Optimised Course Schedule Generator

By:
Doyle D. Bigelow
Diego R. Draguicevich
Brett M. Stricker

Introduction

- What is the project?
- Who is it for?
- What does the project aim to accomplish?

What is the project?

- Optimized Course Schedule Generator (OCSG)
- Tool that will assist colleges and universities with generating their master schedule.
- Will be able to assist with the organization of large pools of data to create the schedule

Who is it for?

- Designed to be used by college and university faculty
- Useful for students and teachers who want to view course and schedule information

What does the project aim to accomplish?

- Aims to create a useful software system that can assist faculty administrators with their jobs
- Create a master schedule that meets the specified requirements.

System Requirements

Requirements

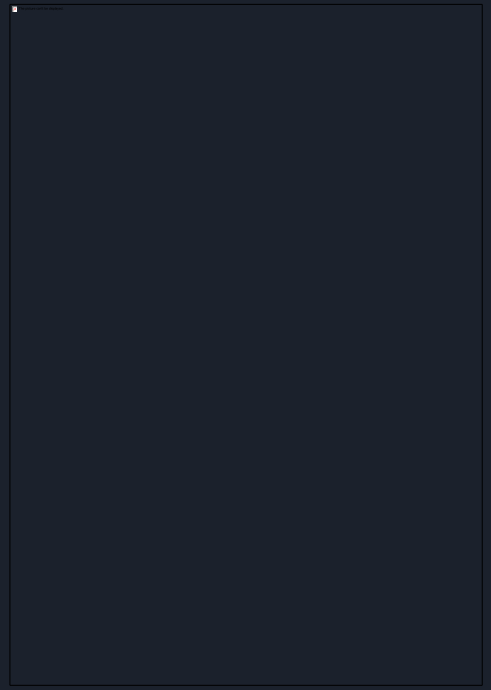
- Client-Server structure
- Use cases
  - View
  - Add
  - Remove
  - Modify
  - Generate master schedule
  - Login
- Authentication management
  - Permission levels

Client-Server Structure

- Server for data storage, schedule generation
- Client to access information, control server

Use Case: View

- Users need access to data
- Allows search of database for specific information
- Open to all authenticated users, but specific data is restricted to specific authorization levels

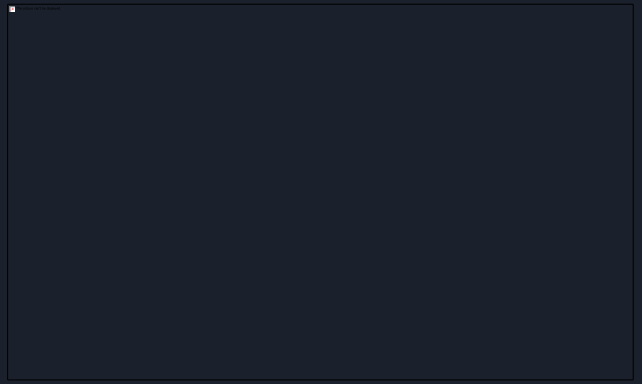


Use Case: Add, Remove, Modify

- Allows for additions, deletions, and modification of data stored in the different database tables
- Limited to Administrator

