

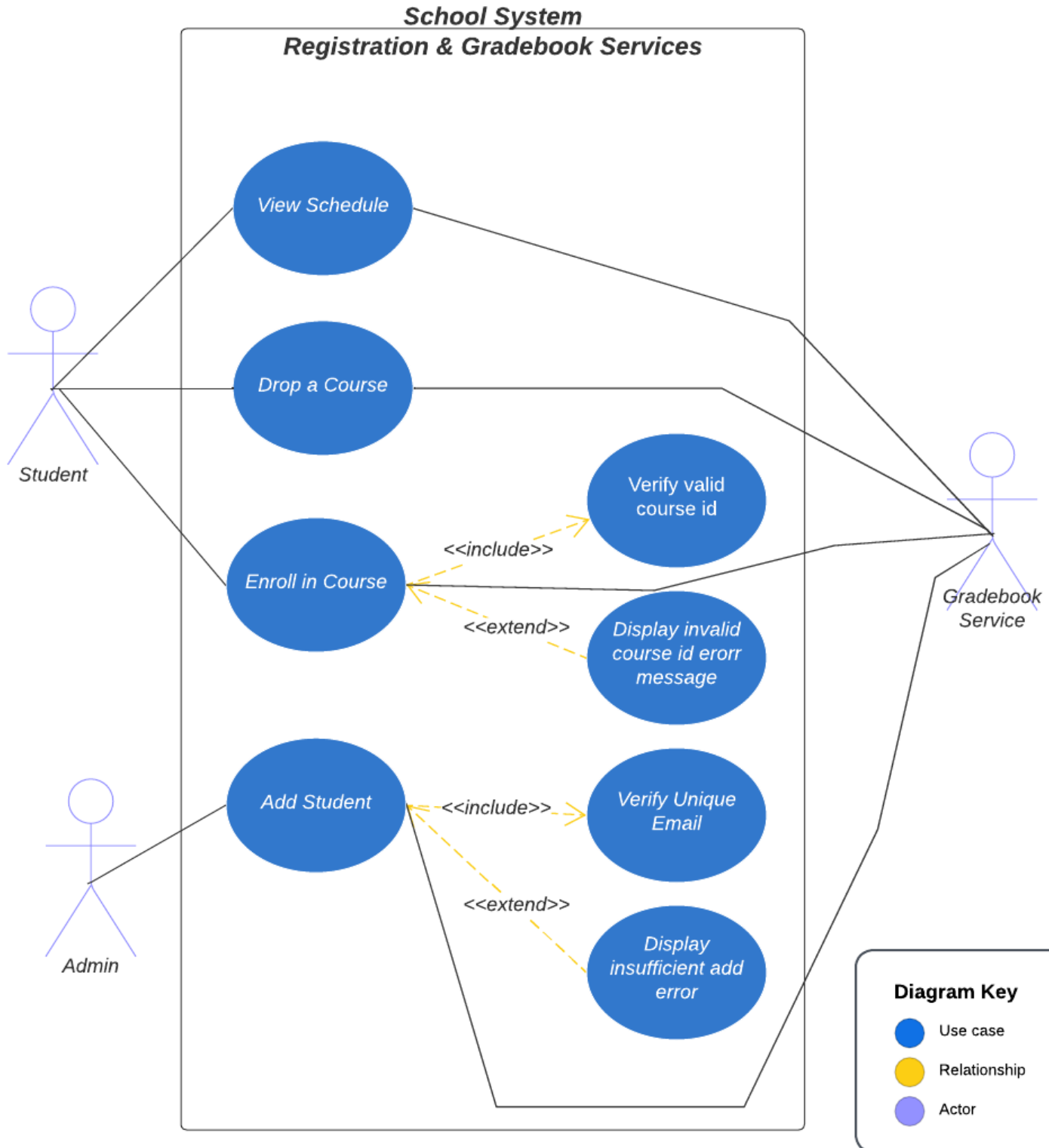
Software Requirements Specifications Document

Registration Service
Jennah Yasin
Partner: Yukio Rivera
CST 438 - Software Engineering

Software Requirements Specification Document

2.1 System Environment

Figure 1 - System Environment



3.2 Functional requirements

3.2.1 Add Student

| | |
|-------------------|---|
| Use Case Name | Add Student |
| XRef | Also see Figure 1 for what is checked/verified by the system when adding a student. |
| Trigger | The Administrator selects to add a new student to the database. |
| Precondition | The Administrator has accessed the Course Registration main screen. |
| Basic Path | <ol style="list-style-type: none"> 1. The system presents empty fields to enter the student information. 2. The Administrator enters the student's information and submits the form. 3. The system checks that the email address and name fields are not blank, and that the email entered doesn't already exist in the database. Then, it updates the database. |
| Alternative Paths | <p>If in step 2, either field is blank, the Administrator is instructed to fill out the empty fields.</p> <p>If in step 3, the entered email already exists in the database, the Administrator receives a "Student NOT added" toast message and is expected to enter a valid/unique email address.</p> |
| Postcondition | The Student has been added to the database. |
| Exception Paths | The administrator may abandon the operation at any time. |
| Others | The student information includes the student's name, email address, status (on hold/no hold), and status code. |

