CS347 Project

Connor Wallace
Nick Albright
William Krzyzkowski
Wesley Llamas

Overview

TA Scheduler / Calendar

We are designing a system to augment Teaching assistant hours for JMU. This system does not intend to completely replace what we are doing now, but simply to enhance the experience.The system will consist of several views including a login system for Teaching Assistants. This will record their preferred schedules and class specialties before the start of each semester. We will also have a manager login for the faculty manager who will be able to view all of the TA's preferences etc. The manager will also be able to schedule TA's, given their preferences. The TA's will also need an interface to request shift covers, and the manager will need an interface to approve or deny these covers. We want to have a calendar that shows how many TA's are in the lab at a given time, but it should not display names, possibly specialties, to the public. Students should be able to sign in to the system, possibly anonymously, and state what class they are there for. From the potential features we want to include how many TA's are scheduled for a shift as well as how many questions are on the board. We will also add a user role, the TA Lead, who will have access to Lead specific information such as faculty contact numbers and schedules with names etc. We will also add information to for all TA's such as when timesheets are due and TA meetings, these will most likely be set by the TA Manager.

** Underline means optional

Required Features

1. Onboarding
    1. Roles:
        1. TA
        2. TA Lead
        3. Manager
    2. register with system
        1. Name
        2. Email
    3. solicit preferences
        1. Scheduling
        2. Courses
        3. Min/max hrs/wk?
2. Login
    1. Based on info from registration
3. Define Semester
    1. Set TA lab dates/hours
4. Manually Schedule Semester
    1. Give Manager a UI: Looking at capabilities, preferences, semester definition
5. Swap duties
    1. Request a cover
    2. Queue potential cover-ers
    3. Coordinator approval of a cover makes it official
6. Show calendar
    1. Different views: public doesn't show TA names
7. Collect data
    1. Student sign in to office hours

Supplementary Features

1. User Roles
    1. already required: TA, Manager
    2. Possible additions: Lead TA
2. TA data
    1. How many TAs are in lab hours?
    2. When do TAs arrive and depart?
    3. What is the current TA schedule for that day?
3. Attendee data
    1. How many people are in lab hours?
    2. What classes are lab hour attendees enrolled in?
4. Notifications of assignments, swap successes, requests for changes
    1. Email, text, push - not sure which yet

Plan

Milestone 1 (Oct 24):

Primary goal: mockup static frontend for display. Dynamic behavior and backend connection will come later

- Frontend HTML? Mockup for all pages. Stubbed out react components > Static web page
    - Pages:
        - Student **(Wesley)**
            - Sign-in/Sign-out function
            - Questionnaire to fill out what class student is in/additional info
        - TA **(Connor)**
            - Schedule Preferences
            - Shift Swap Request/Accept
            - Personal Schedule
        - Lead TA **(Nick)**
            - Above features + additional contact info
            - Access to more detailed information of TA Schedule (schedule for all TAs)
        - Manager **(Will)**
            - Detailed schedule view/schedule maintenance
            - View TA Preferences to aid with scheduling
            - View statistics from student entry
- Set up Git Repository **(Nick)**

Milestone 2 (Oct 31):

- Server - PHP Backend
    - Authentication/login - will just be with a local plaintext until database is set up **(Nick)**
    - Establish Session/Cookie to provision access for each role **(Wesley)**
    - Accessing/storing schedule data for display on frontend **(Connor)**
    - Accessing/storing student responses for display on manger view in frontend **(Will)**

Milestone 3 (Nov 7):

- Database - maybe Mongo (need to discuss with the professor as far as hosting goes) **(Connor)**
- Scheduling implemented
    - Manager shift assignment **(Wesley)**
    - Shift swap request/accept implemented **(Will)**

- Analytics & display of student response feedback **(Nick)**

Milestone 4 (Nov 14):

- Style improvements, bug fixes **(Nick)**
- Calendar implemented **(Connor and Wesley)**
- User Acceptance testing with a couple TAs - have them sign in with mock accounts and report opinions on user interface. **(Will)**

Final Deliverable

- Deployment to server **(Nick)**
- Implement Acceptance Testing recommendations **(Will)**
- Hosting database(s) **(Connor)**
- Implement fixes from professor feedback **(Wesley)**

Questions/Research:

- What database technology should we use? Where should we host it?
- Local development is fine, but where will we eventually deploy our application?
- Should we use a frontend framework like React (and have to take time to learn it), or stick to simple HTML/CSS/JS
- Does it make sense to develop frontend/backend first? How do we mock data before our database is setup/hosted?
- How are Sessions handled in PHP? Need to research this - have done something similar but in Java Spring
- How are we going to find a group to use for user acceptance testing?