

Software Design Document

CSC 4430

Prepared by the OfCourse! Development Team

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
| OfCourse! Development Team | 10/29 | Initial Document/Draft | 1.1 |

Table of Contents

[**Introduction**](#_t6i6saa66jk)[**4**](#_t6i6saa66jk)

[Purpose](#_9rx97kk3ztw) [4](#_9rx97kk3ztw)

[Intended Audience](#_mhpxwq9wt5vc) [4](#_mhpxwq9wt5vc)

[Scope](#_k47b2kotrhsn) [4](#_k47b2kotrhsn)

[Design Summary](#_m7w0x2fap1it) [4](#_m7w0x2fap1it)

[**Conceptual System Architecture**](#_o5s9q2pggsju)[**5**](#_o5s9q2pggsju)

[Overview](#_uzj2i2jrwtbz) [5](#_uzj2i2jrwtbz)

[Weekly Schedule Table](#_u9ie3n4cdgz6) [5](#_u9ie3n4cdgz6)

[Daily Schedule Table](#_gz8qtwjdeg9) [5](#_gz8qtwjdeg9)

[Enrollment Tool](#_mb4t719nrts1) [5](#_mb4t719nrts1)

[**Technical System Architecture**](#_9qe5wswvttyb)[**7**](#_9qe5wswvttyb)

[Overview](#_1bbtadtv2uwm) [7](#_1bbtadtv2uwm)

[Weekly Schedule Table](#_qgw53sxpaa0s) [7](#_qgw53sxpaa0s)

[Daily Schedule Table](#_aogbqv3mc42) [7](#_aogbqv3mc42)

[Enrollment Tool](#_tpuylt24x9ig) [7](#_tpuylt24x9ig)

[**Application Views**](#_3nidrls04btf)[**9**](#_3nidrls04btf)

[Logical View](#_ti7jo7lgbvmj) [9](#_ti7jo7lgbvmj)

[Physical View](#_j54dh1yaf9vu) [10](#_j54dh1yaf9vu)

[Development View](#_tjp9oouv2j73) [11](#_tjp9oouv2j73)

[Work-Assignment View](#_ciqazbpno4bm) [11](#_ciqazbpno4bm)

[**User Interface Design**](#_qc2m1kstwwvo)[**13**](#_qc2m1kstwwvo)

[User Interface Description](#_dem86xt9j3km) [13](#_dem86xt9j3km)

[User Interface Diagram](#_tpvtsspo7n1d) [14](#_tpvtsspo7n1d)

[**Database Design**](#_jitxodvuvvqm)[**15**](#_jitxodvuvvqm)

[Database Description](#_41a2qm1k43jc) [15](#_41a2qm1k43jc)

[Database Diagram](#_kxvifj86qmlf) [15](#_kxvifj86qmlf)

[**Element Catalog**](#_arb1o1swxmiy)[**16**](#_arb1o1swxmiy)

# Introduction

## Purpose

The purpose of this document is to describe the internal and external design of *OfCourse!*, including but not limited to: the conceptual system architecture, the technical system architecture, application views, and other specifications. Throughout this document, features that were outlined in the Software Requirements Specification document will be clarified.

## Intended Audience

This document is intended to be used as an extension of the Software Requirement Specification document to clarify the design, implementation, and features of *OfCourse!*. The audience includes but is not being limited to: financial (such as investors, marketing staff, consultants, etc.), development (programmers, graphic artists, etc.), and stakeholders. Certain sections of this document are more applicable to certain audience members; for example, developers should take note of the Technical Architecture section, while someone interested in marketing would find relevant information in the Conceptual Architecture section.

## Scope

This Software Design Document is intended to give a more detailed overview of the application to those who will be involved with the project such as developers or marketers. *OfCourse!* is an application intended to simplify scheduling courses for LSU students.

## Design Summary

Two tables serve can be seen upon opening the application, allowing a user to view their enrolled classes based on the week and the current day. The sidebar to the right of these tables is where the primary function of the application, enrollment, occurs. The user can select a course by department and level to add to their course schedule. The course schedule is then updated in real-time to reflect the changes.

To demonstrate these features, this document outlines the design from a conceptual and technical view to visualize the aforementioned tables and the methodology of the enrollment feature. As a web-based application, *OfCourse!* combines components of HTML (including CSS, AngularJS, ASP.net, etc.) with a client-server model for database purposes (storing and retrieving course information for users) to provide an efficient tool for querying requests.

# Conceptual System Architecture

## Overview

*OfCourse!* is a course-manager application that is designed to: allow the user to enroll in a course, and; view their courses in a real-time environment. It is designed to work in conjunction with LSU’s existing student web portal, **myLSU**. The application is designed to be accessible and functional for the purposes as outlined above. The user can enroll in courses by filtering them using two categories, Department and Level, or search for them using a manual search engine. After this, the user can select a course from the populated list, determine a time to enroll in the course, and then click the appropriate button to add the course to their schedule. Once added, the user’s schedule is updated to reflect the addition of the course in real-time through usage of two tables which show the weekly and daily schedule of courses.

