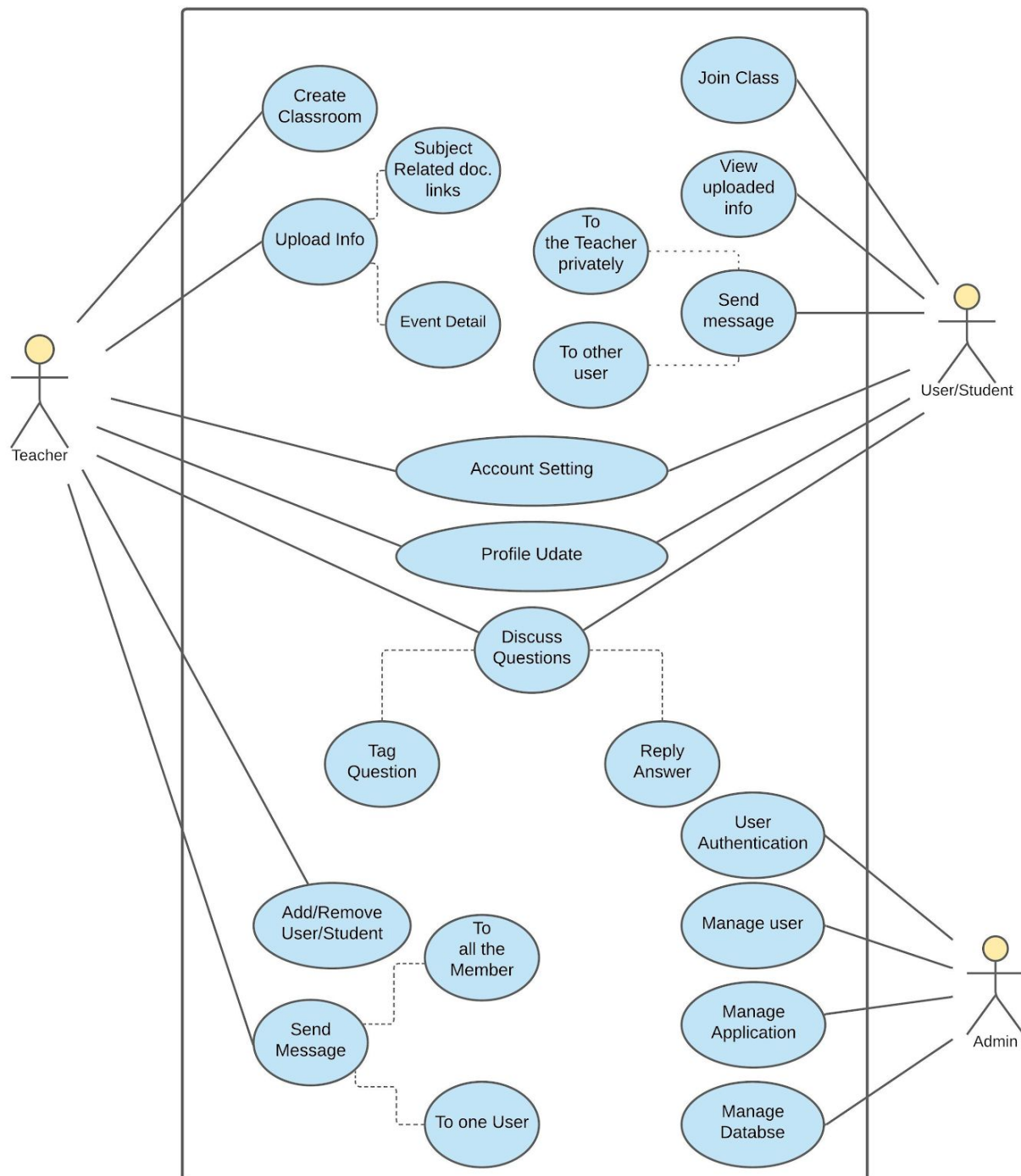


Architecture Decisions Record - ADR

Use-Case Diagram



Status

Proposed

Context

There is a need for having three actors, though one of them is an admin. But we still require two different users unlike other apps where there is only one user and in case of a group he is given some extra functionality. The need for two varieties of users arises due to the creation of CLASSROOM where the creator of the class has more functionality than an ordinary group admin. This has to be done to create a different module where the Class Owner can generate Polls apart from the general discussion box or chat box and also He/She can see the whole record of Polls. It would be difficult to manage all the users with dual functionality and creating another type of dedicated user would be easy.

