COMSATS University Islamabad,
Park Road, Chak Shahzad, Islamabad Pakistan

SOFTWARE DESIGN DESCRIPTION

(SDD DOCUMENT)

for

MOOC

Version 1.1

By

Ali Hassan	**CUI/FA16-BSE-001/ISB**
Nabba Asif	**CUI/FA16-BSE-027/ISB**
Raja Muhammad Saad	**CUI/FA16-BSE-029/ISB**

Supervisor

Mr. Rizwan Rashid

Bachelor of Science in Software Engineering (2016-2020)

Table of Contents

Revision History

Name	Date	Reason for changes	Version

Application Evaluation History

Comments (by committee) ***include the ones given at scope time both in doc and presentation**	**Action Taken**

Supervised by

Mr. Rizwan Rashid

Signature_______________

1. Introduction

1.1 Scope

Our system is a web-based E-learning application which would provide an increased number of ways for student teacher interaction. Instead of directing all the traffic through one or two specified ways of communication, our system would provide the users with facilities like audio calls, conference video calls, screen sharing, direct messaging and classroom chat. The students and teachers both would have different ways of file sharing like direct messaging, portal or classroom chat. The teachers can use portal for uploading, viewing, updating lecture files, marks, assignment, quizzes and videos. The students can view these files, download them and submit their own files. Our system would provide us with two perspectives of E-learning. The first perspective is online learning, where the students can join an online classroom and can use classroom chat for interaction. The teacher can also start conference video call in this online room. Teachers can also use screen sharing feature during conference video calls. The second perspective is offline learning, where students can study from already uploaded content. The teachers can enable auto zoom feature during conference video call for making the content written on board more understandable for student. However, the video quality will not exceed 720p. A good quality external web camera would be required for capturing the videos. The students will be able to view recommended courses and trending courses. The visitors on the web application and the top courses will be shown in the form of graphs, charts and tables to the admin. At the end of course the students will be able to rate and review the teacher and the course. This feedback might help other students who are thinking to take that course.

1.1.1 Modules:

Following are the modules of our system:

1.1.1.1 Module 1: Account Management

This module will help the user to make account in our web application and control it. All the activities that the user will perform related to his account will be handled by this module. These activities include:

- Login
- Register
- Verify Account
- Forgot Password
- View Profile
- Update Profile
- Edit Details
- Change Passwords
- Change Image
- Logout

1.1.1.2 Module 2: Portal

1.1.1.2.1 <u>Sub-module 1: Teacher Portal</u>

The whole interface of the teacher's panel will be a part of this module. All the courses uploaded by the teachers will be in their portals. Teacher portal will help the teachers in:

- Uploading marks
- Uploading, viewing, editing, deleting and downloading assignments
- Uploading, deleting, editing and viewing Quizzes
- Uploading, deleting, editing and viewing lecture files
- Uploading, downloading, deleting and viewing Videos
- Uploading and viewing course calendar

1.1.1.2.2 <u>Sub-module 2: Student Portal</u>

The whole interface of the student's panel will be a part of this module. All the courses taken by the students will be in their portals. Student portal will help the students in:

- Viewing marks
- Submitting, viewing and downloading assignments
- Submitting and viewing Quizzes
- Downloading and viewing lecture files
- Watching Videos
- Viewing and downloading course calendar

1.1.1.3 Module 3: Admin Panel

Admin Panel will help in managing our web application. This module will include all the activities that the admin will perform. These activities include:

- Approve/Disapprove Course: The teacher can add any course, but this course will only be activated when the admin will approve this course. The teacher will write the details of the course and the status of the course will be "Pending for Approval". This course will be sent to the admin, the admin can approve or disapprove the course. If the admin approves the course, the teacher will be notified about it and the status of the course will change to "Activated". If the admin disapproves a course, then the teacher will be notified about it and the course will be automatically removed.
- Block/Unblock Accounts: The users can report an account and the admins will get that report and they can block the user. Likely, an admin can unblock any blocked user.

1.1.1.4 Module 4: Notification and Messaging

1.1.1.4.1 <u>Submodule 1: Notification</u>

The students will be able to receive the notifications when:

- Teacher will upload an assignment, quiz, lecture file, video for the course which student has subscribed.
- The teacher will make an announcement for a course.
- The teacher starts a conference video call.

The teachers will be able to make announcements that will be sent in the form of notifications to the students. The teachers will be notified when

- A student will subscribe for a course
- A student will submit assignment or quiz.

- A student will rate teacher or any of the course that the teacher is teaching.
- Admin will approve or disapprove a course.

1.1.1.4.2 Submodule 2: Today Events

This submodule will work as a reminder for both student and teacher. This submodule will be just the notification submodule with a little amendment in it. This module will give only the daily reminders to the users (e.g., “Today is the last date for XXX assignment submission”).

1.1.1.4.3 Submodule 3: Messages

This submodule will be helpful for users’ interaction with one another. The students can send messages to one another and to their teachers.

1.1.1.4.4 Submodule 4: Classroom Chat

This chat will be in the online rooms where teachers will be using live streaming. During the live broadcasting the students will be able to send their queries to the teachers. This means that this submodule will provide a highly interactive platform to the users. In this online chat the user will be able to send messages, audios and files.

1.1.1.5 Module 5: Communication using VoIP

This module will also help in increasing the ways of interaction between the users. The teacher and student both will be able to make audio calls. These audio calls can be one to one or one to many or many to many. Push-to-Talk functionality will also be provided in this module. The user will push “P” whenever he wants to speak, on releasing the P key microphone will be muted.

1.1.1.6 Module 6: Conference Video Call

This module will be helpful for online courses. Online courses will be the courses in which the participants can join a room for mutual discussions. The teachers will be able to start a live broadcast in that room and students can join that live broadcast. Through this module the teachers will be able to give lectures by simply sharing the web camera. Conference video call will increase the student-teacher interaction. This functionality will only be limited to teachers, student won’t be allowed to use Conference Video Call to share their web camera.

1.1.1.7 Module 7: Auto Zoom

The auto zoom functionality of MOOC will make it stand out among other e-learning systems. The teachers can also use this feature online by enabling it during the conference video call (live broadcast). This module consists of following submodules:

1.1.1.7.1 Sub-module 1: Action Recognition

This submodule will track the teacher by action recognition. Once the teacher is detected through his movements this feature will detect either he is speaking or writing on board.

1.1.1.7.2 Sub-module 2: Smart Zooming

As soon as the action of teacher is detected, our auto zoom feature will start working. If the teacher is speaking, then the teacher will be the center of attention. If he is writing on board then that white board will be zoomed in. This module will be helpful for the students to view the board more properly.

1.1.1.8 Module 8: Screen Sharing

Since not all teachers what to give lecture by making their own video so teachers need an alternate method to show slides or any screen content thus need for computer screen sharing. Teachers will be able to share their laptop screen with students during the live broadcast. The teachers will enable the “Screenshare” option for this functionality. Teacher’s laptop/computer screen will be broadcasted to all the students who are online in the online classroom. This functionality will only be limited to teachers, student won’t be able to use screen sharing to share their laptop screen.

1.1.1.9 Module 9: E-learning Analytics

1.1.1.9.1 Submodule 1: Recommended Courses

Using machine learning algorithm Courses suggestions will be tuned according to the student's subscription only those courses will display to the student that are relevant instead of showing all the courses. It will automatically learn new acquired taste of the student and will modify the recommendations according to it. It will help the student to choose courses that are useful for him and will save time for the student.

1.1.1.9.2 Submodule 2: Top Courses

The most famous that are taken by most of the students will be shown to the students under the category of “Trending Courses”.

1.1.1.9.3 Submodule 3: Trending Courses’ Analytics

This submodule is for admin. The admin will be able to view the top courses based on click and subscriptions. These courses will be shown in the form of tables and charts.

1.1.1.9.4 Submodule 4: Visitors’ Analytics

This submodule will show the number of visitors each week in the web application. These analytics will be shown in the form of graph.

2. Design methodology and software process model

2.1 Design Methodology: Object Oriented Methodology

We will be using object-oriented methodology because:

- It will increase the code reusability.
- It is a modular development and one can do independent testing.
- It will make our system more flexible to changes
- It will make maintainability of system easier.

2.2 Software Process Model: Incremental Process Model

We will be using incremental process methodology because:

- It is more flexible for accommodating changes.
- It is easier to test and debug during a small iteration.
- It uses divide and conquer for breakdown of tasks.

3. System overview

Give a general description of the functionality, context and design of your project. Provide any background information if necessary.

3.1 Architectural design

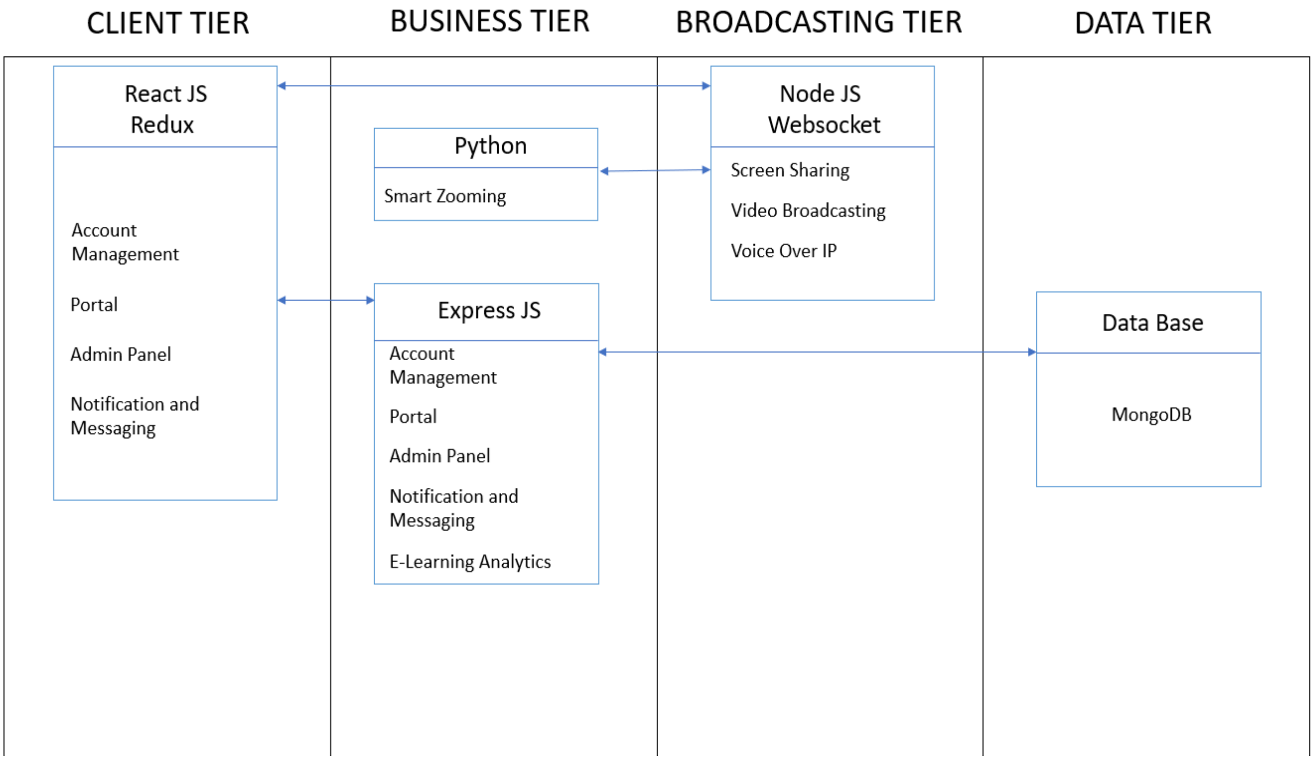


Figure 1: Architecture Diagram

3.1.1 Description of Architecture Diagram

The architecture of our system consists of following 4 tiers with three servers.

3.1.1.1 Client Tier

This is the presentation layer, and this consist of all the user interface of our website. React JS and Redux will be the technology that is used for building this tier. Following are the modules included in this tier:

1. Account Management
2. Portal
3. Admin Panel
4. Notification and Messaging

3.1.1.2 Business Tier

This consist of 2 servers:

1. Python
2. Express JS

Python will be used for the smart zooming of live broadcasting.
Express JS is basically used for building the backend of modules including:

1. Account Management
2. Portal
3. Notification and Messaging
4. E-Learning Analytics

3.1.1.3 Broadcasting tier

This tier will consist of third server, called **websocket** server on **nodejs**. Following features will run in this server:

1. Screen Sharing
2. Video Broadcasting
3. Voice Over IP

All these features will be presented to the user in user interface.

3.1.1.4 Data Tier

Tis tier will consist of database, MongoDB. All of the data entered in the system will be saved in this database in json documents.

3.2 Process flow/Representation

3.2.1 Activity Diagram: Register and Account Verification

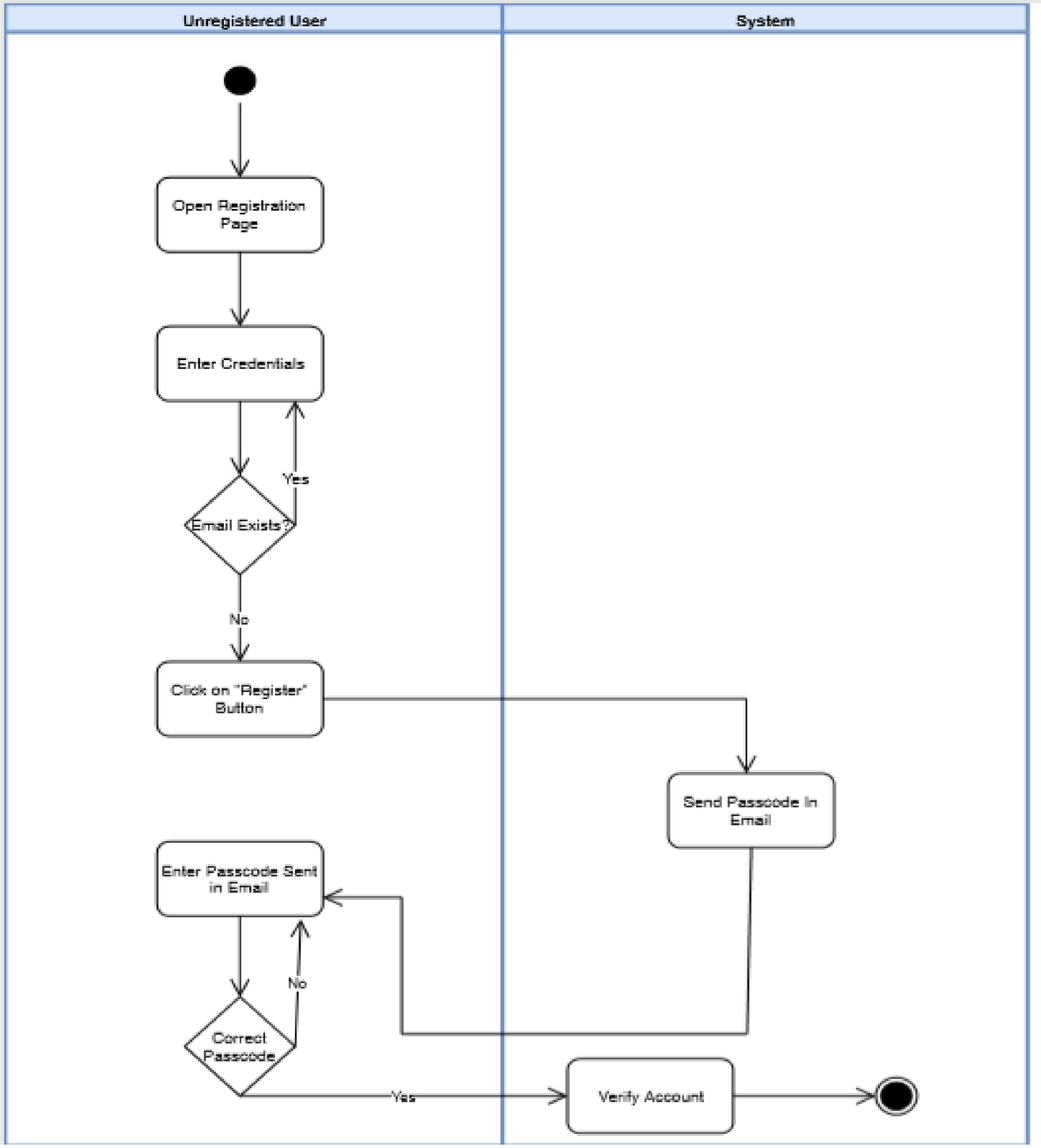


Figure 2: Register and Account Verification

3.2.1.1 Explanation

The activity diagram shows the registration of MOOC. The user will first open MOOC. The he will open the registration page. The user will enter his first name, last name, email, password and confirm password. If the email is already registered, then the user will be asked to select another email. After this the system will display verify account page.

3.2.2 Activity Diagram: Forgot Password

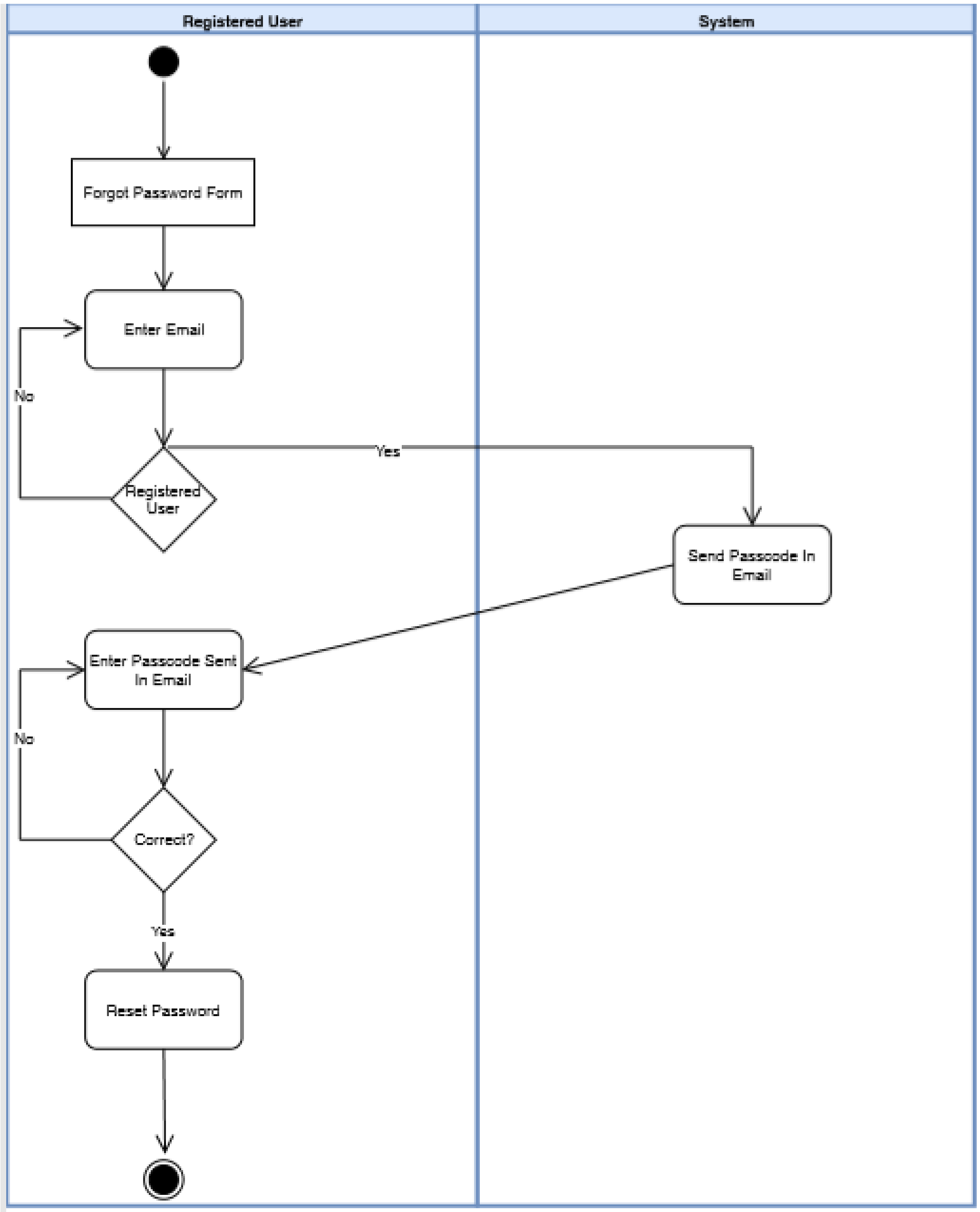


Figure 3: Forgot Password

3.2.2.1 Explanation

The activity diagram shows forgot password option. If the user forgets his password, then he will click on forgot password. The system will open a form where the user must enter the passcode sent to him in his email. Through this the system would successfully verify the user and then will let the user set a new password.

