database to store users' information.

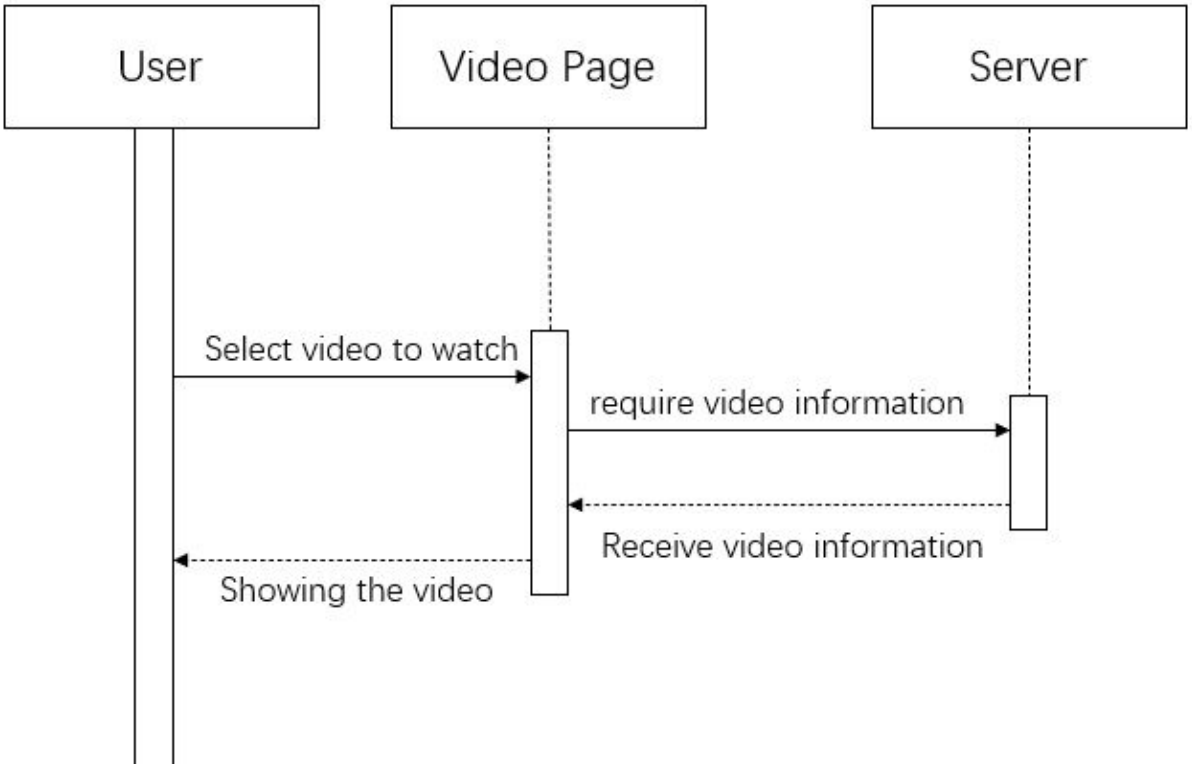
Dynamic Model Diagram



There will be six steps for the user to log in:

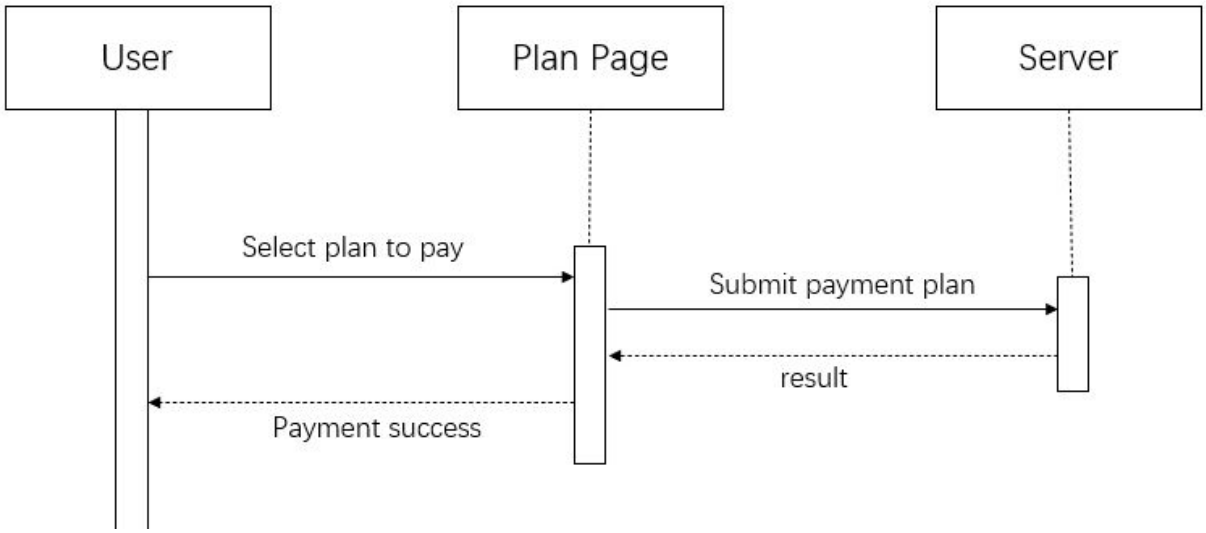
1) Login: users click the login button in the home page
2) Popup sign in table: the home page will show a popup login table for user
3) Validate user: the information that user inputs will pass to the server, and server will check whether the information is valid or not

4) Result: the server will give back the result to the sign in page to show whether the information is valid or not
5) Confirm: the sign in page will confirm this result and go back to home page
6) Success: the home page will show the result to user



