



ADiplT02 – Object-Oriented Design and Programming

TOPIC: GROUP PROJECT REPORT

Group Name: Heisenberg

Members Name:

Pranaya Manandhar

Rujan Shrestha

Sabbir Shahi

Sneha Shah

Submitted date: December 28, 2019

Submitted to: Raj Prasad Shrestha

Contents

1: Project Description	1
1.1: Features:	1
1.2: Applications.....	1
2: Methodology	2
2.1: Agile.....	2
2.2: Why Agile?	2
2.3: Use Case Diagram	3
2.3.1: Explanation	4
2.4: Class Diagram	5
2.4.1: Description of class diagram	6
2.5: Activity Diagram.....	8
2.5.1: Login Case	8
2.5.2: Post Case.....	10
2.6: Sequence diagram	12
2.6.1: Description of above Sequence Diagram:.....	12
2.6.2: Description of above Sequence Diagram:.....	13
2.7: Swimlane Diagram:	14
2.8: N - tier Architecture	16
2.8.1: Why N-tier architecture?	17
2.8.2: Advantages of N-tier architecture are:.....	19
2.8.3: Disadvantages of N-tier architecture are:.....	19
Bibliography.....	20

Figure 1 : Use Case for Phoby	3
Figure 2: Class Diagram for Phoby	5
Figure 3: Activity diagram of Login Case.....	8
Figure 4: Activity Diagram of Post Case	10
Figure 5: Sequence Diagram of Register and Login case	12
Figure 6 Sequence diagram of Post Case	13
Figure 7: Swimlane diagram of Phoby.....	14
Figure 8: N- Tier Architecture	16
Figure 9: 3 Layers of N- Tier Architecture	17
Figure 10: Different Layers in N-Tier.....	18

1: Project Description

The name of the website is “Phoby”. It is basically an idea sharing platform where the users can embrace their hobbies by sharing relevant pictures of the preferred hobby and submerge into it. Users can select their preferred hobby and upload respective pictures and can also follow other users, comment on their posts and download the photo that catch their eye.

1.1: Features:

- ‘Phoby’ allows users to create their own personal accounts and surf the application.
- Users are able to upload photos of their preferred hobby along with suitable captions.
- Users can follow other users which give them the facility to see others post, hobbies, etc.
- Users can also like and comment on any post they desire.
- Users (Consumers) are also given the ability to download pictures posted by other users (Creators).
- Users can also save posts they like or want to see again later.
- Search function is available inside the web application.
- Users can customize their own profile.
- Users can share their hobbies detail through their profile page.
- Notification and news feed are provided to the users.

1.2: Applications

- Phoby basically is a place where we as a user can search for inspirations. Users can follow other users and see their hobbies for inspirations.
- It is a place where people can share their ideas. Users can share their own ideas and hobbies through Phoby.
- Users can also create portfolios.

2: Methodology

2.1: Agile

It is a process of breaking tasks into smaller iterations where the result is observed by the client to see whether the client is satisfied with the result or not. The major function of agile model is to make easy and rapid project achievement. This methodology reduces time and number of errors in the development or coding phase as it involves pair programming.

There are six processes of agile method:

- Requirements gathering
- Design the requirements
- Construction/ iteration
- Testing/ Quality assurance
- Deployment
- Feedback

2.2: Why Agile?

Agile allows a product to be developed more efficiently. Since this is a first project using python and Django, the project is prone to a lot of changes, agile allows an opportunity to change the requirements and add more functions or remove them along the way. This methodology is suitable for team work where there is more than one member and new ideas spark along the way. The process of agile methodology makes it easier for the team to work on a project step wise so, there is less chance of project failure since product is developed fast and frequently delivered. Having agile means having a product ready on all the time from the get go since it is built incrementally and even if the full product is not ready, the product has other functioning features so it is ready to be shared. Working on this methodology helps the team members work efficiently and effectively and late changes can be made quickly based on the feedback by the instructor or the module leader. (w3schools, 2019) (Javapoint, 2019)

