ADipIT02 - Object Oriented Design and Programming

❖ Continuous Assessment – I
❖ Group Report

- **Group: IDEA**

- **Project**: Idea Sharing Platform

- **Members:**
  - Luv Uprety
  - Bibash Poudel
  - Anish Poudel
  - Muskan Khatun

- **Submitted to:** Raj Prasad Shrestha

Table of Contents

1. Project Description.

The goal of this project is to create a platform where anyone can share their ideas with others so that they can get help from others to implement their ideas or to build something. Today's generation is the generation of science and technology and it is the technology through which people are connected with each other. Number of people today are unemployed but have marvelous business ideas, or ideas that may change the future of everyone. To implement such ideas, they need help and resources from other people. So, creating a platform where people with ideas can connect with each other would be able to solve that problem.

1.1 Applications and features:

- People with any business ideas and ways to implement them can share with others.
- Any interested people fascinated by others ideas can connect with them and join on their project.
- People with business ideas can get help form sponsors connected to our sites.
- Users can view each other's profile on request acceptance.

2. Methodology:

Incremental Methodology:

We would be using Incremental Software Development Life Cycle model for the development of the project where the model is incrementally designed, implemented and tested (each time a little more is added) until the product is finished. This includes growth as well as maintenance. When it meets all of its requirements, the product is defined as finished.

Advantage:

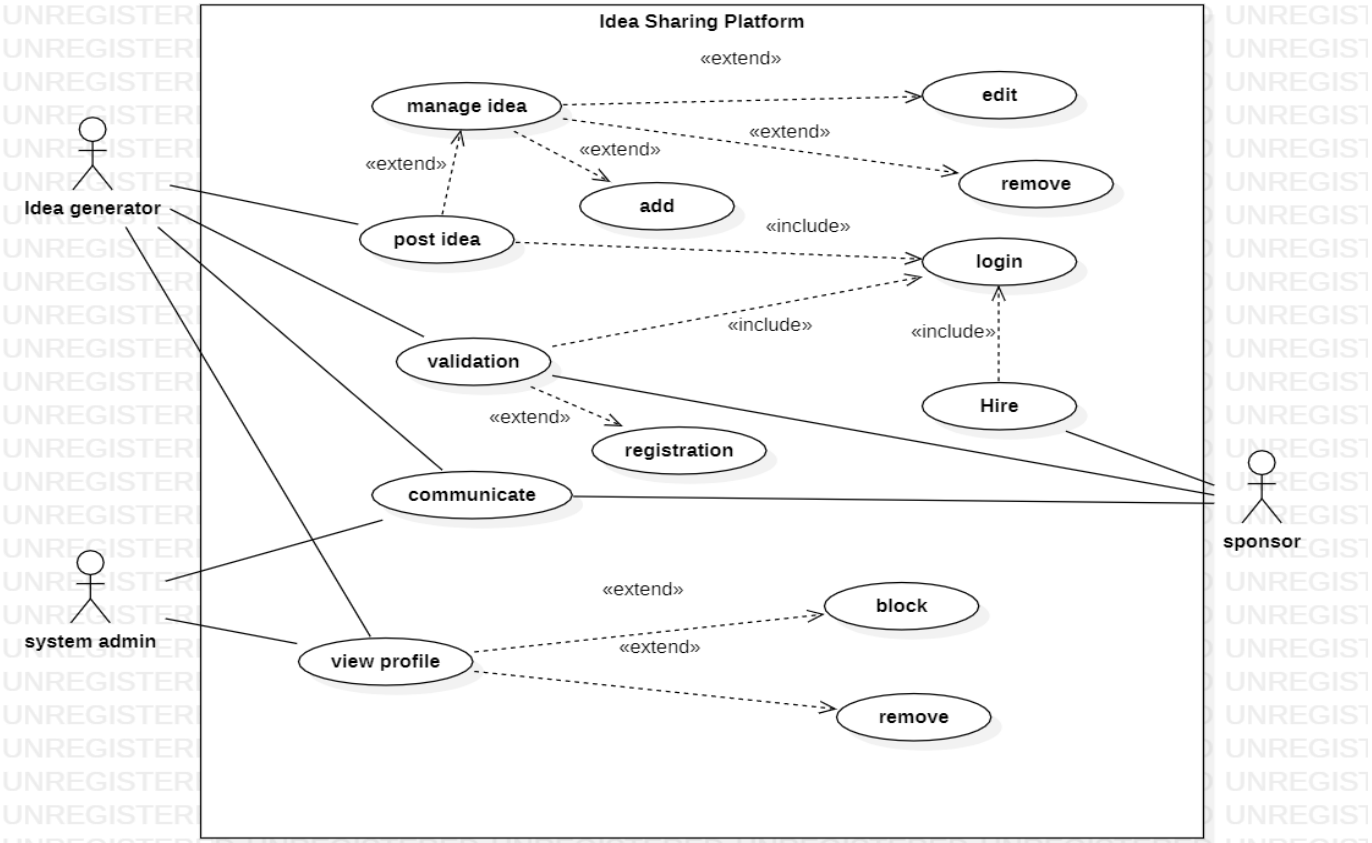
- Generates fast and early running software during the life cycle of the program.
- More versatile and less costly to adjust specifications and range.
- During a smaller iteration, easier to test and debug.
- Easier to manage risk due to the detection and handling of risky items during iteration.
- Every iteration is an achievement that is easily managed.

