School of Electronic Engineering and Computer Science	Interim/Final Report
	Programme of study: Computer Science BSc
	Project Title: **ArtNote**
	Supervisor: George Fazekas
	Student Name: Duy Huu Nguyen
Final Year Undergraduate Project 2020/21	Date: 20/11/2020

Queen Mary
University of London

Abstract

Art demand the artist to be creative and requires a lot of focus and dedication to an artwork, thus rumination occurs and potentially causes depression but with the use of social media, this problem can be resolved. However, social media app , nowadays tends to focus more on sharing or interacting with user's posts and are limited to these actions. This is not a suitable way for artists to interact as discussed in the report and the goal of this project is to develop an application to solve this problem. Sequentially, the report will be doing essential research for the development process, gather the requirements for the project. Furthermore, the report will also be discussing about the design and high level implementation of the project. Lastly, it is important to list the potential risk that may occur during development process.

Table of Contents

Chapter 1:Introduction

In the first chapter, I am going discuss the about the origin of the project, indicating the purpose of the project, what problem is required to be solved

1.1 Problem

Creativity is a crucial part of an artist's life as it is the engine that keeps them motivated and focused during long painting sessions. However, even though creativity might be seen as a powerful tool that lead to success, it also might become a wall that kills the artist. Based on an article by Tanner(2013),in order to be creative, rumination is required, and this is agreed by many psychologists and psychiatrists to be the source of depression in artists. In addition, Tanner also states that rumination a process that demands logical/connective thinking in order to sculpt/combine small details and "Creatives naturally tend to think more and think about their every thought too". Due to this, it is stated that those who ruminate tends to be impacted by pain and suffering more often, because artists is a group that gravitate towards stressful events from time to time. Fortunately, there are methods for alleviate depression, one of it is socializing. According to Nuacht UCD(2012),the more socializing ones become the more likely the person's self-esteem is rebuilt, which possibly implies the growth in having healthy and positive relationships.

Artists tend to have deep thoughts while working on artworks. Nonetheless, this could lead self-isolation due to lack of socializing because of long hours of work. In addition, this is an act of rumination, which is agreed by psychologists and psychiatrists to be the cause of depression because not only do artists have to loop through an experience again and again due to deep thinking, they also might not be able to share to those experiences to anyone because of self-isolation.

1.2 Objectives

This project aims to aid artist in easing the looping cycle of thoughts in artists due to rumination and creating a platform where artists can socialize with each other while still be able to work simultaneously. Because of this, the goal of the project to is to create a platform where artists can interact with each other through socializing and painting, the project can be a great source of inspiration and entertainment for artist as this can act as a meeting for artist to come to paint and have conversation with each other.

Below are the main objectives of the project:

- ✧ Design an application for sharing artworks
- ✧ Design a gallery system for displaying artworks that can be viewed by users(artists)
- ✧ Design a chatting platform for artists to come for chatting and painting
- ✧ The design allows artists to interact with the canvas in real time
- ✧ Design canvas and brush systems with UI and functionality that artists are familiar with
- ✧ Research on similar products, evaluate their pros and cons
- ✧ Do literature review on work relate to Depression, socialization for artist
- ✧ Derive from the research the methods, algorithms and techniques need to be applied to and what can be improved on the project based on other similar projects
- ✧ Decide on what tool/languages/framework to use for implementation
- ✧ Decide which DBMS to use for implementation

1.3 Scope

- ■ **DELIVERABLE**

The project is a web application that allows user to create/join rooms with other users to communicate through voice/text and paint on the same canvas. In addition, each users will have their own gallery which allows them to share artworks and interact with other users on the platform, while having their mood monitored by the system which will then provide suitable artworks from other users that help improve mental health .

- ■ **TIMELINE**

Phase	Task	Start date	Days to Com	Duration
Data gathering and documentation	Requirement and analysis	04-Oct	12-Oct	8
	Diagram(class,Use case)	12-Oct	13-Oct	1
	ERM ,Relational Models,Django ORM	12-Oct	15-Oct	3
Iteration 1	Core functionality for drawing room	15-Oct	17-Oct	2
	Core functionality for displaying gallery	17-Oct	20-Oct	3
	Core functionality for displaying discovery	20-Oct	21-Oct	1
	Creating models for Django	20-Oct	28-Oct	8
	Implement Channels for backend	28-Oct	06-Nov	9
	Apply basic bootstrap to front end	06-Nov	08-Nov	2
Iteration 2	Intergrate Web socket for front end	08-Nov	09-Nov	1
	Establish Web socket connection	09-Nov	11-Nov	2
	Post/get images/user info between client and server	11-Nov	18-Nov	7
	Allow user to create rooms to draw	18-Nov	22-Nov	4
	Save room info	22-Nov	23-Nov	1
Iteration 3	Allow user to join a room without requesting	22-Nov	24-Nov	2
	Allow users to draw on the same canvas	24-Nov	28-Nov	4
	Create chat box for a room	28-Nov	01-Dec	3
	Display rooms joined for each user	01-Dec	06-Dec	5
	Auto Saving canvas	06-Dec	07-Dec	1
	Style frontend	06-Dec	07-Dec	1
	Monitor user's mood for front end	06-Dec	07-Dec	1
Iteration 4	Require user to make a request before joining	06-Dec	10-Dec	4
	Display requests to a room to its host once entered	10-Dec	12-Dec	2
	Allow user to accept/decline request	12-Dec	19-Dec	7
	Display public room activities on discovery but does not allow streaming	19-Dec	20-Dec	1
	Display post according to user's mood	20-Dec	21-Dec	1
Iteration 5	Add livestream ability to each public room displayed	20-Dec	27-Dec	7
	Add voice chat to room	27-Dec	31-Dec	4
	Refine front end	31-Dec	04-Jan	4

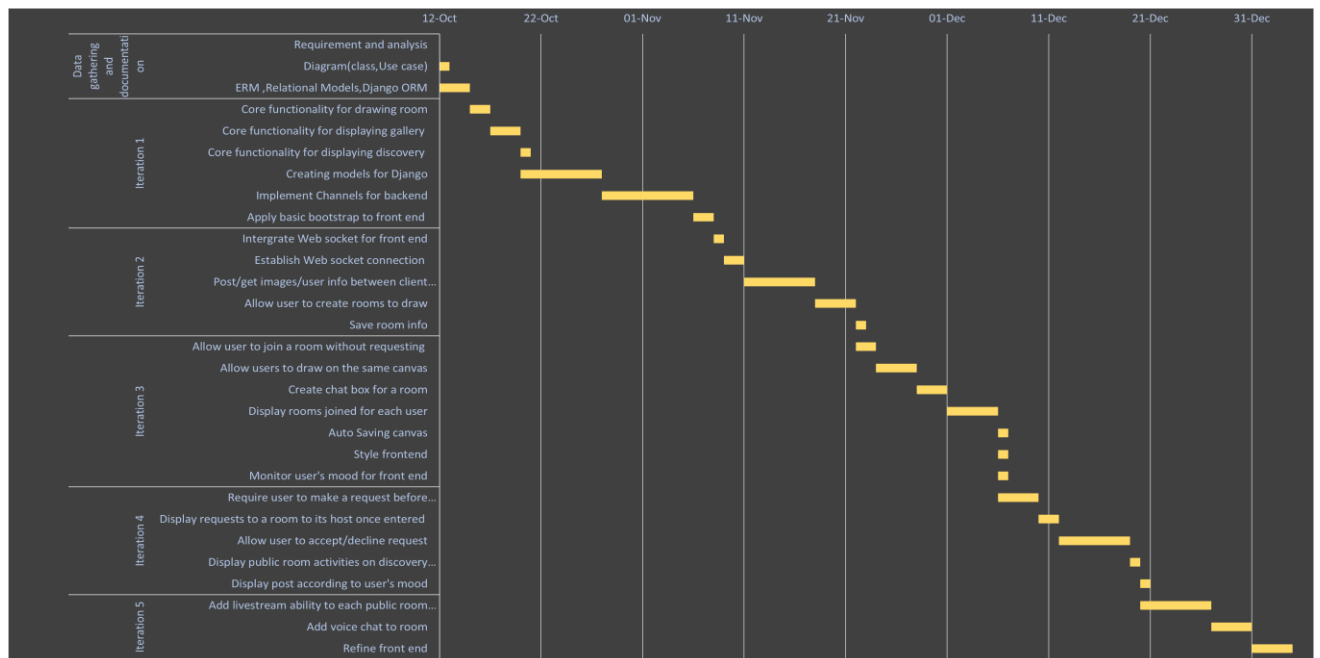**FIGURE 1.1.TIMELINE FOR THE PROJECT PLAN**

Gantt chart:

FIGURE 1.2.GANT CHART OF FIGURE 1.1 TIMELINE

CHAPTER 2:RESEARCH
2.1 Depression

2.1.1 What is depression?

Based on an article from NHS(nimh.nih.gov,https://www.nimh.nih.gov/health/topics/depression/index.shtml), depression is a type of symptom disorder that has impact on ones action, feeling and appetite. Several form of depression are:

◆ Persistent depressive disorder: the person with this form will have various experiences of major depression together with less serious symptoms.

◆ Psychotic depression: is a form of depression ,in which the person experiencing the symptoms is also affected by psychosis such as delusion, find it is unpleasant in things other people don't(hallucination)…

The article stated that a person with depression may suffer from some of these symptom:

- Anxious or empty mood
- Hopelessness
- Loose interest in activities
- Struggle to focus and concentration
- Suicide thoughts,….