2.3: Use Case Diagram

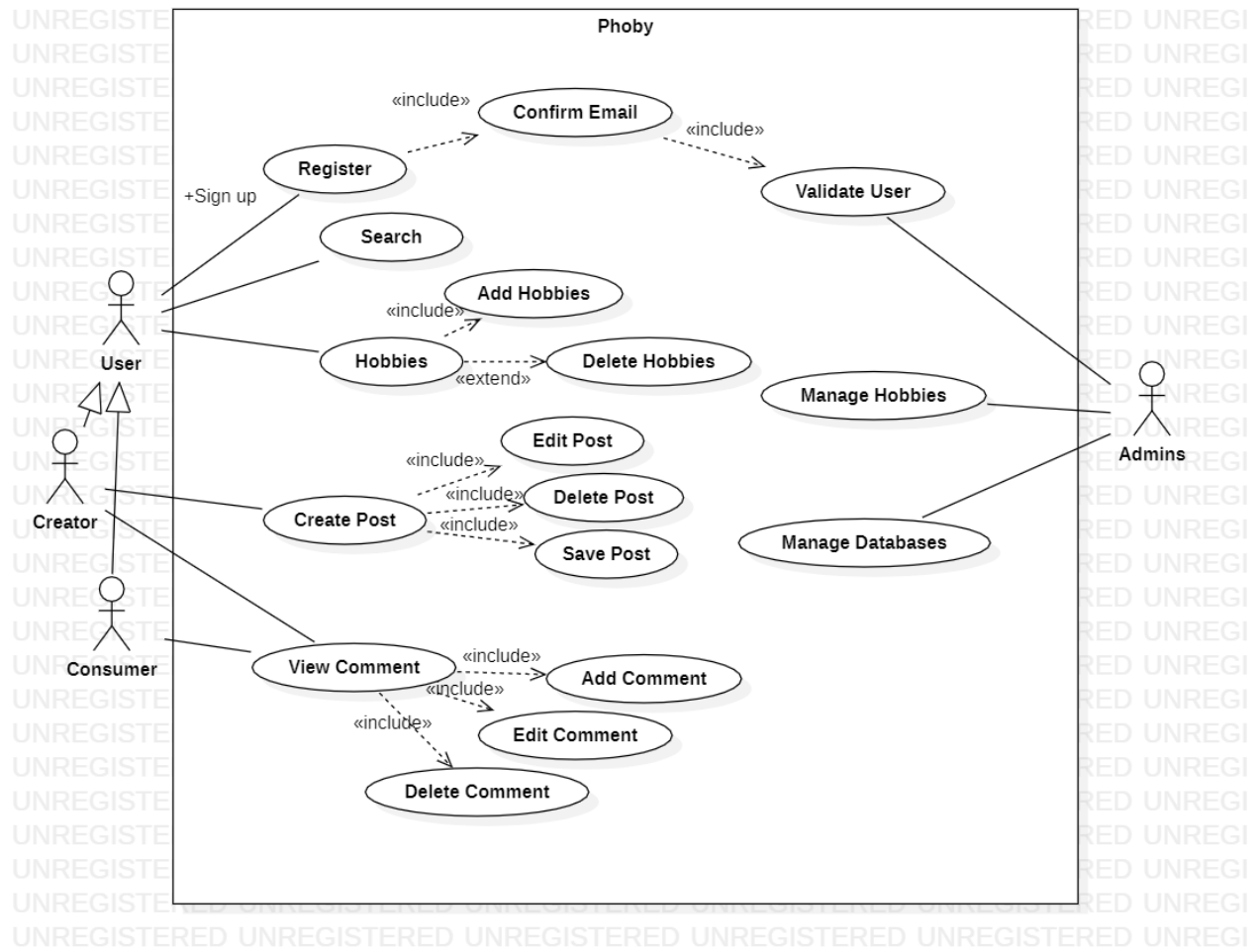


Figure 1 : Use Case for Phoby

2.3.1: Explanation

User Case ID	UC-101
Use Case	People sharing hobbies and searching for inspirations.
Actors	Users (Creators, Consumers), Admins.
Description	Users logs in, shares their hobbies (i.e. art, music, photography, etc.), or checks other's creations for inspirations.
Pre-conditions	Logged in the site.
Flow of Events	<ol style="list-style-type: none">1. User enters site.2. Site prompts user for account validation.3. Customer enters email and password.4. Site authenticates user.5. Site presents user with posts according to their hobbies and choices.6. User uploads posts according to their hobbies.7. User interacts with the post (like, comment and save).8. User view likes and comments.9. Users can use searches to fill their curiosity.10. User can logout or exit site.
Post-conditions	Site is closed or logged out.
Alternative Flows	User may not be able to connect to website, user may not be authenticated,
Priority	High
Assumptions	Users are highly active and communicative.

