Requirement & Design Specification

**for**

**ligevan**

**Version 1.0**

**Created at 18/09/2024**

**Prepared by Group 1 - SE17C.NJS**

**FPT Quy Nhon AI Campus**

# Member

| **#** | **Full name** | **Student ID** | **Account** | **Contributor** |
| --- | --- | --- | --- | --- |
| 1 | Lê Minh Vương | QE170148 | VuongLM | 100% |
| 2 | Đinh Quốc Chương | QE170097 | ChuongDQ | 100% |
| 3 | Nguyễn Thị Thúy | QE170033 | ThuyNT | 100% |
| 4 | Hồ Trọng Nghĩa | QE170173 | NghiaHT | 100% |
| 5 | Lê Mạnh Hùng | QE170213 | HungLM | 100% |
| 6 | Lâm Mỹ Linh | QE170023 | LinhLM | 100% |

# Revision History

| **Version** | **Date** | **A, M, D** | **Assignee** | **Change Description** |
| --- | --- | --- | --- | --- |
| 0.1 | 03/10/2024 | A | VuongLM | Implement base document |
| 0.2 | 03/10/2024 | A | VuongLM | Implement Repository & CI/CD |
| 0.3 | 04/10/2024 | A | VuongLM | Implement Dependencies   * backend * frontend |
| 0.4 | 04/10/2024 | A | ChuongDQ | Implement Collection description |
| 0.5 | 06/10/2024 | A | ChuongDQ | Implement Database Schema |
| 0.6 | 06/10/2024 | A | ChuongDQ | Implement Collection detail:   * admins * centers * course * vouchers |
| 0.7 | 07/10/2024 | A | ChuongDQ | Implement Collection detail:   * teachers * students * classes * slots * attendances * payments * salaries |
| 0.8 | 7/10/2024 | A | ThuyNT | Implement Sequence diagram:   * Login * Student Portal: My class |
| 0.9 | 7/10/2024 | A | HungLM | Implement State diagram:   * Login * Slot Management * Attendance report * Profile Management |
| 1 | 8/10/2024 | A | NghiaHT | Implement Sequence diagram:   * Manage center * Tracking Attendance |
| 1.1 | 9/10/2024 | A | LinhLM | Implement System Context diagram |
| 1.2 | 9/10/2024 | M | LinhLM | Update System Context diagram |
| 1.3 | 10/10/2024 | A | ThuyNT | Implement Sequence diagram:   * Teacher Portal: Attendance Tracking * Admin Portal: Center Management |
| 1.4 | 12/10/2024 | M | ThuyNT | Update Sequence diagram:   * Login * Student Portal: My class * Teacher Portal: Attendance Tracking * Admin Portal: Center Management |
| 1.5 | 13/10/2024 | A | ChuongDQ  NghiaHT | Implement Backend Architectural Design |
| 1.6 | 14/10/2024 | A | LinhLM  HungLM | Implement Architectural Design Analysis Overview |
| 1.7 | 14/10/2024 | A | ThuyNT VuongLM | Implement Frontend Architectural Design |
| 1.8 | 17/10/2024 | M | VuongLM | Update database schema |
| 1.9 | 19/10/2024 | A | ChuongDQ | Implement base for UI/UI Design |
| 2.0 | 20/10/2024 | A | LinhLM | Implement UI/UI Design of Teacher   * Login * View Class List |
| 2.1 | 20/10/2024 | A | NghiaHT | Implement UI/UI Design of Student   * Home Page * View Profile |
| 2.2 | 20/10/2024 | A | HungLM | Implement UI/UI Design of Admin   * Login * Dashboard |
| 2.3 | 27/10/2024 | A | ThuyNT | Implement UI/UX Design |
| 2.4 | 29/10/2024 | A | ThuyNT | Implement UI/UX Design |
| 2.5 | 03/11/2024 | A | ChuongDQ | Implement Deployment in Repository |

**\*A - Added, M - Modified, D - Deleted**

# Table of contents

[**Member 2**](#_u77jcw4x8z9n)

[**Revision History 3**](#_ft87mped2cvi)

[**Table of contents 5**](#_ibtj9aibrb29)

[**I. Overview 8**](#_ki3ezcfs12xz)

[A. Software Requirement Specification 8](#_5f2271wfdx28)

[B. Database design 8](#_bgzqcvxtqmpo)

[1. Database schema 8](#_biepphj98z1l)

[2. Collection description 8](#_sz52jdrm6vwn)

[3. Collection detail description 9](#_ebn2n6klmqdx)

[a) `admins` collection 9](#_t1kh7yukq05m)

[b) `centers` collection 9](#_injmf6i2fmtu)

[c) `course` collection 10](#_wzs5djp9jo0y)

[d) `vouchers` collection 10](#_2a3lebd4znv8)

[e) `teachers` collection 10](#_5xy7ncdxn7oq)

[f) `students` collection 11](#_ru8e5vnn7i6o)

[g) `classes` collection 12](#_l5web58q2lnk)

[h) `slots` collection 12](#_gud3p91asocn)

[i) `attendances` collection 12](#_8qh7k6u3zzxn)

[j) `payments` collection 13](#_p0ihodvppjeh)

[k) `salaries` collection 13](#_n9tsn774gi54)

[C. Repository 15](#_5u8obxgj15qd)

