

Title: System Sequence Diagram and Activity Diagram

Madeline Abbott and Benjamin Hogan

Sequence Diagrams

Use Case 1: Updating Student Records

Actor:

User: The person who interacts with the system (ex. tutor, admin, etc.),

Object:

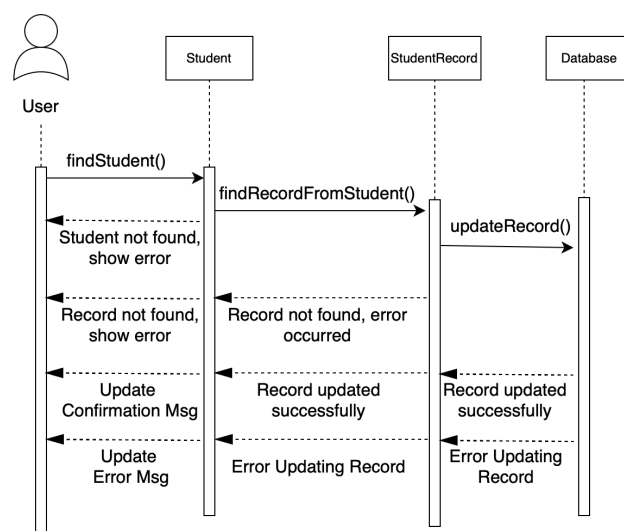
Student: the student whose records are being managed.

StudentRecord: the record of the student.

Database: the storage system for student records.

Steps of Updating a Student Record:

1. The User opens the “Manage Student Records” section in the interface.
2. The User identifies the Student object based on the input of StudentID.
3. The system sends the request to the server and finds the corresponding StudentRecord.
4. The server returns a response finding the record (either a success or error) to the interface.
5. The system processes the request and communicates with the Database by either inserting, updating, or deleting a StudentRecord.
6. The server returns a response updating the record (either a success or error) to the interface.
7. The changes are saved to StudentRecords via the Database.



Use Case 2: Scheduling an Appointment

Actor:

User: The person who interacts with the system (ex. tutor, admin, etc.),

Object:

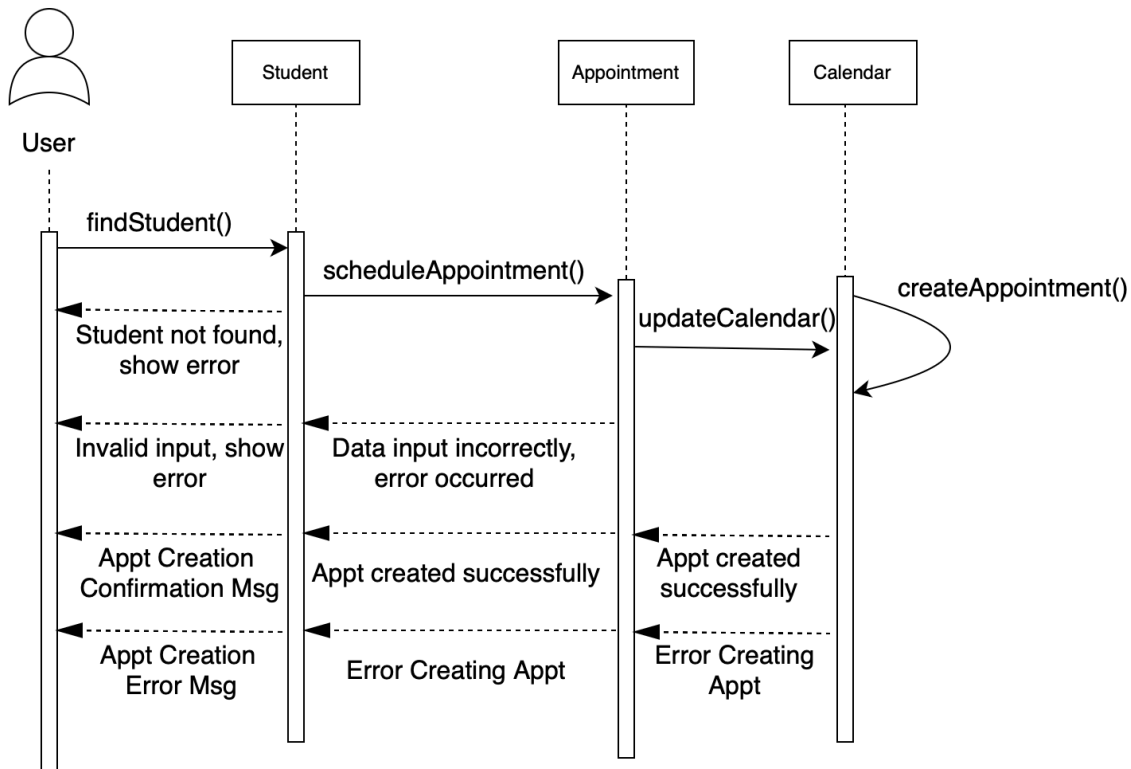
Student: The person for whom the appointment is being scheduled.