3.2.3 Activity Diagram: Update Profile

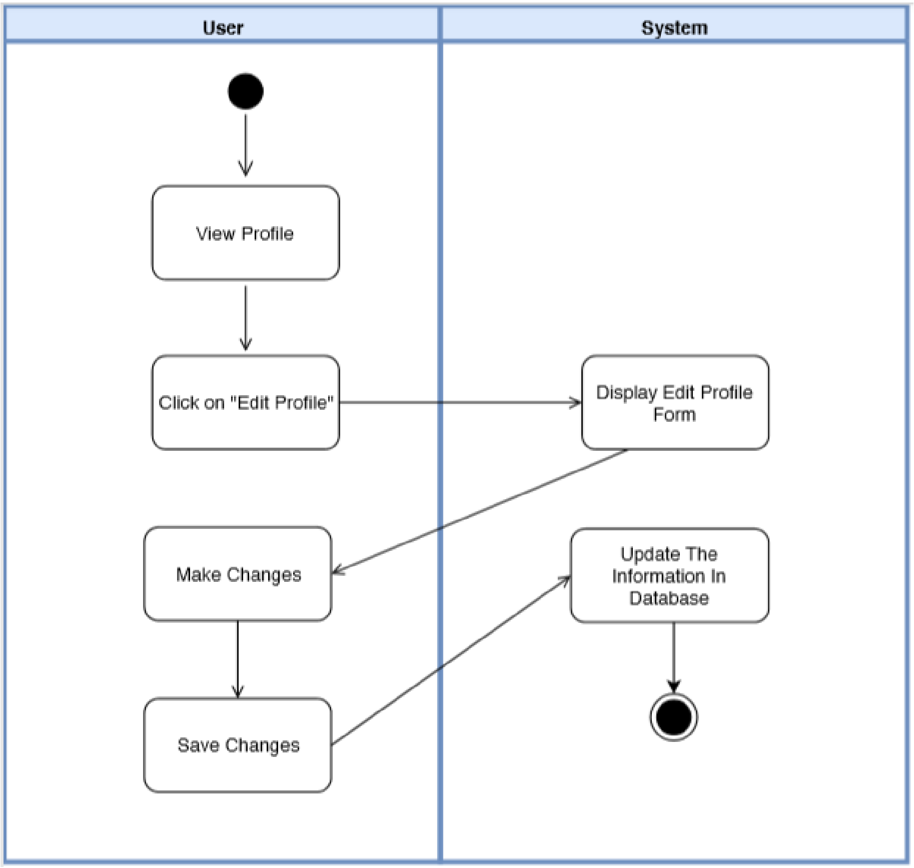


Figure 4: Update Profile

3.2.3.1 Explanation

To update the profile information the user will first view his profile. Then he would click on edit to bring changes in the profile information. A form will come up where he can edit or delete the information. At the end the user will save the changes made.

3.2.4 Activity Diagram: Upload and Approve Course

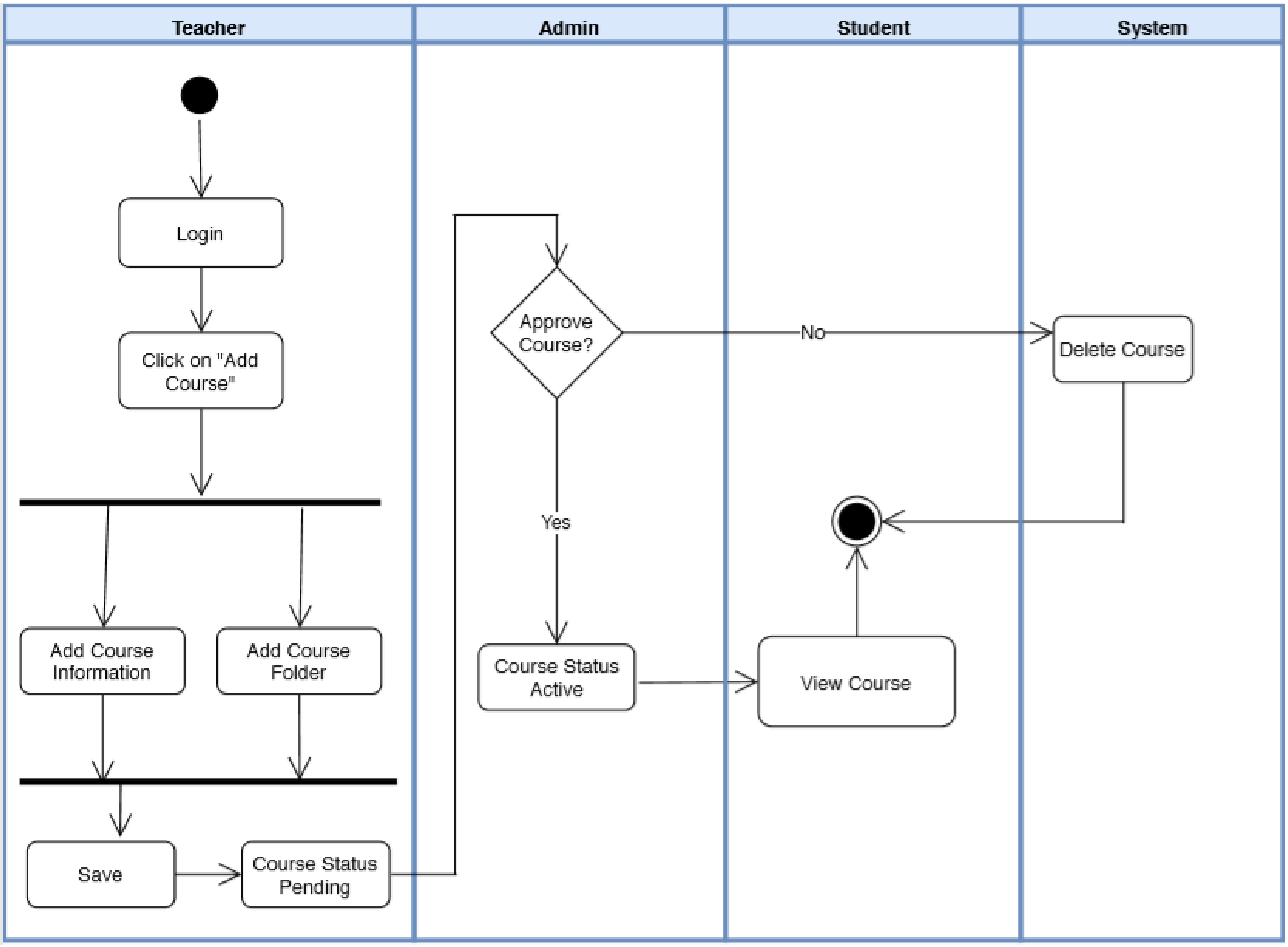


Figure 5: Upload And Approve Course

3.2.4.1 Explanation

The user will click on add course. A form will be displayed to the teacher. In this form he must update all the information about the course to be uploaded and he can also upload the course lecture file (optional). After adding all the information, the user will save the course. The course will be having pending status. This course will be sent to the admin for approval. If the admin approves that course, then the status of the course will change to active and the users will be able to view and subscribe for that course.

3.2.5 Activity Diagram: Quiz

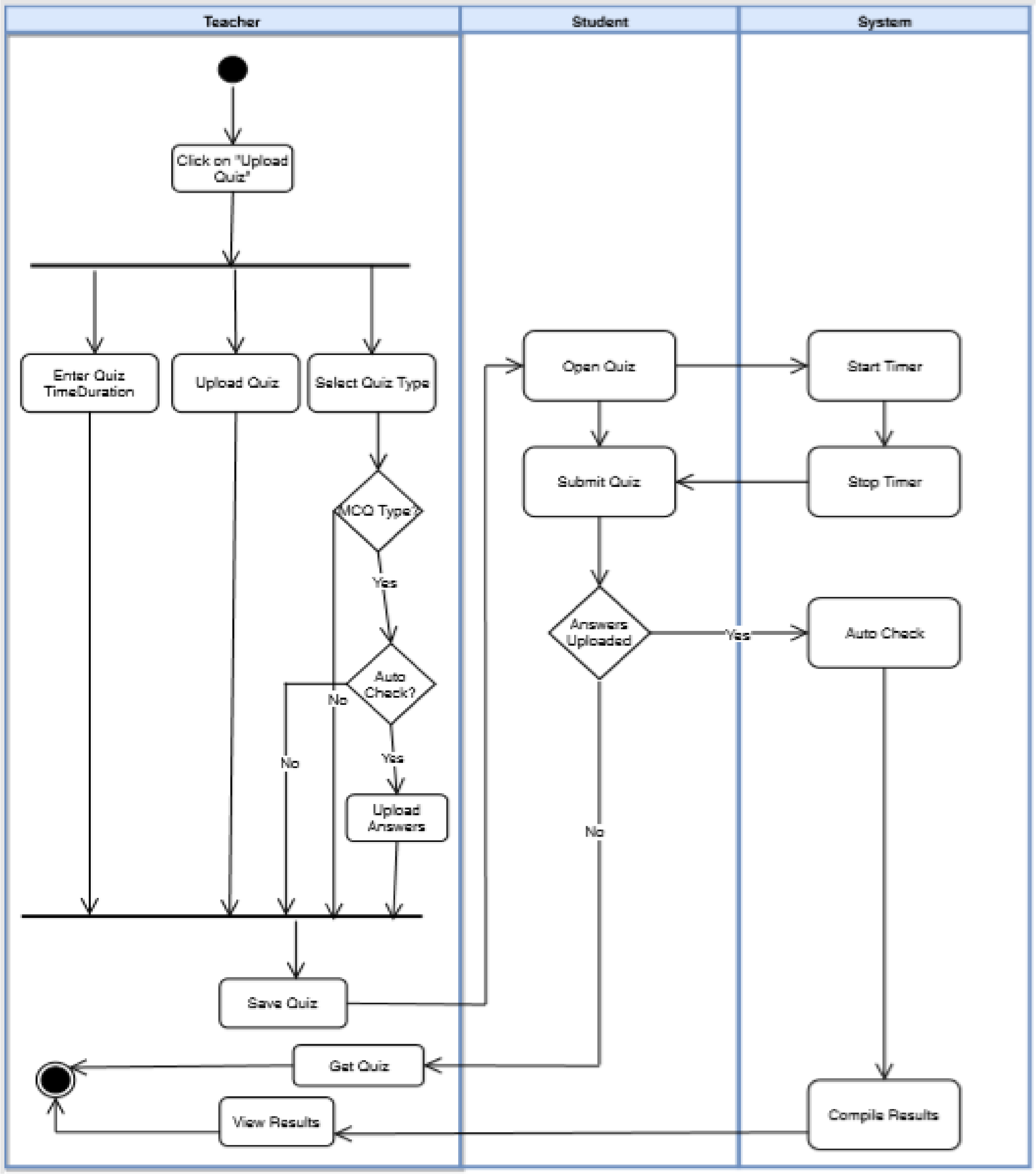


Figure 6: Quiz

3.2.5.1 Explanation

This activity diagram shows the process of uploading the quiz. The teacher will first click on “My Portal”. Then the teacher will select the course whose quiz he wants to upload. The teacher will then open the quiz section. The teacher will click on “Add”. The system will show a form top the teacher. The teacher will fill all the required areas. The teacher will upload a quiz file. If the teacher chooses "MCQ" as the type of quiz. Then he would get and additional functionality of auto check. If the teacher enables this functionality, then he had to upload an answer sheet too (in the required format). After uploading the quiz, the students will be able to view the quiz. For solving the quiz, they will open it. As soon as the student will open the quiz, the system will start the timer. The quiz will be submitted when the timer will end. For the quizzes in which teacher opted for auto check will be checked by the system and results will be sent to the teacher. For other normal quizzes, the teacher will simply receive the submitted quizzes.

3.2.6 Activity Diagram: Upload Assignment and Submit Assignment

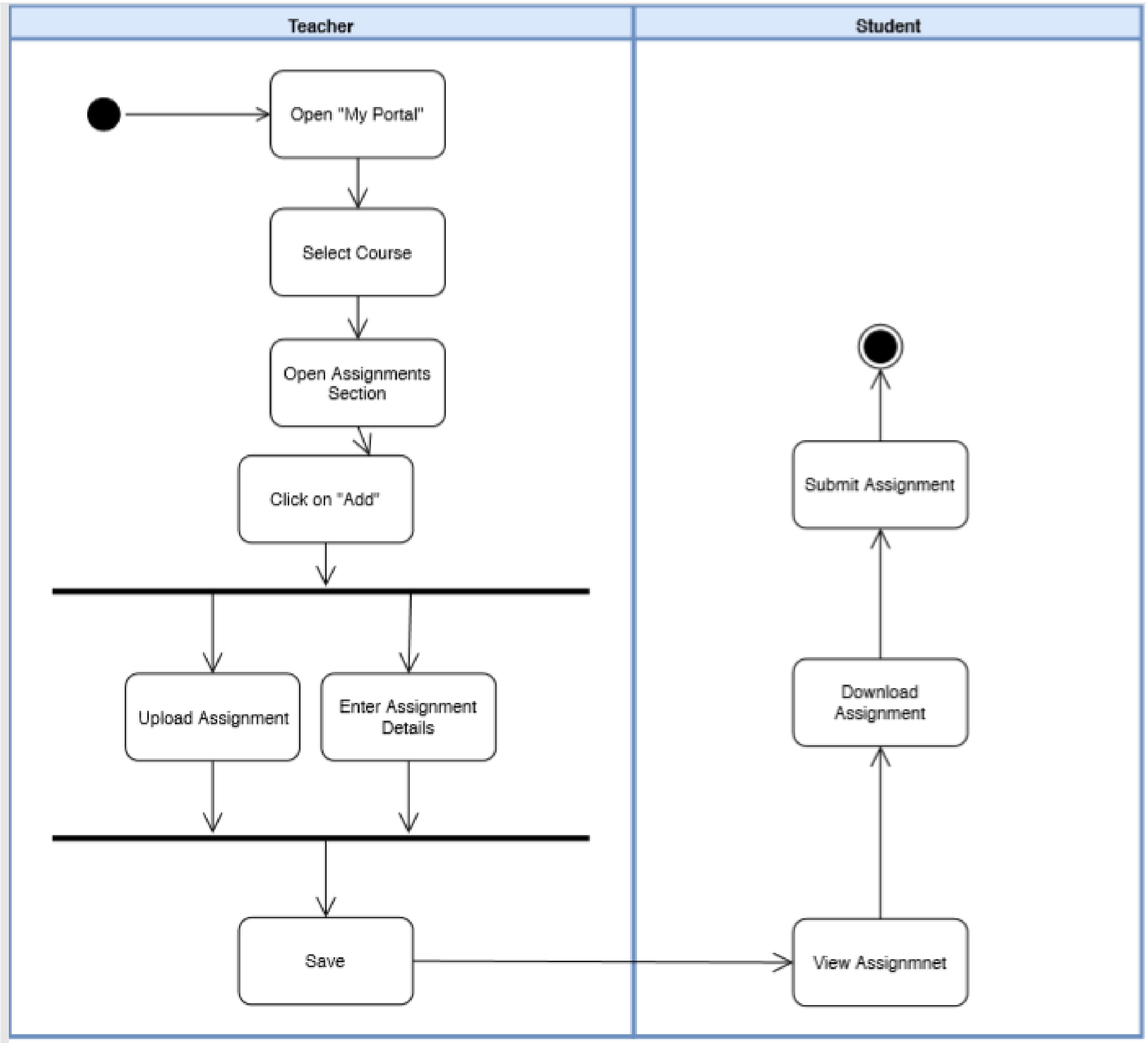


Figure 7: Upload Assignment and Submit Assignment

3.2.6.1 Explanation

For uploading the assignment, the teacher will open "My Portal". After entering in his portal, the teacher will select the course whose assignment he is about to upload. The teach we will a lick on "Add". The teacher will upload the assignment and give related information. At the end the teacher will save the assignment. The student will view the assignment, download the assignment and finally submit his answers.

3.2.7 Activity Diagram: Upload Marks

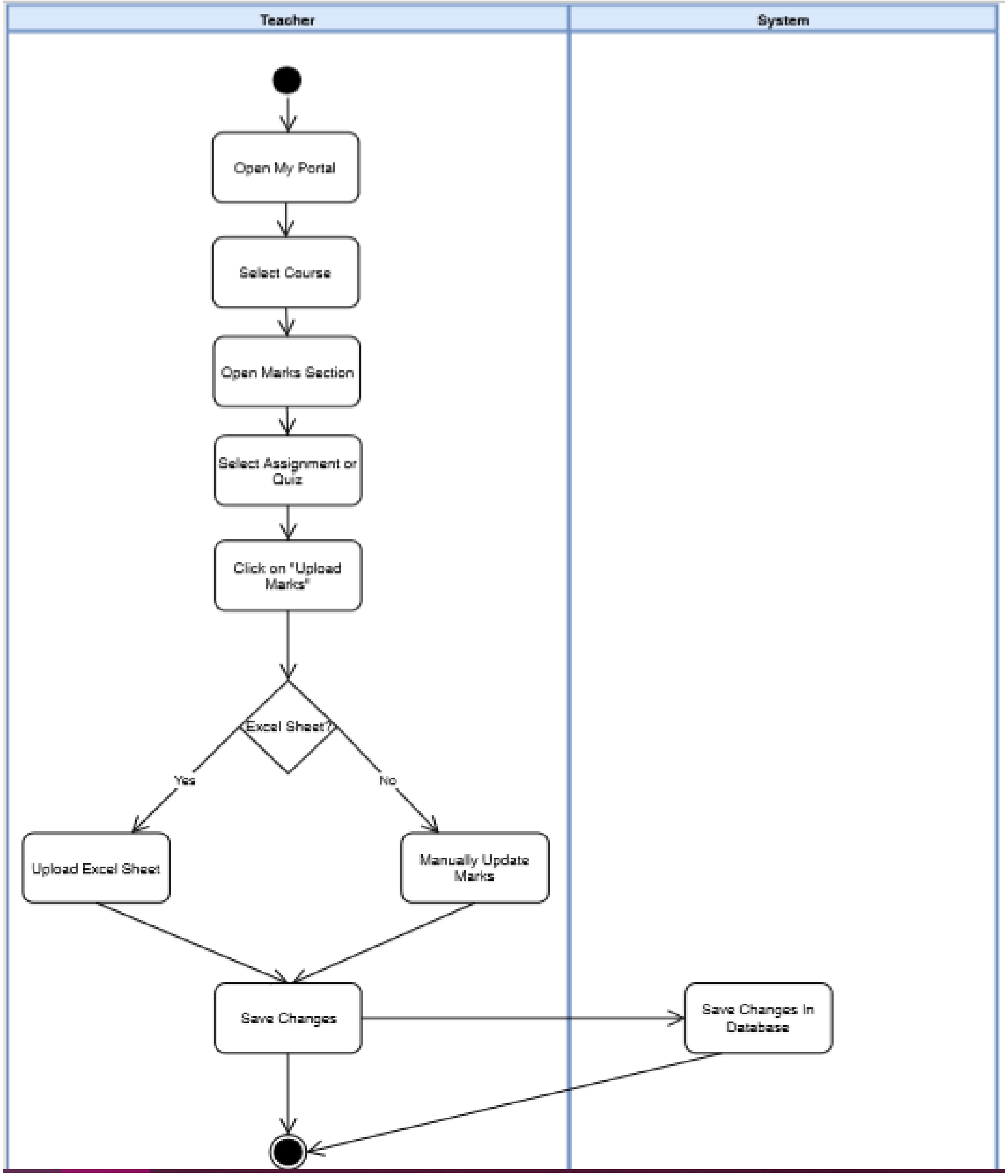


Figure 8: Upload Marks

3.2.7.1 Explanation

For uploading the marks, the teacher will open “My Portal”. The teacher will select the course of which marks he is about to upload. The teacher will then open the marks section. The teacher will select that if he is uploading the marks of assignment or quiz. The teacher will have option of both: uploading the excel sheet or uploading the marks manually. After the cages and the updates, the user will save the video.