This methodology also has some of its disadvantages. Some of them are as follows:

- Each iteration step is stable and does not overlap.
- System design problems that occur because not all specifications are installed at the front for the entire life cycle of the code.

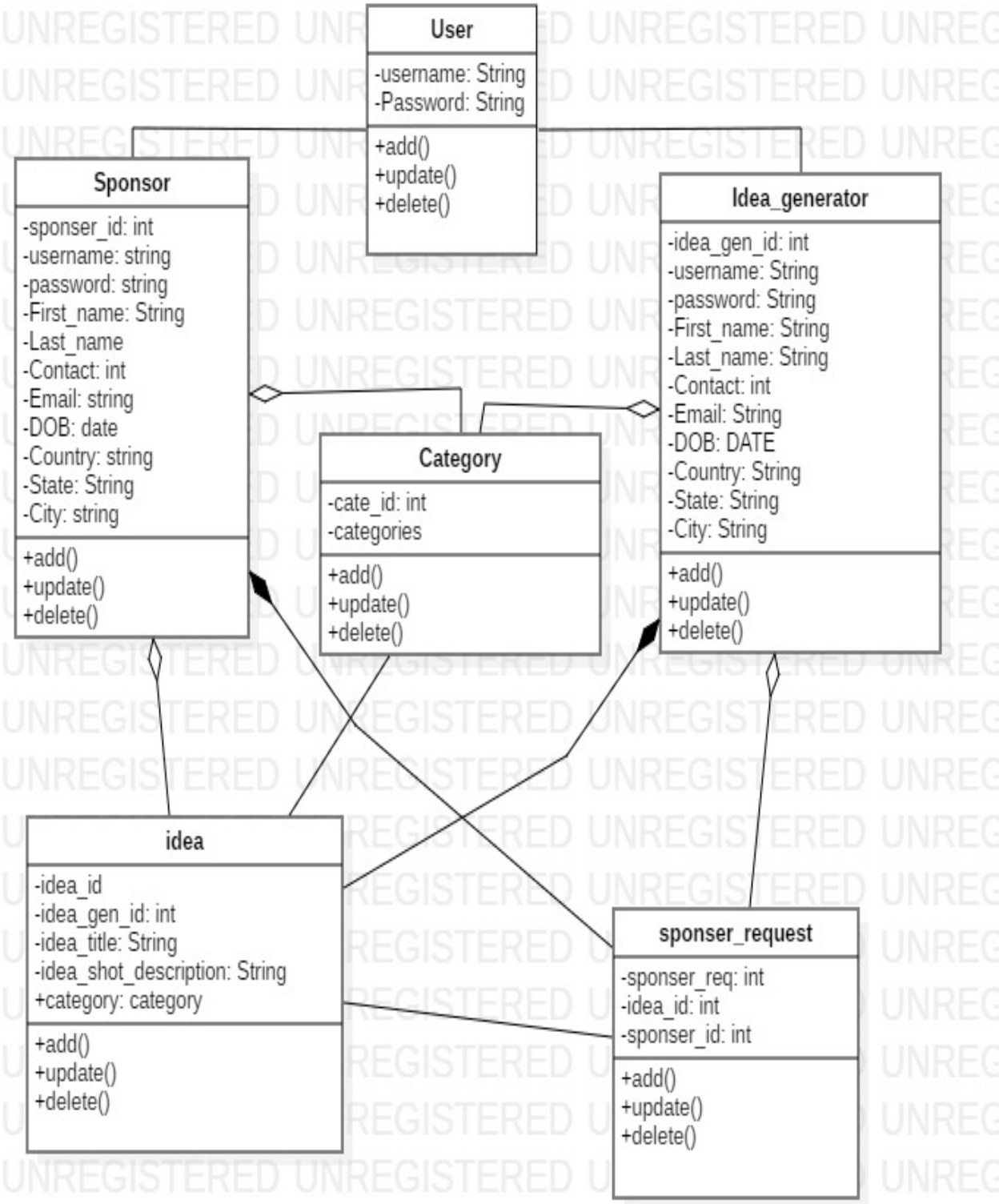
(Testing Excellence, 2018)