From these signs we could conclude that depression not only pessimistically alters a person's mental health but also might have possibly to give a person suicide thoughts.

2.1.2 Artist and depression
Ever since the old days of art, artists and mental health symptom, specifically depression, have always been believed to have a connection with each other. According to Dodgson(2018,https://www.insider.com/the-link-between-creativity-and-mental-health-2018-7),there has always existed a connection between popular artists in the past and hallucination/delusion, a typical example is the famous Vincent Van Goth, who servers his own ears due to a fight with his friends Paul Gauguin and committed suicide 2 years later. It was described by the artist that: "I am unable to describe exactly what is the matter with me"," Now and then there are horrible fits of anxiety, apparently without cause, or otherwise a feeling of emptiness and fatigue in the head … At times I have attacks of melancholy". With this, it could be derived that without self-awareness or being too dedicated, a person will be affected or even absorbed by anxiety and hollow, which could potentially leads to death. In addition, it was also

mentioned in the article, people who work an artistic job are more likely to die by suicide as reported from a study the Office of National Statistic. This once again , strengthen the connection between the artists and the deadly symptoms, depression.

2.1.3 Treatment

Fortunately, there many ways to ease the effects of depression, one of those is socialization. Based on an article by Bulfut(2019,book downloaded,name:SocializationHelpsTheTreatmentOfDepressionInModernLife),psychologist find that an individual's health and mental is improved tremendously when they are surrounded by people who can understand them, share similar challenges, these elements create a feeling of warmth and mutual understanding ,thus making the individual feel cheerful and comfortable. It is also mentioned , unorganised and informal meeting/conversation, face-to-face interaction   could positively leave an impact on a person's lifestyle and mental state.

Another type of treatment is art therapy, in which a person will use activities such as sketching, journaling, painting,… in order to artistically express themselves ,thus relieve stress, written in article from rtor.org(2018,https://www.rtor.org/2018/07/10/benefits-of-art-therapy/).It also mentioned that activities done together as a group assist the connection between people and people, which is likely to minimize the effect of depression. In addition, it is also stated that during the process of art therapy, one does not only have a chance for self-exploration but also for connecting with others, hence overcome emotional difficulties.

2.2 Digital art

2.2.1 What is digital art?

Traditional art is form a of painting using traditional medium such as brushes, canvases, watercolor, gauche,....On the other hand, digital painting uses technology/devices to paint on machine such as a computer or tablet. According to an article on Concept Art Empire(https://conceptartempire.com/digital-painting/),the usage of technology, for example: painting software, tablet,... has become more popular as artists prefer the long-term benefit in terms of investment and this form of art is also required in a variety of industry including animation, illustration, Game,....Overall, digital art might has become the standard for working in industry, which possibly means that people has become more familiar with this format and vice versa, this format has become more accessible to more people.

2.2.2 Tools

In order to get to start with digital art, a number of tools are required.

According to the Adobe's beginner's guide to digital drawing tools(https://www.adobe.com/uk/creativecloud/illustration/discover/digital-pens-digital-tools.html), the below tools are required:

1. Digital stylus/Smart pen

To start off, a type of digital pen is required, and it is mentioned that there are 2 types of pen: Smart pens and digital stylus. Firstly, the former trails the lines and points the user make when drawing on a special paper, take their data and transfer it via the built-in blue tooth of the pen to a computer in real time. Because of this, user's drawing will appear on the screen simultaneously as they drag the pen across the paper. However, the later gives a more natural and traditional feel to the process as it allows user to draw directly on a surface of a device such as a graphic tablet.

2. Device

Base on the type of stylus/pen recommend above by adobe, we could conclude that, some of the devices required might be a computer or a tablet with stylus functionality. In addition , it might be possible to that the choice of which type of those devices is depend on ones work-flow. The reason for this is while a computer might give the user powerful tools, a broad view of their current project due to seeing only the artwork on the computer's screen while working, it is also a stationary device which means portability is a problem due to the size of the device. In contrast to this, a small tablet provides portability ,user could utilize this to comfortably do their job where they want to, for example, when it would be impossible to sketch outside with a tool such as a computer, however, it would likely be capable with a tablet, while still providing the essential features a computer has. In conclusion, each device has its own benefits and drawbacks, which means each of them could bring its own unique profit to different businesses.

3. Drawing software

Lastly, a type of software is compulsory, this aids the artist in various way one of them is correcting mistakes during painting. Since the medium is a software ,it is feasible to erase/undo any errors, making the work flow more enjoyable.

2.3 Django.
2.3.1 What is Django?
According to the Django homepage(https://www.djangoproject.com/start/overview/),it is a python framework that allows rapid, fast development while providing re-usability saving time and effort from implementing existing code. In addition, it also aids the user with modules that help with authentication, administration and many other tasks. Lastly, Django protects developers by providing strong security preventing SQL injection, cross-site scripting, cross-site request forgery, clickjacking.

2.3.2 Usage

Based on Kalpit(https://medium.com/crowdbotics/when-to-use-django-and-when-not-to-9f62f5 5f693b),django could be used to develop these applications:

- Web app or API for the backend

- Quick/rapid development

- Applications that demand strong security

- The codebase encapsulates the web app and the API backend.

- Django does not require directly making database queries, instead it provides its own interface for database transactions.

2.3.3 What is web server?

As written on developer.mozilla.org (https://developer.mozilla.org/en-US/docs/Learn/Common_questions/What_is_a_web _server),a web server can be either a software or hardware, in terms of hardware ,this a storage for website's resources, for example: HTML, CSS, images, videos,.... and in terms of software, it is an Http server that interprets URLs and HTTP, this type of server can be accessed using domain name which then response to the client with the requested resources.

On the basis, in order to access a website, client must send an HTTP request to the web server hosting the files that will return those files after validation.

2.3.4 What ASGI & WSGI?

Based on an article by Positive Stud(https://medium.com/analytics-vidhya/what-is-wsgi-web-server-gateway-interfac e-ed2d290449e)

WSGI(Web server gateway interface): is an interface used to communicate between web server and the python application or frameworks such as Django. For example, once a web server has received requests from many clients, it requires interaction with the python application and for this to happen, WSGI needs to take part in the communication. The negotiation is carried out as follow:

The web server is able to send request to the WSGI container and callable objects with specific functionalities provided by the python application can be called by the WSGI container corresponding to the requests it received. The figure below demonstrates how the process is delivered:
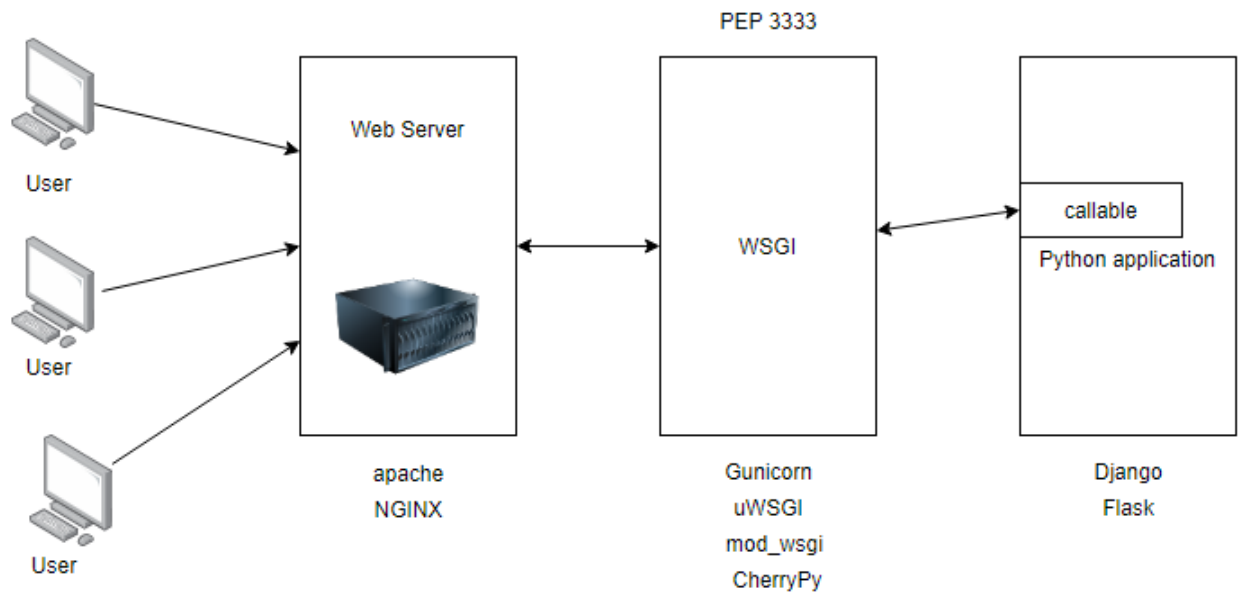
Figure 2.1. Communication between user, web server, WSGI, and Python application(https://medium.com/).

ASGI (Asynchronous Server Gateway Interface):

is derived from WSGI and allows Asynchronous interaction between Python application, frameworks and web server (https://asgi.readthedocs.io/).It was also mentioned that ASGI is constructed using 3 elements:

- scope: a dictionary containing information about a specific connection. Each connection from a client to the ASGI application is a call to the application callable that handles the connection and "scope" store the life span and information about each connection. For example, for Web socket protocol, the live span of the connection is the time it has connection to the python application and scope stores the connection web socket's path

- send: a callable for sending event messages to the client

- receive: a dictionary containing key-value pair of many event messages to the application.

Events: are divided by ASGI into 2:events that the python application receive and event responses that the application send. Each event is a dictionary with a key "type", describing the type of the event's message that be freely invented by the user.

