

1 Statement of Purpose

This document states the architecture on which the scheduling system is based. Additionally, the included class diagram shows the basic structures necessary to the scheduling system and how they interact with each other. The entities in the database are related through the included Entity-Relationship diagram. All of these aspects combine to form a system that effectively schedules the tutors for a given course and meets the previously determined requirements.

2 Architecture

The architecture most appropriate for the system is N -tier. The system's operation is very clearly separated into a tiered structure. First, the course and tutor information must be set up (2.1, 2.2). Then, the tutors must complete the surveys (2.5, 3.2, 2.6). Lastly, the system needs to process the survey responses and output a complete schedule (2.4, 3.4). There is a logical progression of ideas that must be followed for the program to function correctly. This logical progression is most effectively handled by an N -tier architecture.

3 Diagrams

3.1 UML Diagram

The UML diagram, Figure 1, displays how the different segments of the program interact with each other. The Setup object is responsible for initializing the course and its respective tutors. That action includes sending the survey links to the tutors. the Survey object is more of a theoretical object. It exists as an expression of the period of time during which the surveys are being filled out and submitted. The Sched_Writer object takes the received surveys and processes the results, which results in an appropriate tutoring schedule.

3.2 ER Diagram

The ER diagram, Figure 2, serves to visualize the structure of the database back-end. The diagram features how the different tables are related and lists the attributes of each table. Every Course is managed by an Admin. Every Course has 1 to n Tutor objects that are later entered into the schedule.

4 Language for Implementation

For the front-end user interface, it has been decided that HTML, CSS, PHP, and JavaScript will be used, due to the developer's experience and familiarity. Consideration was given to writing the project in AngularJS, but the learning period would have pushed the time-frame for the project's completion beyond the range of the developer's availability. For the database back-end responsibilities, a MySQL database will be used. To create the schedule from the raw data, a Python program will be used. This program will take in files of raw data and output a finalized schedule.

5 Use Cases

5.1. Using the system to set up a course.

5.1.1. The user clicks the link to set up a new course.

5.1.2. The user enters the number of tutors to be assigned to the course.

5.1.3. The user clicks "Submit."

5.1.4. The user enters the course title.

5.1.5. The user enters the necessary personal information for each tutor.

5.1.6. The user clicks "Submit Info."

5.1.7. The user enters the course's tutoring hours and verifies they are correct.

5.1.8. The user clicks "Send Survey."

5.2. Using the system to submit a survey.

5.2.1. The user clicks the survey link in the email he received.

5.2.2. The user enters his personal identifier.

5.2.3. The user selects the tutoring hours he would prefer to tutor.

5.2.4. The user selects the tutoring hours during which he has class.

5.2.5. The user clicks "Finish."

List of Diagrams

1	Tutor Scheduling UML.	4
2	Database ER Diagram.	5

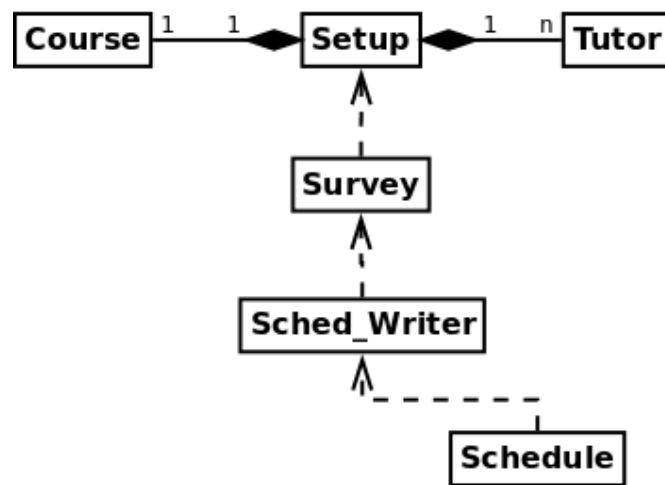


Figure 1: Tutor Scheduling UML.