2.4: Class Diagram

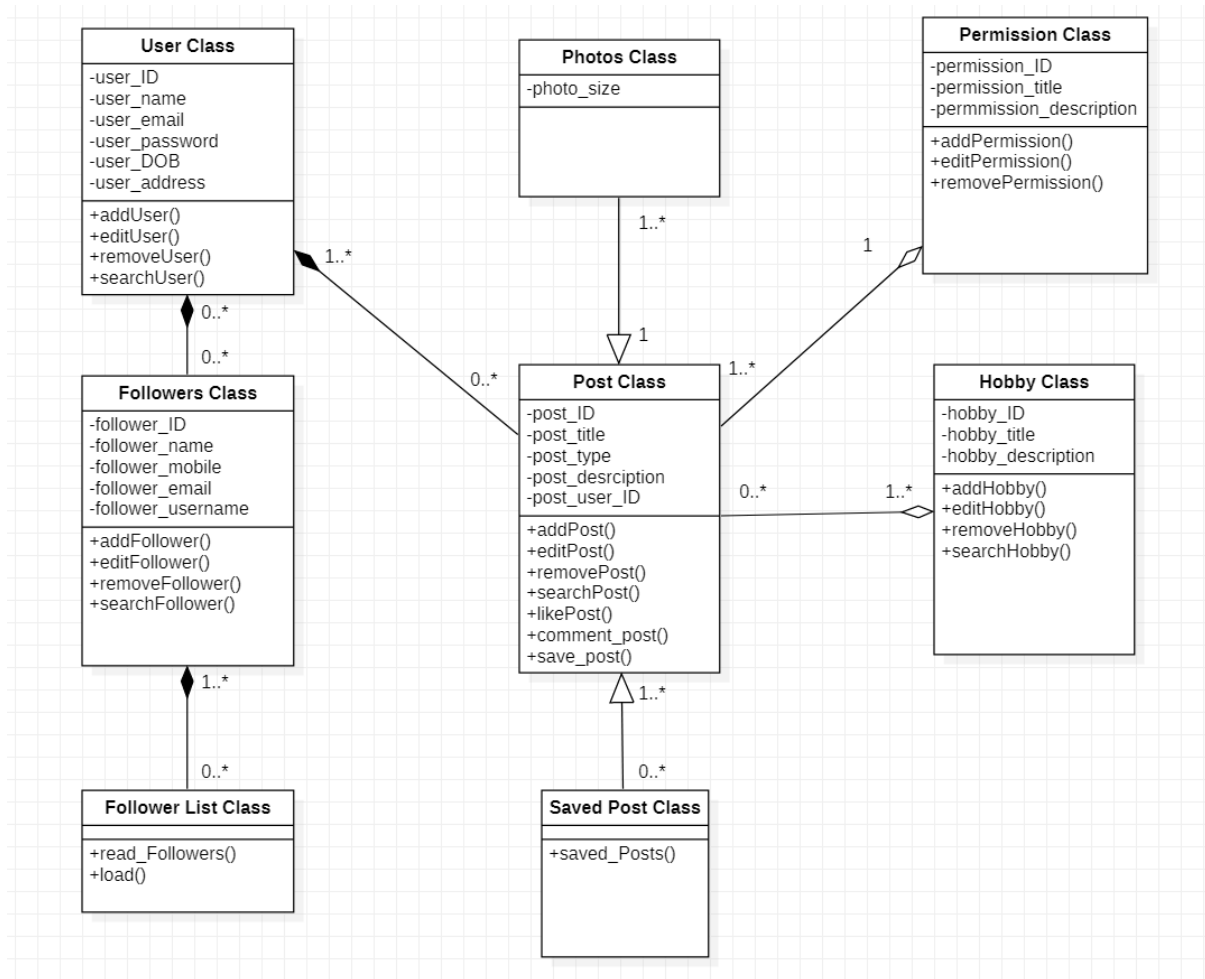


Figure 2: Class Diagram for Phoby

2.4.1: Description of class diagram

The class diagram consists of several classes such as:

- User class
- Followers class
- Post class
- Photos class
- Permission class
- Hobby class

User class consists of attributes and methods that are used or operated by the user. This class consists of basic attributes such as user ID, user name along with their email, password, Date of Birth and address. The methods or functions performed in this class is add, edit, remove and search other users.

Similarly, followers class consists of attributes such as their ID, name, mobile number, email and username. The functions performed in this class are also basic add, edit, remove and search functions through which you can search other followers, add or remove them. Follower List class also inherits the follower class for their attributes. It has aggregation relation with user class and follower list class has aggregation relation with followers class.

Post class consists of functions such as add post, edit, remove and search posts and also the ability to like, comment and search the respective posts. Its attribute consists of post ID, title, type, description i.e. what it's about and the post user ID. Photo class i.e. the photo you post also inherits this class with the extra attribute being photo size. It also allows you to save post and view the saved photos through another saved photo class. Similarly, photos class and saved post class inherits from this class and both hobby and permission class has aggregation relation with this class.

Permission class allows the user/poster to set permission for their posts such as public, private or other options which may later be available. It also has functions to add, edit or remove permission.

Hobby class groups all the available hobbies or interests into one main topic and allows the user to search the hobbies in which they would like to plunge in their free time. It has attributes such as hobby ID i.e. the ID of the hobby, title and the description. Functions include adding, editing, removing or searching the hobby.

Also, as show in this above class diagram one or many users can have zero or more post. Likewise, zero or more user can have zero or more followers and one or more followers can have zero or more followers in their list. One post can have one or more photo and one or more post can save zero or more posts. Similarly, one or more hobby class can have zero or more posts and one or more post can have one and only permission.

On the side note, the above described class diagram is such a specimen and may be subjected to future changes.

