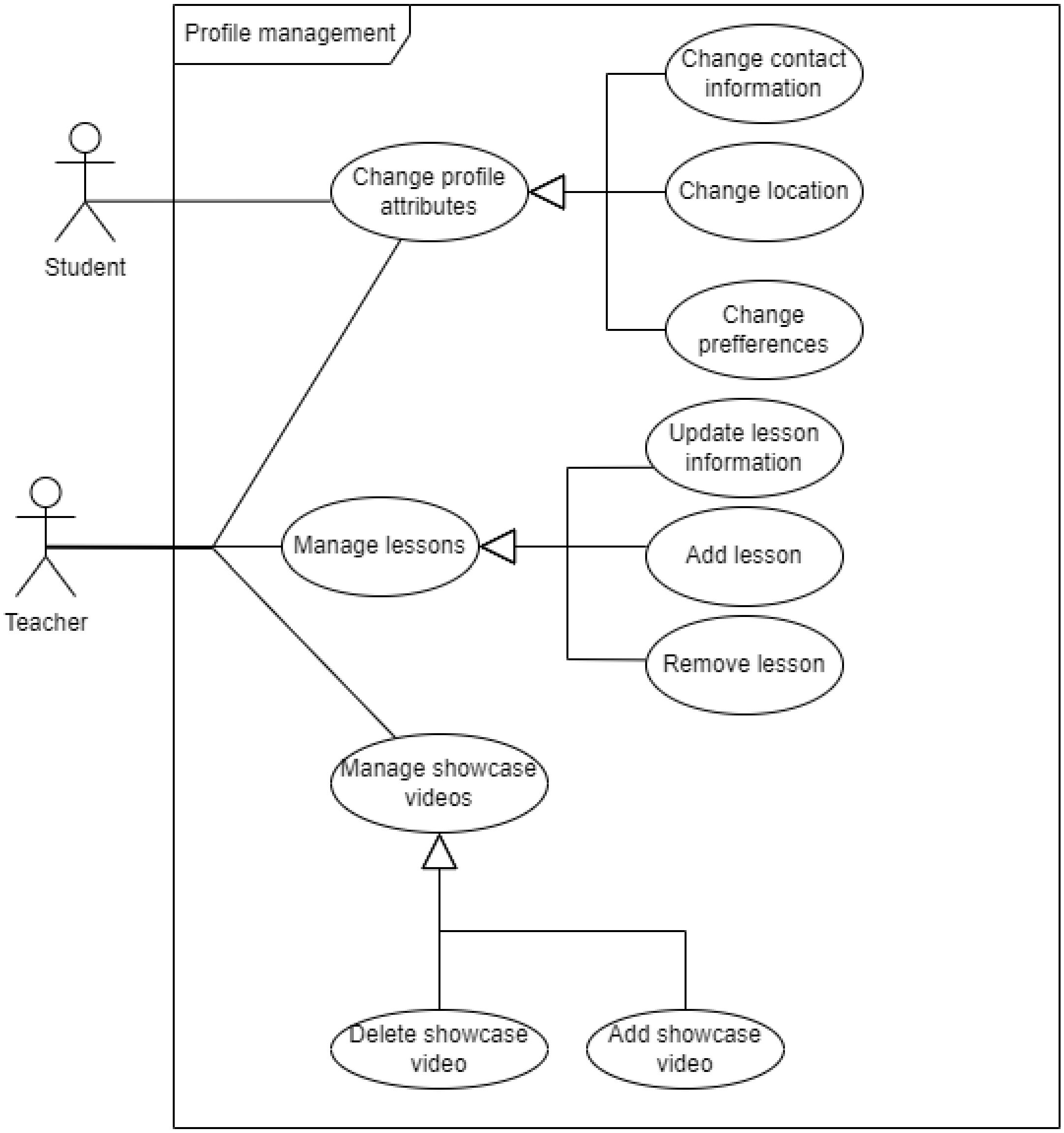
