King Saud University

College of Computer and Information Sciences

Department of Software Engineering

كلية علوم الحاسب والمعلومات
قسم هندسة البرمجيات

Advanced Web Applications Engineering - SWE 481

SMART SCHEDULE

Phase 1: Initial Phase.

#	Name	ID
1	Sarah AlOmran	444200911
2	Roha alswoaiegh	444200551
3	Fatima Alsaggaf	444200112
4	Khawlah Alawadi	444204382

Section #	79619
Group #	4
Supervisor(s)	Bushra Mohammed Alkadhi

Introduction

Project Description.

Manually building academic schedules is time-consuming and error-prone, affecting students, academic advisors, and scheduling committee members. Each semester, students struggle to create feasible schedules that align with their academic plans and personal preferences, while academic advisors spend significant time understanding individual needs to provide guidance.

This project addresses these challenges by developing **SmartSchedule**, a web-based platform that automates and optimizes academic scheduling. Leveraging an AI-powered API, the system generates personalized schedule recommendations tailored to students and faculty.

SmartSchedule also enables real-time collaboration between scheduling and load assignment committees, allowing seamless coordination among stakeholders. Key features include real-time updates with auto-save, version control to track historical changes, and a scalable architecture to support growing institutional needs.

Definitions

Scheduling Tasks: the set of activities carried out to prepare and finalize the university's academic schedule. These include the preparation of course offerings, conducting departmental consultations, distributing requirements, handling feedback, executing schedule copying, managing modification periods, and finalizing the schedules.

Level overview: Visualize the schedule per group in a level, along with the number of sections, the assigned instructors, and the number of students in each section.

Course overview: Visualize classroom assignments to sections, number of students per section, instructors' assignments to sections.

elective course preferences form: a form provided to students in which they indicate their preferred elective courses for the upcoming semester. The form allows students to rank or select courses according to their interests, which the system uses to help generate a fair and balanced course schedule.

Rules: constraints that the schedule must adhere to and be designed following them.

Table of content

Requirements

Functional Requirements

1- User's Requirements:

1.1. Users shall be able to log in via university email and password.
1.2. Users shall be able to reset their password by receiving a one-time verification code via email.
1.3.Users shall be able to log out.
1.4.Users shall be able to search for courses.
1.5.Users shall be able to filter the course search results by instructors' names, the time slots in which the course is taken at, and the sections that take the course.
1.6.Users shall be able to view the course overview.
1.7.Users shall be able to view the level overview.
1.8.Users shall be able to view the replies on their comments.
1.9.Users shall get notified when there is a reply on their comment.

2- Student’s Requirements:

2.1.Students shall be able to submit the elective course preferences form.
2.2.Students shall be able to register into the system by their university email and password.
2.3.Students shall be able to view the preliminary schedule.
2.4.Students shall be able to provide comment on the preliminary schedule.
2.5.Students shall be able to view exams' times.
2.6.Students shall be able to view the updated version of the schedule.
2.7.Students shall be able to receive notifications when the schedule is published.
2.8.Students shall be able to receive notifications when the schedule is updated.
2.9.Students shall be able to receive a notification before the last day of the elective courses preferences form’s deadline if they haven’t submitted it.
2.10.Students shall be able to receive a notification when the elective course preferences form is published.
2.11.Students shall be able to view their comment on the schedule.

3- Faculty’s Requirements:

3.1.Faculty members shall be able to register into the system by their university email, password, and their role.
3.2.Faculty members shall be able to view the preliminary schedule.
3.3.Faculty members shall be able to view the updated version of the schedule.
3.4.Faculty members shall be able to provide comments on the schedules.
3.5.Faculty members shall be able to receive notifications when the schedule is published.
3.6.Faculty members shall be able to receive notifications when the schedule is updated.
3.7.Faculty members shall be able to view their comments on the schedule.

4- Registrar’s Requirements:

4.1.Registrars shall be able to enter irregular student data (name, remaining courses from past levels and the courses needed to prevent the student from falling behind) manually.