3.2.8 Activity Diagram: Report User

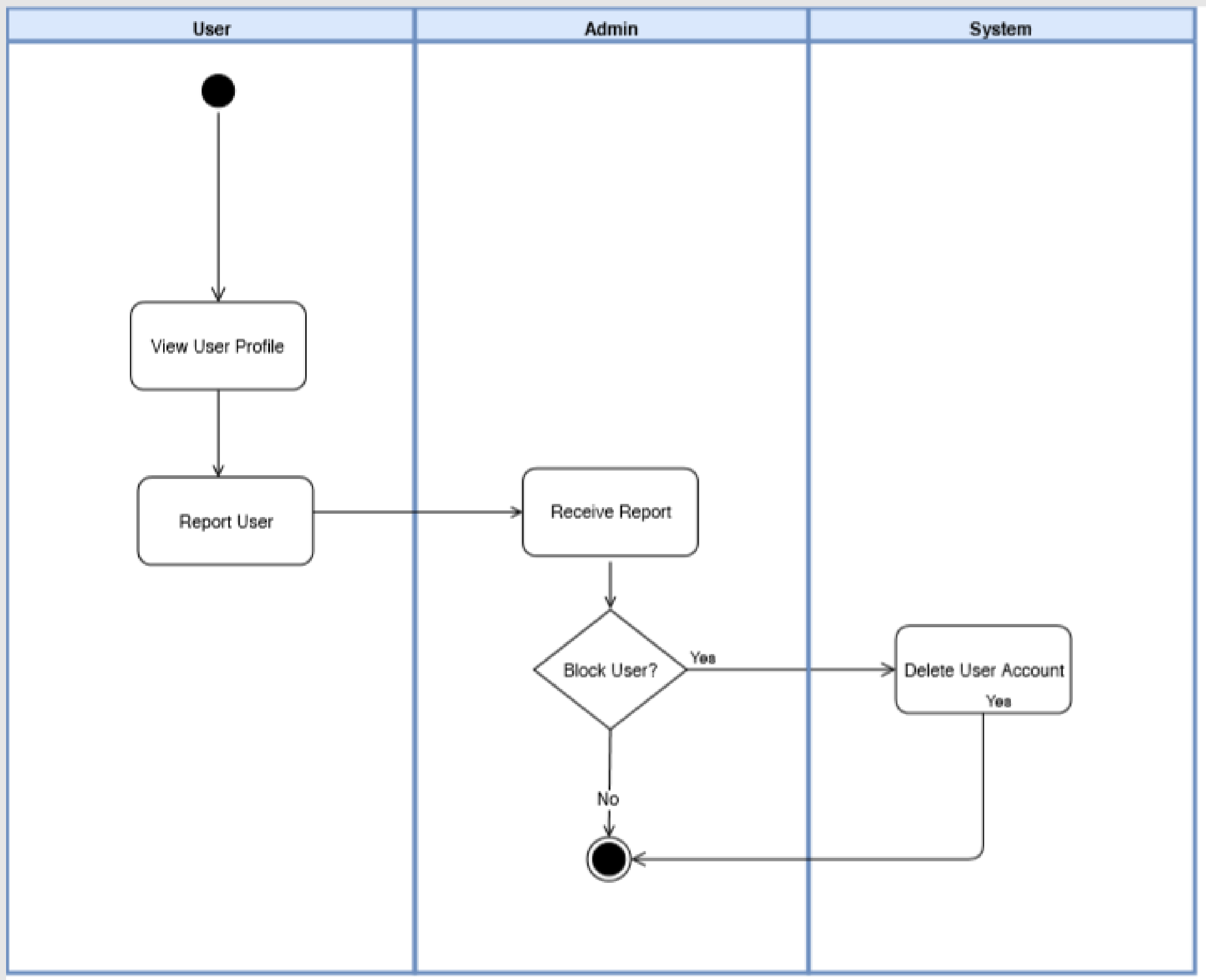


Figure 9: Report User

3.2.8.1 Explanation

The activity diagram shows the process of reporting a user. This report request will be sent to the admin. The admin will have the choice of blocking that user. If the admin blocks the user, the user account will be deleted.

3.2.9 Activity Diagram: Join Live Broadcast

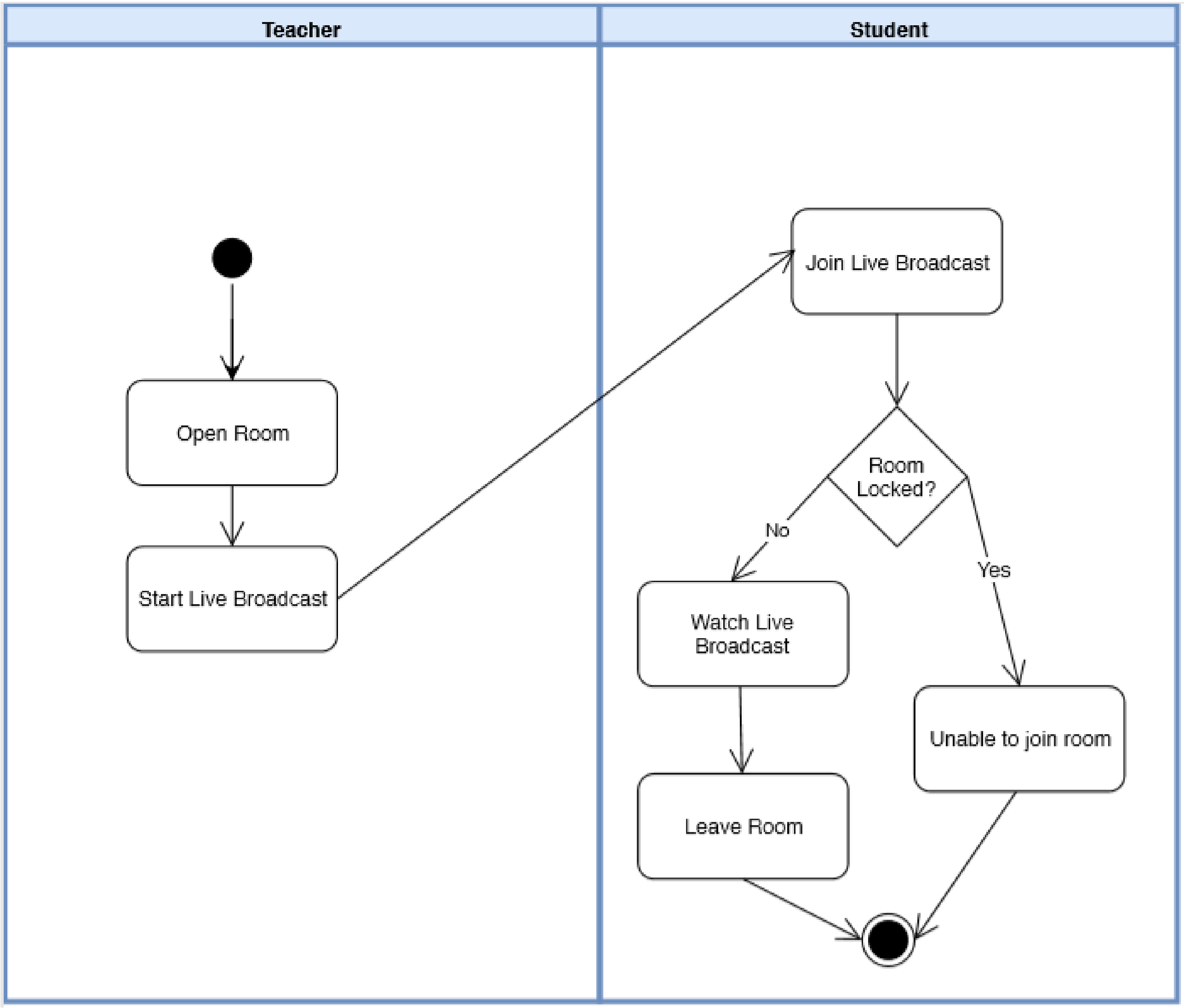


Figure 10: Join Live Broadcast

3.2.9.1 Explanation

The students will be able to join the live broadcast if: The teacher has started the live broadcast and if the room is not locked by the teacher. Of the room is locked then the student will not be able to move in.

3.2.10 Activity Diagram: Smart Zoom

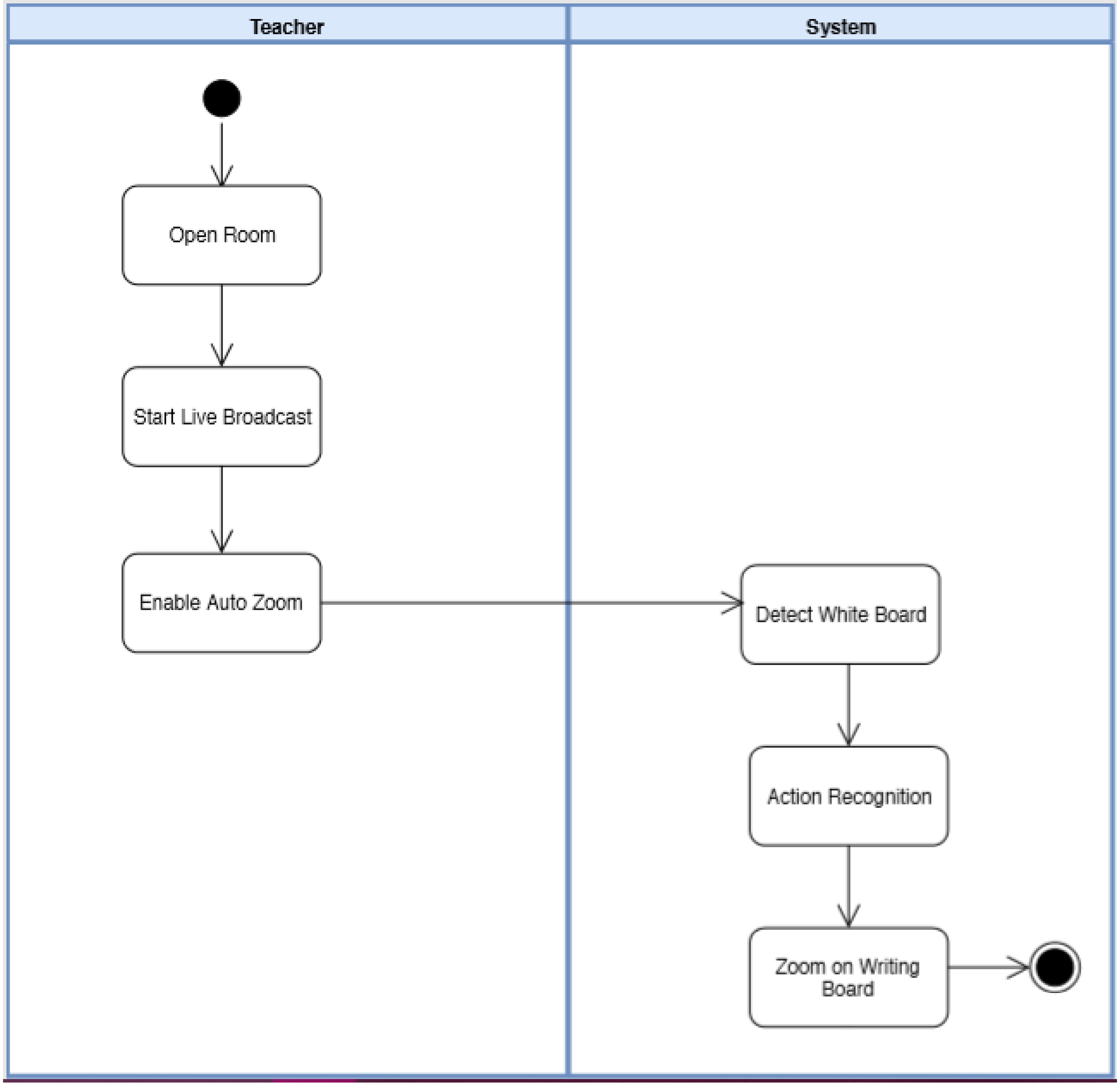


Figure 11: Smart Zoom

3.2.10.1 Explanation

The activity diagram shows that at any time in live broadcast the teacher will be able to enable auto zoom feature. As soon as the teacher will enable this feature the system will detect the white bead. After detecting the white board, the system will recognize the action of the teacher. After successful traction of white board and auction recognition, the system would zoom on whiteboard.

3.2.11 Activity Diagram: Save Video

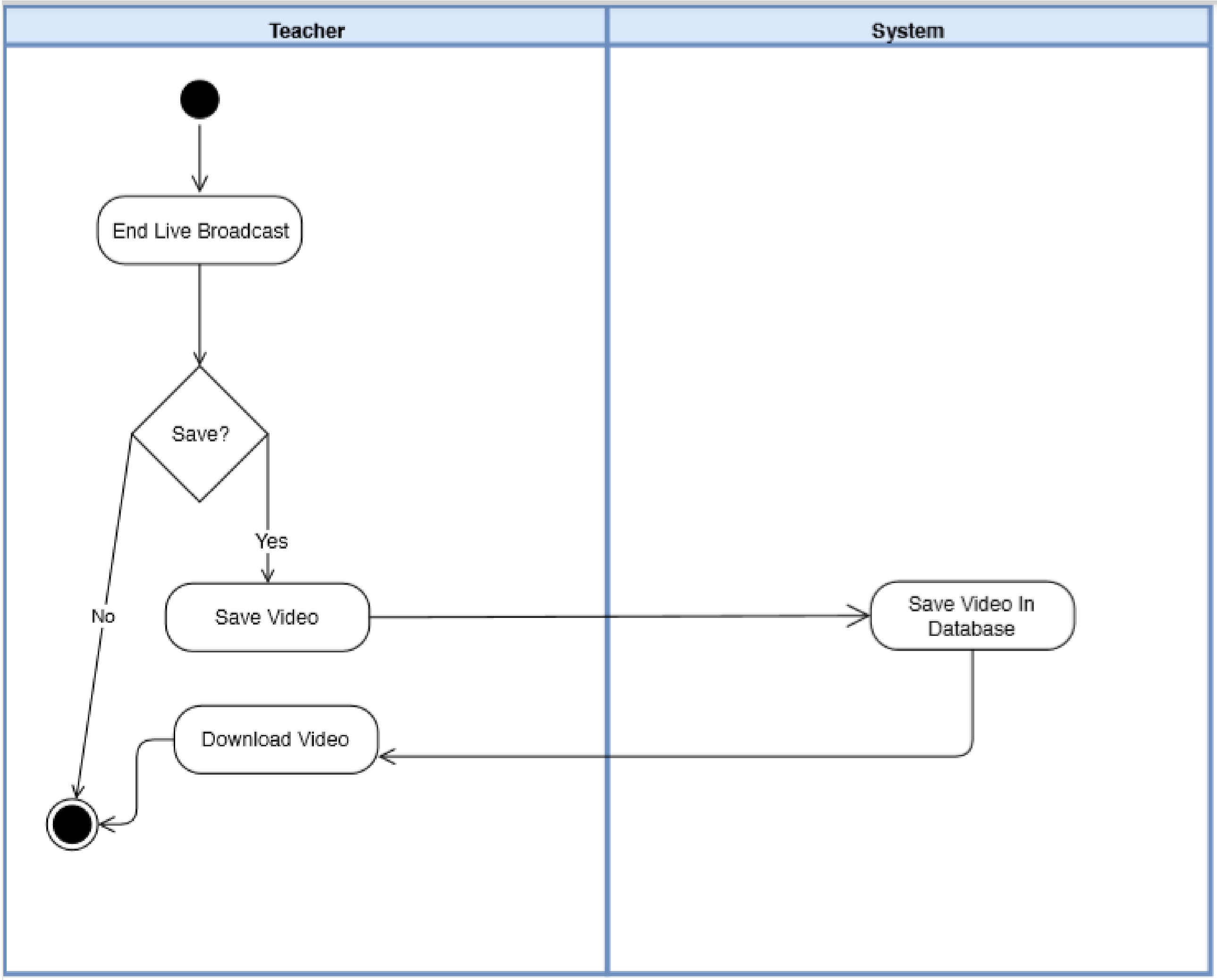


Figure 12: Save Video

3.2.11.1 Explanation

As soon as the live broadcast will end, the teacher will be asked if he wants to save the video. If he saves it then the video will be saved in the course folder. The teacher will then be able to download or delete that video and the student will be able to watch that video.

3.2.12 Activity Diagram: Make Audio Call

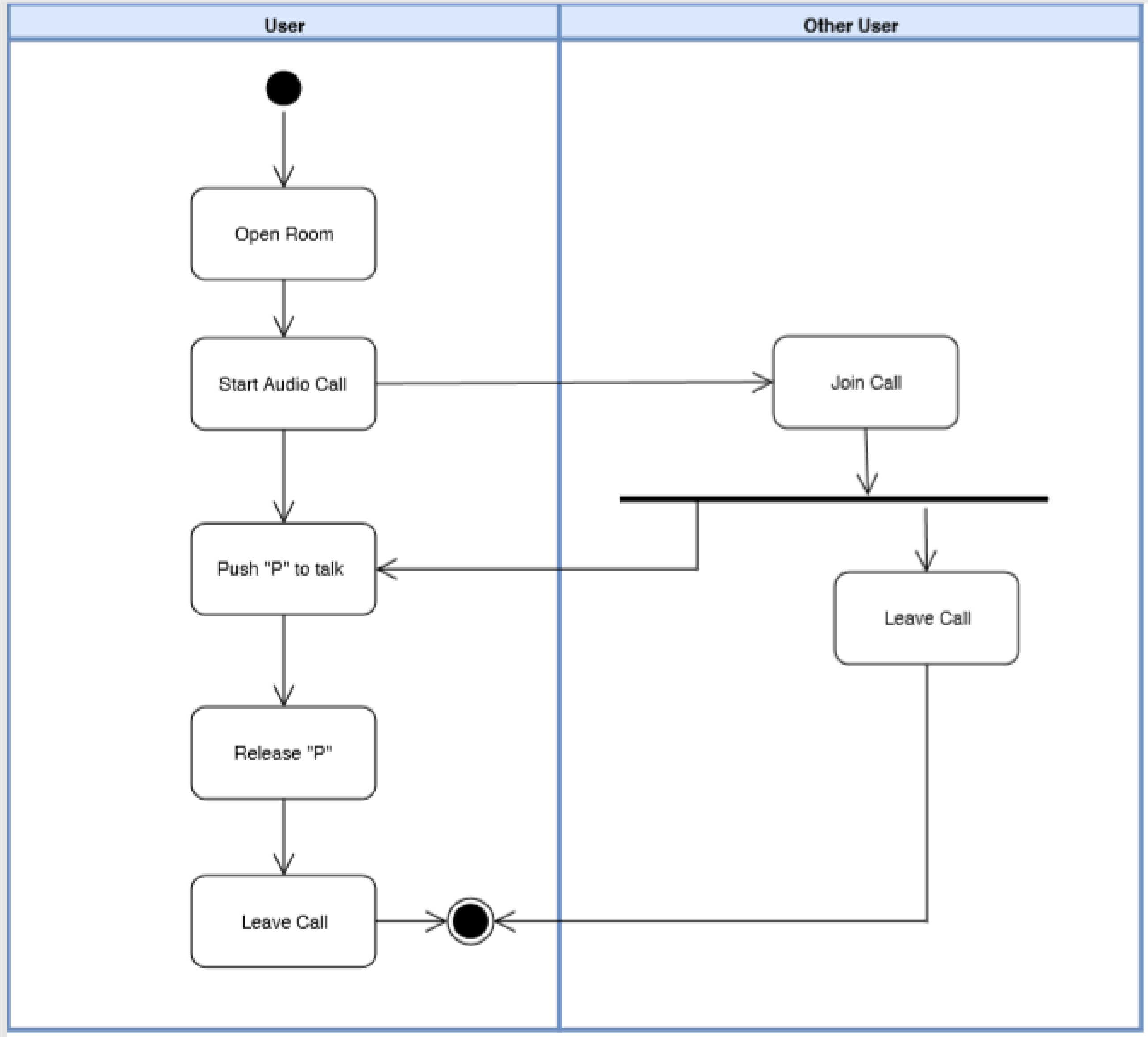


Figure 13: Make Audio Call

3.2.12.1 Explanation

This activity diagram shows audio call in online room. The user (student or teacher) will start the audio call. The other users in the system can join that call if they want to. They will click on “join” to join the call. For speaking the users will press on “P. After the completion of what they have to speak, they will release “P”. The caller may end the call and the other users (who joined the call), may leave the call.