3. Use case diagram



Explanation: The system contains three users. They are: idea generator, sponsor and system admin. Idea generator can post idea, communicate with sponsor and system admin, view profile and more. To post the idea, idea generator should register first if not registered and login. Idea generator can also manage the ideas after posted which includes: add, remove and edit. Sponsors can also communicate with idea generator to hire them, view profile of idea generator. Sponsor must login first to hire idea generator or join on their project. System admin has superior authority, view profile of idea generator and sponsor at once, block profile or remove them on policy violation.

4. Class Diagram



Explanation:

User class: it defines that there are two type of user one is sponsor who can spoons on the idea of the idea generator user.

Sponsor class: this class hold the details of the sponsor. As they can spoons the idea.

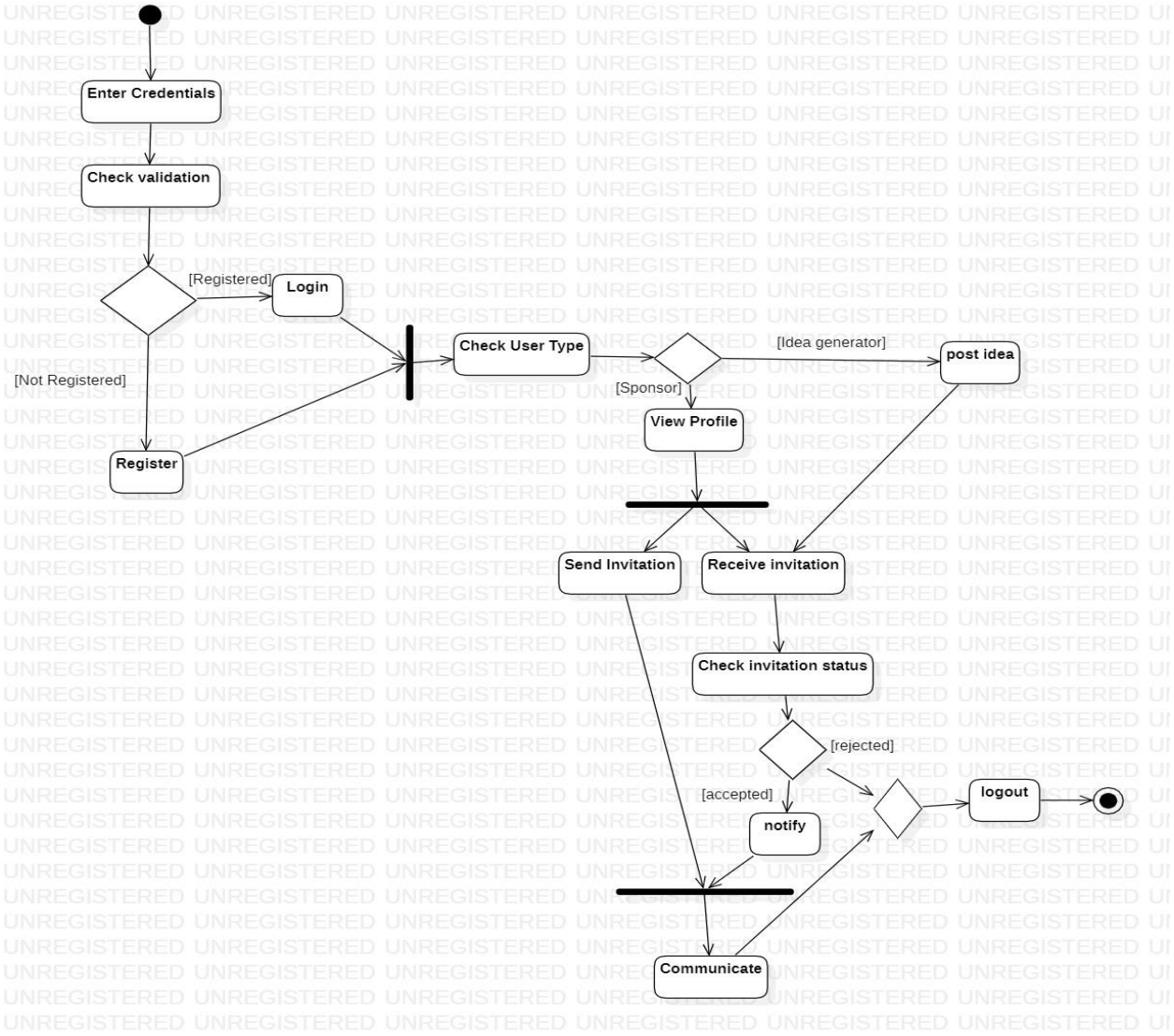
Idea_generator class: this class hold the details of idea generator. As they can post idea.

