Project Report On

"Through The Lens : A Software For Images and Footages"

CSE 306 : Software Engineering & Information System Design Laboratory

Department of Computer Science & Engineering
Premier University Chittagong

Submitted By :
Jannatul Ferdous
ID : 2003810202078

Afra Ibnat
ID : 2003810202079

Semester : 5th
Section : A2
Batch : 38

Submitted To :
Md. Neamul Haque
Lecturer , CSE Department,
Premier University Chittagong.

Index

Chapter 1: Project Proposal

An international Platform for photographers and videographers.

Abstract
This project is to offer a platform to creative photographers and videographers to share and showcase their work, the freedom to monetize their particular works, or to give users permission to use their work as a stock image/footage in other creative endeavors.

Introduction

Motivation

Geographically Bangladesh is a beholder of raw beauty. Mother Nature has gifted us with lush green, endless rivers, sea beaches and whatnot. And our people have established a very unique, versatile, colorful and creative lifestyle over the years. We have seen many young photographers in our country showing a keen passion to capture these moments and beauty through their lens. But unfortunately for us, they don't get good enough platforms to showcase their work proudly. This lack of exposure and mostly recognition for their work and a way to monetize them holds them back to embark on this journey full time.

Solution

Recognizing the major problem stated before, we wanted to give these creators a platform to show their work on an international platform.Where people can use their work as a stock image, rate them, save their work and also buy their particularly monetized works! Meanwhile the artist gets the chance to build a strong profile, get feedback , and also sell them. If the artist makes their work free, the users can download the photos and footage free of copyright, use them , save them for later, rate them, and also buy the monetized works as well.

Proposed Module

The project is divided into four modules :
- Admin Module
- Viewer Module
- User Module
- Pro User module

Admin Module :

- sign the viewers and users in.
- Add or remove features.
- Check the copyright issues and send warnings. If any profile gets it 3 times, the admin can disable the account.
- Admin can remove content if they violate the community terms and policies.

Viewer module:

The viewer can :

● Register using required info (Name, Category Email, Country Name.)
● Log in.
● View photos by searching , or by browsing the most popular category.
● Save photos/ videos for later.
● Download photos in exchange of an honest review about the content (Out of 5 stars)
● Leave comments.
● Chat with the creator.
● Buy monetized content.
● Log out.

User Module:

● Register as a creator using required info (Name, Email, Country, Contact info, scan copy of verified NID)
● Log in
● Update their cover photo, profile photo and Bio
● Can upload their photos and videos. With captions
● Can create albums for their content.
● Also upload the content under specified category (mandatory)
● Their profile will show a average rating based on the rating viewers give them during downloads
● Chat with others.

Premium-User Module:

● There is no requirement to be a pro user. Any academic requirement is irrelevant. They will be judged by their skill.
● Monthly 5 dollar subscription fee.
● If they have their account previously created at the platform, their info will be collected from there. With additional bank account info.
● Sell their albums or content from which the software will get 7% commission
● Log out.

How the monetization will work?

- Creators can access premium features with 5 dollar/month subscriptions fee
- Creators can bid their work to their own accord.
- TTL gets 7% commission for each transaction.

Created with the help of:

Language used:

- HTML
- CSS
- Bootstrap
- PHP

- Js
- MySQL

Conclusion

The mashup of the social media type work showcasing interface (eg pinterest/ instagram) and a strong marketplace - specifically for the magicians behind the lens!

Chapter 2 : Software Development Process Model

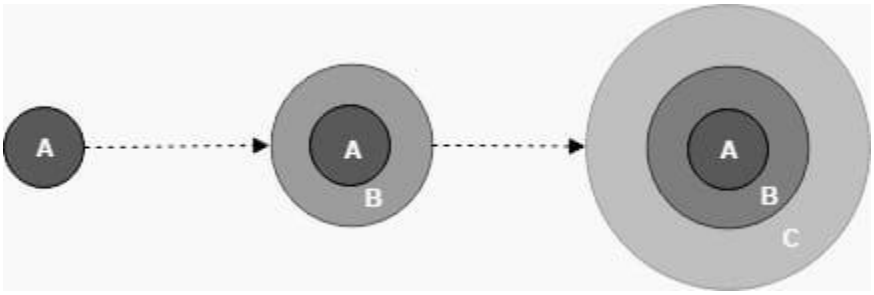
Objective:
To decide on a process model to reduce the complexity and error, and increase the efficiency of developing the proposed system.

Introduction:
Every software development project goes through a complex process of numerous steps, review and reconstruction. The compilation of all the small and key features can easily become overwhelming. Making mistakes, forgetting to add many features, consuming more time than necessary, difficulty in error detection are some very common occurrences if the planning is not good. To avoid all these unwanted hurdles and efficiently develop the software in a systematic way , we are trying to decide on a process model. Process models are compared in accordance with our plan for the software and decided upon by checking which one is most suitable for our project.

Solution:
As our software has a fairly big number of features to add, we compared all the process models in accordance to our requirements. Among them , the incremental process model turns out to be the most efficient and suited for our particular project.

Incremental Process Model:

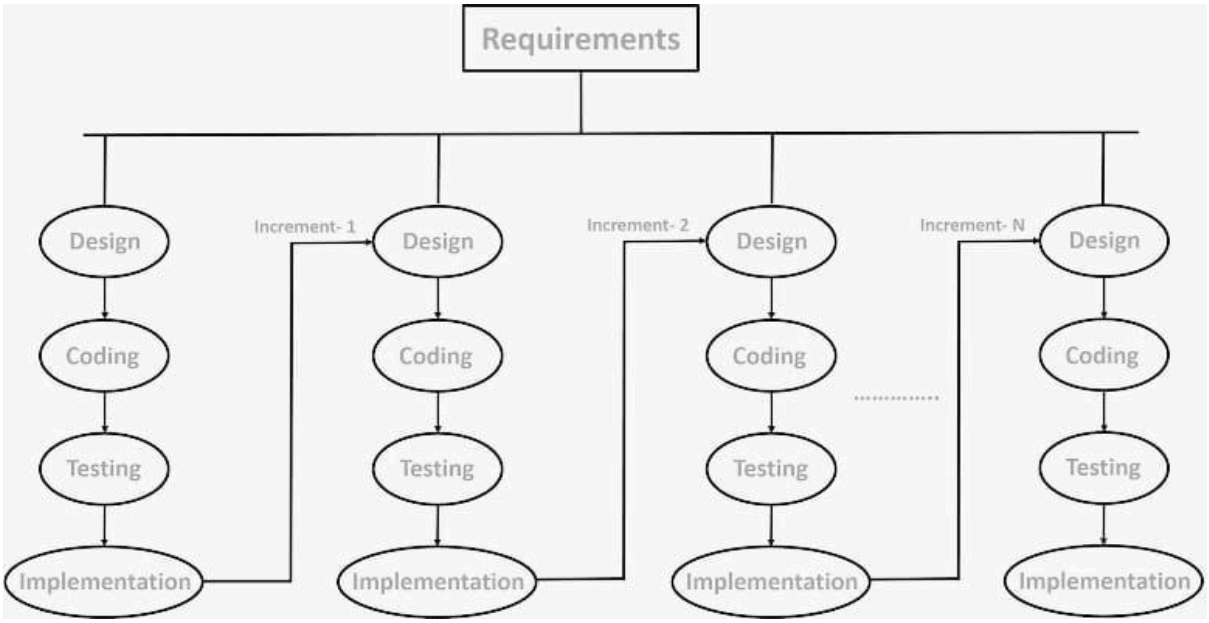
In the incremental process model almost works like a layer upon layer. We first do the key features, make them successfully with feedback, and move onto building the details layer by layer each time. The process looks like the figure given below:



Life cycle activities of the Model:

- Requirements of Software are first broken down into several mini development projects that can be incrementally constructed and delivered.
- At any time, the plan is made just for the next increment rather than a long-term plan.So it is easier to modify according to customer feedback.
- The Development Team first builts the core features of the system.
- Once the core features are fully developed, then the steps of adding details come.
- After each version of the software is built and delivered, the feedback of the Customer is taken and incorporated into the next version.