Decision

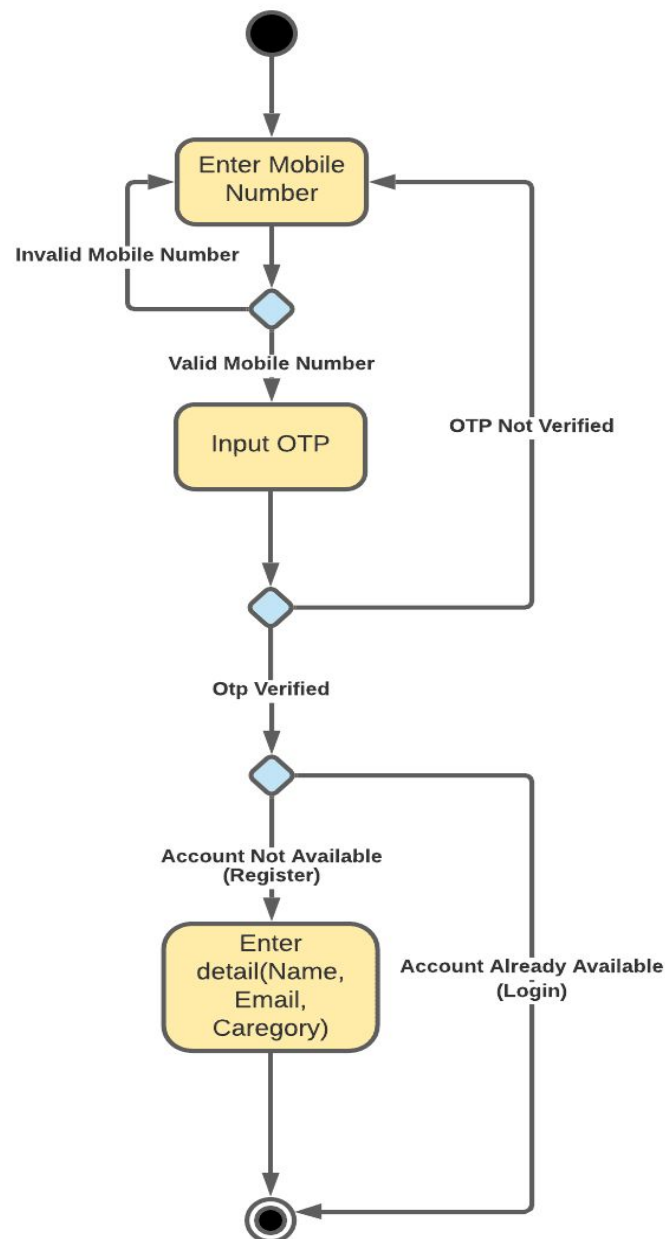
A total of three actors would be used out of which one is an admin and the other two are the different types of user.

Consequences

Adding two users instead of one general user with added functionality would certainly increase the time and space complexity of the application but on the other hand, it will surely ease the logical issues and make it much easier to implement.

Activity Diagram

Registration/Login



Status

Rejected

Context

We first decided to determine the user type early in the Registration/Login Activity so that there would be two parallel flow of activity that also could be forwarded in upcoming activities.