Asynchronous request	Synchronous request
Allows clients to send multiple requests and still be able to continue execution	The client must wait and block until the response is received in order to continue

while waiting for the response	execution

Table 1. Comparison between Asynchronous and Synchronous

2.3.5 ASGI & Web socket

Web socket protocol: is a stateful protocol that uses client-server communication, thus making the life span of the connection last until one of the member disconnect. The process is as follow:

1/Client-server agree on a connection by making a handshake and this connection is kept alive as long as both members are still connected

2/While alive, messages in the communication is bidirectional, which means the client and server can both send and receive messages from each other.

3/If either the client or server dies or disconnect, the communication ends, and the connection is closed

(https://www.geeksforgeeks.org/)

Web socket protocol relationship with asgi:

There are many components for web socket connection through asgi:

✧ Scope:

Web socket scope is stored in the scope dictionary of the callable, this contain meta data about the connection and must contain at least a "type" key describing the type of the connection, for example: http or WebSocket. The keys from scope include:

- type: Describing the type of the connection, for example, http or web socket
- asgi["version"]:version asgi spec
- asgi["spec_version"]:the version of asgi that this server interpret
- http_version:version of http
- scheme:url scheme,such as:ws,http,wss
- path:the url of the request removing the all query string after it
- raw_path:unmodified http path
- query_string:portion of the url after "?"
- root_path:the root path of the application
- headers:a dictionary containing the request's headers
- client:a key-value pair storing the client's IP address and listening port
- server:a key-value pair storing the server's IP address and listening port

- subprotocols:Sub protocols the client advertised

◇ Connect - receive event:

This event happens when the server receive requests from the client when the connection is opened and about to be closed. These requests must be responded with an Accept or Close message before being passed. These messages(Accept or Close) must be sent during the handshake process and the handshake only ends if the server receive a reply. The keys from receive include:

- type: websocket.connect

◇ Accept - send event:

Sent by a client to accept a connection. The keys include:

- type:websocket.accept
- subprotocol:the subprotocol to be to accepted
- headers:a dictionary storing headers of the message.

◇ Receive - receive event:

Messages for the server sent by the client. The keys include:

- type: websocket.receive
- bytes:content sent in binary form
- Text:content sent in text form

◇ Send- send event:

A request from the application to send a message to the client. The keys include:

- type: websocket.send
- bytes:content sent in binary form
- Text:content sent in text form

◇ Disconnect- receive event:

The application receive this if the connection is either lost due to the client losing connection, the server closing connection or socket loss. The keys include:

- type: websocket.disconnect
- code:close code for websocket

✧ Close- send event:

Sent by the application requesting the server to close the connection. The keys include:

- ■ type: websocket.close
- ■ code:close code for websocket

2.3.6 Django channels:

Allows handling other protocols apart from http ,such as ,web socket, chat protocol,... and is based on ASGI. In addition, channel also provide authentication, session system and more.

Since channels is based on ASGI, it also split the connection into sections: connection scope and events.

- ● Scope: Web socket scope last as long as the connection is kept alive. The scope is a dictionary storing information about the connections as discussed in the section before.

- ● Event: These events occur while the scope is kept alive and describe user such as sending a web socket frame. For each scope created ,channel will be instantiated to handle the events specific to that scope.

Sequentially, channel have a concept called Consumer, which takes event as input, then using a routing table direct the request to the corresponding consumer that can be specified as asynchronous or synchronous to handle the requests.

2.4 Real time application.

Having done the research about WSGI,ASGI and web socket, i am able to have a idea about the basic about the architecture of a web socket based application and that means the application will include the use of Web server ,ASGI server, storing scope and handling event requests sent by the ASGI server to the backend application which will then be "consumed" or processed by the consumer that corresponds to the path the request is sent to. In addition, i believe that in order to understand the implementation better, it might be a good idea to take a look at an example of a real time application that utilizes Django's channels and a front end framework to develop responsive website.

2.4.1 Example of real time application and its implementation

Using reactjs and Django stack , an example of a real time application could be a chat box as provided by
Ruben(2018,https://revs.runtime-revolution.com/a-simple-real-time-chat-with-django-channels-and-react-b73edc3a79f2), the implementation is as follow:

Requirements:

- Fronted: React yarn(or node.js)

- Apophony 3,pip,redis server, PostgreSQL

Set up:

Backend

1. Creating and migrating Message model for storing messages from users

2. Setting up Channel and point to Redis server location

3. Define the routing table for channels, this directs all web socket requests to the consumer corresponding to the path in the routing table, which will handle the requests.

Frontend:

1/ In App.js of the react app, create a function called "create", which will add callbacks that are invoked when web socket opens, receives messages closes or error.

2.5 Application critics.

Now that the foundation of the implementation has been established as above, we need to examine and evaluate existing application with similar features that the project desires.

2.5.1 Existing application with similar requirements

As a result of researching, I found that the availability of real time drawing app is limited, even though there are a variety of visualization   application that allows similar features to live drawing, some of them are: Miro ,which is a web application for collaboration that provide tools for visualization such as an infinite canvas along with widgets and the ability to communicate through video calls as well as screen sharing. However, this type of application's goal is to aid project and meeting management whereas my project is focused on providing chat room and drawing tools that artists are familiar with in art industry. Nonetheless, I was able to find an application called Magma studio ,which is an application that has ideas of chatting through artistic visualization. The website allows users to create their own or join another user's room to draw and communicate through text or voice chat as well as providing a real time experience, which means that every user in the same room has access to drawing on the same canvas, possibly resulting in relaxing and comfortable socialization. I also wanted the project to allow users to have a platform for sharing the canvas belong to their room and their own artwork in order to interact with other user and express themselves. Despite sharing important features to my application Magma studio has not had a platform for sharing at the moment, because of this , a type of social media app needs to be examine. In terms of sharing pictures/images, i have found several website that specifically aiming at artist: Artstation and pixiv,both of which satisfy the needs for the project. Below is the comparison with other similar application:

Requirements	Magma studio	Artstation	Pixiv
Communication and live drawing			
Room chat box	Yes		
Live drawing	Yes		
Room creation	Yes		
Individual chat box		Yes	
Canvas	Yes		Yes
Digital tools for painting	Yes		
Live streaming room activities			Yes
Have private or public room			
Send request to join a room	Yes		
Hosting room	Yes		
Voice chat	Yes		
Auto saving canvas of a room	Yes		
Sharing platform			
Each user has a gallery		Yes	Yes
Posting images		Yes	Yes
Like/Comment on posts		Yes	Yes
Real time notification		Yes	Yes
Post canvas			

from a room			
Simple and accessible UI	Yes	Yes	Yes
Profile for each user		Yes	Yes
Discovery page		Yes	Yes

Table 2. Comparison with other existing application

2.5 Conclusion.

As a result of the research, it is possible to conclude that an application focusing on communicating through posting and drawing/chatting is not widely available and the available applications lack the needs for interaction between users as they only a sub part of features in terms of communications, for example ,Magma studio provides the needs to online communications through drawing and chatting, however user can only interact with people they know as there is no platform for communicating with other people such similar to Artstation.The project focus on these aspects, which form its attraction points. The project will have to have a simple to use but artistic design as the targeted stakeholders are artists, this also means there need to have a lot of core tools popular in the digital art industry so that user can easily become familiar with the application quickly.

2.6 Summary.

In this chapter , I have discussed about the problem needed to be solved, which is depression as well as gave an introduction about digital art and the required tools to start digital art.In addition, Django and its uses/operations has also been explained, this allows for further understanding of how the concept of WSGI and ASGI have been applied into Django channels ,thus allows for sending asynchronous requests making the application works in real time. Finally, I have compared the differences between the project and the different existing software to improve the quality of the project.

Chapter3: Requirement specifications

Requirement is an essential aspect for a project because it helps form the foundation and the features the project must have in order to achieve the desired result. In this chapter , we will go through the requirements for the projects which consist of : Functional requirements and Non-functional requirements. These requirements should be able to identify the functionality the project must have as well as the performance of the app as it performs these these operations.

3.1 Functional requirement.

Requirements

General

System must allow many users to create a drawing room

System must allow many users to draw on the same canvas

System must provide a list of brushes and lasso functionalities for users to use

System must let user create an account that has its own profile

Each profile has a gallery storing artworks posted by the profile owner

System must allow users to interact with other users and posts

System must allow user to switch from gallery mode to drawing room mode

System must allow user to invite people to room

System must allow user to join a room

User

User must be allowed to create an account

User must have a join code to be invited to a group

User must be allowed to create a drawing room

User must be allowed to join a room

User must be allowed to be a member of many rooms

User must be allowed to switch from Drawing room mode to Gallery mode

User must be able to login/sign in

User must be able to like post

User must be able to comment on post

User must be able to share post

User must be able to save post

User must be able to search for post/artist

User must be able to post

User must be able to delete post

User must be able to update post

User must able to receive notification in real time

User has these attributes:

firstname,lastname,email,password,occupation,working,location,interest's list, about me section,1-3 preview pictures, following, follower, brushTips, joinCode

User mood

User must has a mood attribute to monitor their mood in 4 range:sad,normal ,happy,excited

User's default mood is happy

User's mood is changed by the type of images they click on

User's mood affect the discovery tab. The tab will display images corresponding to the mood

Drawing mode (DM)

DM must have a collection of room that the user created

DM must have a collection of room that the user is a member of

DM must allow user to change to Gallery mode

DM must display information of each room including: no.member,host's name,on line status, group's name

DM must allow user to search for a group they are hosting or joined using its name

DM must allow user to enter a room number and join it

Drawing room(DR)

Every room must have only 1 single host

DR must have a canvas for many users to draw on

DR must provide a list of tool for users to use on the canvas

DR must provide a brush/eraser system for user to use

DR must provide a layer system for user to use

DR must provide a Color Palete system for user to use

DR must allow user to invite other account to a group using that user's join code.