Idea class: this class hold the idea posted by idea generator. It depends upon the idea generator class

Sponsor_request class: this class hold the id of sponsor id of idea though which id_generator can link to the sponsor. This class depend upon the sponsor class.

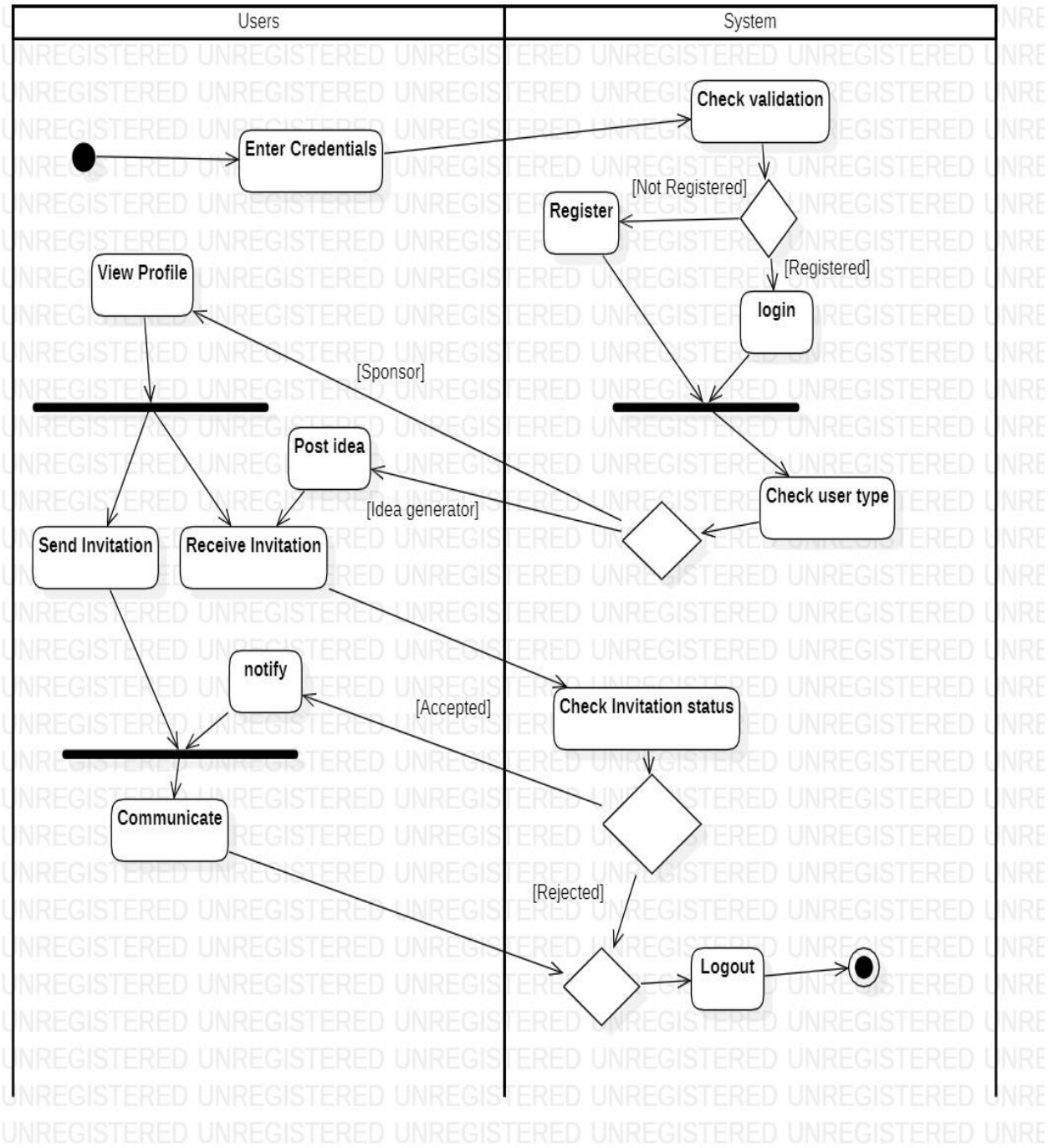
Category class: this call holds the category type of idea e.g. if idea is related to technology then the category is IT and technology.