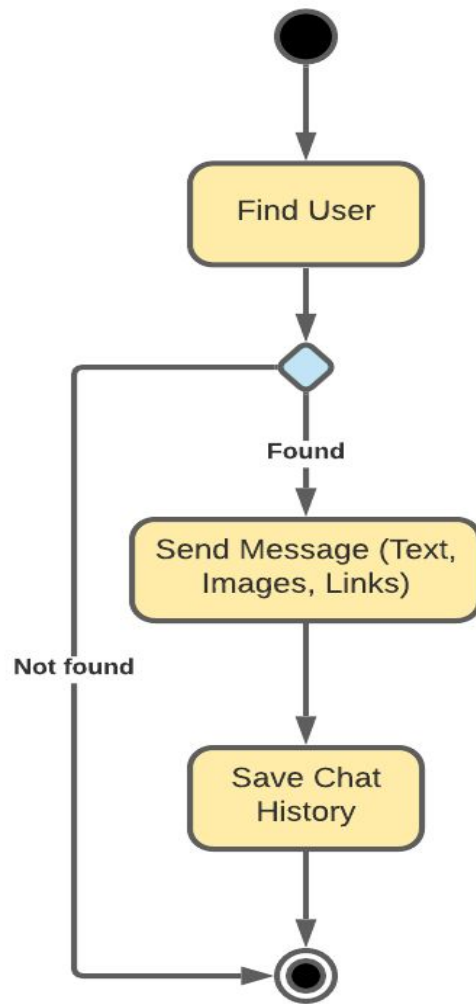
Decision

The user type is determined late in the Registration/Login Process.

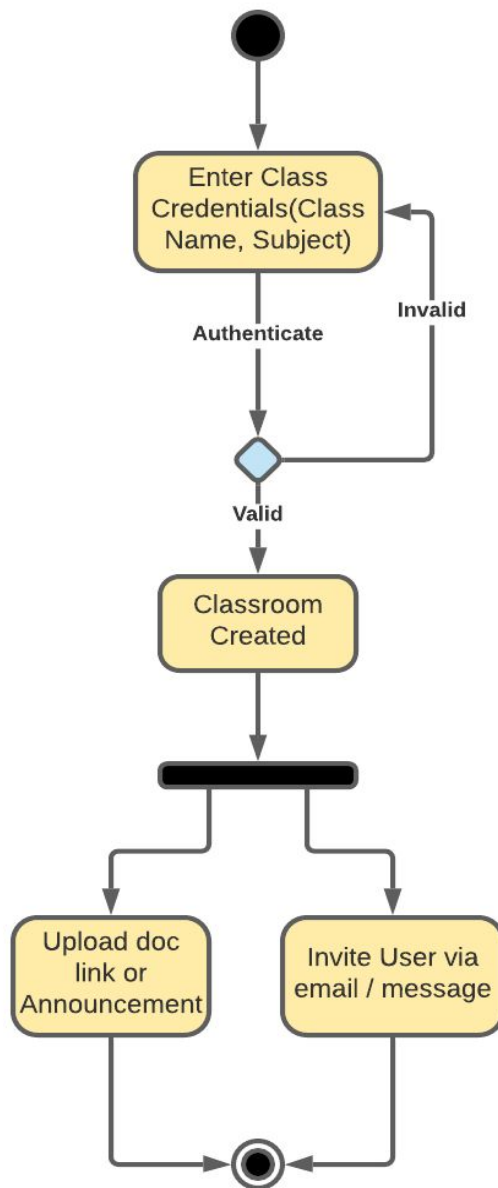
Consequences

Though defining the user type early has it's advantages but would have made the first process very complex. As an effort to lower down the complexity, we will first authenticate the user through OTP and then later make them choose their user type.

Personal Chat



Create Classroom



Status

Proposed

Context

Every Classroom should have it's random ID and a unique name for which the app would even give a suggestions.

