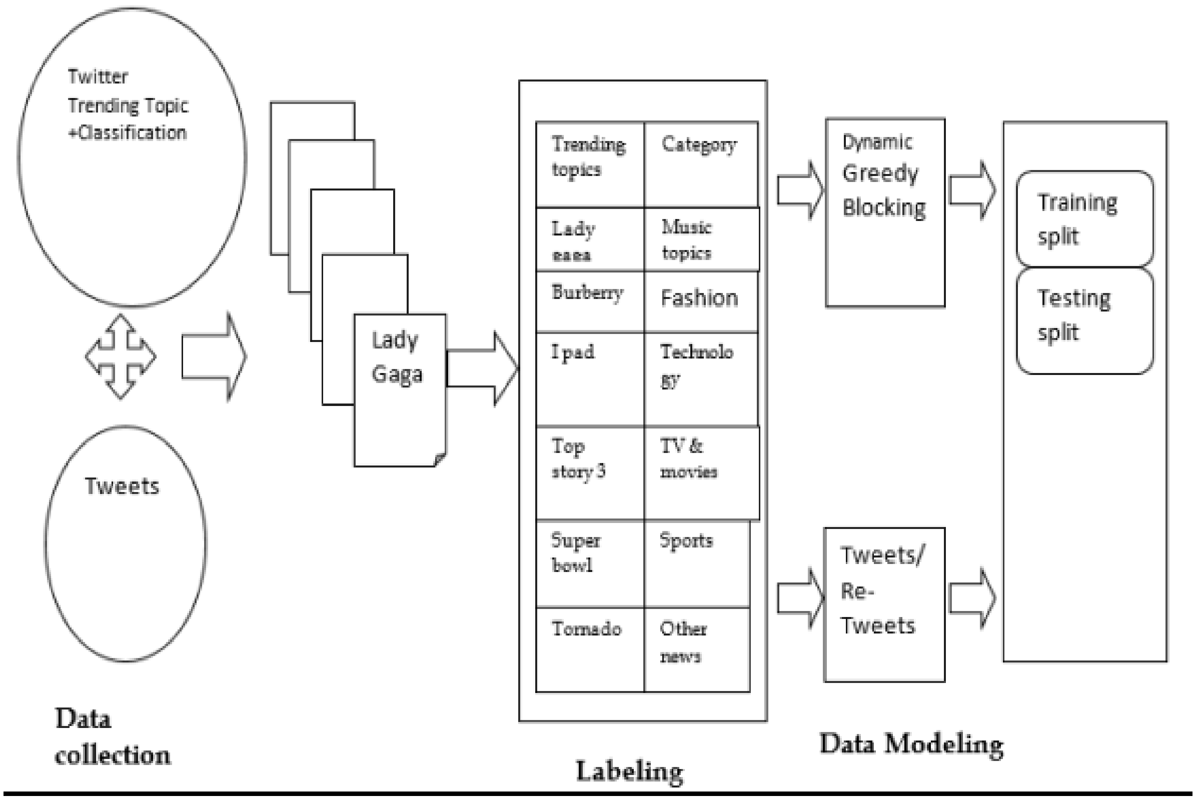
SOFTWARE DESIGN

5.1 SYSTEM ARCHITECTURE

System architecture design refers to the process of defining and designing the structure, components, modules, interfaces, and relationships of a system to fulfill specified requirements. It encompasses creating a blueprint or high-level plan that outlines how different parts of a system.



Trending topics	Category
Lady eaea	Music topics
Burberry	Fashion
I pad	Technology
Top story 3	TV & movies
Super bowl	Sports
Tornado	Other news

Fig 5.1 ARCHITECTURE DIAGRAM

5.2 DATA FLOW DIAGRAM

- The DFD is also called as bubble chart. It is a simple graphical formalism that can be used to represent a system in terms of input data to the system, various processing carried out on this data, and the output data is generated by this system.
- The data flow diagram (DFD) is one of the most important modeling tools. It is used to model the system components. These components are the system process, the data used by the process, an external entity that interacts with the system and the information flows in the system.
- DFD shows how the information moves through the system and how it is modified by a series of transformations. It is a graphical technique that depicts information flow and the transformations that are applied as data moves from input to output.
- DFD is also known as bubble chart. A DFD may be used to represent a system at any level of abstraction. DFD may be partitioned into levels that represent increasing information flow and functional detail.