- Each version of the software has more additional features than the previous ones.



When to use this model :

1. Funding Schedule, Risk, Program Complexity, or need for early release
2. When Requirements are known up-front.
3. When Projects have lengthy development schedules.
4. Requires good planning and design.
5. The total cost is not lower.
6. Well-defined module interfaces are required.

Advantages of the model-

1. Prepares the software fast.
2. Clients have a clear idea of the project.
3. Changes are easy to implement.

Why we are using the incremental process model:

1. Our requirements for the project is known up-front
2. We can break down our project into highest priority key features
3. Breaking the project into mini projects with key features built successfully would be the most effective and efficient way to build the software.
4. Changes will be easy to implement according to clients feedback.
5. Constant feedback and testing will reduce the risk of error.

Disadvantages we may come across:

1. Because of its continuous iterations, the cost increases.

Conclusion :

Considering all the process models and their characteristics, advantages, disadvantages- our chosen model is the incremental model.

Chapter 3 : System Diagram

Objective :
Designing system diagrams for an efficient software development process.

Introduction:
Without a systematic and reasonable plan, every process is bound to become complex. For which the chance of error increases significantly. To avoid that, and make the workflow more efficient, we created different system diagrams such as - ER diagram, Use Case Diagram, and Activity Diagram.

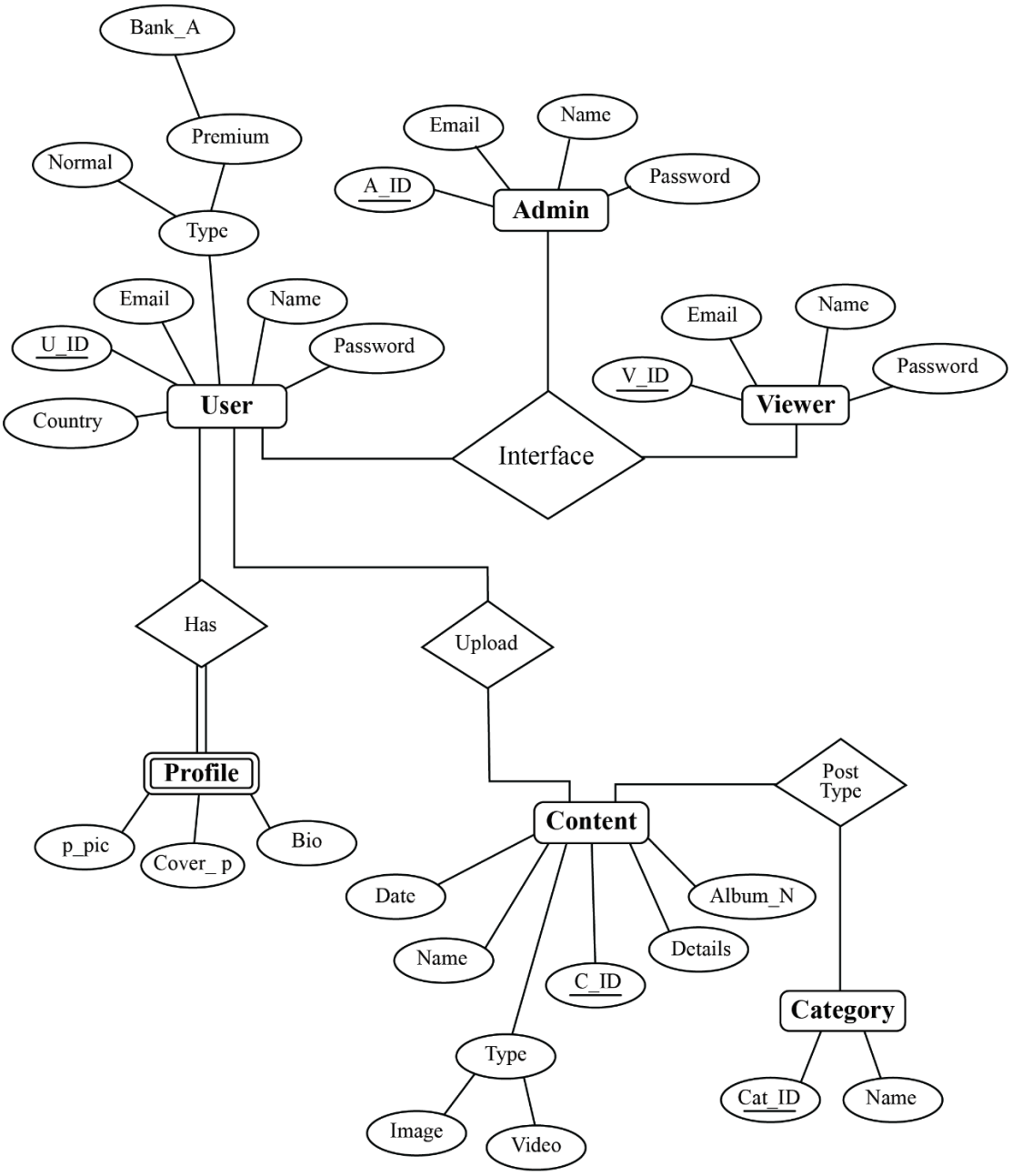
ER diagram:
ER model stands for an Entity-Relationship model. An Entity Relationship (ER) Diagram is a type of flowchart that illustrates how “entities” such as people, objects, or concepts relate to each other within a system.

ER Diagrams are essential for modeling the data stored in a database. It is the basic design upon which a database is built. It is most often used to design or debug relational databases.ER diagrams specify what data we will store: the entities and their attributes. They also show how entities relate to other entities. They use a defined set of symbols such as rectangles, diamonds, ovals, and connecting lines to depict the interconnectedness of entities, relationships, and their attributes.

Why ER diagrams are used:

- It is a high-level data model.
- is used to define the data elements and relationships for a specified system.
- It develops a conceptual design for the database.
- It develops a very simple and easy-to-design view of data.
- In ER modeling, the database structure is portrayed as a diagram called an entity-relationship diagram.

ER diagram for the system of our system:



Use Case Diagram:

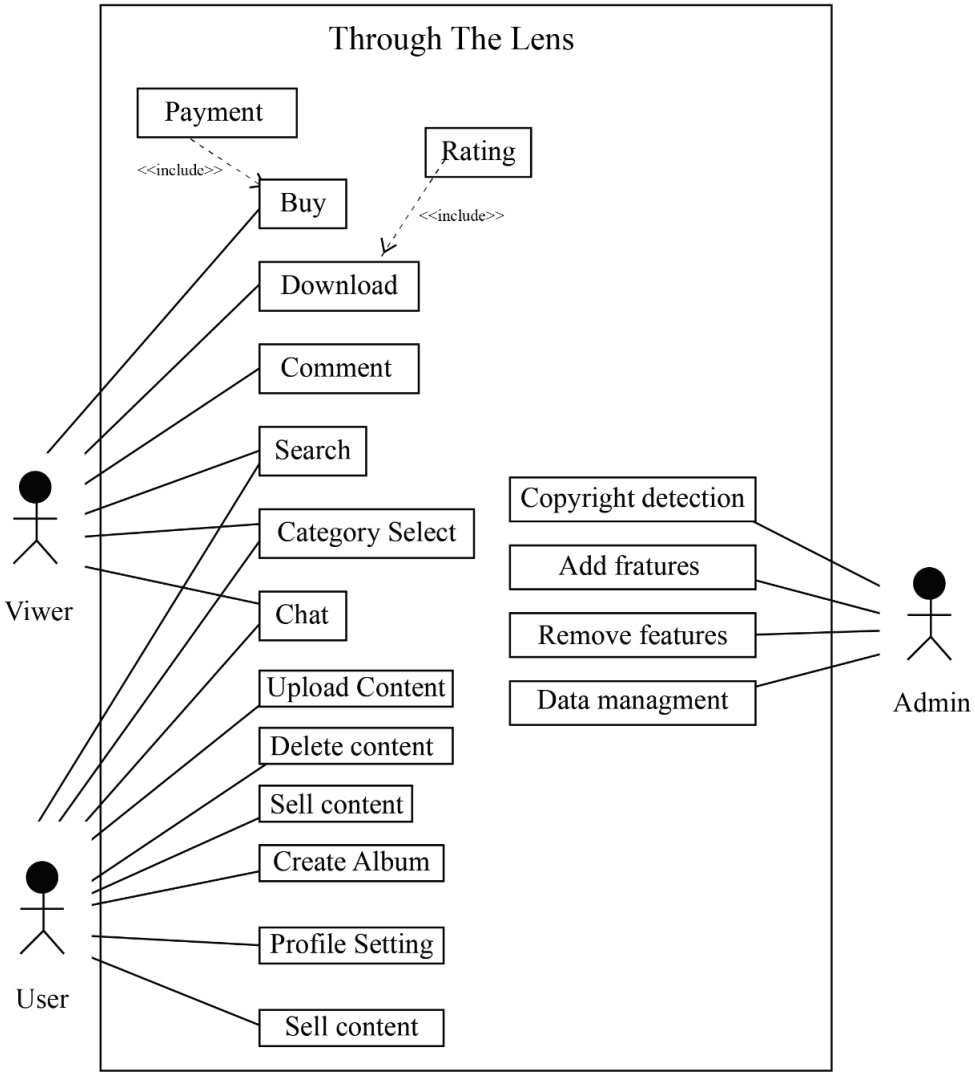
A use case diagram can summarize the details of a system's users/actors and their interactions with the system. Its built by using a set of specialized symbols and connectors. An effective use case diagram can help to discuss and represent :

