Software Requirement Specification

Placement Portfolio

Author(s):
Vignesh kumarS
Date :20.11.2024

Content

- **Abstract**

The main purpose of this web application is to provide for students to crack the interview in easy way. The student can visit the website for specific company interview questions which all are asked in the previous years. The user can either log in or create an account as that is necessary to. The data entered by the user will be validated and if the data is valid, it will automatically be stored in the database.

- **Module**

    - **Home Page**

    The application starts with the Homepage and this page gives the user an overview of what the application is about. At the top-right of the homepage application, you can see the “login/sign-up" tabs. The homepage also allows the user to directly login/sign-up.

    - **Login/Sign-up**

    So now the user can login/sign-up from the login page. Now if the user already has an account, they can directly login. Otherwise, they will have to create an account by

signing up. The user just has to enter the email id, password. Then the admin would validate the data and if the data provided by the user is valid, then it will be stored in the database and the user's account has been created.

- **Specific Company Name**

After login users are able to visit the specific company profile for access the interview questions. They can choose their needed company and able to visit the question asked in that company in the previous years.

- **Interview Questions**

Once a user enters in to any particular company profile. The user can able to see the interview questions like aptitude, technical and HR questions. By using this placement questions, they can able to crack the interview.

- **Functional Requirement**

  - **Login/Sign-up**

So now the user can login/sign-up from the login page. Now if the user already has an account, they can directly login. Otherwise, they will have to create an account by signing up. The user just has to enter the email id, password. Then the admin would validate the data and if the data provided by the user is valid, then it will be stored in the database and the user's account has been created

- **Company name**

  After login users are able to visit the specific company profile for access the interview questions. They can choose their needed company and able to visit the question asked in that company in the previous years.

- **Non-Functional Requirement**

  - **Reusable code**

    Reusable code should always be a key factor that needs to be considered. The code should be reusable because it reduces the time it takes to develop an application. The code should also have the ability to build larger things from smaller parts

  - **User friendly**

    The web application should be user friendly as it is one of the most important factors. If it is not user friendly the users would not want to come back. Everything should be available to the user and the user should not be looking for anything. For example, if the user is looking for a particular product, but the product is not available, under that image of the product, it should be displayed saying the product is out of stock. So, basically whenever the user logs into this web application, the customer should have a very smooth experience.

- **High-Level Design**

- The Homepage will give the user what this online website is. It will contain small snippets of information about the website.
- The login/sign-up button will always be at the top.

- **Low-Level Design**

  - The small snippets of information on the homepage are the type of products.

