Pakistan Vaping Community

Submitted by

Fawad Naeem

Supervisor

Sir: Zeeshan Haider Ali

DEPARTMENT OF COMPUTER SCIENCE

INSTITUTE OF SOUTHERN PUNJAB, MULTAN

July 2020

Pakistan Vaping Community

by
Fawad Naeem

A report
presented to the Institute of Southern Punjab,
Multan in partial fulfillment for the degree
requirement of Bachelor of Science
in
Computer Science
2020
Multan, Southern Punjab, Pakistan

AUTHOR'S DECLARATION

I hereby declare that I am the sole author of this report. This is a true copy of the report, including any required final revisions, as accepted by my examiners. It is further declared, that I have fulfilled all the requirements in line with the Quality Assurance guidelines of the Higher Education Commission.

Abstract

As a student of computer science and having interest in web development technologies I have built a social platform for ex-smokers, smokers and vapers. There are a lot of smokers whom are trying to quit smoking because of their health issues they are facing or for their batter health but most of the smokers whom try to quit smoking without any gaudiness they end up again smoking because of the side effects they face in the process. There's a saying "No one understand unless they've been through it" keeping this quote in mind I have built a platform to provide a circle of like-minded people whom are trying to quit smoking. This platform will provide a circle of like-minded people so they understand each other's problems what they are going through so they can build a better communication channel to help each other my share problems, experience and solutions for the problem. Ex-smokers will be able to help others by sharing while quitting the smoking habit what kind of alternate they use and what alternate worked for them in the process.

Purpose is helping the smoker so they can easily quit smoking without any complications.

Acknowledgements

I would like to express my deep and sincere gratitude for my university, institute of southern Punjab for assigning Sir Zeeshan Haider Ali as my supervisor. I would also like to thank my supervisor for giving me the opportunity to do my project under his supervision and providing invaluable guidance throughout this project. His dynamism, vision, sincerity and motivation have deeply inspired me. He has taught me the methodology to carry out the project and to present my project as clearly as possible. It was a great privilege and honor to work and study under his guidance. I am extremely grateful for what he has offered me. I would also like to thank him for his friendship, empathy and great sense of humor.

Table of Contents

List of Tables

Table of Figures

Chapter 1 Introduction

As a student of computer science and interest in web development technologies I have built a social platform for ex-smokers, smokers and vapers. This platform will be a social circle of those whom are struggling with their smoking habits and want to quit smoking. I am one of those smokers, I struggle with quitting smoking without any proper guides, did not aware of the side effects and again I found myself in this habit after sometime. This platform will be used to speared the awareness of side effect of smoking and the withdrawal while quitting some people will face, this awareness will be speared by the ex-smokers and their experience on the platform. Ex-smokers will share their journey of smoking, how they got this habit and what were the reasons behind their smoking habit, how they struggle with smoking and on what point they decided to quit smoking. While quitting the smoking habit what kind of alternate they use and what alternate worked for them in the process. Vapers will share the experience of their vaping journey and gaudiness for new vapers about the vaping devices and technologies used in the devise so new vapers know what kind of device suites them and what type of device will help them in the process of quitting smoking. Vapers enthusiastic will share about the devices which they used in the past, which device they are using currently and which is the best upcoming device in the market. Vapers enthusiastic will share their reviews about the profile of the E-liquids available in the market and which profile they like to use again and again and share their experience about the E-liquids. New vapers will benefit from their experience and will be able to make a good decision about their device and E-liquids which will help them in their vaping journey.

There will be articles on the platform for the vapers and smokers. Articles will be about the vaping and smoking. Vaping related articles will be about launching of new devices and about their technologies which are used in the devices. Articles will be about new E-liquids and about E-liquids profile the brand is using in their E-liquids. Articles will be on new researches about vaping and smoking. New researches are published frequently about the harms of smoking and their side effects. The articles will be based on the researches to speared the awareness to the smokers. New researches are being published about vaping and their misuse. The articles will be on these researches to spread the awareness of best practices in vaping so people can benefit from vaping and do not end up harming themselves instead of benefiting.

1.1 Problem statement

In existing system there are only few functionalities for users to use. User cannot manage profile information and do not have a proper reading material from official researches. Users cannot find other users by searching their names so they can have a private conversation.

1.2 Motivation

I used to smoke and tried to quit smoking many times but after sometime somehow, I'll found myself in this habit again. I tried different alternatives so I can quit smoking habit of mine, wasted my money and time in this process so it thought about a platform where all the like-minded people whom face the same problem can gather on like Pakwheels because Pakwheels provided a platform so like-minded people can gather on one platform.

1.3 Objectives

This platform will provide all of the needs of a smokers to quit smoking. Like-minded people whom face the same problem and different kind of alternate they used to quit this habit and vaper's enthusiastic sharing their experience and journey so others smokers can switch smoothly to vaping with proper gaudiness as an alternate.

1.4 Scope of the project

Project scope includes a profile module for user to maintain user information, a status module for socialization between users and a comment module so users will be able to comment on other user status. An article module where users can read articles and a module for commenting on articles.

1.5 SDLC models

SDLC is a process followed for a software project, within a software organization. It consists of a detailed plan describing how to develop, maintain, replace and alter or enhance specific software. The life cycle defines a methodology for improving the quality of software and the overall development process.

1.5.1 Waterfall SDLC Model

Waterfall – is a cascade SDLC model, in which development process looks like the flow, moving step by step through the phases of analysis, projecting, realization, testing, implementation, and

support. This SDLC model includes gradual execution of every stage completely. This process is strictly documented and predefined with features expected to every phase of this software development life cycle model.

Use cases for the Waterfall SDLC model:

- The requirements are precisely documented
- Product definition is stable
- The technologies stack is predefined which makes it not dynamic
- No ambiguous requirements
- The project is short

1.5.2 Iterative SDLC Model

The Iterative SDLC model does not need the full list of requirements before the project starts. The development process may start with the requirements to the functional part, which can be expanded later. The process is repetitive, allowing to make new versions of the product for every cycle. Every iteration (which last from two to six weeks) includes the development of a separate component of the system, and after that, this component is added to the functional developed earlier. Speaking with math terminology, the iterative model is a realization of the sequential approximation method; that means a gradual closeness to the planned final product shape.

Use cases for the Iteration model:

- The requirements to the final product are strictly predefined
- Applied to the large-scale projects
- The main task is predefined, but the details may advance with the time

1.5.3 Spiral SDLC Model

Spiral model – is SDLC model, which combines architecture and prototyping by stages. It is a combination of the Iterative and Waterfall SDLC models with the significant accent on the risk analysis. The main issue of the spiral model – is defining the right moment to make a step into the next stage. The preliminary set time frames are recommended as the solution to this issue. The shift to the next stage is done according to the plan, even if the work on the previous stage isn't done yet. The plan is introduced basing on the statistical data, received during the previous projects even from the personal developer's experience.

Use cases for the Spiral model

- Customer isn't sure about the requirements
- Major edits are expected during the development cycle
- The projects with mid or high-level risk, where it is important to prevent these risks
- The new product that should be released in a few stages to have enough of client's feedback

1.5.4 V-shaped SDLC Model

V-shaped SDLC model is an expansion of classic waterfall model and it's based on associated test stage for every development stage. This is a very strict model and the next stage is started only after the previous phase. This is also called "Validation and verification" model. Every stage has the current process control, to make sure that the conversion to the next stage is possible.

Use cases for the V-shaped model:

- For the projects where an accurate product testing is required
- For the small and mid-sized projects, where requirements are strictly predefined
- The engineers of the required qualification, especially testers, are within easy reach.

1.5.5 Agile SDLC Model

In the agile methodology after every development iteration, the customer is able to see the result and understand if he is satisfied with it or he is not. This is one of the advantages of the agile software development life cycle model. One of its disadvantages is that with the absence of defined requirements it is difficult to estimate the resources and development cost. Extreme programming is one of the practical uses of the agile model. The basis of such model consists of short weekly meetings – Sprints which are the part of the Scrum approach.

Use cases for the Agile model:

- The users' needs change dynamically
- Less price for the changes implemented because of the many iterations
- Unlike the Waterfall model, it requires only initial planning to start the project

1.6 Which SDLC Model you chose and why?

I have used V-shaped software development lifecycle (SDLC) model for my project development. My project requires testing and verification in the early stages because the next stage is started only after the previous phase and my project requirements are static and clear.

1.6.1 Procedure

Procedure of V-shaped software development lifecycle method includes

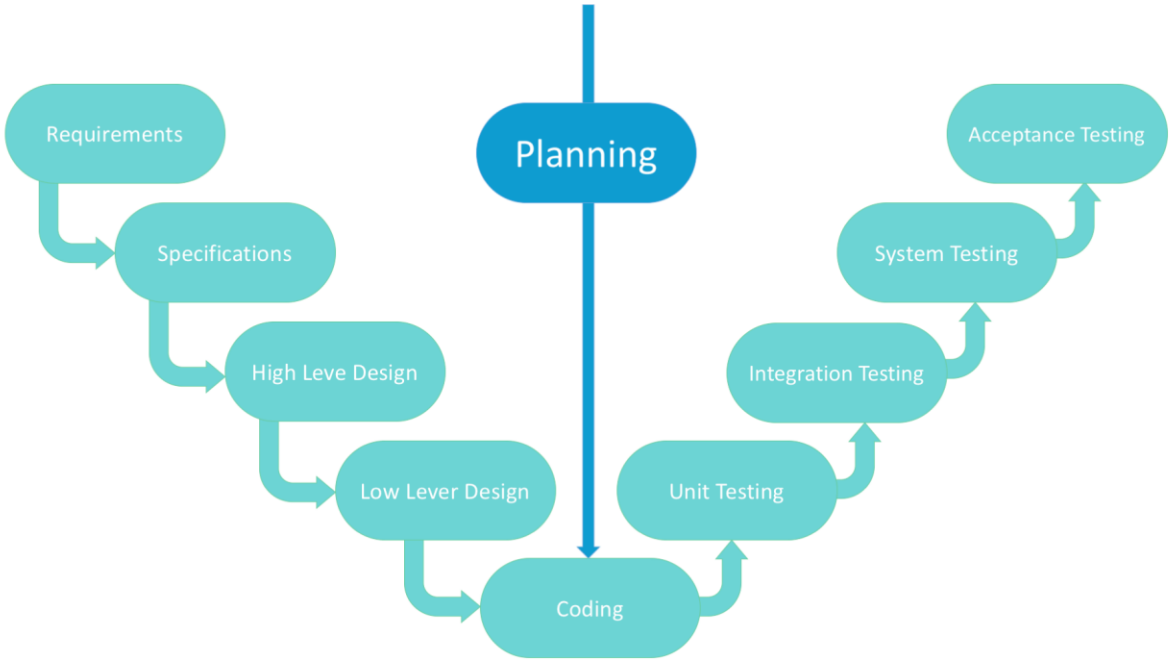

Figure 1

1.6.2 Requirements

In requirement phase I have collected the requirements for the project. Requirements for a social platform where users will make a profile to maintain user information, a user should be able to share thoughts by sharing a photo and by writing a status. A user should be able to socialize with other users on the platform by commenting on other user status and should be able to contact other users by privately messaging. A user should be able to read articles and should be able to comment on those articles which are written by the platform.

1.6.3 Specifications

Specification includes a profile module for user to maintain user information, a module status for socialization between users and a comment module so users will be able to comment on other user status. An article module where users can read articles and a module for commenting on articles.

1.6.4 High level design

In high level design front end technologies are used. In Front end technologies to design the front-end HTML are used, for styling the front-end CSS and BOOTSTRAP are used. To make a user-friendly design JAVASCRIPT is used.

1.6.5 Low level design

In low level design back end technologies are used. In back end technologies to design the back end MYSQL is used to design and connect to the database, PHP is used to show data from the database.

1.6.6 Coding

In coding phase all the modules are coded which we gathered in requirement phase. Front-end and back-end coding take place including the modules which are the specification for this project

1.7 Software tools

Visual Studio Code
XAMPP
Adobe photoshop

Chapter 2 Literature review and analysis

The proposed social platform and the existing social platform are analyzed and compared according to their functionalities. The existing social platform has only few functionalities to offer whereas proposed social platform contains more functionalities than the existing social platform. These functionalities help the users for friendly experience and encourage the user to use the platform more frequently.

2.1 Existing system

Existing system offers a user to make a prolife by providing the basic information name, email and quitting date of smoking habit but a user cannot maintain the profile by adding more information and by adding a profile picture or a cover photo. Existing system provide a status functionality where user can share thought by posting a written status and status commenting functionality where user can comment on other user's status. Existing system provide a functionality to user where user can privately message other users to have a private conversation.

2.2 Proposed system

Proposed system offers a user to make a prolife by providing the basic information name, email, birthday and country and a user can also maintain the profile by adding more information and by adding a profile picture or a cover photo. Proposed system provides a status functionality where user can share thought by posting a written status and status commenting functionality where user can comment on other user's status. Proposed system provides a functionality to user where user can privately message other users to have a private conversation. Proposed system offers a functionality where users can read new articles every day and can comment on those articles. Users can also search for the users on the platform if they want a private conversation with the user. Proposed system offers a search functionality where user can search for keywords used in the status to find the posts which are related to the keyword.

2.2.1 Information gathering

Information is gathered by online search and studying the existing system which is already up and running on web. Vapingcommunity.co.uk is the similar platform to this project which provide some functionality to the users.