- Scenarios in which the system or application interacts with people, organizations, or external systems.
- Goals that the system or application helps those entities/ actors achieve
- The scope of your system

When to use Use Case Diagram:

- Represent the goals of systems and users.
- Specify the context in a system should be viewed in.
- Specify system requirements.
- Provide a model for the flow of events when it comes to user interactions.
- Provide an outside view of a system.
- Show external and internal influences on a system.

Use Case Diagram for the system of our software:



Activity Diagram:

Activity Diagrams describe dynamic aspects of the system. An activity diagram is essentially an advanced version of a flowchart that models the flow from one activity to another activity.

When do we Use an Activity Diagram:

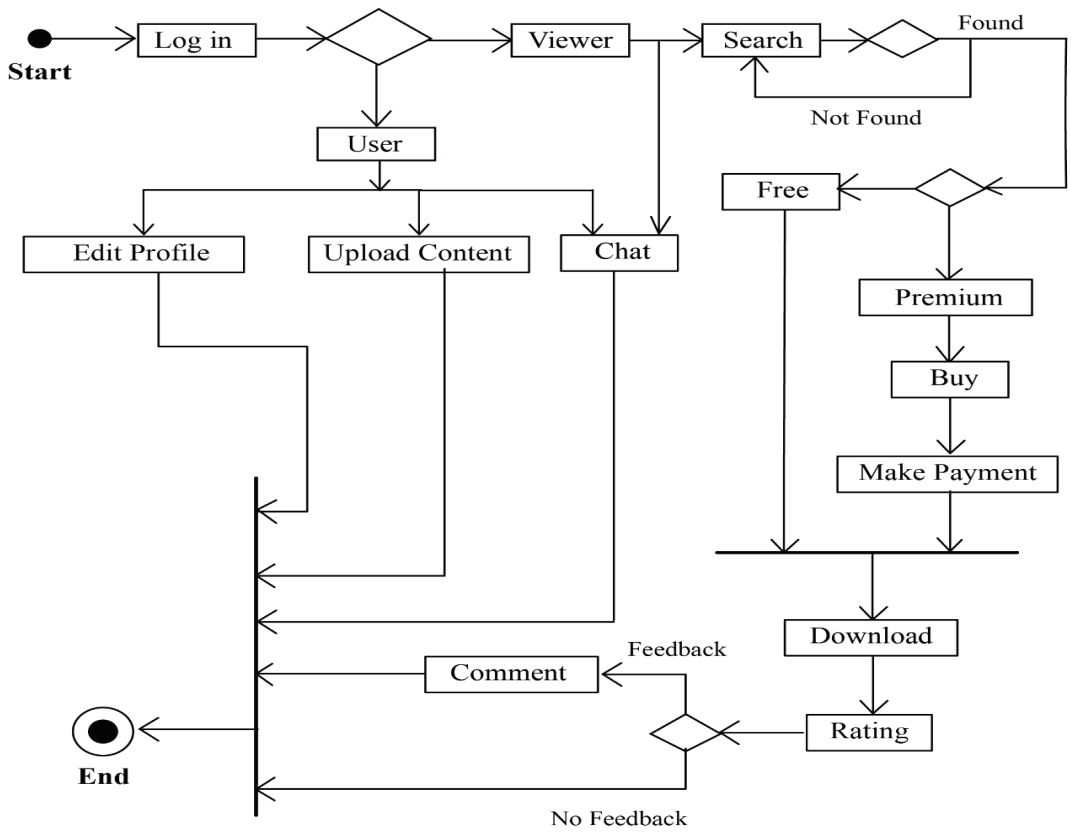
1. To Identify candidate use cases, through the examination of business workflows
2. Identify pre- and post-conditions for use cases
3. Model workflows between/within use cases
4. Model complex workflows in operations on objects

The advantage of activity diagrams:

Can be used to:

- capture flow from one system to another and include capabilities
- Illustrate a business process or workflow between users and the system.
- Simplify and improve any process by clarifying complicated use cases.
- Model software architecture elements, such as method, function, and operation.

Activity Model for the system of our system:



Conclusion :

We developed an ER, Use Case and Activity diagram to reduce the complexity and increase the clarity of the development process of our software system.

Chapter 4 : Class Diagram

Objective :
Designing a class diagram for an efficient software development process.

Class Diagram :
A class represents the main program unit in object-oriented programming. Classes are data structures that contain code and represent objects in an application. A class diagram is a visual representation of class objects in a model system, categorized by class types. Each class type is represented as a rectangle with three compartments for the class name, attributes, and operations.

Benefits of using a class diagram

Using a class diagram is a good way for businesses to communicate a class structure to project stakeholders and teams. A class diagram is especially useful for communicating class hierarchies and collaborations between classes.

Class Notation

A class notation consists of three parts:

1. **Class Name**
    - The name of the class appears in the first partition.
2. **Class Attributes**
    - Attributes are shown in the second partition.
    - The attribute type is shown after the colon.
    - Attributes map onto member variables (data members) in code.
3. **Class Methods**
    - Operations are shown in the third partition. They are services the class provides.
    - The return type of a method is shown after the colon at the end of the method signature.
    - The return type of method parameters is shown after the colon following the parameter name.
    - Operations map onto class methods in code

Relationships between Classes

- **generalization:** an inheritance relationship
    - inheritance between classes
    - interface implementation
- **association:** a usage relationship
    - dependency
    - aggregation
    - composition