## Weekly Schedule Table

The weekly schedule table displays the user’s enrolled courses. This table orders a user’s schedule based on the day in which the class is to be attended into their respective day (Monday thru Friday), and sorts this list based on the attendance time of the course (i.e., the start time). The courses to be placed into this table are derived from a database which stores the information for security and ease-of-access.

## Daily Schedule Table

The daily schedule table displays the user’s enrolled courses in a one-column, n-row table. This table orders a user’s schedule based on: that the course be scheduled for that day, i.e. that the course is on the day in which the application is accessed, and; the start time of the course. The courses to be placed into this table are derived from a database which stores the information for security and ease-of-access.

## Enrollment Tool

The enrollment feature of *OfCourse!* provides users with the ability to search for and add courses to their existing schedule in real-time. This tool combines two components - the schedule booklet and “Schedule Request” feature - found on **myLSU**. The first subfeature of the enrollment tool is the search tools for locating courses. The user can search for classes depending on two qualities; Department and Level, in which Level denotes the course’s level (meaning 1000-level courses, 2000-level courses, etc.). This search can be performed in one of two ways; by selecting from a list of departments and levels based on a drop-down menu, or a manual search for a course if the user knows the specific information for the course.

The next subfeature occurs after the search, in which the area below the search is populated with courses depending upon the aforementioned search. The user can select from the list of populated courses, which will produce a pop-up box displaying various information about the selected course, including a full name, description, attendance day/time, etc. A button is also in the box allowing the customer to enroll in the course. If the customer is able to schedule the course (i.e., the course does not interfere with any existing time slots), the course is added to the user’s schedule in real-time and corresponding tables are updated to reflect the change.

# 

# 

# Technical System Architecture

## Overview

*OfCourse!* is designed to be a real-time course manager. As such, it is important that each function perform in a timely and efficient manner. The application utilizes a combination of client/server technologies with a graphical interface provided by a website to accomplish this goal.

The weekly and daily tables derive their information from a database that contains the user’s enrolled courses. However, each table respectively incorporates a different algorithm for displaying the information. The tables, which are designed in HTML, act as a visually decipherable means for displaying this information.

The enrollment tool, similar to the tables, derives its information from a database as well, as courses are populated based on department and can furthermore be sorted by level. The user can select a course from the list of courses, view information about it, and add it to their schedule.

## Weekly Schedule Table

This table, created using HTML, populates by placing the course into a “slot” in the table based on: The attendance day of the course (Monday thru Friday), and; the start time of the course, arranged in ascending order. Included in the display are: department; ID, and; start time. The courses to be placed into this table are derived from a database which stores the information for security and ease-of-access.

## Daily Schedule Table

This table, created using HTML, populates by placing the courses which matches the day in real-time, in ascending order of start time. Included in the display are: department; ID; instructor; location, and; start time; The courses to be placed into this table are derived from a database which stores the information for security and ease-of-access.

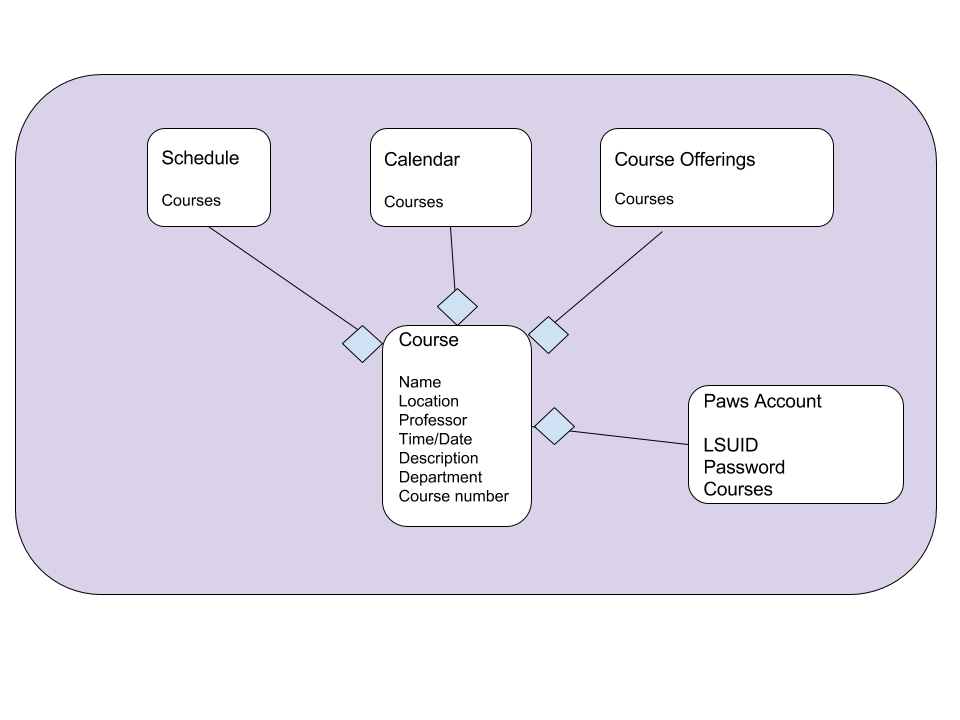
## Enrollment Tool

Once an user performs a search, a query is performed to locate the course based on the search, followed by those courses listed below the search section. The user can then select a course, which populates a pop-up box detailing information about the course (ID, department, description, etc.). Included in this pop-up box is a button that allows the user to enroll in the course.