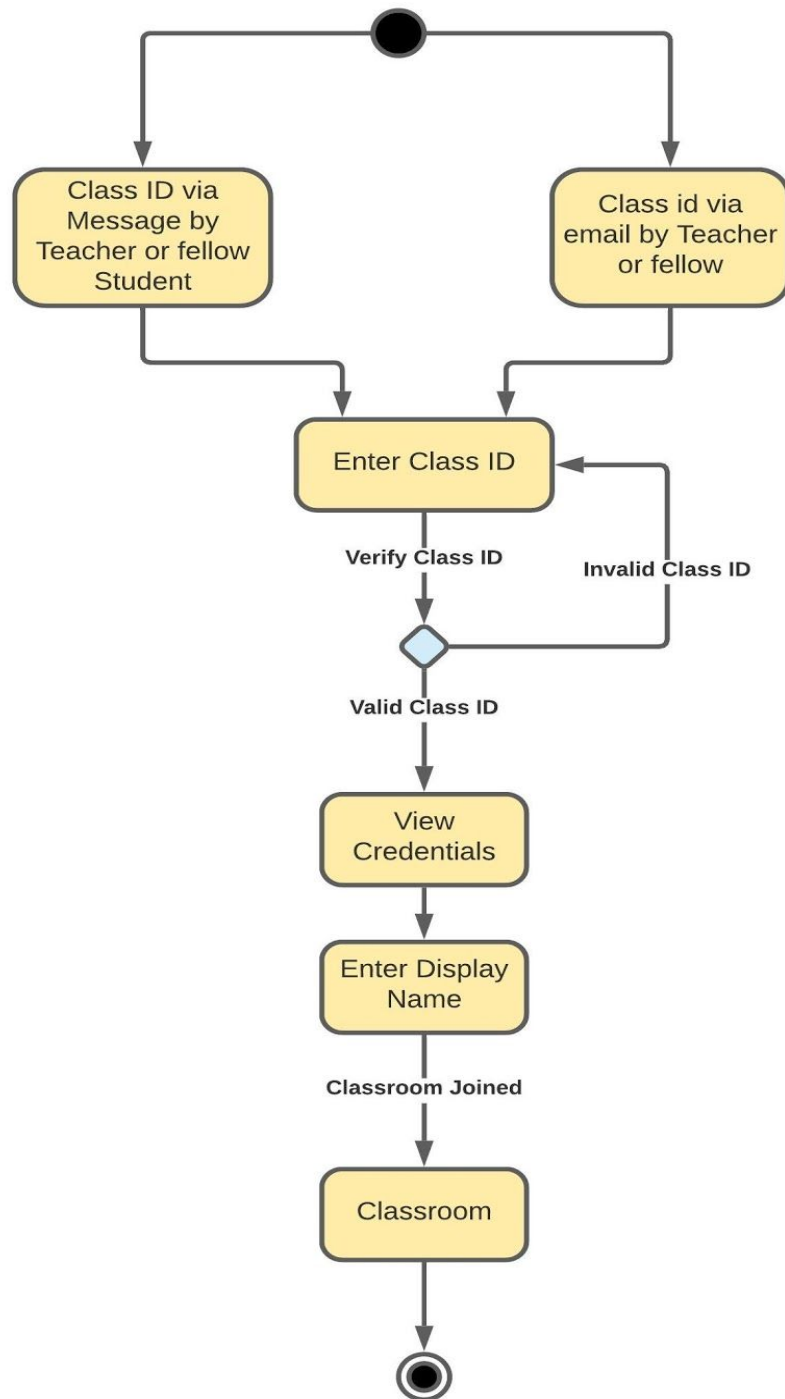
Decision

Every classroom will have it's random ID and a unique name.