[1. Repository 15](#_yc0wqhvfoqz5)

[2. CI/CD 15](#_wwu4u1shyx5e)

[a) Commit message validation 15](#_3wcxkqidz18)

[b) PR validation 15](#_b51spcaijv2z)

[c) Publish backend image 15](#_ffzv0r3hx7e9)

[d) Publish frontend image 15](#_rcho24v61ltc)

[e) Validate branches on creation 15](#_62tzg1ylro3i)

[3. Deployment 15](#_t3erbv3xpppy)

[D. Dependencies 15](#_aeskcd3g9dd8)

[1. Backend 15](#_i6a2eurytql6)

[2. Frontend 16](#_c92ww5uj3n5n)

[**II. System Design 17**](#_mxcrdb205jpk)

[A. System Context 17](#_2bkswgpeu8v4)

[B. Analysis models 18](#_vl0k3djvg1d3)

[1. Interaction diagrams - Sequence diagram 18](#_4aa5qxovt2l0)

[a) Authentication/Login/Logout 18](#_asw7a9oop2hs)

[b) Student Portal: My Classes/Attendance Record/Payment Management 18](#_s6f56mf3aiv2)

[c) Teacher Portal: Attendance Tracking 20](#_luu48j732iaz)

[d) Admin Portal: Center Management 21](#_toc7g2mdjij)

[2. State diagram 24](#_ljzno7g5oycu)

[a) State Login diagram 24](#_ejxu8fb9z3i8)

[b) State Slot Management diagram 25](#_17dcl4pl5w1i)

[c) State Attendance report diagram 26](#_e9j7rzdk27tz)

[d) State Profile management diagram 27](#_svj1gerl09nz)

[C. Design specification - Communication Diagram 29](#_xdpx9rii5xv6)

[1. Manage Center: Add new, VIew all, View details, Edit, Delete 29](#_4khawm33up7n)

[2. Tracking Attendance 30](#_nyxsw3f0idlh)

[**III. Architectural Design Analysis 30**](#_vjaimtdmv241)

[A. Overview 30](#_lrvbeop3zp2g)

[1. Client-Server (Single Server - Multiple Frontend) 30](#_naxam5ax7pc2)

[2. Diagram of the Client-Server (Single Server - Multiple Frontend) 31](#_c5akbat06aex)

[B. Frontend: 31](#_mz42zyqoxgv7)

[1. Architectural Overview 31](#_bj1xu3vefdp9)

[2. Key Components 31](#_ib84qlp8isna)

[3. Modular of the Frontend Architecture 32](#_xtfrxw16uf97)

[C. Backend 32](#_9an89awbyr3z)

[**IV. UI/UX Design - Mockup 35**](#_un5vkfspbh3l)

[A. Admin 35](#_pnrmfcac3snk)

[1. Login Page 35](#_7svsbhdekjmv)

[2. Change password 36](#_9iars8qymzme)

[3. Dashboard 36](#_f56s6emcra4k)

[4. View Centers List 37](#_3r5zordpe50c)

[a) Create new center 37](#_9h3hhc32pcwj)

[b) View Center Detail 38](#_di1m0xl78ahh)

[c) Update Center 38](#_dydz8ndw3tk9)

[d) Delete Center 39](#_63o5v72z5j49)

[e) View classes list 39](#_b2mxocakl8kp)

[f) Create new class 40](#_thxzmc4halb4)

[g) View Class Detail 40](#_gquky1nu71un)

[h) View Teachers List 41](#_pqo7mhp025zn)

[i) Create New Teacher 41](#_8cqmqzacsucu)

[j) View Teacher Detail 41](#_bawf4wbgrdf9)

[5. View Courses List 42](#_8juh7hjj7bvv)

[a) Create new course 42](#_uxqacdtrzcyz)

[b) View Course Detail 43](#_k17v0od81j5c)

[c) Update Course 43](#_260xk058561x)

[d) Delete Course 44](#_efc777s8gfb8)

[6. View Vouchers List 45](#_rmpdgjctv85h)

[a) Create New Voucher 45](#_n0ar49psd9w7)

[b) View Voucher Detail 45](#_n6h3uy2sizxe)

[c) Update Voucher 46](#_4qpmwumpzt4s)

[d) Delete Voucher 46](#_a9sc8dhjkljb)

[7. Schedule 47](#_1f28pinnaxjt)

[a) View class schedule 47](#_7733d22sq6)

[b) Update Schedule 48](#_azpxhj7c6fey)

[8. View Teacher Salary List 48](#_sfb41pfshwei)

[a) View Teacher Salary List 48](#_ebpqyy34l0xj)

[b) Salary Calculation For Teachers 49](#_6eg932wa77x8)

[B. Teacher 49](#_r2qixu5ugqhj)

[1. Login Page 49](#_xi0th91sesr5)

[2. Change Password 49](#_eaa2e46hry5r)

[3. View Profile 50](#_jezdtzdzn26f)

[4. Update Profile 50](#_vhuzh73e1l1m)

[5. View Class List 51](#_mo4pypguo0j4)

[6. View Class Detail 51](#_iu6hgyvjg2dd)

[7. View Student Detail 52](#_r5hje5hxdsl7)

[8. Teaching Schedule 52](#_qs352t5ghh3u)

[a) View Teaching Schedule 52](#_rhi7a5gn6vaf)