DR must allow only host to be able to accept or decline a join/invitation request

Each join/invite request to a room from non-host user must be put on pending and wait for the host to accept/decline

DR must display the pending request to host user only

Creating a room
User must first enter: name of room/search and on the room to join if its a public room or send for pending to join a room
User then can optionally choose to invite a user to group by entering there join code if the group is private
A room must have these information: ID,room name,roomCode,members,host,pending,publicity

Joining request(Private room)
User must enter the roomCode to enter a room
User requesting to join a room must wait for the host to approve the request before joining.
Join request must have:ID,requestUser,group

Invite request
User must enter the joinCode of a user to make an invite request for that user to the group
The invited user must wait for the host to approve the request before joining the group
Invite request must have: ID,inviter,invitee,group

Canvas
Canvas must be allowed to have many layers with the maximum of 10 layers
User must be allowed to select a portion of the canvas to draw in using the tool system
Canvas must store these attributes: ID,room

Layer
System must have a panel for storing and modifying the layers
User must be able to create a layer for a canvas
User must be able to delete a layer for a canvas
A layer must have these properties: mode and opacity
A layer must have different mode including:Normal,Multiply,ColorDodge,Color,Darken
A layer can be toggle off/on
Layer must store these attribute:ID,Canvas,Image

Tool
List of tool must include:Move tool,square selector,circle selector,lasso tool,eyedropper,paintBucket,Brush,Text,goBack,goForward,zoom in, zoom out

Brush
User choose the brush tool to use the brush
A brush has many tips and has 5 tips by default, default is a round tip
A brush has these properties:size,density,opacity,hardness,color
User can import and upload their own brush tips
User can apply pressure to these properties:size,density,opacity

Gallery mode

Sign up
User is asked for their first name,last name,email,password,password confirmation ,occupation,where they are working,other information they would like to provide to their bio.
User is asked to choose their interest from a list
User is asked to write an About me summary and provide 3 pictures of their artwork
System must save into database the user's firstname,lastname,email,password,occupation,working location,interest's list,about me section and its 3 pictures and assign an ID number to the user

Login
User must enter their user email ,password
System must validate those information with records from database
System must provide a message confirming the user that they have logged in successfully
System must provide a message confirming the user that they have logged in unsuccessfully
User must be able to logout

Post
A post must have these attribute:list of images(maximum 10),summary,number of like,currentInteraction(updated every hour),lastInteraction(updated with currentInteraction every hour) , number of comment,comments,mood,interest,the author.These must be stored in the database and assigned with the author's ID

For user different from the post's owner,there is a like/share/save button
Owner must have option to edit/delete the post
Owner while editing the post can delete/add images and change the summary ,once the images are changed,the post mood is updated.These information is then used to update the record in the database
Posting a post
User must upload images or videos and write a summary
User must enter at least 1 type of interest the post belong to such as : illustration,3d moddeling ,2d art
System must determine the mood of the image by monitoring the Dominant Color of the image
Discovery
The tab must show trending/latest/following posts
The tab must allow user to decide what type of post to show based on type of interest
The content shown must be based on the trending/latest/following filter, type of interest and the user mood
User homepage
Homepage has Gallery page storing user's post
Homepage has Follower page showing who is following the user
Homepage has Following page showing who the user is following
Homepage has About me page storing user's information including a summary and 3 pictures
Homepage has a follow button for account that is no the user
Home page always display user's name,occupation,working location,other information input by the user

Table 3.Functional requirement table

3.2 Non-functional requirement.

- Scalability: The project is a social and communication application which stores a large number of data that need to be extended once the limit is exceeded, if this happens , the increase in the number of node for storing this data is critical. In order to do this it is important to choose a DBMS that allows scalability ,for

example, a type of database that allows the increase in the number of nodes like mongo dB.

- Reliability: The application is real time performance dependent, this demands the system to be running with minimal crashing chance of 1-5% and alive time of the server must be above 90% of the time together with real time updates of user data and notifications.

- Latency: The runs on a large database of users ,which mean there will be a large number of transactions that need to deliver in time so that the user feel satisfies. In order for this to happen, transactions might have to take advantage of indexes in Misusing algorithm from the back-end for sorting data into a specific format in the shortest time possible or sending data in batches so that there is more time for sorting. For the front end , lazy loading could allow some time for the back end to sort data as this only requires the back-end to sequentially send small amount of data.

- Security: The system stores and make transactions with user's data which need to be save and inaccessible to other, this requires a secure way of transferring as well as preventing Injections, Cross-site scripting and other form of hijacking. The application must also protect the privacy of user from different chat room by allowing only the host of the room to be able to accept/decline request from users.

3.3 Use case examples.

The application has 2 main pages, draw&chat page and gallery page, in this section, I am going to discuss 2 core use cases for each of them.

- Draw&Chat:

Use case 1:Creating a room

Prerequisite: User have created an account on the website

1/Initially, when the user enter page , they will need to provide their user name and password to login

2/Once logged in, if the user has rooms that they are hosting or they have already joined more than 1 room, they will be directed their room's list(list of rooms they have joined or are hosting).

 2.1/User click on the "Create room" button to create a room

 2.2/User enter room's information

 2.3/User click submit

 2.4/Once the room is created, the user is directed to that room and provided with a canvas belong to that room along with the tool UI

3/If the user is a new account, meaning they have not joined or are hosting a room yet, they are directed to the room create form

 3.1/User create room by entering the room info

3.2/User click submit

3.3/User is directed to the created room

Use case 2: Draw

Prerequisite: User have created an account on the website

1/Initially, when the user enter page, they will need to provide their user name and password to login

2/if the user has created or joined more than 1 room, they will be presented with a list of those rooms:

2.1/The user click on one of the rooms they want to start drawing in

2.2/The user enter the room and presented with a canvas and tool for drawing

3/If the user has not created a room yet, they have an optional choice to create a room or join a room once login

3.1/User create or join room by following the create/join room use case

3.2/Once one of those use cases is done, they will be directed to the room they have chosen or created

3.3/User start drawing in the room

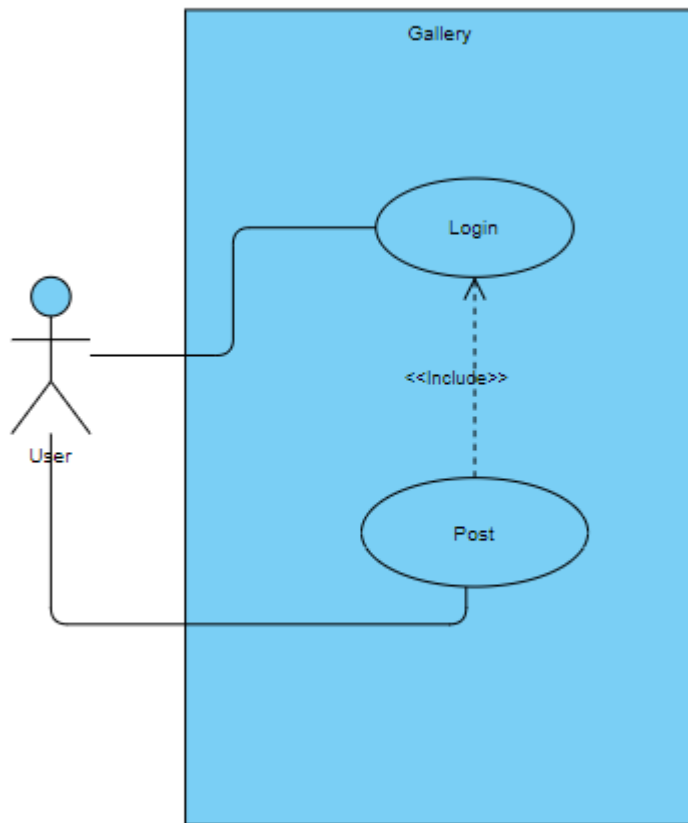
Below is the Use case diagram for these 2 use cases:

FIGURE 3.1.USE CASE DIAGRAM FOR DRAW USE CASE AND CREATE ROOM USE CASE

● Gallery:

Use case 1: Posting image

Prerequisite: User have created an account on the website

1/Initially, when the user enter page, they will need to provide their username and password to login

2/User is directed is checked whether they have created or joined a room yet, they can optionally answer this and is allowed skip to the room's list page

3/Once in the room's list, on the top left corner, user click on a button to change to gallery mode

4/Once in gallery mode, user will already have their own profile created and can start posting.

5/User can start posting by clicking on the "+" symbol on the top right

6/User is directed to the Post submission page.