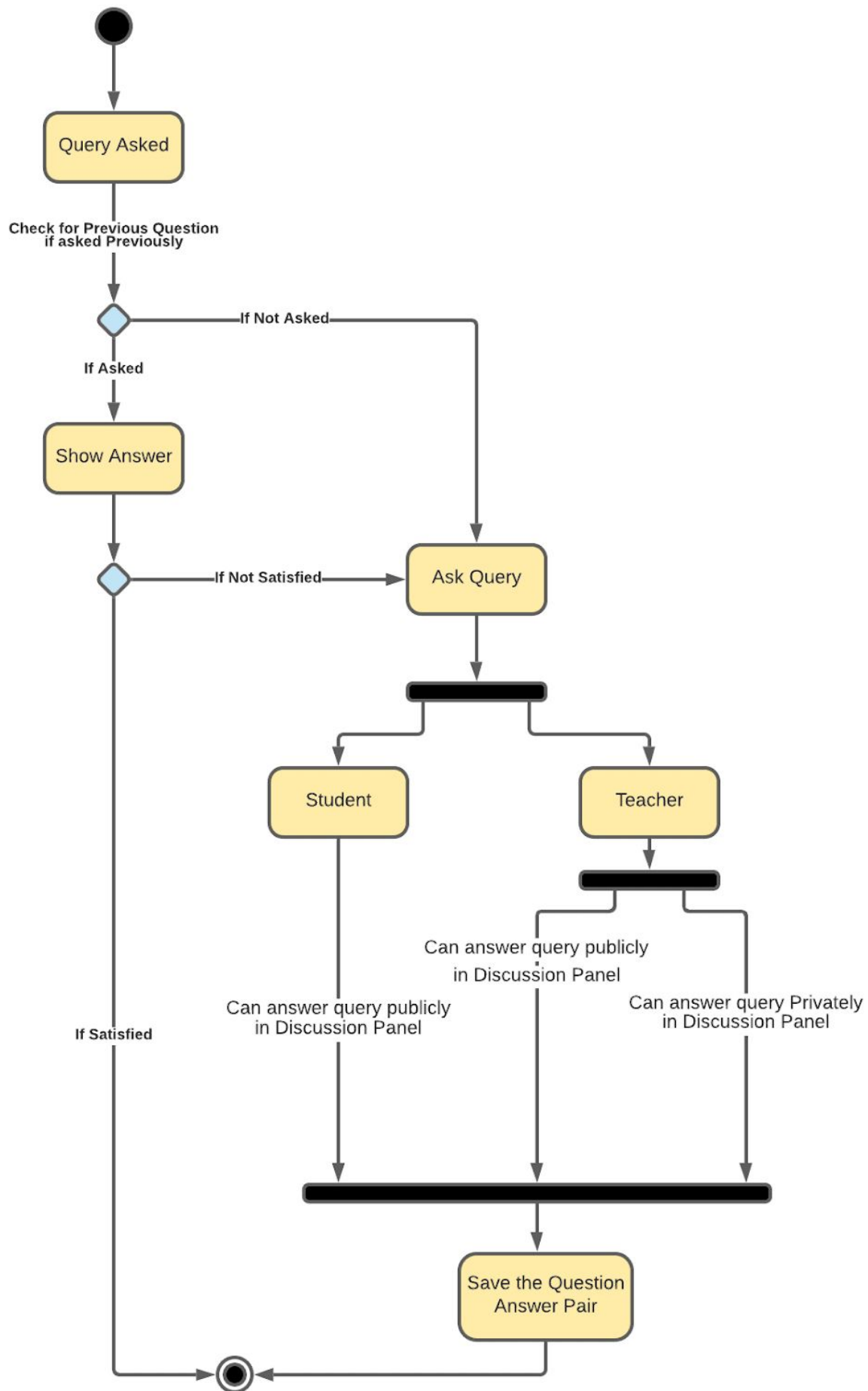
Consequences

Having a unique name for the Classroom will help in searching and identifying the Classroom much faster.

Join Classroom



Discussion



Status

Accepted

Context

Creating and joining a classroom is one of the major functions of our apps. As a way to ease the joining process, at least, two modes of joining should be there like one from a direct message link in the app itself and the other through an email link.

Decision

Two modes of joining a classroom will be provided.

Consequence

Though having two modes of joining a classroom will increase the burden on the developing team, it will make the whole activity very dynamic and easy.

Status

Proposed

Context

Members in a chat group seldom see the display name of every other participant, on the contrary they only see those whose contacts are saved and the rest participants are shown just as random mobile numbers.