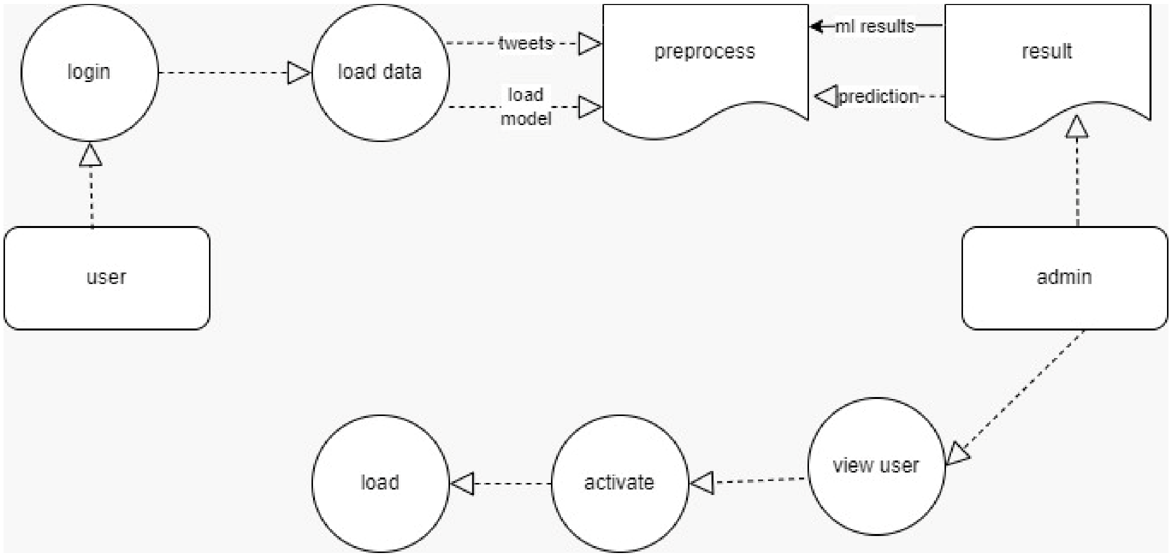


Fig 5.2 DATA FLOW DIAGRAM

5.3 UML DIAGRAMS

UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group.

The goal is for UML to become a common language for creating models of object-oriented computer software. In its current form UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML.

The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems.The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems.

The UML is a very important part of developing objects-oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

5.3.1 GOALS

The Primary goals in the design of the UML are as follows:

1. Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.
2. Provide extendibility and specialization mechanisms to extend the core concepts.
3. Be independent of particular programming languages and development process.
4. Provide a formal basis for understanding the modeling language.
5. Encourage the growth of OO tools market.
6. Support higher level development concepts such as collaborations, frameworks, patterns and components.
7. Integrate best practices.

5.3.2 USE CASE DIAGRAM

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

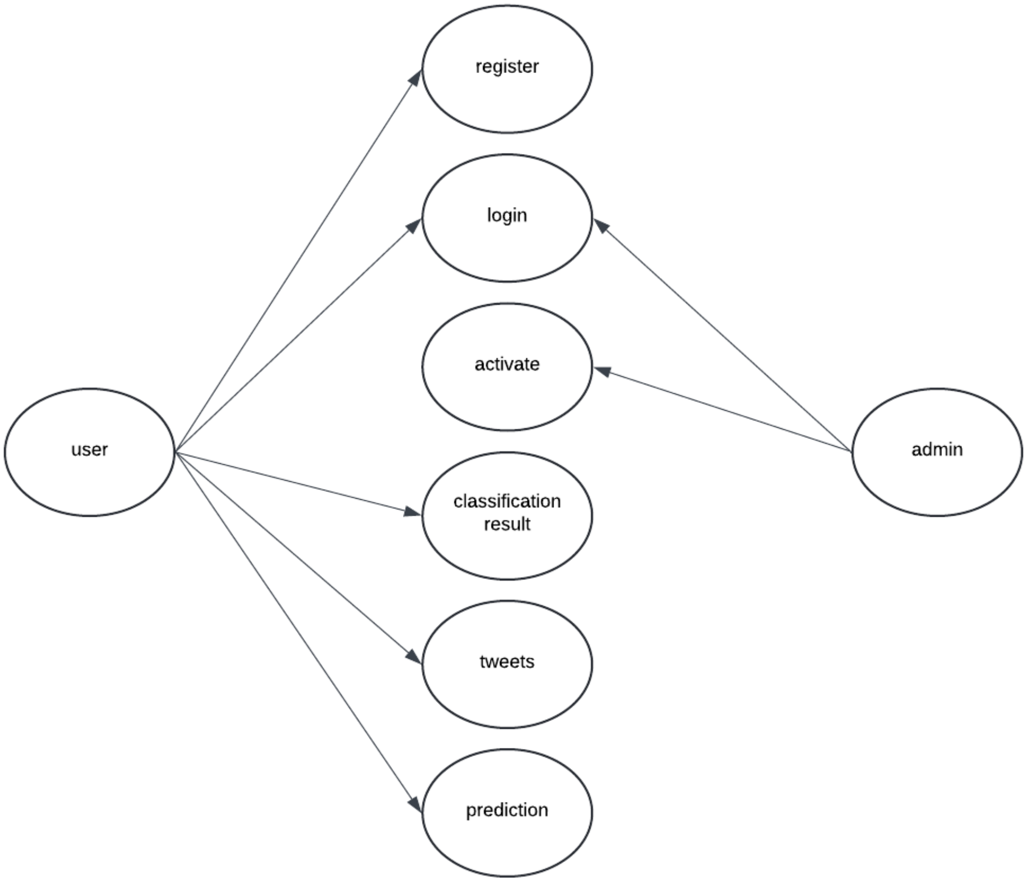


Fig 5.3.2 USECASE DIAGRAM

5.3.3 CLASS DIAGRAM

In software engineering, 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 the classes. It explains which class contains information.