2.2.2 Interview

Interviewed the users whom used the Vapingcommunity.co.uk platform. Most of the users used this platform are smokers or used to smoke and they switch to vaping for their health benefit.

2.2.3 Questionnaires

Asked about the difficulties they faced using the vapingcommunity.co.uk platform. What functionalities are missing from this platform which are beneficial for the them. What type of functionality will be engaging to users for using the platform more frequently?

2.3 Comparison table

Table 1

Pakistanvapingcommunity.com	**Vapingcommunity.co.uk**
Prolife module	Prolife module
Maintaining profile module	Non
Find users module	Non
Article module	Non
Article commenting module	Non
Messaging module	Messaging module
Status module	Status module
Status commenting module	Status commenting module

2.4 Feasibility report

Technical Feasibility.

Technical requirements to build this project meet the todays technologies. All the technologies used in the making of this project are commonly used.

Economic Feasibility.

Economically this project is really affordable. Open source programming languages are used to code this project, domains are relatively cheap now a days and hosting fees are affordable.

Operational Feasibility.

Operational cost is affordable for this project. This project operational needs are domain and hosting. Domains are affordable now a days and hosting fees are accordingly the use of hosting services.

2.5 Gantt chart

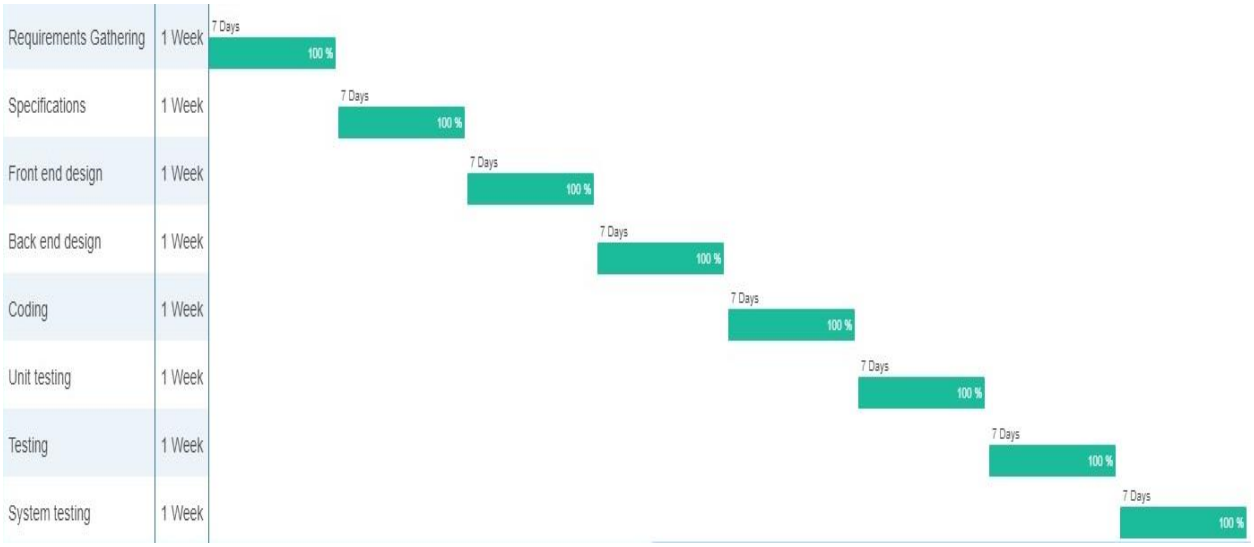

Figure 2

Requirements Gathering

Duration (one week)

Developing specifications

Duration (one week)

Front-end design

Duration (one week)

Back-end design

Duration (one week)

Coding

Duration (one week)

Unit testing

Duration (one week)

System testing

Duration (one week)

Chapter 3 Design

3.1 Introduction

Design phase include the designing of the platform where each step of development is proposed in a design which include the data flow diagram, use case diagram, Sequence diagram Class diagram, Entity relationship diagram, database, database schema, normalization and used types of normalization in the database.

3.2 Data flow diagram

A data-flow diagram is a way of representing a flow of data through a process or a system. The DFD also provides information about the outputs and inputs of each entity and the process itself. A data-flow diagram has no control flow, there are no decision rules. Data flow diagram shows how information enters and leaves the system, what changes the information and where information is stored. The purpose of a DFD is to show the scope and boundaries of a system as a whole. It is used as a communications tool between a systems analyst and any person who plays a part in the system that acts as the starting point for redesigning a system.

External Entity

An external entity can represent a human, system or subsystem. It is where certain data comes from or goes to. It is external to the system we study, in terms of the business process. For this reason, people used to draw external entities on the edge of a diagram.

Figure 3

Process

A process is a business activity or function where the manipulation and transformation of data take place. A process can be decomposed to a finer level of details, for representing how data is being processed within the process.

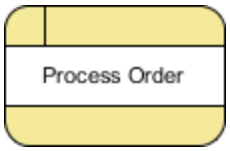

Figure 4

Data Flow

A data flow represents the flow of information, with its direction represented by an arrowhead that shows at the end(s) of flow connector.

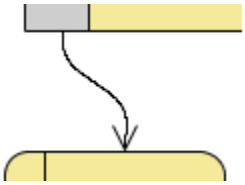

Figure 5

3.2.1 level-0

Level 0 DFD is also known as context diagrams. Context diagram are the most basic data flow diagrams. It provides a broad view that is easily digestible but offers little detail. Level 0 data flow diagrams show a single process node and its connections to external entities.

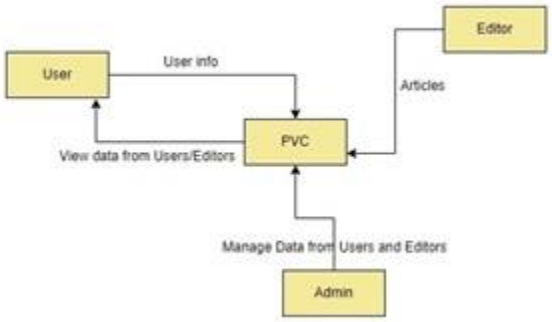

Figure 6

3.2.2 level-1

Level 1 DFD is still a general overview but it goes into more detail than a context diagram. In a level 1 data flow diagram the single process node from the context diagram is broken down into

subprocesses. As these processes are added, the diagram will need additional data flows and data stores to link them together.

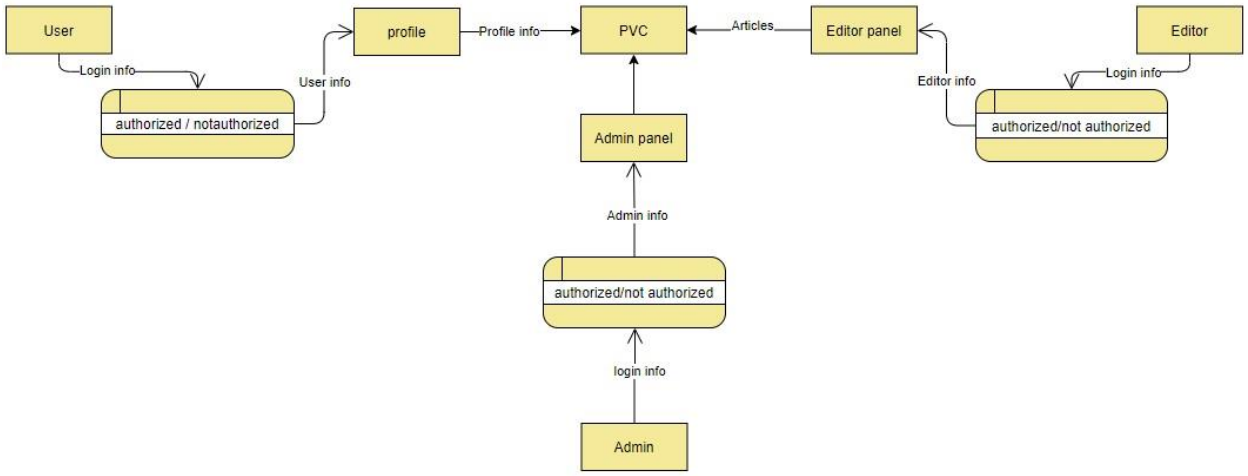

Figure 7

3.2.3 level-3

Level 2 DFD simply break processes down into more detailed subprocesses. It goes one process deeper into parts of 1-level DFD. It is be used to project or record the specific/necessary detail about the system's functioning.

Login

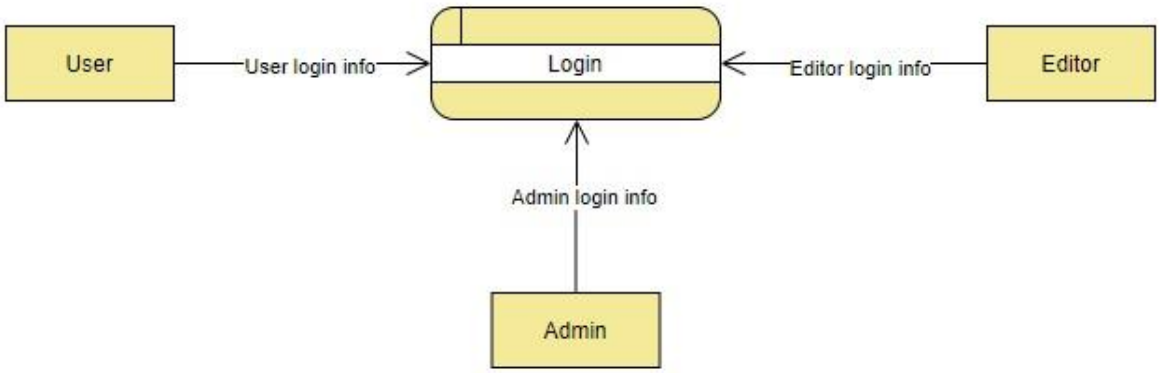

Figure 8

User login info to process

User Editor info to process

User Admin info to process

Articles

Figure 9

User data flow User info to article

Editor data flow article to article

Article commenting

Figure 10

User data flow comment and user info to comments
Editor data flow approved/unapproved to comments

User post

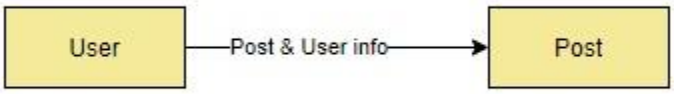

Figure 11

User data flow post and user info to post

Post comment

Figure 12

User data flow comment and user info to comment

User Messages

Figure 13

User data flow messages and user info to messages

3.3 Use case diagram

Use case diagrams are typically developed in the early stage of development and we often apply use case modeling for the following purposes:

- Specify the context of a system
- Capture the requirements of a system
- Validate a systems architecture
- Drive implementation and generate test cases
- Developed by analysts together with domain experts

Actor

- Someone interacts with use case (system function).
- Named by noun.
- Actor plays a role in the business
- Similar to the concept of user, but a user can play different roles
- For example:
    - A prof. can be instructor and also researcher
    - plays 2 roles with two systems
- Actor triggers use case(s).
- Actor has a responsibility toward the system (inputs), and Actor has expectations from the system (outputs).

Figure 14

Use Case

- System function (process - automated or manual)
- Named by verb + Noun (or Noun Phrase).
- i.e. Do something
- Each Actor must be linked to a use case, while some use cases may not be linked to actors.

Figure 15

Communication Link

- The participation of an actor in a use case is shown by connecting an actor to a use case by a solid link.
- Actors may be connected to use cases by associations, indicating that the actor and the use case communicate with one another using messages.