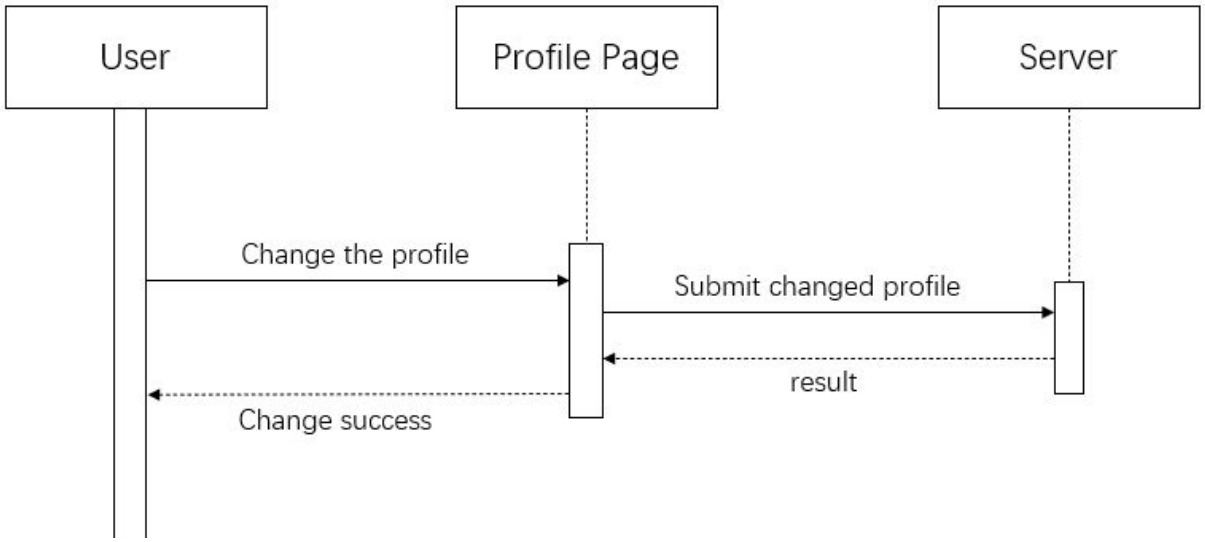
There will be four steps for users to watch video:

1) Select video to watch: users will select a video they want to watch
2) Require video information: video page will request the video information from server
3) Receive video information: the server will give back the information to video page
4) Showing the video: video page will show the video to users



There will be four steps for users to pay their plan:

1) Select plan to pay: users will select the plan they want to pay
2) Submit payment plan: plan page will receive the payment plan and pass it to the server
3) Result: the server will store the plan which user select and give the result back to plan page
4) Payment success: the plan page will show that the payment success to users

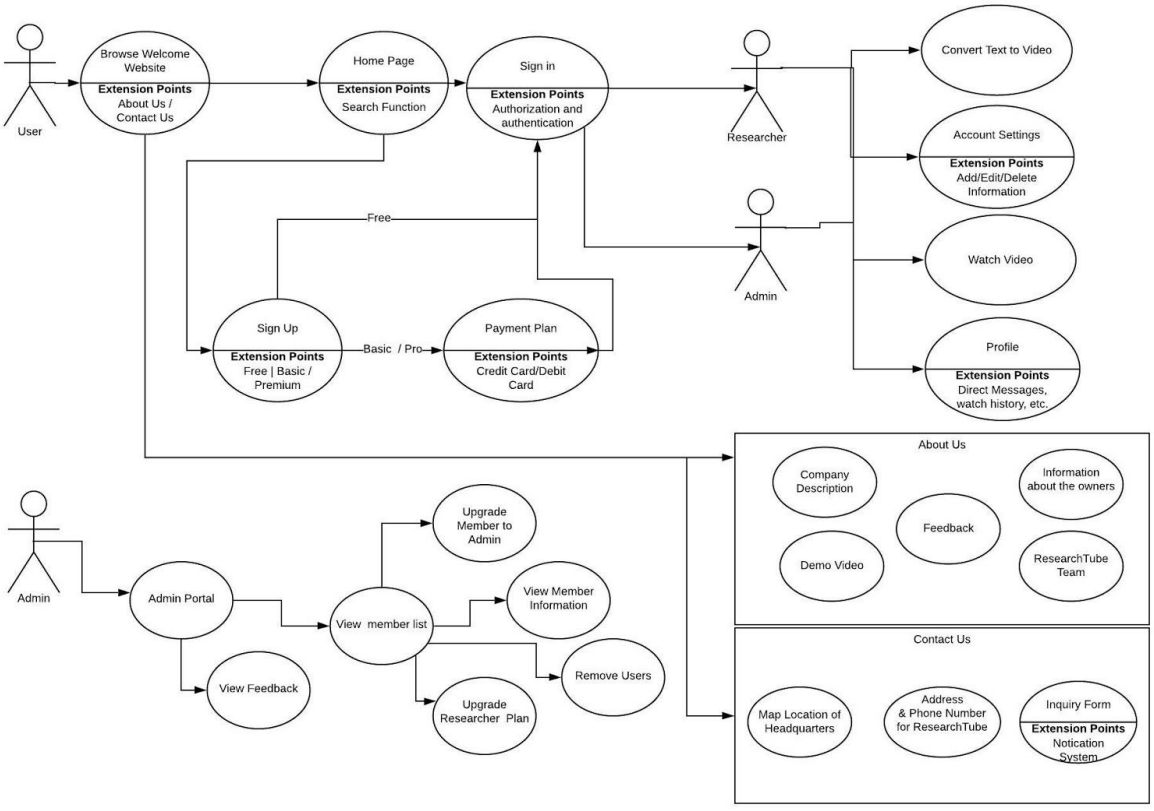


There will be four steps for users to change their profile:

1) Change the profile: users will change their profile in the profile page, and press submit button
2) Submit changed profile: the profile page will submit the information what users change to the server