4. Design models along with descriptions

The applicable models may include:

4.1 Class Diagram

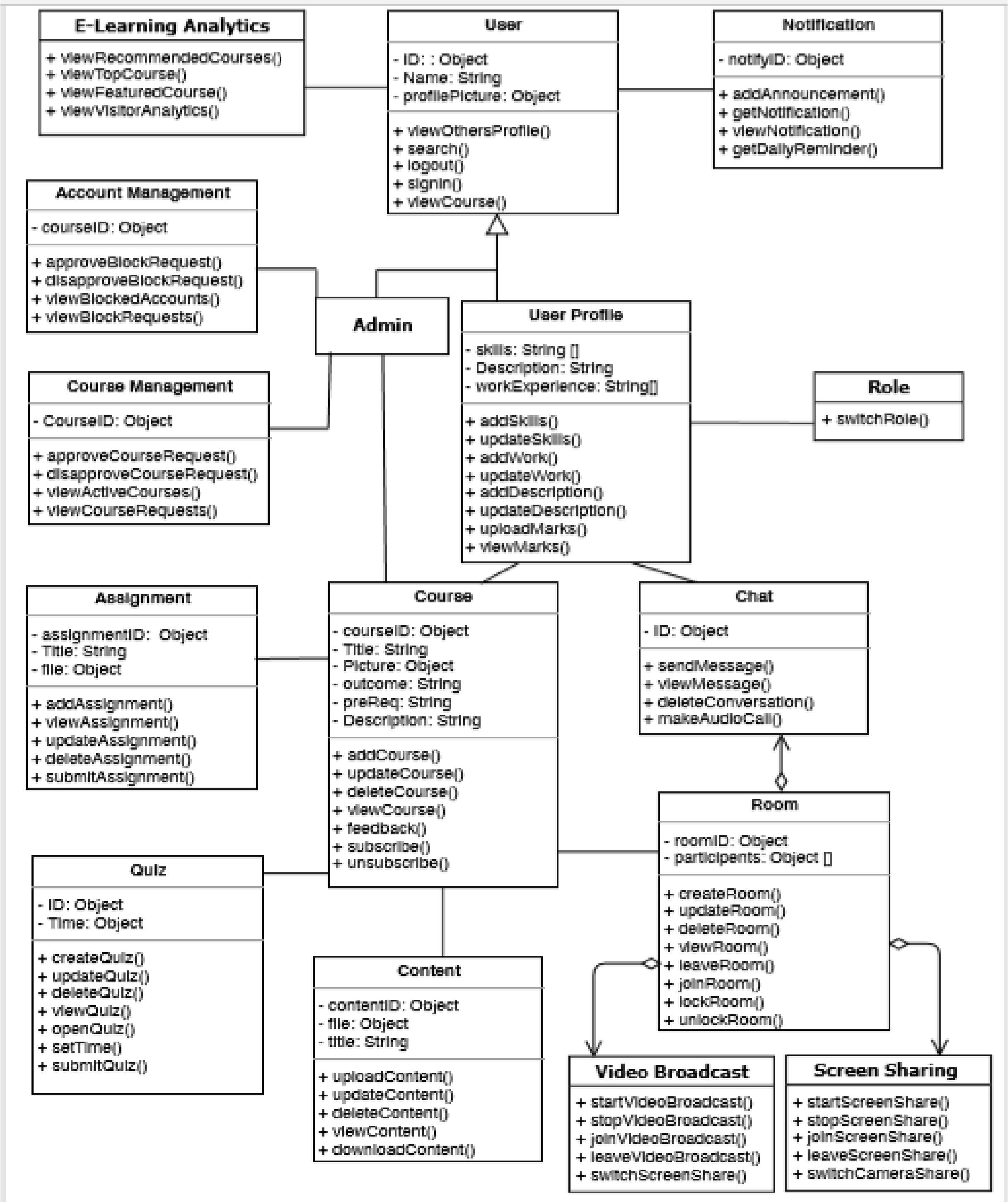


Figure 14: Class Diagram

4.2 Sequence Diagram

4.2.1 Sequence Diagram: Add and Approve Course

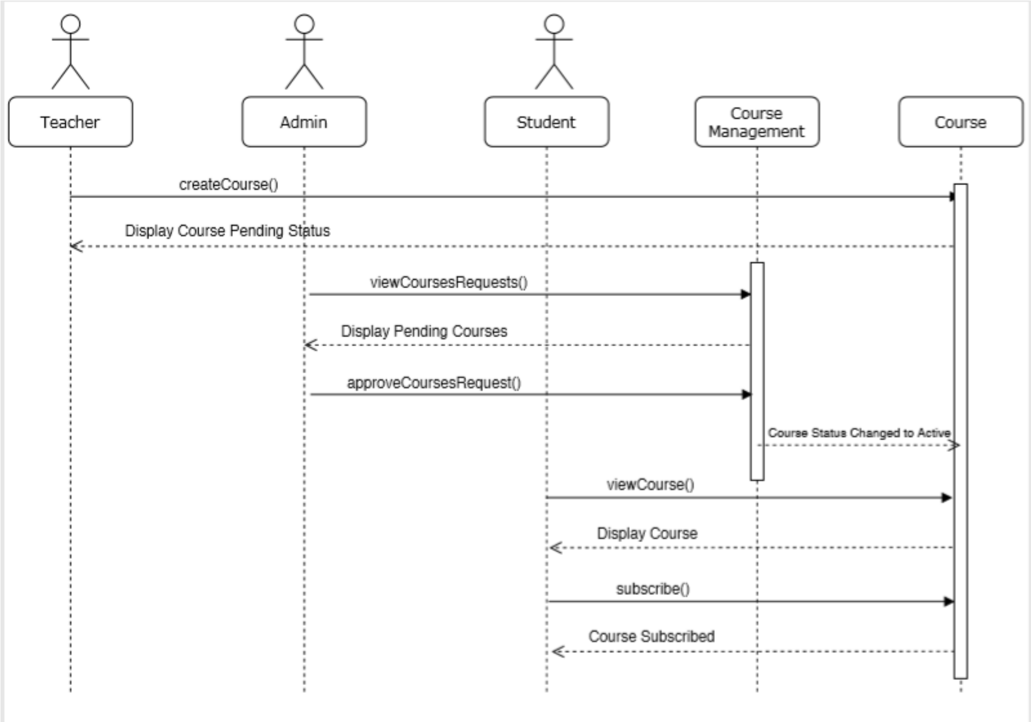


Figure 15: Add And Approve Course

4.2.2 Sequence Diagram: Room Chat

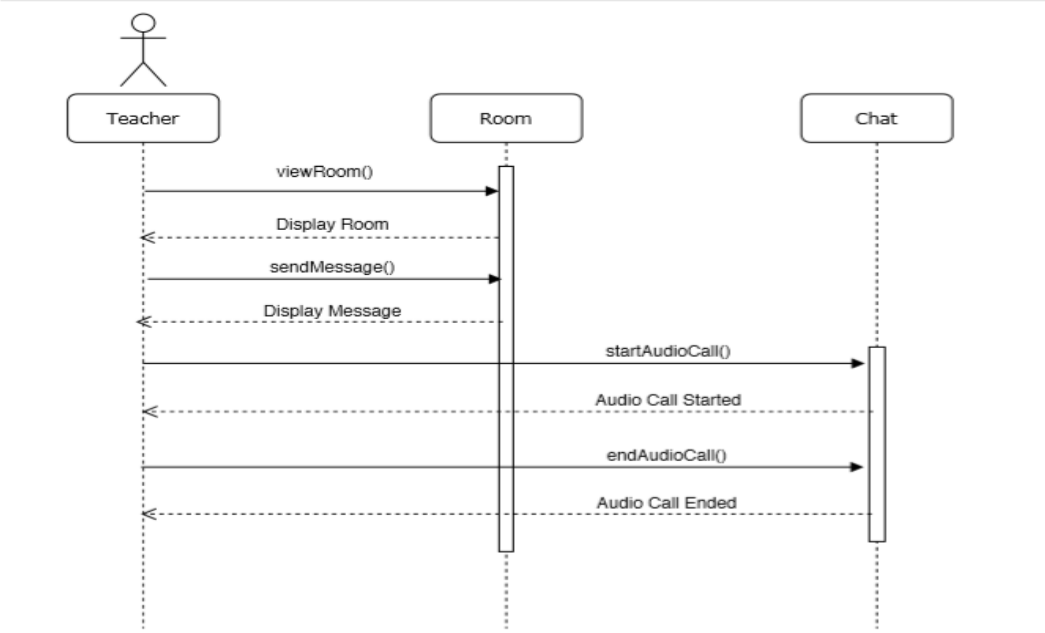


Figure16: Room Chat

4.2.3 Sequence Diagram: Admin

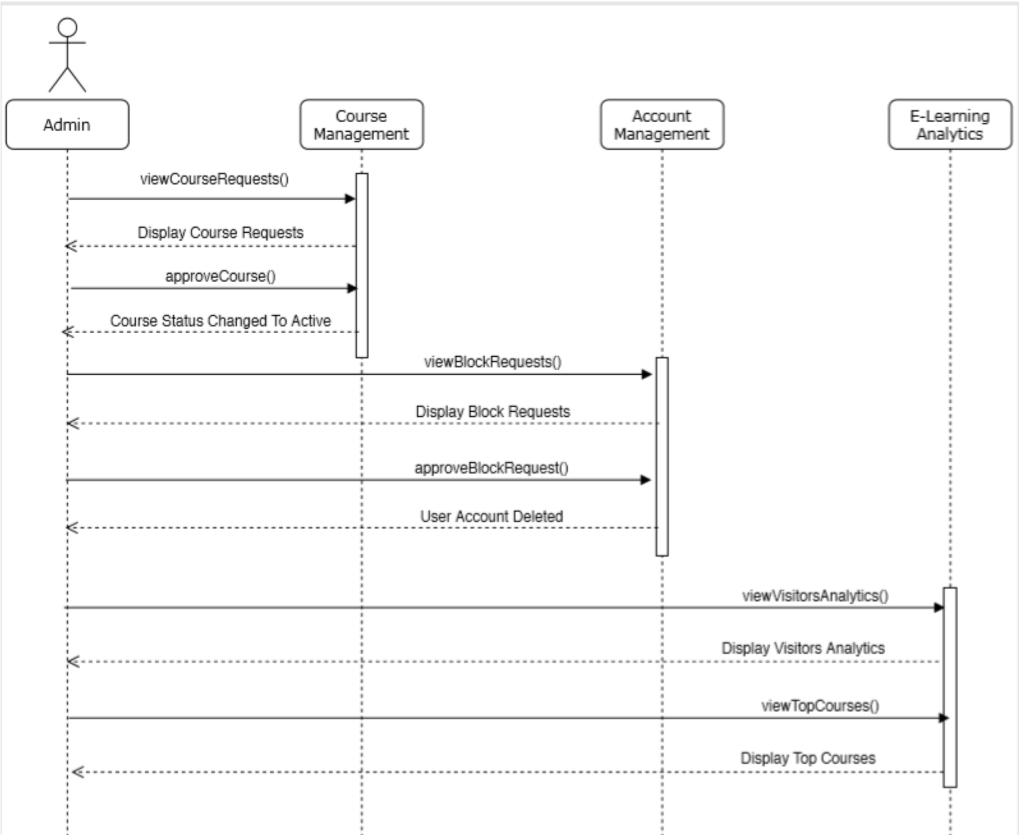


Figure 17: Admin

4.2.4 Sequence Diagram: Assignment

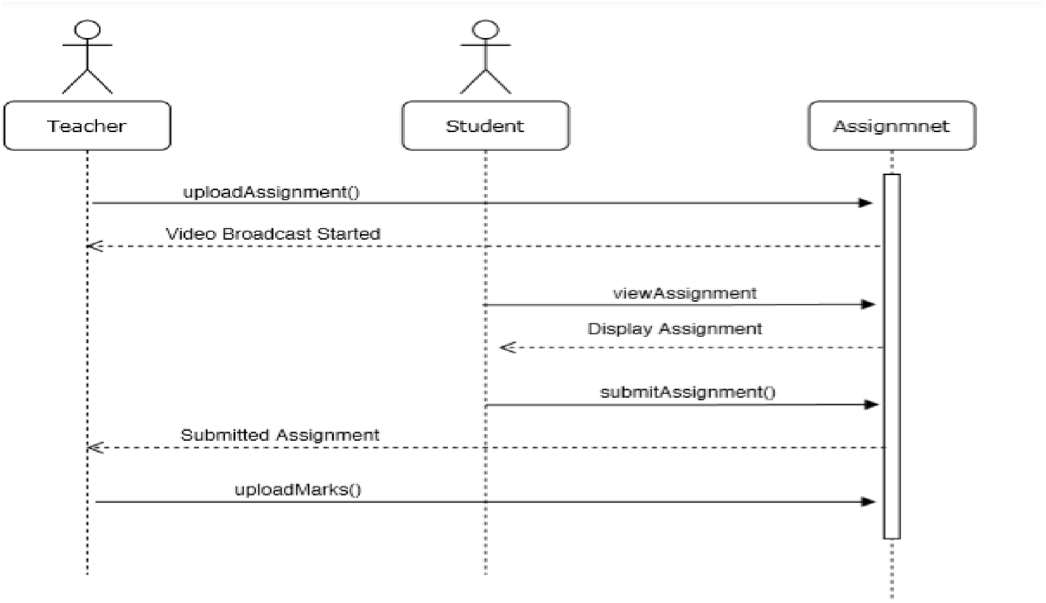


Figure 18: Assignment

4.2.5 Sequence Diagram: Video Broadcast

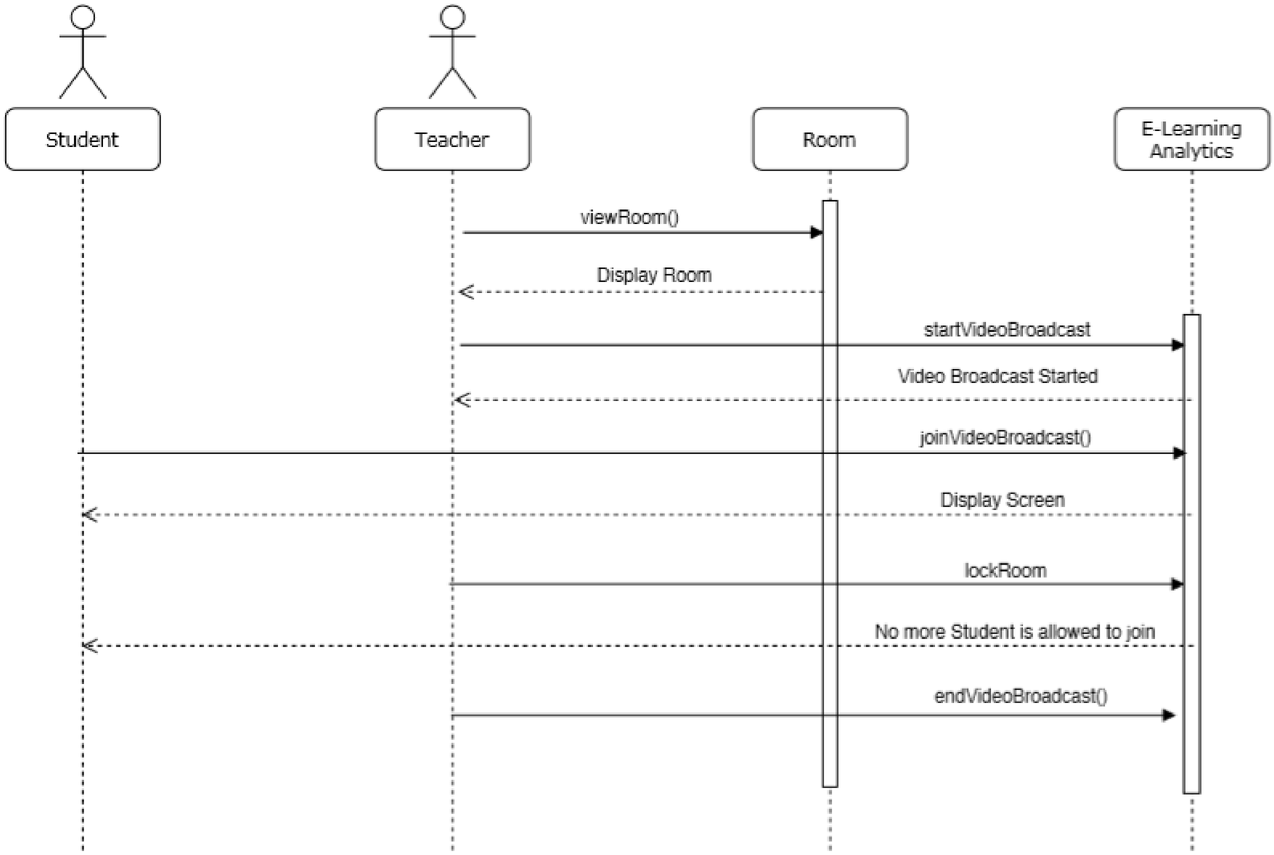
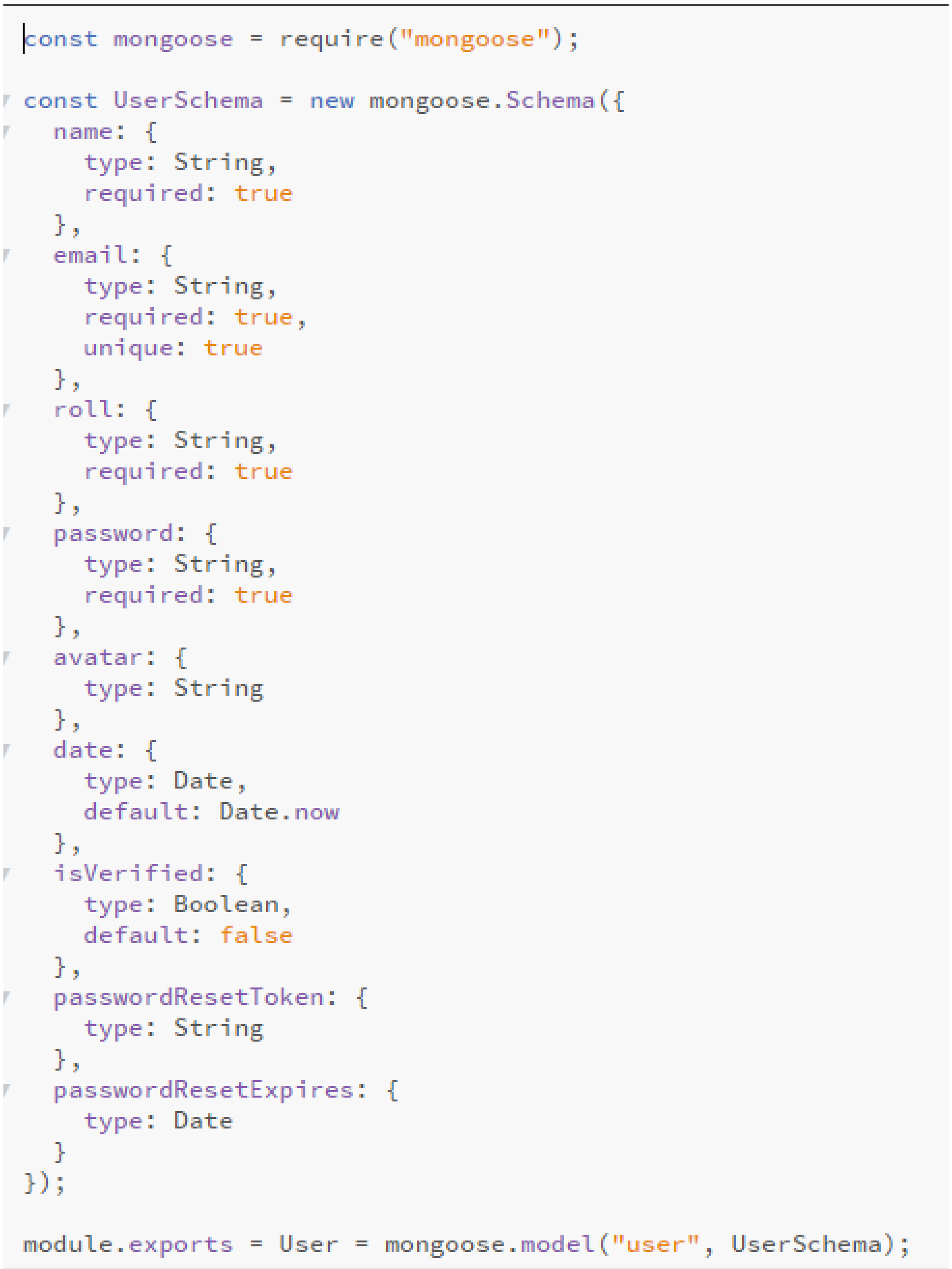