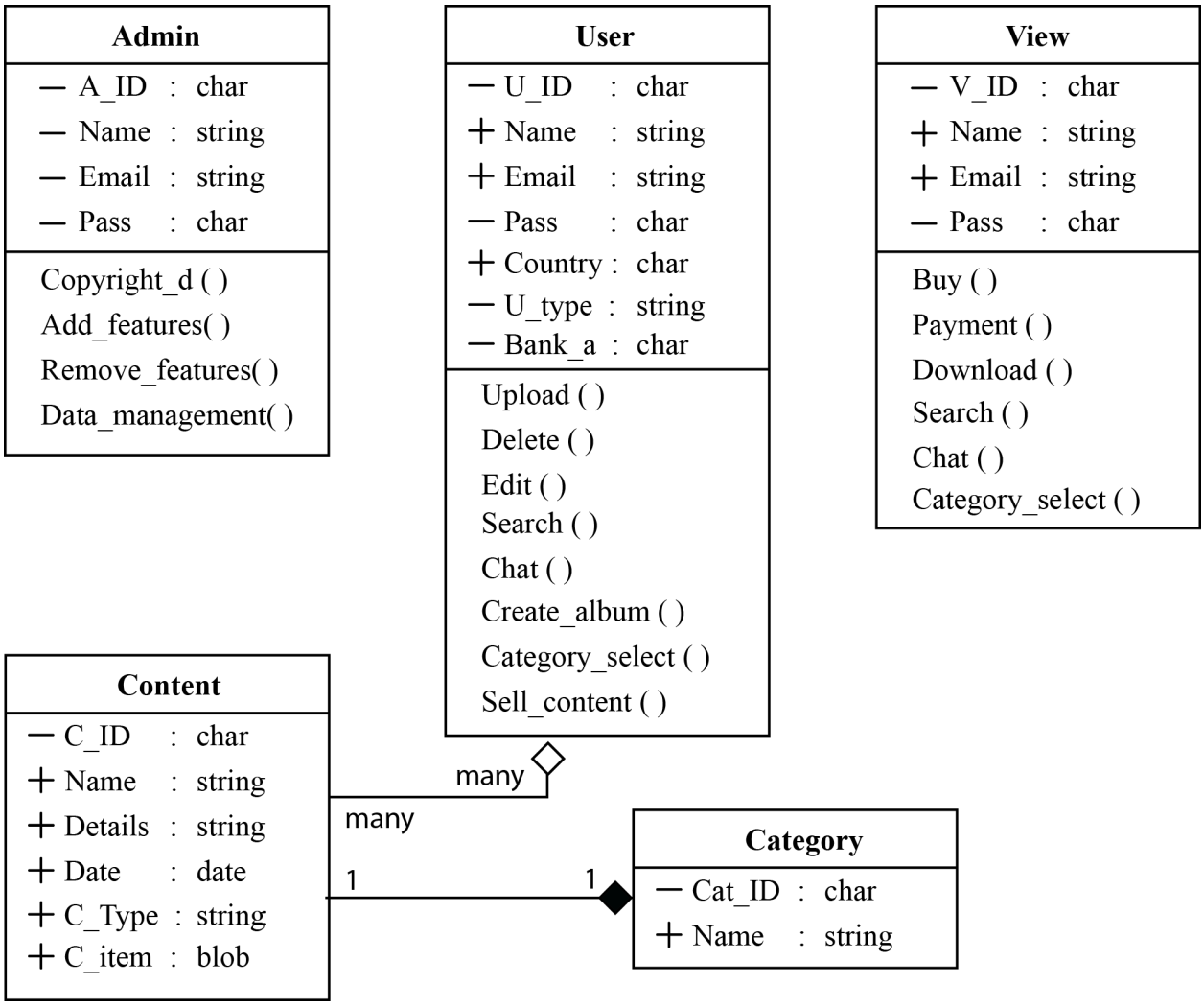
Member access modifiers

All classes have different access levels depending on the access modifier (visibility). Here are the access levels with their corresponding symbols:

- Public (+)
- Private (-)
- Protected (#)
- Package (~)

Class Diagram Relationship Type	Notation
Association	
Inheritance	
Realization/ Implementation	
Dependency	
Aggregation	
Composition	

Class diagram for the system of our system:



Conclusion:

We can design the system better by using the class diagram designed for our system

Chapter 5: Sequence Diagram

Objective: Designing a sequence diagram for an efficient software development process.

Sequence Diagram : Sequence diagrams are commonly used by developers. It creates a model of the interactions between objects in a use case. They illustrate how the different parts of a system interact with each other to carry out a function. They also show the order in which the interactions occur.
In simpler terms, a sequence diagram shows how different parts of a system work in a ‘sequence’ to get something done.

Benefits of using a sequence diagram :

These are some of the main benefits of using UML sequence diagrams:

1. Help in discovering architectural, interface and logic problems early.

2. It is easy to generate.

3.If the sequence diagrams are used as collaboration tools, communication is made easy during design discussions - as one can see interactions between entities

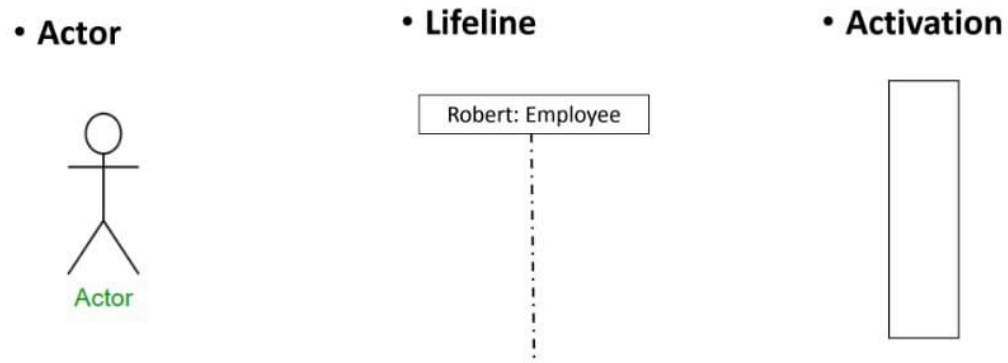
4.Enhances Productivity as it helps in identifying logic problems early, and see how it handles the special cases.

5. The UML sequence diagram can document the system design at various levels. We can extract details at any level,can also monitor the system’s behavior and make any possible changes.

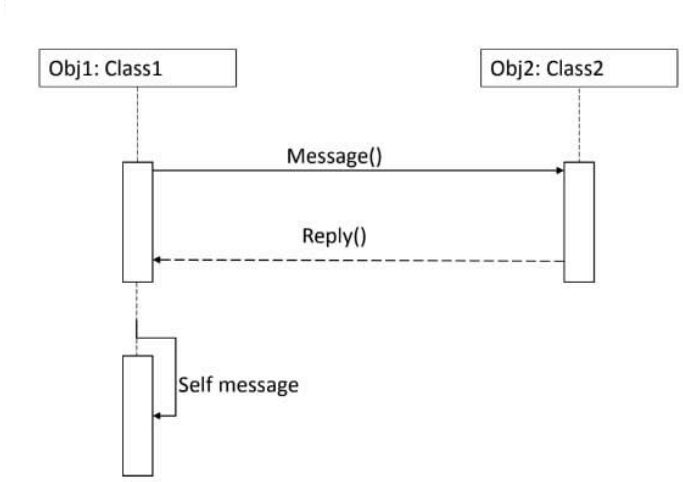
6.Provides a better understanding of the system.

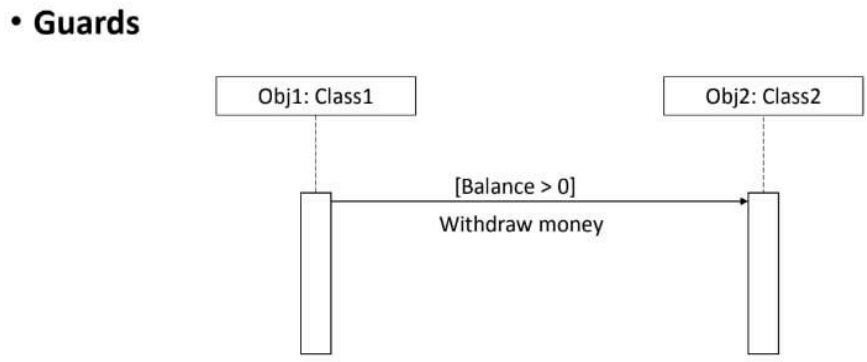
7. Saves time.