Boundary of system

- The system boundary is potentially the entire system as defined in the requirements document.
- For large and complex systems, each module may be the system boundary.
- For example, for an ERP system for an organization, each of the modules such as personnel, payroll, accounting, etc.
- can form a system boundary for use cases specific to each of these business functions.
- The entire system can span all of these modules depicting the overall system boundary

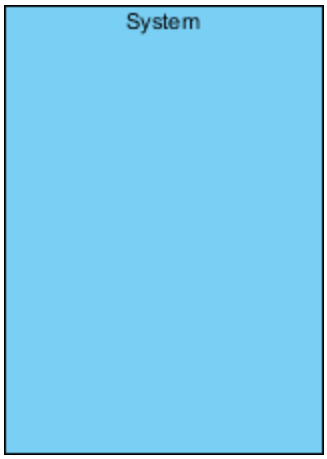

Figure 16

3.3.1 Admin

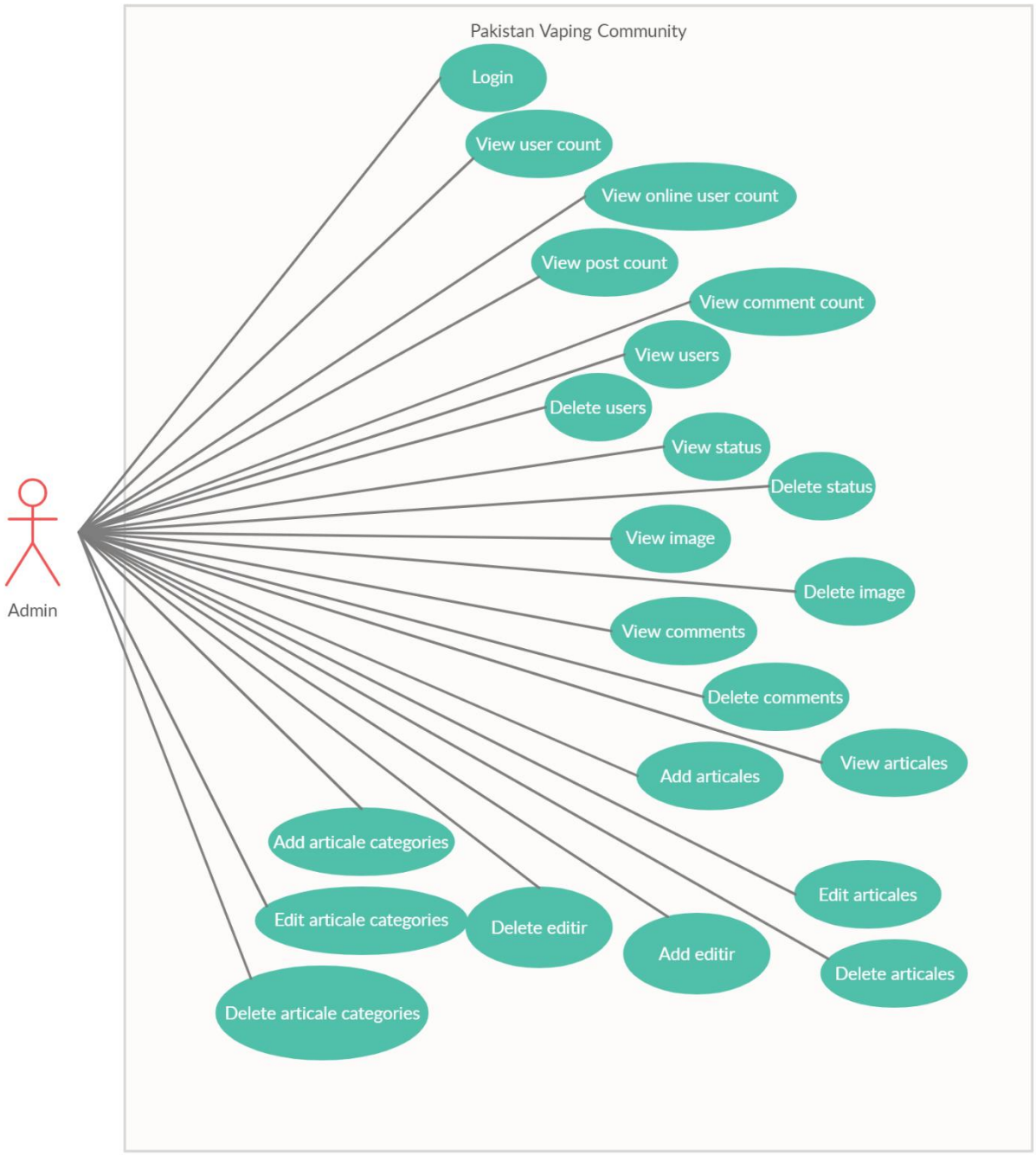

Figure 17

Admin use case

Login

In login use case admin will be able to provide login information for login process by providing email and password.

View user count

In view user count use case admin will be able to view all the member count signed up on the platform

View online user count

In view online user count use case admin will be able to view all the member count whom are signed up and currently online on the platform

View post count

In view post count use case admin will be able to view all the posts by the members on the platform

View comment count

In view comment count use case admin will be able to view all the member comment count which are commented by the members on the platform

View users

In view user use case admin will be able to view all the member signed up on the platform

Delete users

In delete user use case admin will be able to delete a member signed up on the platform

View status

In view status use case admin will be able to view all the status posted by the members on the platform.

Delete status

In delete status use case admin will be able to delete a status posted by a member on the platform.

View image

In view image use case admin will be able to view all the images posted by the member on the platform.

Delete image

In delete image use case admin will be able to delete the image posted by the member on the platform.

View comment

In view comment use case admin will be able to view all the comments posted by the members on the platform.

Delete comment

In delete comment use case admin will be able to delete a comment posted by the members on the platform.

View articles

In view article use case admin will be able to view all the articles posted by the editor on the platform.

Add articles

In add article use case admin will be able to add articles on the platform by providing post title, post author, post category, post content, post tags, post status and post image.

Edit articles

In edit article use case admin will be able to edit articles on the platform by editing post title, post author, post category, post content, post tags, post status and changing the post image.

Delete articles

In delete article use case admin will be able to delete an article by clicking the delete button.

Add article category

In add article category use case an admin will be able to add article category by providing article category and clicking add category button.

Edit article category

In edit article category use case, an admin will be able to edit article category by providing article category and clicking edit button.

Delete article category

In delete article category use case, an admin will be able to delete article category by clicking delete button.

Add editor

In add editor use case an admin will be able to add editor by providing username, email, password and role.

Delete editor

In delete editor use case, an admin will be able to delete an editor by clicking the delete button.

3.3.2 Editor

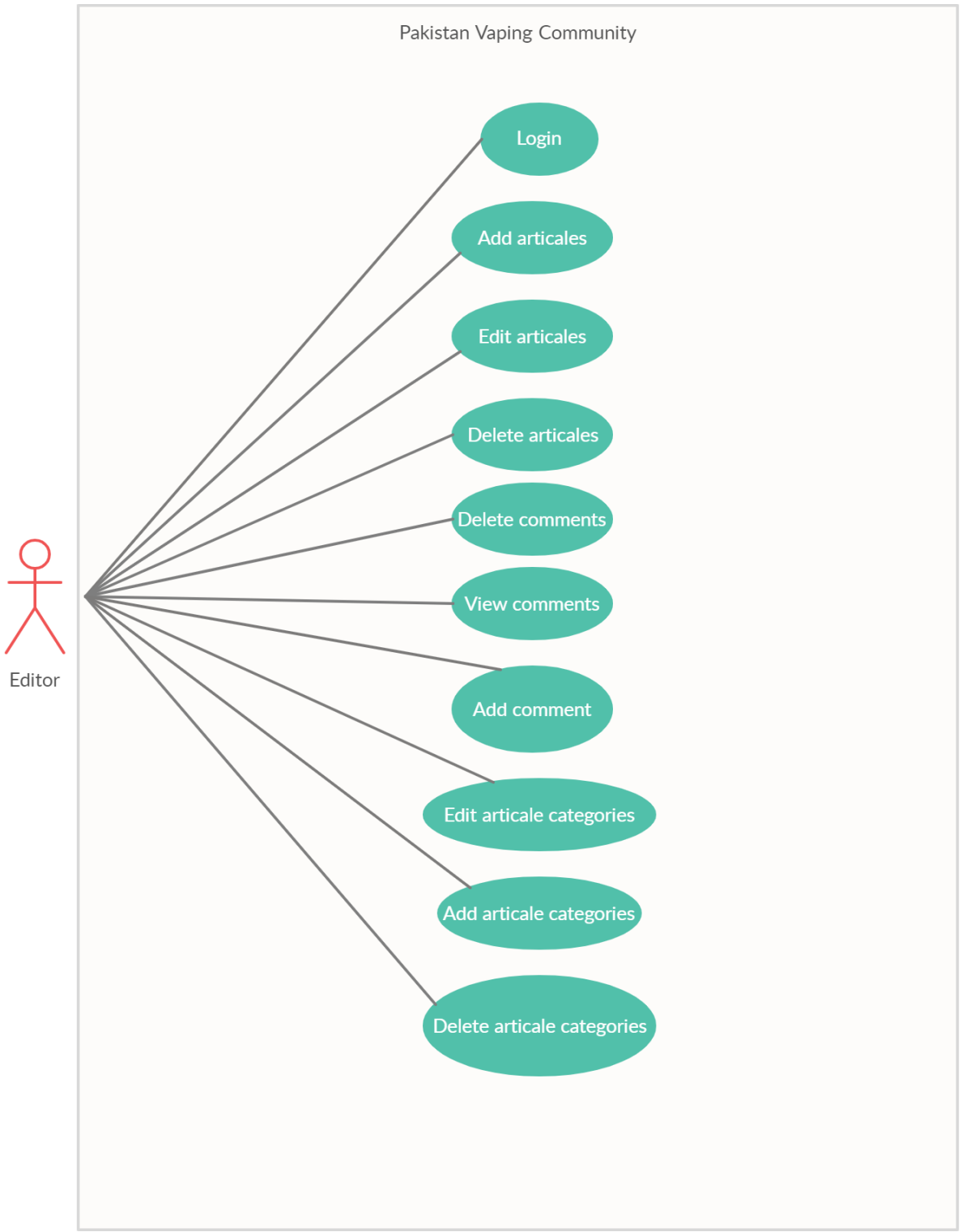

Figure 18

Editor use case

Login

In login use case editor will be able to provide login information for login process by providing email and password.

Add articles

In add article use case editor will be able to add articles on the platform by providing post title, post author, post category, post content, post tags, post status and post image.

Edit articles

In edit article use case editor will be able to edit articles on the platform by editing post title, post author, post category, post content, post tags, post status and post image.

Delete articles

In delete article use case editor will be able to delete an article by clicking the delete button.

Add comment

In add comment use case an editor will be able to add comment by written a text string in the text area and clicking the comment button.

View comment

In view comment use case, an editor will be able to view all the comment on the articles by members of the platform.

Delete comment

In delete comment use case, an editor will be able to delete comment by clicking the delete button.

Add article category

In add article category use case an editor will be able to add article category by providing article category and clicking add category button.

Edit article category

In edit article category use case, an editor will be able to edit article category by providing article category and clicking edit button.

Delete article category

In delete article category use case, an editor will be able to delete article category by clicking delete button.

3.3.3 User

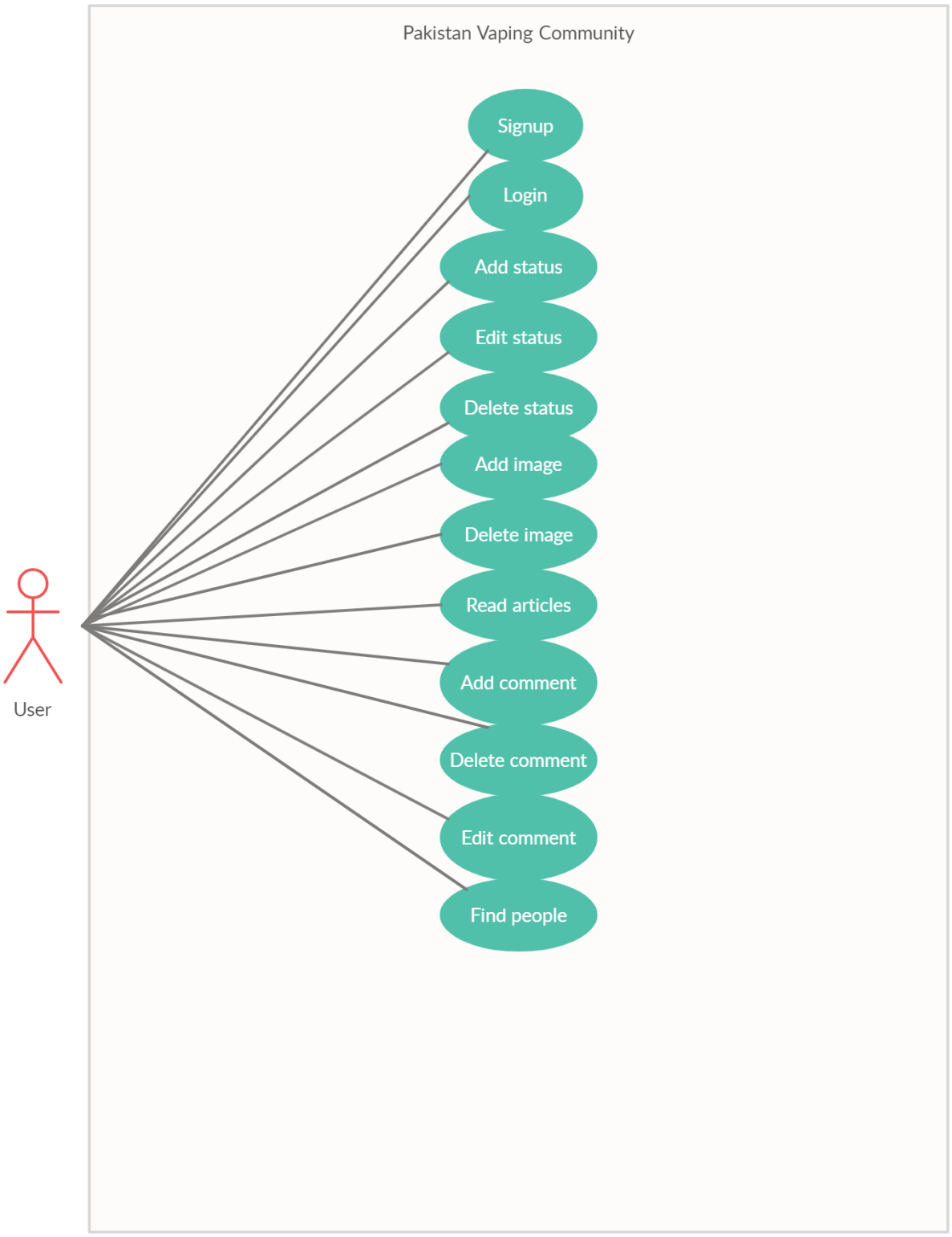

Figure 19

User use case

Signup

In signup use case user will be able to provide basic information for signup process which includes first name, last name, password, email, country name, gender and full date of birth.

Login

In login use case user will be able to provide login information for login process which include email and password.

Add status

In add status use case user will be able to add a status by written a text string in text area and selecting image file from select image button.