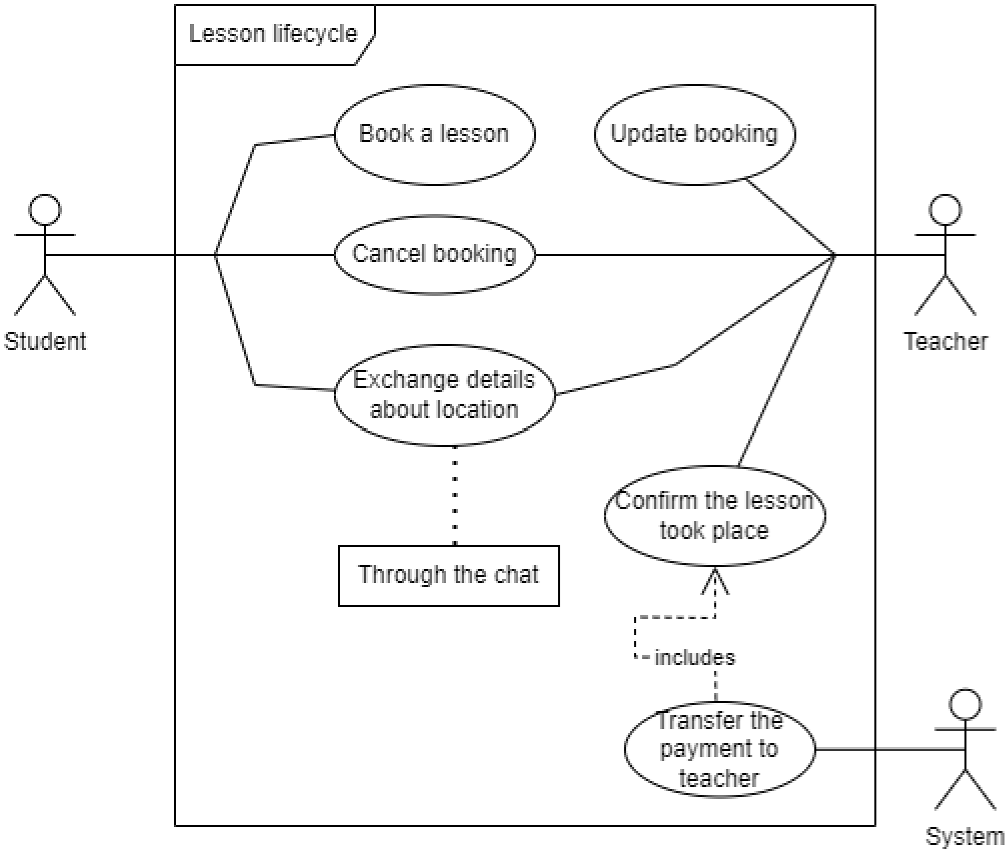
NoteBridge 1