Sequence Diagram Notation :

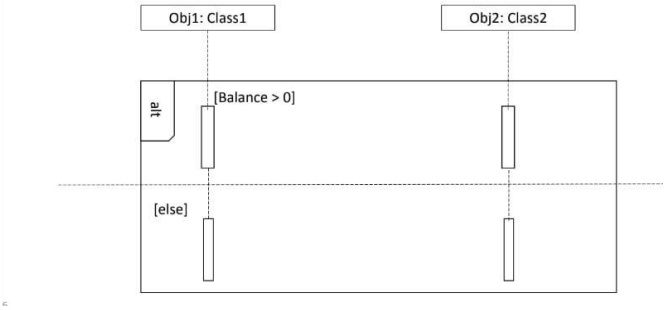


Message:

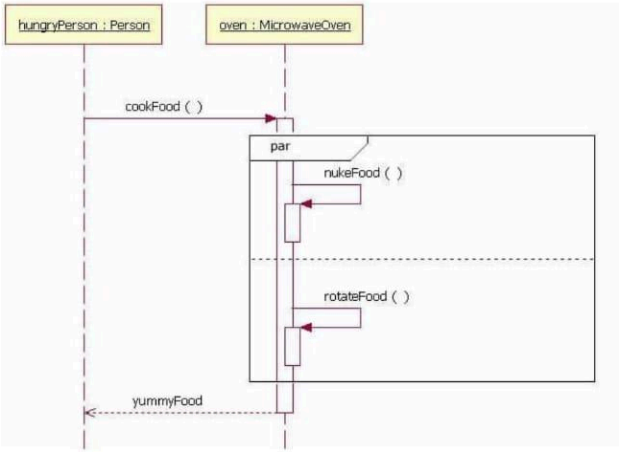




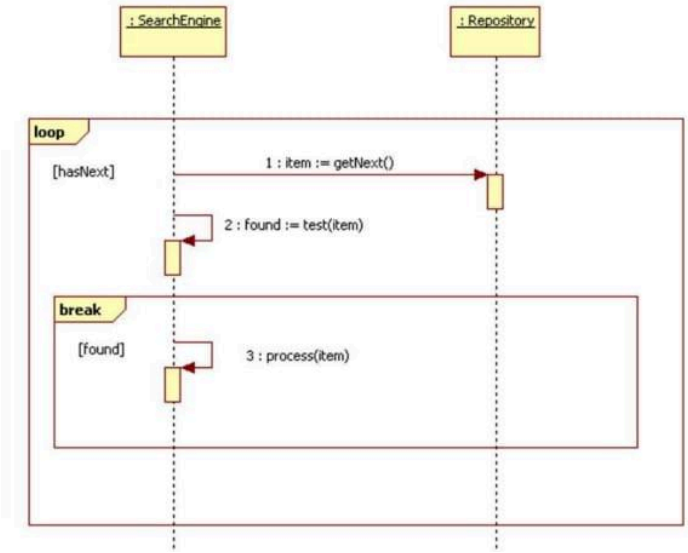
• **Alternatives**



Parallel activities:



Loop:



The Sequence Diagram of "Through The Lens":