Edit status

In edit status use case user will be able to edit status which is written a text string in text area.

Delete status

In delete status use case user will be able to delete status by clicking the delete button.

Add image

In add image use case user will be able to add image by clicking the select image button.

Delete image

In delete image use case user will be able to delete image by clicking the delete button.

Read articles

In read article use case a user will be able to view all the articles on the platform.

Add comment

In add comment use case a user will be able to add comment to other user post by written a text string in the text area and clicking the comment button.

Delete comment

In delete comment use case, a user will be able to delete comment by clicking the delete button.

Edit comment

In edit comment use case, a user will be able to edit comment by clicking edit button.

Find people

In find people use case a user will be able to find other members on the platform by written text string in the search input field and clicking the search button.

3.4 Sequence diagram

Sequence Diagrams are interaction diagrams that show the detail how operations are carried out. They capture the interaction between objects in the context of a collaboration. Sequence Diagrams are time focus and they show the order of the interaction visually by using the vertical axis of the diagram to represent time what messages are sent and when.

Purpose of Sequence Diagram

- Model high-level interaction between active objects in a system
- Model the interaction between object instances within a collaboration that realizes a use case
- Model the interaction between objects within a collaboration that realizes an operation
- Either model generic interactions (showing all possible paths through the interaction) or specific instances of an interaction (showing just one path through the interaction)

Lifeline

- A lifeline represents an individual participant in the Interaction.

Figure 20

Activations

- A thin rectangle on a lifeline) represents the period during which an element is performing an operation.
- The top and the bottom of the of the rectangle are aligned with the initiation and the completion time respectively

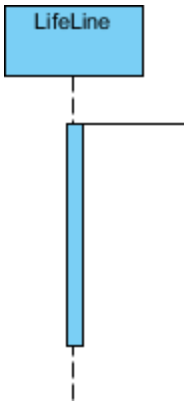

Figure 21

Call Message

- A message defines a particular communication between Lifelines of an Interaction.
- Call message is a kind of message that represents an invocation of operation of target lifeline.

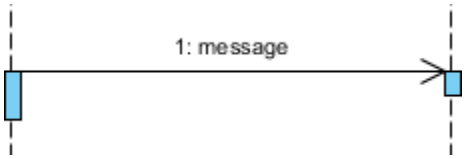

Figure 22

Return Message

- A message defines a particular communication between Lifelines of an Interaction.
- Return message is a kind of message that represents the pass of information back to the caller of a corresponded former message.

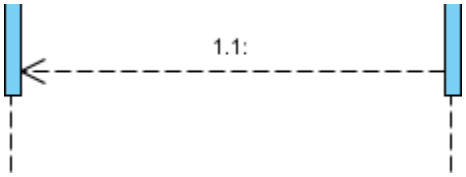

Figure 23

Sequence diagram

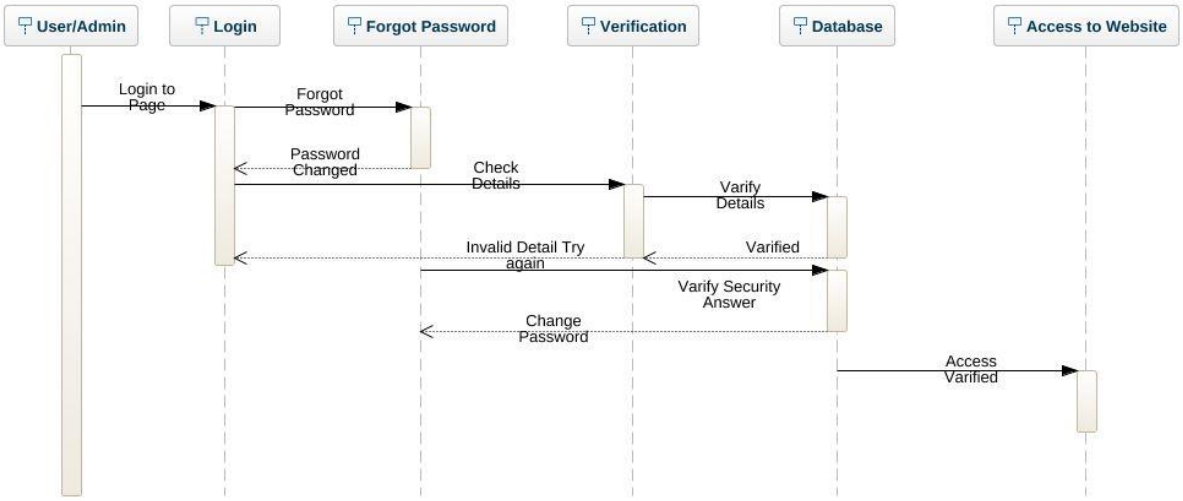

Figure 24

User/admin

User/admin call message to login

Login

Login call message to forget password

Forget password return message to login

Login call message to verification

Verification return message to login

Forget password

Forget password call message to database

Database return message to forget password

Verification

Verification call message to database

Database return message to verification

3.5 Class diagram

A class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects.

Purpose of Class Diagrams

1. Shows static structure of classifiers in a system
2. Diagram provides a basic notation for other structure diagrams prescribed by UML
3. Helpful for developers and other team members too
4. Business Analysts can use class diagrams to model systems from a business perspective

A UML class diagram is made up of:

- A set of classes and
- A set of relationships between classes

What is a Class

A description of a group of objects all with similar roles in the system, which consists of:

- Structural features (attributes) define what objects of the class "know"
    - Represent the state of an object of the class
    - Are descriptions of the structural or static features of a class
- Behavioral features (operations) define what objects of the class "can do"
    - Define the way in which objects may interact
    - Operations are descriptions of behavioral or dynamic features of a class

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 Operations** (Methods)
    - Operations are shown in the third partition. They are servicing 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

Class Relationships

A class may be involved in one or more relationships with other classes. A relationship can be one of the following types: (Refer to the figure on the right for the graphical representation of relationships).

Inheritance:

- Represents an "is-a" relationship.
- An abstract class name is shown in italics.
- SubClass1 and SubClass2 are specializations of Super Class.
- A solid line with a hollow arrowhead that point from the child to the parent class

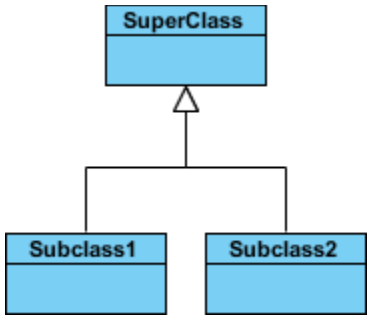

Figure 25

Simple Association:

- A structural link between two peer classes.
- There is an association between Class1 and Class2
- A solid line connecting two classes

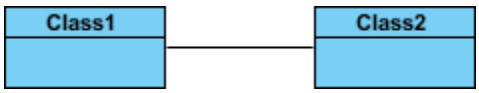

Figure 26

Dependency:

- Exists between two classes if the changes to the definition of one may cause changes to the other (but not the other way around).
- Class1 depends on Class2
- A dashed line with an open arrow

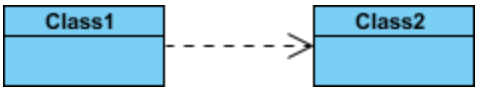

Figure 27

Class diagram

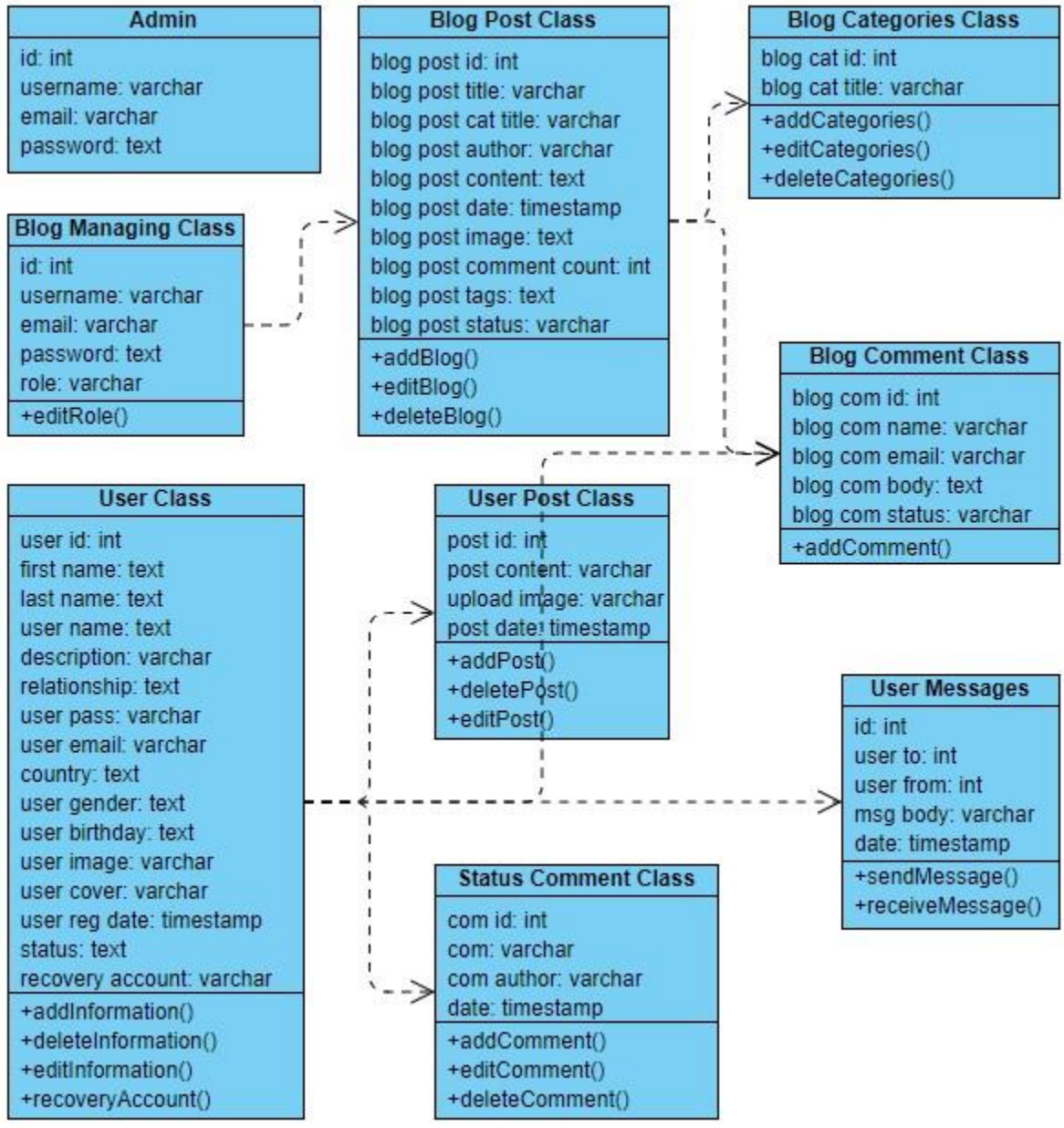

Figure 28

Class name

Admin

Class Attributes

1. id
2. Username
3. Email
4. Password
5. Role

Class Operations

Delete user ()

Delete comment ()

Delete post ()

Delete article ()

Class Name

Blog post

Class Attributes

1. Blog_post_id
2. Blog_post_title
3. Blog_post_cat_title
4. Blog_post_author
5. Blog_post_content
6. Blog_post_date
7. Blog_post_image
8. Blog_post_comment_count
9. Blog_post_tags
10. Blog_post_status
11. Blog_user_id
12. Blog_cat_id

Class Operations

Add blog ()

Edit blog ()

Delete blog ()

Class Name

Blog comment

Class Attributes

1. Blog_com_id
2. Blog_comment_name
3. Blog_comment_email
4. Blog_comment_body
5. Blog_comment_status
6. Blog_post_id

Class Operations

Add comment ()

Class Name

Blog categories

Class Attributes

1. Blog_cat_id
2. Blog_cat_tittle

Class Operations

Add category ()

Delete category ()

Edit category ()

Class Name

User

Class Attributes

1. user_id
2. f_name
3. l_name
4. user_name
5. description
6. relationship
7. user_pass
8. user_email
9. user_country
10. user_gender
11. user_birthday
12. user_image
13. user_cover
14. user_reg_date
15. status
16. posts
17. user_recovery_acc

Class Operations

Edit profile ()

Class Name

User messages

Class Attributes

1. id
2. user_id
3. user_to
4. msg_body
5. date

Class Operations

Send message ()

Class Name

Posts

Class Attributes

1. Post_id
2. Post_content
3. Upload_image
4. Post_date
5. User_id

Class Operations

Add post ()

Edit post ()

Delete post ()

Class Name

Comments

Class Attributes

1. Com_id
2. Comment
3. Comment_author
4. Date
5. Post_id
6. User_id

Class Operations

Add comment ()

Edit comment ()

Delete comment ()

3.6 ERD (Entity relationship Diagram)

Entity Relationship Diagram, also known as ERD, ER Diagram or ER model, is a type of structural diagram for use in database design. An ERD contains different symbols and connectors that visualize two important information: The major entities within the system scope, and the inter-relationships among these entities

Entity

An ERD entity is a definable thing or concept within a system, such as a person/role (e.g. Student), object (e.g. Invoice), concept (e.g. Profile) or event (e.g. Transaction) (note: In ERD, the term "entity" is often used instead of "table", but they are the same). When determining entities, think of them as nouns. In ER models, an entity is shown as a rounded rectangle, with its name on top and its attributes listed in the body of the entity shape. The ERD example below shows an example of an ER entity.