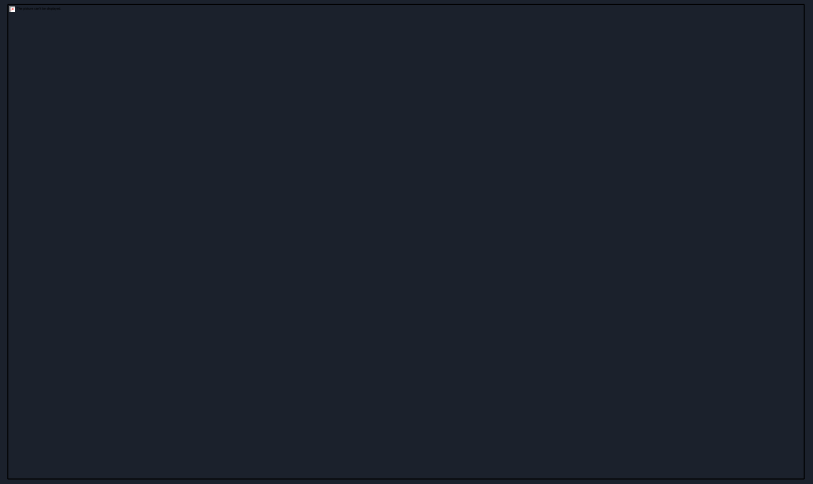
Use Case: Generate Master Schedule

- Primary purpose of system is to create course sections, that together serve as the Master Schedule
- Requires the tables (besides courses) to have sufficient data in them
- Output saved to database as the courses table
- Limited to administrator



Use Case: Login

- Prerequisite of all other use cases
- Established user permission level



Authentication Management

- Permission levels
  - Student
  - Teacher
  - Administrator

System Design

Proposed Architecture

- Client-Server Architecture
- 5 Subsystems

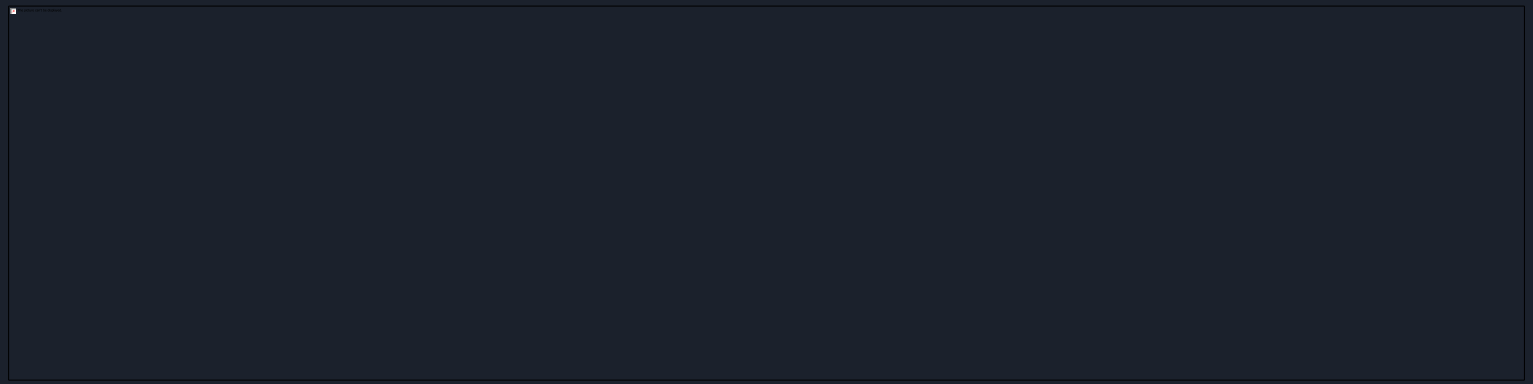
Subsystems

- UI
- User Management
- Schedule Generation
- Authentication
- Database Management

Class Design

- 13 total classes
- 4 major class diagrams
  - Database Connection
  - Load Data
  - Generation
  - Save Data
- The default class design is multiton
- Any class name that is followed by a (s) is a singleton

