Service Specification

1. **Add student service**

Inputs:

email, password, picture, first-name, last-name, father’s name, birth-date, phone-num, edu-level

Outputs:

Message of successful operation or error message in case of failure.

Notes:

All input fields are to be provided, otherwise empty field error is thrown.

Email and phone number must be unique.

1. **Log in service**

Inputs:

email, password

Outputs:

Message of successful operation and navigate to dashboard or error message in case of failure.

Notes:

All input fields are to be provided, otherwise empty field error is thrown.

Email doesn’t exist or password mismatch are the errors.

1. **View class panel service**

Inputs:

Student’s ID

Outputs:

User’s current class(teacher, book,…),

next session time(to remind),

recent announcements,

last assignments(with info like score,…) and open assignments.

Notes:

User has to be authenticated and authorized as student.

All input fields are to be provided.

In case ID doesn’t exist or student has no active class errors are thrown.

1. **Payment service**

Inputs:

Student’s ID

Required Payment

Outputs:

Redirect to payment gateway(service providers api needs to be set up).

Notes:

User has to be authenticated and authorized as student.

Payments records must be checked(in case user has paid already).

All input fields are to be provided.

In case ID doesn’t exist or student has no active class errors are thrown.

1. **Full report of educational status and history service**

Inputs:

Student’s ID

Outputs:

Redirect to educational status report page.

Contains all previous class enrollments of the student(with level names and detailed grades) and the average of the final grades of all term as overall grade.

Notes:

User has to be authenticated and authorized as student.

All input fields are to be provided.

In case ID doesn’t exist or student has no enrollments errors are thrown.

1. **View teacher’s classes service**

Inputs:

Teacher’s ID

Outputs:

All classes that are held by the teacher.

Notes:

User has to be authenticated and authorized as teacher.

All input fields are to be provided.

Error is thrown if the teacher ID doesn’t exist or the teacher has no active class.

1. **Create Timeslot Service**

Inputs:

day, capacity, start\_time, end\_time

Outputs:

New timeslot is created in database

Notes:

User has to be authenticated and authorized as personnel.

All input fields are to be provided.

start\_time and end\_time should be stored as military time

Error is thrown if start\_time and end\_time don’t follow clock conventions.

Error is thrown if day of week is invalid.

1. **View Enrolled students service**

Inputs:

classID

Outputs:

List of all students in the class(who relate with the classID).

Notes:

User has to be authenticated and authorized as teacher or personnel.

All input fields are to be provided.

Throw error if classID is non-existant.

1. **Grading Student Service**

Inputs:

ID of Enrollment, studentID, grades[attendance, midterm, final]

Outputs:

Message of successful operation or error message in case of failure.

Notes:

User has to be authenticated and authorized as teacher.

All input fields are to be provided.

Error is thrown if the teacher ID or EnrollmentID doesn’t exist or the teacher has no active class or teacher and class aren’t related or the class is not held in current semester.

Error is thrown if studentID doesn’t exist or the student and class aren’t related.

Error is thrown if grades aren’t in correct range.

Warning is given if student already has grades.

1. **Personnel schedule classes service**

Inputs:

classID, timeslotID

Outputs:

Message of successful operation or error message in case of failure.

Notes:

User has to be authenticated and authorized as personnel.

All input fields are to be provided.

Error is thrown if timeslotID or classID doesn’t exist or the class is not active.

Error is thrown if the timeslot has no capacity.