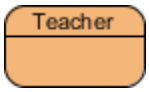

Figure 29

Entity Attributes

Also known as a column, an attribute is a property or characteristic of the entity that holds it. An attribute has a name that describes the property and a type that describes the kind of attribute it is, such as varchar for a string, and int for integer. When an ERD is drawn for physical database development, it is important to ensure the use of types that are supported by the target RDBMS.

The ER diagram example below shows an entity with some attributes in it.

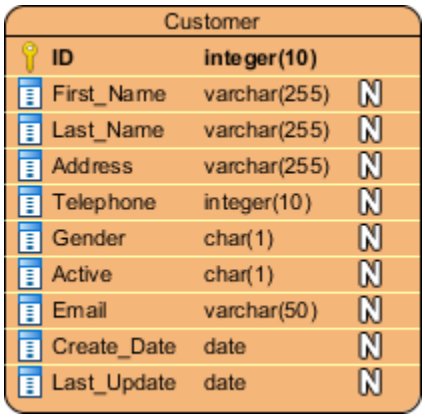

Figure 30

Primary Key

Also known as PK, a primary key is a special kind of entity attribute that uniquely defines a record in a database table. In other words, there must not be two (or more) records that share the same value for the primary key attribute. The ERD example below shows an entity 'Product' with a primary key attribute 'ID', and a preview of table records in the database. The third record is invalid because the value of ID 'PDT-0002' is already used by another record.

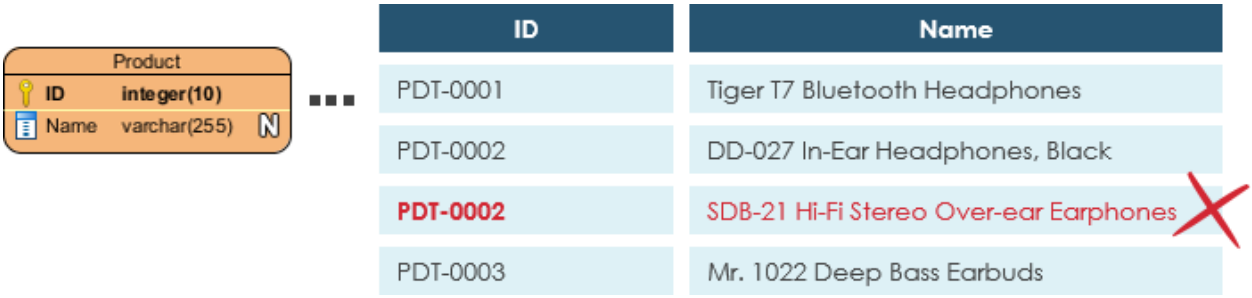

Figure 31

Foreign Key

Also known as FK, a foreign key is a reference to a primary key in a table. It is used to identify the relationships between entities. Note that foreign keys need not be unique. Multiple records can share the same values. The ER Diagram example below shows an entity with some columns, among which a foreign key is used in referencing another entity.

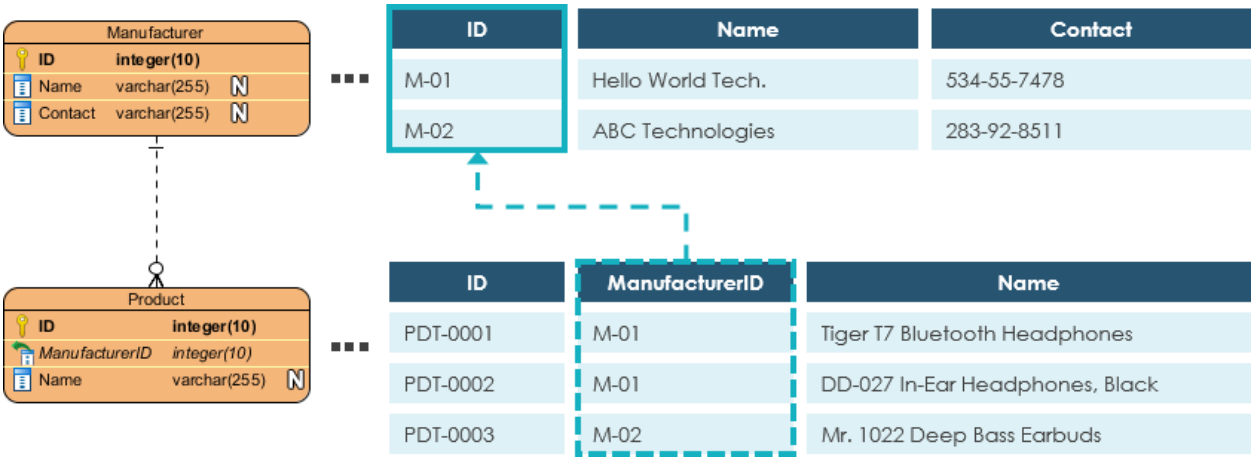

Figure 32

Relationship

A relationship between two entities signifies that the two entities are associated with each other somehow. For example, a student might enroll in a course. The entity Student is therefore related to Course, and a relationship is presented as a connector connecting between them.

Cardinality

Cardinality defines the possible number of occurrences in one entity which is associated with the number of occurrences in another. For example, ONE team has MANY players. When present in an ERD, the entity Team and Player are inter-connected with a one-to-many relationship.
In an ER diagram, cardinality is represented as a crow's foot at the connector's ends. The three common cardinal relationships are one-to-one, one-to-many, and many-to-many.

One-to-One cardinality example

A one-to-one relationship is mostly used to split an entity in two to provide information concisely and make it more understandable. The figure below shows an example of a one-to-one relationship.

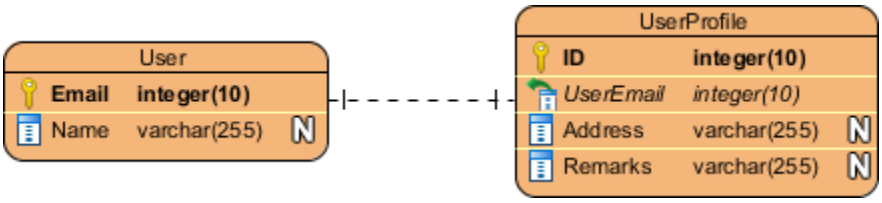

Figure 33

One-to-Many cardinality example

A one-to-many relationship refers to the relationship between two entities X and Y in which an instance of X may be linked to many instances of Y, but an instance of Y is linked to only one instance of X. The figure below shows an example of a one-to-many relationship.

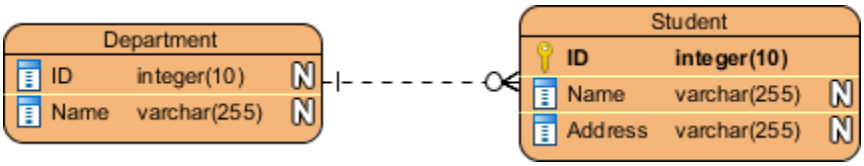

Figure 34

Many-to-Many cardinality example

A many-to-many relationship refers to the relationship between two entities X and Y in which X may be linked to many instances of Y and vice versa. The figure below shows an example of a many-to-many relationship. Note that a many-to-many relationship is split into a pair of one-to-many relationships in a physical ERD. You will know what a physical ERD is in the next section.

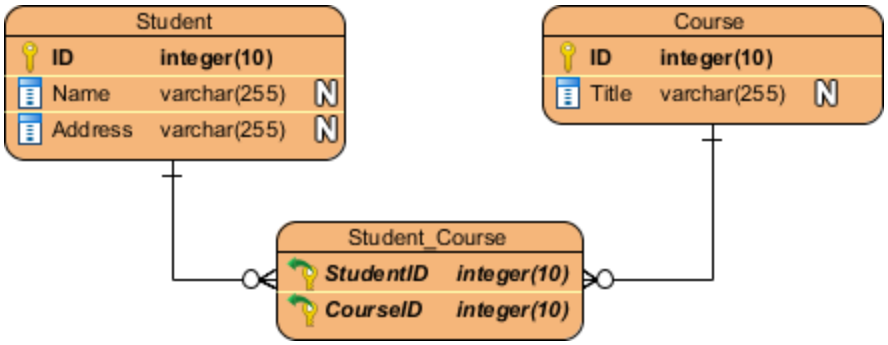
Student
ID integer(10)
Name varchar(255) N
Address varchar(255) N
Course
ID integer(10)
Title varchar(255) N
Student_Course
StudentID integer(10)
CourseID integer(10)

Figure 35

Entity relationship Diagram

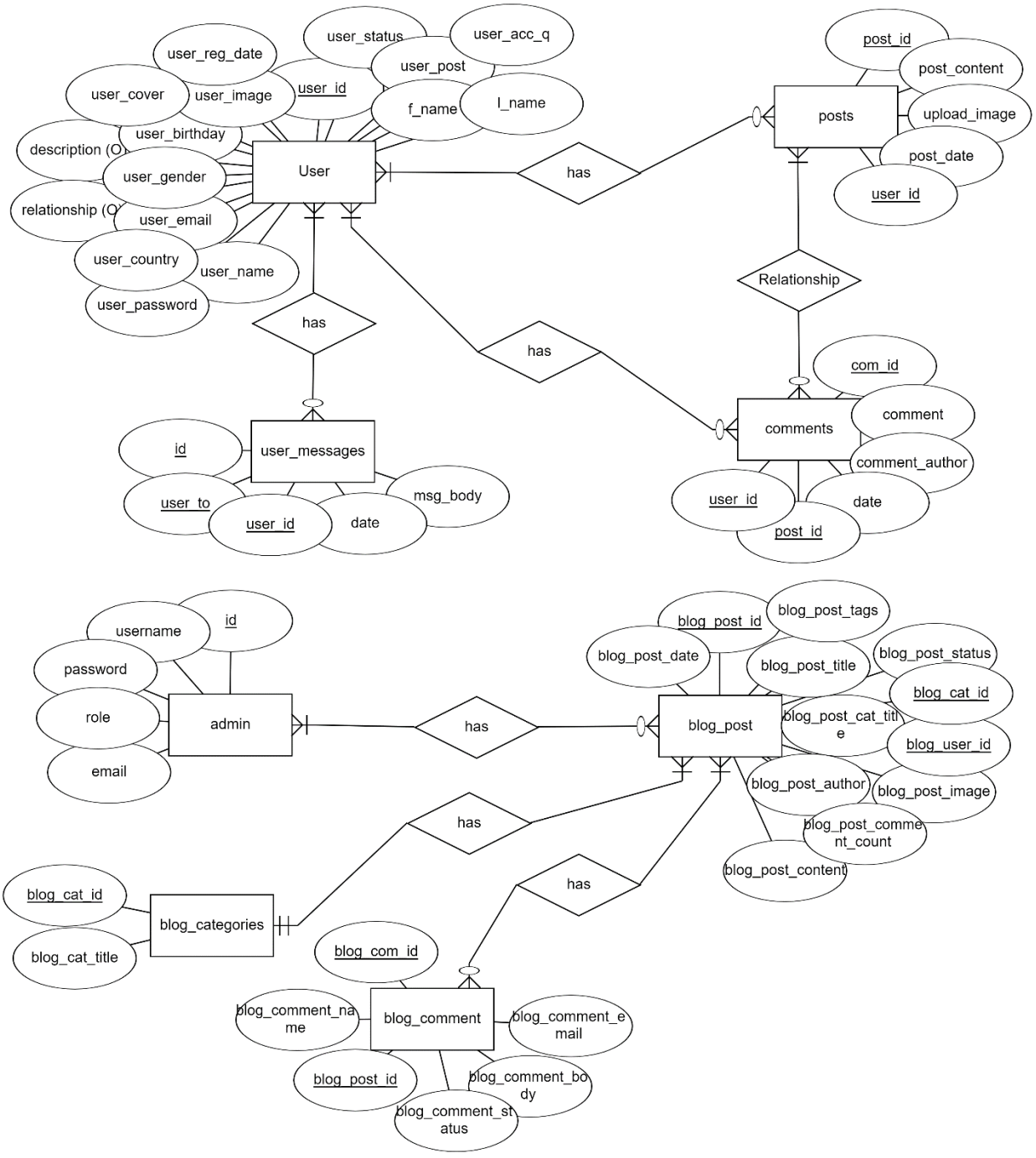

Figure 36

Users to message

User mandatory many messages optional many

Messages to user

Messages mandatory many User mandatory many

User to comment

User mandatory many user comments optional many

Comments to user

Comments mandatory many user mandatory many

User to post

User mandatory many to user post optional many

Post to users

Post mandatory many user mandatory many

Admin to blog post

Admin mandatory many blogs post optional many

Blog post to admin

Blogs post mandatory many admin mandatory many

Admin to blog categories

Admin mandatory many bog categories optional many

Blog categories to admin

Bog categories mandatory many admin mandatory many

Admin to blog comment

Admin mandatory many blog comments optional many

Blog comment to admin

Blog comments mandatory many admin mandatory many

3.7 Database

Entities

- Admin
- Blog_post
- Blog_comment
- Blog_categories
- Users
- User_messages
- Posts

- Comments

Attributes of entities

- **Admin**
    1. id
    2. Username
    3. Email
    4. Password
    5. Role
- **Blog_post**
    1. Blog_post_id
    2. Blog_post_title
    3. Blog_post_cat_title
    4. Blog_post_author
    5. Blog_post_content
    6. Blog_post_date
    7. Blog_post_image
    8. Blog_post_comment_count
    9. Blog_post_tags
    10. Blog_post_status
    11. Blog_user_id
    12. Blog_cat_id
- **Blog_comment**
    1. Blog_com_id
    2. Blog_comment_name
    3. Blog_comment_email
    4. Blog_comment_body
    5. Blog_comment_status
    6. Blog_post_id
- **Blog_categories**
    1. Blog_cat_id

2. Blog_cat_tittle

- **User**
    1. user_id
    2. f_name
    3. l_name
    4. user_name
    5. description
    6. relationship
    7. user_pass
    8. user_email
    9. user_country
    10. user_gender
    11. user_birthday
    12. user_image
    13. user_cover
    14. user_reg_date
    15. status
    16. posts
    17. user_recovery_acc
- **user_messages**
    1. id
    2. user_id
    3. user_to
    4. msg_body
    5. date
- **Posts**
    1. Post_id
    2. Post_content
    3. Upload_image
    4. Post_date
    5. User_id

- **Comments**
    1. Com_id
    2. Comment
    3. Comment_author
    4. Date
    5. Post_id
    6. User_id

3.7.1 Database schema

A database schema is the skeleton structure that represents the logical view of the entire database. It defines how the data is organized and how the relations among them are associated. It formulates all the constraints that are to be applied on the data.
A database schema defines its entities and the relationship among them. It contains a descriptive detail of the database, which can be depicted by means of schema diagrams. It's the database designers who design the schema to help programmers understand the database and make it useful.