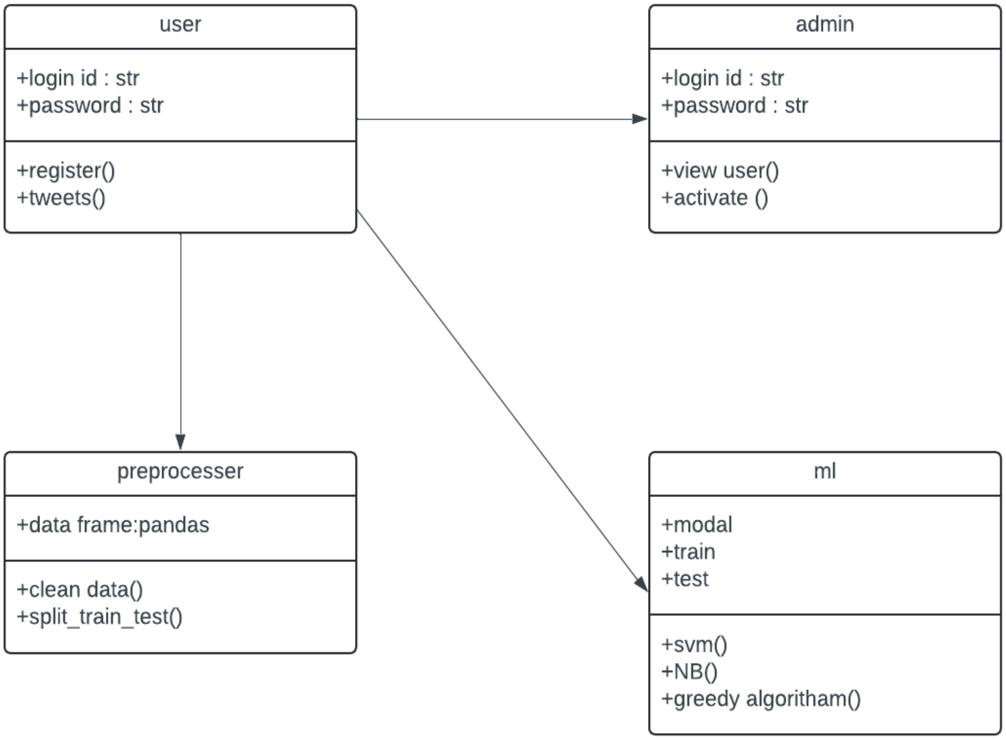


Fig 5.3.3 CLASS DIAGRAM

5.3.4 SEQUENCE DIAGRAM

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.

