

Team Members

Roll No.	Gr No.	Name
323023	21910480	Ketaki Hadnurkar
323045	21910513	Aditi Nikam
323052	21910956	Roshan Purohit
323071	21910163	Shlok Zanwar

SRS Document

Code Rooms

1. INTRODUCTION

1.1 PURPOSE

The purpose of this document is to build an online learning management system to ease the process of assignment submissions.

1.2 DOCUMENT CONVENTIONS

This document uses the following conventions.

DB	Database
DDB	Distributed database
ER	Entity Relationship
UI	User Interface

1.3 INTENDED AUDIENCE AND READING SUGGESTIONS

This project is a development for the submission management system and it is restricted within the college premises. This project is useful for the teachers and as well as to the students.

1.4 PROJECT SCOPE

The purpose of the online learning management system is to ease submissions management and to create a convenient and easy-to-use application for

teachers, trying to collect/review submissions and for students to submit their assignments. The system is based on a relational database with its different types of logins, questions, user interfaces. We will have a database server supporting users across India. Above all, we hope to provide a comfortable user experience along with easy-to-use unique features.

1.5 REFERENCES

- Other learning management systems (Google Classroom, Online GDB Classroom)
- Documentations of technologies used.

2. OVERALL DESCRIPTION

2.1 PRODUCT PERSPECTIVE

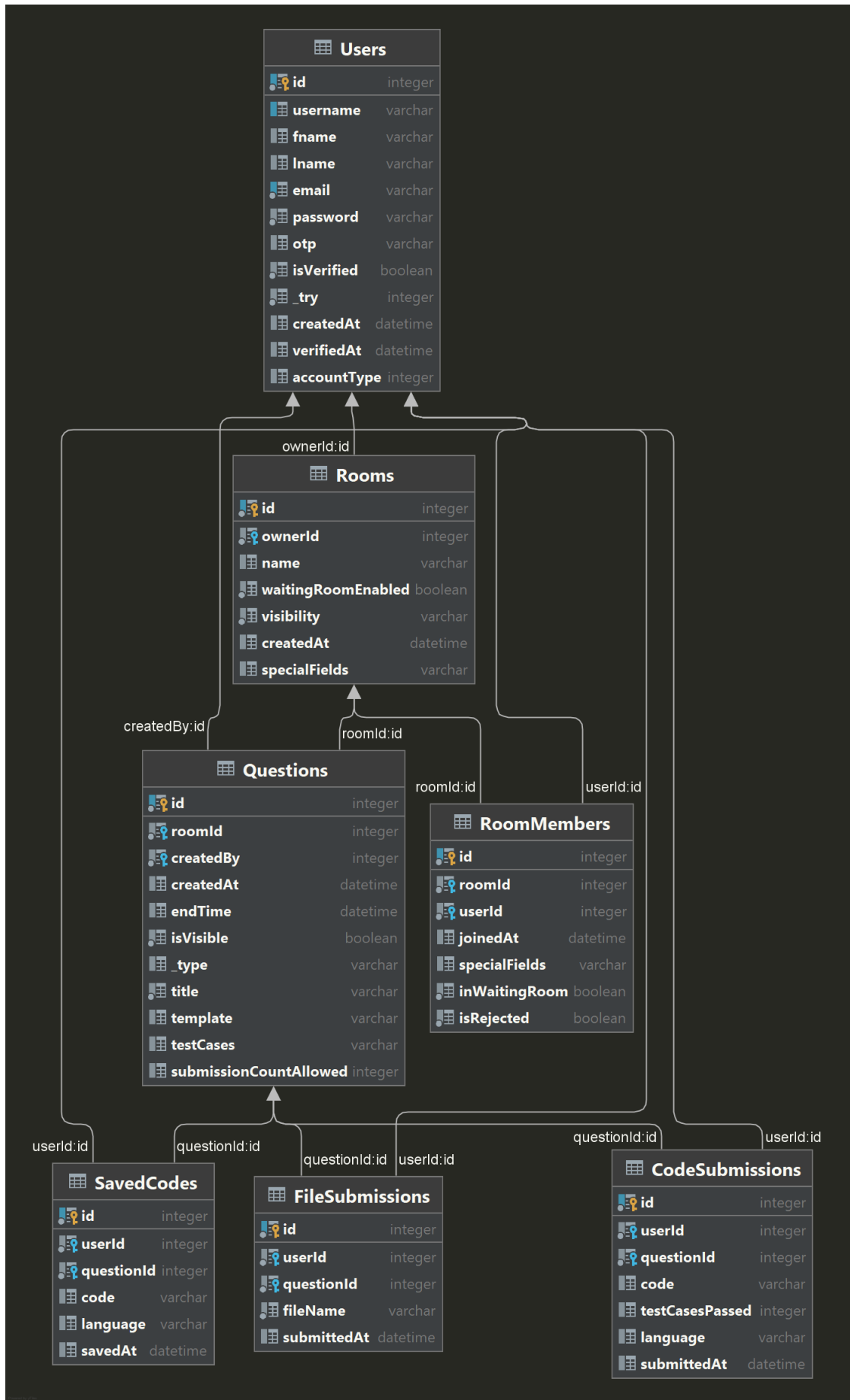
A distributed database system stores the following information.