The report provides an overview of the UML use cases, class diagram, and SQL schema.

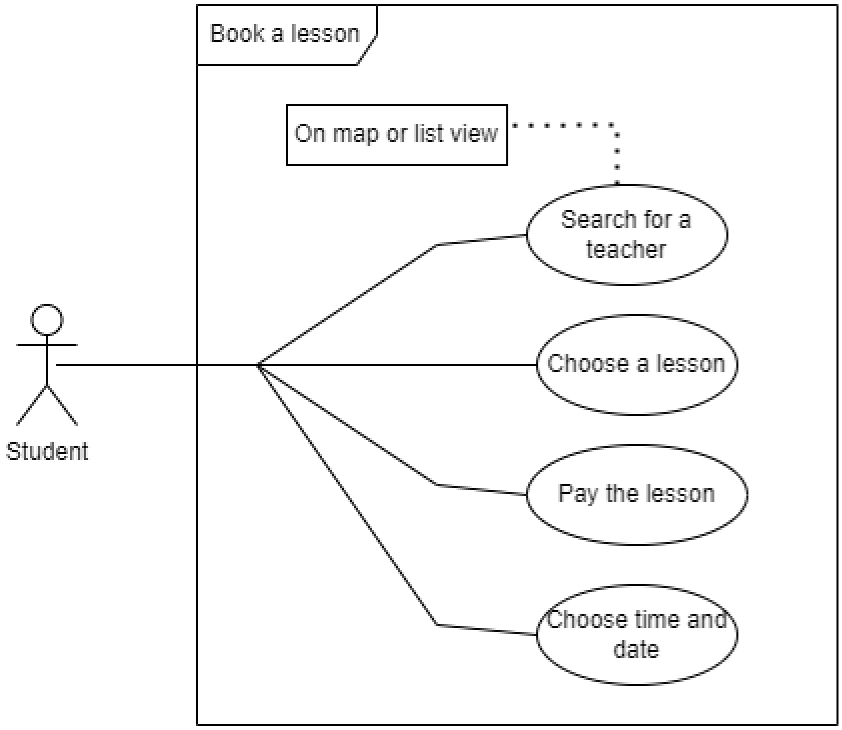
1. Use Case Diagrams



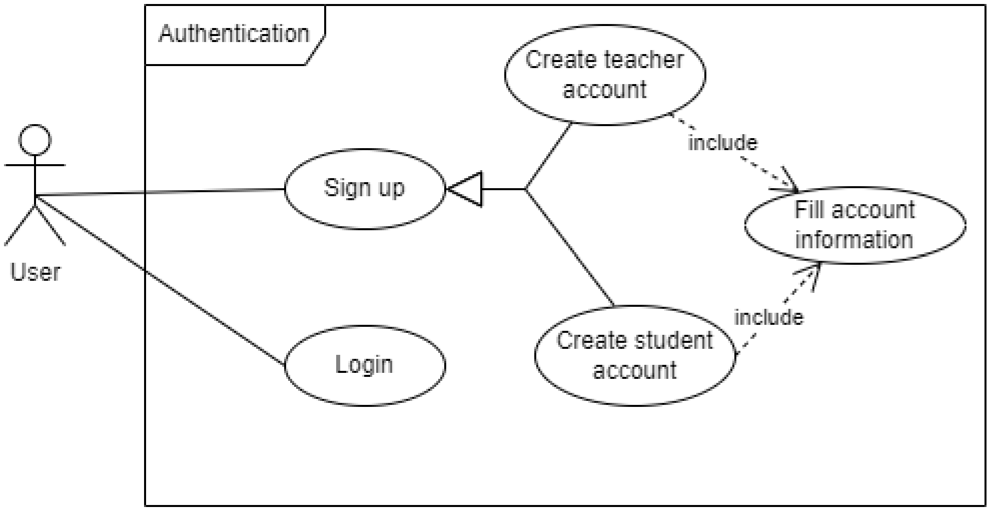
The profile management use case diagram displays what modifications users can do with their profiles. There are 2 types of users: *student* and *teacher*. Both *teacher* and *student* can change (general) profile attributes, which includes changing contact information (mail. phone number, etc.), changing their location and changing preferences (online/offline, instruments taught/learnt). A *teacher* can also manage their lessons, which implies adding, removing and updating them. Also, *teachers* can display showcase videos on their profile, which implies that they could both add and remove videos from their profile



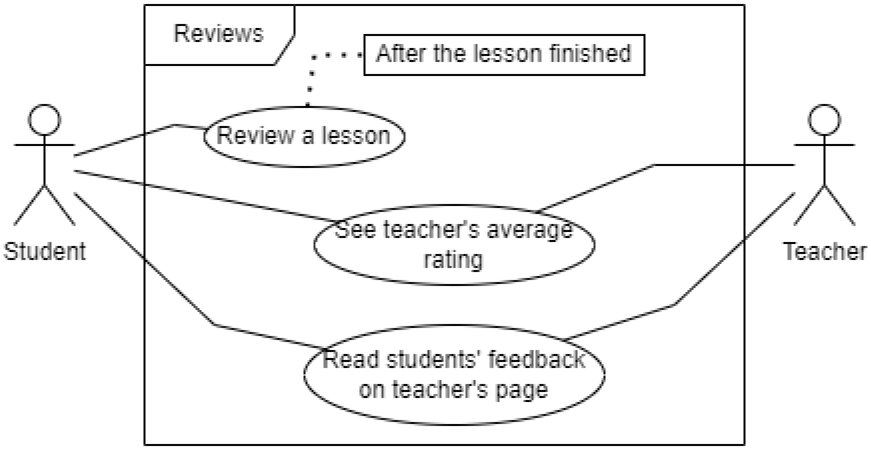
Here is a diagram that displays the lifecycle of a lesson. Firstly, a *student* can book a lesson. This implies also paying for it (the money doesn't go directly to the teacher). Afterwards, the *teacher* has to accept or reject a booking (updating the booking status from pending). In case the *teacher* accepts the booking, both *teacher* and *student* exchange details about where the lesson will take place (does not matter if online or offline). This is done by communicating on the implemented chat on the website. At any point in time, a *teacher* or a *student* can cancel a booking or it can be rescheduled by the *teacher.* After the lesson took place the *teacher* has to confirm that the lesson took place, so that the *system* could transfer the payment to the teacher.