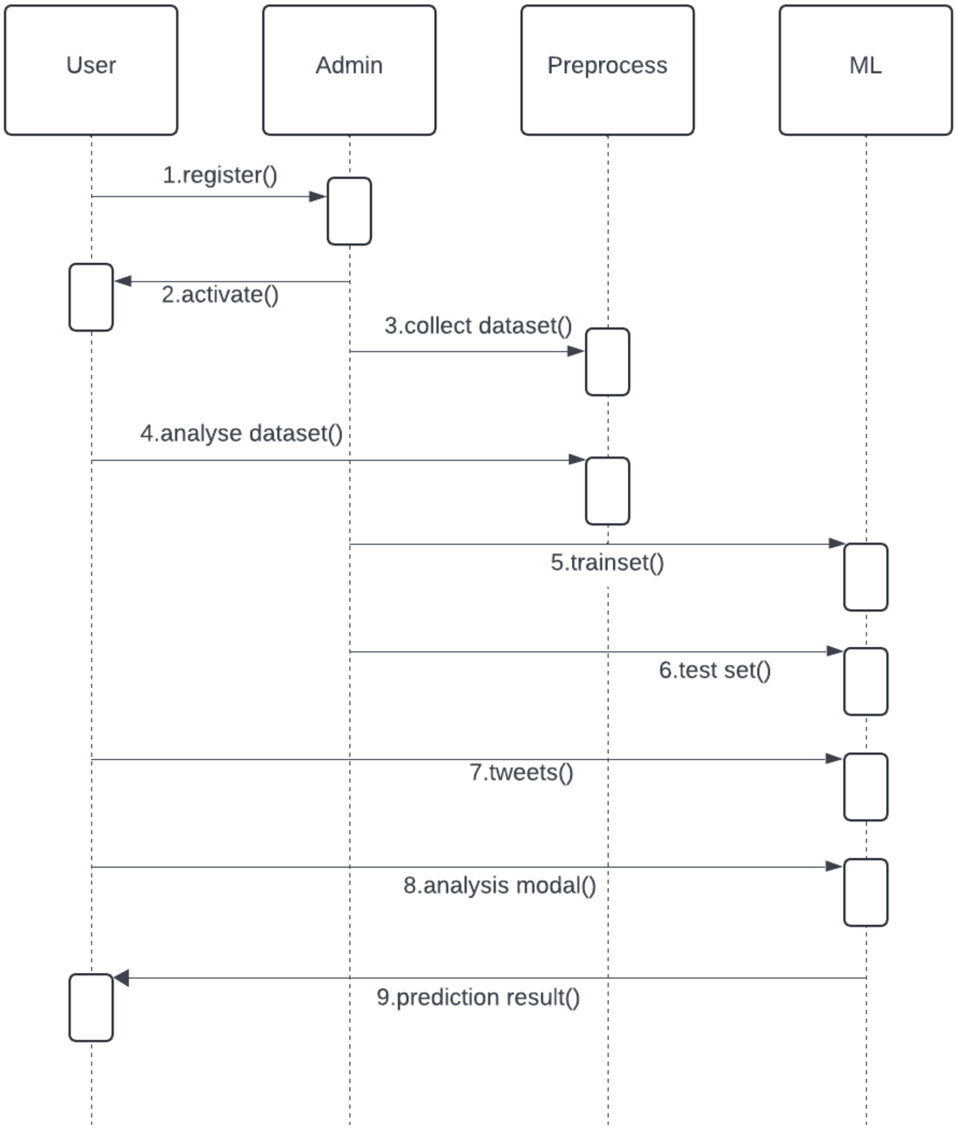


Fig 5.3.4 SEQUENCE DIAGRAM

5.3.5 ACTIVITY DIAGRAMS

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-bystep workflows of components in a system. An activity diagram shows the overall flow of control.

