

Student Information System

Front End Technologies

React

Back End Technologies

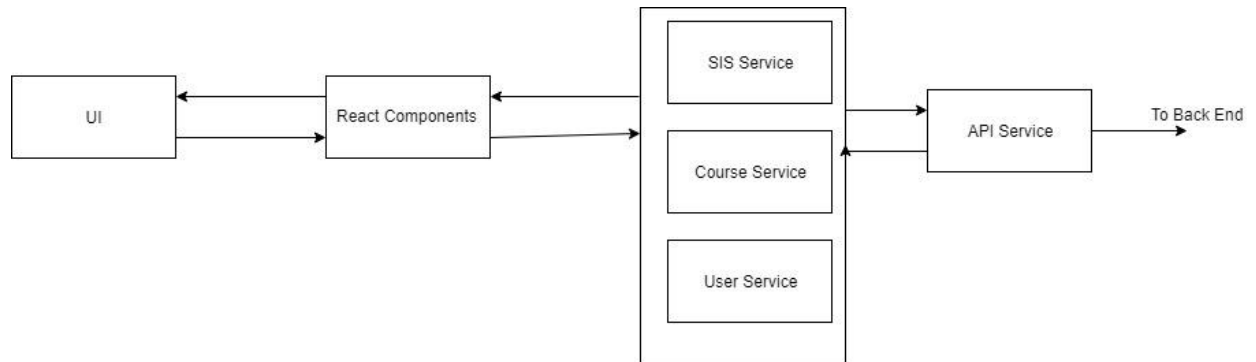
Node JS

Spring Boot

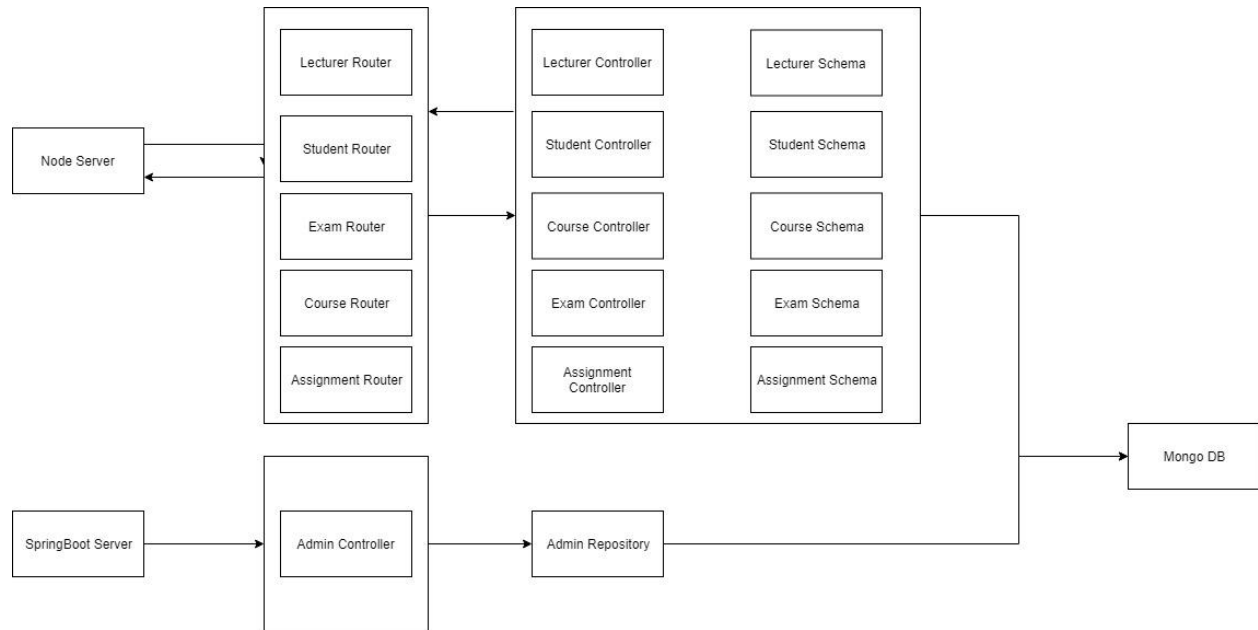
High Level Architecture



Front End Architecture



Back End Architecture



- All the API calls are tokenized
- All the passwords are encrypted using md5 hash
- Token expires after 2 hours and 30 minutes