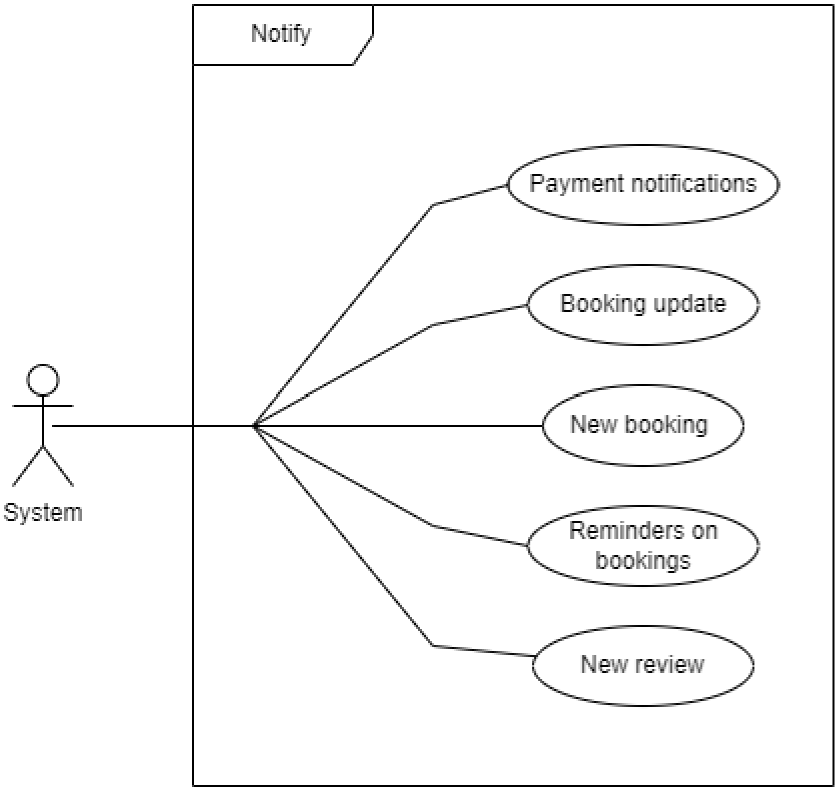
This diagram extends the use case *Book a lesson* from the previous use case diagram. Here we can see that the *student* can search for a teacher using the map or list view, choose a lesson of a selected teacher, choose an available time slot and pay for the lesson.