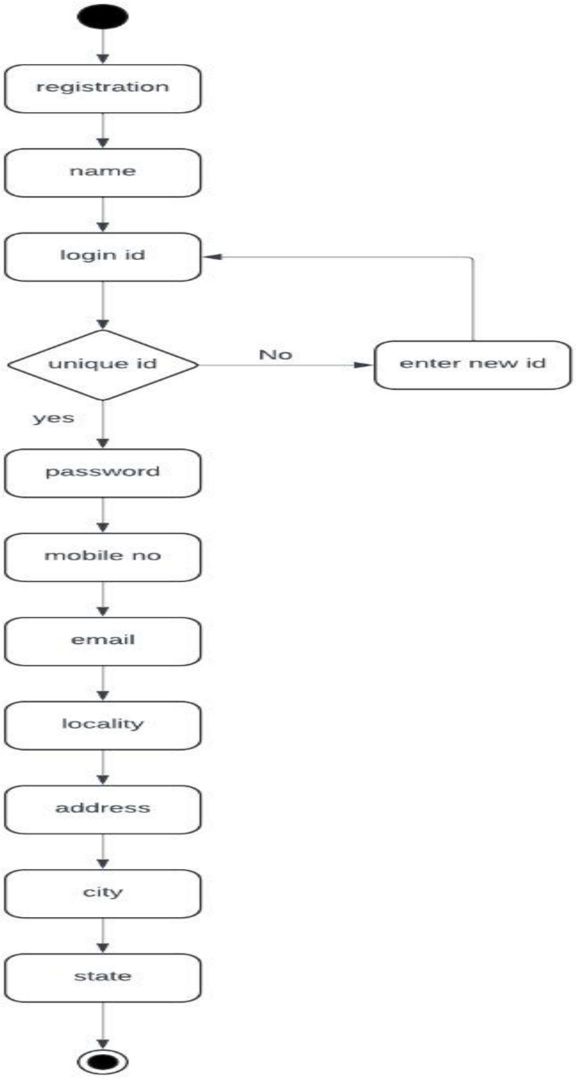


Fig 5.3.5.1 ACTIVITY DIAGRAM FOR USER REGISTRATION PASSWORD

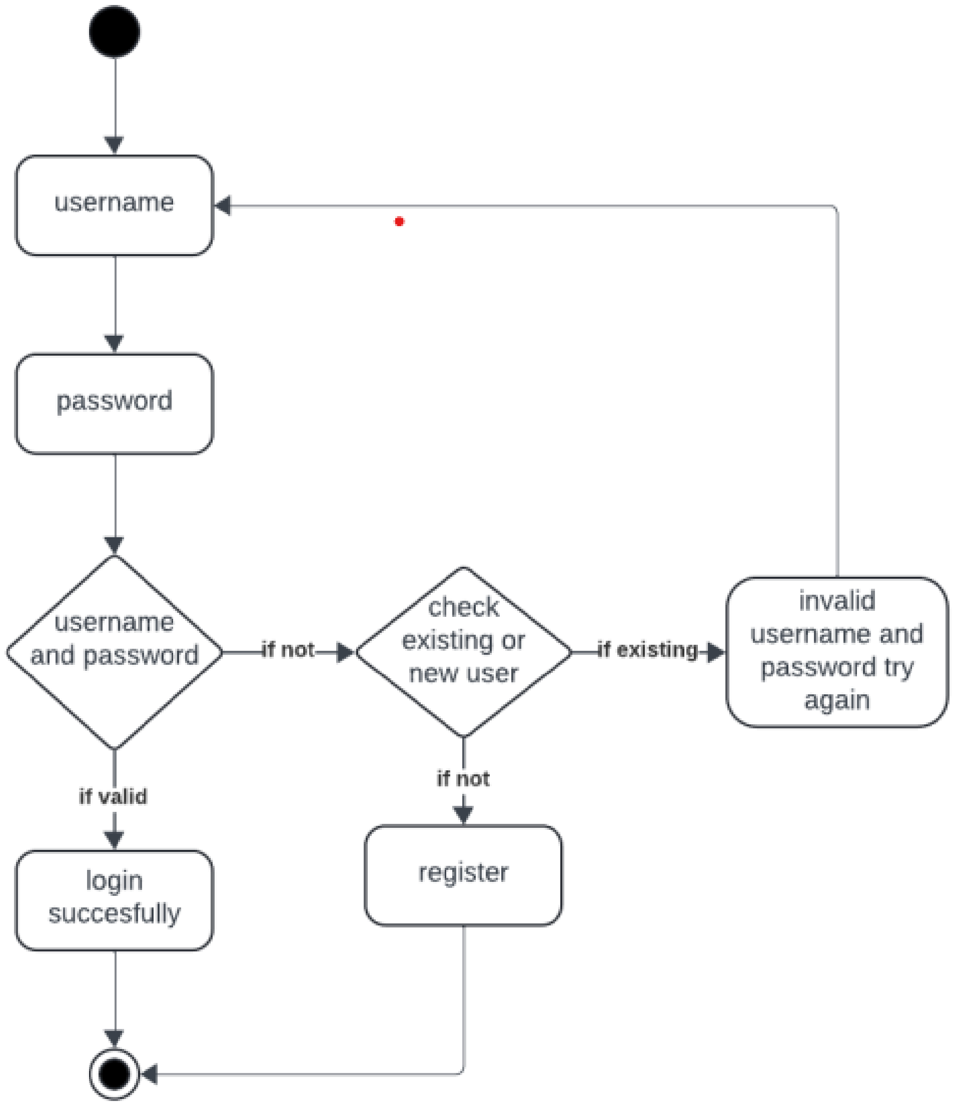


Fig 5.3.5.2 ACTIVITY DIAGRAM OF USER LOGIN

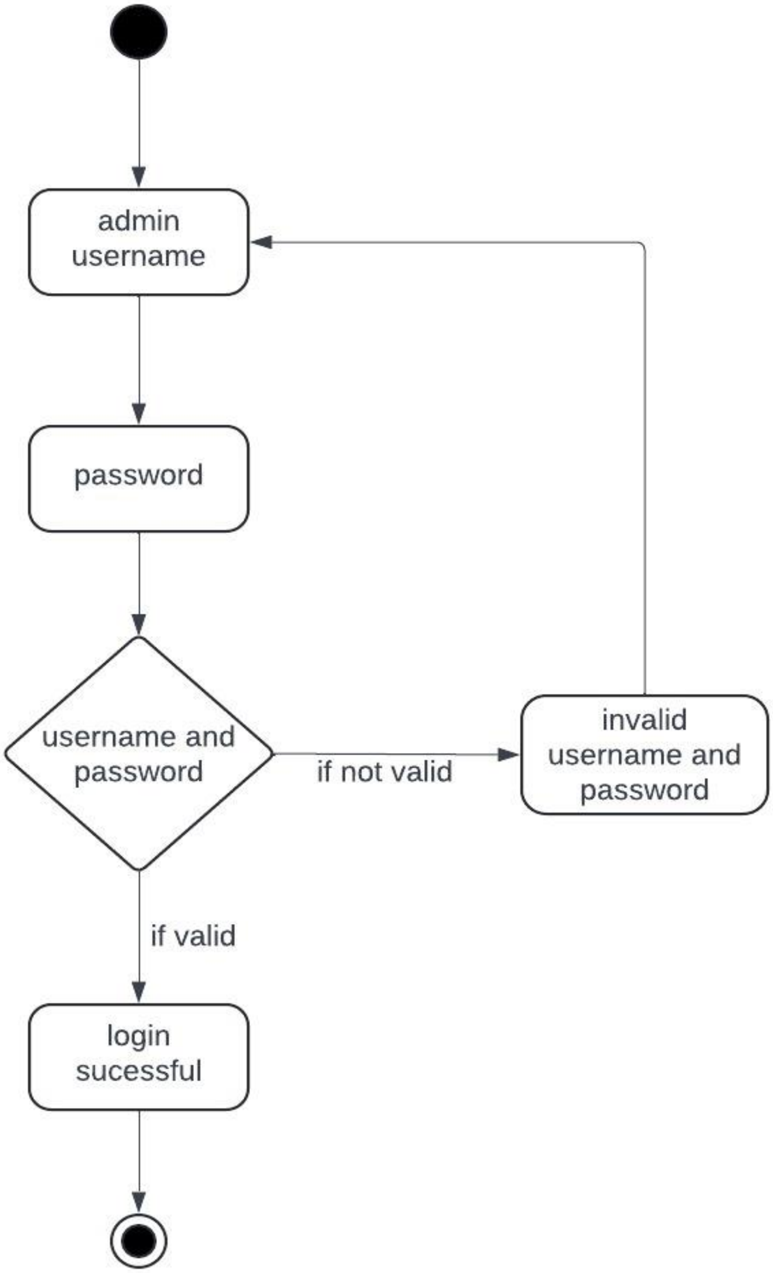


Fig 5.3.5.3 **ACTIVITY DIAGRAM OF ADMIN LOGIN**

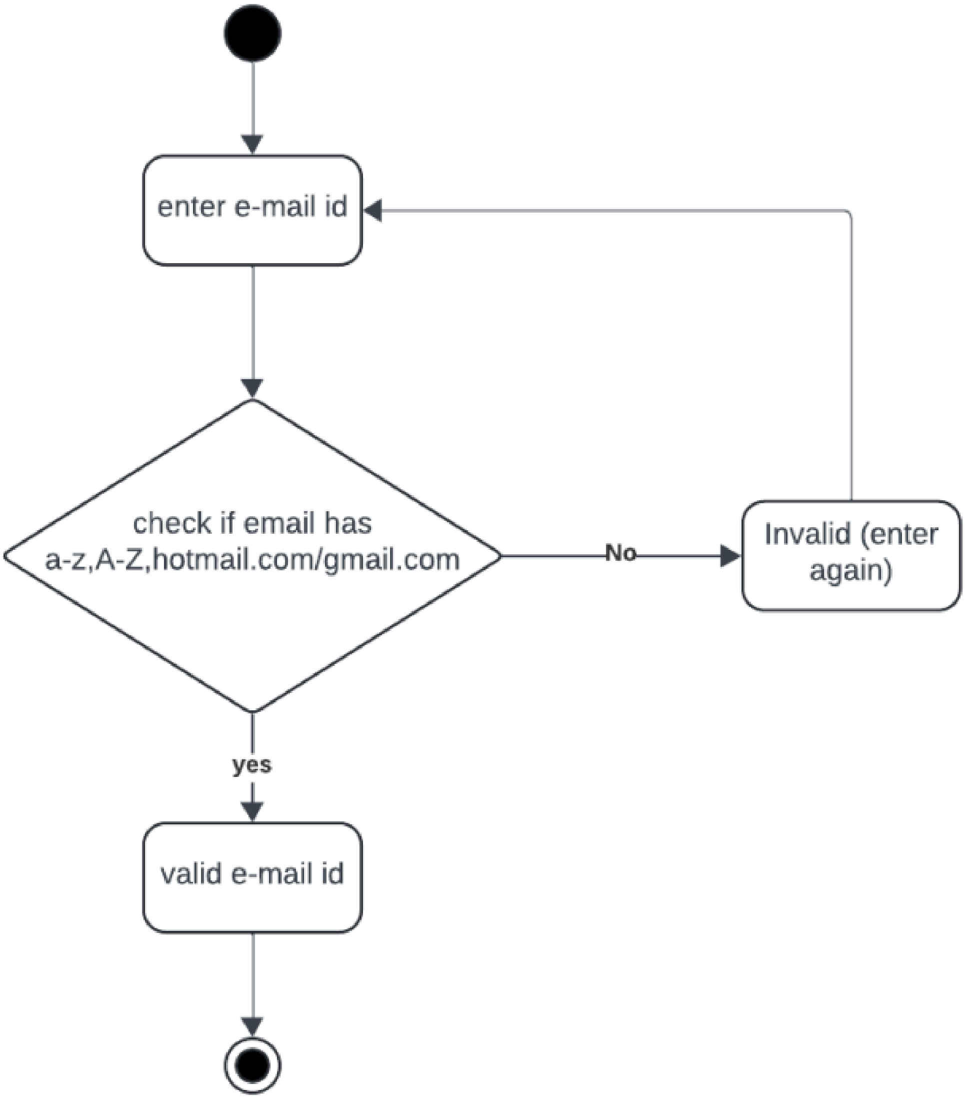


Fig 5.3.5.4 **ACTIVITY DIAGRAM OF EMAIL**

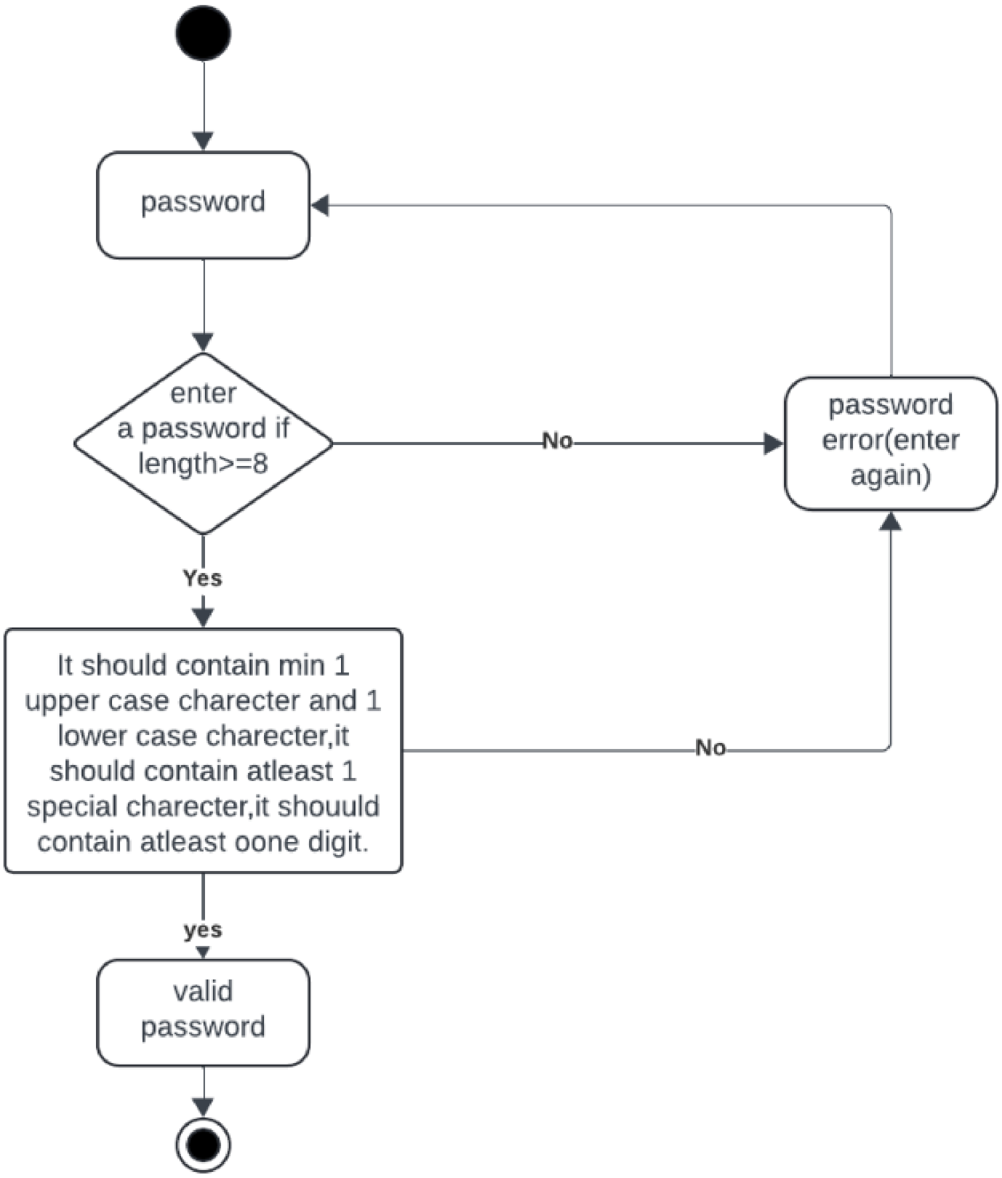