RESTful HTTP URLs used for CRUD functions

URL	DESCRIPTION
http://localhost:3000/api/assignment/	Post – insert assignment
http://localhost:3000/api/assignment/allAssignments	Get – get all assignments
http://localhost:3000/api/assignment/FilterByUser/:CreatedBy	Get – get assignments of a lecturer
http://localhost:3000/api/assignment/FilterByKeys/:CourseId/:AssignmentId	Get – get assignments by keys
http://localhost:3000/api/assignment/:CourseId/:AssignmentId	Put – update assignments
http://localhost:3000/api/assignment/:CourseId/:AssignmentId	Delete- Delete assignments
http://localhost:3000/api/exam/allExams	Get – get all exam
http://localhost:3000/api/exam/FilterByUser/:CreatedBy	Get – get all exams of a lecturer
http://localhost:3000/api/exam/FilterByKeys/:CourseId/:AssignmentId	Get – get exam by keys
http://localhost:3000/api/exam/	Post – insert exam
http://localhost:3000/api/exam/:CourseId/:AssignmentId	Put – update exam
http://localhost:3000/api/exam/:CourseId/:AssignmentId	Delete – delete exam

Schemas for Assignments and Exam Management

```
//Assignment
const AssignmentSchema = new mongoose.Schema({
  CourseId: {
    type: String,
    required: true
  },
  AssignmentId: {
    type: String,
    required: true
  },
  Title: String,
  Description: String,
  StartDate: Date,
  EndDate: Date,
  CreatedBy: {
    type: String,
    required: true
  }
});

//Exam
const ExamSchema = new mongoose.Schema({

  CourseId: {

    type: String,

    required: true

  },

  ExamId: {

    type: String,

    required: true

  },

  Title: String,

  Type: String,

  TotalMarks: Number,

  PerFinalMark: Number,

  Date: String,

  Duration: String,

  CreatedBy: {

    type: String,
```

```
        required: true
    }
});
```

Sample Data for the Models

StudentInteractiveSystem.exams

```
{ "_id": "5d0f0a4ee4ba361f5460274c", "CourseId": "SE3080", "ExamId": "E500", "Title": "New ps exam", "Type": "WRITTEN", "TotalMarks": 100, "PerFinalMark": 25, "Date": "2019-06-25", "Duration": "4hrs", "CreatedBy": "test123", "__v": 0 }

{ "_id": "5d0f0b34e4ba361f5460274e", "CourseId": "SE3080", "ExamId": "E501", "Title": "MC written exam", "Type": "ONLINE", "TotalMarks": 100, "PerFinalMark": 34, "Date": "2019-06-26", "Duration": "45hrs", "CreatedBy": "test123", "__v": 0 }

{ "_id": "5d0f08b6e4ba361f5460274b", "CourseId": "SE3080", "ExamId": "00400", "Title": "PS. spot Exam", "Type": "SPOT", "TotalMarks": 40, "PerFinalMark": 10, "Date": "2019-06-12", "Duration": "240", "CreatedBy": "Anuradha", "__v": 0 }

{ "_id": "5d0f0707e4ba361f54602747", "CourseId": "SE3080", "ExamId": "E010", "Title": "PS final exam", "Type": "WRITTEN", "TotalMarks": 45, "PerFinalMark": 15, "Date": "2019-06-25", "Duration": "2hrs", "CreatedBy": "test123", "__v": 0 }
```