7/User enter the information for the post

8/User click submit

FIGURE 3.2 USE CASE DIAGRAM FOR POSTING USE CASE

Use case 2: Posting canvas from room

Prerequisite: User have created an account on the website and already have joined or created at least 1 room

1/Initially, when the user enter page, they will need to provide their user name and password to login

2/User is directed is checked whether they have created or joined a room yet, they can optionally answer this and is allowed skip to the room's list page

3/In the room's list page, user choose a room containing the canvas they want to post

4/User is directed to the room , where user click on the button to post the current state of the canvas to their profile
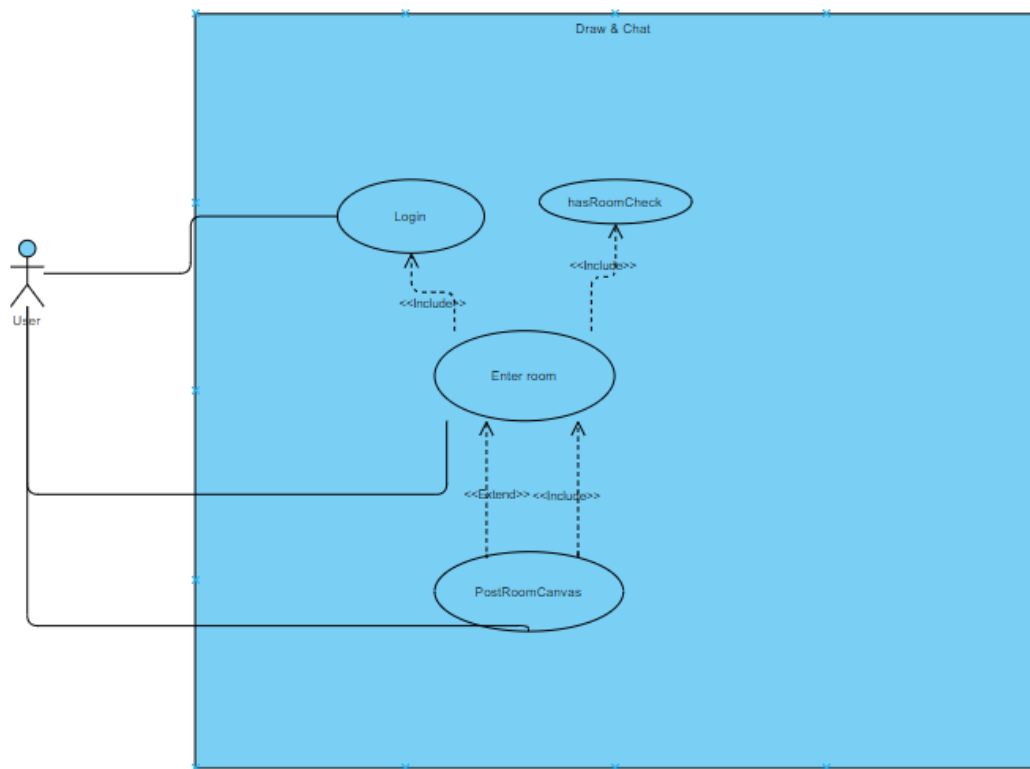
FIGURE 3.3. USE CASE DIAGRAM FOR POSTING CANVASES FROM A ROOM

3.4 Summary.

This chapter has identified the functional requirements while mentioning the nonfunctional requirements which the system needs to satisfy in order to give a good stream of experience to the user. In addition, 4 essential use cases of the application has also been discussed, this allows understanding of the system's purpose and give a general idea of how the process of each of those operations might be approached.

Chapter 4: Design and Implementation

In chapter 4, we are going to discuss about the design pattern that form the structure of the application, this give a broad view of the project's architecture and a general idea of component's relationships/behaviours, what they are for and the reason for choosing them. Furthermore, through the progression of the chapter i will also be narrowing down the topic from general to detail by explaining the implementation of the components and give clarification to the design choice.

4.1 Design

4.1.1 Overview

The application is web-based and developed using JavaScript for the front-end and python for the back-end. In addition, for the database, SQLite has been chosen as it is integrated as part of the python framework that will be discussed later. The database provides a REST API which is useful as this allows the development to focus only on the front-end/back-end integration and not data transaction.

4.1.2 Design pattern

The design pattern i have chosen to form the architecture of the application is Model View Template(MVT).The framework separates logic, interface and data into sections managed by Django framework, this allows encapsulation across different component ,thus making the testing process more approachable.

- M - Model stores data and relationship between class-based entities, which are then migrated to the built-in SQLite of Django framework. Model in Django has ORM mechanism for interpreting relationship type: 1 - 1, 1 - many, many - many.

- V - View receives request and data from URL dispatcher and client , then performs logic depending on the request and response to the client.

- T - Template can be html files that receives context and is rendered by view. Using context and JavaScript , the template displays to the client the desired information using the passed context with bootstrap or CSS.
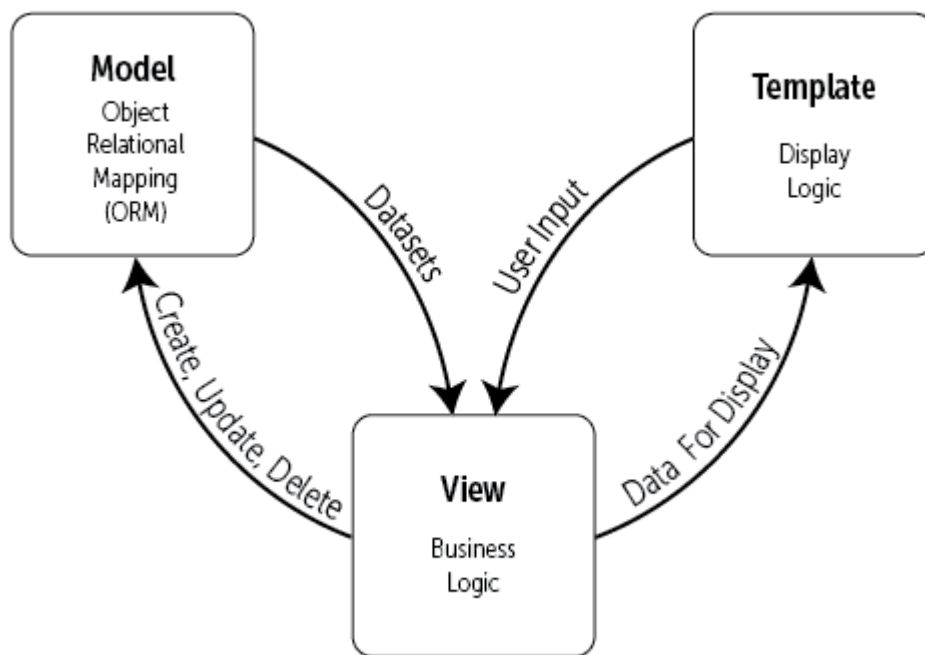
Figure 4.1 MVT model (Figure 3.1 https://djangobook.com/mdj2-django-structure/)

4.1.3 Framework

The project is divided into 3 parts: front-end, backend and database

- Front-end: Instead of using template for displaying information, the design uses Reactjs for a responsive design, this allows modularity as Reactjs website is built using combination of components along with the a wide range of module ,thus allows re-usability of code.

- Backend: Implemented using Django framework, this allows for past pace development for quick result, provide ORM which can be modified if needed and minimize the use of SQL query as the framework has API for translating python code to Sql queries and receive as query jQuery.

- Database: Django provides built-in database, SQLite. Moreover, Django has REST API for SQLite database ,this gives easy access and manipulation to database records.

