Senior Project

Design Document

Drop-in Tutoring Web Application

Team Members:

Timothy Baker, Jeremiah Caban, Eric Ray Carpizo, Aanchal Chaturvedi, Huy Ly, Christopher Mariani, and Bryan Nunez

|  |  |
| --- | --- |
| Table of Contents | Page |
| 1. Summary |  |
| 1. Technology Stack    1. HTML / CSS    2. JavaScript    3. PHP    4. Java |  |
| 1. Application Views |  |
| * 1. Home   2. Log In   3. Drop-in Tutoring Schedule   4. Survey/Feedback Form   5. Navigation Bar   6. Tutor Session Logbook   7. Shared Knowledgebase   8. Session Report Page |  |
| 1. RESTful API |  |
| 1. Database Design |  |
| 1. Implementation Plan    1. Features: Log-in, API implementation    2. Views: Home, Log-in, Navigation bar    3. Database instantiation |  |

Summary

The group will design and create a web application that will assist various departments with the drop-in tutoring services that they offer. Tutors themselves will be able to log in to access a session logbook, which will help keep track of each individual tutoring session as they take place. They will also have access to a shared knowledgebase for passing back and forth useful information to other tutors as they see fit. Admins, typically a departmental chair or secretary, will be able to log in to view detailed reports of all sessions that have been held over any given period. The system will allow for admins to easily add new tutors or courses to the schedule to keep up with demand as time goes on. Basic features (no login required) will be made available for students, these include: home screen with scrolling news panel, survey/feedback form, and a schedule viewer for the drop-in tutoring services.