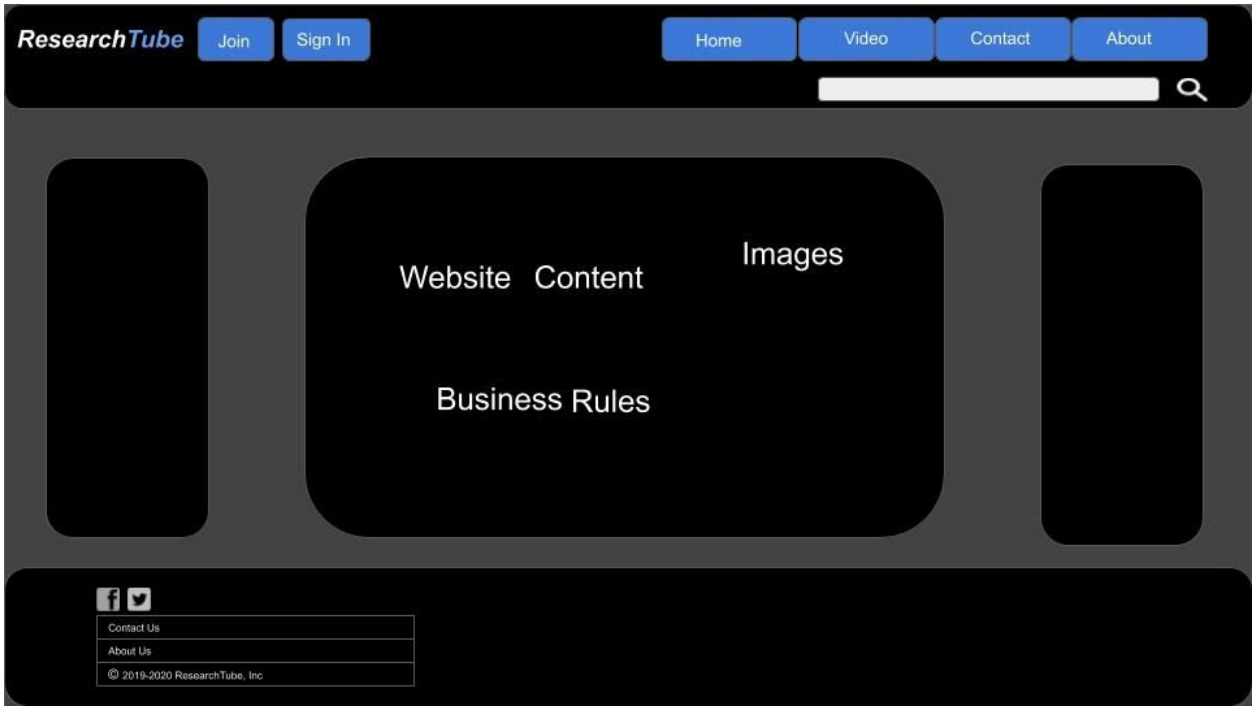
3) Result: the server will give back the result which shows that the change success to the profile page,
4) Change success: the profile page will show the result which shows that the change success to the users

UML Use Case Diagram

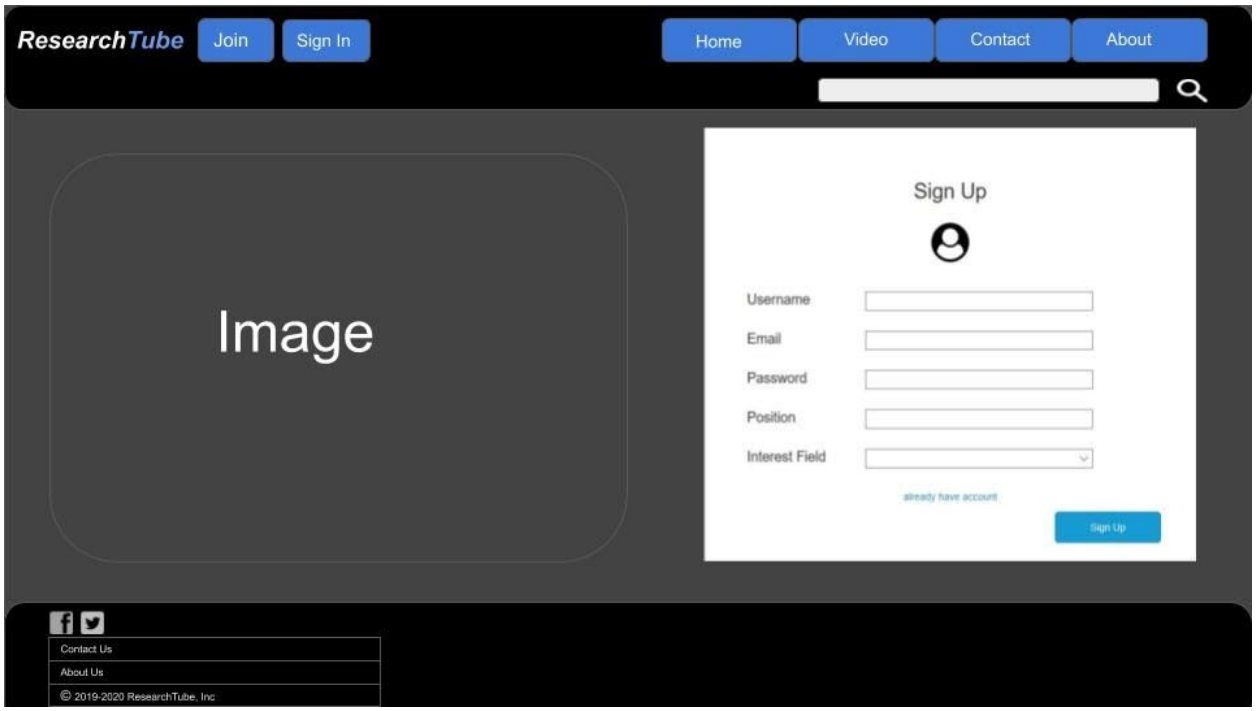


UI Mockups

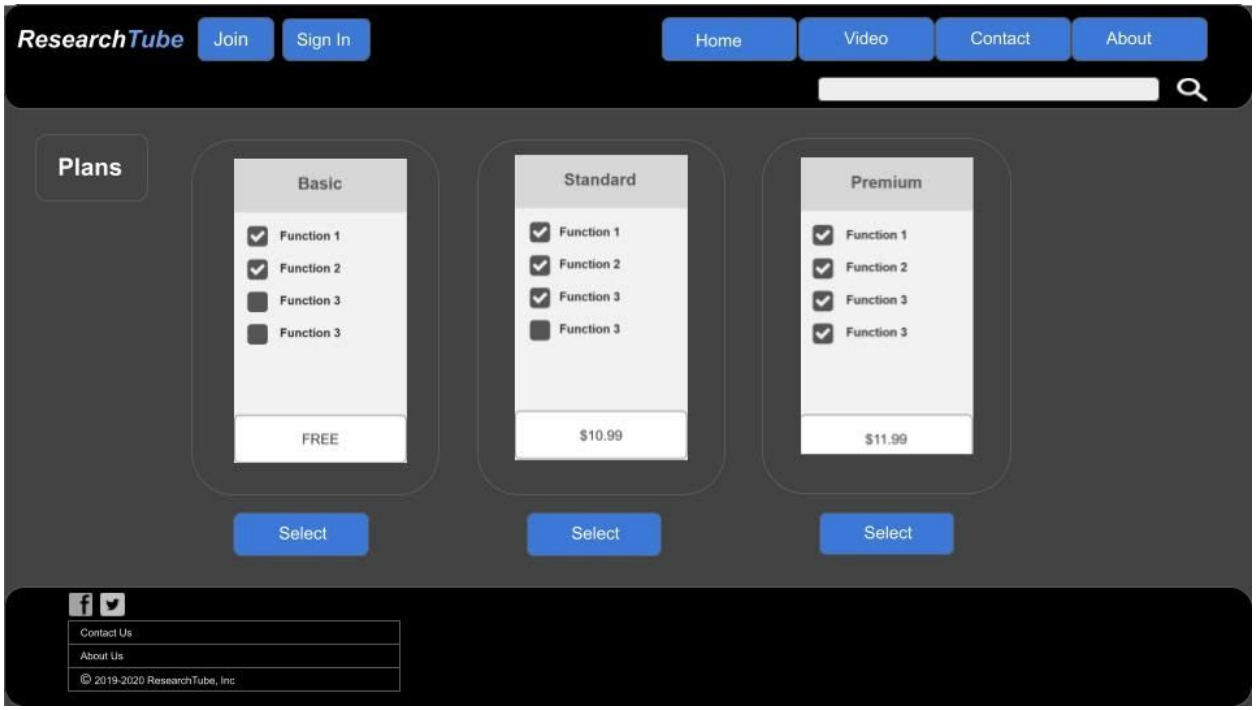
Home Page:

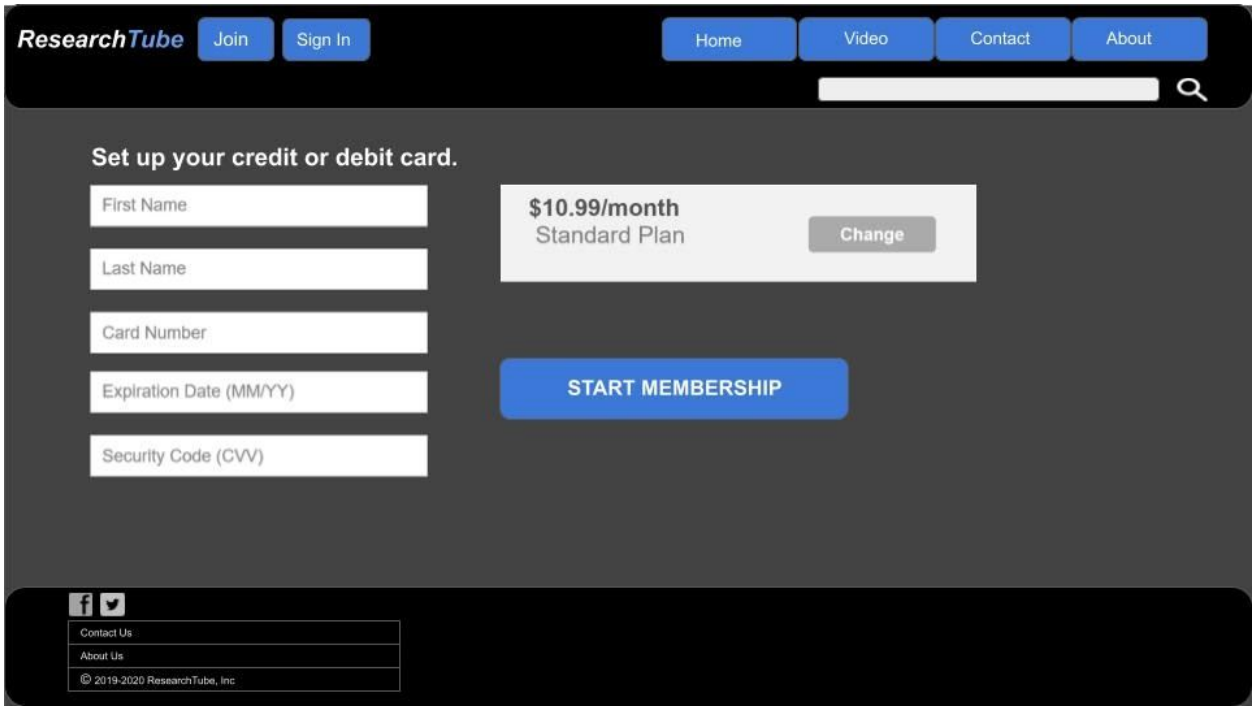


Sign Up Page:

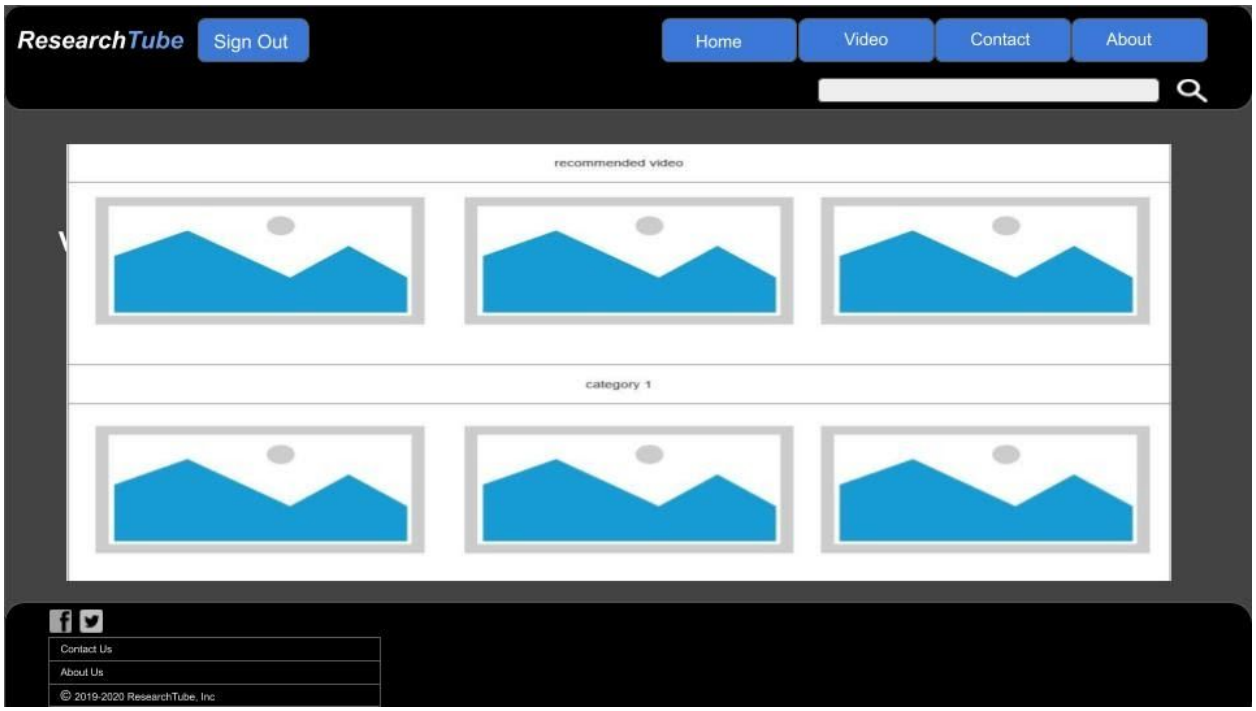

ResearchTube
Join
Sign In
Home
Video
Contact
About
Image
Sign Up
Username
Email
Password
Position
Interest Field
already have account
Sign Up
Contact Us
About Us
© 2019-2020 ResearchTube, Inc

Payment Plan Page:



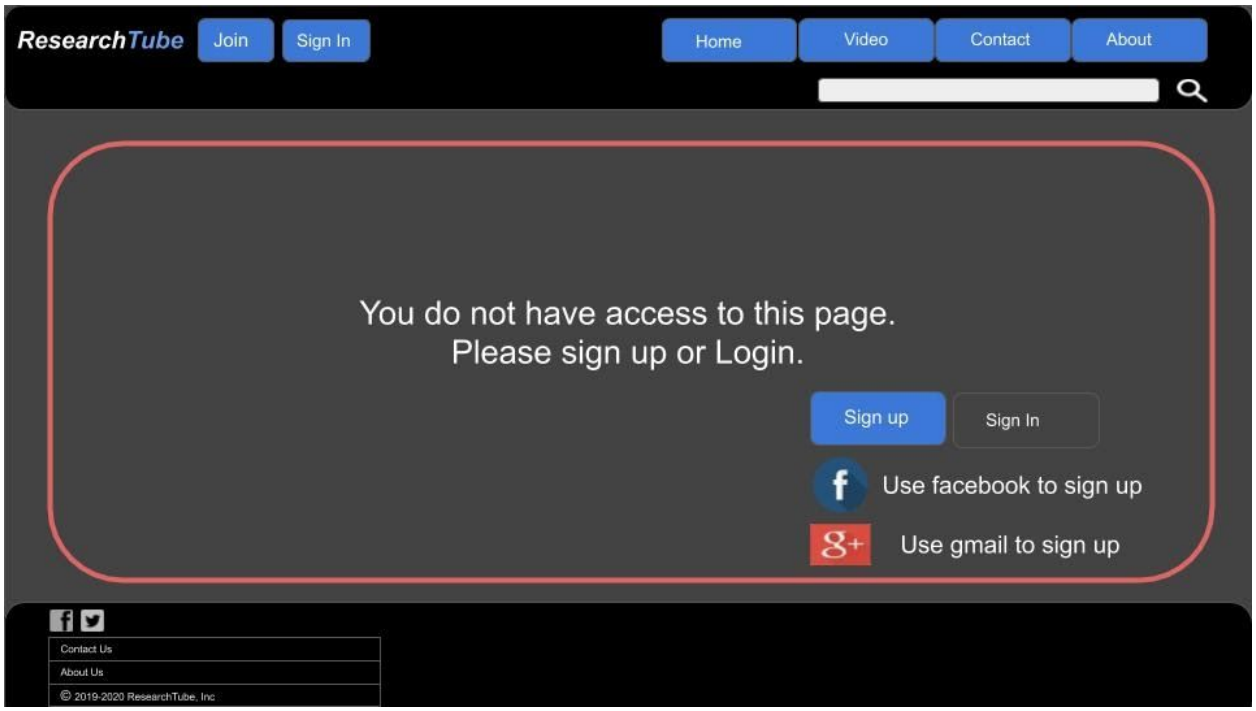


Video Page:

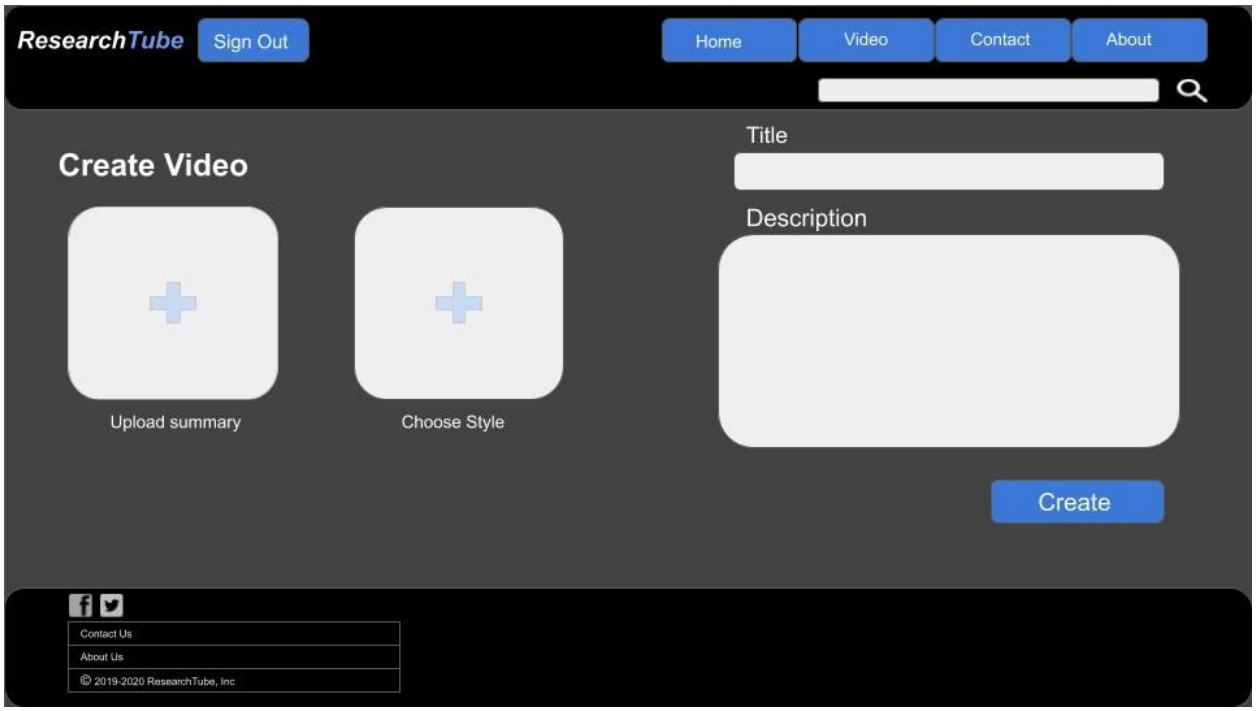

ResearchTube
Sign Out
Home
Video
Contact
About
recommended video
category 1
Contact Us
About Us
© 2019-2020 ResearchTube, Inc

