Design:

# Normalization:

Normalization is a technique used in database design to organize tables in a way that reduces redundancy and dependency. It is a process of organizing the columns and tables of a database to minimize data redundancy and improve data integrity. This is done by dividing larger tables into smaller, more focused tables and defining relationships between them. Normalization typically involves dividing a database into two or more tables and defining relationships between the tables. The goal of normalization is to produce database tables that are efficient to maintain and update.

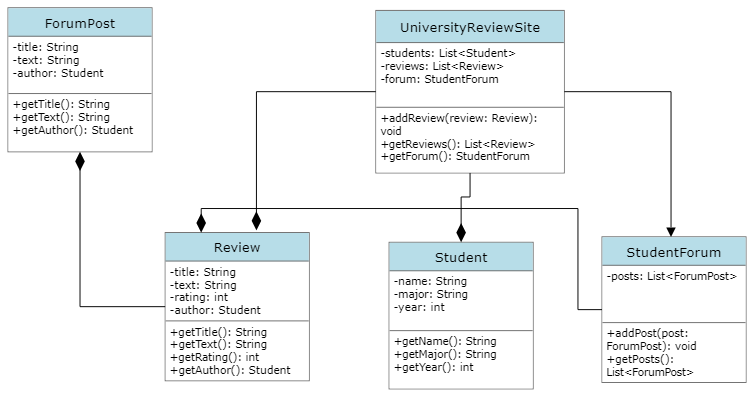
Normalization can improve our database management system (DBMS) in several ways. First, it can reduce data redundancy, which means that the same data is not stored multiple times in the database. This can help to save storage space and improve the efficiency of the database. Second, it can reduce the likelihood of data inconsistencies, which can occur when the same data is stored in multiple places and is not updated consistently. This can help to ensure that the data in the database is accurate and up-to-date. Third, normalization can improve the performance of database queries by reducing the amount of data that needs to be processed. Finally, normalization can make it easier to modify the structure of the database as the needs of the system change over time.

There are several guidelines for normalizing a database. These include the following:

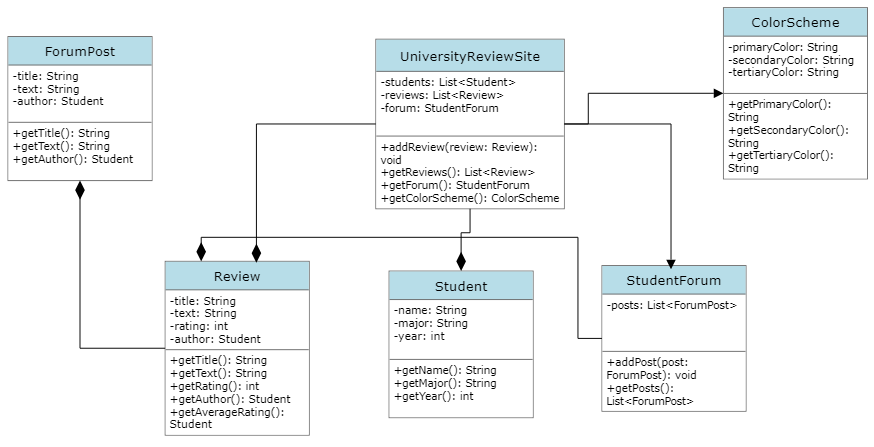
1. Divide the database into smaller, more focused tables. Each table should contain data about a single subject, such as customers or orders.
2. Define relationships between the tables. For example, a customer table might have a relationship with an orders table, indicating that each order is associated with a specific customer.
3. Remove redundant data from the tables. For example, if the same data is stored in multiple tables, it should be consolidated into a single table.
4. Use primary and foreign keys to identify unique records and establish relationships between tables.
5. Normalize the data to the third normal form (3NF) or higher. This involves applying a series of rules to the tables to ensure that they are fully normalized and do not contain any redundant data.
6. Test the database to ensure that it is working properly and that queries are being executed efficiently.
7. Regularly review and update the database to ensure that it continues to meet the needs of the system.

# UML Diagram:

This is the final version of our UML diagram which details the design of our system:



The final UML diagram:

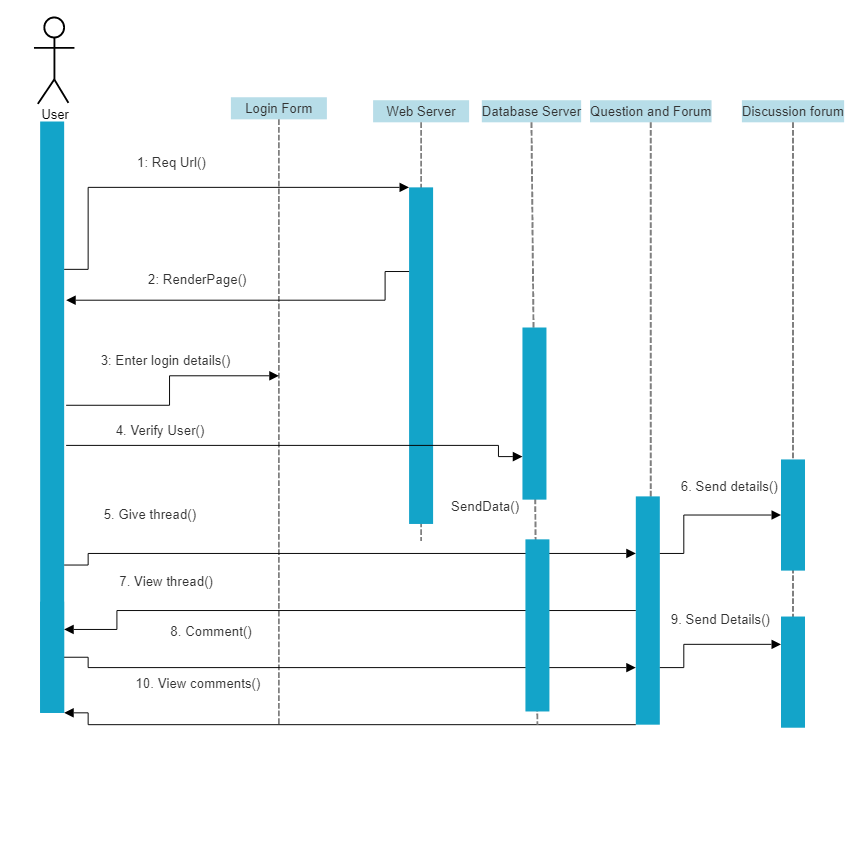


# Sequence Diagram:

User creating a review on the website:

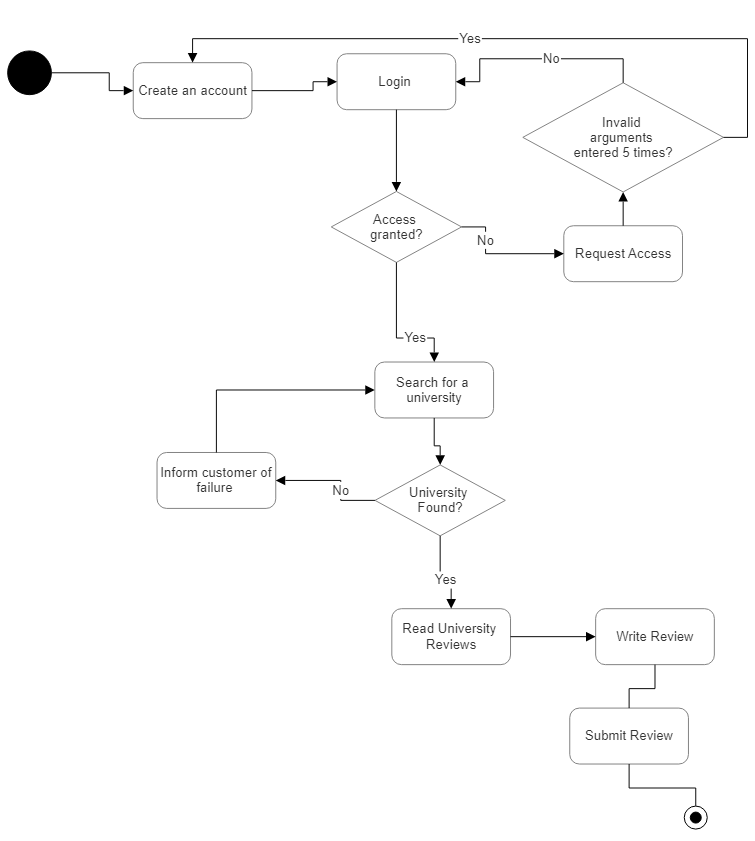


User interacting with the student forum:

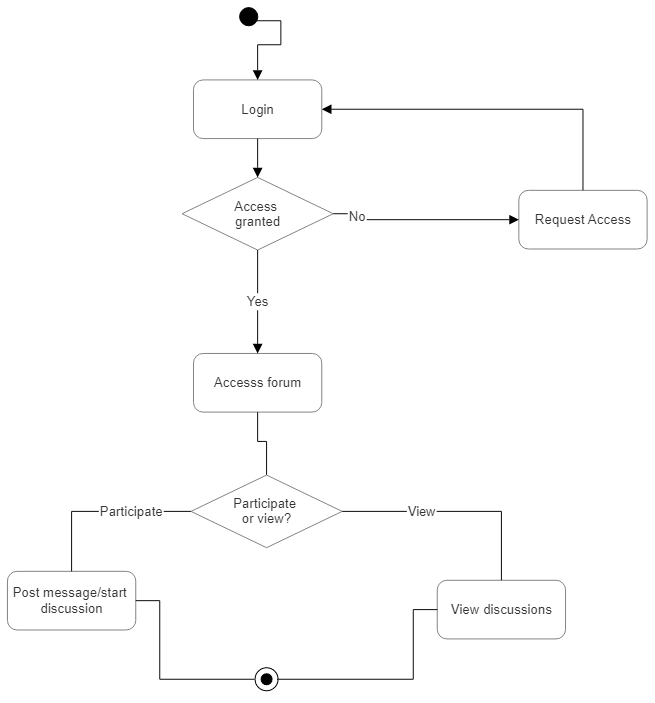


# Activity Diagrams:

Activity diagram showing a user viewing a university then writing a review:

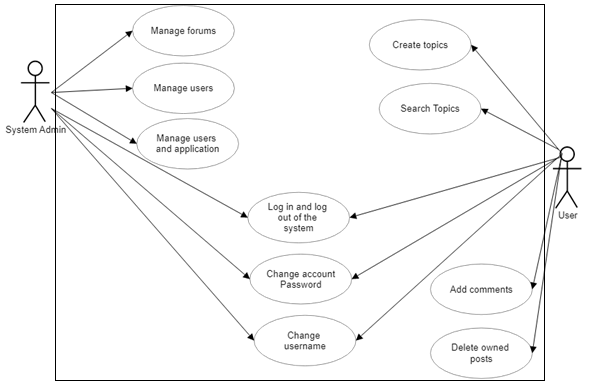


Student forum Activity Diagram:



# Use Case diagrams:

Original use case for admins and users interacting on the forum section of the website:



Updated Diagram:

