

Attendance Manager App Tutorial: Guide for Future Project

Jesus Cornejo
cornejoj@oxy.edu
Occidental College

1 Introduction

1.1 Why this tutorial?

For my computer science project, I am aiming to develop a Campus Event app for the campus community. Users will be able to view events happening on campus listed through the app. The app's goal is to promote student engagement on campus by providing easy access to event information and enabling users to express interest in attending events. The Attendance Management app [1], which tracks student attendance in a MySQL database, provided a solid foundation for the Campus Event app as similar classes and activities are needed for its implementation. The tutorial provided insightful knowledge on creating an interactive app and connecting it to a database. Although installing unfamiliar software like XAMPP to run MySQL was challenging, configuring the files and completing the rest of the tutorial process was straightforward.

2 Methods

2.1 Tutorial Overview

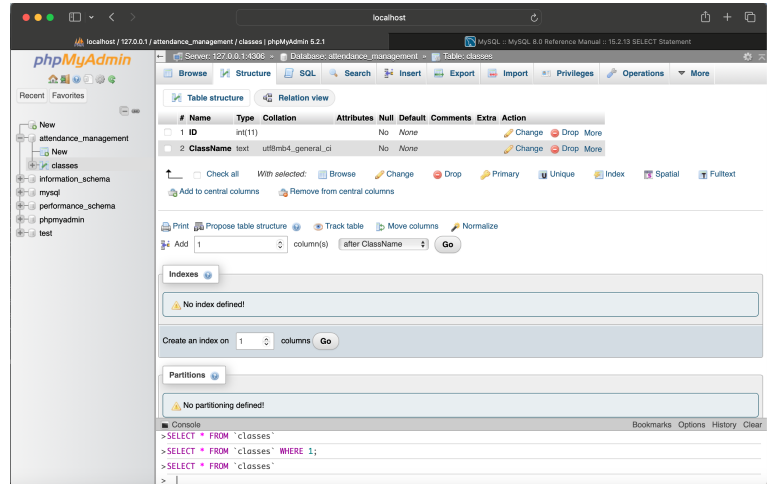
The Attendance Management app is built in Java using Android Studio. The application makes use of a MySQL database through *phpadmin* from a software called *XAMPP*.

2.2 Classes and How They Work:

1. Database.java: This class handles database operations such as retrieving, adding, updating, and deleting classes, students, and sessions from the database. It interacts with the MySQL database using JDBC (Java Database Connectivity).

2. Class.java: Represents a class entity with properties like ID and className. Instances of this class are used to store class information.

3. Student.java: Represents a student entity with properties like ID, firstName, lastName, email, and tel. Instances of



this class are used to store student information.

4. Session.java: Represents a session entity with properties like ID, students, subject, and date. Instances of this class are used to store session information, including the students attending the session.

5. SessionEditor.java: This activity allows users to create or edit sessions. Users can set the subject and date for a session, and they can also add or remove students from the session.

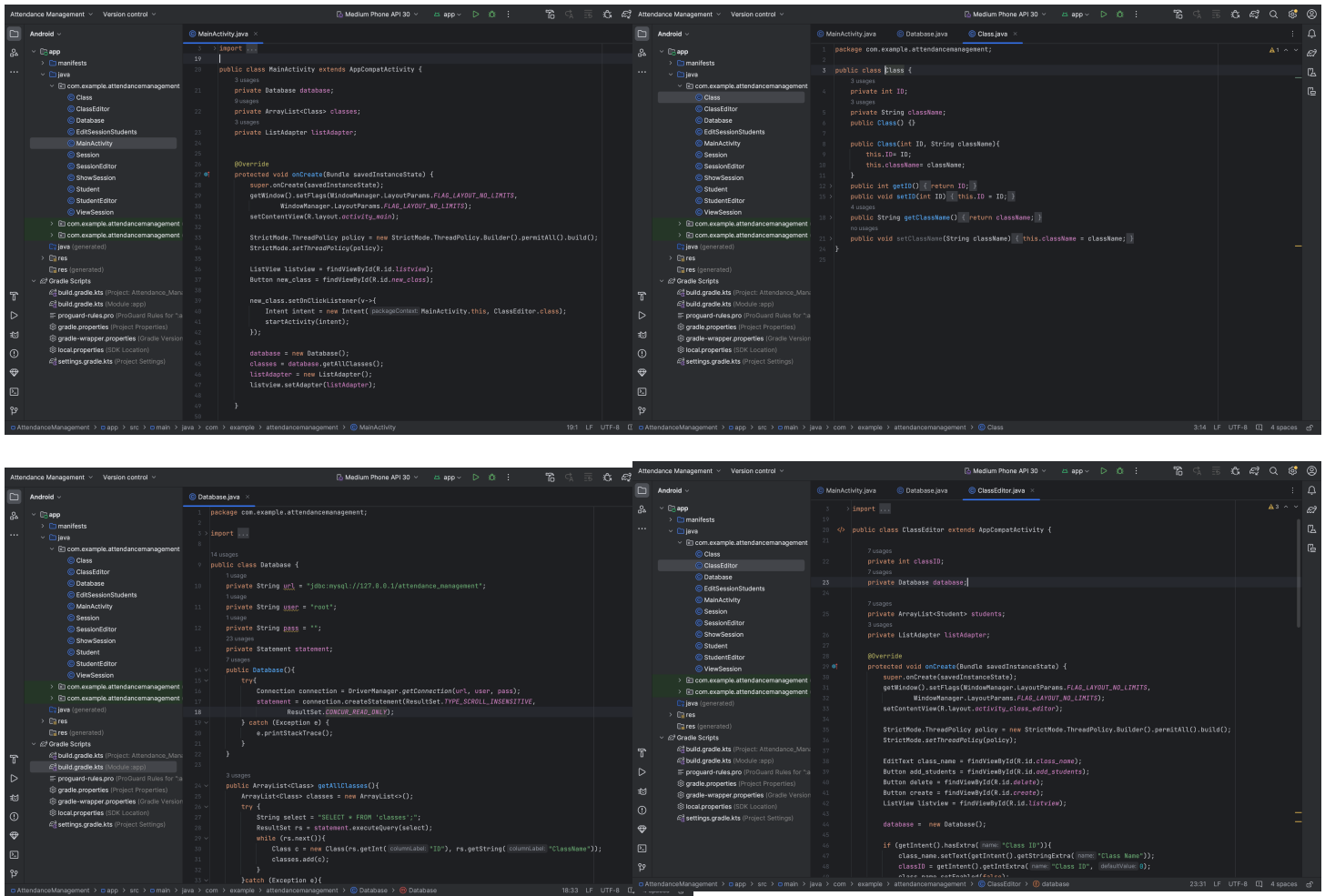
6. StudentEditor.java: This activity allows users to create or edit student information. Users can set the first name, last name, email, and telephone number for a student.

7. ClassEditor.java: This activity allows users to create or edit classes. Users can set the class name and add or remove students from the class.

8. ShowSessions.java: This activity displays a list of sessions for a specific class. Users can create a new session or click on a session to edit its details.

9. ViewSession.java: This activity displays the details of a specific session, including the list of present and absent students. It retrieves session information from the database and separates students into present and absent categories.

10. EditSessionStudents.java: This activity allows users to edit the list of students attending a session. Users can check or uncheck students to mark them as present or absent, re-



spectively.

These classes work together to create a comprehensive attendance management system. The Database class serves as the central hub for database interactions. It utilizes JDBC (Java Database Connectivity) [2] to establish connections with the MySQL database. Through this class, users on the app can do operations like retrieving, adding, updating, and deleting data entries like classes, students, and sessions.

Furthermore, the classes 'Class.java', 'Student.java', and 'Sessions.java' are meant for entity management. The 'Class.java' class represents a class entity, meaning that instances of this class holds data about different classes like ID and class name. Similarly, the 'Student.java' class represents a student, with its instances containing information such as ID, first/last name, email, and phone number. Lastly, 'Session.java' is meant to represent a class/school session (could be a lecture, field trip, event, etc.) by saving the date, name of the subject, ID, and a list of attending students. The other classes and activities provide user interfaces for creating, editing, and managing sessions, students, and classes.

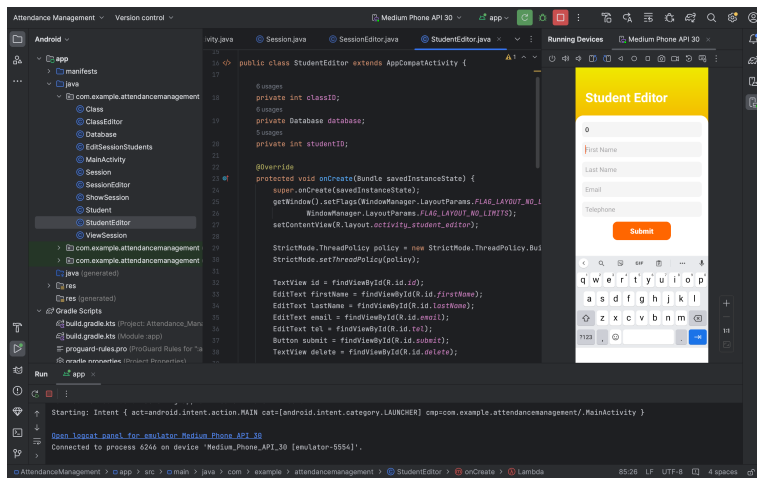
3 Metrics and Results

The goal of the app was to create a database of student classes and connect it to the attendance manager app in order to make changes to the database. Since the application can create and delete new instances of students and classes in the database, the goal of the app was achieved.

In order to create an app that fosters student engagement on campus, it is important to know how many students are attending events. With the attendance manager app, users can take attendance through a student-classes database. One desired activity for the event app is to see how many students are interested in attending an event, helping both event planners and students prepare for the attendance.

4 Reflection

The process of following this tutorial was not too difficult. The tutorial I found at YouTube channel *Tech with Bob's* was straightforward, but there are a few issues for my project moving forward. The process of setting up the



database through MySQL was not covered in the tutorial I followed, so more knowledge will be required to setup the events database. I am also a bit worried about the other activities and functions that I will need to develop on my own for my app to be singular to the school. I do however enjoy this topic and have found a bit of passion in trying to create working apps.

References

- [1] Bob, Tech with. *Attendance Manager App for Android using Java and MySQL*. Last accessed 11 March, 2024. 2023. URL: <https://www.youtube.com/playlist?list=PL-cxxMmn1xXEcfgae8Gjv21VMWxyilrVn>.