Technology Stack

HTML / CSS

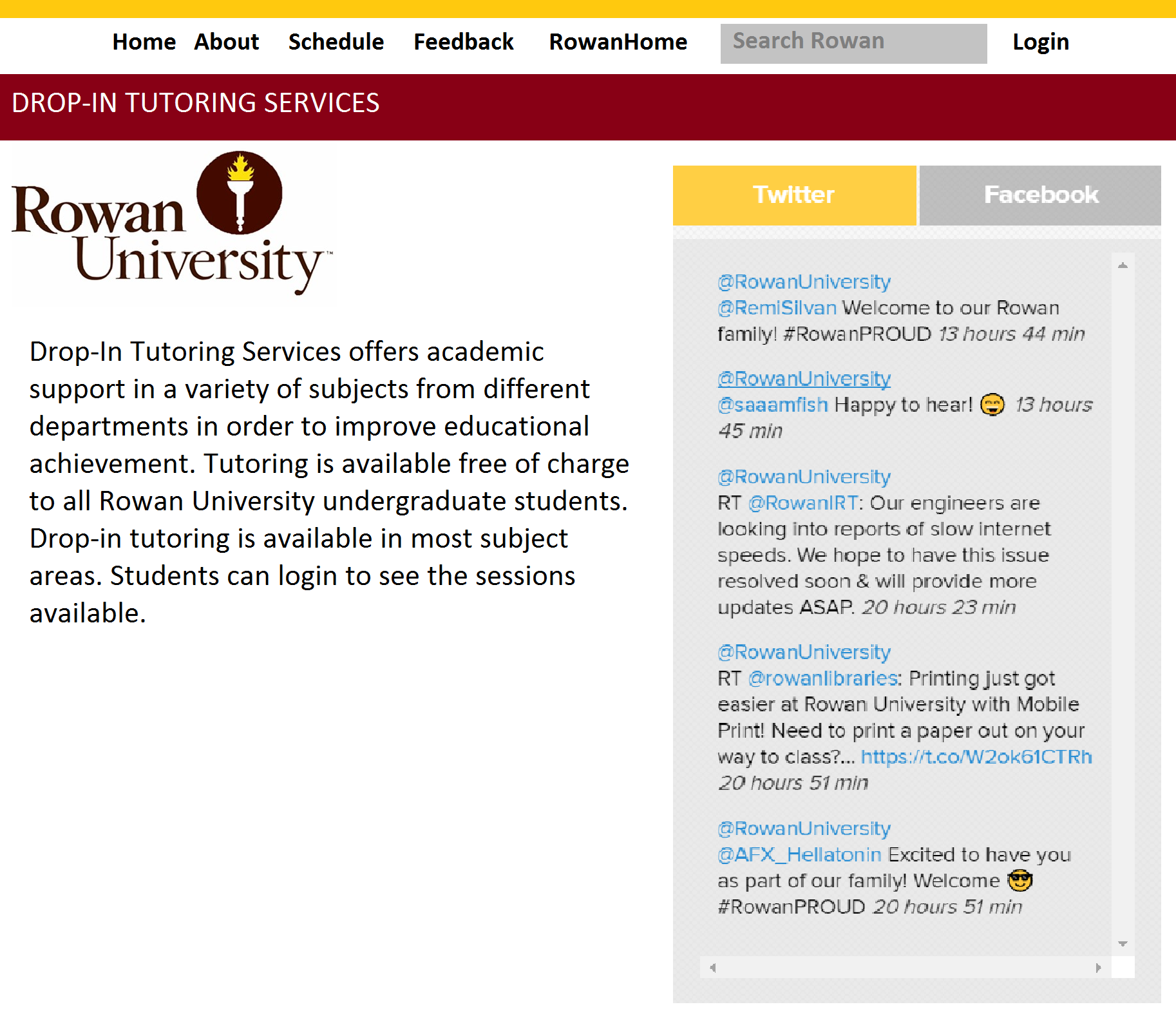
JavaScript

PHP

Java

Application Views

# Home



# Log In

# Drop-in Tutoring Schedule

# Survey/Feedback Form

# Navigation Bar

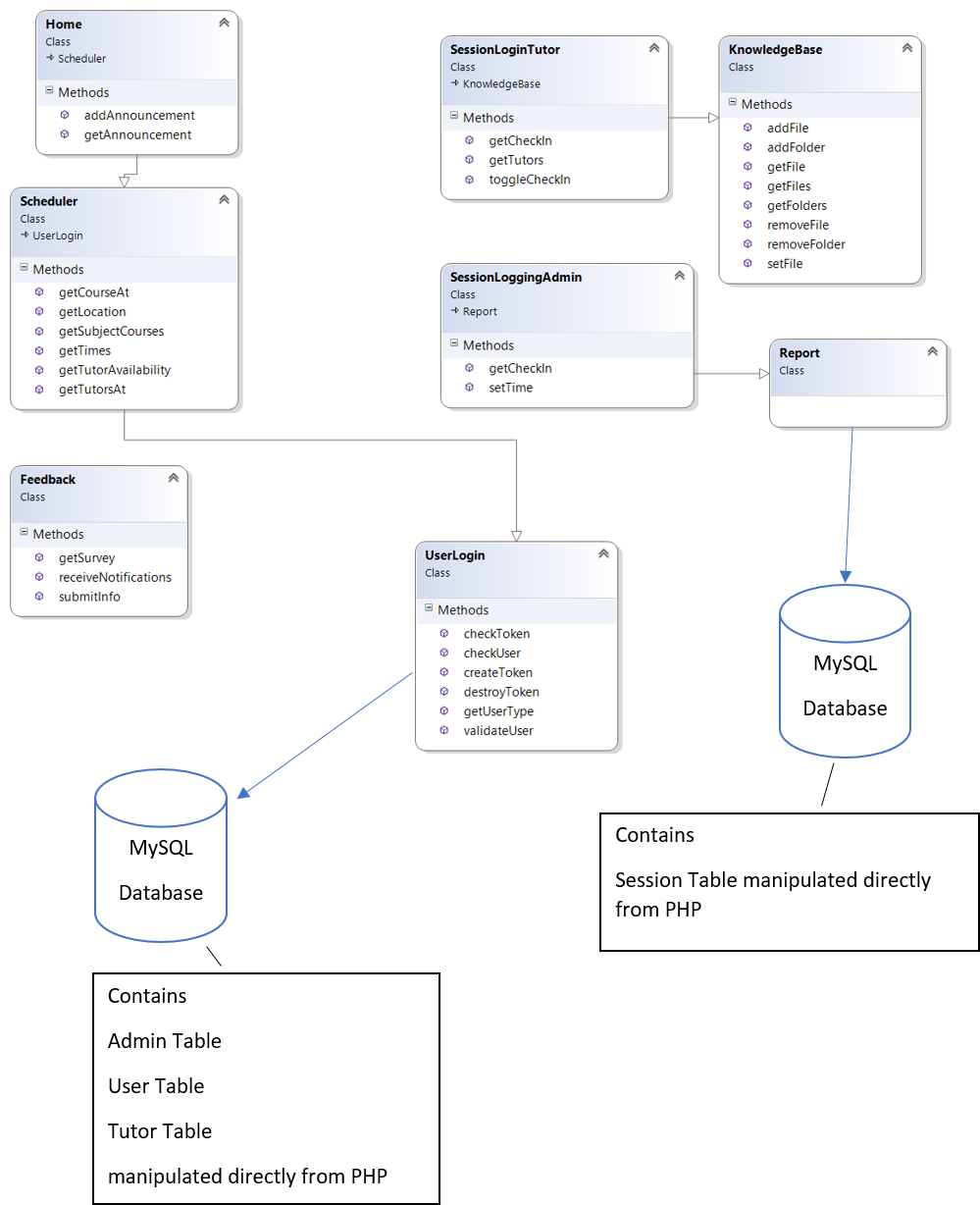
# Tutor Session Logbook

# Shared Knowledgebase

# Session Report Page

RESTful API

# Class Diagram



All endpoints in the RESTful service. For each endpoint, provide all inputs and outputs. Reflect briefly (in pseudo code) on how you will implement the endpoint. As an example, consider the “cities in a country” endpoint in my sample code. This might be written up as follows:  
  
Endpoint: POST /cities/:country  
Purpose: Returns cities in a given country  
Inputs:  
        country name: path parameter  
Output: JSON array of  
        city-name: String  
        population: integer  
Implementation:  
        Join the city and country tables listed in the database section and retrieve desired information

Database Design



Implementation Plan

Write am implementation plan for the first two weeks in the implementation phase. This must include all the action items for this time period and an assignment of responsibilities for individual team members. Note that I will use your writeup to assess how you have progress at the midway point in the implementation phase.

Features: Log-in, API implementation

Views: Home, Log-in, Navigation bar

Database instantiation