Andy Morrison
Ben Ratner

CCEL Project STEP-UP organization web app

Software design document

1. Table of contents

2. Introduction

2.1 Purpose

This software design document outlines the structure of the CCEL Project STEP-UP organization web app. It is intended to be used for the developers of the project, as well as any developer maintaining or altering the project after its initial release. The document will summarize the modules of the project and their interactions, and relate them to the functional and non-functional requirements specified in the Software Requirements Specification[1].

2.2 **Scope**

The software is meant to serve as a comprehensive system for organizing the participants and tutoring session of the Center for Civic Engagement and Learning (CCEL) Project STEP-UP tutoring program. This system is meant to replace the current method of organizing the program, which consists of paper- and private email-based communication. The project will take the form of a website built on top of a database of Project STEP-UP information. The website will be augmented with JavaScript to allow it to have fast, intuitive functionality, and to be compatible with both desktop and mobile computers. The project will allow students to better take advantage of the session offered through Project STEP-UP, and will allow tutors to be in better contact with session location hosts so volunteer time can be utilized as efficiently as possible.

2.3 **Overview**

This document will give an overview of the architecture of this software project. It will list all planned modules with a brief description of each, as well as a schema specification and brief description of the database to serve as the core of the system. A description of the planned user interface layout will also be given. Finally, a requirements matrix will be given, showing how each of the planned modules corresponds to the requirements listed in the SRS document.

2.4 **References**

[1] CCEL Project STEP-UP organization web app: Software Requirements Specification; Andy Morrison and Ben Ratner

3. **System Overview**

The CCEL Project STEP-UP web app is meant to eliminate reliance on private email communication and paper-based scheduling in Project STEP-UP. The core of a project is a relational database of students, tutors, sessions, and session locations. This database is augmented with additional tables representing interactions between users, such as a table of messages and a table indicating which students plan to attend which sessions. PHP modules are used to retrieve information from the database. Then JavaScript modules are used to dynamically insert information into HTML web pages.

The website will allow users to register with a unique username. There will be different roles for different users, depending on what role is performed within Project STEP-UP by the owner of a given username. The primary function of the website is to provide a searchable list of tutoring sessions, which are created by tutors and project STEP-UP officials. Furthermore, students will be able to send messages to Project STEP-UP tutors and officials, and tutors and officials will be able to respond and send messages to each other.

4. System architecture

4.1 Architectural design

The architecture of the system is depicted in Figure 1 below. Each horizontal “layer” of the diagram roughly represents a subsystem, each containing several modules. When a user accesses the website, a Layout Manager module determines the screen size of the user’s device and returns an appropriate HTML home page. The layer below the Layout Manager contains all HTML files of the system. Access to HTML files is always done through a JavaScript module titled “home.js.” This module serves as an interface between the HTML files and any PHP files. In this way, JavaScript Ajax functions within home.js can dynamically update HTML files without any need for full page refreshes.

In the next layer is user.php. This PHP file serves as the interface through which home.js accesses and modifies the database. If the user is not logged into the website, user.php provides all information retrieving services available to an unregistered user, the most important of which is the ability to see the list of scheduled tutoring sessions.

Upon logging in, user.php performs the role of verifying the user’s login credentials and setting necessary cookies upon success. Once those cookies have been set, the user has access to the messenger module and its JavaScript interface, as well as one of the three “role modules” in the layer below user.php. At the bottom layer is the database itself.