2.5: Activity Diagram

2.5.1: Login Case

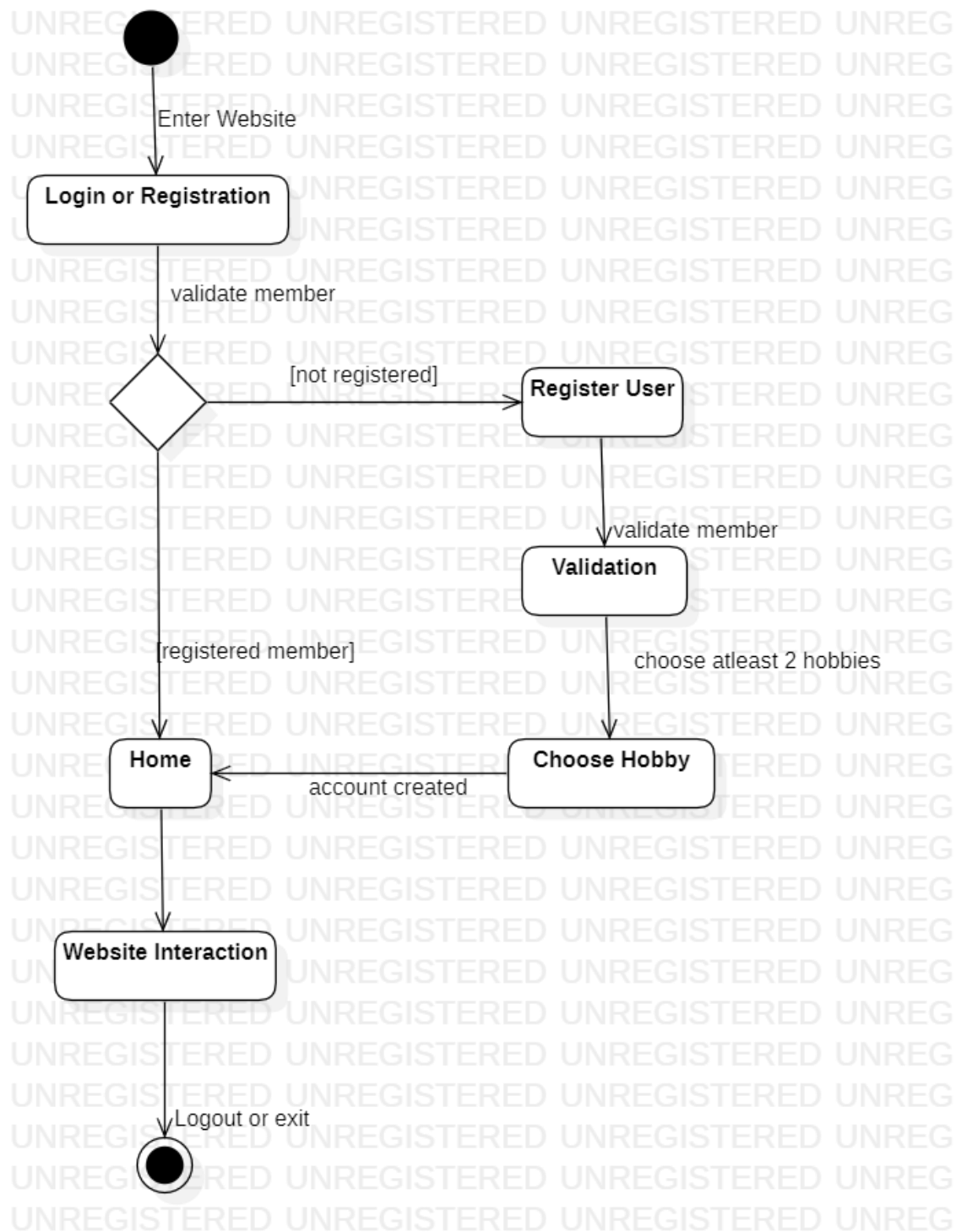


Figure 3: Activity diagram of Login Case

Here, there a user/actor enters the website and arrive at the login/register page. Here If the user is registered then the user logs in their credentials in the logins and arrives at the home page.

But if the user is not registered then they are registered to the website and they are validated by the system. After this the user chooses at least 2 hobbies and the new user is directed to the homepage where they interact with other apps in the website and logout.

2.5.2: Post Case

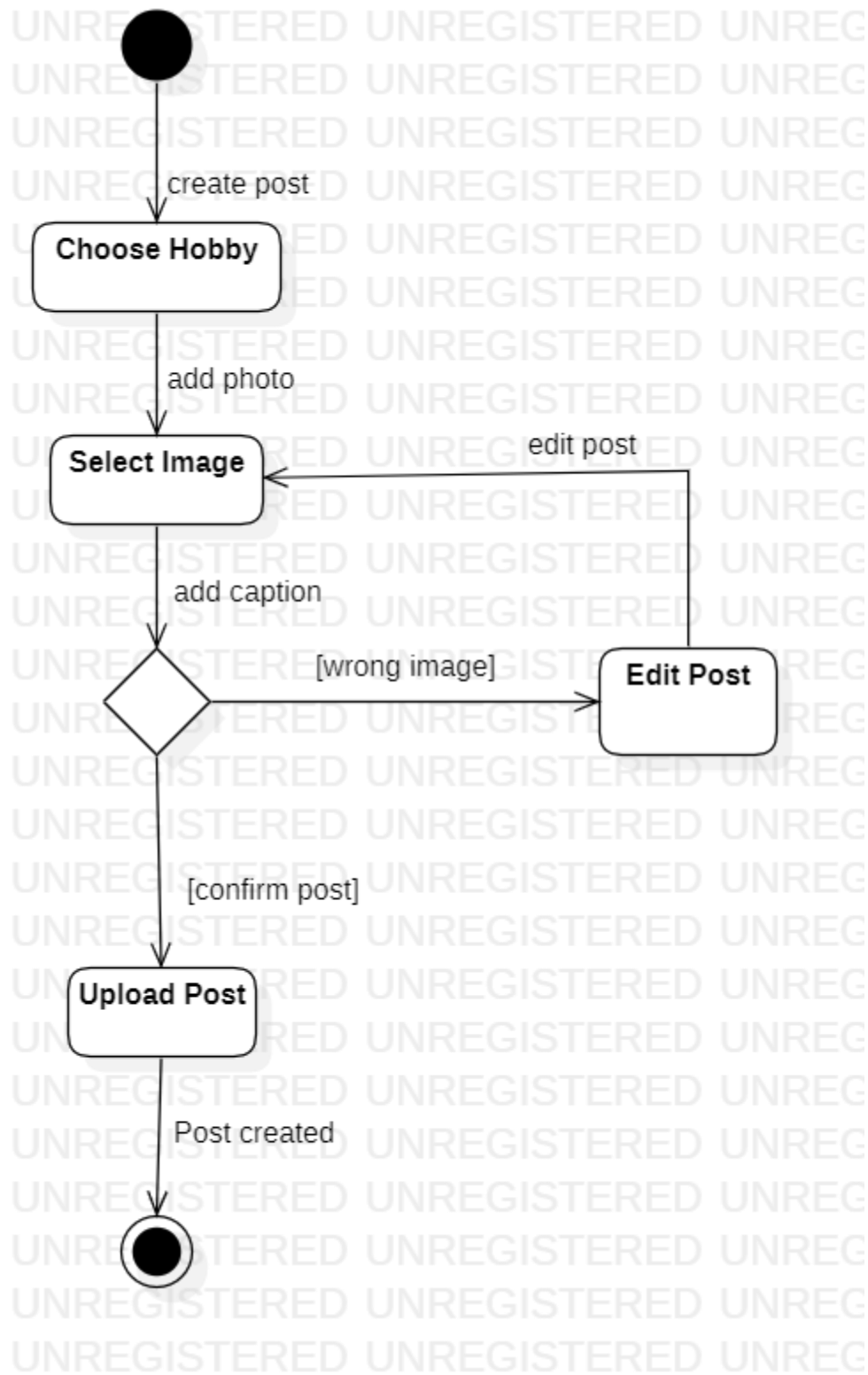


Figure 4: Activity Diagram of Post Case

Here, where a logged in user(creator) is creating a new post according to their hobby. First, the creator goes to create post and selects an image. After selecting image then the user is given an option to add a caption. If the process till this moment has no problems then the user uploads the post thus completing the post action. But if there were any problems in the process then the user would be directed back to the select image through edit post.