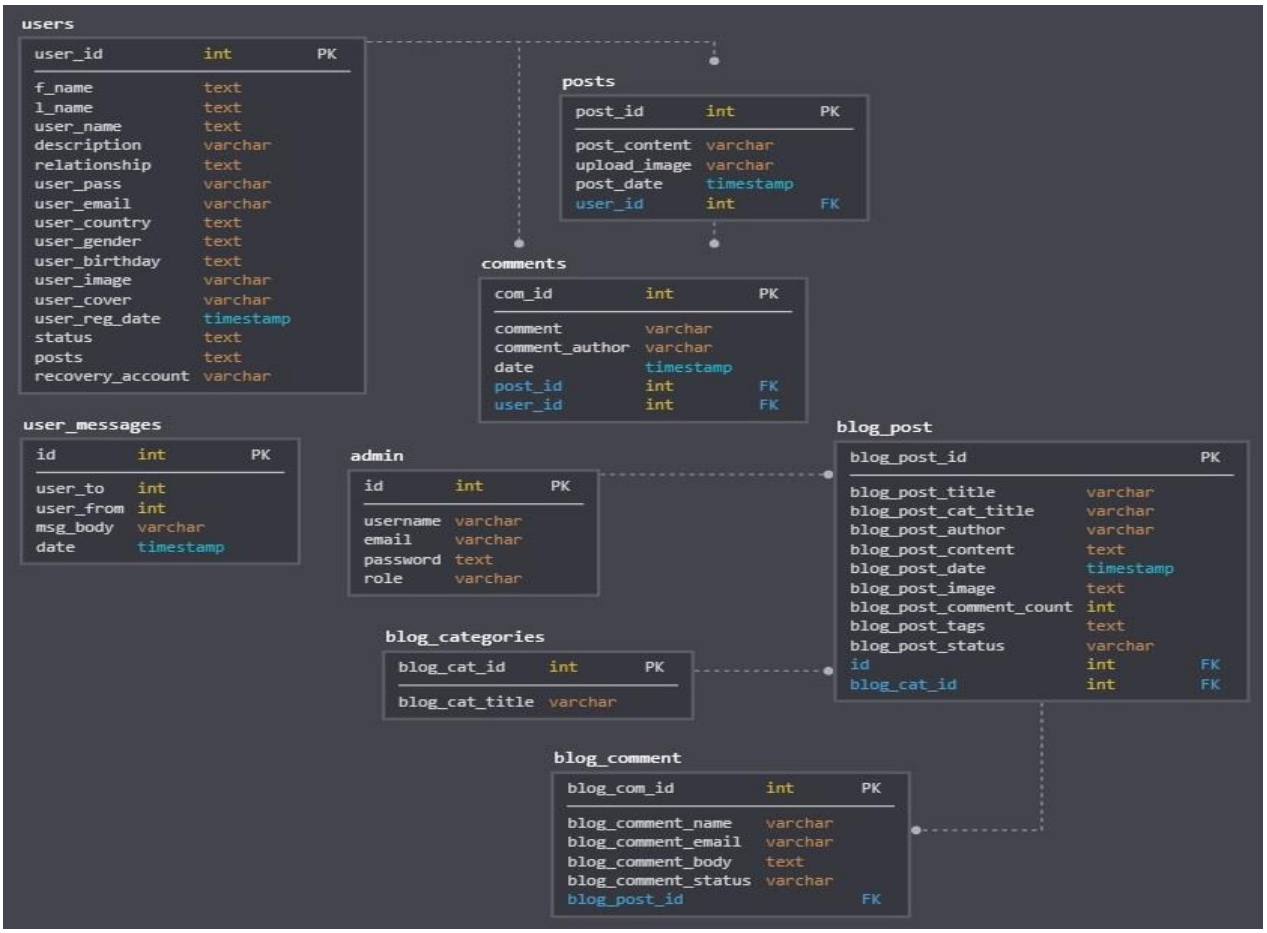

Figure 37

User table

User table contain all the user information and user table primary key User id is used as foreign key in Post table and comment table.

Post table

Post table contain all the user post information and post table primary key is used as foreign key in comments table.

Comments table

Comment table contain all the user, post and comment information. Comment table has two foreign key post id from post table and user id from user table.

User messages table

User message table contain all the user and message information. User message has one foreign key user to from user table.

Admin table

Admin table contain al the information for the admin.

Blog post table

Blog post table contain all the information for the blog post. Blog post table has two foreign key admin id from admin table and category id from blog category table.

Blog categories table

Blog category table contain all the information for the category of blog post.

Blog comment table

Blog comment table contain all the information of comment and a foreign key from blog post table.

3.8 Normalization

Purpose of normalization is, if a table is not properly normalized and have data redundancy then it will not only eat up extra memory space but will also make it difficult to handle and updating the database without facing data loss. Insertion, updating and deletion anomalies are very frequent, if database is not normalized.

3.8.1 Types of normalization

First Normal Form (1NF)

For a table to be in the First Normal Form, it should follow the following 4 rules:

1. It should only have single(atomic) valued attributes/columns.
2. Values stored in a column should be of the same domain
3. All the columns in a table should have unique names.
4. And the order in which data is stored, does not matter.

Table 1

Table 2

User name	User role	Blog title	Blog category	Blog author	Blog content	Blog image
Fawad	Admin	Laptop	Technology	Fawad	New laptop	../images/519sdf6d876b.jpg
Jawad	Editor	Mobile	Technology	Hammad	New mobile	../images/51fsfsdb6a4i4.jpg
Fawad	Admin	Tab	Technology	Saad	New tab	../images/5sdfsdfs5c1c4.jpg
Jawad	Editor	computer	Technology	Jawad	New computer	../images/5196dsf601c4.jpg

Second Normal Form (2NF)

For a table to be in the Second Normal Form,

1. It should be in the First Normal form.
2. And, it should not have Partial Dependency.

Table 1

Table 3

User name	User role	Blog title	Blog category	Blog author	Blog content	Blog image
Fawad	Admin	Laptop	1	Fawad	New laptop	../images/519sdf6d876b.jpg
Jawad	Editor	Mobile	1	Hammad	New mobile	../images/51fsfsdb6a4i4.jpg
Fawad	Admin	Tab	1	Saad	New tab	../images/5sdfsdfs5c1c4.jpg
Jawad	Editor	computer	1	Jawad	New computer	../images/5196dsf601c4.jpg

Table 2

Table 4

Category id	Category
1	Technology

Third Normal Form (3NF)

A table is said to be in the Third Normal Form when,

1. It is in the Second Normal form.

2. And, it doesn't have Transitive Dependency.

Table 1

Table 5

User name	User role	Blog title	Blog category	Blog author	Blog content	Blog image
2	2	Laptop	1	Fawad	New laptop	../images/519sdf6d876b.jpg
1	1	Mobile	1	Hammad	New mobile	../images/51fsfsdb6a4i4.jpg
2	2	Tab	1	Saad	New tab	../images/5sdfsdfs5c1c4.jpg
1	1	computer	1	Jawad	New computer	../images/5196dsf601c4.jpg

Table 2

Table 6

Category id	Category
1	Technology

Table 3

Table 7

User id	User name
1	Jawad
2	Fawad

Table 4

Table 8

Role id	Role
1	editor
2	admin

Chapter 4 Development

4.1 Development plan

Requirements Gathering (Week 1)

In this week all the requirements are collected for the project development. All the requirements for social platform are collected by searching and interviewing the users. Collected requirements includes where a user should be able to maintain user profile information, a user should be able to share thoughts by sharing a photo and by writing a status. A user should be able to socialize with other users on the platform by commenting on other user status and should be able to contact other users by privately messaging. A user should be able to read articles and should be able to comment on those articles which are written by the platform.

Developing specifications (Week 2)

In this week all the requirements are analyzed carefully for which specification are needed which includes a profile module for user to maintain user information, a module status for socialization between users and a comment module so users will be able to comment on other user status. An article module where users can read articles and a module for commenting on articles.

Front-end design (Week 3)

In this week all the front-end design is finalized. The user interface is designed by using mock flow web-based app. Designed interface includes user login interface, signup interface, user news feed interface, profile interface, searched interface, article interface and messaging interface.

Back-end design (Week 4)

In this week all the back-end designs are designed by analysis the requirements. Firstly, Data flow diagram is designed. After designing the data flow diagram, the use case diagram is designed accordingly to use cases for users. Sequence diagram is built after designing sequence diagram the class diagram is designed. Entity relationship diagram is designed for database. All the entities and attributes are designed in database schema and normalized the database.

Coding (Week 5)

In this week front-end designs and back-end designs are coded according to the front-end design and back end-design using front-end technologies and back-end technologies. All the modules are coded which we gathered from requirement phase.

Unit testing (Week 6)

In this week all the modules are tested which are included in the project, profile module for user to maintain user information, status module for socialization between users and commenting module for users to comment on other user status. An article module where users read articles and module for commenting on articles.

System testing (Week 7)

In this week the platform is tested. All the modules are working accordingly to the requirements of the project and there is no occurring of errors while running the platform. In acceptance testing all the requirements for the project are accessed and all the modules fulfil the needed requirements.

4.2 Platform

Landing Page

Figure 38

On landing page of the platform user will have two options

- **Signup**
- **Login**

Signup Page

On signup page there are two options for the user signup or already have an account?

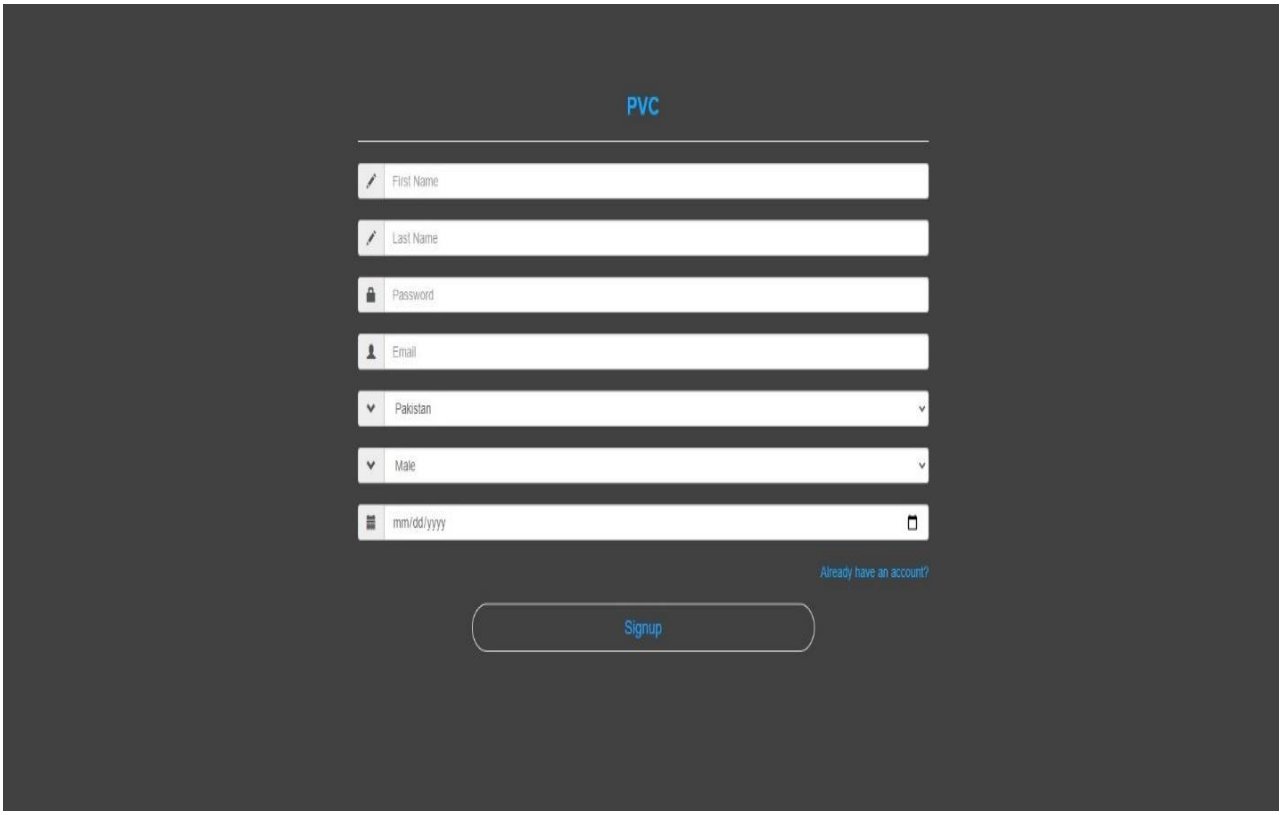

Figure 39

- **Signup**

Signup page require basic information of the user to sign up on the platform.