Validation Error

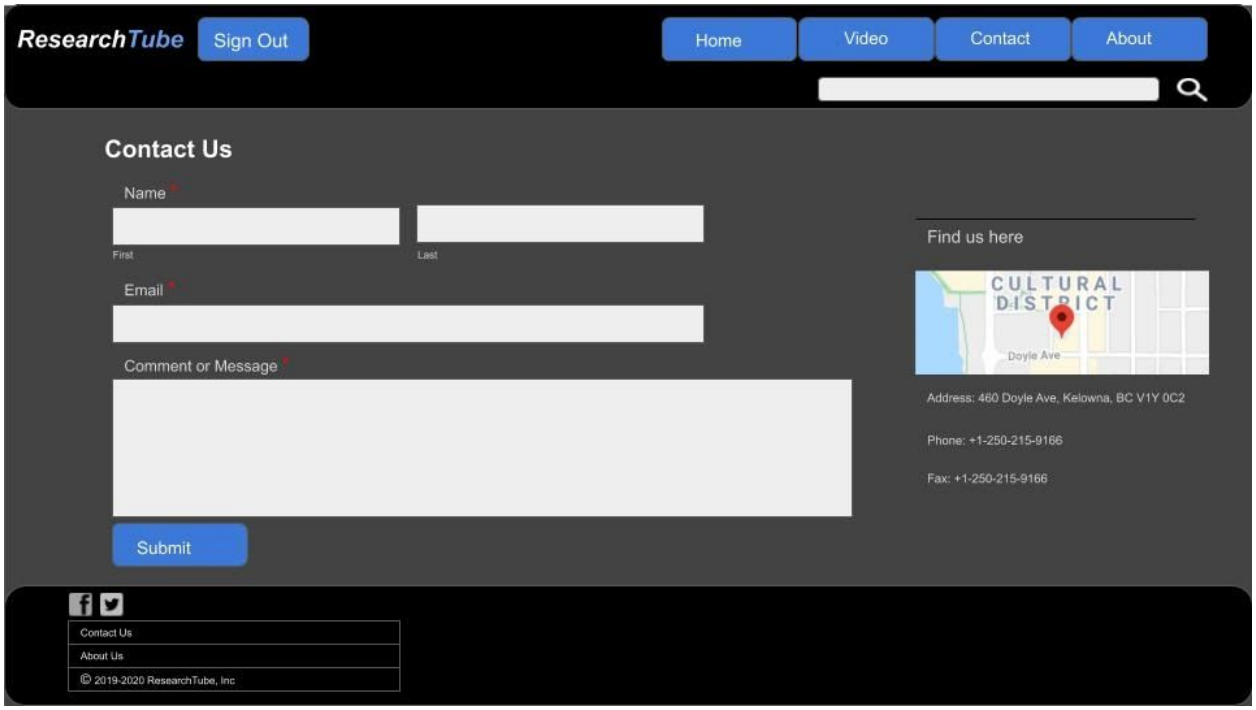
If a user tries to access the videos without login or out of his/her access level. An feedback will be provided and the advice necessary action needs to be taken further.



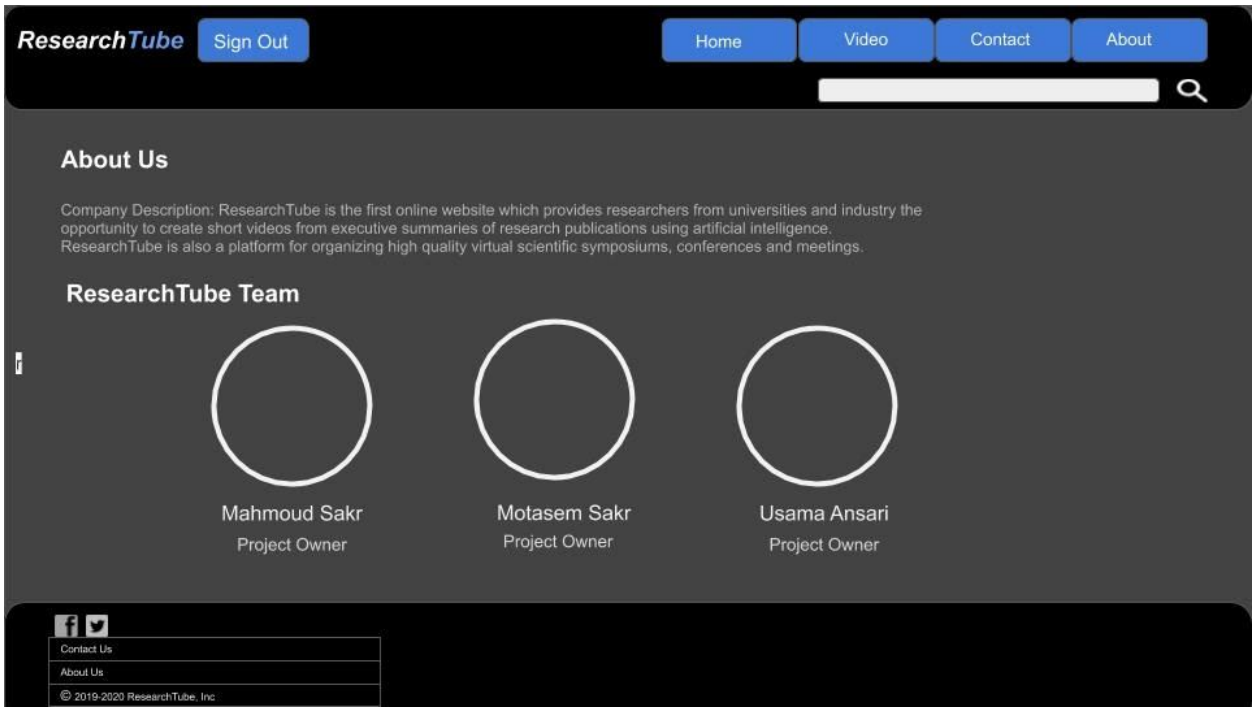
Create Video:



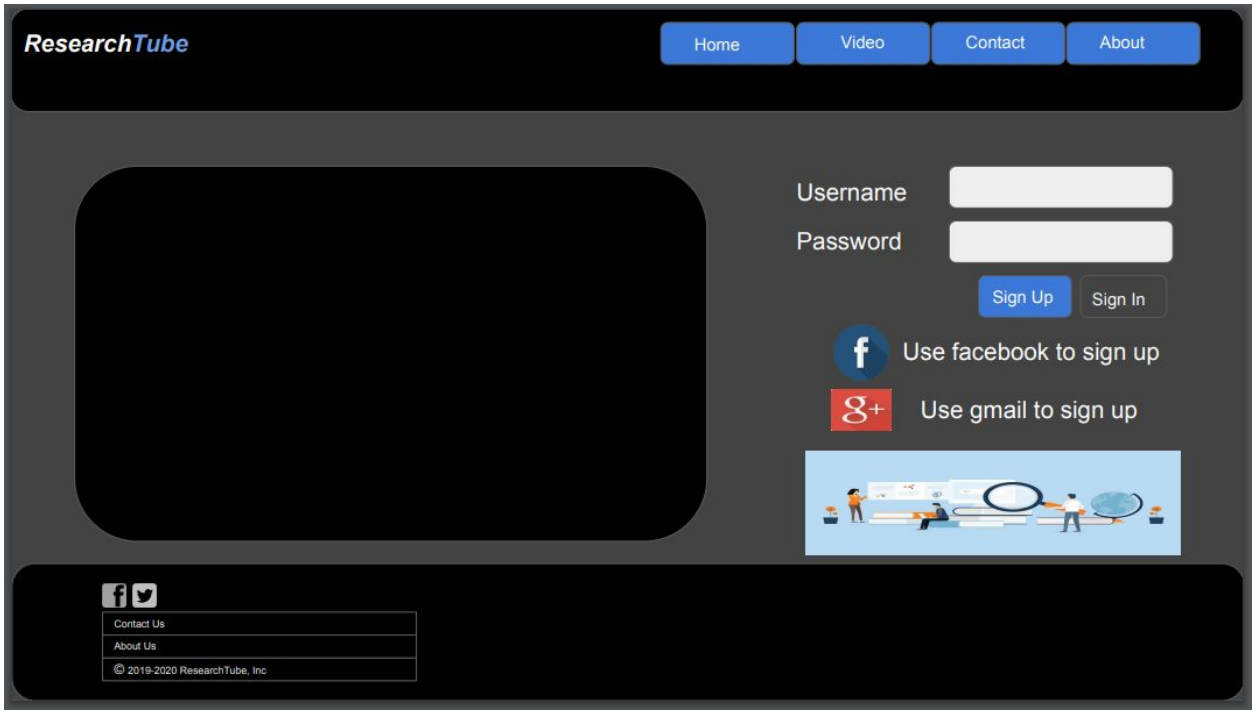
Contact Us Page:



About Us Page:

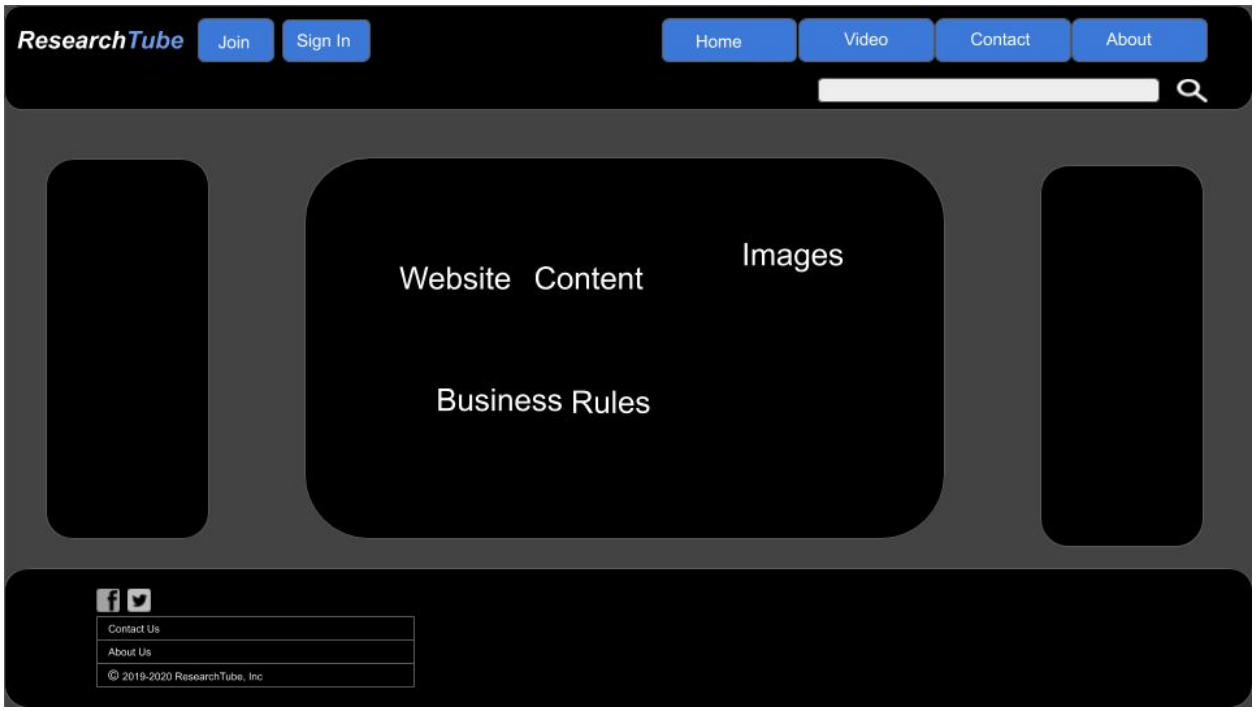


Initial Index Page Design:



Client requested to have join and sign-in buttons rather than traditional style login and sign up.