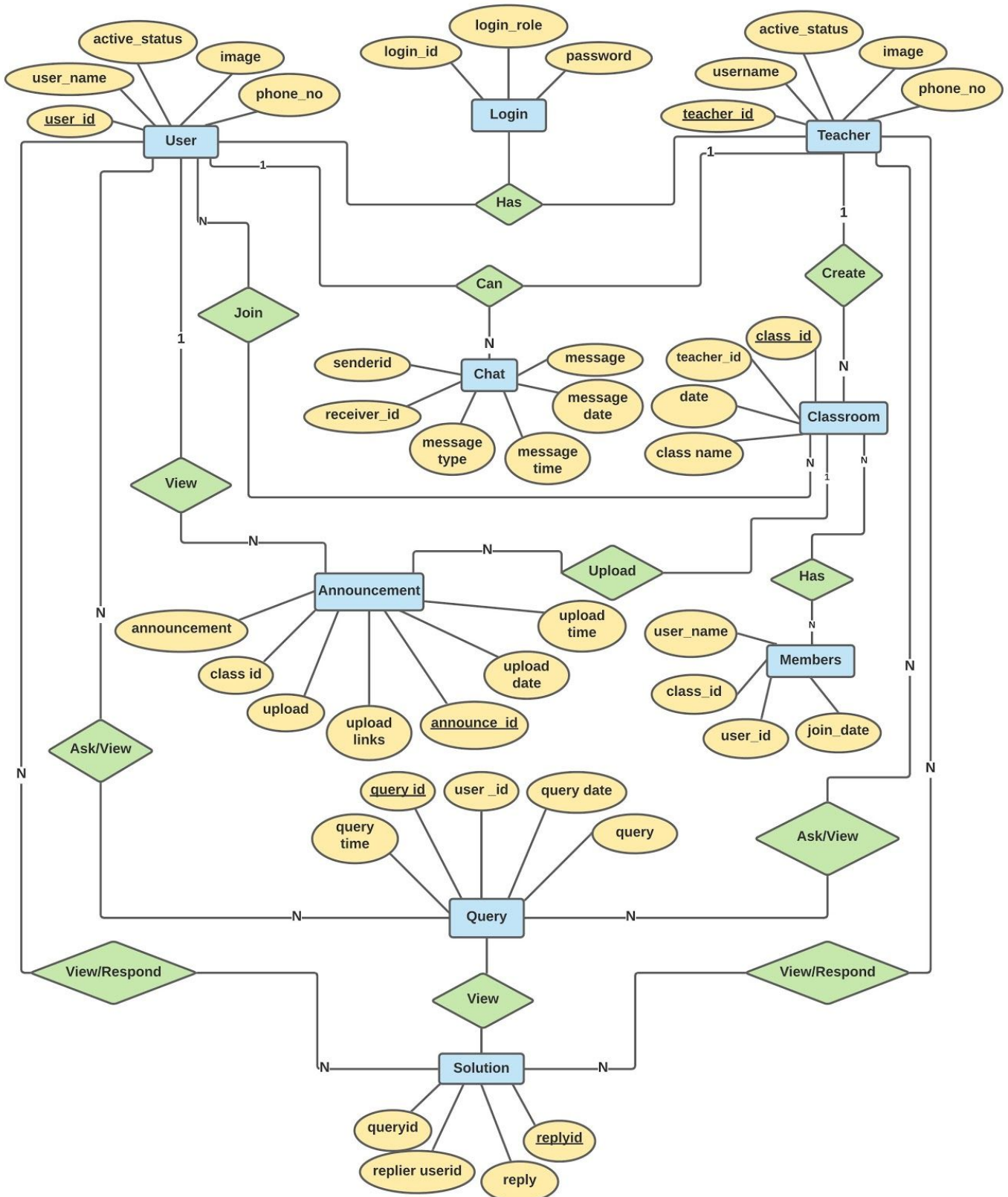
Decision

Members of a classroom will have a display name compulsorily, whose format can be pre-decided by the teacher itself.

Consequences

Though having an extra display name would certainly increase the space and logical complexity in the database but it will also add more meaning to the classroom because, in an environment like a class, students should know each other and not just be replying to random phone numbers.

E-R Diagram



Status

Proposed

Context

Databases are an integral part of any live real project. We have decided to use a NoSQL database. The database will be Firebase realtime database and Cloud Fire Store.

Decision

A NoSQL real-time database is used for data storage and manipulation.

Consequence

Where relational databases require data to be put into tables and columns to be accessed and analyzed, the various data model capabilities of NoSQL databases make them extremely flexible when it comes to handling data.

Status

Proposed

Context

Login should be created as a separate entity as it will help in maintaining the two different type of users and would help determine the role of a user early in the process.