[b) Check Attendance 53](#_89sjm3y5je7m)

[C. Student 53](#_5xznzfadmax9)

[1. Home Page 53](#_dp9gt4x20c81)

[2. Login Page 54](#_tjjjd0uxrtfc)

[a) Login 54](#_xj8nth9ar3bg)

[b) Login With Google 55](#_mv1ng961oela)

[3. View Profile 55](#_sye40yv5qo9y)

[a) Profile 55](#_xn0z4mvebase)

[b) Update Profile 55](#_oymvlitd0yg9)

[4. View Class List 56](#_pfmc9cxh0804)

[a) View Class List 56](#_t5e7ieoqh91b)

[b) View Class Detail 56](#_cjohbeuac23h)

[c) View Class schedule 57](#_d811wdia4ekh)

[5. Payment History Detail 57](#_s4kpcamecg1j)

[a) View Payment History 57](#_w29bff1wjri9)

[b) View Payment History Detail 58](#_vrcb0ou81rjx)

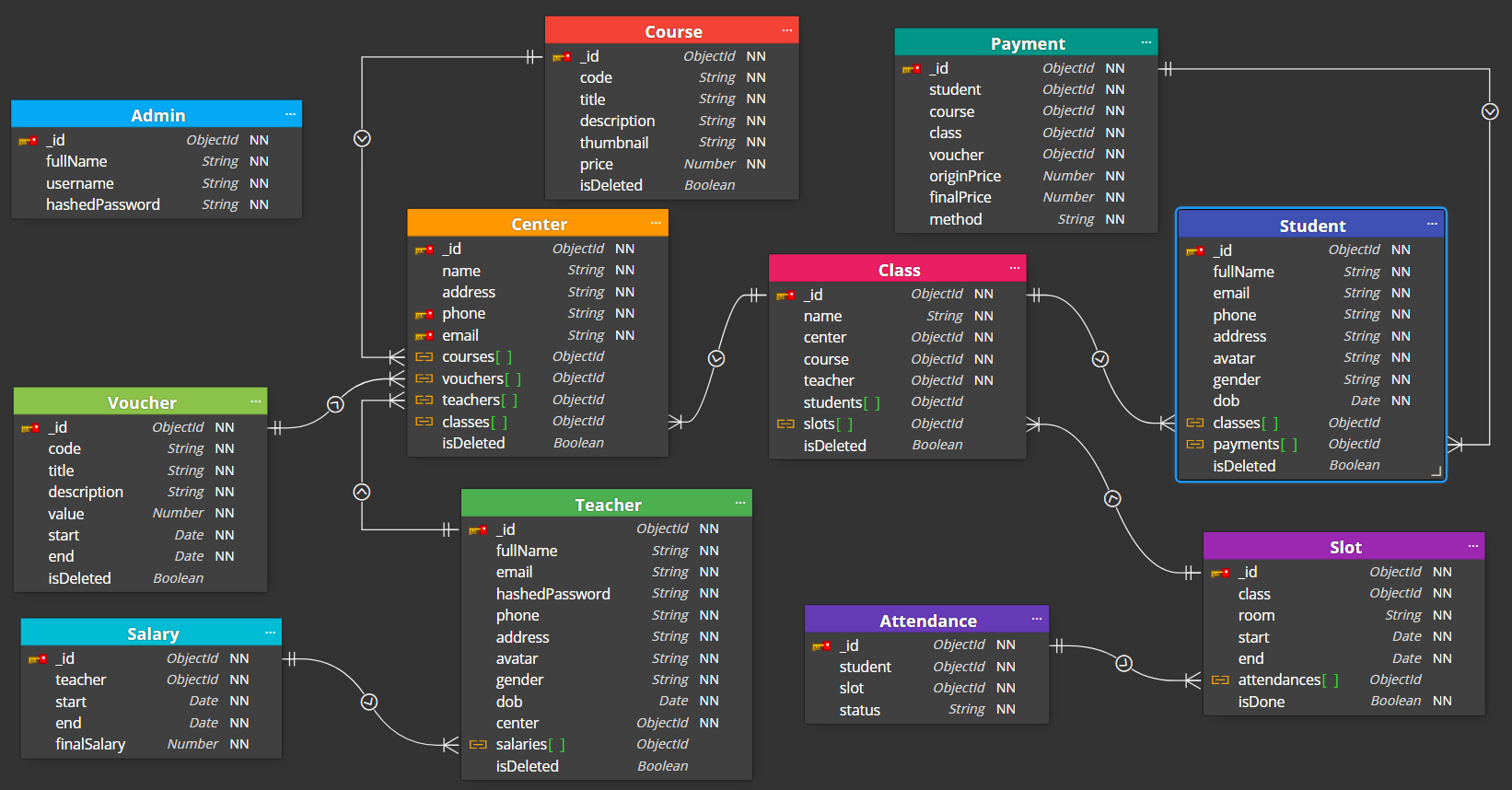
# Overview

## Software Requirement Specification

[ligevan - SRS](https://docs.google.com/document/d/16O2i5cR0zmeRglXDZpUKKc2BEnbO2rx2/edit?usp=sharing&ouid=108875959429454212061&rtpof=true&sd=true)