5. Activity Diagram



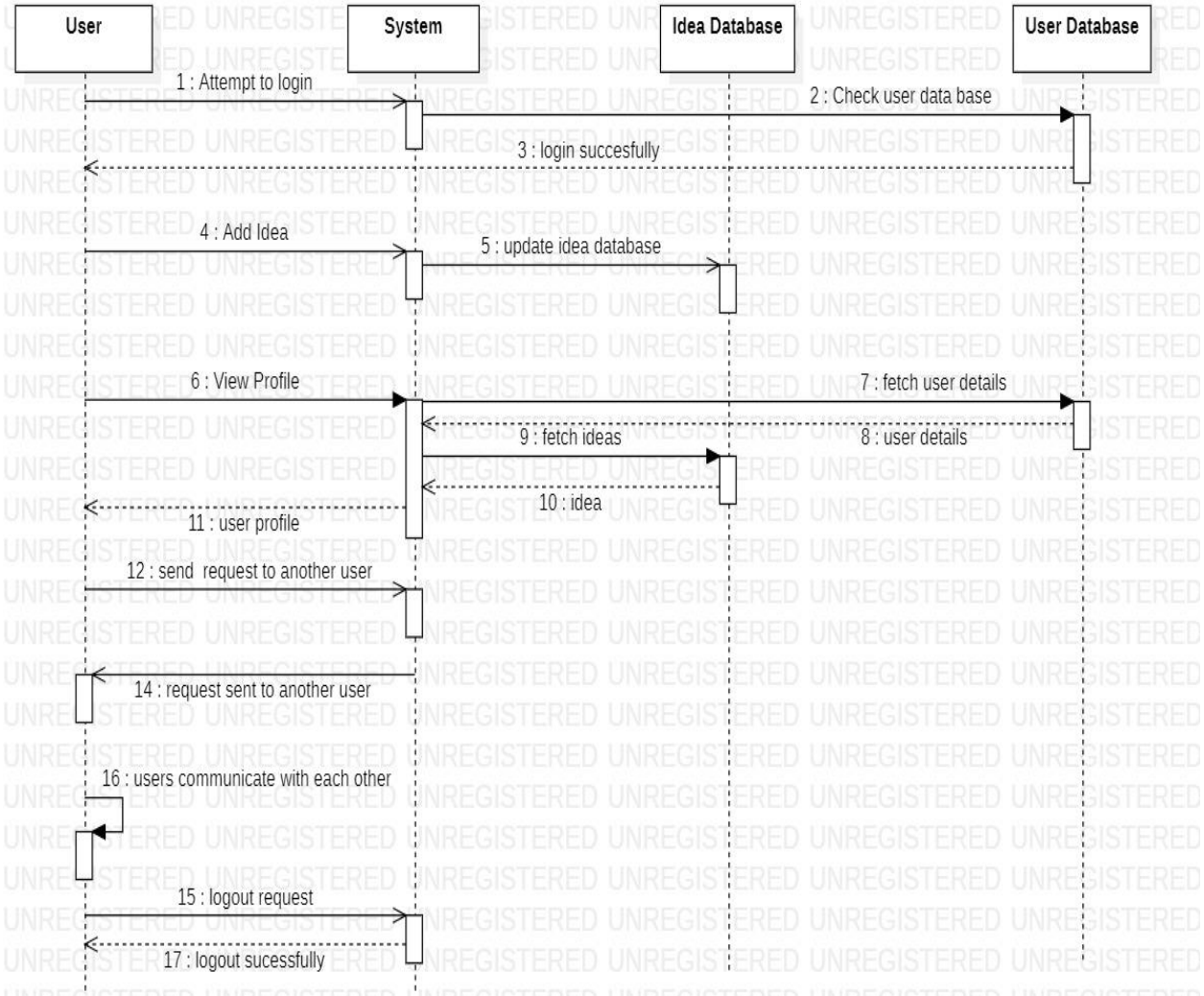
Explanation: Here the user enters the login credentials first. Then the system checks the validation of the user. If the user has already registered then the user is successfully logged in. If not then the user has to register first. After that system checks the user type (Idea Generator or Sponsor). If the user is idea generator then the system takes the user to idea generator dashboard where the user can post his/her ideas. If the user is sponsor then the system takes the user to sponsor dashboard where the user can view different ideas generator profiles, their ideas and send invitation request to communicate. If the idea generator accepts the invitation request then both the user can communicate with each other. If not then the users can logout.