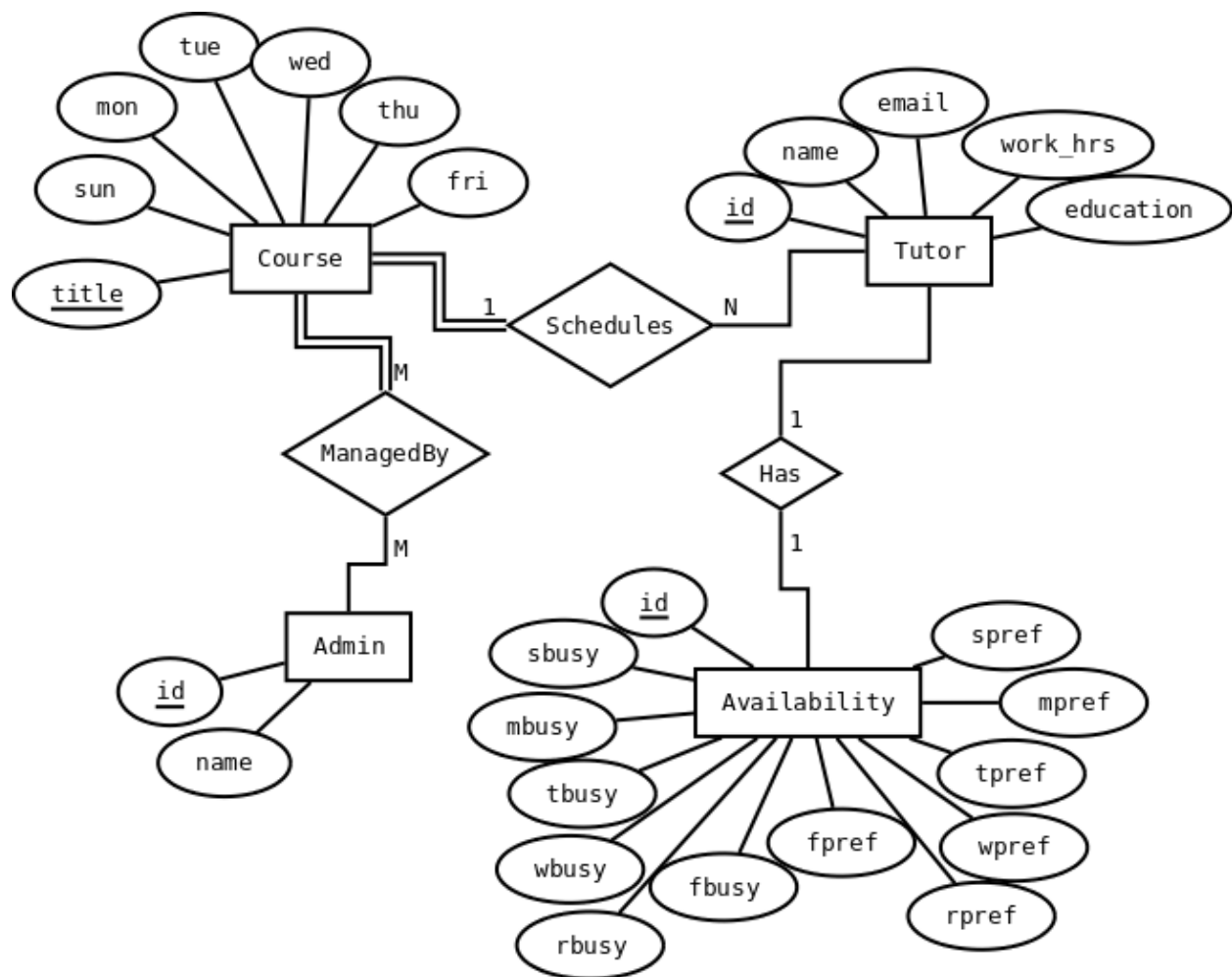


Figure 2: Database ER Diagram.