4.2 High level design
4.2.1 Model

Model contains classes representing entities and their relationships, which are handled by Django or can be manually defined. In addition , beside class diagram, since SQLite is being used, ERD can be used for designing relational tables, normalization to minimize redundancy for better performance. The classes are defined as below:
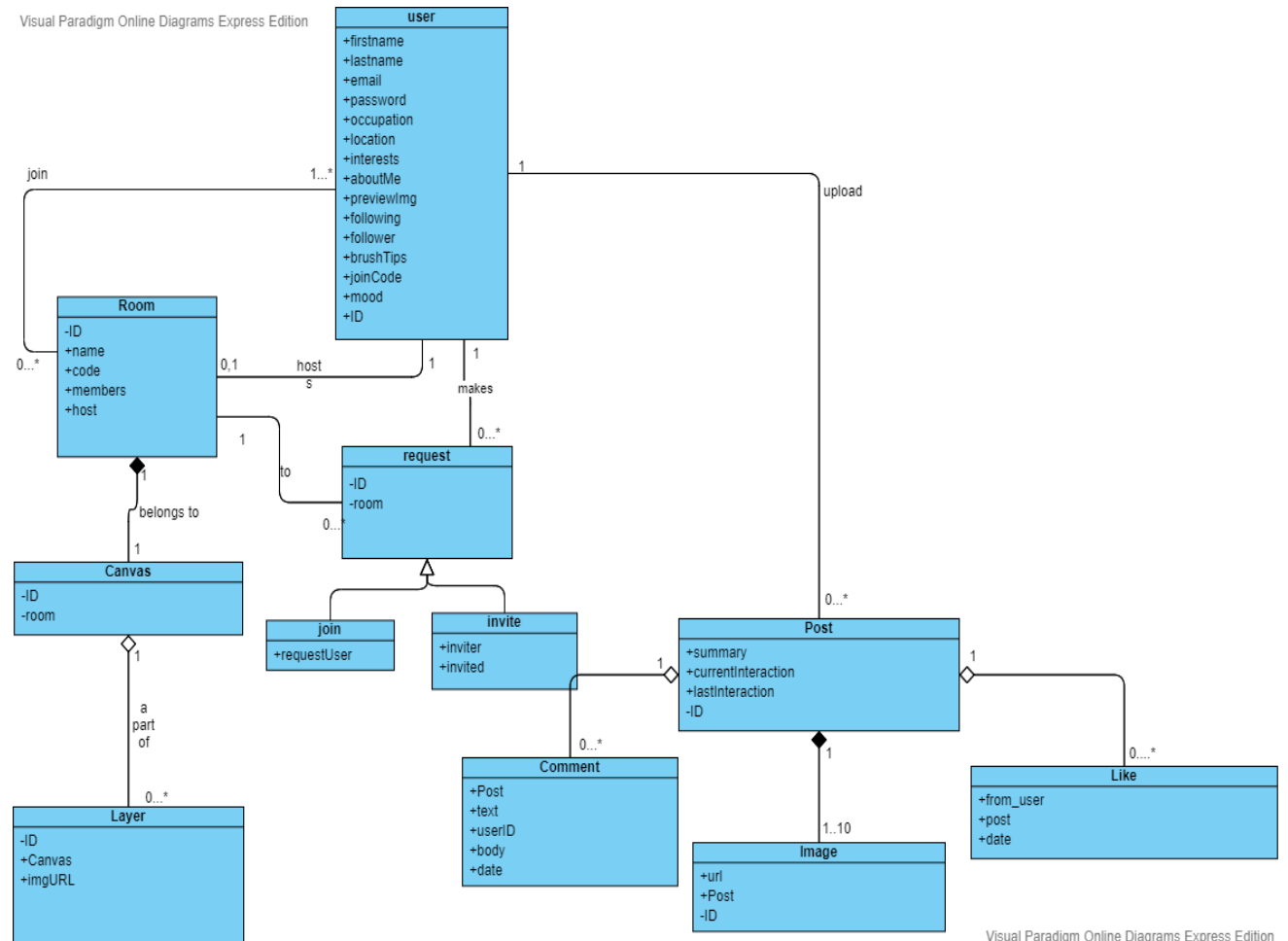
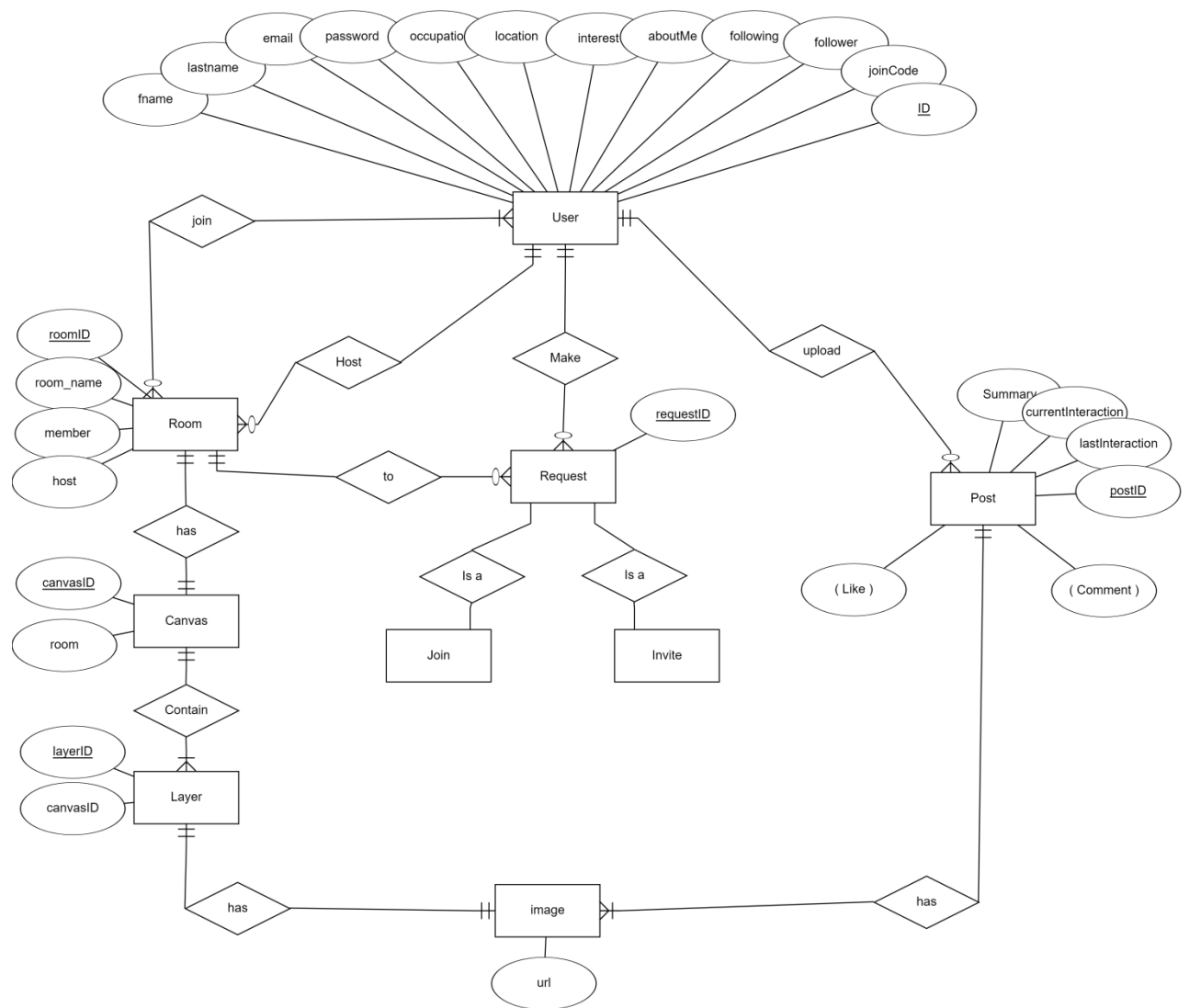**FIGURE 4.2.**CLASS DIAGRAM DESCRIBING THE ENTITIES IN THE PROJECT

FIGURE 4.3. ENTITY RELATIONSHIP DIAGRAM DESCRIBING THE MODELS REQUIRED

4.2.2 View

View contains logic that needs to be performed for different request from the client. When the client sends a request to the back-end handled by Django, the request initially arrives at the URL dispatcher, which is then forward to the view corresponding to the URL that the request is sent to in the URL table. Sequentially, the called view function will take as input, the request ,which is a dictionary containing the sent data that can be manipulated by the view. Finally, once the data has been processed , the response will be parse into json data and delivered to the client. In conclusion, View acts as a handler for different requests from the client and is called based the URL corresponding to it in the URL dispatcher. In general, the main views include:
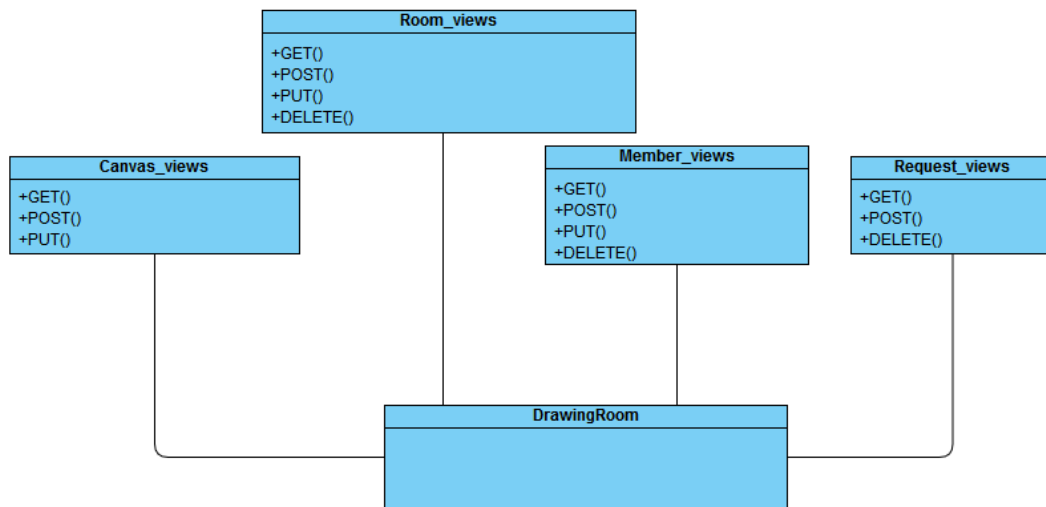
Drawing room:

FIGURE 4.4. VIEWS REQUIRED TO HANDLE **DRAWING** ROOM REQUEST FROM CLIENT
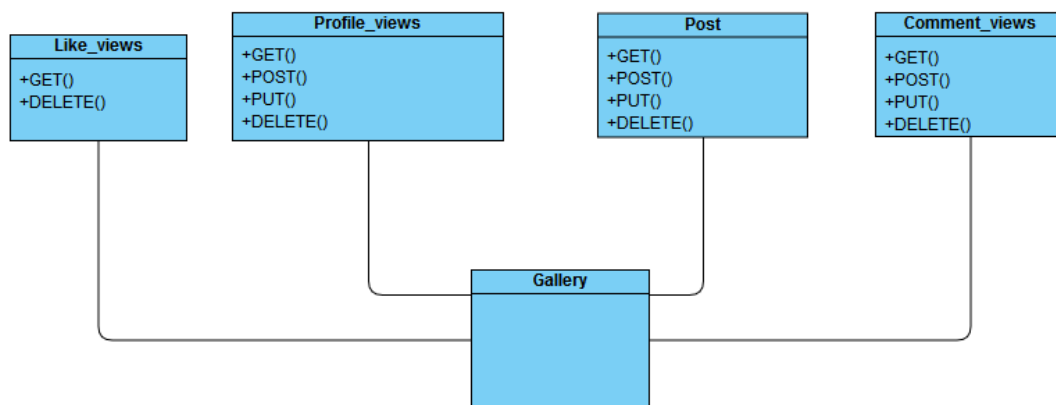
Gallery:

FIGURE 4.5. VIEWS REQUIRED TO HANDLE GALLERY REQUEST FROM CLIENT

4.2.3 Template

The front end take use of the tools and modules provided by Reactjs. The UI preserve the concept of modularity as it can be built upon layers of components that is specified by the developer. The goal of the front end is to take inputs from the client , format it so that it could be interpreted by Django and allow for sending data to the back-end API as well as receiving response data through web-socket protocol. The front end applies the use of web socket to the back end to create a bidirectional communication, which make the application dynamic and real time interactive. In general, the UI will consists of:
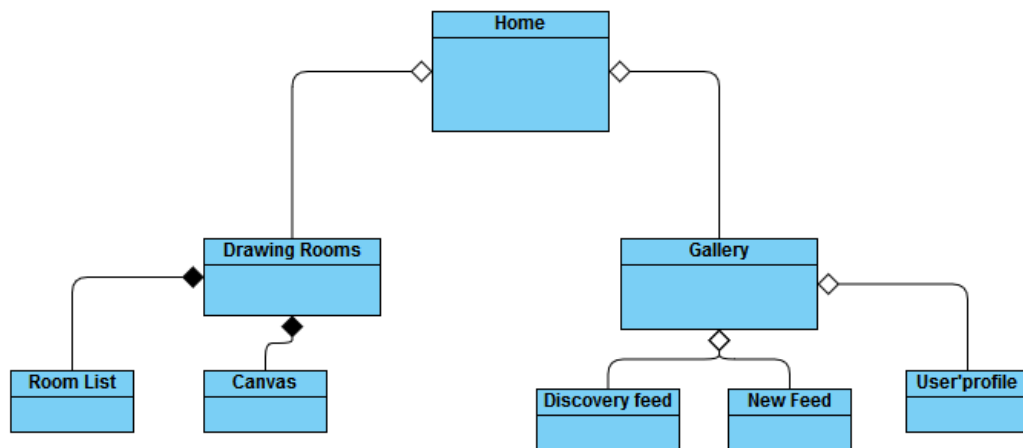
4.2.4 Database

The project is using Django built in SQLite. The reason is that the framework has an API for the database, this allows for quick data insertion/delete for testing and debugging. Even though ,the MySQL is more suitable in terms of scalability, security, since this a small social application ,SQLite is much quicker to use since does not require authentication, a server while still using sql syntax.
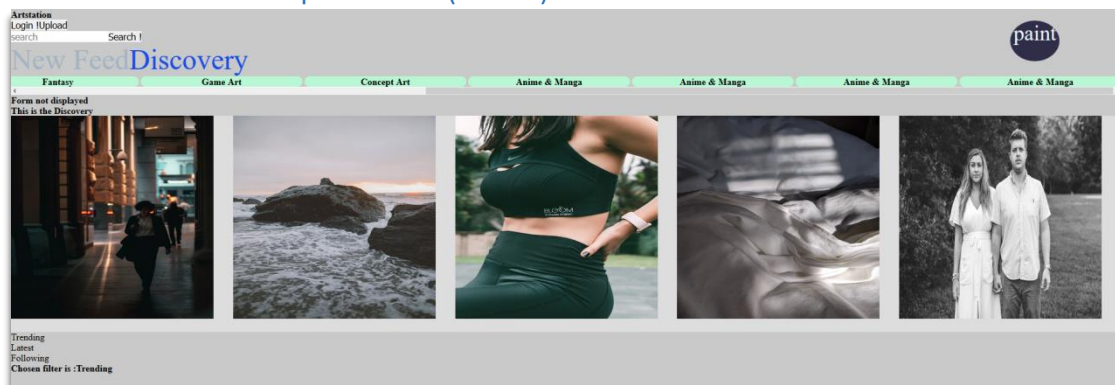
4.3Minimum vial product (MVP)

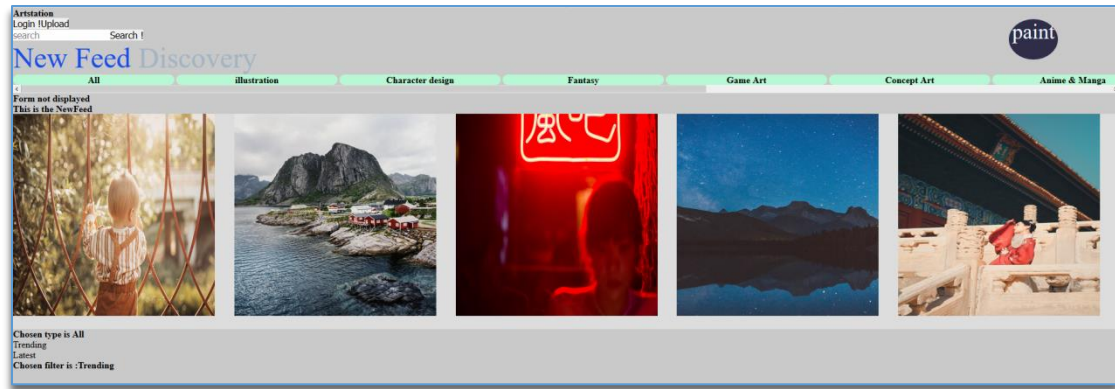**FIGURE 4.6 HOMEPAGE FOR VIEWING/POSTING IMAGES FROM FOLLOWING USERS**

FIGURE 4.7 HOMEPAGE FOR VIEWING/POSTING IMAGES FROM ALL USERS

THIS THE GALLERY PAGE, THE COMPONENTS OF THE PAGE INCLUDE:

- The main area for displaying posts from users that the account is following.

- A tab containing all the type of post: All, illustration, character design..., which can be clicked on by the user to filter all the post currently in the post area.

- A tab to filter trending or latest posts.

- Each post can be clicked on and then the user will be led to the post page displaying information about the post and its owner's account.

- A search bar for searching posts by keyword.

- The paint button will direct the user to the drawing page that will be documented later below.
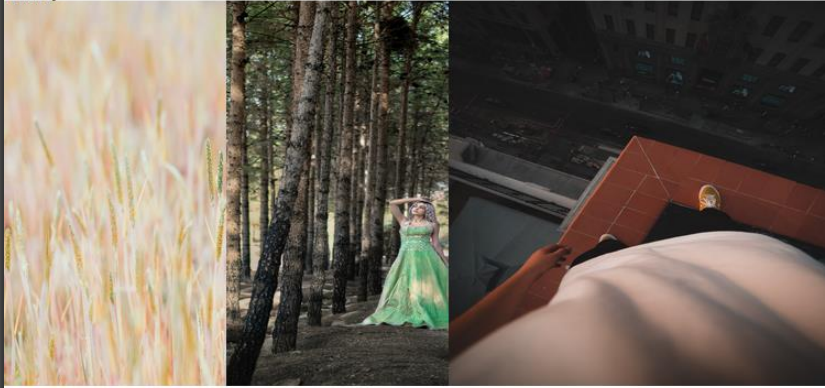
FIGURE 4.8 POST PAGE, DISPLAYING INFORMATION ABOUT A POST.

THIS IS THE POST PAGE FOR DISPLAYING A POST THE USER CLICK ON, THIS PAGE INCLUDE:

- A comment sections
- The post's image
- Like/comments number
- Post date
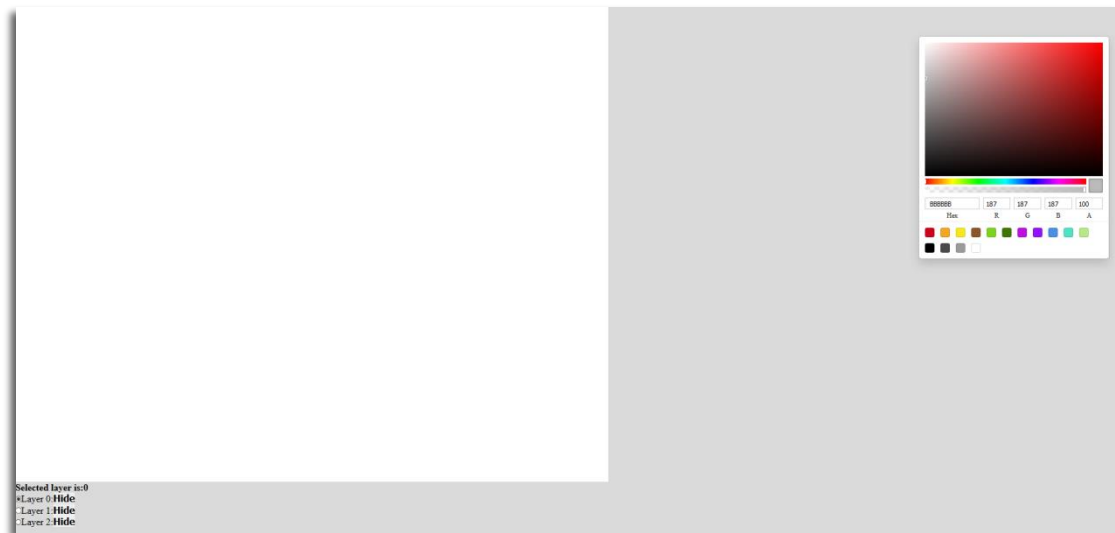- Follow/Like/Save button
- User's gallery

