COMP 464 WEB PROGRAMMING AND INTERNET TECHNOLOGIES

FINAL PROJECT REPORT

Course Scheduling System

İlknur Atalay 041701026

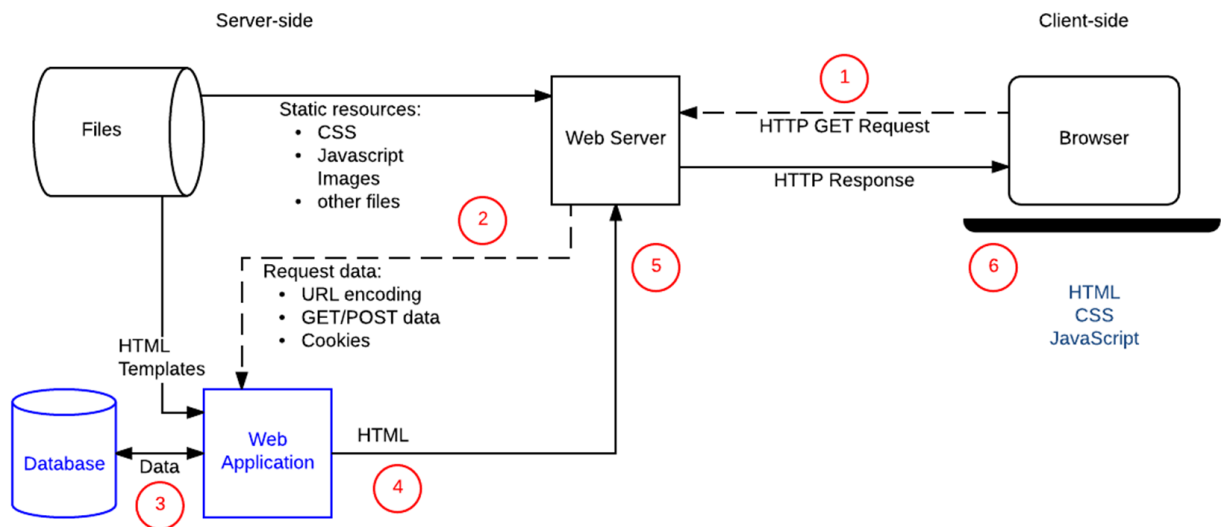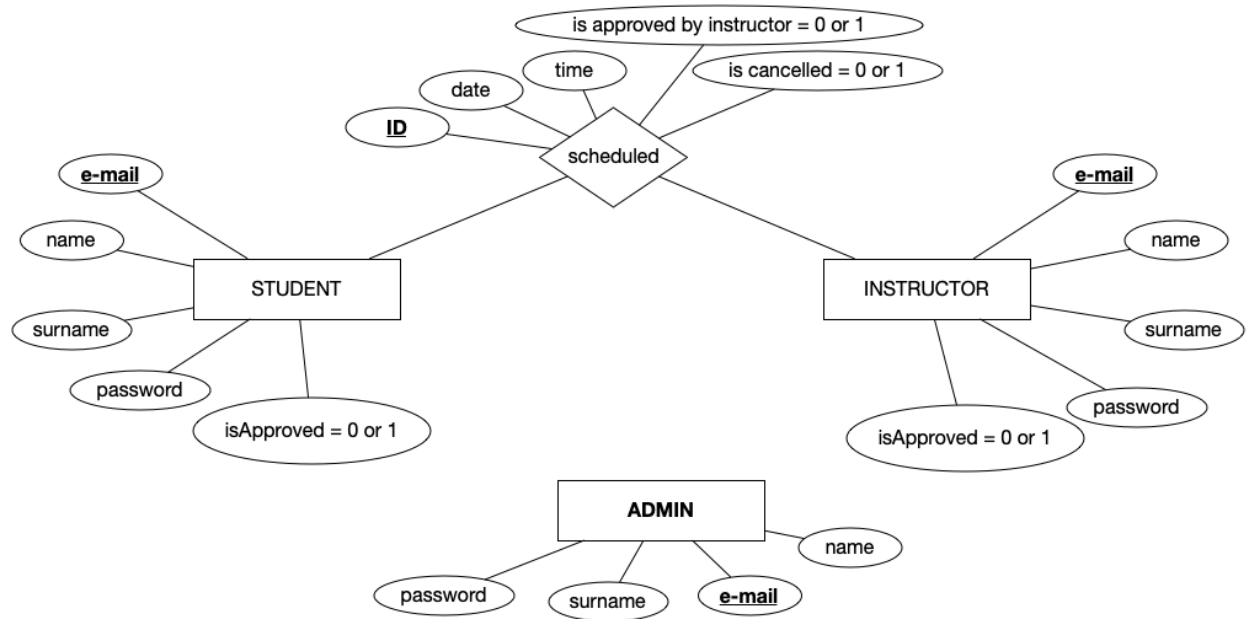Yasemin İzel Çangal 041701021

Ertuğrul Yılmaz 041701030

Team

Yasemin and Ilknur from our team members initially worked on the front end. While they were doing the front end part, Ertuğrul started the backend part. When the front end part is finished, the back end part is completed and the pages are combined as a team. Team members are satisfied with the division of labor and put in equal effort.

Client side

When we first open our website, we have a welcome page. Similar to the Mef.sis page, there are tabs for student, instructor and name. Then we have the User Account Form. It is a basic sign-up page. Email validation is done. When the account is created, a warning (flash) message is given saying 'Account has been created'. On our Course Scheduling Form page, when you log in as a student, you can see the names of all the instructors, choose the instructor you want, the appropriate day and time, and create your meeting. All hours are displayed and the desired time is selected from there. The same operations can be done for the Instructors on the Instructor page, which is a separate page. Each meeting has the feature of being cancelled. If logged in as admin, you can see the information kept in the database. All users are working separately. We use the POST method for receiving inputs.

Server side

For the backend side we used Flask, python, Psycopg2 to access database in python and we used Postgre to hold our database. Our ER diagram is as it appears in the image.

Entity-Relationship Diagram

scheduled (relationship):
- is approved by instructor = 0 or 1
- is cancelled = 0 or 1
- time
- date
- **ID**

STUDENT:
- **e-mail**
- name
- surname
- password
- isApproved = 0 or 1

INSTRUCTOR:
- **e-mail**
- name
- surname
- password
- isApproved = 0 or 1

ADMIN:
- password
- surname
- **e-mail**
- name

Web Architecture Diagram

Server-side

Client-side

Files → Web Server

Static resources:
- CSS
- Javascript Images
- other files

(1) HTTP GET Request

HTTP Response

(2) Request data:
- URL encoding
- GET/POST data
- Cookies

HTML Templates

Database — Data — Web Application **(3)**

(4) HTML

(5)

Web Server

(6) Browser

HTML
CSS
JavaScript

Enhancements

We did not develop your application on a serverless architecture.

Presentation

The link of your website

What to do next

If we had three more weeks to work on the project, we would have put effort into improving the UI design. Our database could be improved and new functions could be added.