3.2.2 View Schedule

| | |
|---------------|--|
| Use Case Name | View Schedule |
| XRef | N/A |
| Trigger | The Student selects to view their schedule. (GET SCHEDULE button) |
| Precondition | The Student has accessed the Course Registration main screen and selected a term. |
| Basic Path | <ol style="list-style-type: none"> 1. The student selects the semester term for the schedule they want to |

| | |
|-------------------|---|
| | see and clicks the GET SCHEDULE button. 2. The system queries for all of the courses the student is registered in for that semester, and displays them on the screen. |
| Alternative Paths | If in step 1, the user does not select a semester term, they will be instructed to do so before being able to get their schedule. |
| Postcondition | The student views their schedule. |
| Exception Paths | The student may abandon the operation at any time. |
| Others | The student's schedule contains information about the classes they are enrolled in. Specific information includes the time, building, room, grade and number of each course. The student also has the option to Add a course or drop a course they are already enrolled in. |

3.2.3 Enroll in course

| | |
|-------------------|--|
| Use Case Name | Enroll in course |
| XRef | Also see View Schedule |
| Trigger | The Student selects to add a course to their schedule. |
| Precondition | The Student has accessed the Course Registration main screen and selected a term and clicked the "GET SCHEDULE" button. |
| Basic Path | <ol style="list-style-type: none"> 1. The student selects the semester term for which they want to add a course to and clicks the GET SCHEDULE button. 2. The system displays the student's schedule with an "ADD COURSE" button. 3. The student clicks the "ADD COURSE" button and is presented with an empty field for the course id from the system. 4. The student enters a valid course id (course id that actually exists). 5. The course is added to the student's schedule. |
| Alternative Paths | If in step 4, the student does not enter a valid course id, the system presents an "Error when adding" toast message and is expected to enter a course id for a course that actually exists. |
| Postcondition | The student is enrolled in the course. |
| Exception Paths | The student may abandon the operation at any time. |
| Others | When a student enrolls in a new course, the course will be shown on the |

| | |
|--|---|
| | student's schedule instantly. The student will also have the option to drop a course in their schedule. |
|--|---|

3.2.4 Drop a course

| | |
|-------------------|--|
| Use Case Name | Drop a course |
| XRef | Also see View Schedule |
| Trigger | The Student selects to drop a course from their schedule |
| Precondition | The Student has accessed the Course Registration main screen and selected a term and clicked the "GET SCHEDULE" button. |
| Basic Path | <ol style="list-style-type: none"> 1. The student selects the semester term for which they want to delete a course from and clicks the GET SCHEDULE button. 2. The system displays the student's schedule with an "DROP" button next to each course they are enrolled in. 3. The student clicks the "DROP" button and the system displays a small window asking for confirmation of dropping the course. 4. The student confirms the message by clicking the "OK" button. 5. The system deletes the dropped course from the student's schedule and in the database. |
| Alternative Paths | If in step 4, the student decides to click "cancel" on the message confirmation, the system will not proceed with the deletion. The student is expected to select and confirm a course they want to drop. |
| Postcondition | The student is dropped from a course. |
| Exception Paths | The student may abandon the operation at any time. |
| Others | When a student drops a course from their schedule, the course will be removed from their schedule list instantly. |