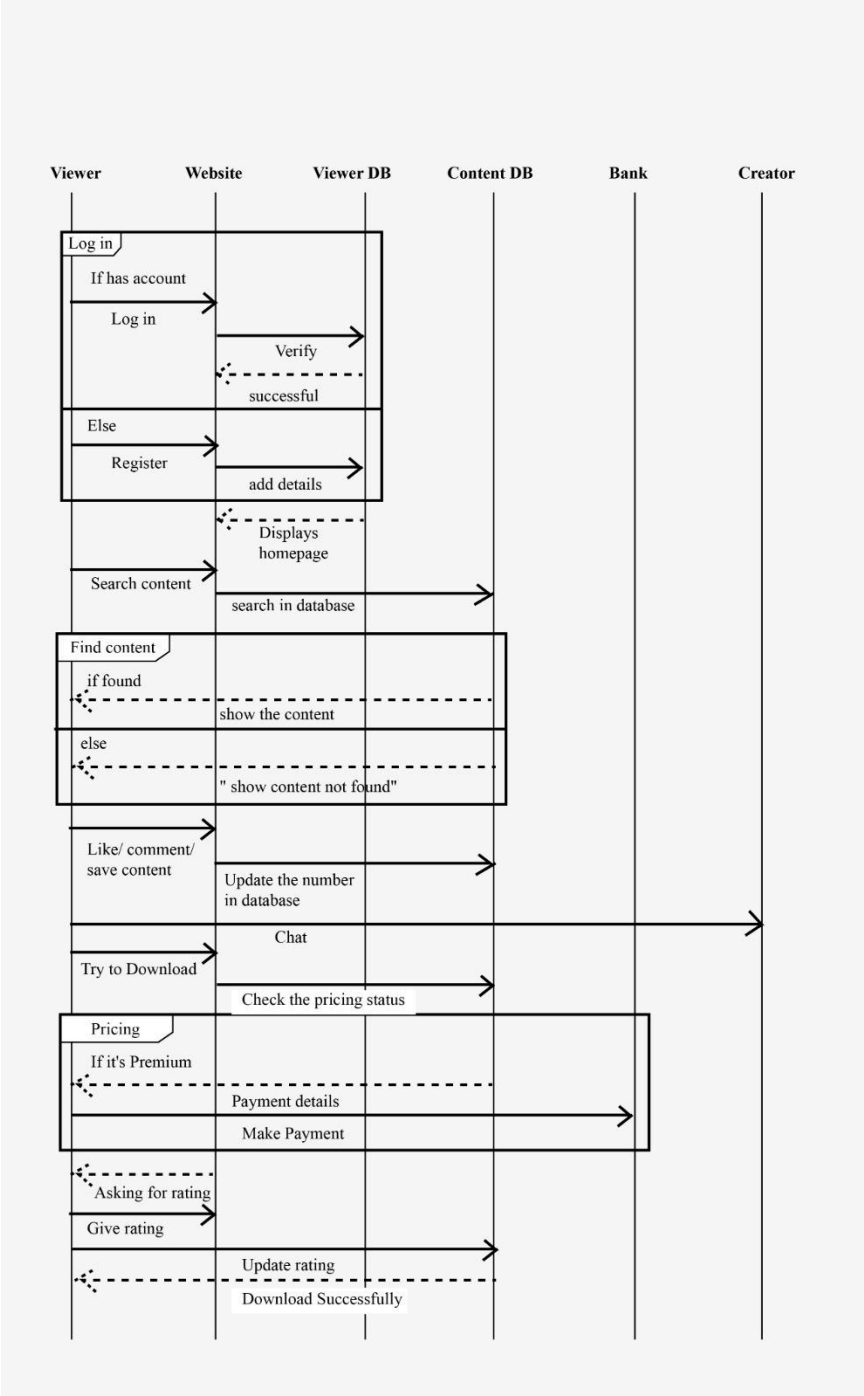


Fig: viewers actions' sequence diagram

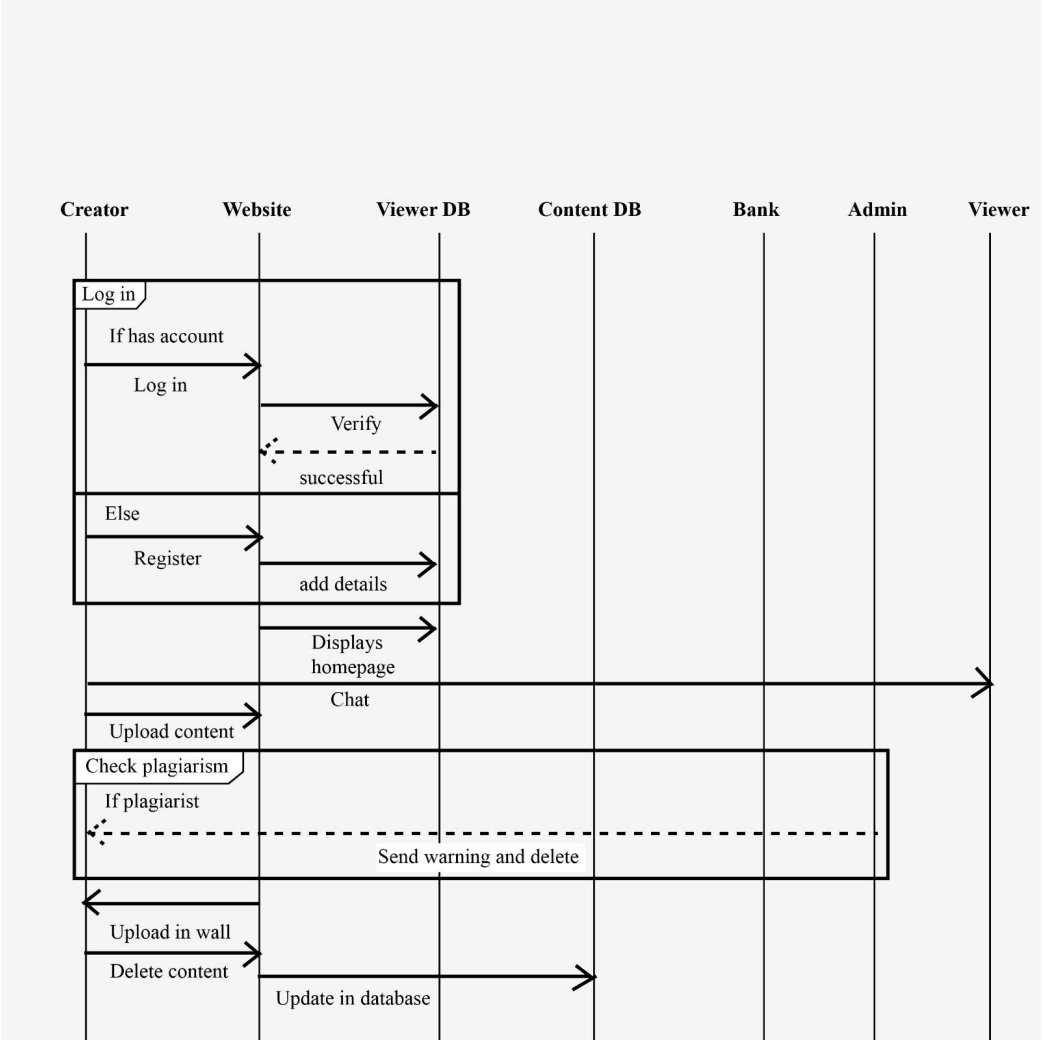
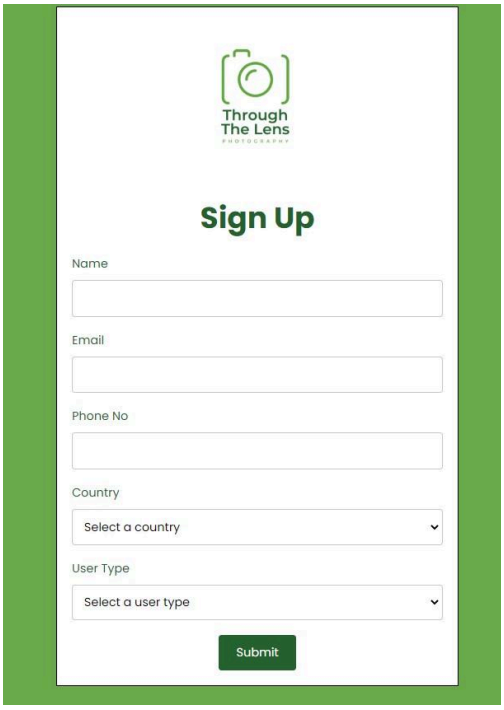


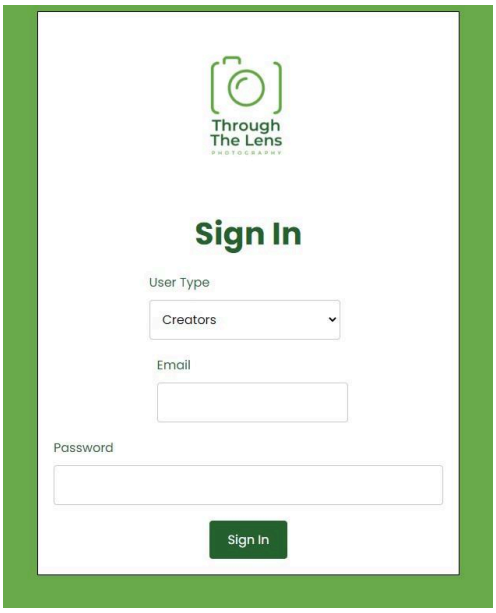
Fig: Creators actions' sequence diagram

Conclusion: Using the Sequence Diagram designed for our system will hopefully make our system design and the documentation process much easier. We can get all the benefits, understand the systems working process better and check for logical errors faster.

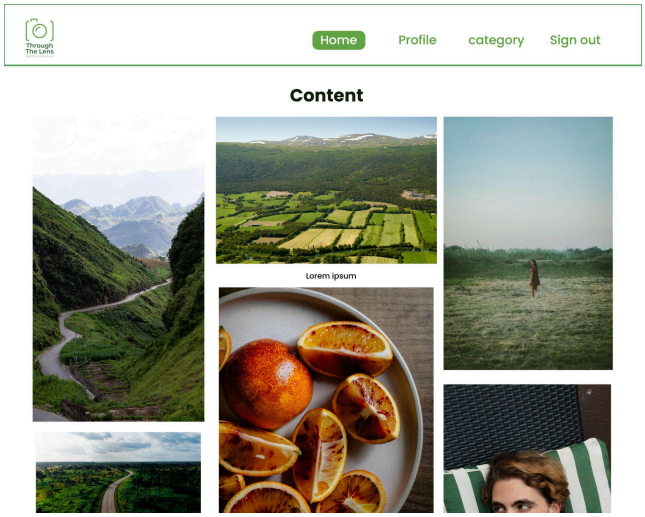
Chapter 6 : Output



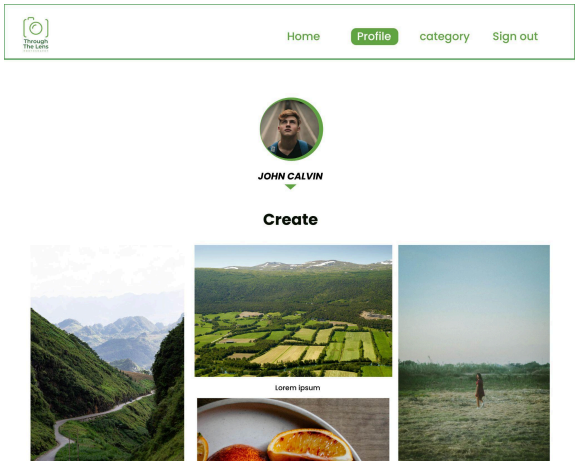
Img 1 : Sign Up page



Img 2 : Sign in page

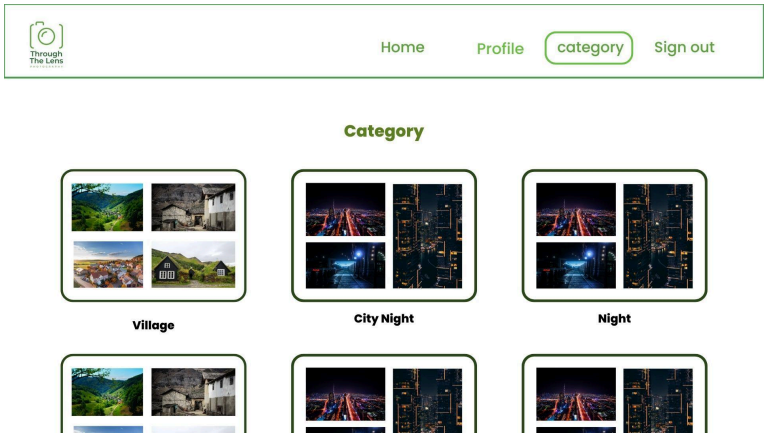


Img 3: Home Page



Img 4:Profile

Img 4:Profile edit



Img 4:Category

Chapter 07 : Software Requirements Specification (SRS Document)

1. Introduction :

1.1 The scope of the system :

The system will be a web-based platform for creators to share and users to view photographs and videos. The platform will have features such as user registration and login, content display, downloading, and premium content.

Creators will be able to upload photographs and videos to the platform, and users will be able to search and view them. Users can download the content, but before doing so, they must pay the price set by the creator to access the content. The platform will also have a premium content feature that allows creators to make their content premium.Users can also like the contents, and creators will be able to see the number of likes their contents have received.