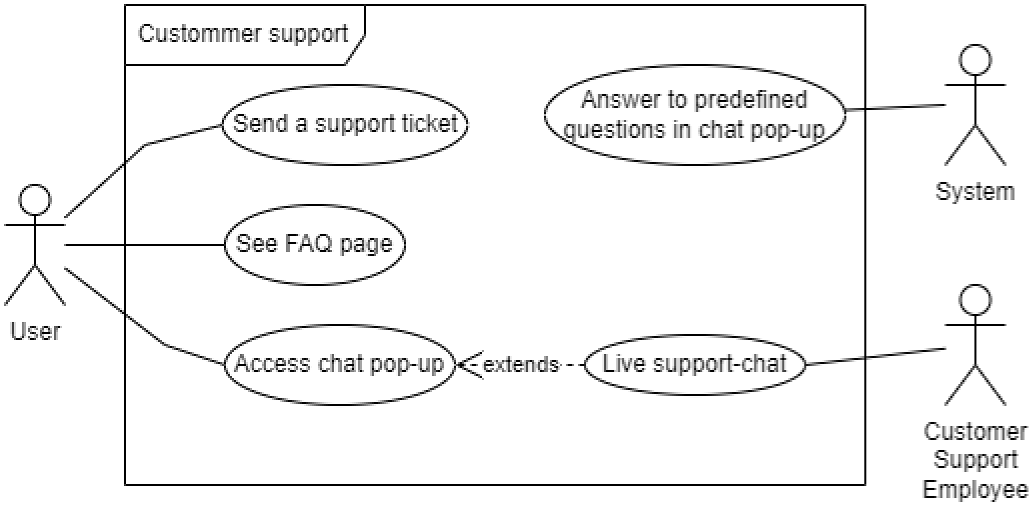
Here is a simple diagram that shows how authentication works. If the *user* has an account, they login, otherwise they have to sign up. When signing up they have to choose if they want to be a teacher or a student (a teacher is also a student, but with extra functionalities). Both of these decisions imply filling the required account information (name, phone number, email, etc.)


Reviews
After the lesson finished
Review a lesson
See teacher's average rating
Read students' feedback on teacher's page
Student
Teacher

The *Reviews* use case diagram shows us that a *student* can review a lesson (after it finished) and afterwards both the *student* and *teacher* can see the teacher's average rating and read students' feedback on his page.



This use case diagram shows what notification are pushed to users by the *system*



This diagram shows us how customer support works. If a *user* has a question/problem they could try to look for an answer/solution on the FAQ page or send a support ticket. Otherwise, they can access the cat pop-up and if the predefined Q&A don't satisfy the *user*, a *customer support employee* will solve their issue through the live support-chat.

2. Class diagram

The following is a description of the structure of our project's database (and class diagram).

1. Users: This table stores information about users of the platform. It includes attributes such as id, email, password, full_name, country_code, and city. The country_code attribute references the id column of the Country table.

2. Country: This table maintains a list of countries with their respective id and name values. The id column in the Users table references this table.

3. Admin: The Admin table represents the administrators of the platform. It contains an id column that serves as a primary key, referencing the id column in the Users table.

4. Teacher: The Teacher table stores information specific to the platform's teachers. It includes attributes such as id, description, experience, instruments, and avg_rating. The id column references the id column in the Users table.

5. Teacher_instruments: This table represents the association between teachers and the instruments they can teach. It includes the teacher_id and instrument_id columns, both of which reference the corresponding tables.

6. Instrument: The Instrument table contains a list of musical instruments with their corresponding id and name values.

7. Teacher_schedule: This table represents the schedule of teachers. It includes attributes like teacher_id, start_time, duration, and online (true or false). The teacher_id column references the id column in the Teacher table.

8. Lesson: The Lesson table represents the available music lessons on the platform. It includes attributes like id, teacher_id, price, instrument_id, skill_id, online, and description. The teacher_id, instrument_id, and skill_id columns reference the corresponding tables.

9. Booking: This table stores information about lesson bookings made by students. It includes attributes like id, student_id, lesson_id, teacher_id, start_time, is_canceled, and is_finished. The student_id, lesson_id, and composite foreign key teacher_id and start_time reference the corresponding tables.

10. Review: This table holds the reviews submitted by students for the teachers. It contains attributes like id, teacher_id, student_id, rating, and comment. The teacher_id and student_id columns reference the id column in the Teacher and Users tables, respectively.

11. Skill: This table stores various musical skills with their respective id and name values.

12. Payment: The Payment table keeps track of payment transactions associated with bookings. It contains attributes such as id, booking_id, amount, payment_timestamp, and status. The booking_id column references the id column in the Booking table.

13. Message: The Message table stores messages exchanged between users of the platform. It includes attributes such as id, sender_id, receiver_id, message_text, and timestamp. The sender_id and receiver_id columns reference the id column in the Users table.

