

Group 4 Project Design Specifications

Summary

This project will be to create a simple course management system, focused on assignments and timetables.

PostgreSQL will be used as the database engine, with Python and Django for the backend.

The system will be equally accessible through an Android app, and an HTML web frontend.

Students will be able to use the system to register for courses, view timetables, and receive and submit assignments.

Courses will be taught in sections to students, each section by one teacher. An assignment, once assigned to the class, can be completed by individual students, or groups of students.

Use Case Analysis

Admin:

The admin must be able to login, create and delete courses and although not his main task also be able to create and delete sections. Furthermore, the admin is able to look up all existing courses and sections. The admin must also verify teacher's account.

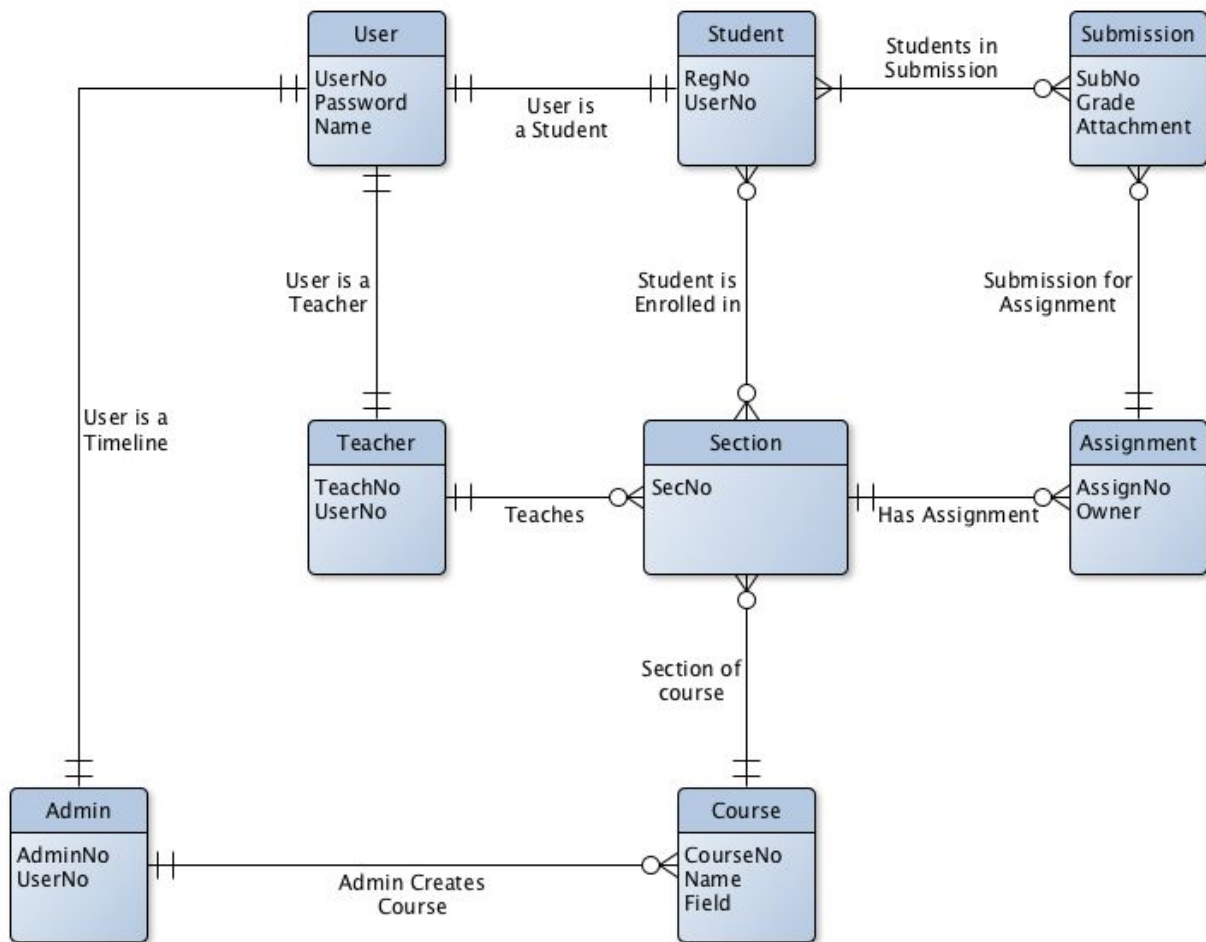
Teacher:

The teacher can create an account, which needs to be verified by the admin. With an existing account the teacher can login. The teacher's tasks are creating and deleting sections for existing courses. The teacher can create, delete and update assignments for courses he teaches himself. Assignments can be either individual or group assignments. For each assignment the teacher must specify how much they count for the final grade. The teacher also grades these assignments. Moreover, the teacher can look up all students that are registered in his/her course including their grades. The teacher is also able to look up the timetable containing all taught courses.

Students:

A student is able to create an account and login with the existing one. Students can register for sections and drop sections they are registered in. All registered courses are displayed in a timetable. A student can look up the assignments for each course and submit their solution as an individual or group assignment, depending on the assignment type created by the teacher. A submission can be changed or deleted afterwards. But no submission or change can be done after the deadline.

Conceptual Data Model



Description of Technology

The backend will be programmed in Python, using Django. Postgres will be used as the database engine.

The Android backend will be coded in C# in Xamarin, and interface with the backend via a REST API using JSON.

An HTML web frontend will be created through Django. SASS will be used to generate stylesheets. As this is not going to be used on a large scale, Python's builtin web server will be used for web hosting, reducing overall complexity.

There will be a number of complex queries and some many-to-many relationships. Django's handling of these will be analysed, and possible optimizations will be planned.

Weekly milestones

Oct 22	<ul style="list-style-type: none">• Environment Familiarization• Design table specifications• Design Android UI
Oct 29	<ul style="list-style-type: none">• Set up Django DB pt.1• Connect Android to Django with REST APIs
Nov 5	<ul style="list-style-type: none">• Implement Android UI• Set up Django DB pt.2
Nov 12	<ul style="list-style-type: none">• Finish Android functionality• Set up SASS
Nov 19	<ul style="list-style-type: none">• Finish web front end• Query optimization
Nov 26	<ul style="list-style-type: none">• Testing, optimization, and bye week

We've given ourselves plenty of time to finish the milestones by the projected dates. Knowing that project delays or school can provide unforeseen setbacks, these deadlines afford breathing room in the event the project gets pushed back. This can be seen in that the last week of the project has been allotted as a bye week, with the project otherwise being finished by Nov 19th.

Work Division

Android – Thom Watkin

Django – Jimmy Pardey, Stefanie Fritz

Postgres/SASS/Django consulting – Chris Papke