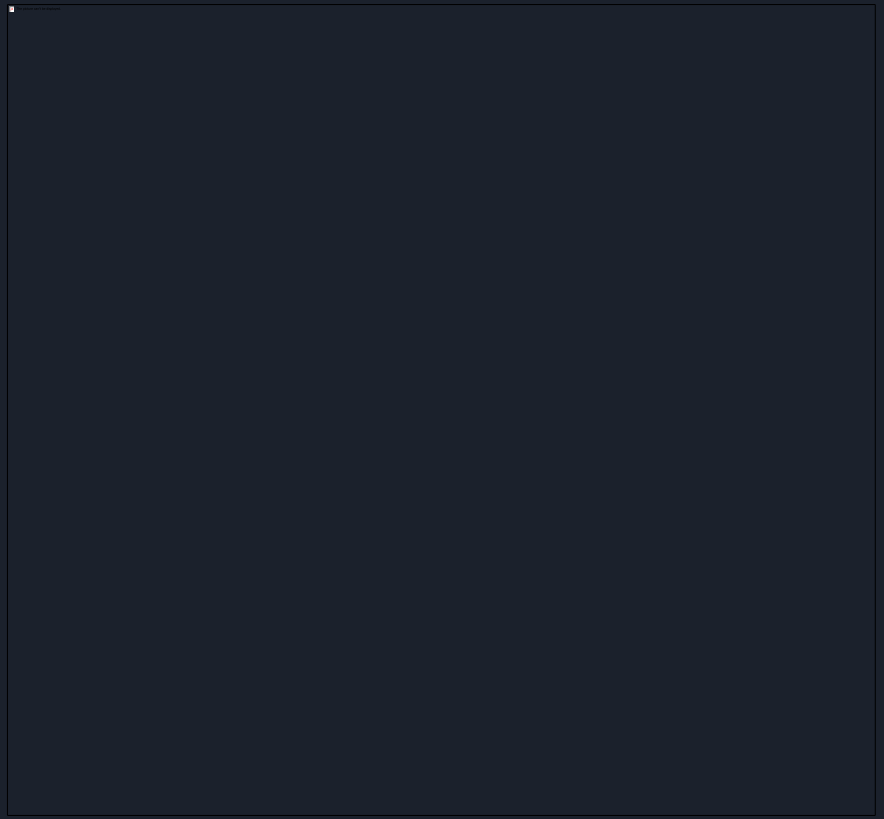
Database Connection Class Diagram



Load Data Class Diagram

Generate Class Diagram

Save Data Class Diagram

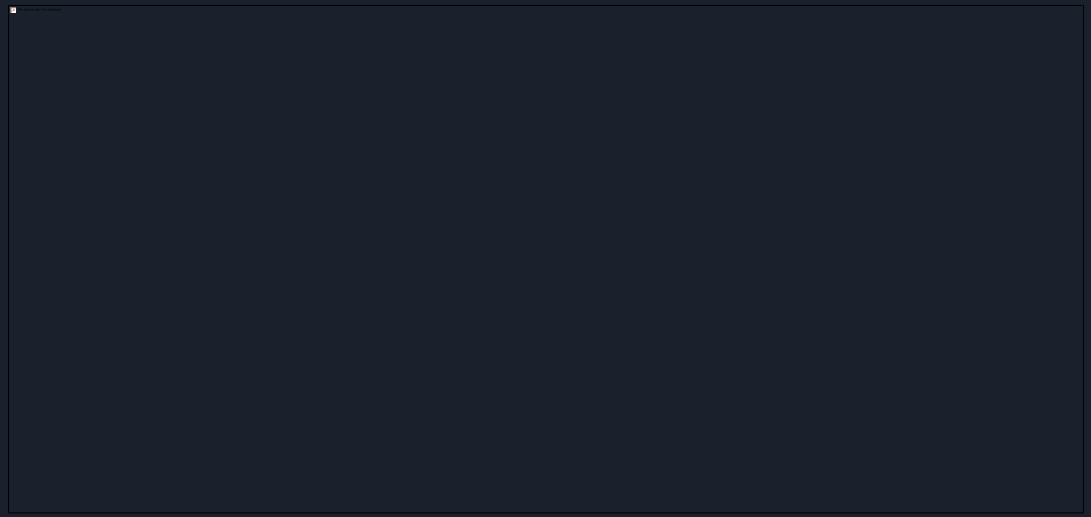


Object Interaction

- 5 Sequence Diagrams
  - Authentication
  - View Data
  - User Management
  - Data Management
  - Generate
- Each correspond to a use case

Authentication Sequence Diagram

View Data Sequence Diagram



User Management Sequence Diagram

Data Management Sequence Diagram

Generate Sequence Diagram

Prototype

- The following slides are screenshots from a prototype of the client side application for the OCSG.
- The prototype is setup under the assumption someone with administrator privileges has logged in.