Figure 19: Video Broadcast

5. Data design

5.1 Data dictionary

5.1.1 Users



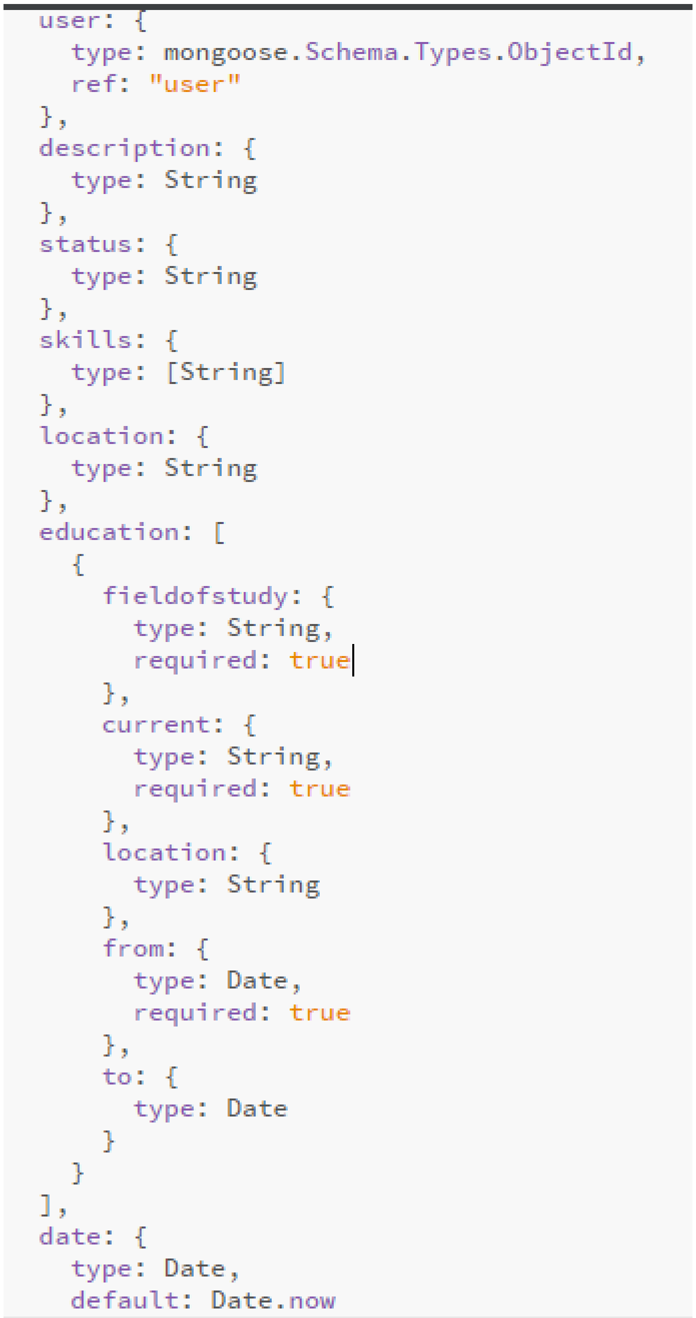
```
const mongoose = require("mongoose");

const UserSchema = new mongoose.Schema({
  name: {
    type: String,
    required: true
  },
  email: {
    type: String,
    required: true,
    unique: true
  },
  roll: {
    type: String,
    required: true
  },
  password: {
    type: String,
    required: true
  },
  avatar: {
    type: String
  },
  date: {
    type: Date,
    default: Date.now
  },
  isVerified: {
    type: Boolean,
    default: false
  },
  passwordResetToken: {
    type: String
  },
  passwordResetExpires: {
    type: Date
  }
});

module.exports = User = mongoose.model("user", UserSchema);
```

Figure 20: Data Dictionary of Users

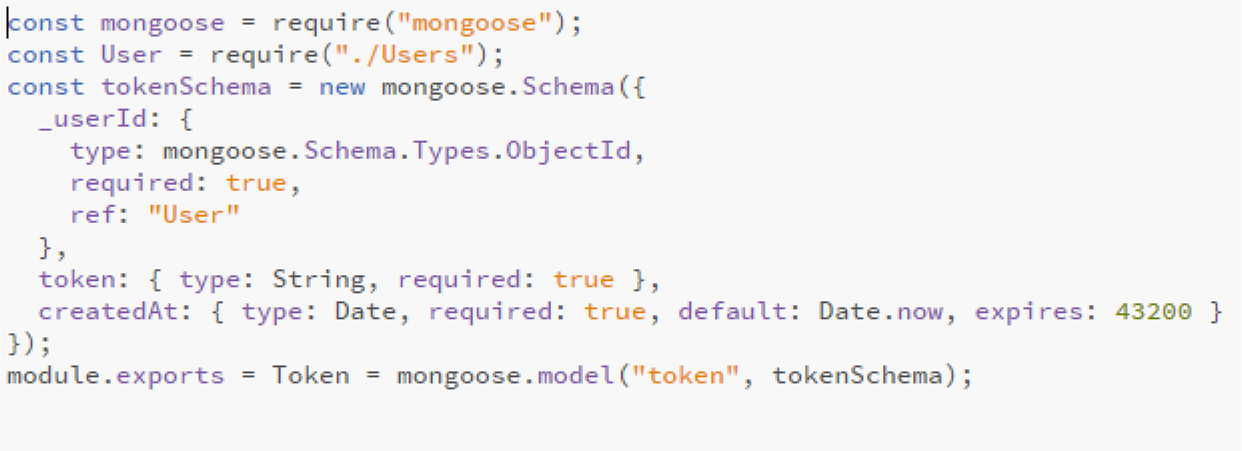
5.1.2 Profile



```
  user: {
    type: mongoose.Schema.Types.ObjectId,
    ref: "user"
  },
  description: {
    type: String
  },
  status: {
    type: String
  },
  skills: {
    type: [String]
  },
  location: {
    type: String
  },
  education: [
    {
      fieldofstudy: {
        type: String,
        required: true
      },
      current: {
        type: String,
        required: true
      },
      location: {
        type: String
      },
      from: {
        type: Date,
        required: true
      },
      to: {
        type: Date
      }
    }
  ],
  date: {
    type: Date,
    default: Date.now
```

Figure 21: Data Dictionary of Profile

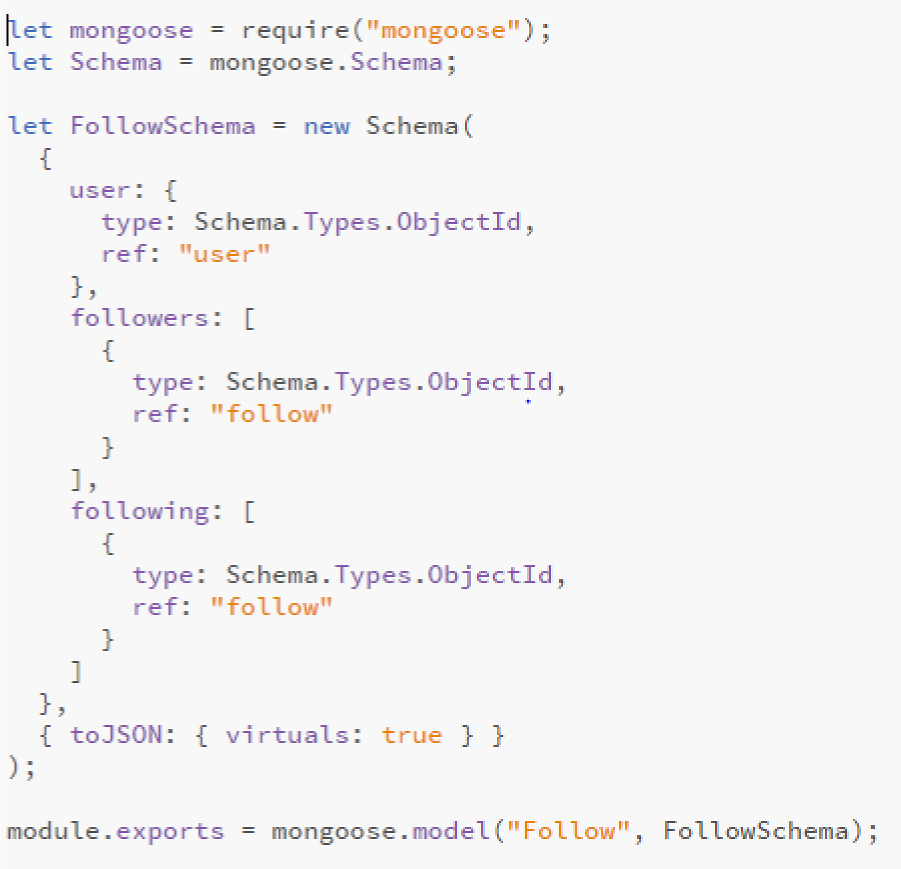
5.1.3 Token



```
const mongoose = require("mongoose");
const User = require("./Users");
const tokenSchema = new mongoose.Schema({
  _userId: {
    type: mongoose.Schema.Types.ObjectId,
    required: true,
    ref: "User"
  },
  token: { type: String, required: true },
  createdAt: { type: Date, required: true, default: Date.now, expires: 43200 }
});
module.exports = Token = mongoose.model("token", tokenSchema);
```

Figure 22: Data Dictionary of Token

5.1.4 Follow



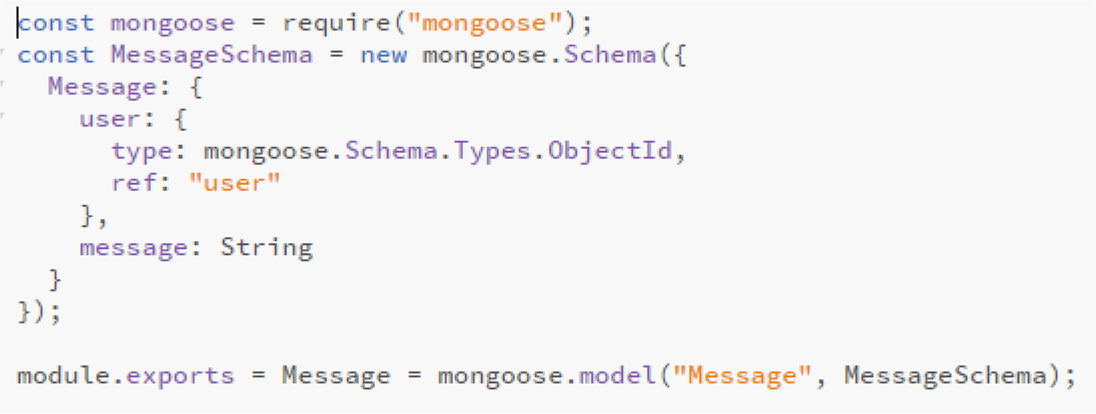
```
let mongoose = require("mongoose");
let Schema = mongoose.Schema;

let FollowSchema = new Schema(
  {
    user: {
      type: Schema.Types.ObjectId,
      ref: "user"
    },
    followers: [
      {
        type: Schema.Types.ObjectId,
        ref: "follow"
      }
    ],
    following: [
      {
        type: Schema.Types.ObjectId,
        ref: "follow"
      }
    ]
  },
  { toJSON: { virtuals: true } }
);

module.exports = mongoose.model("Follow", FollowSchema);
```

Figure 23: Data Dictionary of Follow

5.1.5 Message

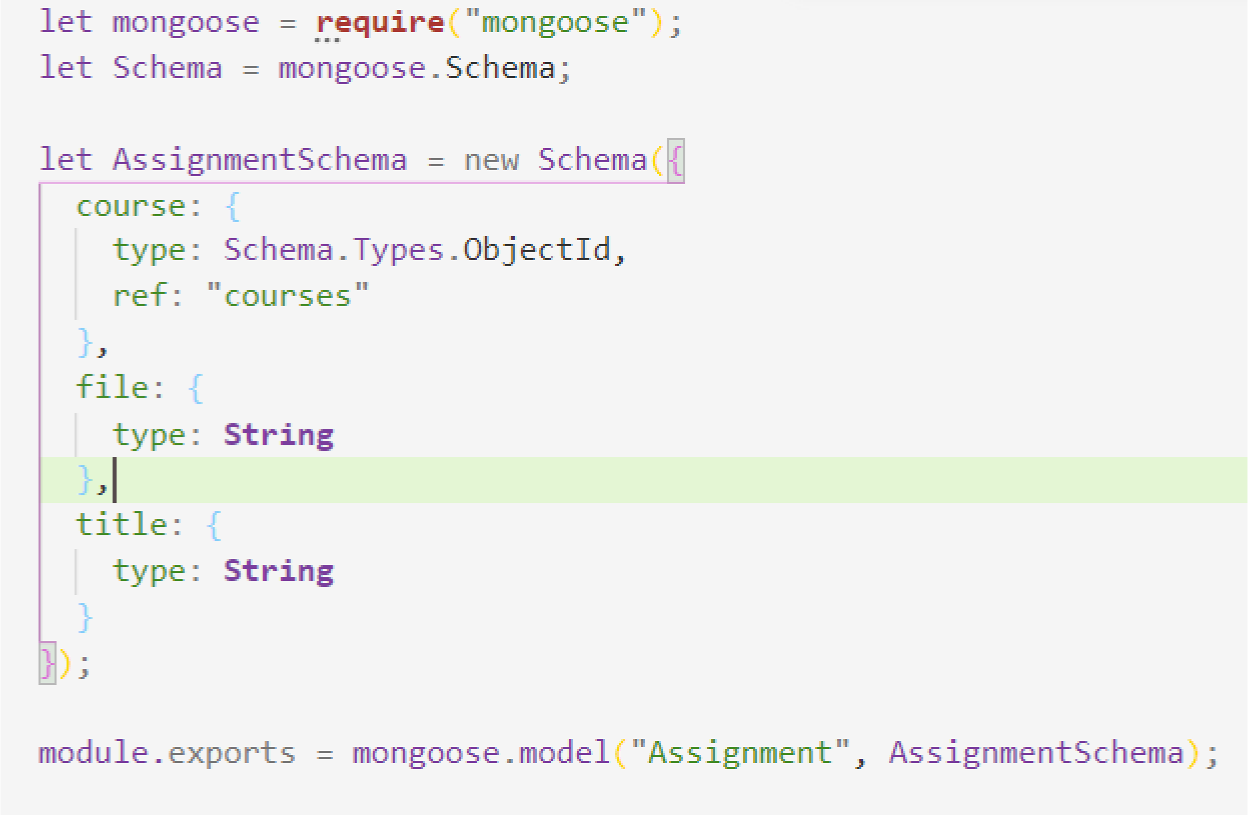


```
const mongoose = require("mongoose");
const MessageSchema = new mongoose.Schema({
  Message: {
    user: {
      type: mongoose.Schema.Types.ObjectId,
      ref: "user"
    },
    message: String
  }
});

module.exports = Message = mongoose.model("Message", MessageSchema);
```

Figure 24: Data Dictionary of Message

5.1.6 Assignment



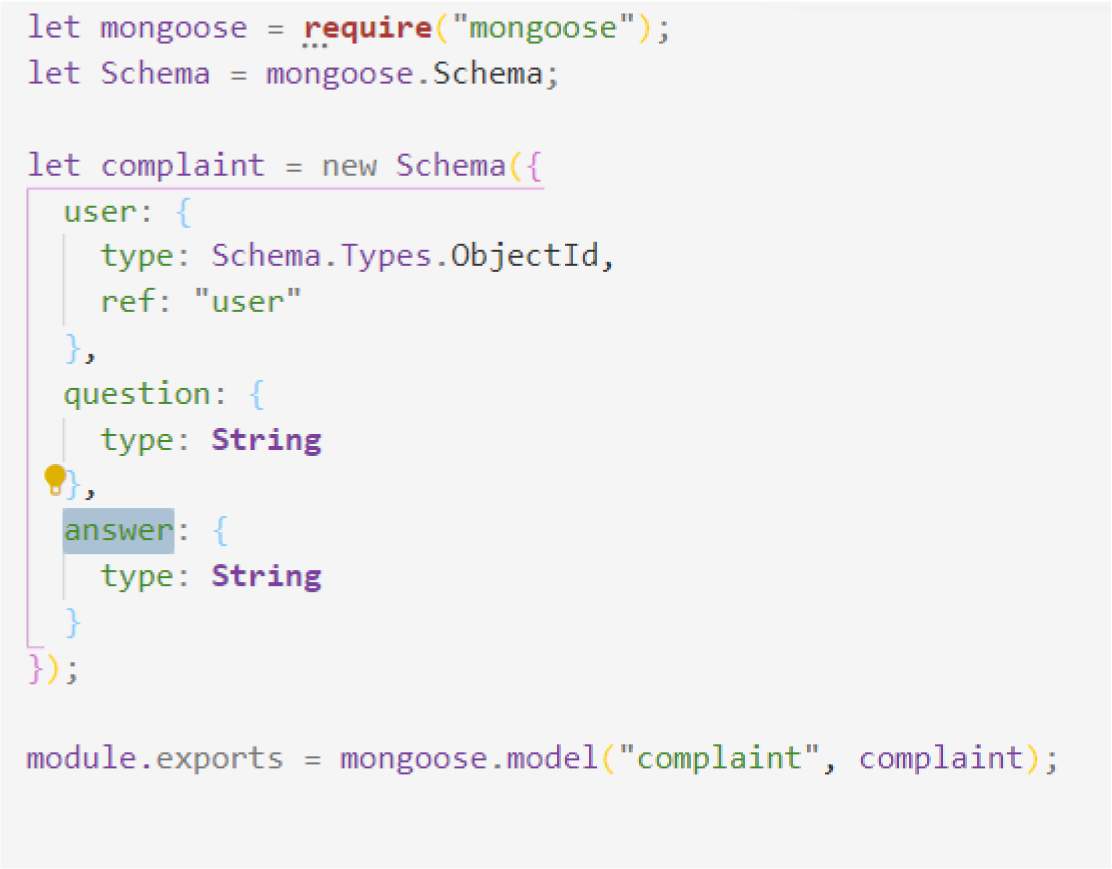
```
let mongoose = require("mongoose");
let Schema = mongoose.Schema;

let AssignmentSchema = new Schema({
  course: {
    type: Schema.Types.ObjectId,
    ref: "courses"
  },
  file: {
    type: String
  },
  title: {
    type: String
  }
});

module.exports = mongoose.model("Assignment", AssignmentSchema);
```

Figure 25: Data Dictionary of Assignment

5.1.7 Complaint



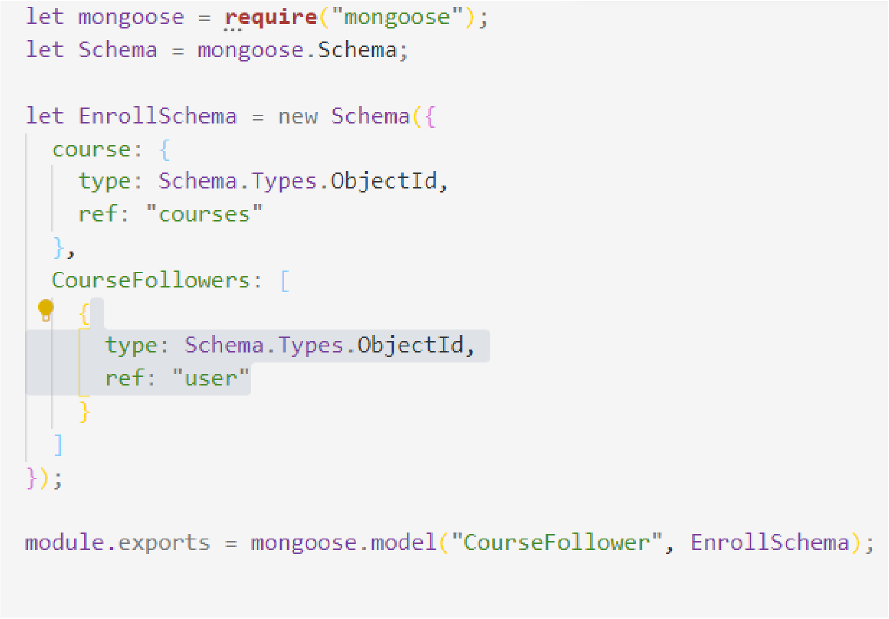
```
let mongoose = require("mongoose");
let Schema = mongoose.Schema;

let complaint = new Schema({
  user: {
    type: Schema.Types.ObjectId,
    ref: "user"
  },
  question: {
    type: String
  },
  answer: {
    type: String
  }
});

module.exports = mongoose.model("complaint", complaint);
```