1.2 The project overview:

Our Team has a total of two members. So the tasks have been delegated among us:

Phase 1

This phase involves identifying the requirements of the system. The following table represents the requirements upon which each member contributed.

Name	Requirements number
Jannatul Ferdous	R1,R5
Afra Ibnat	R2,R3,R4,R6

Phase 2

This phase involves developing test cases for the system. The following table represents the test cases which each member worked on.

Name	TEST CASE ID
Jannatul Ferdous	TC 001 - TC010
Afra Ibnat	TC 011 - TC018

Phase 4:

This phase involves documenting bugs that were identified during testing. The following table represents the bugs which were reported by each member

Name	BUG ID
Jannatul Ferdous	BUG 001 - BUG004
Afra Ibnat	BUG 005 - BUG006

2. System Testing :

2.1 System Requirements

Requirements for the system are listed in the following table.

Requirements	Requirement No.	Description
R01: User Registration and Login	R01.01	Users can create an account by providing a unique username, email address, and password.
	R01.02:	The system verifies that the username and email address are not already in use.
	R01.03	The system requires users to verify their email address before they can log in
	R01.04:	Users can log in using their username/email and password.
	R01.05:	The system displays an error message if the login credentials are invalid.
	R02.01	: Creators can upload photographs and videos.
	R02.02	The system verifies that the file types are supported and that

R02: Content Upload		the file size does not exceed the maximum limit.
	R02.03	Creators can add captions to each item they upload.
	R02.04	The system generates a unique URL for each uploaded content.
R03 : Content Display	R03.01	The system displays the uploaded photographs and videos in a grid format.
	R03.02	Users can click on an item to view it in full-screen mode.
	R03.03	Users can like, comment, and save the contents.
	R03.04	The system tracks and displays the number of likes, comments, and saves for each content.
R04: Downloading and Premium Content	R04.01	Users can download the contents.
	R04.02	Before downloading, users must rate the content using a 5-star rating system.
	R04.03	The system calculates the average rating for each creator and displays it on their profile.

	R04.04	Creators can choose to make their content premium, which requires users to pay the price set by the creator to access the content.
R05: Chatting	R05.01	Viewers and creators can chat with each other using a chat system integrated into the platform.
	R05.02	The system displays the chat history and allows users to send and receive messages in real-time.
	R05.03	The system allows users to block other users or report inappropriate messages.
R06: User Profile	R06.01	The system allows users to view their own profile and edit their information.
	R06.02	The system displays a list of uploaded contents and their statistics.
	R06.03	The system displays the creator's average rating and provides a link to their premium contents.

2.2 System Test Cases

Test case ID:	TC001
Test case title:	Registration page loads correctly
Test case type:	Functional test
Test Steps:	1. Navigate to the registration page. 2. Verify that the registration page loads correctly and all required fields are present and labeled correctly.

Expected Result:	The registration page should load correctly and all required fields should be present and labeled correctly.
Actual Result:	The registration page loads correctly and all required fields are present and labeled correctly.
Pass/Fail:	Pass

Test case ID:	TC002
Test case title:	Verify successful login with valid credentials
Test case type:	Functional test case
Test Steps:	1. Navigate to the login page. 2. Enter valid login credentials (username and password). 3. Click on the login button. 4. Verify that the system redirects the user to their account page.
Expected Result:	The user should be able to log in successfully with valid credentials and be redirected to their account page.
Actual Result:	The test case does not specify which type of user account is being used to test the login functionality. For example, it could be an admin account, a user account, or a creator account with specific privileges. This lack of clarity could lead to confusion and inconsistent test results.
Pass/Fail:	Fail
Revised Test Case	Verify successful login with valid credentials for a customer account. The user should be able to log

	in successfully with valid credentials and be redirected to their dashboard page.Verify successful login with valid credentials for a user account. The user should be able to log in successfully with valid credentials and be redirected to their dashboard page.

Test case ID:	TC003
Test case title:	Verify successful logout
Test case type:	Functional test case
Test Steps:	1. Log in to the system using valid credentials. 2. Click on the "Logout" button. 3. Verify that the system successfully logs out the user and redirects them to the login page.
Expected Result:	The system successfully logs out the user and redirects them to the login page after they click on the "Logout" button.
Actual Result:	The system successfully logs out the user and redirects them to the login page after they click on the "Logout" button.
Pass/Fail:	Pass

Test case ID:	TC004
Test case title:	Verify successful upload of photographs and videos
Test case type:	Functional test case
Test Steps:	1. Log in to the system as a creator.

	2. Click on the "Upload" button. 3. Select a photograph or video to upload. 4. Enter a caption for the photograph or video. 5. Click on the "Upload" button to complete the process. 6. Verify that the photograph or video is successfully uploaded and appears on the platform.
Expected Result:	The photograph or video should be successfully uploaded to the platform and should appear in the creator's profile.
Actual Result:	The photograph or video is successfully uploaded to the platform and appears in the creator's profile.
Pass/Fail:	Pass

Test case ID:	TC005
Test case title:	Verify successful creation of album and addition of captions
Test case type:	Functional test case
Test Steps:	1. Log in to the system as a creator. 2. Click on the "Create Album" button. 3. Enter a name for the album and click on the "Create" button. 4. Click on the newly created album to view its contents. 5. Click on the "Add Item" button. 6. Select a photograph or video to upload. 7. Enter a caption for the photograph or video. 8. Click on the "Add" button to complete the process. 9. Verify that the photograph or video is successfully added to the album with the correct caption.
Expected Result:	The album should be successfully created with the correct name and should contain the added

	photograph or video with the correct caption.
Actual Result:	The album is successfully created with the correct name and contains the added photograph or video with the correct caption.
Pass/Fail:	

Test case ID:	TC006
Test case title:	Verify successful search by keyword and tag
Test case type:	Functional test case
Test Steps:	1. Log in to the system as a user. 2. Click on the search bar. 3. Enter a keyword or tag related to the photograph or video being searched for. 4. Click on the "Search" button. 5. Verify that the search results include photographs or videos with the entered keyword or tag. 6. Click on a search result to view the photograph or video. 7. Verify that the photograph or video is displayed correctly
Expected Result:	The search results should include photographs or videos with the entered keyword or tag. The photograph or video should be displayed correctly after being clicked on.
Actual Result:	The test case does not define the expected search results for a successful search by keyword and tag.
Pass/Fail:	Fail
Revised test case	Verify successful search by keyword and tag, specifying the expected search results to ensure consistent and accurate testing.

Test case ID:	TC007
Test case title:	Verify successful view of photographs and videos in a grid format
Test case type:	Functional
Test Steps:	1. Log in to the system as a user. 2. Click on the "Photographs" or "Videos" tab. 3. Verify that photographs or videos are displayed in a grid format. 4. Click on a photograph or video to view it in full-screen mode. 5. Verify that the photograph or video is displayed correctly in full-screen mode.
Expected Result:	The photographs or videos should be displayed in a grid format. The photograph or video should be displayed correctly in full-screen mode.
Actual Result:	The photographs or videos are displayed in a grid format. The photograph or video is displayed correctly in full-screen mode.
Pass/Fail:	Pass

Test case ID:	TC008
Test case title:	Verify successful view of photographs and videos in full-screen mode
Test case type:	Functional
Test Steps:	1. Log in to the system as a user. 2. Click on the "Photographs" or "Videos" tab. 3. Click on a photograph or video to view it in full-screen mode. 4. Verify that the photograph or video is displayed correctly in full-screen mode. 5. Check that the user can exit full-screen mode and return to the grid view.

Expected Result:	The photograph or video should be displayed correctly in full-screen mode. The user should be able to exit full-screen mode and return to the grid view.
Actual Result:	The photograph or video is displayed correctly in full-screen mode. The user can exit full-screen mode and return to the grid view
Pass/Fail:	Pass

Test case ID:	TC009
Test case title:	Verify that users can like and comment on photographs and videos.
Test case type:	Functional
Test Steps:	1. Navigate to the platform's home page. 2. Search and select a photograph or video to view. 3. Verify that the photograph or video is displayed correctly and the user can view it. 4. Click the "Like" button below the photograph or video. 5. Verify that the "Like" button changes colour or displays a count indicating that the user has successfully liked the content. 6. Add a comment in the comment section below the photograph or video. 7. Verify that the comment is successfully posted and displayed below the content. 8. The photograph or video is displayed correctly, and the user can view it. 9. The "Like" button changes color or displays a count indicating that the user has successfully liked the content. 10. The comment is successfully posted and displayed below the content.
Expected Result:	The user should be able to successfully like and comment on photographs and videos, and the likes and comments should persist after refreshing the page.