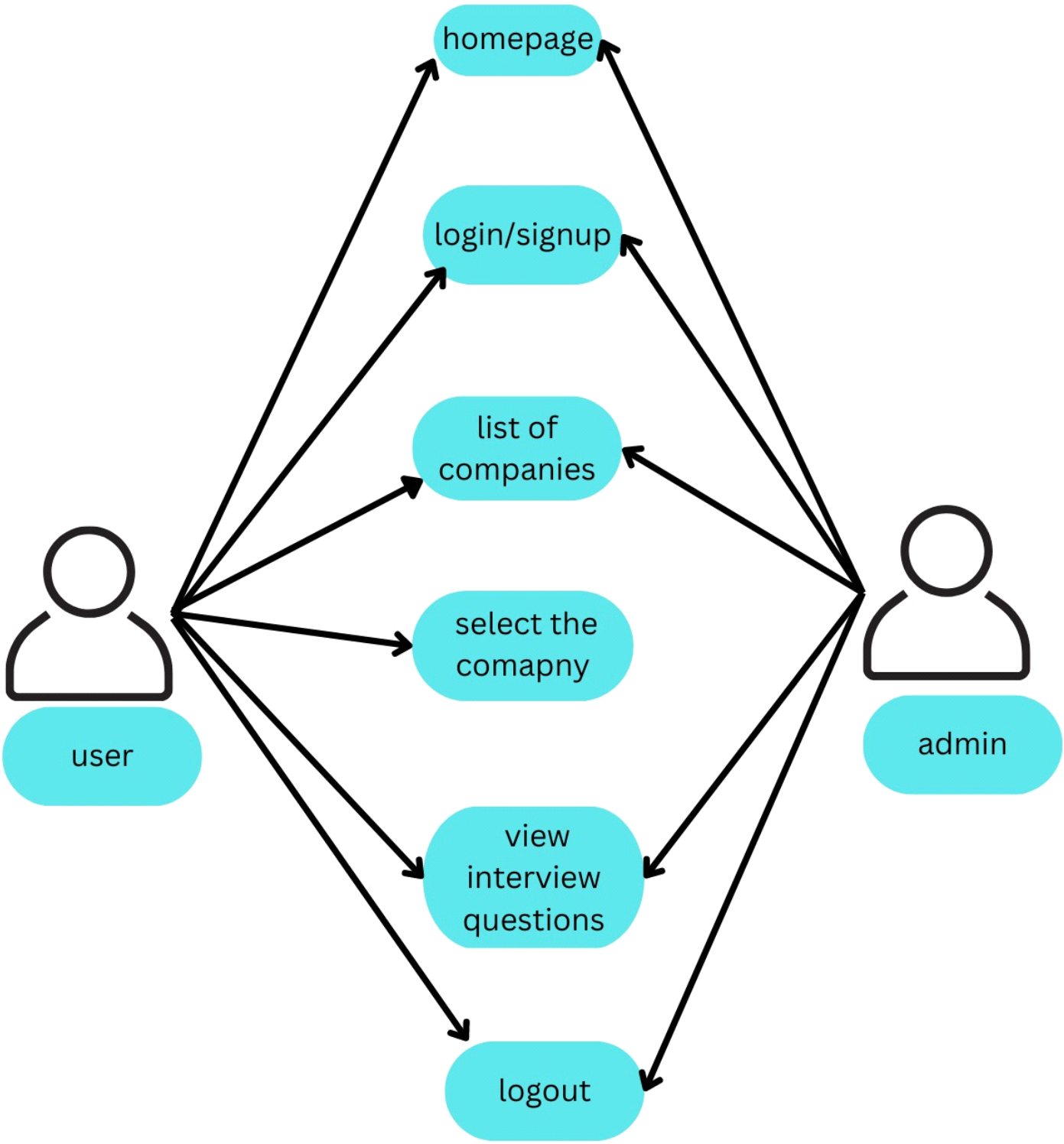
  - Some basic information about how it works.

Various other Non-Functional Requirements are:

- Security
- Reliability
- Maintainability
- Portability
- Extensibility
- Reusability
- Compatibility
- Resource Utilization