If the time slot of the tentatively-enrolled course does not interfere with any existing time slots via usage of a query to check the start and end times of the currently enrolled times, then the user is enrolled into the course. A message below the list of courses will appear, with a prompt that reads a message along the lines of “Successfully enrolled in [Department] [ID]” or an error message informing the user that they could not be enrolled in the course.

# Application Views

## Logical View



## Physical View



## Development View



## Work-Assignment View

We have chosen to delegate tasks as follows:

* Project Managers: Kameron Smart and Jackie Robinson
* Web Development/Front-End Design: Nina Mohammed and Jackie Robinson
* Database Development: Kameron Smart and Phillip Breland
* Design/Documentation/Communication: Tyler Anderson and Che’La Fleming

Jackie and Kameron serve as co-project managers by coordinating decisions pertaining to the design of the database and user interface.

The Front-End Design and Web Development for *OfCourse!* is handled by Jackie and Nina. They have the most experience with HTML and CSS and are the most qualified to implement the web page design.

Kameron and Phillip have the strongest background in database development, so they were tasked with implementing the database that stores relevant course information, such as course title, instructor, and attendance day. The database implementation is imperative for the scheduling application, as this will allow the user store their own scheduling information and sync it with their calendars.

Che’La and Tyler are in charge of documenting group meetings and relaying information between group members. They also handle the design of the *OfCourse!* logo. They assist in the planning and coordination of tasks in meetings.

# 

# 

# 

# User Interface Design

## User Interface Description

It was important that we considered the fact that: first, **myLSU** is a web-based portal for students, meaning that it should be accessible over the Internet and not act as its own application.Secondly, many of the other features available under **myLSU** behave as extensions of the main portal, such as the Additional Services application for adding services to one’s fee bill. Therefore, we developed *OfCourse!* to be a web-application that could be accessed from virtually any location and device (see SRS Document for more details on Hardware and Software Specifications).

The website acts as the front-end implementation of our application. It is used to demonstrate the back-end’s capabilities as it: keeps track of an user’s courses; displays the courses in two tables, and; maintains a list of courses in which the user can enroll in. The two tables include: a weekly schedule, which appears on the upper-left hand side of the screen, detailing the enrolled courses over the span of a week, and; a daily schedule which appears on the lower-left hand side of the screen and details information about courses based on the day the application was accessed.

On the right-hand side of the screen is a sidebar, aligned appropriately to the right of the aforementioned tables, which serves as housing for the enrollment feature. A section for searching is left-adjoined to the sidebar, providing users with a way to search for a course using either: the two drop-down menus or a manual search in text-form. Both allow the user to search by department and ID. Below the search section is a list of populated courses based on the search. When selected, a pop-up box appears that details information about the course alongside a button for enrolling in the course. Finally, a section below this list serves as the message-system for the user on the status of their enrollment depending upon their success or failure in enrolling in the course.

The website is designed to be visually appealing but professional, in particular comply with LSU’s color requirements (as the purple and gold the University uses for much of its apparel is of a particular RGB). The website implements industry-standard coding practices, such as HTML, CSS, and AngularJS to perform tasks such as button selection, drop-down menus, and tables.

## User Interface Diagram



# Database Design

## Database Description

We noted that the central feature of the application, enrollment, would need a proper means for maintaining and accessing courses, which could be performed via usage of a database. Though it would had been possible to save this information in the client via some data structure, we concluded that: it could be easily compromised, leading to malicious attacks and other deficiencies, and; for the sake of the design-choices, a database would ensure ease-of-access and reduce redundancy.

The database encapsulates a many-to-many relationship between students and courses; many students can enroll in many classes and vice-versa; many classes can have many students enrolled in it.

Because our application exists off the premises that it would be used as an extension of the **myLSU** web portal, our schema for the database encompassed the *Courses* table, as the Student schema is assumed to exist as the user is logged into their account. This table would have the the following attributes:

## Database Diagram



# Element Catalog

|  |  |
| --- | --- |
| Database, Diagram, Data ... | This object represents a database in any usage. |
|  | This object represents an entity in any usage. |
| Open ... | This object represents a server in any usage. |
| Computer, Desktop, Workstation ... | This object represents a device (in particular, a device used to connect to something) in any usage. |
| Folder, Explorer, Files ... | This device represents a directory in any usage. |
|  | This object represents a relationship between entities in any usage. |