Tables:

1. **Users**

Column	Data Type	Description	Constraint
id	Integer	Stores the id of the user	Primary key
email	Varchar(255)	Stores the email of the user	
password	Varchar(64)	Stores the password of the user	
full_name	Varchar(255)	Stores the full name of the user	
country_code	Varchar(3)	Stores the country code of the user	Foreign key
city	Varchar(50)	Stores the city of the user	

2. **Country**

Column	Data Type	Description	Constraint
id	Varchar(3)	Stores the id of the country	Primary key
name	Varchar(50)	Stores the name of the country	

3. Admin

Column	Data Type	Description	Constraint
id	Varchar(3)	Stores the id of the admin	Primary key, Foreign key

4. Teacher

Column	Data Type	Description	Constraint
id	Integer	Stores the id of the teacher	Primary key, Foreign key
description	Varchar(255)	Stores the description of the teacher	
experience	Varchar(50)	Stores the experience of the teacher	
instruments	json	Stores the instruments	
avg_rating	float	Stores the average rating of the teacher	Check

5. **Review**

Column	Data Type	Description	Constraint
id	Int	Stores the id of the review	Primary key
teacher_id	Int	Stores the id of the teacher	Foreign key
student_id	Int	Stores the id of the student	Foreign key
rating	Float	Stores the password of the user	
comment	Varchar(255)	Stores the comment of the student	

6. **Teacher Schedule**

Column	Data Type	Description	Constraint
id	Integer	Stores the id of the user	Primary key
email	Varchar(255)	Stores the email of the user	
password	Varchar(64)	Stores the password of the user	
full_name	Varchar(255)	Stores the full name of the user	
country_code	Varchar(3)	Stores the country code of the user	Foreign key
city	Varchar(50)	Stores the city of the user	

7. **Instrument**

Column	Data Type	Description	Constraint
id	Integer	Stores the id of the instrument	Primary key
name	Varchar(100)	Stores the name of the instrument	Unique

8. **Skill**

Column	Data Type	Description	Constraint
id	Integer	Stores the id of the skill	Primary Key
name	Varchar(100)	Stores the name of the skill	Unique

9. **Lesson**

Column	Data Type	Description	Constraint
id	Integer	Stores the id of the lesson	Primary key
teacher_id	Integer	Stores the id of the teacher	Foreign key
price	Decimal	Stores the price of the lesson	
Instrument_id	Integer	Stores the instrument id of the lesson	Foreign key
skill_id	Integer	Stores the skill id of the lesson	Foreign key
online	Boolean	Stores if the lesson was online or offline	

description	Varchar(255)	Stores the description of the lesson	

10. Booking

Column	Data Type	Description	Constraint
id	Integer	Stores the id of the booking	Primary key
student_id	Integer	Stores the id of the student	Foreign key
lesson_id	Integer	Stores the id of the lesson	Foreign key
teacher_id	Integer	Stores the id of the teacher	Foreign key
start_time	Timestamp	Stores the start time of the booking	Foreign key
is_canceled	Boolean	Stores if the booking was canceled	
is_finished	Boolean	Stores if the booking was finished	

11. Payment

Column	Data Type	Description	Constraint
id	Integer	Stores the id of the payment	Primary key
booking_id	Integer	Stores the id of the booking	Foreign key
amount	Decimal	Stores the amount payed	
payment_stamp	Timestamp	Stores the time of the payment	
status	Varchar(20)	Stores the status of the payment	

12. Teacher_instruments

Column	Data Type	Description	Constraint
teacher_id	Integer	Stores the id of the teacher	Primary key, Foreign key
instrument_id	Integer	Stores the id of the instrument	Primary key, Foreign key

13. Message

Column	Data Type	Description	Constraint
id	Integer	Stores the id of the message	Primary key
sender_id	Integer	Stores the id of the sender	Foreign key
receiver_id	Integer	Stores the id of the receiver	Foreign key
message_text	Varchar(255)	Stores the message	
timestamp	Timestamp	Stores the time	