## Database design

### Database schema



### Collection description

| **#** | **Collection** | **Description** |
| --- | --- | --- |
| 1 | admins | Stores admin infomation |
| 2 | centers | Stores center infomation |
| 3 | teachers | Stores teacher details |
| 4 | students | Stores student information |
| 5 | classes | Stores class infomation |
| 6 | courses | Stores course infomation |
| 7 | vouchers | Stores voucher infomation |
| 8 | slots | Stores slot infomation of class |
| 9 | attendances | Stores attendance infomation of student |
| 10 | payments | Stores payment infomation of student |
| 11 | salaries | Stores salary infomation of teacher |

### Collection detail description

#### `admins` collection

| **Field** | **Type** | **Constraint** | **Description** |
| --- | --- | --- | --- |
| fullName | string | required | Admin full name |
| username | string | required, unique | Admin username |
| hashedPassword | string | required | Admin hashed password |

#### `centers` collection

| **Field** | **Type** | **Constraint** | **Description** |
| --- | --- | --- | --- |
| name | string | required | Center name |
| address | string | required, unique | Center address |
| phone | string | required, unique | Center phone number |
| email | string | required, unique | Center email |
| courses | Object | ref: 'Course', required | Courses offered by the Center |
| vouchers | Object | ref: 'Voucher' | Vouchers offered by the Center |
| teachers | Object | ref: ‘Teacher’, required | Teachers offered by the Center |
| classes | Object | ref: 'Class', required | Classes offered by the Center |
| isDeleted | boolean | default: false | Soft delete indicator for the Center |

#### 

#### `course` collection

| **Field** | **Type** | **Constraint** | **Description** |
| --- | --- | --- | --- |
| code | string | required, unique | Course code |
| title | string | required | Course title |
| description | string | required | Course description |
| thumbnail | string | required | Course thumbnail |
| price | number | required | Course price |
| isDeleted | boolean | default: false | Soft delete indicator for the Course |

#### `vouchers` collection

| **Field** | **Type** | **Constraint** | **Description** |
| --- | --- | --- | --- |
| code | string | required, unique | Voucher code |
| title | string | required | Voucher title |
| description | string | required | Voucher description |
| value | number | required | Monetary discount amount (%) |
| start | date | required | Voucher start date |
| end | date | required | Voucher end date |
| isDeleted | boolean | default: false | Soft delete indicator for the Voucher |

#### `teachers` collection

| **Field** | **Type** | **Constraint** | **Description** |
| --- | --- | --- | --- |
| fullName | string | required | Teacher full name |
| email | string | required, unique | Teacher email |
| hashed Password | string | required | Teacher hashed password |
| phone | string | required, unique | Teacher phone number |
| address | string | required | Teacher address |
| avatar | string | required | Teacher avatar |
| gender | string | required,  ‘male’ | ‘female’ | Teacher gender |
| dob | date | required | Teacher date of birth |
| center | Object | ref: ‘Center’, required | Center offered by the Teacher |
| salaries | Object | ref: 'Salary’, required' | Salaries offered by the Teacher |
| isDeleted | boolean | default: false | Soft delete indicator for the Teacher |

#### 

#### `students` collection

| **Field** | **Type** | **Constraint** | **Description** |
| --- | --- | --- | --- |
| fullName | string | required | Student full name |
| email | string | required, unique | Student email |
| phone | string | required, unique | Student phone number |
| address | string | required | Student address |
| avatar | string | required | Student avatar |
| gender | string | required,  ‘male’ | ‘female’ | Student gender |
| dob | date | required | Student date of birth |
| classes | Object | ref: ‘Class’, required | Classes offered by the Student |
| payments | Object | ref: 'Payment’ | Payments offered by the Student |
| isDeleted | boolean | default: false | Soft delete indicator for the Student |

#### 

#### `classes` collection

| **Field** | **Type** | **Constraint** | **Description** |
| --- | --- | --- | --- |
| name | string | required | Class name |
| center | Object | ref: ‘Center’, required | Center offered by the Class |
| course | Object | ref: ‘Course’, required | Course of class |
| teacher | Object | ref: 'Teacher’, required | Teachers offered by the Class |
| students | Object | ref: 'Student’, required | Students offered by the Class |
| slots | Object | ref: 'Slot’, required | Slot offered by the Class |
| isDeleted | boolean | default: false | Soft delete indicator for the Class |

#### 

#### `slots` collection

| **Field** | **Type** | **Constraint** | **Description** |
| --- | --- | --- | --- |
| class | Object | ref: ‘Class’, required | Class offered by the Slot |
| room | string | required | Room of Slot |
| start | date | required | Start time of Slot |
| end | date | required | End time of Slot |
| attendances | Object | ref: ‘Attendance’ | Attendances offered by the Slot |
| isDone | boolean | default: false | Teacher can completed Slot |

#### `attendances` collection

| **Field** | **Type** | **Constraint** | **Description** |
| --- | --- | --- | --- |
| student | Object | ref: ‘Student’, required | Student offered by the Attendance |
| slot | Object | ref: ‘Slot’ | Slot offered by the Attendance |
| status | string | default: not yet,  ‘not yet’ | ‘absent’ | ‘attended’ | Attendance status |