2.6: Sequence diagram

Register:

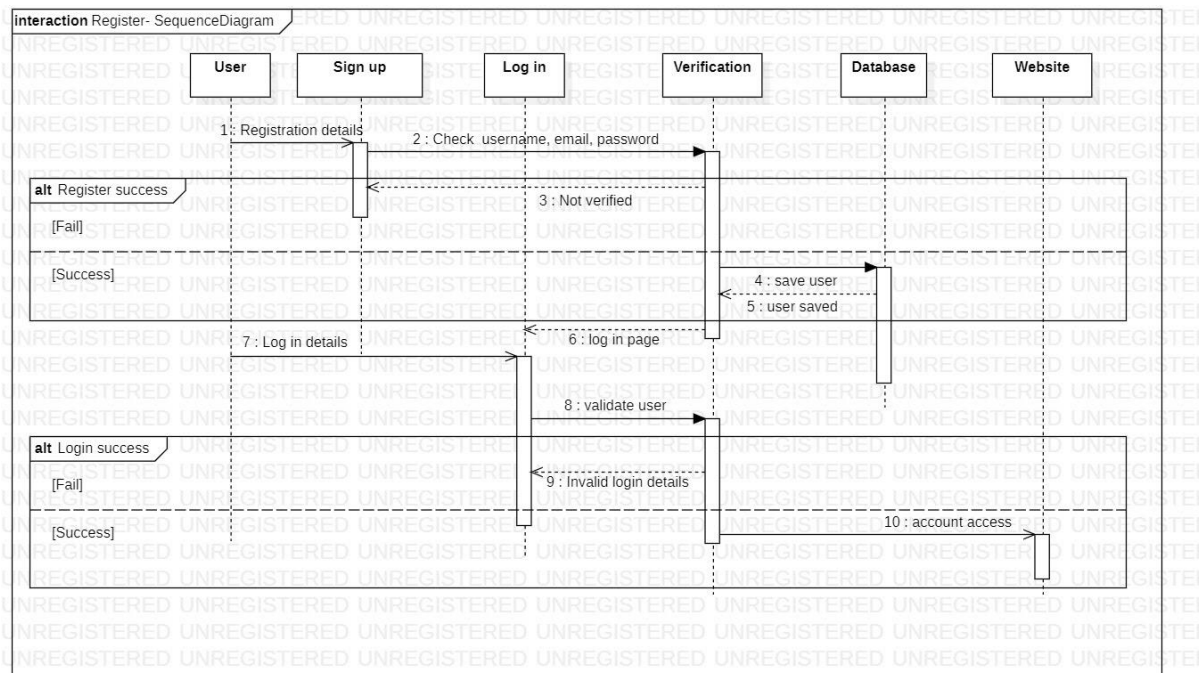


Figure 5: Sequence Diagram of Register and Login case

2.6.1: Description of above Sequence Diagram:

The sequence diagram consists of user, sign up, log in, verification, database and Website.

User starts by providing registration details to register (sign up) account. The System then checks whether the user name provided by the user is valid or not. If the provided username already exists, the user is taken back to the sign up page. If the credentials provided by the user is satisfactory then the account is saved in the database. After the account has been made, login page is opened, now the user provides login details like username and password to gain access to the website. The system checks whether the user is a valid user or not. If not then the user is returned back to the login page and if user is valid then user is accessed to the website.

Post:

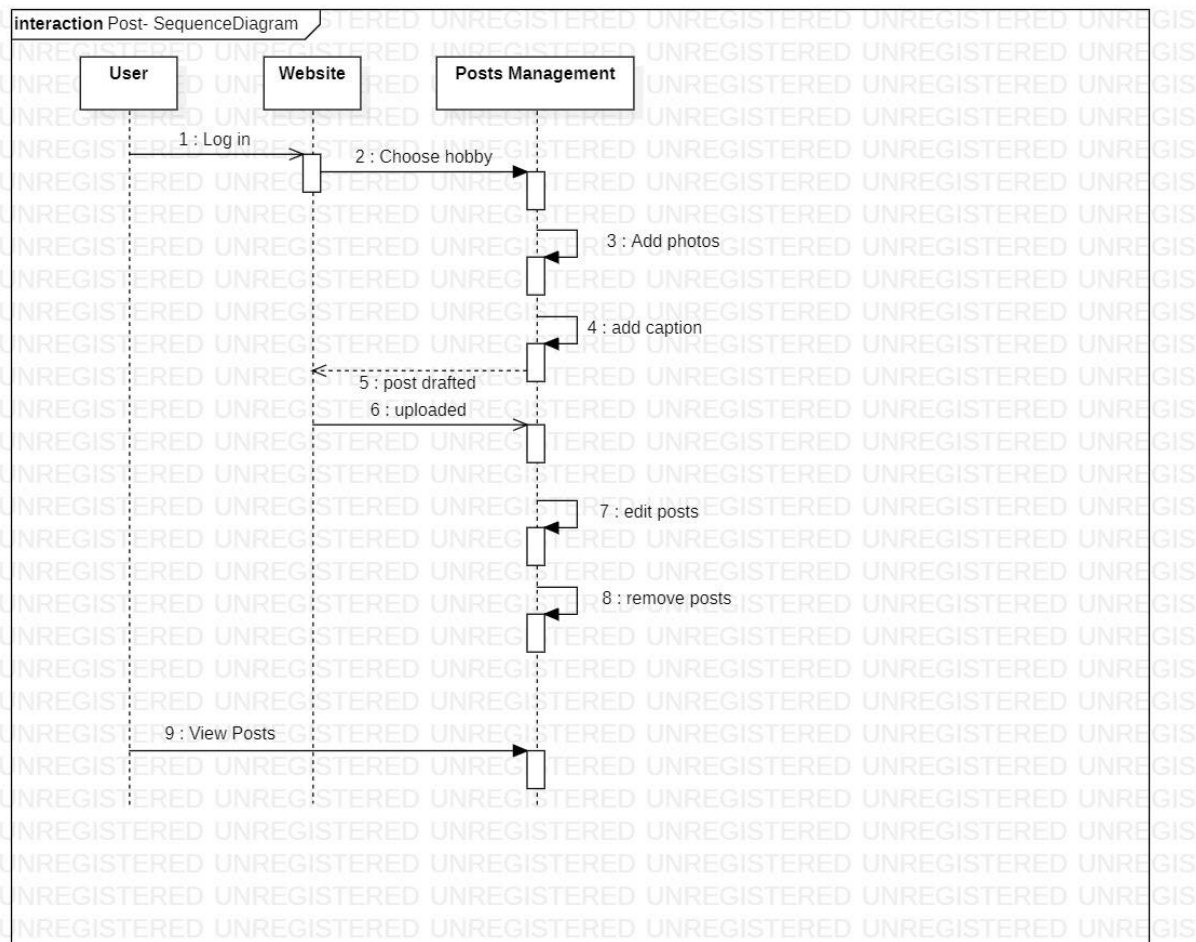


Figure 6 Sequence diagram of Post Case

2.6.2: Description of above Sequence Diagram:

The sequence diagram consists of user, Website and Post management.

After the user logs in to the account, the user has to choose a hobby. The user adds photos with captions then the post is drafted. After uploading the post, the user can edit the post as well as delete the post. Whereas other users can view the post.

2.7: Swimlane Diagram:

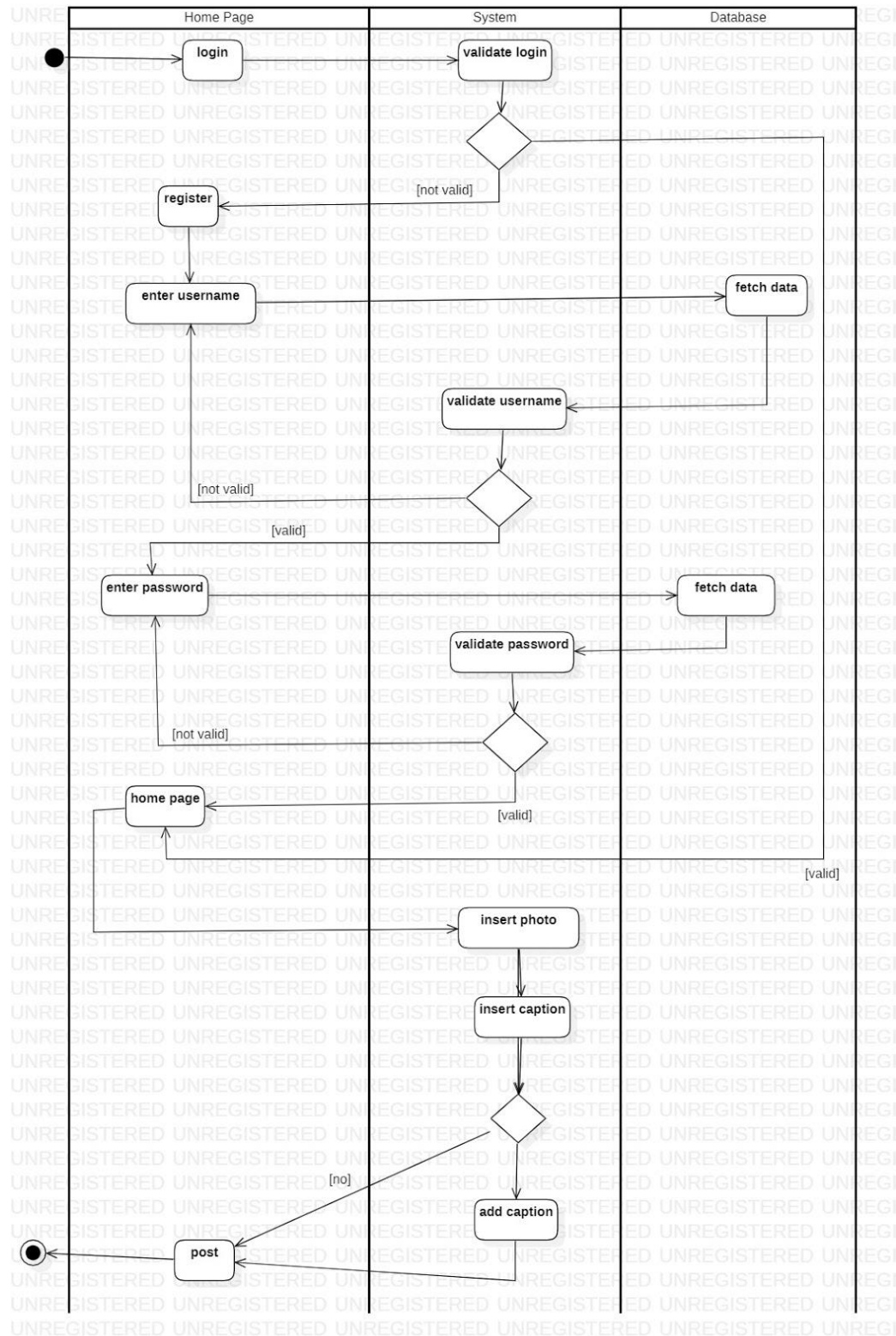


Figure 7: Swimlane diagram of Phoby

Above figure represents swim lane diagram where the workflow shows the process of the web app.