- **User details:**
It includes the data such as first name, last name, email, username etc to uniquely identify any user.
- **Room details:**
It includes owner id, name, and other settings required for the classroom. This information will be used for identifying different classrooms.
- **Room members:**
This table takes care of enrolment of students in different rooms.
- **Question details:**
It includes room id (to which the question belongs), created by, title, description and other question parameters.
- **Submissions details:**
It has all the details of submissions of assignments/questions in a well-structured form categorized by question (to which submission belongs), user (the owner of submission) etc.

2.2 PRODUCT FEATURES

- Create Rooms with features like waiting room authority and visibility on the go with one click.
- Create questions of different types on the go with side by side expected user preview on the left with all editable components on right in resizable panels.
- Code type questions have option to add automated testcases which will be processed on submissions.
- Question settings like End time and visibility can be edited anytime.
- Students can also get a detailed of questions view with good formatting of submission due time and other settings.
- Code type questions can be viewed with user friendly resizable code editor with options like language-selection, run, save and submit.
- File type questions provide a preview before submission for clarity.

The major features of database system as shown in below [entity–relationship model](#) (ER model).



2.3 USER CLASS and CHARACTERISTICS

The system will support two types of user privileges, Teacher and Student.

Student type users of the system should be able to retrieve of all the rooms enrolled by them, all the questions of those rooms and all other room details from the database.

On the other hand, teacher users should be able to create rooms, create/edit questions, view submissions anytime.

Students will have access to student functions, and the teachers will have access to both student and teacher functions.

The teacher should be able to do the following functions:

- Create a new room (classroom)
 - Modify room settings (Name, visibility, waiting rooms etc).
 - Create Question
 - Create question of two types (Code / File Submission).
 - Edit question template.
 - Modify settings (Due date, visibility, etc)
 - Set automated testcases (Code type question)
- Delete any of his/her room.
- Get all the details of students enrolled in a particular room.
- Remove students from room.
- View submissions of all students for all questions of that room.
- View statistics of a particular room.

The student should have following functionalities:

- Join any room by a joining link.
- View all details of that room.
- Submit any question visible to him from any of his/her enrolled rooms.
- Review already submitted questions / modify them if due date isn't over.

Other common features:

- Every user should have access to updating their profile / password.
- Code Playground
 - Easy to use free code editor and compiler.
 - Save codes locally and modify them anytime.

2.4 OPERATING ENVIRONMENT

Operating environment for the learning management system is as listed below.

- Distributed database
- Client / Server system
- Operating system: Any supporting a web browser.
- database: MySQL database
- platform:
 - Frontend: ReactJS, Html, CSS
 - Backend: Python, FastApi

2.5 DESIGN and IMPLEMENTATION CONSTRAINTS

1. The global schema, fragmentation schema, and allocation schema.
2. SQL commands for above queries/applications
3. Adding google analytics so that usage can be tracked.
4. Implement the database at least using a centralized database management system.