Figure 26: Data Dictionary of Complaint

5.1.8 Enroll



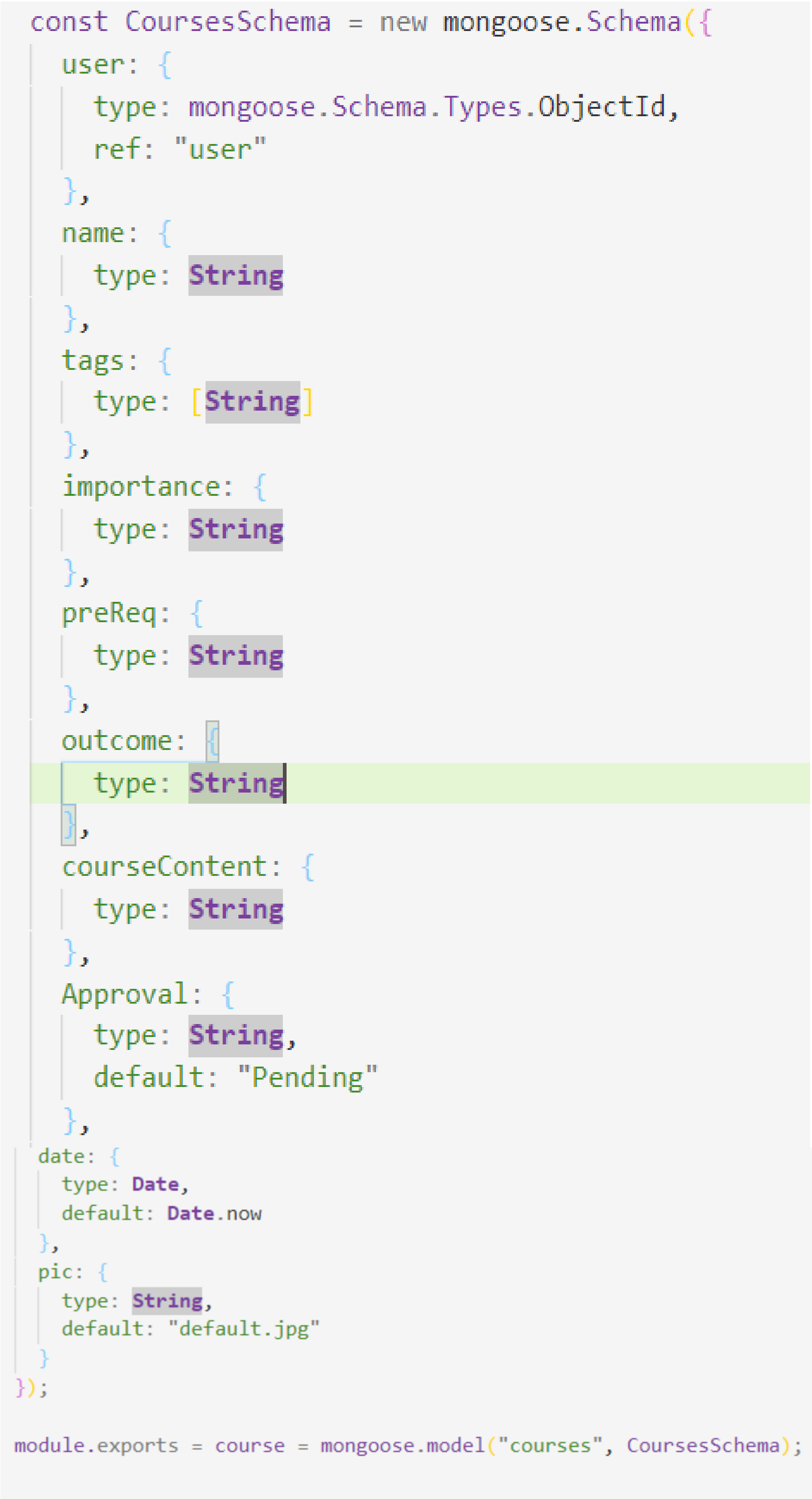
```
let mongoose = require("mongoose");
let Schema = mongoose.Schema;

let EnrollSchema = new Schema({
  course: {
    type: Schema.Types.ObjectId,
    ref: "courses"
  },
  CourseFollowers: [
    {
      type: Schema.Types.ObjectId,
      ref: "user"
    }
  ]
});

module.exports = mongoose.model("CourseFollower", EnrollSchema);
```

Figure 27: Enroll

5.1.9 Course

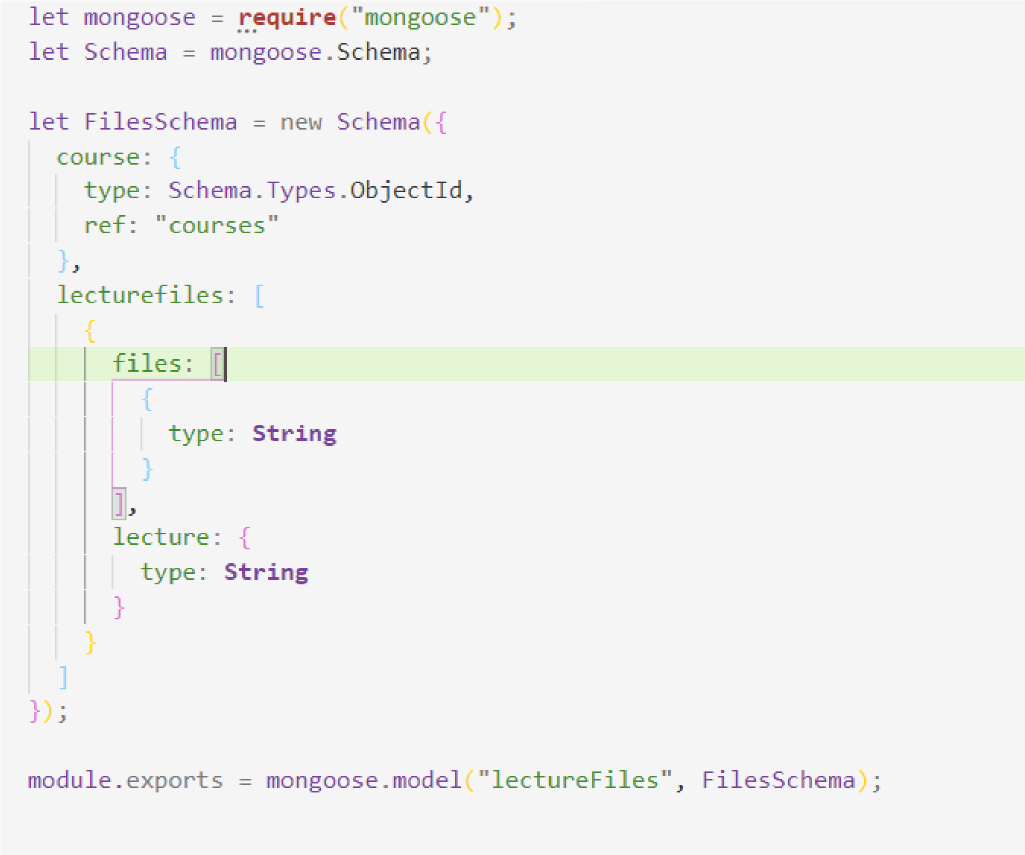


```
const CoursesSchema = new mongoose.Schema({
  user: {
    type: mongoose.Schema.Types.ObjectId,
    ref: "user"
  },
  name: {
    type: String
  },
  tags: {
    type: [String]
  },
  importance: {
    type: String
  },
  preReq: {
    type: String
  },
  outcome: {
    type: String
  },
  courseContent: {
    type: String
  },
  Approval: {
    type: String,
    default: "Pending"
  },
  date: {
    type: Date,
    default: Date.now
  },
  pic: {
    type: String,
    default: "default.jpg"
  }
});

module.exports = course = mongoose.model("courses", CoursesSchema);
```

Figure 28: Data Dictionary of Course

5.1.10 Files



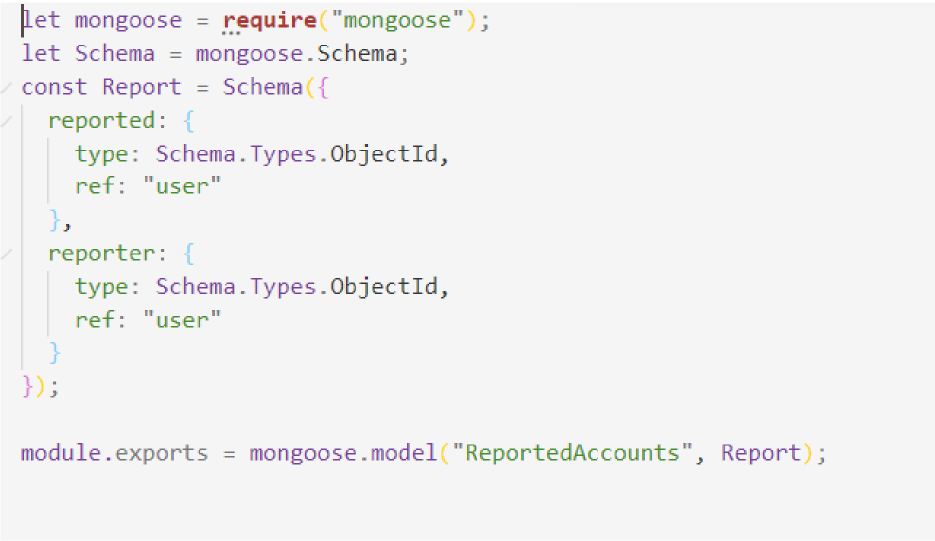
```
let mongoose = require("mongoose");
let Schema = mongoose.Schema;

let FilesSchema = new Schema({
  course: {
    type: Schema.Types.ObjectId,
    ref: "courses"
  },
  lecturefiles: [
    {
      files: [
        {
          type: String
        }
      ],
      lecture: {
        type: String
      }
    }
  ]
});

module.exports = mongoose.model("lectureFiles", FilesSchema);
```

Figure 29: Data Dictionary of Files

5.1.11 Report

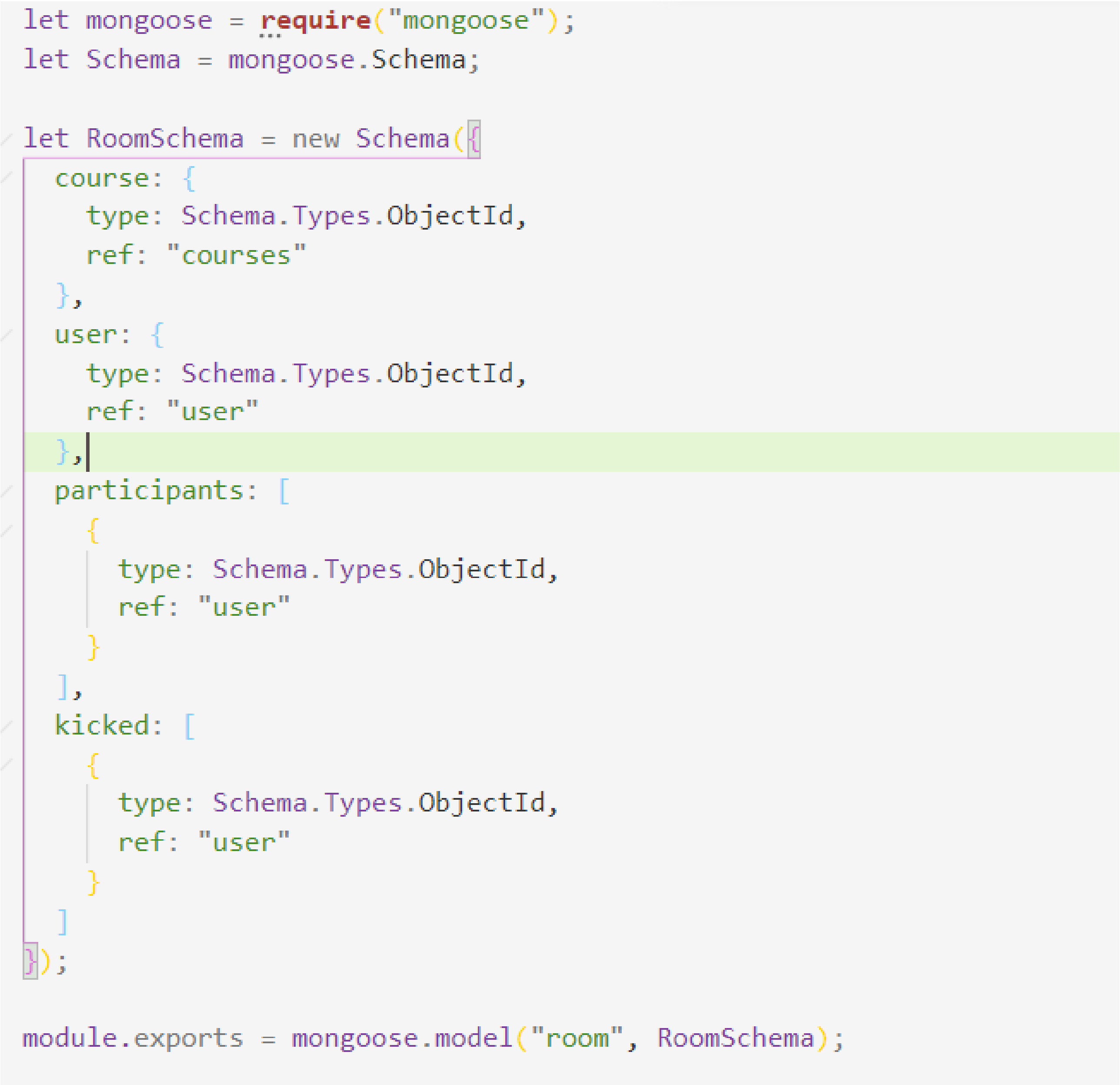


```
let mongoose = require("mongoose");
let Schema = mongoose.Schema;
const Report = Schema({
  reported: {
    type: Schema.Types.ObjectId,
    ref: "user"
  },
  reporter: {
    type: Schema.Types.ObjectId,
    ref: "user"
  }
});

module.exports = mongoose.model("ReportedAccounts", Report);
```

Figure 30: Data Dictionary of Report

5.1.12 Room



```
let mongoose = require("mongoose");
let Schema = mongoose.Schema;

let RoomSchema = new Schema({
  course: {
    type: Schema.Types.ObjectId,
    ref: "courses"
  },
  user: {
    type: Schema.Types.ObjectId,
    ref: "user"
  },
  participants: [
    {
      type: Schema.Types.ObjectId,
      ref: "user"
    }
  ],
  kicked: [
    {
      type: Schema.Types.ObjectId,
      ref: "user"
    }
  ]
});

module.exports = mongoose.model("room", RoomSchema);
```

Figure 31: Data Dictionary of Room

6. Algorithm & Implementation

6.1 Change Password

6.1.1 Input

Current Password
User id

New password

6.1.2 Steps

1. Send input parameters with axios to nodejs
2. Get input parameters from axios
3. Fetch the user with the user id.
4. Check if the user against that user id exists
5. Check if current password entered matches to the password of that user
6. Update Password
7. Update user

6.1.3 Output

Status: Password Changed

6.2 Forgot Password

6.2.1 Input

Email
New password

6.2.2 Steps

1. send input parameters with axios to nodejs
2. get input parameters from axios
3. Fetch the user with that email
4. Check if the user exists, then generate a random token of 20 characters.
5. Save the token against that user id
6. Send that token in the email to that user
7. User will enter the token sent to him
8. Check if the token entered is valid.
9. Check if the token exists against that user
10. Reset Password

6.2.3 Output

Status: Password Changed

6.3 Update Profile

6.3.1 Input

User Inputs in all fields in profile

6.3.2 Steps

1. Send input parameters with axios to nodejs
2. Get input parameters from axios
3. Fetch the user with user id
4. Update profile

6.3.3 Output

Status: Profile Updated

6.4 Send Message

6.4.1 Input

Message
Sender ID
Receiver ID
Room ID

6.4.2 Steps

1. Socket listen for the server
2. Server is available
3. Socket connected to the server
4. Socket on event join send message SenderID has joined the chat in room
5. Socket on event message send, forward the message to the server for that senderID
6. Socket broadcast message

6.4.3 Output

Message

6.5 Receive Message

6.5.1 Input

Message
Receiver ID
Room ID

6.5.2 Steps

1. Create socket
2. Socket on connection event, make the connection async
3. Socket on broadcast message send to receiver ID in room.

6.5.3 Output

Message

6.6 Video Broadcast

6.6.1 Input

access to webcam and microphone
userID
RoomID

6.6.2 Steps

1. Create a signalling server for the socket which will process coordinating communication
2. Generate a random RoomID with Math.Random
3. Emit socket to the new-channel request if the user selects create Room
4. Else if user clicks on join Broadcast, Socket will emit the user to the already created room
5. Upon the call of sendFunction socket will start emitting the data
6. onRemoteStream function is then called when a user clicks on join
7. It gets the data from the hostStream and show the video to the broadcast users
8. When room is found UI will be updated
9. List of Room will be displayed below
10. When the user creates a room his webcam stream is captured
11. Upon the permission to access the webcam boradcasting will start
12. Otherwise if the permission is denied an error message will be diplayed

6.6.3 Output

Participants
BroadcastStream

7. Software requirements traceability matrix

Req. Number	Design Component	Ref. Item	Component Items
FR-001	Class Diagram	User	Display Registration Form
FR-002	Class Diagram	User	Mismatch password and confirm password
FR-003	Class Diagram	User	Already registered email
FR-004	Class Diagram	User	Display Login Form
FR-005	Class Diagram	User	Invalid email or password
FR-006	Class Diagram	User	Send Passcode Via Email

FR-007	Class Diagram	User	Display Verify Account Form
FR-008	Class Diagram	User	Wrong passcode entered
FR-009	Class Diagram	User	Change Password
FR-010	Class Diagram	User	Display Change Password Form
FR-011	Class Diagram	User	Incorrect Current Password Entered
FR-012	Class Diagram	User	Set New Password
FR-013	Class Diagram	User	Send Passcode
FR-014	Class Diagram	User	Wrong Passcode Entered
FR-015	Class Diagram	User Profile	Upload Picture
FR-016	Class Diagram	User Profile	Picture Format Validation
FR-017	Class Diagram	User Profile	Picture Size Validation
FR-018	Class Diagram	User Profile	Delete Picture
FR-019	Class Diagram	User Profile	Change Image
FR-020	Class Diagram	User Profile	Display Profile
FR-021	Class Diagram	User Profile	Display Education Information Form
FR-022	Class Diagram	User Profile	Save Education Information
FR-023	Class Diagram	User Profile	Delete Education Information
FR-024	Class Diagram	User Profile	Display Work Experience Form
FR-025	Class Diagram	User Profile	Save Work Experience
FR-026	Class Diagram	User Profile	Delete Work Experience
FR-027	Class Diagram	User Profile	Display Personal Information Form
FR-028	Class Diagram	User Profile	Save Personal Information
FR-029	Class Diagram	User Profile	Delete Personal Information
FR-030	Class Diagram	User Profile	Set Privacy
FR-031	Class Diagram	User Profile	Edit Education

			Information
FR-032	Class Diagram	User Profile	Edit Work Experience
FR-033	Class Diagram	User Profile	Edit Personal Information
FR-034	Class Diagram	User Profile	Update Privacy Settings
FR-035	Class Diagram	User	Logout
FR-036	Class Diagram	User Profile	Report User
FR-037	Class Diagram	User Profile and Account Management	Report Request Sent to Admin
FR-038	Class Diagram	User Profile	Delete Account
FR-039	Class Diagram	Assignment	Upload Assignment
FR-040	Class Diagram	Assignment	Maximum 4 Assignments to be posted
FR-041	Class Diagram	Assignment	File Size Validation
FR-042	Class Diagram	Assignment	File Format Validation
FR-043	Class Diagram	Assignment	Display Assignment List
FR-044	Class Diagram	Assignment	Download Assignment
FR-045	Class Diagram	Assignment	Delete Assignment
FR-046	Class Diagram	Assignment	Edit Assignment
FR-047	Class Diagram	Assignment	Submit assignment
FR-048	Class Diagram	Quiz	Upload Quiz
FR-049	Class Diagram	Quiz	Set Quiz Time Duration
FR-050	Class Diagram	Quiz	Delete Quiz
FR-051	Class Diagram	Quiz	Edit Quiz
FR-052	Class Diagram	Quiz	Submit Quiz
FR-053	Class Diagram	Quiz	Auto Check Quiz
FR-054	Class Diagram	Quiz	Auto Submit Quiz
FR-055	Class Diagram	Quiz	Start Timer
FR-056	Class Diagram	Quiz	Stop Timer
FR-057	Class Diagram	User Profile	Upload Marks
FR-058	Class Diagram	User Profile	Display Marks
FR-059	Class Diagram	User Profile	Edit Marks
FR-060	Class Diagram	Content	Upload Lecture Files