The three classes of this diagram are Home Page, System and Database. The process begins with a black circle at the top. Each rectangle box (action) which represents a task in the process, each diamond box (decision) represents query that must be asked before the next step. The black circle at the bottom ends the process.

Starting the process first, the user logs in the home page. The user enters their credentials and if the login fails, the user should register a username and password. The data of the used/available username is fetched from the database. After fetching the data, the username and password is verified. If it is validated, the system processes the user to the home page. If the username and password is not verified or validated, the user must enter the username or password respectively again. If the login succeeds, then the system proceeds to the home page. After that, the user can upload any photo of their liking in the respective hobby and insert respective caption. Inserting the caption is optional so if the user wants to write captions, the caption is added and if not, then the post is uploaded without the caption.

2.8: N - tier Architecture

N - Tier architecture, also known as multi-tier architecture divides application into physical and logical tiers.

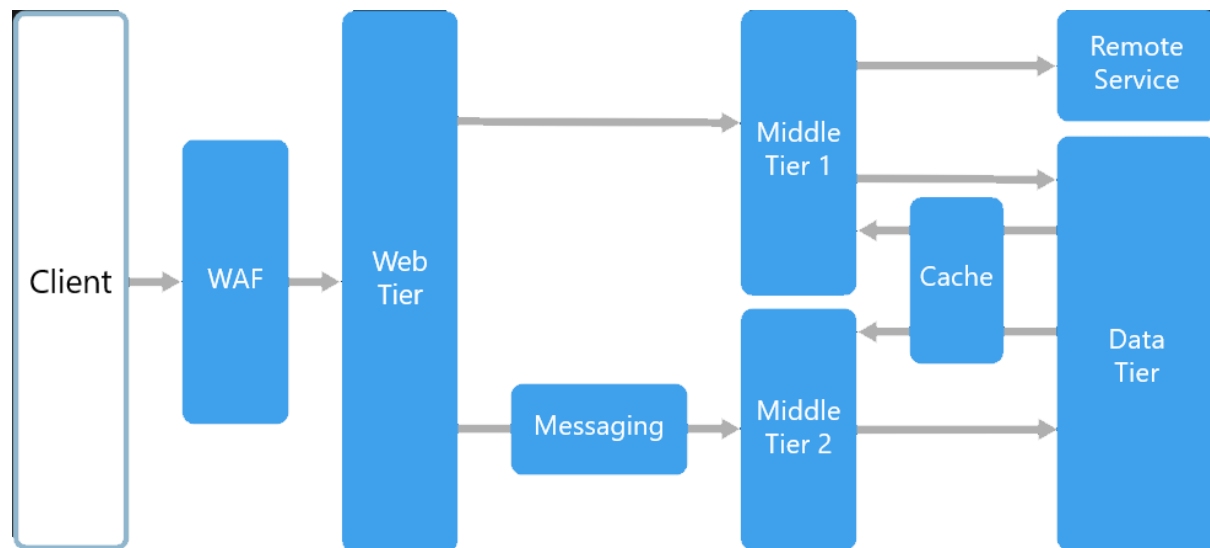


Figure 8: N- Tier Architecture

(Microsoft, 2018)

Tiers (physical unit) are a way to separate and manage dependencies. Each tier is separated physically and has a specific responsibility. A tier can call another tier directly, or use a message queue (asynchronous messages).

Similarly, layers are a logical unit i.e. how to organize a code. A single layer might be hosted on a single tier or many layers might be hosted on the same tier. A higher layer in a n-tier architecture may use the services of a lower layer but a lower layer cannot use the services of a higher layer.

An N-tier application can have closed layer architecture or an open layer architecture:

- In closed layer architecture, a layer can only call the next layer immediately down.
- Whereas, in an open layer architecture, a layer can call any of the layers below it.

A closed layer limits the dependencies between layer however creates an unnecessary network traffic in the process. (Microsoft, 2016)

2.8.1: Why N-tier architecture?

We chose N-tier because N-tier architectures are not restricted to three tiers or any number of tiers. 3-tiers always has 3 tiers, while N-tier has a variable number of tiers. Basically, 3-tier is N-tier where $N = 3$.

A typical 3-tier architecture consists of:

- Presentation Tier
- Application Tier
- Data Tier

As shown below:

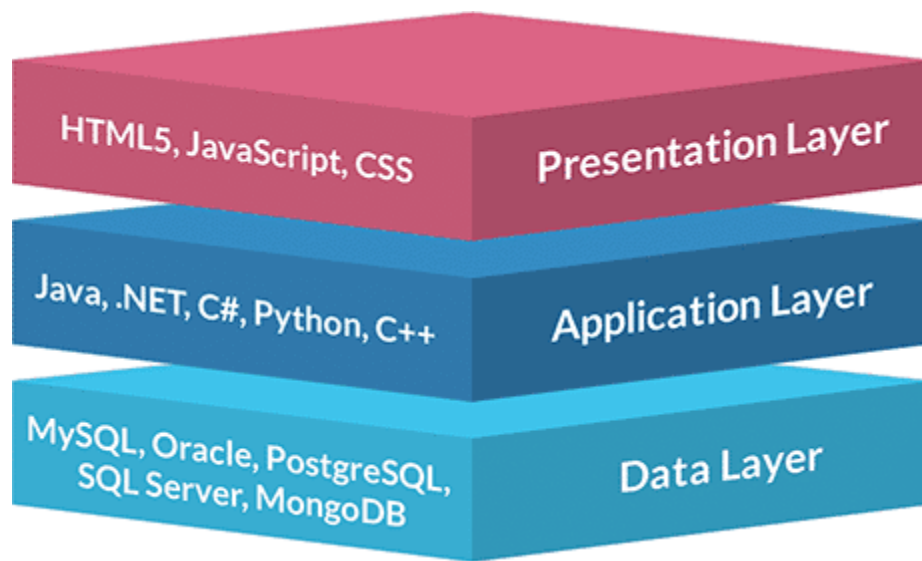


Figure 9: 3 Layers of N- Tier Architecture

(JReport, 2019)

Some layers in 3-Tier can be broken further into more layers. These broken layers may be able to run in more tiers.