StudentInteractiveSystem.assignments

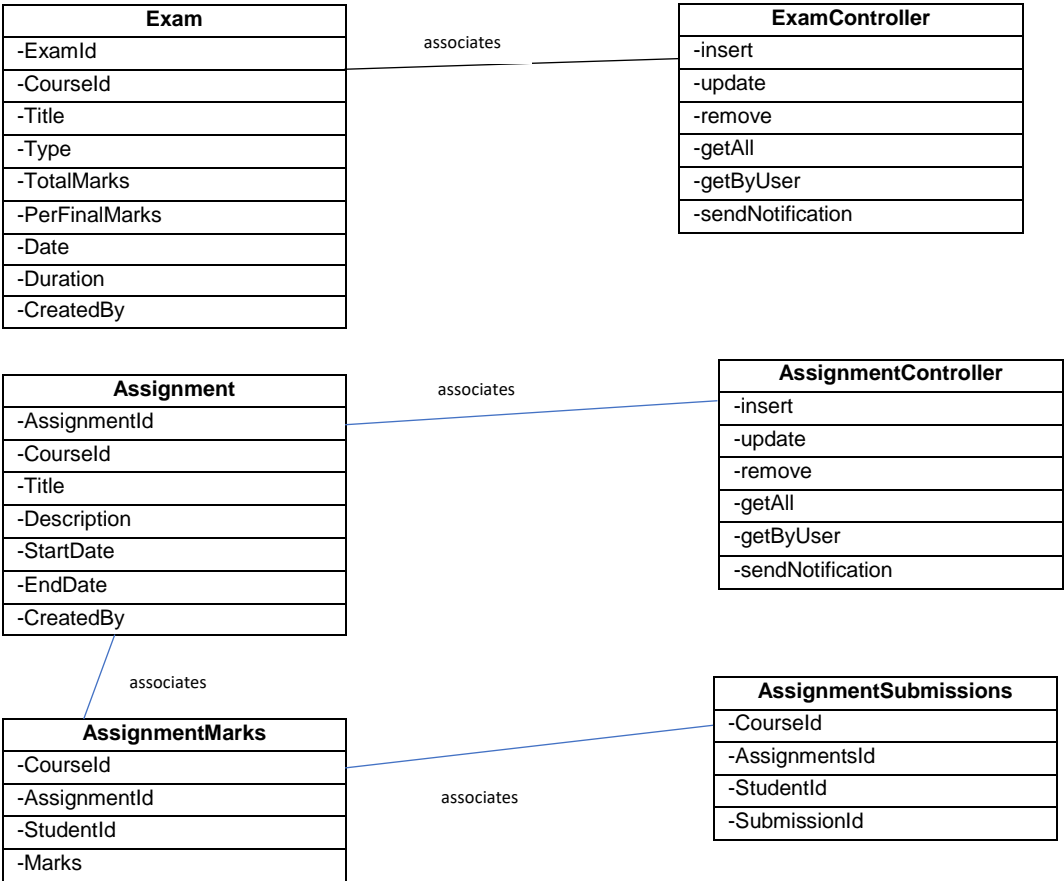
```
{ "_id": "5d0d6eee8fa26469f01c52b3", "CourseId": "SE3080", "AssignmentId": "A001", "Title": "Eclipse lab", "Description": "This is mandatory", "StartDate": "2019-06-25T00:00:00.000Z", "EndDate": "2019-06-28T00:00:00.000Z", "CreatedBy": "Anuradha", "__v": 0 }

{ "_id": "5d0daf6df59c1669b4b9679e", "CourseId": "SE3080", "AssignmentId": "A009", "Title": "JMS LAB", "Description": "Java related", "StartDate": "2019-06-03T00:00:00.000Z", "EndDate": "2019-07-04T00:00:00.000Z", "CreatedBy": "Anuradha", "__v": 0 }

{ "_id": "5d0db05dfa636816cc29a655", "CourseId": "SE3080", "AssignmentId": "A015", "Title": "CLOUD LAB", "Description": "Java related", "StartDate": "2019-06-03T00:00:00.000Z", "EndDate": "2019-07-04T00:00:00.000Z", "CreatedBy": "Anuradha", "__v": 0 }

{ "_id": "5d0e315216222082fc3b61f3", "CourseId": "SE3080", "AssignmentId": "A090", "Title": "NEW LAB", "Description": ".Net related", "StartDate": "2019-06-03T00:00:00.000Z", "EndDate": "2019-07-04T00:00:00.000Z", "CreatedBy": "Anuradha", "__v": 0 }
```

Class Diagram



Test Cases for Assignment Management

TestId	Description	Test Steps	Test Data	Expected Output	Actual Output	Result
TASMNT_ADD01	User cannot insert a previous date for start date when adding assignment	Fill the form with test data and click and click Add Assignment button	CourseID = SE3080 AssignmetId = A702 Title = EJB LAB Description = Refer the lectures and be prepared Start Date = 01.05.2018 End Date = 04.10.2019	Display window alert " The End Date should not be Today or a Previous Date"	Display window alert " The End Date should not be Today or a Previous Date"	Pass
TASMNT_ADD02	User cannot insert a same date for start and end date when adding assignment	Fill the form with test data and click and click Add Assignment button	CourseID = SE3080 AssignmetId = A702 Title = EJB LAB Description = Refer the lectures and be prepared Start Date = 04.10.2019 End Date = 04.10.2019	Display window alert "Start Date & End Date should not be equal"	Display window alert "Start Date & End Date should not be equal"	Pass
TASMNT_ADD03	User cannot insert day prior to end date for start date when adding assignment	Fill the form with test data and click and click Add Assignment button	CourseID = SE3080 AssignmetId = A702 Title = EJB LAB Description = Refer the lectures and be prepared Start Date = 25.10.2019 End Date = 04.10.2019	Enter Valid Start Date and End Date	Enter Valid Start Date and End Date	Pass