Appointment: Represents the transaction of scheduling an appointment including the time, date, and subject.

Calendar: The tool used to represent dates and times, as well as schedule appointments.

Steps of Scheduling an Appointment:

1. The User selects the Student which will have an Appointment by using their StudentID.
2. The User selects a time and date for the Appointment.
3. The system checks the Calendar to validate the time and date for the Appointment.
4. The system either returns an error or confirmation message for the Appointment.
5. The Appointment is saved in the Calendar.



Activity Diagrams

Use Case 1: Creating a Student and Student Record

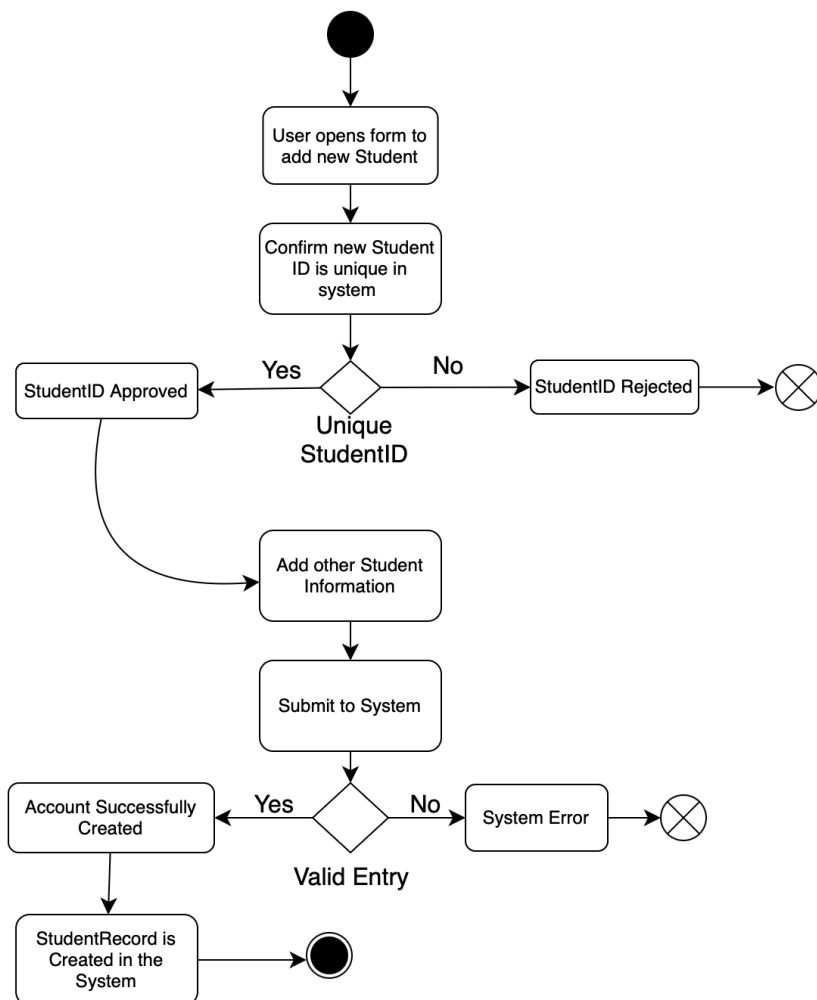
States:

Initial State: The user opens the system and begins the student creation process.

Final State: 1. The user receives confirmation of the new student creation, and a new student is added to the database. 2. The user receives an error message with an explanation, and a new student is not added to the database.

Actions:

The user is logged in and selects the “Add New Student” option. The user is given a form and inputs student information including name, student ID, date of birth, major, address, contact information, tutoring concerns, and other notes. User submits Student details. System validates the entered information. If the validation is successful, the system creates a StudentRecord. System saves the new StudentRecord to the database. System returns a success or error message to the interface accordingly.



Use Case 2: Deleting a Student and Student Record

States:

Initial State: The user opens the system and begins the student deletion process.

Final State: 1. The user receives a confirmation message, and the student and their record are successfully deleted. 2. The user receives an error message, and the student and their record remain in the system. 3. The user cancels the deletion, and the student and their record remain in the system.

Actions:

The user is logged in and selects the “Delete Student Record” option. The user is given a form to confirm the student they want to delete, where the user inputs the student ID. If both attributes do not match a student, the system returns an error. Otherwise, the system displays the student information along with a textbox that says “Are you sure you want to delete this student’s information?” along with two buttons, each saying “yes” or “no.” If the user selects no, a confirmation message that the record was not deleted will be displayed. Otherwise, the system will delete the student and their record. If the process is successful, the system will return a confirmation message of the deletion. Otherwise, the system will return an error message that assures that the student and their record were not deleted.