#### `payments` collection

| **Field** | **Type** | **Constraint** | **Description** |
| --- | --- | --- | --- |
| student | Object | ref: ‘Student’, required | Student offered by the Payment |
| course | Object | ref: ‘Course’, required | Course offered by the Payment |
| class | Object | ref: ‘Class’, required | Class offered by the Payment |
| voucher | Object | ref: ‘Voucher’ | Voucher offered by the Payment |
| originPrice | number | required | Origin Price of Payment |
| finalPrice | number | required | Final Price of Payment |
| method | string | required  'vn-pay' | 'momo' | 'zalo-pay' | Payment method |

#### `salaries` collection

| **Field** | **Type** | **Constraint** | **Description** |
| --- | --- | --- | --- |
| teacher | Object | ref: ‘Teacher’, required | Teacher offered by the Salary |
| start | date | required | Start date to calculate salary |
| end | date | required | End date to calculate salary |
| finalSalary | number | required | Final salary of the teacher |

## 

## Repository

### Repository

* GitHub repository: [GitHub](https://github.com/hardingadonis/ligevan)
* Project management: [Project](https://github.com/users/hardingadonis/projects/14)

### CI/CD

#### Commit message validation

* Description: To validate the commit messages when push code into GitHub repository.
* Source: [commit-msg-validation.yml](https://github.com/hardingadonis/ligevan/blob/main/.github/workflows/commit-msg-validation.yml)

#### PR validation

* Description: Checks PRs to ensure they target the ***dev*** branch instead of ***main***.
* Source: [pr-validation.yml](https://github.com/hardingadonis/ligevan/blob/main/.github/workflows/pr-validation.yml)

#### Publish backend image

* Description: Builds and publishes the backend Docker image on pushes to ***main***.
* Source: [publish-backend-image.yml](https://github.com/hardingadonis/ligevan/blob/main/.github/workflows/publish-backend-image.yml)

#### Publish frontend image

* Description: Builds and publishes the frontend Docker image on pushes to ***main***.
* Source: [publish-frontend-image.yml](https://github.com/hardingadonis/ligevan/blob/main/.github/workflows/publish-frontend-image.yml)

#### Validate branches on creation

* Description: Validates newly created branches, ignoring ***main***, ***dev***, and ***imgbot/\****.
* Source: [validate-branches-on-creation.yml](https://github.com/hardingadonis/ligevan/blob/main/.github/workflows/validate-branches-on-creation.yml)

### Deployment

* Backend APIs: <https://ligevan.eastasia.cloudapp.azure.com:3000>
* Swagger Docs: <https://ligevan.eastasia.cloudapp.azure.com:3000/api-docs>
* Frontend Web: <https://ligevan.vercel.app>

## Dependencies

### Backend

| **#** | **Package** | **Version** |
| --- | --- | --- |
| 1 | @nestjs/common | 10.4.4 |
| 2 | @nestjs/config | 3.2.3 |
| 3 | @nestjs/core | 10.4.4 |
| 4 | @nestjs/jwt | 10.2.0 |
| 5 | @nestjs/mongoose | 10.0.10 |
| 6 | @nestjs/platform-express | 10.4.4 |
| 7 | @nestjs/swagger | 7.4.2 |
| 8 | argon2 | 0.41.1 |
| 9 | class-transformer | 0.5.1 |
| 10 | class-validator | 0.14.1 |
| 11 | module-alias | 2.2.3 |
| 12 | mongoose | 8.6.3 |
| 13 | morgan | 1.10.0 |
| 14 | reflect-metadata | 0.2.2 |
| 15 | rxjs | 7.8.1 |

### Frontend

| **#** | **Package** | **Version** |
| --- | --- | --- |
| 1 | @reduxjs/toolkit | 2.2.7 |
| 2 | antd | 5.21.0 |
| 3 | axios | 1.7.7 |
| 4 | react | 18.3.1 |
| 5 | react-dom | 18.3.1 |
| 6 | react-redux | 9.1.2 |

### 

# System Design

## System Context

|  |
| --- |
| *Figure II-A: System Context Diagram* |

* Entities:
* Admin: The administrator of the system, who has access to all data and can perform administrative tasks.
* Teacher: manage class, student, take attendance.
* Student: can register course, view data of course, class, slot.
* Payment Gateway: A third-party service that handles payment transactions.
* Google Authentication Service: A service used for user authentication and authorization.

## Analysis models

### Interaction diagrams - Sequence diagram

#### Authentication/Login/Logout

|  |
| --- |
| *Figure II-B-1-a: Sequence diagram for Login* |

* Diagram Description:

Access the Portal and authenticate: The user accesses the Portal and submits a login request to the Authentication Server. If the provided credentials are valid, the user is verified and redirected to the homepage.

#### Student Portal: My Classes/Attendance Record/Payment Management

|  |
| --- |
| *Figure II-B-1-b: Sequence diagram for My Classes* |

* Diagram Description:

My Classes: The student selects 'My Classes' from the Sidebar. The browser sends a request to the Web Server to retrieve data, the Web Server fetches the information from the Database and returns the class schedule of courses.

#### Teacher Portal: Attendance Tracking

|  |
| --- |
| *Figure II-B-1-c: Sequence diagram for Attendance Tracking* |

* Diagram Description:
* Access Teacher Portal: The teacher logs into the Portal, and the browser sends a request to retrieve the current class list.
* Retrieve class list: The web server fetches the class list from the database and sends it back to the browser for the teacher to mark attendance.
* Record attendance: The teacher records each student’s attendance (present/absent) on the browser, which is then sent to the web server and saved in the database.
* Generate attendance report: The teacher requests to export the attendance report, and the browser sends a request to the web server. The web server retrieves the attendance data from the database and provides the report for download.