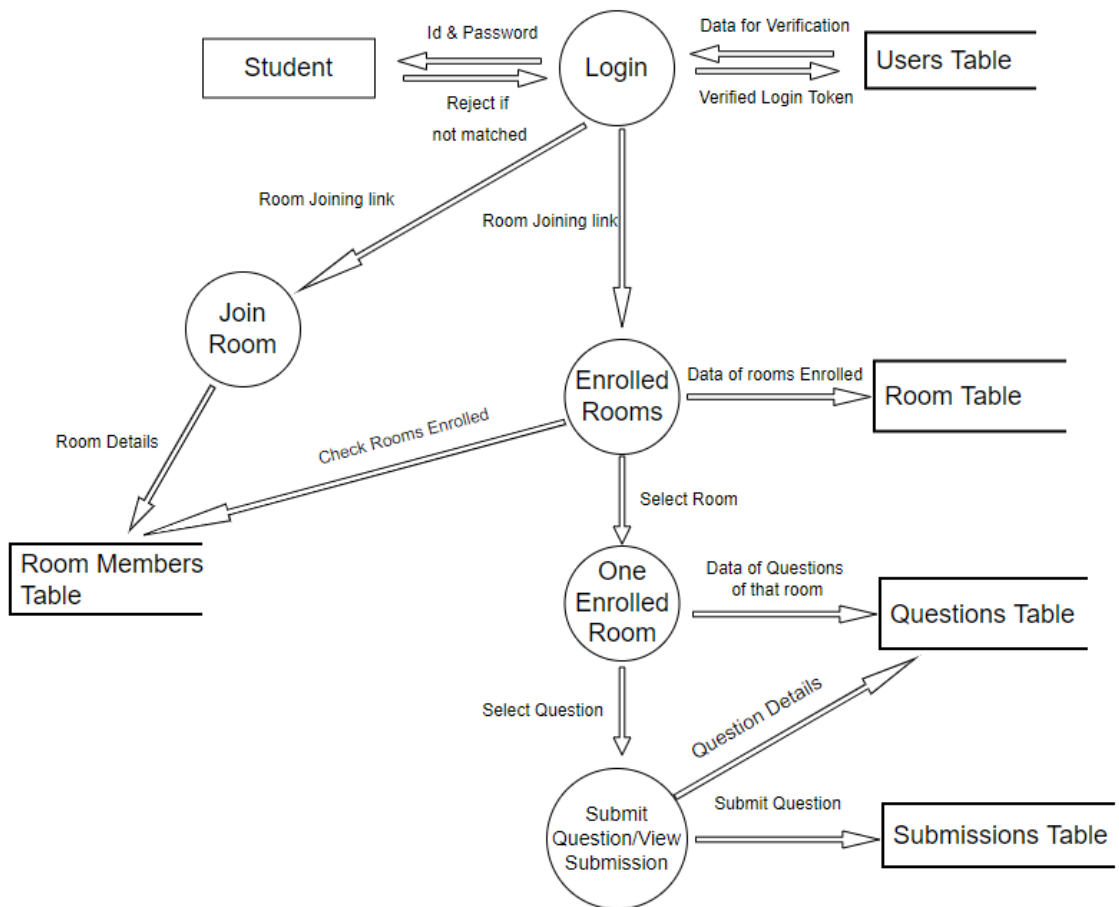
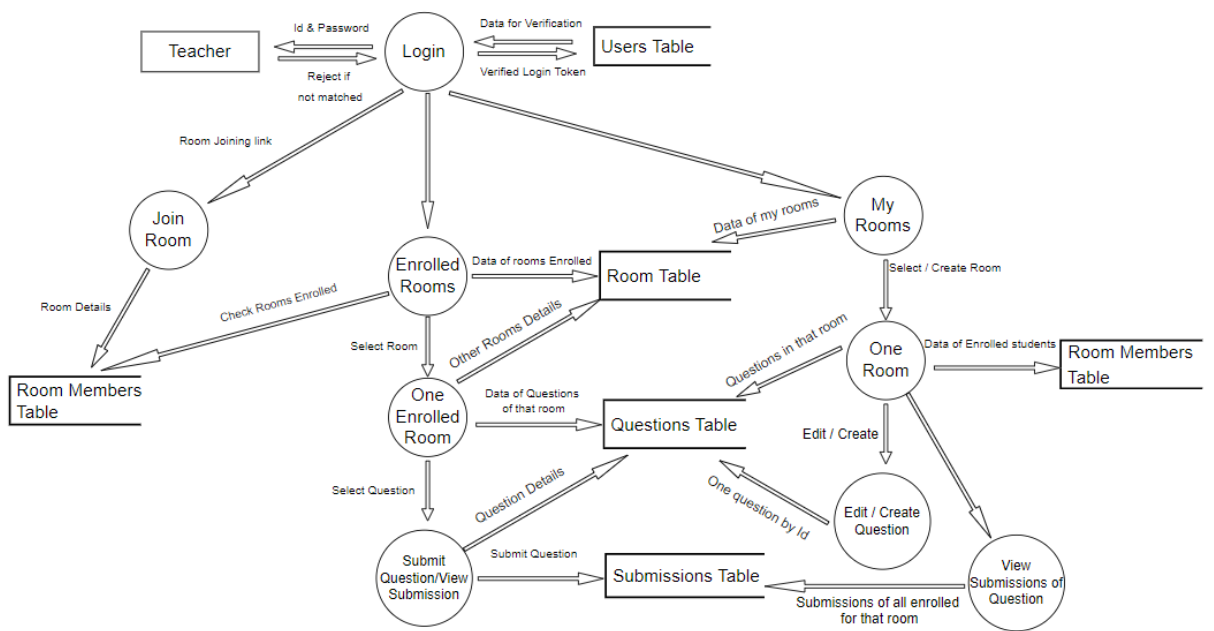
2.5 SYSTEM DESIGN SPECIFICATIONS

Data Flow Diagram (DFD)

It is a way of representing system requirements in graphical form; this led to modular design. A DFD describes a data flow(logical) rather than how they are processed. So, they do not depend upon software, hardware, data structure or file organization. It is also known as 'bubble chart'. A DFD is a structured analysis and a design tool that can be used for flowcharting in place of, or in association with, information-oriented and process- oriented system flowcharts. A DFD is considered as an abstract of the logic of information oriented or process-oriented system flowchart.