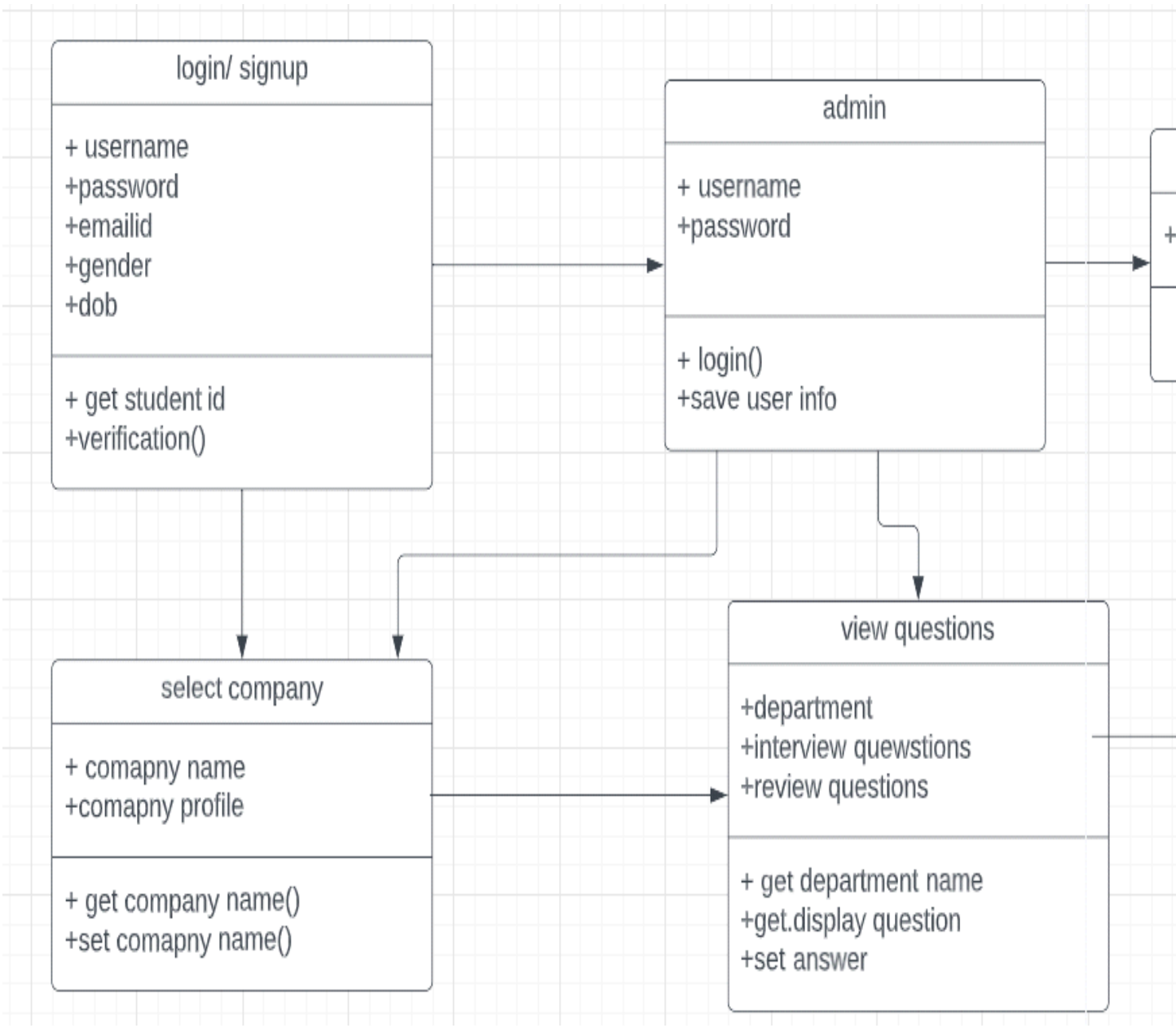
- **Use case Diagram**

Use-case diagrams describe the high-level functions and scope of a system. These diagrams also identify the interactions between the system and its actors. The use cases and actors in use-case diagrams describe what the system does and how the actors use it, but not how the system operates internally.