#### Admin Portal: Center Management

|  |
| --- |
| *Figure II-B-1-d-1: Sequence diagram for Center Management* |

|  |
| --- |
| *Figure II-B-1-d-2: Sequence diagram for Center Management* |

* Diagram Description:
* Center Management: The administrator opens Center Management to view the list of branches of the center.
* The browser sends a request to the web server to retrieve the list of branches from the database.
* The administrator can search, filter branches, or view detailed information for each branch.

### State diagram

#### State Login diagram

|  |
| --- |
| *Figure II-B-2-a: State Login Diagram* |

* Diagram Description: The login state diagram illustrates the complete lifecycle of a user authentication process. It begins at the Start state, leading to the Login Page where users input their credentials. The system then moves to the Validate state to verify the information. Upon successful authentication, users are directed to the Main Page. If authentication fails, an Error Message is displayed, allowing users to retry.

#### State Slot Management diagram

|  |
| --- |
| *Figure II-B-2-b: State Slot Management Diagram* |

* Diagram Description: This flowchart illustrates a streamlined process for creating, assigning, and publishing class schedules, with the flexibility to update before final publication.

#### State Attendance report diagram

|  |
| --- |
| *Figure II-B-2-c: State Attendance report Diagram* |

* Diagram Description: This state diagram tracks student attendance: starting as "Not Yet," then moving to "Present" or "Absent," and ending with a "Tracking Report."

#### State Profile management diagram

|  |
| --- |
| *Figure II-B-2-d: State Profile management Diagram* |

* Diagram Description: This state diagram manages user profiles: starting with "Viewing," it transitions to "Editing" or "Deleting," with options to save, cancel, or confirm actions.

## Design specification - Communication Diagram

To transition from analysis to design and to identify the subsystems, it is necessary to

synthesise an initial software design based on the analysis conducted so far. This involves

integrating the use case based interaction diagrams developed as part of the dynamic model.

The integrated collaboration diagrams will then incorporate all interactions within the

system. For large and complex systems, this integration may be performed for each subsystem

individually. In this project, we will integrate the collaboration diagrams for the use cases

we have developed:

### Manage Center: Add new, VIew all, View details, Edit, Delete

|  |
| --- |
| *Figure II-C-1:Integrated communication diagram of Manage centers.* |

* Diagram Description: The diagram illustrates the flow of interactions between key objects such as Admin, Admin Dashboard View, Manage Center View, Center Controller, Center Service. This integration helps identify the classes and methods required to implement the use case. For example, the diagram shows that Admin manages the retrieval, addition, and update of information for centers.

### Tracking Attendance

|  |
| --- |
| *Figure II-C-2:Integrated communication diagram of Tracking Attendance.* |

* Diagram Description: The diagram illustrates the flow of interactions between key objects such as Teacher, TeacherPortalView, ClassListView, AttendanceTrackingView, ClassController, and ClassService. This integration helps identify the classes and methods required to implement the use case. For example, the diagram shows that the Teacher manages the retrieval of current slots and student lists, and updates attendance records.

# Architectural Design Analysis

## Overview

### Client-Server (Single Server - Multiple Frontend)

* Multiple frontends:
* Each user group (Student, Teacher, Admin) has its own dedicated frontend interface, optimized for their specific tasks and responsibilities.
* These frontends interact with the same backend server through RESTful APIs to retrieve and send data.
* Backend (Single Server):
* The server handles all business logic, including managing centers, courses, users,...
* It serves all the frontends by providing APIs and communicates with a central database to manage data.

### Diagram of the Client-Server (Single Server - Multiple Frontend)

|  |
| --- |
| *Figure III-A: Diagram of the Client Server (Single Server - Multiple Frontend)* |

## Frontend:

### Architectural Overview

* **React**: The core of the frontend library, responsible for building reusable UI components and efficiently updating the UI when data changes.
* **Ant Design (AntD)**: A design system integrated into React to provide a consistent, modern, and customizable UI component library, streamlining the development of user interfaces.
* **Redux**: The state management library used to centralize the application's state, ensuring consistent state updates and making it easier to manage complex application states.
* **Flux Architecture**: The design pattern around which Redux is built, where unidirectional data flow is emphasized for predictable state changes.

### Key Components

* UI Layer (React + AntD):
* *React Components*: Build the interactive UI, responsible for rendering data from the state and capturing user interactions.
* *AntD Components*: Pre-designed UI elements for forms, tables, modals, etc., facilitating rapid development.
* State Management Layer (Redux):
* *Actions*: User interactions or system events dispatch actions (objects containing type and payload).
* *Reducers*: Functions that take the current state and an action, process them, and return a new state.
* *Store*: A central repository holding the application's state.
* Data Flow (Flux Architecture):
* *Unidirectional Flow*: Data flows in one direction—starting from the user interaction (view), moving to the dispatcher (Redux action), updating the store, and reflecting the changes in the UI (React components).