For example, the application layer can be broken into business layer, persistence layer or more.

Presentation layer can be broken into client layer and client presentation layer.

A simple example of .net framework is given below:

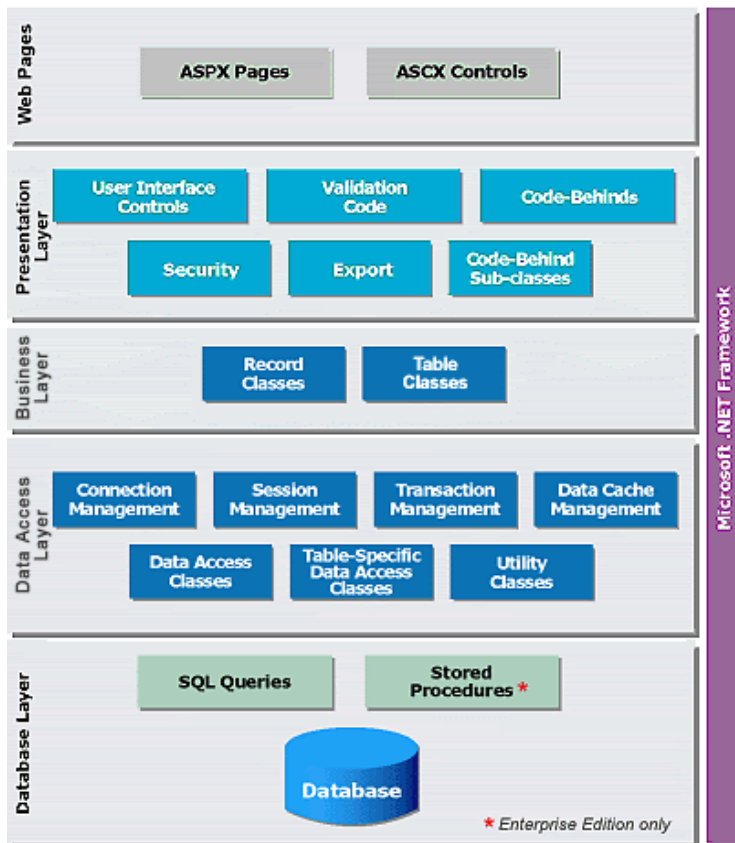


Figure 10: Different Layers in N-Tier

(c-sharpcorner, 2015)

So basically, it is 3-tier but more scalable.

2.8.2: Advantages of N-tier architecture are:

- **Secure:** Each of the tiers can be secured separately using different methods.
- **Scalable:** This is due to its capability of multiple tier deployment and the tier decoupling it brought. For example, the data tier can be scaled up by database clustering without other tiers involving.
- **Flexible:** Apart from isolated scalability, each tier can be expanded in any manner that the requirements dictate.
- **Better fault tolerance ability:** for example, the databases in data layer can be clustered for failover or load balance purpose without affecting other layers.
- **Independent tier upgrading and changing without affecting other tiers:** in object-oriented world, Interface-dependency implementation can decouple all layers very well so that each layer can change individually without affecting other layers too much.
- **Friendly for maintenance:** N-Tier architecture groups different things together mainly by functionality and then makes things clear, easily understandable and manageable.
- **Friendly for new feature addition:** Due to the logical grouped components and the decoupling brought by N-Tier architecture, new features can be added easily without affecting too much on the whole system.
- **Better reusability:** This is due to the logically grouped components and the loose couplings among layers. Loosely-coupled component groups are usually implemented in more general ways, so they can be reused by other applications.

2.8.3: Disadvantages of N-tier architecture are:

- The performance of the whole application may be slow if the hardware and network bandwidth aren't good enough because more networks, computers and processes are involved.
- More cost for hardware, network, maintenance and deployment because more hardware and better network bandwidth are needed. (Hu, 2013)

Bibliography

c-sharpcorner, 2015. *C-sharpcorner*. [Online]

Available at: <https://www.c-sharpcorner.com/forums/what-is-tier-tier-difference-between-tier-and-layer>

[Accessed 28 December 2019].

Hu, W., 2013. *Code Project*. [Online]

Available at: <https://www.codeproject.com/Articles/430014/N-Tier-Architecture-and-Tips>

[Accessed 25 December 2019].

Javapoint, 2019. *Javapoint/agilemodel*. [Online]

Available at: <https://www.javatpoint.com/software-engineering-agile-model>

[Accessed 10 December 2019].

JReport, 2019. *JReport*. [Online]

Available at: <https://www.jinfony.com/resources/bi-defined/3-tier-architecture-complete-overview/>

[Accessed 28 December 2019].

Microsoft, 2016. *Microsoft*. [Online]

Available at: <https://docs.microsoft.com/en-us/visualstudio/data-tools/n-tier-data-applications-overview?view=vs-2015&redirectedfrom=MSDN>

[Accessed 27 December 2019].

Microsoft, 2018. *Microsoft*. [Online]

Available at: <https://docs.microsoft.com/en-us/azure/architecture/guide/architecture-styles/n-tier>

[Accessed 28 December 2019].

w3schools, 2019. *w3schools/sdlcagile*. [Online]

Available at: <https://www.w3schools.in/sdlc-tutorial/agile-model/>

[Accessed 10 December 2019].