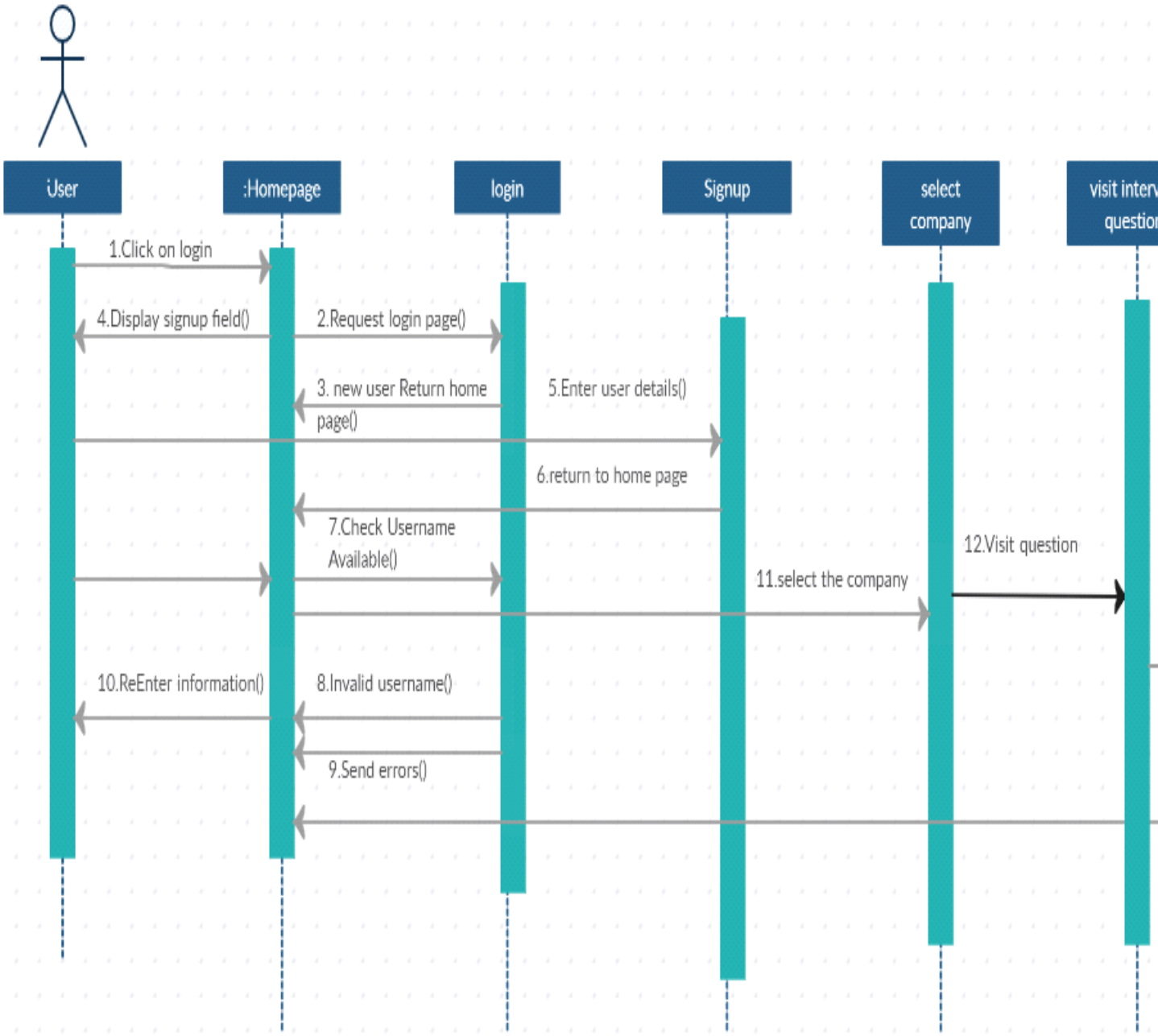
- **Class Diagram**

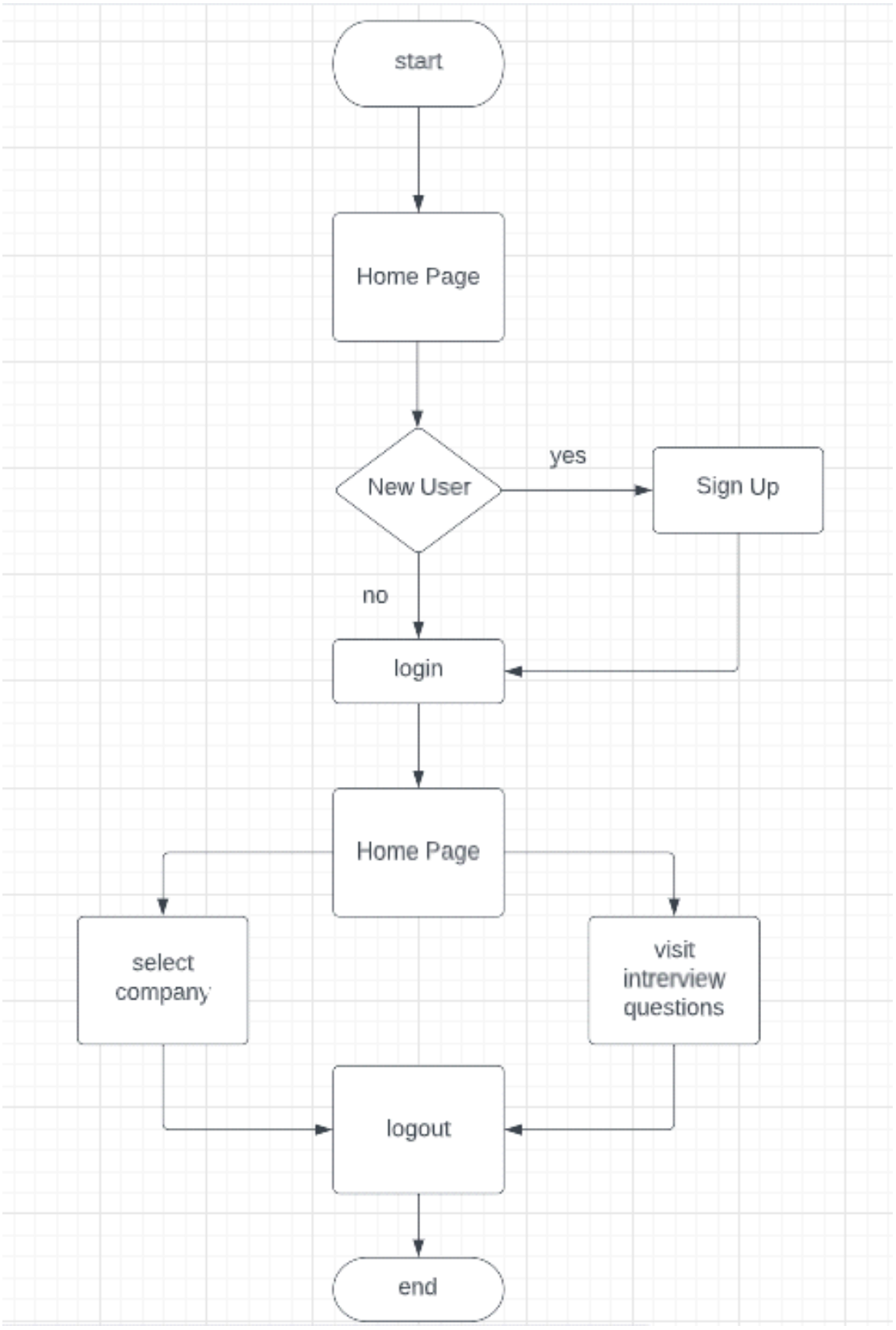
Class diagrams are the blueprints of your system or subsystem. You can use class diagrams to model the objects that make up the system, to display the relationships between the objects, and to describe what those objects do and the services that they provide. Class diagrams are useful in many stages of system design.