First Name

User have to provide first name in this input field

Last Name

User have to provide last name in this input field

Password

User have to set a password in this input field

Email

User have to provide email address in this input field

Country

User have to provide country name in this input field

Gender

User have to provide gender in this input field

Birthday

User have to provide full date of birth as DD/MM/YYYY in this input field

- **Already have an account?**

On already have account? If a user is already signup on the platform then user can use this option and User will be requiring to input the email and password which is used in the signup process.

Login Page

On login page a user will have two options login or don't have an account?

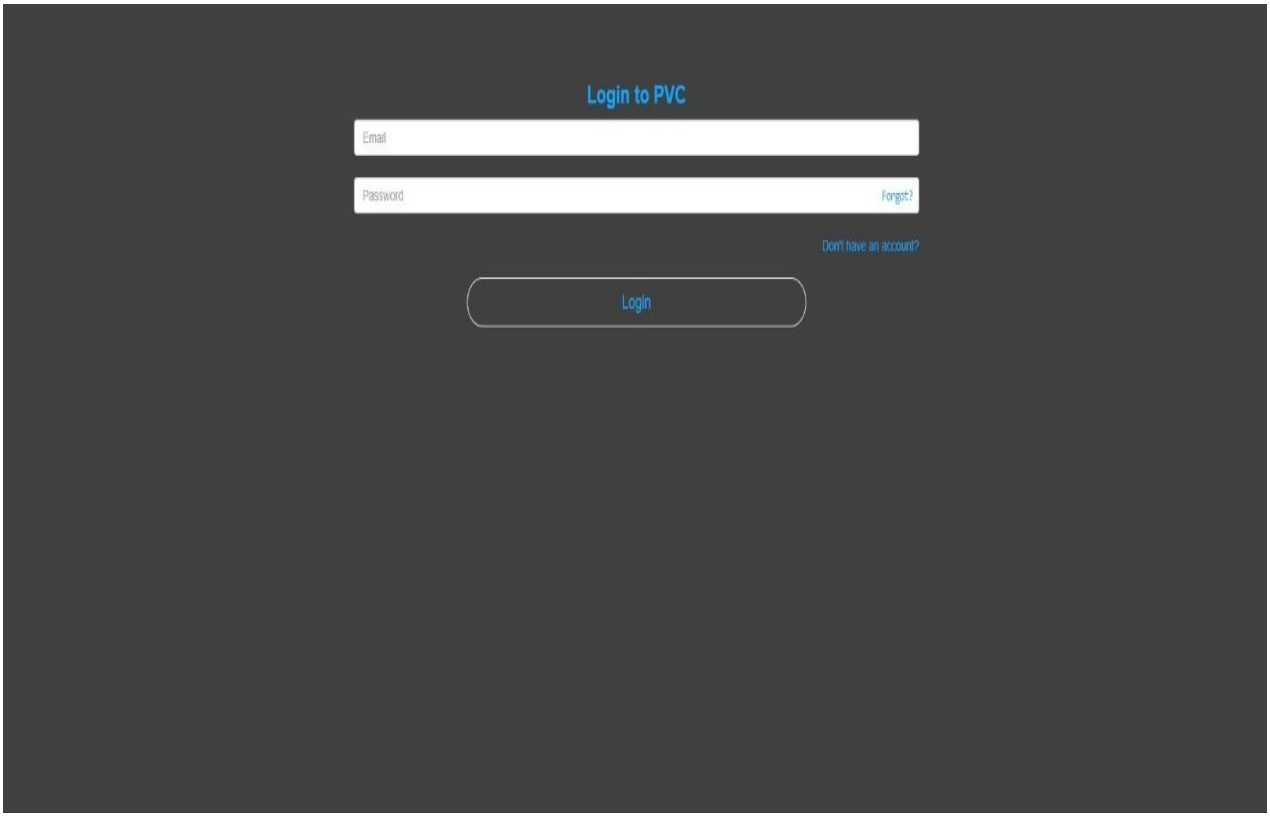

Figure 40

- **Login**

Login page will require email and password which is used in the signup process.

- **Don't have an account?**

Don't have an account will give the user an option to signup by providing basic information.

First Name

Last Name

Password

Email

Country

Gender

Birthday

News Feed Page

A user after logging in will have the option of posting a status and commenting on others post.

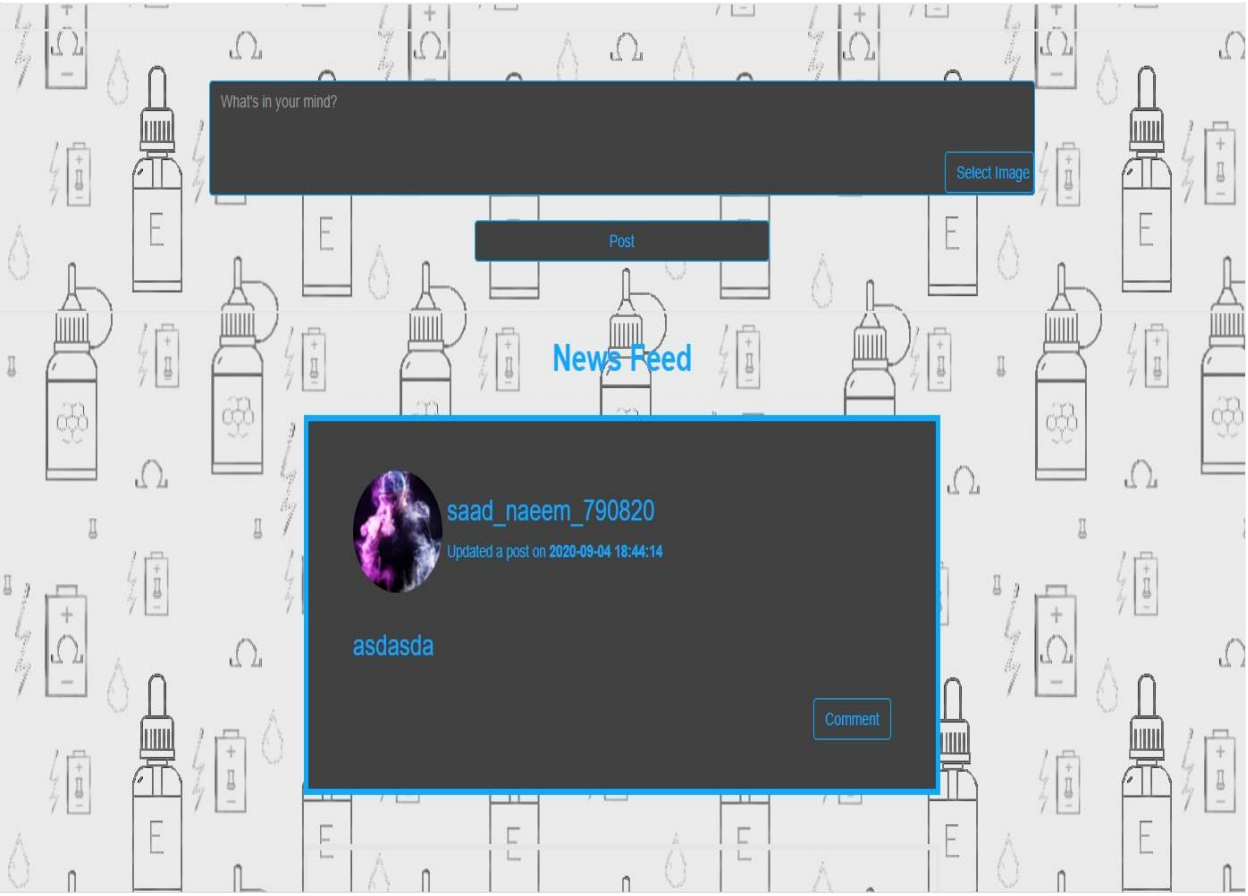

Figure 41

- **Status**

A user a post text status or a text status with an image, a user has to select the image from SELECT IMAGE button from the gallery. After written a status and selecting the image file the user has to click the POST button to upload the status to news feed.

- **Commenting**

A user comment on a post by written a comment on the post and by clicking the COMMENT button the comment will be posted on the user post.

Find New People Page

On find new people page a user can find other users on the platform by searching.

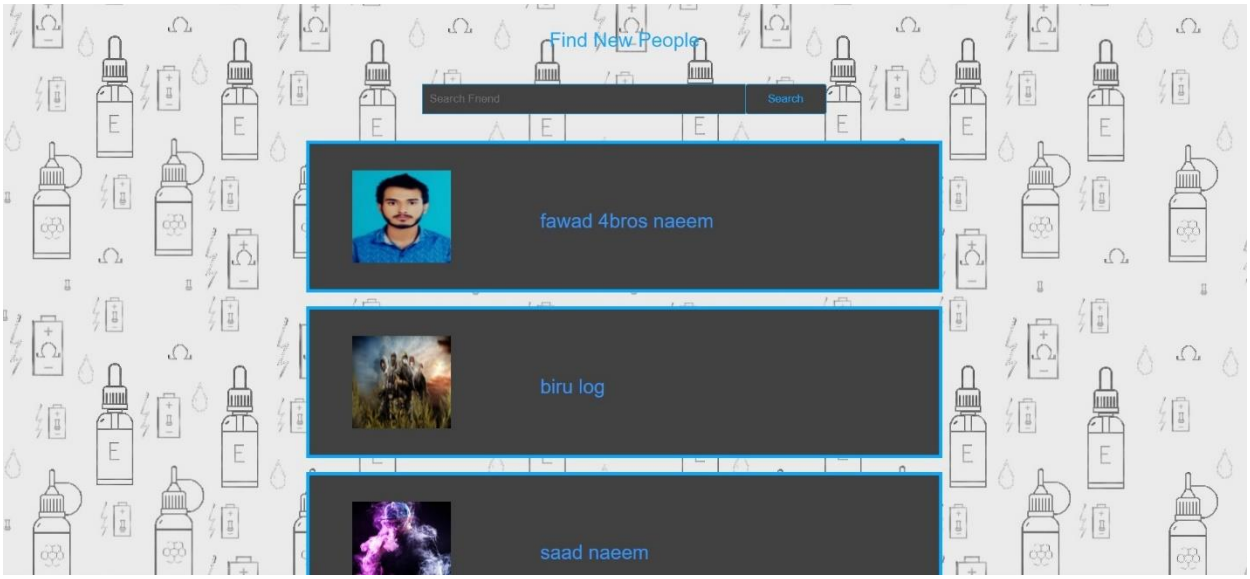

Figure 42

- **Search**

A user has to write the other user first name or last name in the search text area and by clicking the SEARCH button all the users which have the keywords in their name will appear.

Edit Profile Page

On edit profile page a user will have the option to edit the profile information.

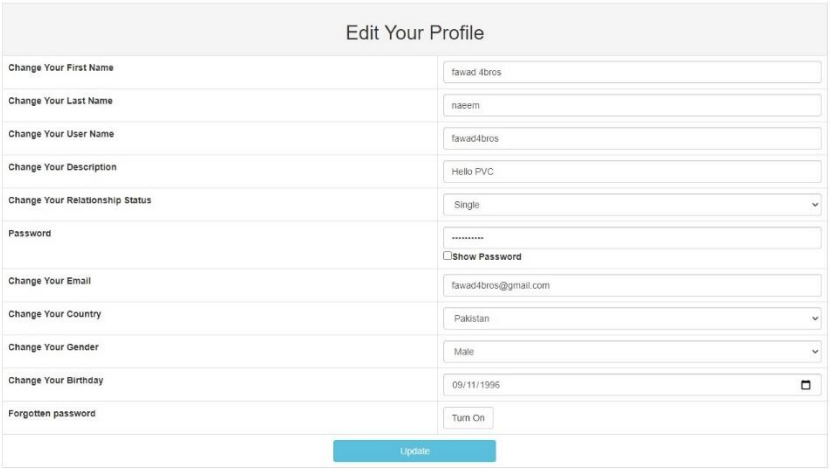

Figure 43

Change first name

User can changer first name by this option

Change last name

User can changer Change last name by this option

Change description

User can changer Change description by this option

Change password

User can changer Change password by this option

Change country

User can changer Change gender by this option

Change gender

User can changer Change gender by this option

Change birthday

User can changer Change birthday by this option

Article Page

On article page user will read about new articles published by the admin and a user will be able to select a category for article reading.