### Modular of the Frontend Architecture

|  |
| --- |
| *Figure III-B: Modular of the Frontend Architecture* |

## Backend

|  |
| --- |
| *Figure III-C: Modular Backend Design Architecture* |

* **Admin Module:**
* Contact with other Modules: Admin module can manage the entire system, including creating, updating, and deleting objects such as Center, Class, Course, Teacher, Student, etc.
* Specifically: Admin module can send a request to Center Module to update center information, or Teacher Module to add new teachers.
* **Center Module:**
* Contact with Class Module and Course Module: Each center can organize many classes and courses. Therefore, Center Module will connect with Class Module to manage the center's classes, and with Course Module to organize courses.
* Specifically: Class Module gets center information from Center Module to assign classes to the corresponding center. Similarly, the Course Module also gets information from the Center Module.
* **Class Module:**
* Contact with Teacher Module and Student Module: Each class will have teachers and students participating. The Class Module will interact with the Teacher Module to assign teachers to each class, and with the Student Module to manage the list of students in the class.
* Contact with the Attendance Module: To manage attendance, the Attendance Module will need class information from the Class Module, and from there know the list of students attending.
* **Course Module:**
* Contact with the Class Module: A course can include many different classes. The Course Module will interact with the Class Module to know which class belongs to which course.
* Contact with the Voucher Module: To apply promotional vouchers to the course, the Course Module will interact with the Voucher Module to confirm valid voucher codes.
* **Voucher Module:**
* Contact with the Course Module and Payment Module: The Voucher Module connects with the Course Module to determine which vouchers are applied to which course. It will also interact with the Payment Module to give discounts when students pay for the course.
* Specifically: When students use vouchers, the Payment Module will confirm the information with the Voucher Module to calculate the amount to be paid after applying the discount code.
* **Teacher Module:**
* Contact the Class Module and Salary Module: The Teacher Module will connect with the Class Module to assign teachers to specific classes. In addition, to manage salaries, the Teacher Module will contact the Salary Module to calculate teachers' salaries based on the number of classes taught.
* Contact the Attendance Module: Teachers will take attendance for students in the class, so the Attendance Module will need teacher information from the Teacher Module to confirm the attendance list.
* **Slot Module:**
* Contact the Class Module and Teacher Module: The Slot Module manages class times. It will connect with the Class Module to determine the class times for each class, and with the Teacher Module to confirm that teachers can participate in teaching during that time frame.
* Contact the Student Module: To ensure that there are no overlapping class times, the Slot Module will check the student's class schedule through the Student Module.
* **Student Module:**
* Contact with Class Module, Course Module and Attendance Module: Student Module interacts with Class Module to manage students in each class, and with Course Module to confirm the courses that students register for. It also connects with the Attendance Module to track student attendance.
* Contact with Payment Module: When students register for a course, Payment Module will need to confirm information from Student Module to process payment for that course.
* **Salary Module:**
* Contact with Teacher Module: Salary Module will calculate salary for teachers based on the number of classes taught or other criteria from Teacher Module.
* Contact with Payment Module: When teachers receive salary, Salary Module will connect with Payment Module to make salary payment to teachers.
* **Attendance Module:**
* Contact with Class Module, Teacher Module and Student Module: Attendance Module needs information from Class Module to know the list of students in the class, from Teacher Module to manage teacher attendance, and from Student Module to track the attendance of each student.
* Contact with Slot Module: To confirm attendance schedule by time slot, Attendance Module will need information from Slot Module.
* **Payment Module:**
* Contact with Student Module and Voucher Module: Payment Module interacts with Student Module to process student tuition payment and with Voucher Module to apply discount code.
* Contact with Salary Module: Payment Module will make salary payment

# UI/UX Design - [Mockup](https://www.canva.com/design/DAGS3BmB__c/cXos9_7vzM014nCtzr72Nw/edit)

## Admin

### Login Page

|  |
| --- |
| *Figure IV-A-1: Login Page* |

* Description: Allow administrator to login home page

### Change password

|  |
| --- |
| *Figure IV-A-2: Change password* |

* Description: Allow administrator can change password

### Dashboard

|  |
| --- |
| *Figure IV-A-3: Dashboard Admin* |

* Description: Administrators can monitor the system through statistics in the form of charts.

### View Centers List

|  |
| --- |
| *Figure IV-A-4: View Center List* |

* Description: Administrators can view the current list of centers in the system.

#### Create new center

|  |
| --- |
| *Figure IV-A-4-a: Create new center* |

* Description: Administrators can create new center

#### View Center Detail

|  |
| --- |
| *Figure IV-A-4-b: View Center Detail* |

* Description: Administrators can view detailed information of each center.

#### Update Center

|  |
| --- |
| *Figure IV-A-4-c: Update Center* |

* Description: Administrators can edit information of each center.

#### Delete Center

|  |
| --- |
| *Figure IV-A-4-d: Delete Center* |

* Description: Administrators can delete the center.

#### View classes list

|  |
| --- |
| *Figure IV-A-4-e: View classes list* |

* Description: Administrators can view classes list

#### Create new class

|  |
| --- |
| *Figure IV-A-4-f: Create new class* |

* Description: Administrators can create new class

#### View Class Detail

|  |
| --- |
| *Figure IV-A-4-g: View Class Detail* |

* Description: Administrators can view info detail of class

#### View Teachers List

|  |
| --- |
| *Figure IV-A-4-h: View Teachers List* |

* Description: Admin can view teachers list.