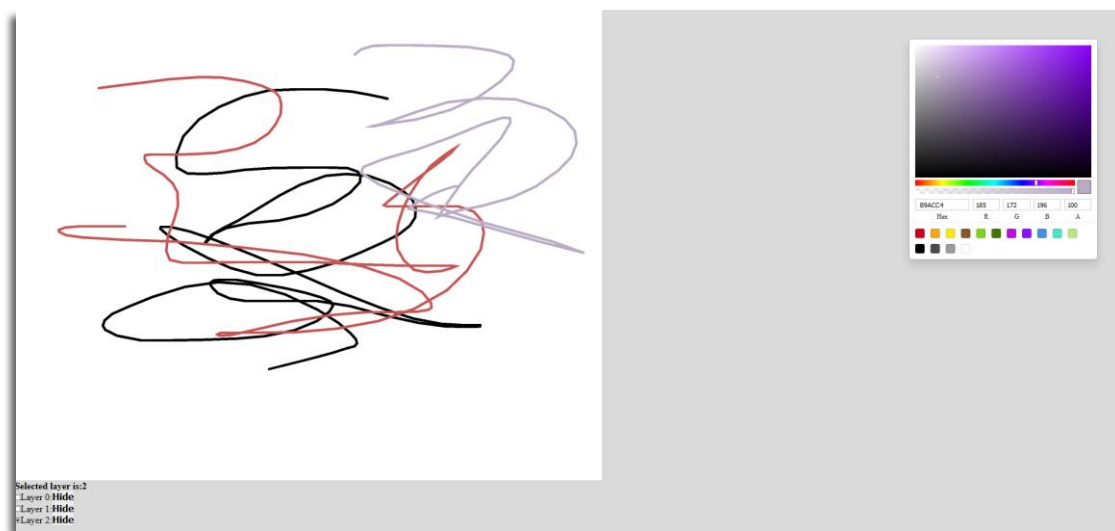**FIGURE 4.9. DRAWING ROOM PAGE , WITH CANVAS USED FOR DRAWING**

FIGURE 4.10. DRAWING ROOM PAGE, WITH CANVAS USED FOR DRAWING DESCRIBING USE OF LAYERING SYSTEM

THIS IS THE DRAWING ROOM , LATER IN THE DEVELOPMENT THERE WILL BE A LIST OF ROOM AND EACH ROOM WILL HAVE ITS OWN CANVAS AS ABOVE, AND EACH CANVAS WILL HAVE MANY LAYERS THE USER CAN DRAW ON. THIS PAGE'S COMPONENTS INCLUDE:

- A canvas with a list layers that can be turn on and off and user can decide on which layer to draw on

- A color board for choosing color

- Each layer be turn on/off in terms of visibility and interaction and user to check one and only one of the layers that they want to draw on.

Chapter 5: Risk management

Implementation				
Name	Description	Frequency	Impact	Solution
Developer experience	This risk happens during development when the developer has implementation related problems such as: not knowing how to solve a problem with using frameworks/programming languages, how to integrate backend into frontend, understanding how web socket protocol works,....	High	High impact	Avoidance: Do research about the tools and framework/modules needed for the development before implementation Solution: Scale the project down to suitable needs while still providing the initial requirements
Version control	The code is stored on the disk which means if the disk was to be broken or damaged, its data will be lost , thus losing all the project's resource	High	High impact	Avoidance: Use of version control such as git to store resources on a server so that we have a copy that can be pulled from or push to. This allows the code to be safely preserved for most of the time Solution: Scale down the project and find tools most suitable for fast

Name	Description	Frequency	Impact	Solution
				development.
Lack of resource preparation	This problems occurs when at the start of the project, after specifying requirements, a lists of tools need to be evaluated and selected to be used for development. However ,later in the project 's life-cycle ,an event such as requirement changes or an external frameworks/tools that have not been included initially are needed, this requires the integration of those tools into the application but several obstacles might appear including: developer is not familiar with the tools ,putting more time on planning the project leading to delaying.	Low	Medium impact	Avoidance: None Solution: Evaluate if the tools will take a long time to adapt , if not, proceed to apply plan and apply to the project, otherwise scale the use of the tool for the project down so that only the core elements are implemented.

<table>
<tr><td colspan="5" style="background:black;color:#f5a800;text-align:center">Requirement</td></tr>
</table>

Name	Description	Frequency	Impact	Solution
False collection	The foundation of a project is collecting the required elements from the client and knowing the aim of the application, if not prepared probably could leads to developing a product not in need or not satisfying a target group of stakeholder	Medium	High impact	Avoidance: Focus on the targeted requirement and stakeholder of the application and implements core features instead extra features that contribute little to the product and compare the aim of the end product with the requirements noted.
Requireme	Application is usually develop based a number	Medium	Medium impact	Avoidance: None

nt change	of predefined requirements, however ,their might be changes need to be made due to the need of the customer has changed or the scale of the projects, thus demanding evaluation and adjustment to the code			Solution: Analyze and apply framework/tools the take the shortest to adapt to satisfy the changes
Requireme nt difficulty	Requirements form the foundation of project, however if the requirements exceed the capability of the developer, this could be a problem as this requires knowledge of different methods to satisfy the requirements	High	Medium impact	Avoidance: Understand the fundamentals required to handle to the project, such as , frame works and protocol. Solution: Analyze choose the tool that provide the quickest time to learn while providing the fast pace development to complete the requirement in the most effective way

Time management				
Name	Description	Frequency	Impact	Solution
Planning	Once the analysis process is done., next step is to create the timeline and milestone for the project. This is important as the project must meet the deadline and satisfy the requirement, if the planning and time-line is not planning equally in terms of work effort and length and not stating the core milestone for the core features, it would be confusing or	High	High impact	Avoidance: Give more time than needed for each planned task to ensure that there is extra time to handle time consuming problems that possibly occur later Solution: Identify the big milestones for project and using those milestones to establish the timeline plan.

	might leave out some core aspect of the application			
Delay from changes	In the life cycle of the project, many problem could occur in term or changes: requirement changes, lack resources leading to extra needed for adapting to new tools,....As a result , the plan deadline couldn't be met.	High	High impact	Avoidance: None Solution: Scale down the project to be suitable for the timeline.

Personal

Name	Description	Frequency	Impact	Solution
Personal health	During development, it is unpredictable what cold or disease a person might catch, if this happens project delay will happen	Low	High impact	Avoidance: Spread the time so that there is rest time between milestone while still meeting the deadline and taking care personal health
Environment	A project from a group or individual can have member from anywhere ,whether it is a different country or has a different time zone or the place the member living in is unideal, for example lack of internet	Low	Medium impact	Avoidance: Divide the work equally depending on the environment condition of member
Other projects	There are other projects a member might have, this also contribute to the ability to work punctually and meeting the deadlines. This is because, for example, if ones has to do a side project along with this project and still has to follow the planned timeline, the result	Low	Medium impact	Avoidance: Divide the work equally so that there is time for other projects and task Solution: Divide the work equally so that there is time for other projects and task

Name	Description	Frequency	Impact	Solution
	might be not being able to finish his/her part at the right time or the task is done poorly			
	Copyright			
Name	Description	Frequency	Impact	Solution
Plagiarism	Using resources on the from other existing software is a way to identify the flaw of the project however, it is important that the project does not copy the implementation of the others	Medium	High impact	Avoidance: Only compare the project and the others existing ones to identity and understanding the implementation then think of a way to implement.
User plagiarism	The end product might be plagiarized or stolen by other individual who is not a part of the project.	Medium	High impact	Avoidance: Avoid exposing or leaking project's resources before publishing and have strong security to protect source code.

Table 4.Risk management

Reference:

1. Tanner Christensen(July 2013) .https://creativesomething.net [Online]

   https://creativesomething.net/post/55508909341/the-link-between-depression-and-creativity-and

2. Nuacht UCD(June 2012),www.ucd.ie[Online]
   https://www.ucd.ie/news/2012/06JUN12/130612-Socialising-helps-to-alleviate-symptoms-of-depression.html

3. NHS[Online]www.nimh.nih.gov

https://www.nimh.nih.gov/health/topics/depression/index.shtml

4. Lindsay Dodgson(July 2018)[Online]

https://www.insider.com/the-link-between-creativity-and-mental-health-2018-7

5. Bulfut(2019) SocializationHelpsTheTreatmentOfDepressionInModernLife

6. rtor.org(July 2018)[Online]

https://www.rtor.org/2018/07/10/benefits-of-art-therapy/

7. Concept Art Empire[Online]

https://conceptartempire.com/digital-painting/

8. Adobe.Adobe's beginer's guide to digital drawing tools[Online]

https://www.adobe.com/uk/creativecloud/illustration/discover/digital-pens-digital-tools.html

9. Django.Django.com [Online]

https://www.djangoproject.com/start/overview/

10. Kalpit(Aug 2018).When To Use(And When not to).medium.com[Online]

https://medium.com/crowdbotics/when-to-use-django-and-when-not-to-9f62f55f693b

11. developer.mozilla.org[Online]

https://developer.mozilla.org/en-US/docs/Learn/Common_questions/What_is_a_web_server

12.   Positive Stud(September 2020).What is WSGI(Web Server Gatewat Interface).www.medium.com[Online]

https://medium.com/analytics-vidhya/what-is-wsgi-web-server-gateway-interface-ed2d290449e

13. ASGI(March 2019).ASGI(Asynchronous Server Gateway Interface)[Online]

https://asgi.readthedocs.io/en/latest/specs/main.html

14. Geeksforgeeks(Dec 2019)[Online]

https://www.geeksforgeeks.org/what-is-web-socket-and-how-it-is-different-from-the-http/

15. Ruben Dinis(2018).A simple Real Time Chat with Django Channels and React[Online]

https://revs.runtime-revolution.com/a-simple-real-time-chat-with-django-channels-and-react-b73edc3a79f2

16. Djangobook.Django's structure-a heretic's eye view,Figure 3.1[Online]

https://djangobook.com/mdj2-django-structure/