6. Swimlane



Explanation: Here the activity diagram is presented in a Swimlane categorized into Users and System where activities carried out are shown respectively. The description of the swimlane is same as the activity diagram which you can find in the paragraph above the swimlane diagram.

7. Sequence diagram



Explanation: Here the user attempt to login into the system first. Then the system checks the user database for the current user as per the login credentials provided. If the credentials match with the database then the user is successfully logged in. Then the user can add his/her ideas which is stored to idea database by the system. If the user wants to view his/her profile or others profile then the system will fetch the details from user database and ideas from the ideas database and provide it to the user as a profile. If the user is sponsor then he/she can send invitation request to idea generator through the system. If the idea generator accepts the request then both the user can communicate and finally users can logout.

8. Software Architecture Pattern

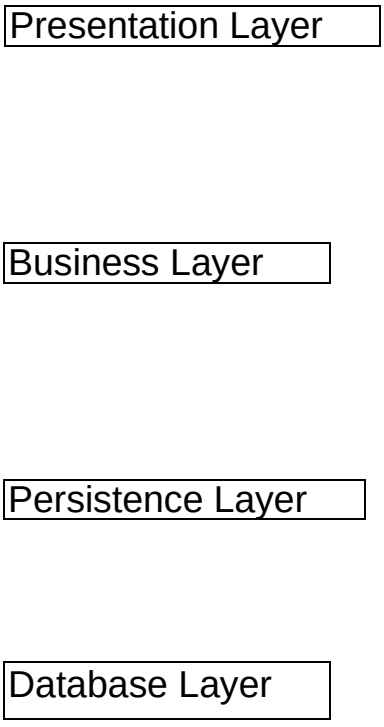
It defines how the component of a software system are organized, assembled and communicate with each other.

For our web application, we have planned to use Layered pattern / N TIRE – Pattern.

Layered Pattern:

This layer contains a different layer having its responsibility. Each layer performs a specific role in its application. Where request more from layer to layer.

There are different layers in the layered pattern they are as follows:



Each layer has its own responsibility

Presentation Layer:

It is also called the UI layer where users can view different posts and ideas.

And also have a different option.

Business Layer:

In this layer different activities take place like delete, update, edit, write and hire.

Persistence layer:

This layer decides according to the business layer, which makes simple SQL call to the database

Database layer:

In this layer data are store

Why we choose it:

For Example, to fetch the Idea-generator profile

Presentation layer response to a request from the sponsor to retrieve the idea-generator profile (data). The presentation layer pass request to the business layer, which simply passes the request to the persistence layer, which then make a simple SQL call to the database layer to retrieve the idea-generator profile (data).

Advantage:

- Easily Maintainable
- Easily Testable

- Make easy to assign separate "roles"
- Makes us easy to update and enhance layers separately

Disadvantage:

- The modules don’t have clear roles or relationships.
- Much of the code can be devoted to passing data through layers without using any logic.
- Layer isolation, which is an important goal for the architecture, can also make it hard to understand the architecture without understanding every module.
- Create tight coupling and produce a logical mess full of complex interdependencies.

Small changes can require a complete redeployment of the application.