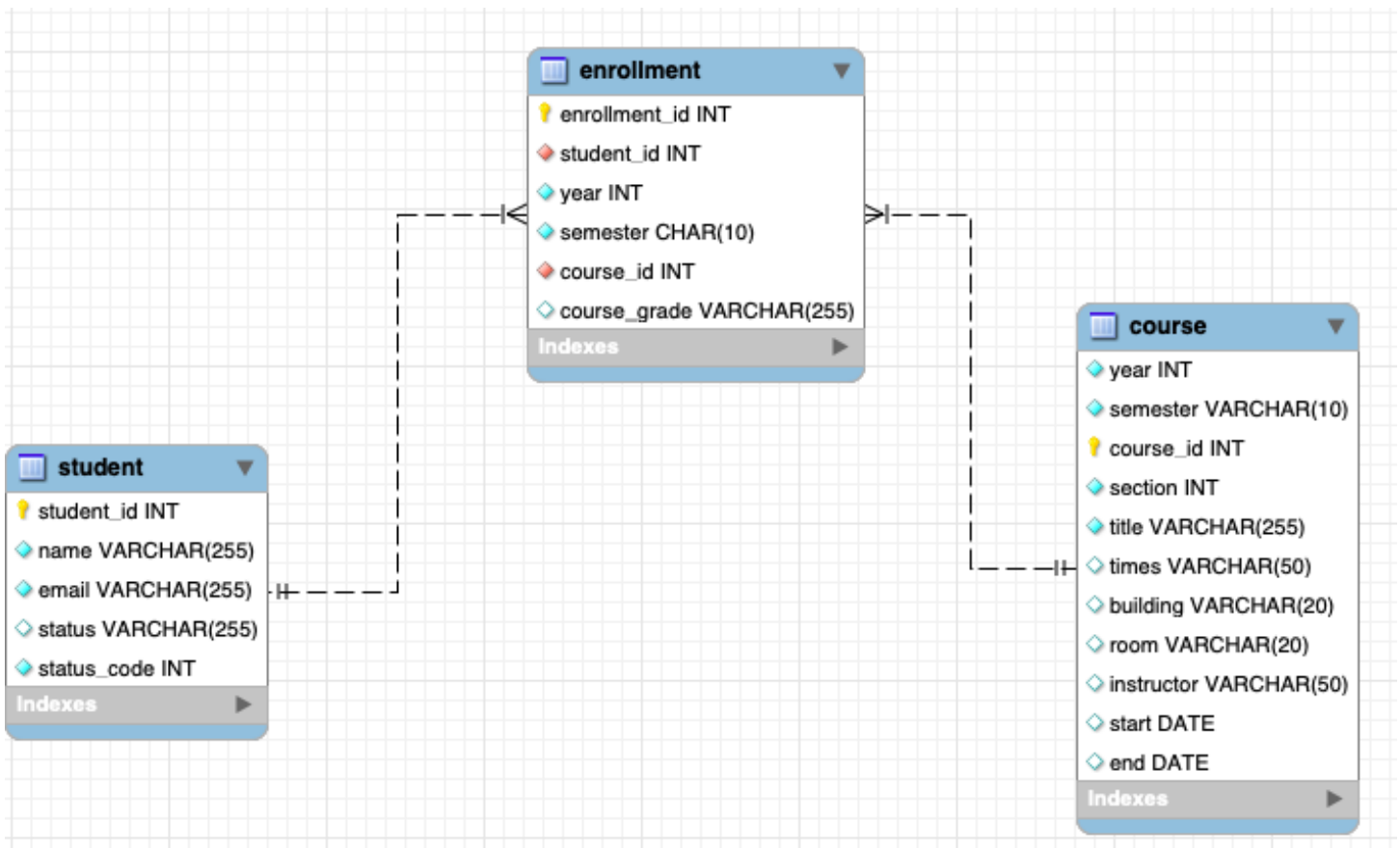
3.3 Non-Functional Requirements

The registration service of the system is developed to ensure a high quality experience and performance for the users by having a user-friendly interface. The users are also able to navigate through the website with reliable security of their personal information through the use of OAuth in the website. Basically, only authorized users (either students or admins) are able to access the student information and the courses they are registered in. In terms of browser compatibility, this service is compatible with current known browsers and their versions, like Google Chrome, Safari, and Firefox.

3.4 Logical Database Diagram

Figure 2: ER Diagram

Register Database



Database Diagram Explanation:

- This diagram shows the relationships between the three entities of the register database. The entities are: enrollment, student, and course.

- A student can enroll in many courses, and a course has many enrollments.

Here is a description of each attribute of each entity of the database:

Student Entity:

| Data Item | Type | Description |
|-------------|---------|--|
| student_id | Integer | The student's id. |
| name | Text | The student's name. |
| email | Text | The student's email address. |
| status | Text | The student's account status (on hold/no hold) |
| status_code | Integer | 1: on hold, 0: no hold |

Enrollment Entity:

| Data Item | Type | Description |
|---------------|---------|--|
| enrollment_id | Integer | The id of the enrollment. |
| student_id | Integer | The student's id. |
| year | Integer | The year that the enrollment is in. |
| semester | Text | The semester name of the term. |
| course_id | Integer | The id of the course. |
| course_grade | Text | The grade that the student received in the course. |

Course Entity:

| Data Item | Type | Description |
|-----------|---------|---|
| year | Integer | The year that the course begins in. |
| semester | Integer | The semester that the course exists in. |
| course_id | Integer | The id of the course. |
| section | Integer | The section number of the course |
| title | Text | The name of the course. |
| times | Text | The time schedule of the course. |

Software Requirements Specifications Document

| | | |
|------------|------|---|
| building | Text | The building number that the course takes place in. |
| room | Text | The room number that the course takes place in. |
| instructor | Text | The name of the instructor for the course. |
| start | Date | The date for the start of the course |
| end | Date | The date for the end of the course. |