4.2. Registrars shall be able to input the number of students in each level.

4.3. Registrars shall be able to receive notification from scheduling committee to enter irregular students' data.

5- Scheduling Committee's Requirements:

5.1.Scheduling Committee shall be able to register into the system by their university email, password, and their role.
5.2.Scheduling committee shall be notified of important upcoming scheduling tasks deadlines from the university's official timeline.
5.3.Scheduling committee shall be able to modify the default number of students which is 25 in each section.
5.4.Scheduling committee shall be able to notify registrars to enter irregular students' data.
5.5.Scheduling Committee shall be able to manually enter slots and course information (name, credit hours, and course code) received from other departments.
5.6.Scheduling committee shall be able to modify the generated schedule.
5.7. Scheduling committee shall be notified when the generated schedule is published.

5.8. Scheduling committee shall be able to add new rules.
5.9. Scheduling committee shall be able to modify rules.
5.10. Scheduling committee shall be able to delete existing rules.
5.11. Scheduling committee shall be able to view other users' comments.
5.12. Scheduling committee shall be able to reply on other users' comments.
5.13. Scheduling committee shall be able to delete other users' comments.
5.14. Scheduling committee shall be able to approve or decline load committee modifications.
5.15. Scheduling committee shall be able to view all versions of the schedule.
5.16. Scheduling committee shall be able to restore a schedule from previous versions.

6- Load Committee's Requirements:

6.1. Load committee shall be able to register into the system by using the university email, password, and selecting their role.
6.2. Load committee shall be able to view the initial schedule.
6.3. Load committee shall be able to view the preliminary version of the schedule.
6.4. Load committee shall be able to view the updated versions of the schedule.
6.5. Load committee shall be able to provide comments on the initial schedule.
6.6. Load committee shall be able to receive notifications when the schedule is updated.
6.7. Load committee shall be able to add new rules.
6.8. Load committee shall be able to modify rules.
6.9. Load committee shall be able to delete existing rules.
6.10. Load committee shall be able to view their comment on the schedule.
6.11. Load committee shall be able to suggest modifications on the schedule.

7- System’s Requirements:

7.1. The system shall provide elective course forms to collect student preferences.

7.2. The system shall automatically save user changes to persistent storage at intervals not exceeding 60 seconds of activity.

7.3. The system shall automatically calculate the ideal number of course sections by dividing the number of enrolled students by the default section size of 25.

7.4. The system shall prevent users from submitting the elective course form more than once, providing a notification if already submitted.

7.5. The system shall check the schedules of irregular students before applying changes to other schedules.

7.6. The system shall cancel proposed schedule changes if they conflict with irregular students’ schedules.

7.7. The system shall notify schedulers when proposed schedule changes are canceled.

7.8. The system shall regenerate the schedule based on changes approved by the scheduling committee if no conflicts exist with irregular students’ schedules.

7.9. The system shall ensure elective courses are scheduled to provide equal registration opportunities for students of all levels.

7.10. The system shall automatically reserve 12:00 PM to 1:00 PM daily as a break period in all schedules.

7.11. The system shall schedule courses and their prerequisite-linked courses in the same time slot.

7.12. The system shall ensure days off are evenly distributed among all student groups.

7.13. The system shall schedule elective courses in early morning or late afternoon time slots.

7.14. The system shall enforce scheduling of extended sessions as continuous blocks.

7.15. The system shall automatically generate initial schedules based on predefined rules, course slots from other departments, and preferences of regular and irregular students.

7.16. The system shall provide a dashboard for level overview.

7.17. The system shall provide a dashboard for course overview.

Non-Functional Requirements

1- Scalability

1.1. The system shall support at least 10,000 concurrent users across multiple institutions.

2- Useability

2.1. Users shall be able to navigate to any page at the website within 3 clicks.
2.2. User shall be able to view schedule in 1 click.

3- Security

3.1. The system shall validate data to prevent any sort of data breach.
3.2. The system shall authenticate users based on their roles (faculty member, student, scheduling committee member, load committee member).

4- Performance

4.1. The system shall propagate user changes for example (text insertion, deletion, cursor movement) to all other collaborating clients within 200 milliseconds under standard network conditions.
4.2. The system shall render pages and process user interactions such as (form submissions, filtering) with a maximum latency of 2 seconds under normal load.

5- Portability

The system shall be designed to allow deployment on both web and Android mobile platforms with minimal code modifications.

Links

- The GitHub setup : gitHub
- The Demo: Demo