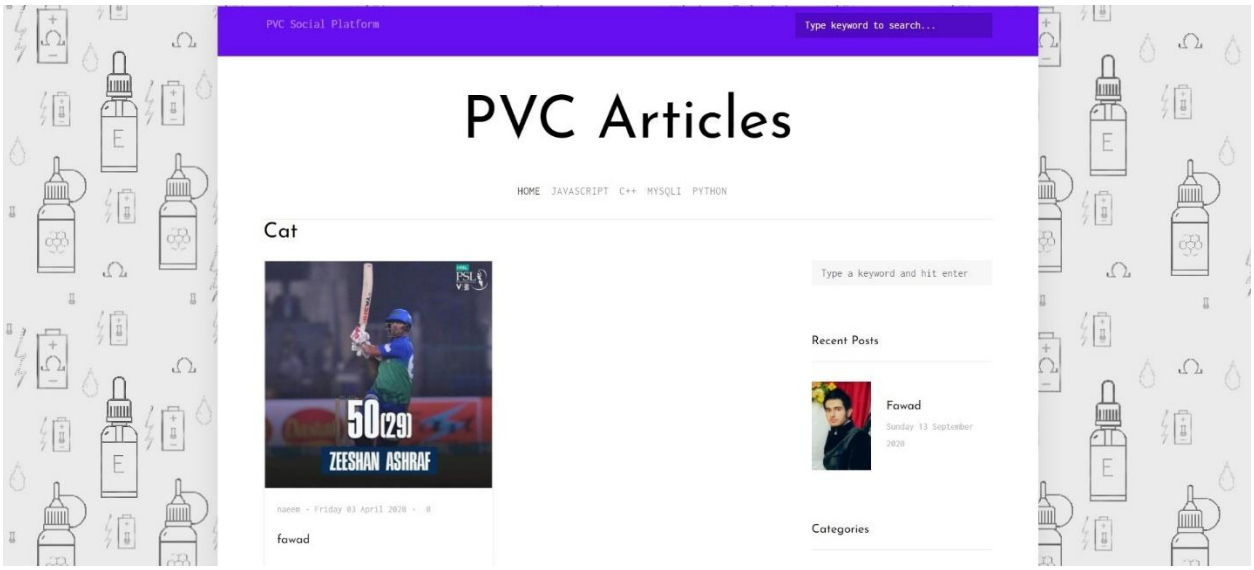

Figure 44

Category

A user will be able to select a category by clicking on the category name.

Article Comment Section

A user will be able to leave a comment under the article by providing the username and email.

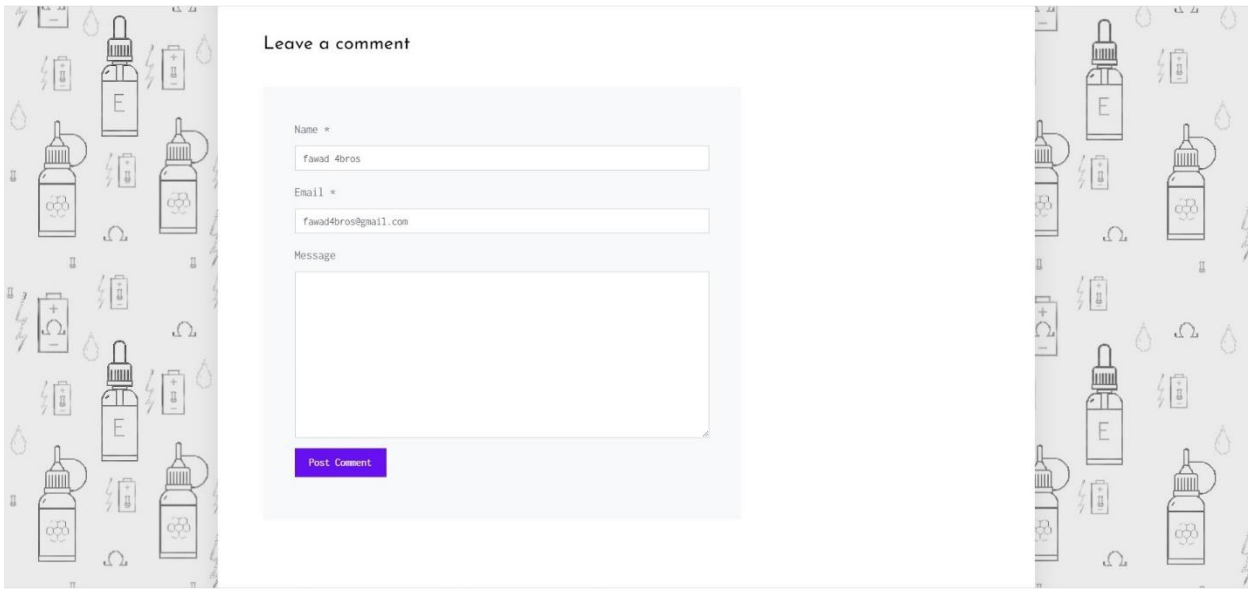

Figure 45

Name

A user which is already signup on the platform will not have to provide the name because the platform will auto pick the user name and user which are not sign up they have to provide the user name in the input field.

Email

A user which is already signup on the platform will not have to provide the email because the platform will auto pick the user email and user which are not sign up, they have to provide the user email in the input field.

Comment

A user will be able to leave a comment in this input field by text, after written a comment user have to click the COMMENT button to post the comment on the article.

4.3 Language

HTML

CSS

BOOTSTRAP

PHP

JAVASCRIPT

MYSQL

Chapter 5 Deployment

In this chapter testing of the system takes place. Black box testing, white box testing, unit box testing, acceptance testing and test cases are analyzed.

5.2 Testing of the system

In system testing the platform is tested by various testing techniques which includes black box testing, white box testing, unit box testing and acceptance testing. All these techniques have some advantages and disadvantages.

5.2.1 Black box

In black box testing we have tested our platform where we kept these points in our mind while testing the platform.

Incorrect or missing functions

Interface errors

Errors in data structures or external database access

Behavior or performance errors

Initialization and termination errors

Tests are done from a user's point of view and it help in exposing discrepancies in the specifications.

5.2.2 White box

In white box testing we have done our testing based on an analysis of the internal structure of the system. We have studied the implementation code of fields on a webpage and determines all valid and invalid inputs and verifies the outputs against the expected outcomes. This test is determined by studying the implementation code.

5.2.3 Unit box

In unit box testing the main purpose was to validate that each unit of our platform performs as designed or not. We have test individual units, function and procedures of our platform

5.2.4 Acceptance testing

In acceptance testing we have tested our platform on a level of testing where platform is tested for acceptability. The purpose of this test was to evaluate the platform compliance with the user requirements and assess whether it is acceptable for delivery or not. Formal testing is done with respect to user needs, requirements, and business processes conducted to determine whether or not this platform satisfies the acceptance criteria.

5.3 Test case

A test case is a document, which has a set of test data, preconditions, expected results and postconditions, developed for a particular test scenario in order to verify compliance against a specific requirement.

Test Case acts as the starting point for the test execution, and after applying a set of input values, the application has a definitive outcome and leaves the system at some end point or also known as execution postcondition.

Test case title: Member login

Table 9

Test case 1: Login	**Priority (H, L):** High
Test objective: verifying login	
Test descriptions: "user enter the required field, press login button", client program contact to the server, server contact with database, database update the result and send the results to the user.	
Requirements verified: Yes	
Test environment: IIS is in running state, Database contain appropriate table and link is established between server and client program.	
Test setup/ pre condition: IIS in running state. The entire mandatory field is entered.	
Location	**Expected result**

The user login to aces platform	"Login successfully". Display main page.
Pass: Yes	**Condition**: Pass **Fail:** No
Problem /issues: NIL	
status: successfully executed	

Test case title: Member Signup

Table 10

Test case 2: Signup	**Priority (H, L):** High
Test objective: verifying login	
Test descriptions: "user enter the required field, press signup button", client program contact to the server, server contact with database, database update the result and send the results to the user.	
Requirements verified: Yes	
Test environment: IIS is in running state, Database contain appropriate table and link is established between server and client program.	
Test setup/ pre condition: IIS in running state. The entire mandatory field is entered.	
Location	**Expected result**
The user signup to aces platform	"Signup successfully". Display login page.
Pass: Yes	**Condition**: Pass **Fail:** No
Problem /issues: NIL	
status: successfully executed	

Test case title: Member Status

Table 11

Test case 3: Status	**Priority (H, L):** High
Test objective: Member posting a status	
Test descriptions: "user enter the required field, press post button", client program contact to the server, server contact with database, database update the result and send the results to the user.	
Requirements verified: Yes	
Test environment: IIS is in running state, Database contain appropriate table and link is established between server and client program.	
Test setup/ pre condition: IIS in running state. The entire mandatory field is entered.	
Location	**Expected result**
The member post status on platform	"Status post successfully". Display news feed page.
Pass: Yes **Condition**: Pass **Fail:** No	
Problem /issues: NIL	
status: successfully executed	

Test case title: Member Comment

Table 12

Test case 4: Commenting	**Priority (H, L):** High
Test objective: Member commenting on a post	
Test descriptions: "user enter the required field, press comment button", client program contact to the server, server contact with database, database update the result and send the results to the user.	

Requirements verified: Yes		
Test environment: IIS is in running state, Database contain appropriate table and link is established between server and client program.		
Test setup/ pre condition: IIS in running state. The entire mandatory field is entered.		
Location	**Expected result**	
The member comment on status	"Comment post successfully". Display post page.	
Pass: Yes	**Condition**: Pass	**Fail:** No
Problem /issues: NIL		
status: successfully executed		

Test case title: Member Message

Table 13

Test case 5: Message		**Priority (H, L):** High
Test objective: Member message to another member		
Test descriptions: "user enter the required field, press send button", client program contact to the server, server contact with database, database update the result and send the results to the user.		
Requirements verified: Yes		
Test environment: IIS is in running state, Database contain appropriate table and link is established between server and client program.		
Test setup/ pre condition: IIS in running state. The entire mandatory field is entered.		
Location	**Expected result**	
The member message sends to another member on	"Message send successfully". Display messages	

platform	page.
Pass: Yes	**Condition**: Pass **Fail:** No
Problem /issues: NIL	
status: successfully executed	

Test case title: Member article comment

Table 14

Test case 3: Comment on article	**Priority (H, L):** High
Test objective: Member commenting on article	
Test descriptions: "user enter the required field, press comment button", client program contact to the server, server contact with database, database update the result and send the results to the user.	
Requirements verified: Yes	
Test environment: IIS is in running state, Database contain appropriate table and link is established between server and client program.	
Test setup/ pre condition: IIS in running state. The entire mandatory field is entered.	
Location	**Expected result**
The member comment on article on platform	"Comment post successfully". Display article page.
Pass: Yes	**Condition**: Pass **Fail:** No
Problem /issues: NIL	
status: successfully executed	

Test case title: Article

Table 15

Test case 3: Article	**Priority (H, L):** High
Test objective: Posting article on platform	
Test descriptions: "user enter the required field, press post button", client program contact to the server, server contact with database, database update the result and send the results to the user.	
Requirements verified: Yes	
Test environment: IIS is in running state, Database contain appropriate table and link is established between server and client program.	
Test setup/ pre condition: IIS in running state. The entire mandatory field is entered.	
Location	**Expected result**
The article post on platform	"Article post successfully". Display article page.
Pass: Yes **Condition**: Pass **Fail:** No	
Problem /issues: NIL	
status: successfully executed	

Test case title: Edit article

Table 16

Test case 3: Edit article	**Priority (H, L):** High
Test objective: Editing article on platform	
Test descriptions: "user enter the required field, press save button", client program contact to the server, server contact with database, database update the result and send the results to the user.	

Requirements verified: Yes	
Test environment: IIS is in running state, Database contain appropriate table and link is established between server and client program.	
Test setup/ pre condition: IIS in running state. The entire mandatory field is entered.	
Location	**Expected result**
The editor edits the article on platform	"Edit article successfully". Display article page.
Pass: Yes **Condition**: Pass **Fail:** No	
Problem /issues: NIL	
status: successfully executed	

Test case title: Delete article

Table 17

Test case 3: Delete article	**Priority (H, L):** High
Test objective: Deleting article on platform	
Test descriptions: "user enter the required field, press post button", client program contact to the server, server contact with database, database update the result and send the results to the user.	
Requirements verified: Yes	
Test environment: IIS is in running state, Database contain appropriate table and link is established between server and client program.	
Test setup/ pre condition: IIS in running state. The entire mandatory field is entered.	
Location	**Expected result**
The editor delete article on platform	"Article delete successfully". Display article page.

Pass: Yes	**Condition**: Pass	**Fail:** No
Problem /issues: NIL		
status: successfully executed		

Conclusion

The proposed platform will provide more functionalities and friendly user interface to the user rather than the existing platform. On Proposed platform user functionalities includes a profile module for user to maintain user information, a status module for socialization between users and a comment module so users will be able to comment on other user status. An article module where users can read articles and a module for commenting on articles whereas existing platform provides only few functionalities for users to use. User cannot manage profile information and do not have a proper reading material from official researches. Users cannot find other users by searching their names so they can have a private conversation.

References

https://app.creately.com/ For use case diagram and Process flow diagram.

https://mockflow.com/ For user interface design.

https://app.genmymodel.com/ For sequence diagram.

https://online.visual-paradigm.com/ For Data flow diagram.

https://cloud.smartdraw.com/ For activity diagram

https://cloud.smartdraw.com/ For system diagram.

https://erdplus.com/ For entity relationship diagram (ERD).

https://sqldbm.com/ For schema diagram.

https://vapingcommunity.co.uk/ Reference for project.

https://online.visual-paradigm.com/ For class diagram.