- **Sequence Diagram**

A sequence diagram is a Unified Modeling Language (UML) diagram that illustrates the sequence of messages between objects in an interaction. A sequence diagram consists of a group of objects that are represented by lifelines, and the messages that they exchange over time during the interaction.


User
:Homepage
login
Signup
select company
visit interv
questio
1.Click on login
4.Display signup field()
2.Request login page()
3. new user Return home page()
5.Enter user details()
6.return to home page
7.Check Username Available()
12.Visit question
11.select the company
10.ReEnter information()
8.Invalid username()
9.Send errors()

- 
start
Home Page
New User
yes
Sign Up
no
login
Home Page
select company
visit intrerview questions
logout
end

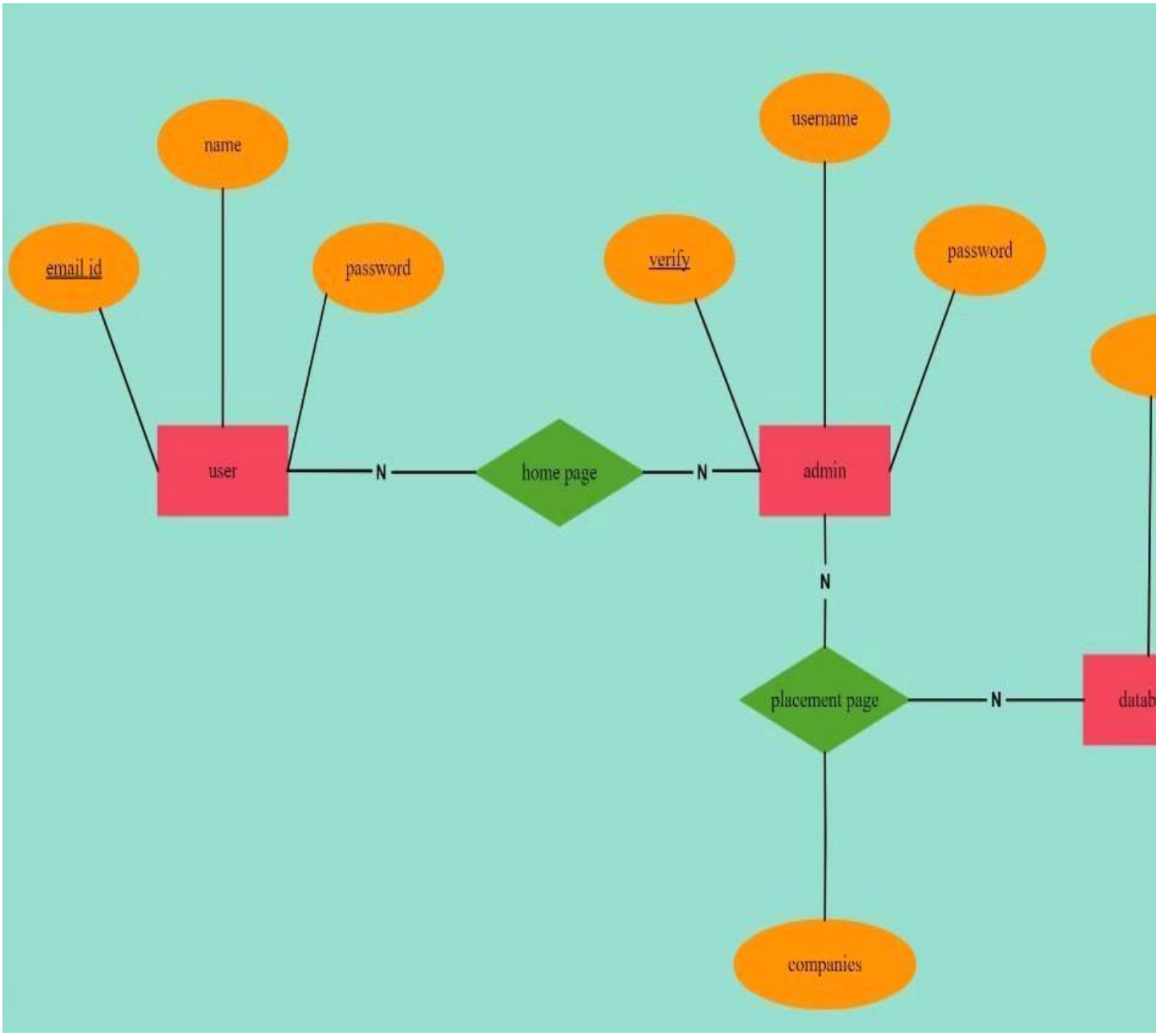
Flow Chart

A flowchart is a picture of the separate steps of a process in sequential order. It is a generic tool that can be adapted for a wide variety of purposes, and can be used to describe various processes, such as a manufacturing process, an administrative or service process, or a project plan.

+

- **Entity Relationship Diagram**

An entity relationship diagram (ERD), also known as an entity relationship model, is a graphical representation that depicts relationships among people, objects, places, concepts or events within an information technology (IT) system



- **Test Cases**

SECURITY TESTING

Testing is vital for the success of any software no system is ever perfect. Testing is also carried in two phases .First phase is during the software engineering that is during the module creation. Second phase is after the completion of software this is system testing which verifies that the whole set of programs hanged together.

WHITE BOX TESTING:

This type of testing technique deals with testing the internal structure, logic design, and implementation of different modules. Here, the tester uses his / her preferred input or exercises paths via code to determine the proper or exact output. As it is also termed as codeoriented testing, it contains technical tests and script-based testing as part of its testing phase.

BLACK BOX TESTING:

This testing is also known as Behavioral Testing. The software tests the internal structural, design, and implementation and UI and UX of the product being tested, which is unknown to the tester. Black box testing is both functional or non-functional, but most of the time, it is usually functional.

This testing technique is named black box because the software or the product is not known or acknowledged in advance to the tester, and hence you can say the tester's eye is blind-folded like a black box, and you can see nothing inside. This technique of testing tries to find errors.

ALPHA TESTING:

Alpha Testing can be defined as a form of acceptance testing carried out to identify various types of issues or bugs before publishing the build or executable of the software public or market. This test type focuses on real users through black box and white box testing techniques.

BETA TESTING:

Beta Testing can be defined as the second stage of testing any product before release, where a sample of the released product with minimum features and characteristics is being given to the intended audience for trying out or temporarily using the product.

UNIT TESTING:

In this testing level, individual sections or parts of software or product are tested. The idea of this is to confirm every part or unit of the product after the test.

INTEGRATION TESTING:

In this software testing level, individual parts need to combine and test as a single cluster. This testing level's main idea is for exposing the faults while interacting between integrated units of the project.

VALIDATION TESTING:

Validation testing was performed to ensure that all the functional and perfromance requirement are met.

SYSTEM TESTING:

In this software testing level, the whole, integrated software or project is tested. The principle for this testing is to assess the system's conformity with its intended requirements.

THE STEPS IN THE SOFTWARE TESTING

- Preparation of the test cases.
- Preparation of the possible test data with all the validation checks.
- Complete code review of the module.
- Actual testing done manually.
- Modifications done for the error found during testing.
- Prepared the test result scripts.