Actual Result:	The test case does not verify whether likes and comments made by users are accurately recorded and displayed on photographs and videos.
Pass/Fail:	Fail
Revised test case	Verify that users can like and comment on photographs and videos, and verify that the likes and comments made by users are accurately recorded and displayed.

Test case ID:	TC010
Test case title:	Verify saving photographs and videos to profile
Test case type:	Functional
Test Steps:	1. Log in as a user. 2. Click on a photograph or video. 3. Save the photograph or video. 4. Navigate to the user's profile. 5. Verify that the saved photograph or video appears in the user's saved items or bookmarks list.
Expected Result:	The user should be able to successfully save photographs and videos to their profile and view them in their saved items or bookmarks list.
Actual Result:	Test case does not specify expected behavior when a user saves a photograph or video to their profile.
Pass/Fail:	Fail
Revised test case	Verify saving photographs and videos to profile and specify expected behavior.

Test case ID:	TC011
Test case title:	Verify that creators can set a price for their premium content
Test case type:	Functional
Test Steps:	1. Log in to the system as a creator. 2. Upload content and mark it as premium. 3. Set a price for the premium content. 4. Save the changes. 5. Log out of the system. 6. Log in to the system as a regular user. 7. Search for the premium content. 8. Attempt to download the premium content. 9. Verify that the user is prompted to pay the price set by the creator.
Expected Result:	The creator should be able to set a price for their premium content, and the user should be prompted to pay the price to download the premium content
Actual Result:	The creator was able to set a price for their premium content, and the user was prompted to pay the price to download the premium content.
Pass/Fail:	Pass

Test case ID:	TC012
Test case title:	Verify that creators can view analytics for their content
Test case type:	Functional
Test Steps:	2. Log in to the system as a creator. 3. Upload content. 4. View the analytics for the uploaded content. 5. Verify that the number of views and downloads is displayed. 6. Verify that the total earnings from selling

	premium content is displayed.
Expected Result:	The creator should be able to view the number of views and downloads for their content, as well as their total earnings from selling premium content.
Actual Result:	The creator was able to view the number of views and downloads for their content, as well as their total earnings from selling premium content.
Pass/Fail:	Pass

Test case ID:	TC013
Test case title:	Verify that users can rate content before downloading it
Test case type:	Functional
Test Steps:	3. Log in to the system as a regular user. 4. Search for content to download. 5. Attempt to download the content. 6. Verify that the user is prompted to rate the content on a 5-star scale before downloading it. 7. Rate the content. 8. Download the content.
Expected Result:	The user should be prompted to rate the content on a 5-star scale before downloading it, and should be able to download the content after rating it.
Actual Result:	Test case does not specify expected behavior after a user rates content before downloading it.
Pass/Fail:	Fail
Revised test case:	Verify user rating of content before downloading it and define expected behavior, such as

	displaying the user's rating or calculating an overall rating.

Test case ID:	TC014
Test case title:	Verify that users can rate content on a 5-star scale before downloading it.
Test case type:	Functional
Test Steps:	1. Log in to the system as a user. 2. Search for a piece of content. 3. Click on the content to view the details page. 4. Rate the content using the 5-star scale. 5. Try to download the content. 6. Verify that the user can download the content after rating it.
Expected Result:	The user should be able to rate the content and download it after rating it.
Actual Result:	The user can rate the content and download it after rating it.
Pass/Fail:	Pass

Test case ID:	TC015
Test case title:	Verify that creators can set a price for their premium content.
Test case type:	Functional
Test Steps:	2. Log in to the system as a creator. 3. Upload a piece of content. 4. Set a price for the content. 5. Publish the content. 6. Log out of the system. 7. Log in to the system as a user. 8. Search for the premium content. 9. Try to download the premium content. 10. Verify that the user is prompted to pay

	the price set by the creator.
Expected Result:	The creator should be able to set a price for their premium content and the user should be prompted to pay the price to access the content.
Actual Result:	The creator can set a price for their premium content and the user is prompted to pay the price to access the content.
Pass/Fail:	Pass

Test case ID:	TC016
Test case title:	Verify that creators can view the number of views and downloads for their content.
Test case type:	Functional
Test Steps:	1. Log in to the system as a creator. 2. View the dashboard or analytics section. 3. Verify that the number of views and downloads are displayed for each piece of content.
Expected Result:	The creator should be able to view the number of views and downloads for their content.
Actual Result:	The creator can view the number of views and downloads for their content
Pass/Fail:	Pass

Test case ID:	TC017
Test case title:	Verify that viewers and creators can chat with each other on the platform
Test case type:	Functional
Test Steps:	1. Log in to the system as a viewer or

	creator. 2. Search for a piece of content. 3. Click on the content to view the details page. 4. Start a chat with the creator or viewer. 5. Send a message. 6. Verify that the message is sent and received.
Expected Result:	Viewers and creators should be able to chat with each other on the platform and send and receive messages.
Actual Result:	Viewers and creators can chat with each other on the platform and send and receive messages.
Pass/Fail:	Pass

Test case ID:	TC018
Test case title:	Verify that creators can view the number of views and downloads for their content.
Test case type:	Functional
Test Steps:	1. Log in to the system as a creator. 2. View the dashboard or analytics section. 3. Verify that the number of views and downloads are displayed for each piece of content.
Expected Result:	The creator should be able to view the number of views and downloads for their content.
Actual Result:	The creator can view the number of views and downloads for their content
Pass/Fail:	Pass

2.3 Report on Execution of System Test Cases

Test Case ID	Pass/ Fail	Comments	Bug ID	Execution Date	Test Executor
TC001	**Pass**			27.02.2023	Jannatul Ferdous
TC002	**Fail**	The test case does not specify which type of user account is being used to test the login functionality.	**BUG001**	27.02.2023	Jannatul Ferdous
TC003	**Pass**			27.02.2023	Jannatul Ferdous
TC004	**Pass**			28.02.2023	Jannatul Ferdous
TC005	**Pass**			28.02.2023	Jannatul Ferdous
TC006	**Fail**	Verify successful search by keyword and tag, specifying the expected search results to ensure consistent and accurate testing.	**BUG002**	02.03.2023	Jannatul Ferdous
TC007	**Pass**			02.03.2023	Jannatul Ferdous
TC008	**Pass**			02.03.2023	Jannatul Ferdous
TC009	**Fail**	The test case does not verify whether likes and comments made by users are accurately recorded and displayed on photographs and videos.	**BUG003**	04.03.2023	Jannatul Ferdous
TC010	**Pass**			05.03.2023	Jannatul Ferdous
TC011	**Pass**			05.03.2023	Afra Ibnat

TC012	**Pass**			05.03.2023	Afra Ibnat
TC013	**Fail**	Test case does not specify expected behavior after a user rates content before downloading it.	**BUG005**	05.03.2023	Afra Ibnat
TC014	**Pass**			06.03.2023	Afra Ibnat
TC015	**Pass**			06.03.2023	Afra Ibnat
TC016	**Pass**			06.03.2023	Afra Ibnat
TC017	**Fail**	Test case does not specify expected behavior of chat feature between viewers and creators.	**BUG006**	06.03.2023	Afra Ibnat

3. Conclusion

The system developed successfully meets the requirements specified in the SRS document for User Registration and Login, Content Upload, Content Display, Downloading and Premium Content, Chatting, and User Profile. Out of the 18 test cases run by us, the system passed 18 test cases, and the 6 bugs found were revised. The system is expected to provide an excellent user experience, enabling creators to share their content and users to engage with it.

Testing Team Task Delegation Metrics: Jannatul Ferdous [50%], Afra Ibnat [50%].