Updated design:



Header Color: Black - #000000
Background Color: Dark gray - #333333
Footer Color: Black - #000000

Text Color: White-FFFFFF
Body Text Size: 16px
Heading SIze: 18px to 32px - Depend on the content

Button color: Blue - #0033FF
Active button color: Light blue - #0099FF

Technical Specification

- It will require an IIS server/Docker platform to host the application.
- The server that will handle the api requests to handle the data and displaying the content.

- GitHub will be used for version control. When we merge our code into the main branch.
- On GitHub, we will use TravisCI to automate the build process.
- For testing, we will be using Dotnet / xUnit for C# language.
- The web portal will be built using JavaScript, HTML, and CSS, and will be running on the client's personal computer.
- The application will be written in C# with MVC architecture.
- Net core framework will be used.
- Docker will be used for containerization.
- ADO.net data access services will be used for data access/connection.
- SQL server will be used at backend.

Detailed test plan

1. Scope

The purpose of this test plan is to ensure we meet all of our functional, non-functional, technical, and user requirements in a timely and organized manner. Our overall test plan will consist of unit testing, integration testing, performance testing, and bug fixing. Testing is critical to the successful completion of this project, and will be a continuous process throughout its duration

2. Process

We will be using test-driven development to cover human/programmer error, logical error and performance constraints(how long does a server call take to respond). This technique will help us to make sure if the web-app performs its required functions in a robust, correct and efficient manner. These processes are explained in detail below.

Unit Testing

All the functions will be used to fulfill the high-level functional requirements of our software. An example of such a component would be uploadUserData() method to upload data on database or userAuthentication() method to user-authenticate system. Major components will be such as uploadVideo() to upload a video on server or such as getData() to fetch data from server. All the major components should be tested and verified before any coding is undertaken. A unit test evaluates a part of your application's logic. Unit tests will be employed after a new feature/function is established. xUnit will be used as a unit testing tool for this project..

Integration Testing

Integration testing is used to test the functionality of components working together. We will have to write separate test cases from the unit testing. Integration testing will be making sure that adding a new feature will persist the current state of the software and will not have any negative effect. One example of a function performed by many integrated components is testing when a user upgrades his/her plan(basic to pro), he/she must have all the rights as per the new access level. This will require to update user status, access level to the videos, and payment plan. Example: The process for generating integration tests is as follows:

- Identify that existing features will be interacting with the new features without breaking the existing functionality.
- Identify all the edges cases so that adding new features can communicate or provide desired output.
- Before merging new components to the master branch, make sure all the previous test cases pass.

Usability Tests

The usability test is ensuring that users have a good experience with the website. This includes: Ease of learning, Navigation, General Appearance, User Satisfaction. This works along with User Tests but is more focused on the aesthetics. An example would include asking a randomized user their thoughts on the types of buttons or boxes.

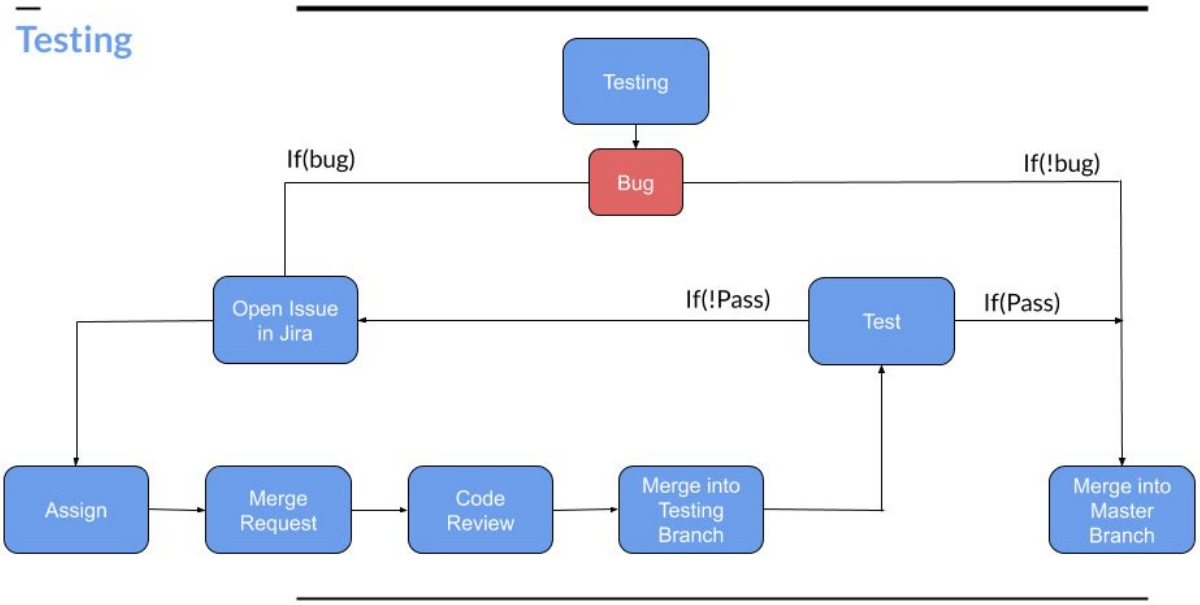
Performance Tests

Performance testing is the process of determining the speed, responsiveness and stability of the software under a workload, in order for us to compare different data structures and algorithms which one has the highest performance.
Example:Performance checks will be as follow:

- We will make sure a 2 minutes video can be uploaded within 30 seconds to 1 minute.
- Server calls should not take too long. If so, we will make sure that there are no promises(javascript promises) deadlock and also the developers will make sure the async methods should not be blocked.
- If a query takes longer to return the result, query optimization will be used. Developers will make sure table joining and where clauses will be used only when required. To do this, where clause and joines will be used as function parameters and use appropriate join to where clause as per the API call.

Bug Life-Cycle

Bugs are inevitable and can be encountered in the software throughout the course of
development and beyond. The process for documenting and correcting standards bugs will be as follows:



- When a bug is encountered, document it immediately in the form of a Jira task, assign it an initial priority.
- Developers will keep checking the Jira application and grab the higher priority task.
- Once the task is done, the developer will request for merging the particular branch.
- After code review, code will be uploaded on the testing branch, and let the QA person know the new code is pushed to the testing branch.
- Once QA gives a go ahead, the integration lead person will be pushing the code into the master branch.

Summary of Tools

C#
.NET Core Framework
MVC Architecture
IIS Server
Docker
Unit Testing / xUnit
JavaScript(Vanilla)
HTML
CSS
SQL Server

Approvals

Project Owner Signature: _____

Pardeep Singh

Project Manager Signature: _____