#### Create New Teacher

|  |
| --- |
| *Figure IV-A-4-i: Create New Teacher* |

* Description: Admin can create teacher to add new teacher to the center.

#### View Teacher Detail

|  |
| --- |
| *Figure IV-A-4-j: View Teacher Detail* |

* Description: Admin can view teacher details.

### View Courses List

|  |
| --- |
| *Figure IV-A-6: View Courses List* |

* Description: Admin can view course list.

#### Create new course

|  |
| --- |
| *Figure IV-A-5-a: Create new course* |

* Description: Admin can create new course.

#### View Course Detail

|  |
| --- |
| *Figure IV-A-5-b: View Course Detail* |

* Description: Admin can view course details.

#### Update Course

|  |
| --- |
| *Figure IV-A-5-c: Update Course* |

* Description: Admin can update course info.

#### Delete Course

|  |
| --- |
| *Figure IV-A-5-d: Delete Course* |

* Description: Admin can delete course.

### View Vouchers List

|  |
| --- |
| *Figure IV-A-6: View Vouchers List* |

* Description: Admin can view vouchers list.

#### Create New Voucher

|  |
| --- |
| *Figure IV-A-6-a: Create New Voucher* |

* Description: Admin can create new voucher.

#### View Voucher Detail

|  |
| --- |
| *Figure IV-A-6-b: View Voucher Detail* |

* Description: Admin can view the details of the voucher.

#### Update Voucher

|  |
| --- |
| *Figure IV-A-6-c: Update Voucher* |

* Description: Admin can edit voucher info.

#### Delete Voucher

|  |
| --- |
| *Figure IV-A-6-d: Delete Voucher* |

* Description: Admin can delete the voucher.

### Schedule

#### View class schedule

|  |
| --- |
| *Figure IV-A-7-a: View class schedule* |

* Description: Allow admin to view class schedules list.

#### Update Schedule

|  |
| --- |
| *Figure IV-A-7-b: Update Schedule* |

* Description: Allow admin can update class schedules.

### View Teacher Salary List

#### View Teacher Salary List

|  |
| --- |
| *Figure IV-A-8-a: Salary Calculation For Teachers* |

* Description: Allows admin to view the salary of teachers in the center.

#### Salary Calculation For Teachers

|  |
| --- |
| *Figure IV-A-8-b: Salary Calculation For Teachers* |

* Description: Allows admin to manage and calculate salaries for teachers of the center.

## Teacher

### Login Page

|  |
| --- |
| *Figure IV-B-1: Login Page* |

* Description: Allow teachers to enter email and password, login to the system.

### Change Password

|  |
| --- |
| *Figure IV-B-2: Change Password* |

* Description: Allow the teacher to change the password.

### View Profile

|  |
| --- |
| *Figure IV-B-3: View Profile* |

* Description: Allow teachers to view their profiles.

### Update Profile

|  |
| --- |
| *Figure IV-B-4: Update Profile* |

* Description: Allows teachers to update information.

### View Class List

|  |
| --- |
| *Figure IV-B-5: View Class List* |

* Description: Allows teachers to view class lists.

### View Class Detail

|  |
| --- |
| *Figure IV-B-6: View Class Detail* |

* Description: Allows teachers to view detailed information for each class.

### View Student Detail

|  |
| --- |
| *Figure IV-B-7: View student Detail* |

* Description: Teachers can view student details.

### Teaching Schedule

#### View Teaching Schedule

|  |
| --- |
| *Figure IV-B-8-a: View Teaching Schedule* |

* Description: Teachers can view their teaching schedules.

#### Check Attendance

|  |
| --- |
| *Figure IV-B-8-b: Check Attendance* |

* Description: Teachers can view the class roster for each slot, and mark attendance for each student.

## Student

### Home Page

|  |
| --- |
| *Figure IV-C-1: Home Page* |

* Description: This is the main screen of the website. It consists of a Header (containing a dropdown list of centers, our logo, and a Google login button for students), a list of courses for the center selected on the Header, and finally the Footer.

### Login Page

#### Login

|  |
| --- |
| *Figure IV-C-2-a: Login Page* |

* Description: Students can log in to the system.

#### Login With Google

|  |
| --- |
| *Figure IV-C-2-b: Login With Google* |

* Description: Students can log in to the system by account google.

### View Profile

#### Profile

|  |
| --- |
| *Figure IV-C-3-a: View Profile Page* |

* Description: This is the page used to view student account information, including name, email, phone number, address, gender and date of birth. Below is a button to go to the student information update page.

#### Update Profile

|  |
| --- |
| *Figure IV-C-3-b: Update Profile* |

* Description: Students can update their profiles.

### View Class List

#### View Class List

|  |
| --- |
| *Figure IV-C-4-a: View Class List* |

* Description: Students can view their class list.

#### View Class Detail

|  |
| --- |
| *Figure IV-C-4-b: View Class Detail* |

* Description: Students can view detailed class information.

#### View Class schedule

|  |
| --- |
| *Figure IV-C-4-c: View Class schedule* |

* Description: Students can view their class schedules.

### Payment History Detail

#### View Payment History

|  |
| --- |
| *Figure IV-C-5-a: View Payment History* |

* Description: Students can view payment history.

#### View Payment History Detail

|  |
| --- |
| *Figure IV-C-5-b: Payment History Detail* |

* Description: Students can view detailed payment history information.