Fig 5.3.5.5 ACTIVITY DIAGRAM OF PASSWORD

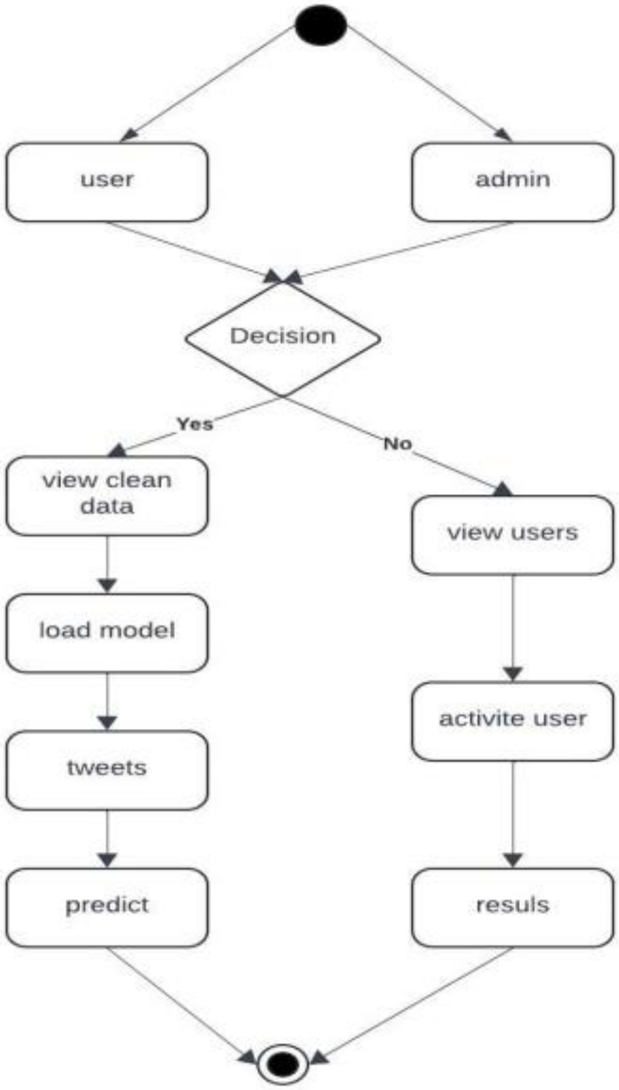
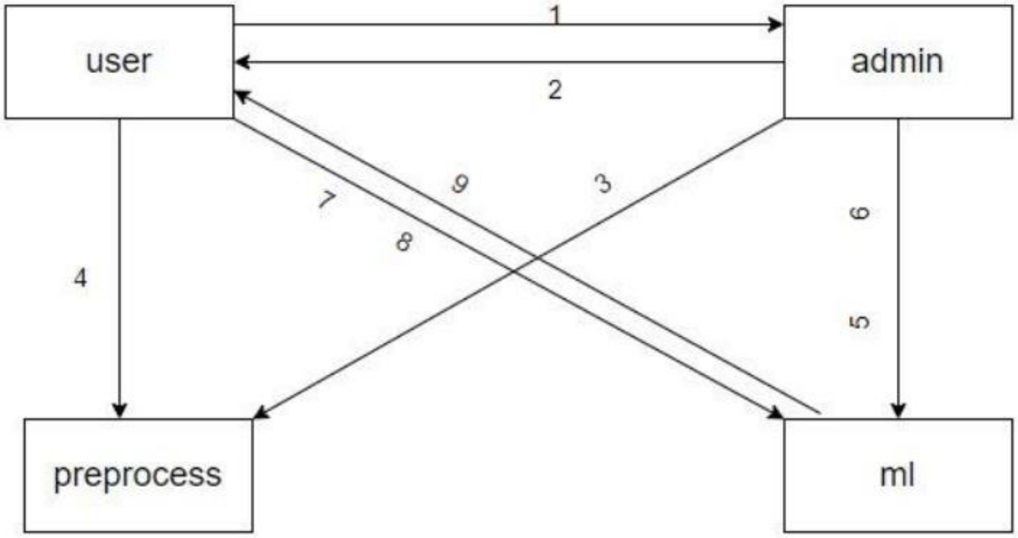


Fig 5.3.5.6 ACTIVITY DIAGRAM FOR OVERALL WORKFLOW OF THE SYSTEM

5.3.6 COLLABORATION DIAGRAM

A collaboration diagram, also known as a communication diagram, is a type of interaction diagram in Unified Modeling Language (UML) that illustrates the interactions and relationships among objects or components within a system. In a collaboration diagram, objects are represented as rectangles, and the interactions between them are depicted by labeled arrows indicating the messages passed between the objects. These messages can represent method calls, signals, or any form of communication between objects. The diagram emphasizes the dynamic behavior of the system by showing how objects collaborate to achieve a certain functionality or perform a specific task. It helps in visualizing the flow of control and communication among the objects during run time.



1.Register

2.Activate

3.Collect data

4.Analyse Data set

5.Train set

6.Test set

7.Tweets

8.Analysis Modal

9.Prediction Result

Fig 5.3.6 COLLABORATION DIAGRAM