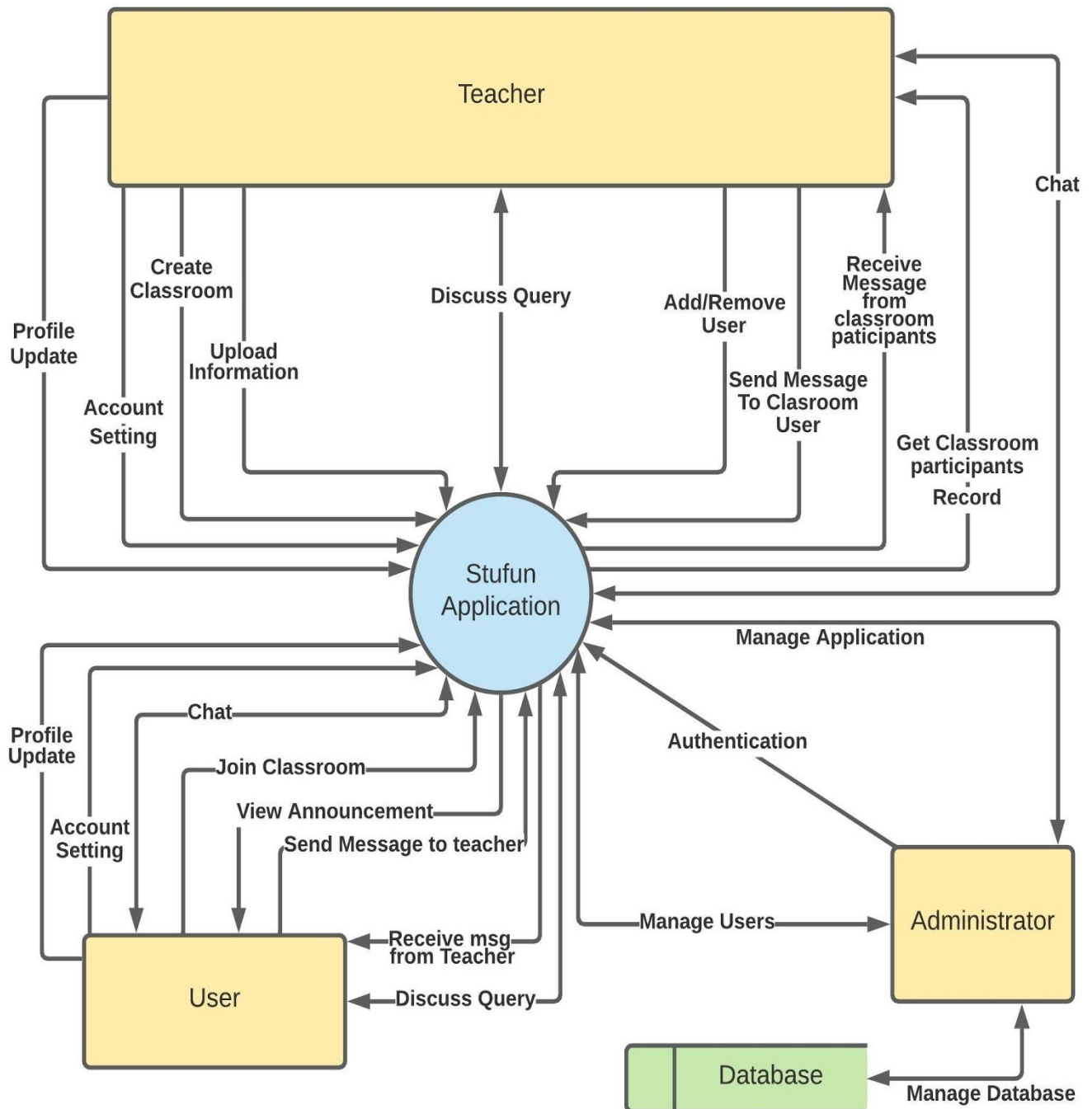
Decision

Login has been created as a separate entity.

Consequences

Using login as an attribute would disrupt the sense and flow of normalisation, also using it as an entity would help to determine user's role and also their password data would be separated from their main data table.

Data Flow Diagram



Status

Proposed

Context

A level 0 Data Flow Diagram is to be made. The STUFUN application is the main process and all other external entities like Teacher, User and admin will be linked to it therefore it should be the central process.