The four basic symbols used to construct a dataflow diagram are:

- Rectangle: Rectangle represents a data source or destination.
- Direct line represents flow of data.
- Oval represents a process that transforms into streams.
- An open-ended rectangle represents storage.



3. SYSTEM FEATURES

- **DESCRIPTION and PRIORITY**

The learning management system maintains information on rooms, enrolments, questions, submissions and settings. Of course, this project has a high priority because there are not many easy-to-use platforms with features such as in this one.

- **STIMULUS/RESPONSE SEQUENCES**

- **FUNCTIONAL REQUIREMENTS**

Other system features include:

CLIENT/SERVER SYSTEM

The term client/server refers primarily to an architecture or logical division of responsibilities, the client is the application (also known as the front-end), and the server is the DBMS (also known as the back end).

A client/server system is a distributed system in which,

- Some sites are client sites and others are server sites.
- All the data resides at the server sites.
- All applications execute at the client sites.

4. EXTERNAL INTERFACE REQUIREMENTS

4.1 USER INTERFACES

- Front-end software: ReactJS, Html, CSS
- Back-end software: Python, FastApi, MySQL

4.2 HARDWARE INTERFACES

- A device with a browser which supports CGI, HTML & JavaScript.

4.3 SOFTWARE INTERFACES

Following are the software used for the management of online application.

Software Used	Description
Operating system	This product is independent of any OS and can run on any system which supports browser with CGI, HTML & JavaScript.
Backend	FastApi is a framework of python to ease the process handling REST Api's and hosting server. Python has different libraries to simplify the work of managing the database.
Database	To save all records we have chosen MySQL database.
Frontend	To implement this project, we have chosen ReactJS. As React is an opensource library it has access to many free to use components which help in building any app in less time.

4.4 COMMUNICATION INTERFACES

This project supports all types of web browsers.

We are using custom logic and components for the application all based on JavaScript.

5. NONFUNCTIONAL REQUIREMENTS

5.1 PERFORMANCE REQUIREMENTS

The steps involved to perform the implementation of database are as listed below.

A) E-R DIAGRAM

The E-R Diagram constitutes a technique for representing the logical structure of a database in a pictorial manner. This analysis is then used to organize data as a relation, normalizing relation and finally obtaining a relation database.

- **ENTITIES:** Which specify distinct real-world items in an application.
- **PROPERTIES/ATTRIBUTES:** Which specify properties of an entity and relationships.
- **RELATIONSHIPS:** Which connect entities and represent meaningful dependencies between them.

B) NORMALIZATION:

The basic objective of normalization is to reduce redundancy which means that information is to be stored only once. Storing information several times leads to wastage of storage space and increase in the total size of the data stored.

If a database is not properly designed, it can give rise to modification anomalies. Modification anomalies arise when data is added to, changed or deleted from a database table. Similarly, in traditional databases as well as improperly designed relational databases, data redundancy can be a problem. These can be eliminated by normalizing a database.

Normalization is the process of breaking down a table into smaller tables. So that each table deals with a single theme. There are three different kinds of modifications of anomalies and formulated the first, second and third normal forms (3NF) is considered sufficient for most practical purposes. It should be considered only after a thorough analysis and complete understanding of its implications.

5.2 SAFETY REQUIREMENTS

If there is extensive damage to a wide portion of the database due to catastrophic failure, such as a disk crash, the recovery method restores a past copy of the database that was backed up to archival storage (typically tape) and reconstructs a more current state by reapplying or redoing the operations of committed transactions from the backed-up log, up to the time of failure.

5.3 SECURITY REQUIREMENTS

Security systems need database storage just like many other applications. However, the special requirements of the security market mean that vendors must choose their database partner carefully.

5.4 SOFTWARE QUALITY ATTRIBUTES

- **AVAILABILITY:** The platform should be always live, and the backend server should always be active.
- **CORRECTNESS:** Room / questions data should only be visible to the teacher owning the room / student enrolled in the room.
- **MAINTAINABILITY:** The teachers should be able to maintain all rooms and manage all settings for questions/rooms.
- **USABILITY:** The UI should be simple with important functionalities accessible in least number of clicks.