FR-061	Class Diagram	Content	Lecture File Size Validation
FR-062	Class Diagram	Content	Lecture File Format Validation
FR-063	Class Diagram	Content	Display Lecture File
FR-064	Class Diagram	Content	Download Lecture File
FR-065	Class Diagram	Content	Delete Lecture File
FR-066	Class Diagram	Course	Save Video
FR-067	Class Diagram	Content	Watch Video
FR-068	Class Diagram	Content	Download Video
FR-069	Class Diagram	Content	Upload Video
FR-070	Class Diagram	Content	Video Size Validation
FR-071	Class Diagram	Content	Delete Video
FR-072	Class Diagram	Course	Subscribe Course
FR-073	Class Diagram	Course	Unsubscribe Course
FR-074	Class Diagram	Course	Rate Teacher
FR-075	Class Diagram	Course	Review Teacher
FR-076	Class Diagram	Course	Rate Course
FR-077	Class Diagram	Course	Review Course
FR-078	Class Diagram	Course	Add Course
FR-079	Class Diagram	Course	Add Course Details
FR-080	Class Diagram	Course	Course Status Pending
FR-081	Class Diagram	Course	Edit Course
FR-082	Class Diagram	Course	Delete Course
FR-083	Class Diagram	Course	Display Course
FR-084	Class Diagram	Course	Display Subscribers
FR-085	Class Diagram	User Profile	Display Profile
FR-086	Class Diagram	Room	Display Rooms
FR-087	Class Diagram	Room	Join Room
FR-088	Class Diagram	Room	Leave Room
FR-089	Class Diagram	Room	Create Room
FR-090	Class Diagram	Room	Edit Room

FR-091	Class Diagram	Room	Delete Room
FR-092	Class Diagram	Room	Display The Room
FR-093	Class Diagram	Room	Remove Student From Room
FR-094	Class Diagram	User	Display Search Results
FR-095	Class Diagram	User Profile	Send Complaint
FR-096	Class Diagram	Quiz	Open Quiz
FR-097	Class Diagram	Course Management	Approve Course
FR-098	Class Diagram	Course Management	Disapprove Course
FR-099	Class Diagram	Account Management	Block User
FR-100	Class Diagram	Course Management	Display Active Courses
FR-101	Class Diagram	Course Management	Display Pending Course Request
FR-102	Class Diagram	Account Management	Display Pending Account Request
FR-103	Class Diagram	Admin	Display Complaint
FR-104	Class Diagram	Admin	Send Reply
FR-105	Class Diagram	User	Add Reply to FAQ
FR-106	Class Diagram	Chat	Send Message
FR-107	Class Diagram	Chat	Send Message In Chat
FR-108	Class Diagram	Chat	Delete Conversation
FR-109	Class Diagram	Chat	Display Full Conversation
FR-110	Class Diagram	Notification	Add Announcement
FR-111	Class Diagram	Notification	Send Announcement As a Notification
FR-112	Class Diagram	Notification	Display Notification
FR-113	Class Diagram	Notification	Delete Announcement
FR-114	Class Diagram	Notification	Display Events
FR-115	Class Diagram	Chat	Start Audio Call
FR-116	Class Diagram	Chat	Join Call
FR-117	Class Diagram	Chat	Leave Call
FR-118	Class Diagram	Chat	End Call

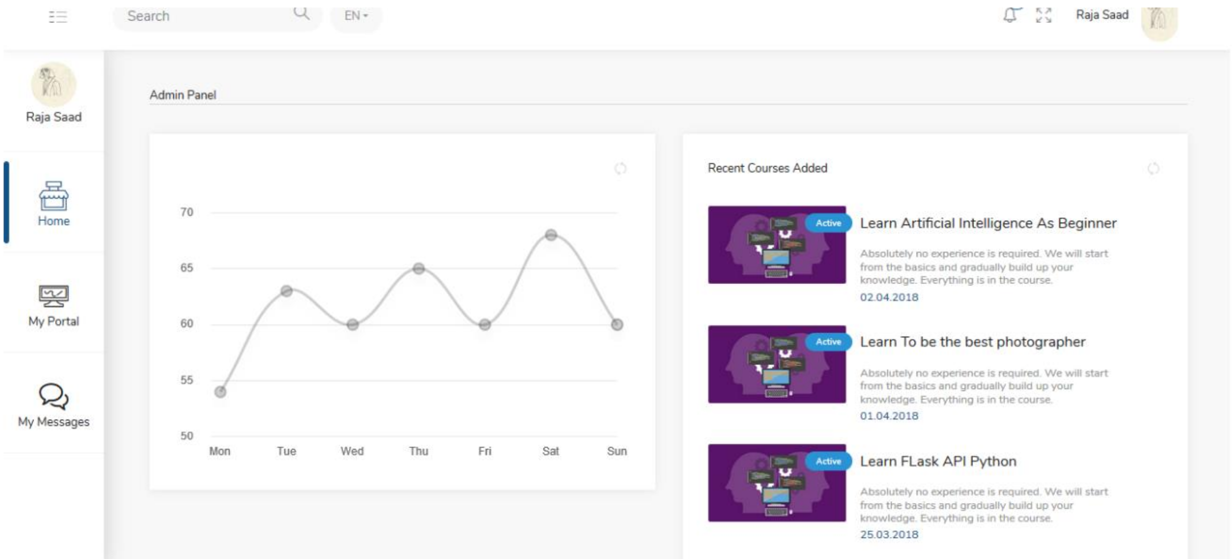
FR-119	Class Diagram	Call	Reject Call
FR-120	Class Diagram	Call	Accept Call
FR-121	Class Diagram	Video Broadcast	Go Live
FR-122	Class Diagram	Video Broadcast	End Video Call
FR-123	Class Diagram	Video Broadcast	Join Broadcast
FR-124	Class Diagram	Video Broadcast	Leave Broadcast
FR-125	Class Diagram	Video Broadcast	Switch To Screen Sharing
FR-126	Class Diagram	Video Broadcast	Auto Zoom
FR-127	Class Diagram	Video Broadcast	Input Video Frame
FR-128	Class Diagram	Video Broadcast	Feature Extraction
FR-129	Class Diagram	Video Broadcast	Classification
FR-130	Class Diagram	Video Broadcast	Localization
FR-131	Class Diagram	Video Broadcast	Pre-processing Training Data
FR-132	Class Diagram	Video Broadcast	Action Label Database
FR-133	Class Diagram	Video Broadcast	Create Model
FR-134	Class Diagram	Video Broadcast	Feature Extraction
FR-135	Class Diagram	Video Broadcast	Compile Model
FR-136	Class Diagram	Video Broadcast	Input Video Frame
FR-137	Class Diagram	Video Broadcast	Segmentation
FR-138	Class Diagram	Video Broadcast	Action Classification
FR-139	Class Diagram	Video Broadcast	Zoom On Writing Board
FR-140	Class Diagram	Video Broadcast	Disable Auto Zoom
FR-141	Class Diagram	Screen Sharing	Start Screen Sharing
FR-142	Class Diagram	Screen Sharing	Join Screen Sharing
FR-143	Class Diagram	Screen Sharing	Leave Screen Sharing
FR-144	Class Diagram	Screen Sharing	Switch To Camera Sharing
FR-145	Class Diagram	Screen Sharing	End Screen Sharing
FR-146	Class Diagram	E-Learning Analytics	Display Recommended Courses
FR-147	Class Diagram	E-Learning Analytics	Display Trending Courses
FR-148	Class Diagram	E-Learning Analytics	Display Visitor’s

			Analytics
FR-149	Class Diagram	E-Learning Analytics	Display Top Courses

8. Human interface design

8.1 Screen images

8.1.1 Admin Panel



Learn Artificial Intelligence As Beginner

Absolutely no experience is required. We will start from the basics and gradually build up your knowledge. Everything is in the course.

02.04.2018

Learn To be the best photographer

Absolutely no experience is required. We will start from the basics and gradually build up your knowledge. Everything is in the course.

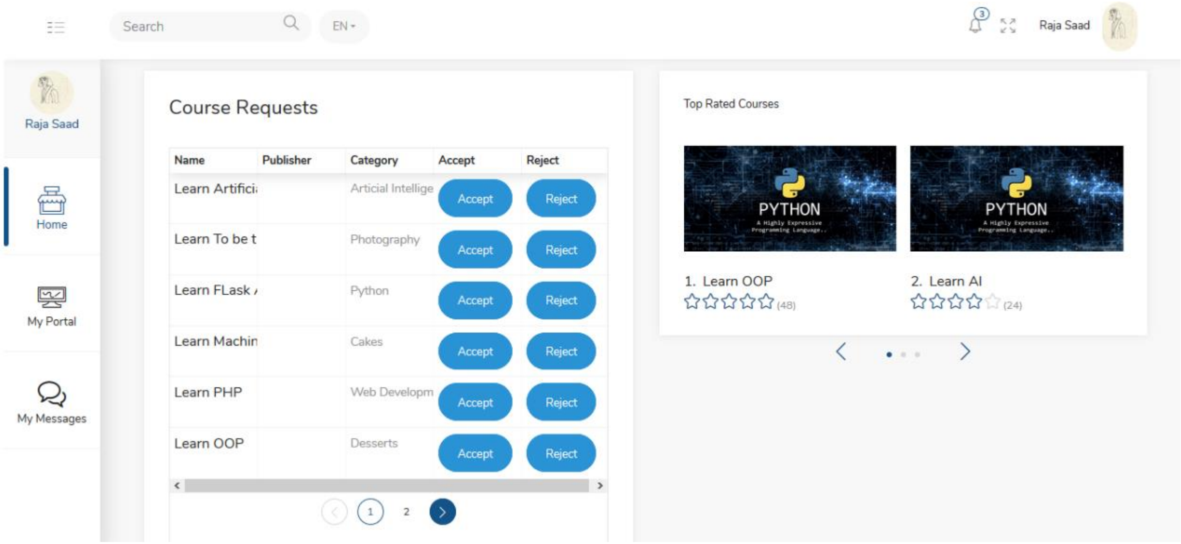
01.04.2018

Learn FLask API Python

Absolutely no experience is required. We will start from the basics and gradually build up your knowledge. Everything is in the course.

25.03.2018

Figure 15: Recent Added Courses Screen



Course Requests

Name	Publisher	Category	Accept	Reject
Learn Artifici		Articial Intellige	Accept	Reject
Learn To be t		Photography	Accept	Reject
Learn FLask /		Python	Accept	Reject
Learn Machin		Cakes	Accept	Reject
Learn PHP		Web Developm	Accept	Reject
Learn OOP		Desserts	Accept	Reject

Top Rated Courses

1. Learn OOP ☆☆☆☆☆ (48)
2. Learn AI ☆☆☆☆☆ (24)

Figure 16: Course Requests Screens

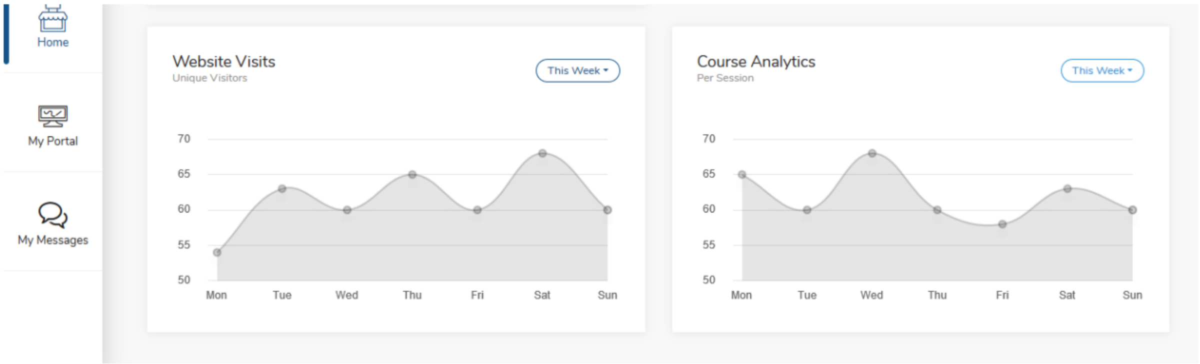
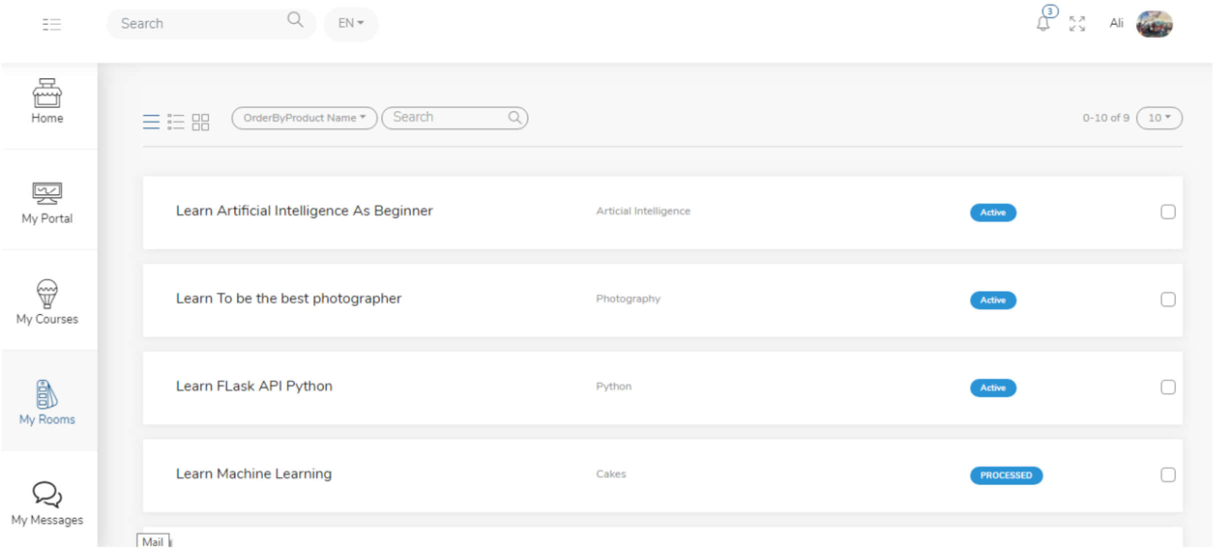


Figure 17: Visitor's Analytics Screen

8.1.2 Student: Rooms



Learn Artificial Intelligence As Beginner	Articial Intelligence	Active	□
Learn To be the best photographer	Photography	Active	□
Learn FLask API Python	Python	Active	□
Learn Machine Learning	Cakes	PROCESSED	□

Figure 18: Rooms List

8.1.3 Teacher: Courses

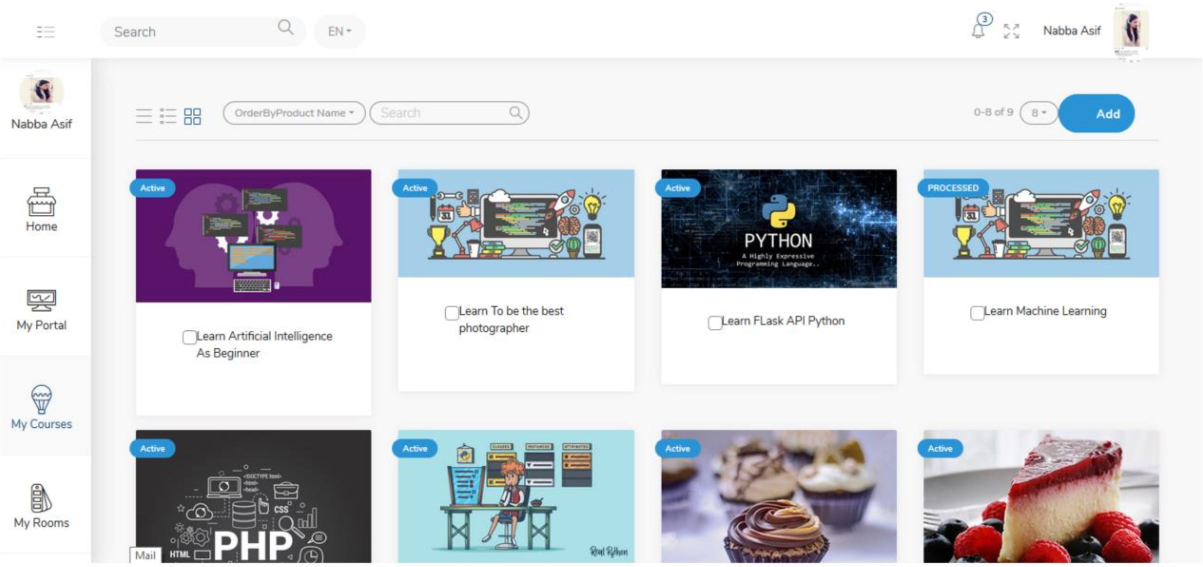
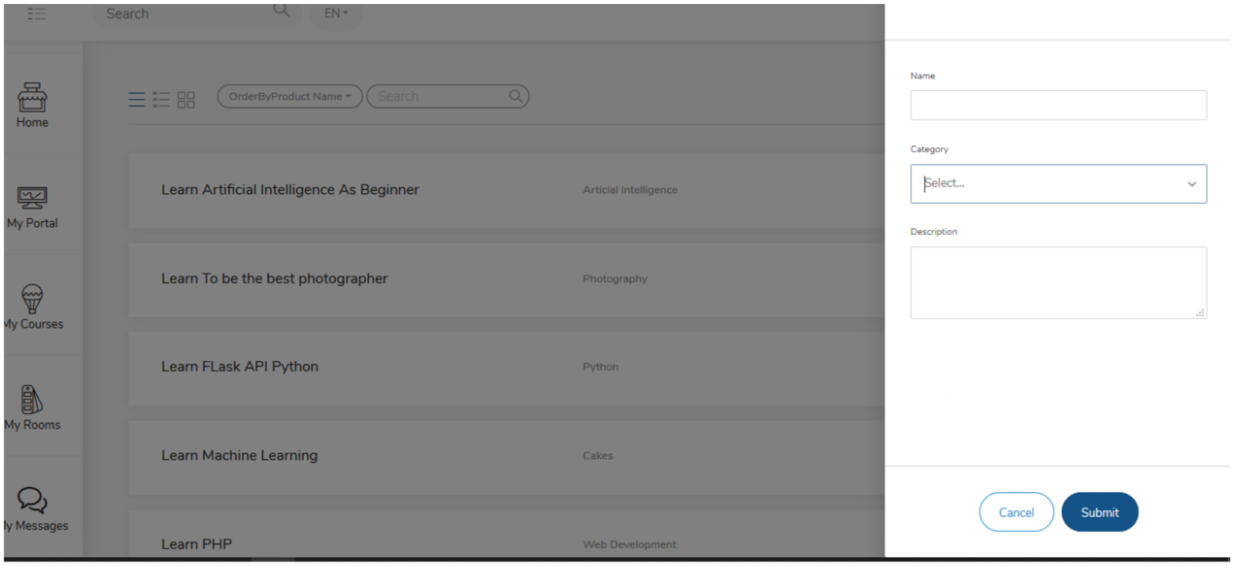


Figure 19: Courses List Screen

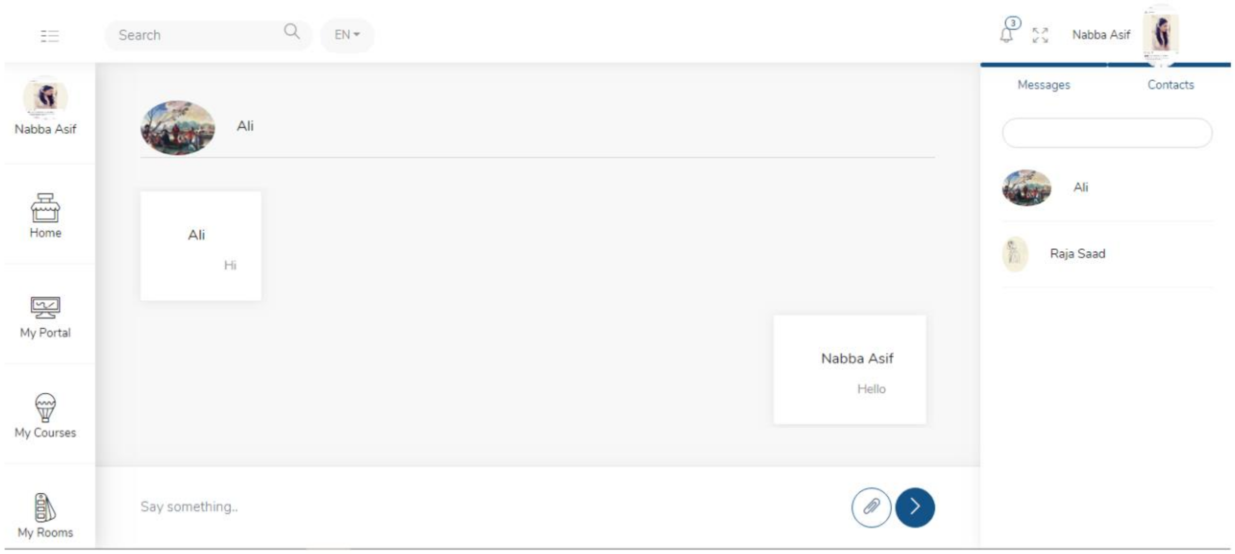
8.1.4 Teacher: Add Course



Learn Artificial Intelligence As Beginner	Articial Intelligence
Learn To be the best photographer	Photography
Learn FLask API Python	Python
Learn Machine Learning	Cakes
Learn PHP	Web Development

Figure 20: Add Item Screen

8.1.5 Message



Ali

Ali
Hi

Nabba Asif
Hello

Figure 21: Direct Message Screen

8.1.6 Profile

8.1.6.1 Add Picture

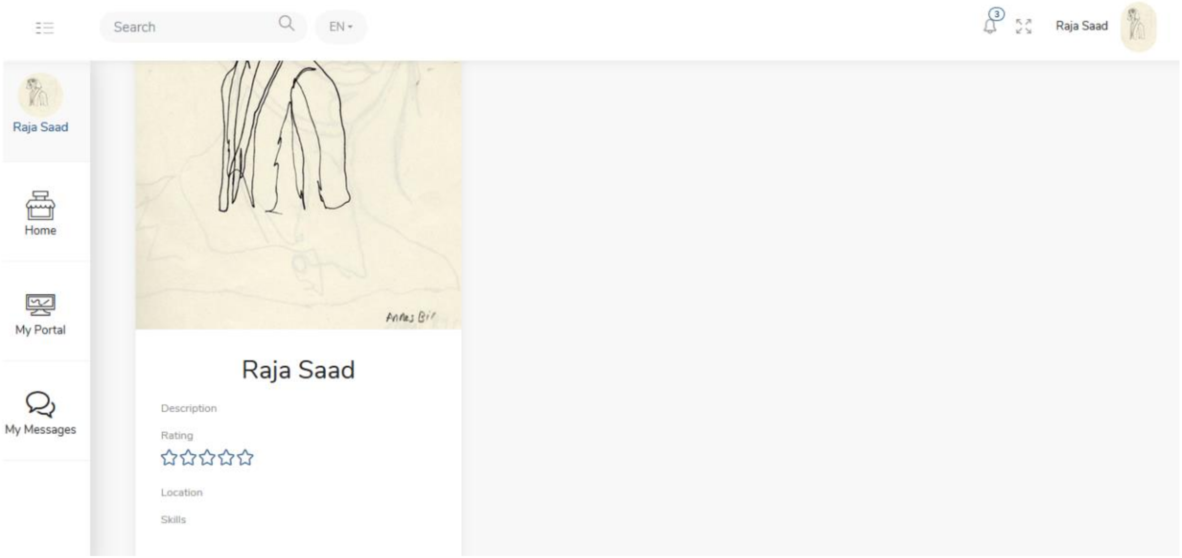
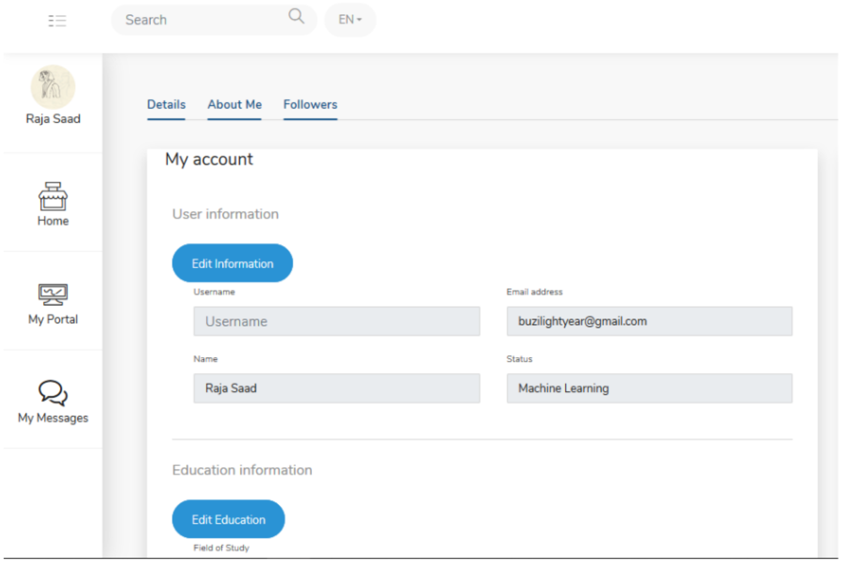


Figure 22: Change Profile Picture Screen

8.1.6.2 Add Information



My account

User information

Username: Username

Email address: buzilightyear@gmail.com

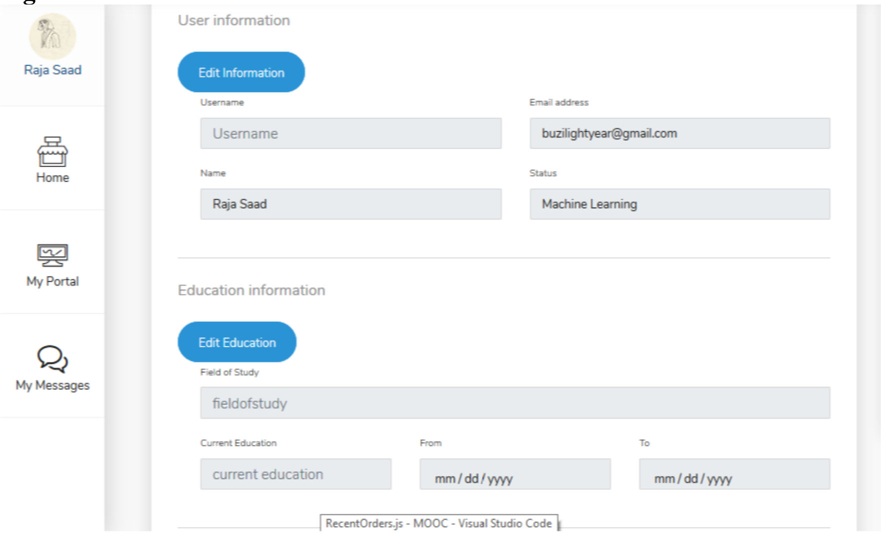
Name: Raja Saad

Status: Machine Learning

Education information

Field of Study

Figure 23: Add Personal Information Screen



User information

Username: Username

Email address: buzilightyear@gmail.com

Name: Raja Saad

Status: Machine Learning

Education information

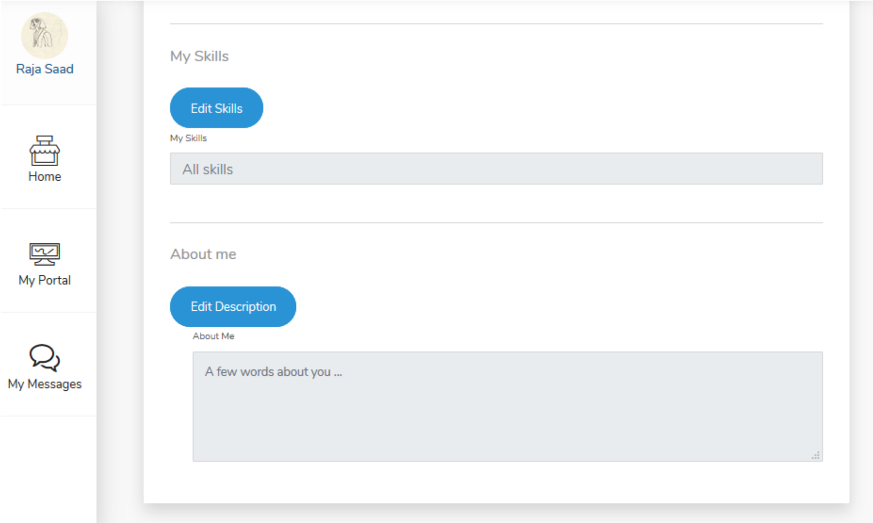
Field of Study: fieldofstudy

Current Education: current education

From: mm / dd / yyyy

To: mm / dd / yyyy

Figure 24: Add Education Information Screen



My Skills

Edit Skills

My Skills

All skills

About me

Edit Description

About Me

A few words about you ...

Figure 25: Add My Skills Screen

8.2 Screen objects and actions

8.1.7 Example 1: When teacher is writing on board

8.1.7.1 Smart Zoom Disabled

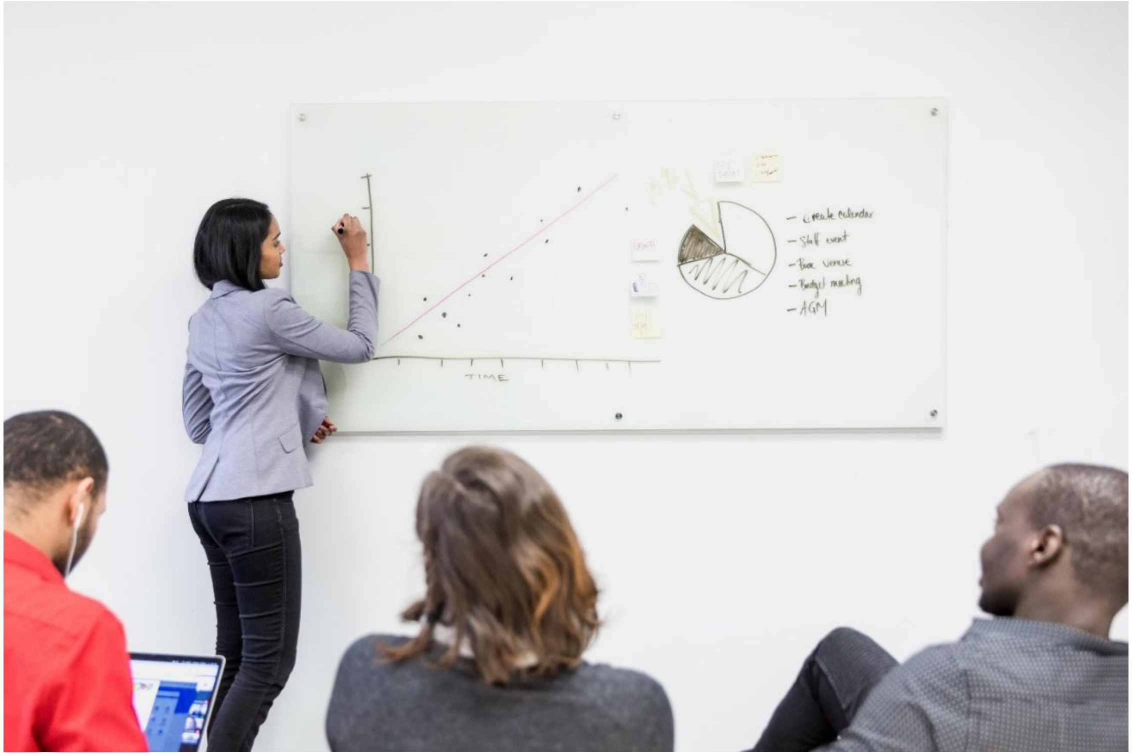


Figure 26: Example 1- Smart Zoom Disabled

8.1.7.1.1 Explanation

In the case where teacher is writing on board and If the smart zoom is disabled then the whole frame will be shown to the viewer.

8.1.7.2 Smart Zoom Enabled

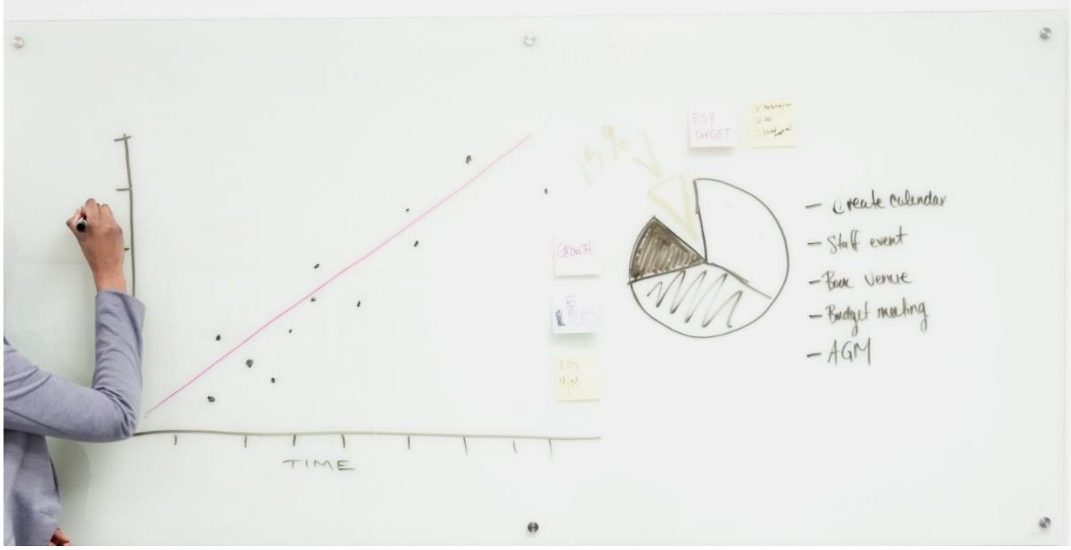


Figure 27: Example 1- Smart Zoom Enabled

8.1.7.2.1 Explanation

After enabling the smart zoom, the system will crop the frame to just the white board as shown above.

8.1.8 Example 2: When teacher is not writing (speaking or doing any other action)

8.1.8.1 Smart Zoom Disabled

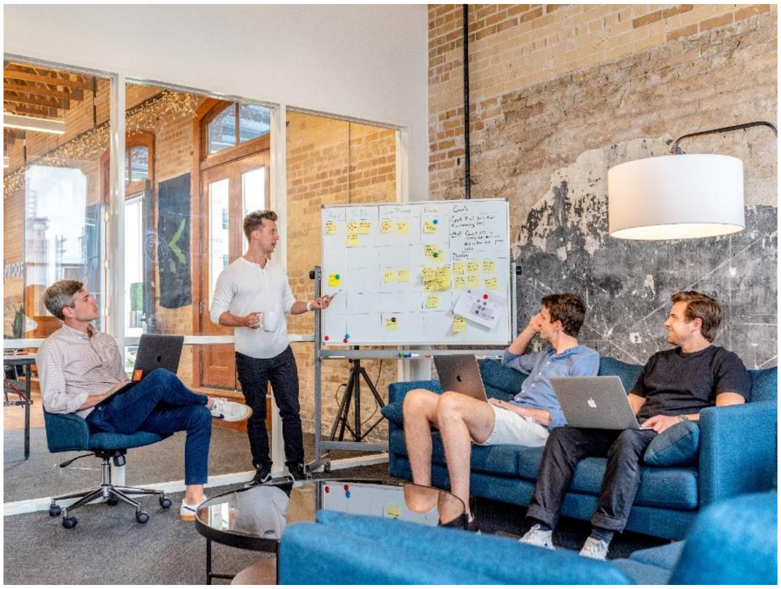


Figure 28: Example 2- Smart Zoom Disabled

8.1.8.1.1 Explanation

If the teacher is speaking and the autozoom is disabled then the system would show whole frame.

8.1.8.2 Smart Zoom Enabled

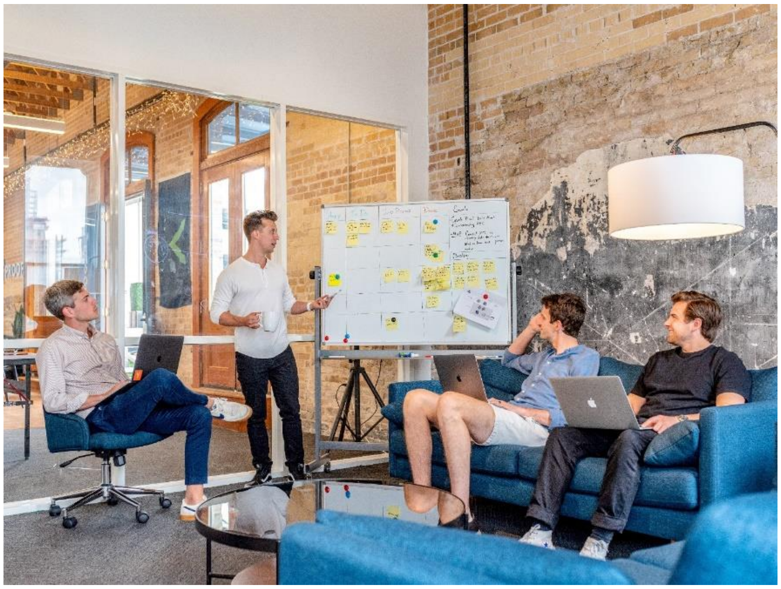


Figure 29: Example 2- Smart Zoom Enabled

8.1.8.2.1 Explanation

If the teacher is speaking and not writing on board then enabling the auto zoom would not crop the frame.

8.1.9 Example 3: When teacher is writing on board

8.1.9.1 Smart Zoom Disabled

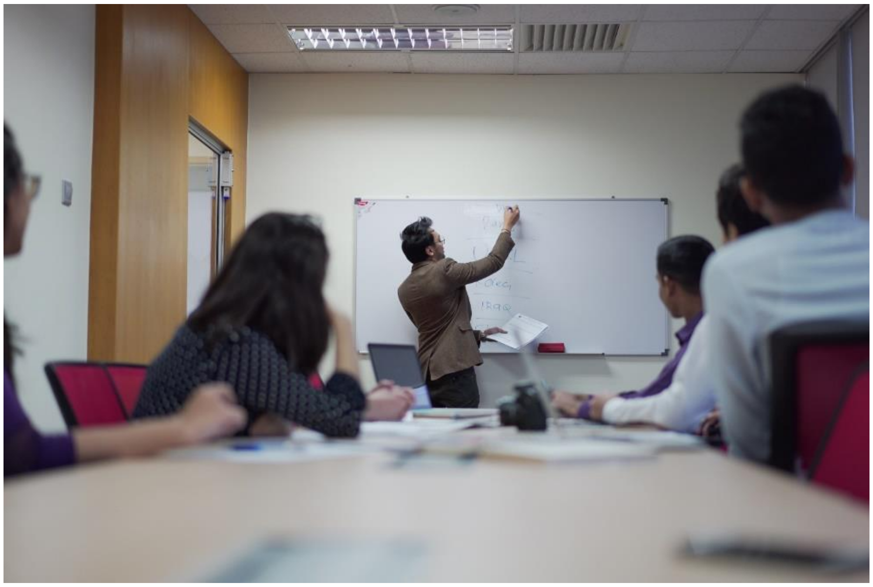


Figure 30: Example 3- Smart Zoom Disabled

8.1.9.1.1 Explanation

In the case where teacher is writing on board and If the smart zoom is disabled then the whole frame will be shown to the viewer.

8.1.9.2 Enable Smart Zoom

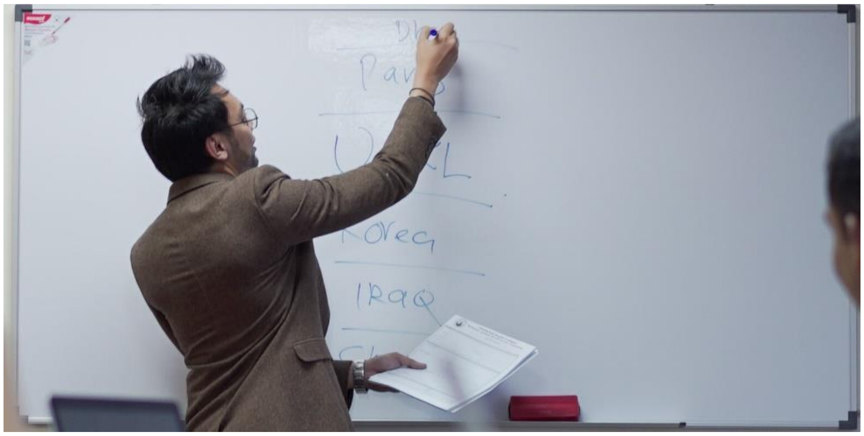


Figure 31: Example 3- Smart Zoom Enabled

8.1.9.2.1 Explanation

After enabling the smart zoom, the system will crop the frame to just the white board as shown above.

9. References

https://www.udemy.com/
https://www.udacity.com/
https://www.wiziq.com/
https://zoom.us/
https://discordapp.com/