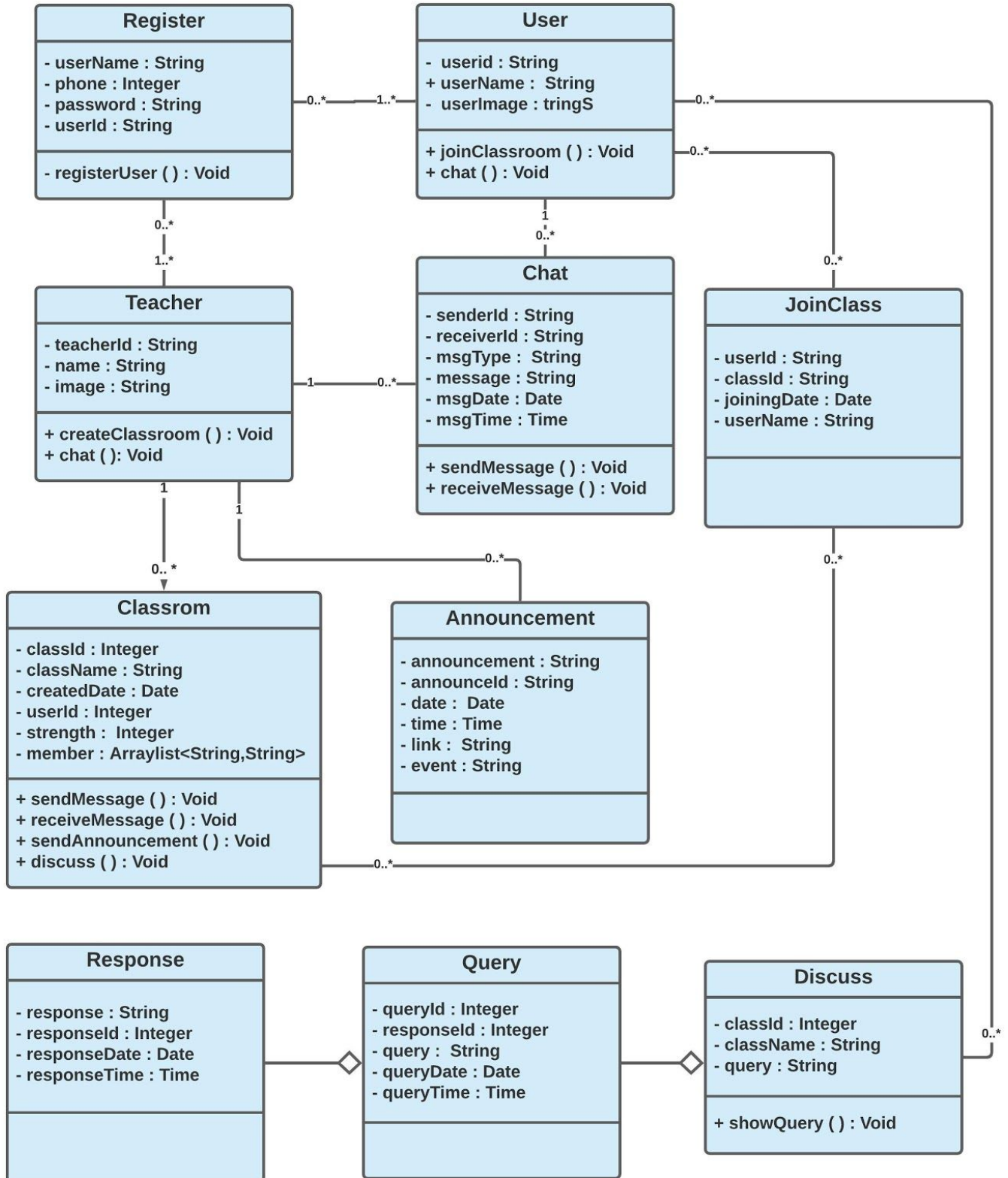
Decision

The STUFUN application is the main and only central process and the whole flow of data will be around this central process.

Consequence

Having a centralised process makes it easier to understand the whole process easy and helps in connecting the various entities easily and also give a better sense of data flow.

Class Diagram



Status

Accepted

Context

All the attributes of all the classes are kept private except for the username that will be shown to the primary user and secondary user.

Decision

The username is kept public

Consequence

Keeping the username public would be the first step towards a chat application as we can see the name of a user and when we communicate with him/her. Rest all attributes are kept private to keep them safe from foreign intrusion

Status

Accepted

Context

The response, Query and discussion should be having an aggregation as they combined wholly give more important service rather than their functionality.

Decision

The response, Query and Discussion are joined together as an aggregation and not as a composition.

Consequences

After changing Response, Query and Discussion classes as aggregation they entirely serve as a combined function, if there had been a composition between them we would have to work with them as individual functions and then they wouldn't be a part of the chat and they would be incorporated from the outside like a separate entity.