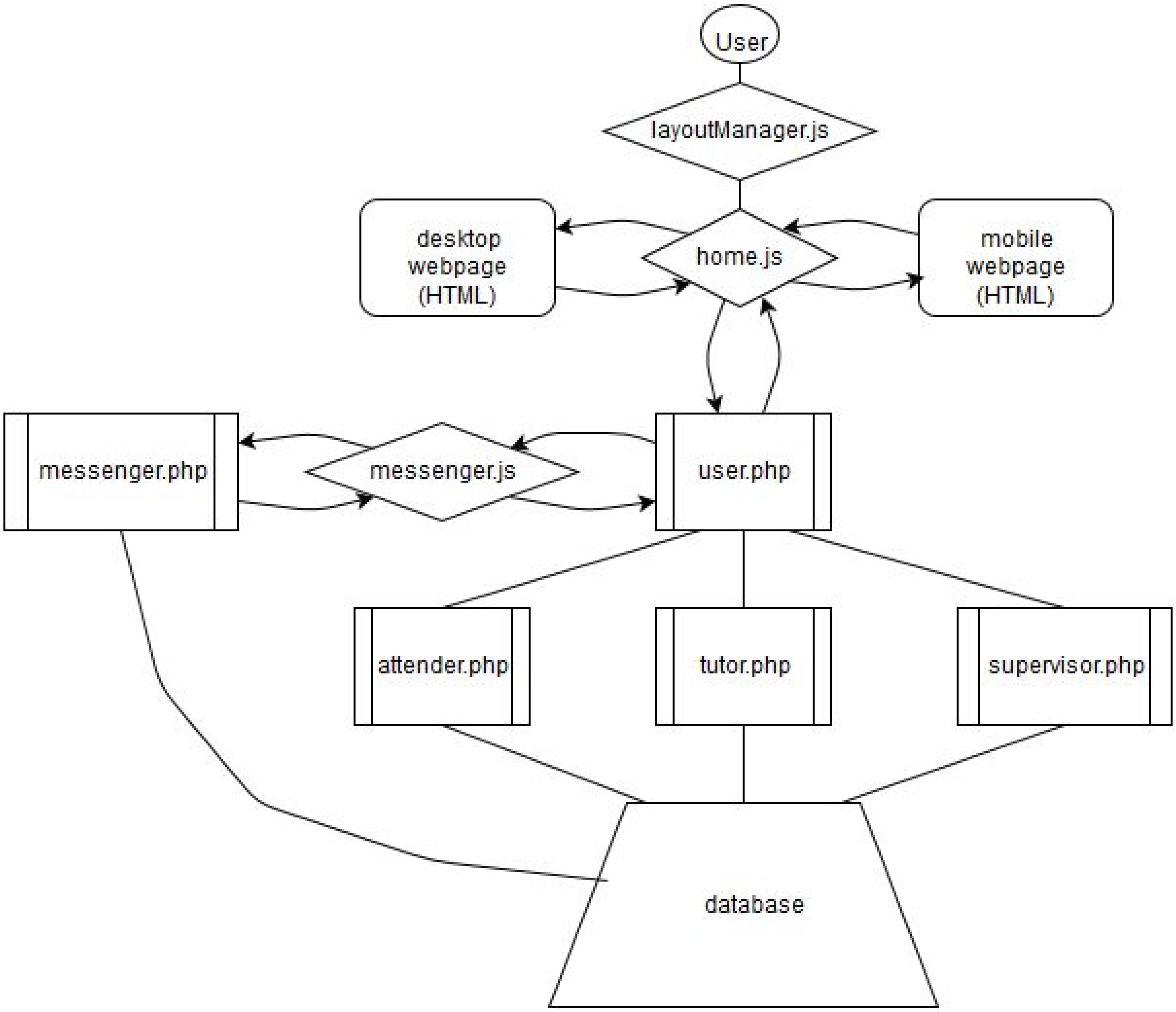

Figure 1: System architecture

4.2 **Design rationale**

The design of this system is based on the need for information organization within Project STEP-UP. At its core, the system is a response to the extreme utility of setting up a database for Project STEP-UP. That is why a simple relational database serves as the basis for the entire system. PHP is used to access the database due to PHP's object-oriented capabilities, which are applicable considering the various roles of participants in Project STEP-UP.

Heavy use of JavaScript is planned to ensure the website is fast, modern, and user-friendly, and to completely encapsulate and hide the usage of PHP and a database at the system's core. JavaScript will also ensure the website is versatile for all screen sizes. We chose to build a versatile website rather than building a desktop website accompanied by a mobile app to maintain simplicity upon initial deployment of the system. A self-adjusting website will be accessible to all Project STEP-UP participants.

4.3 List of modules

- user.php - Contains functions to set cookies when a user logs in; contains functions to retrieve user info, retrieve sessions, and interface with more specialized user role PHP files.
- attender.php - Contains functions for performing operations that can only be performed when a student user is logged in, as defined in the SRS.
- tutor.php - Contains functions for performing operations that can only be performed when a tutor user is logged in, as defined in the SRS.
- supervisor.php - Contains functions for performing operations that can only be performed when a CCEL supervisor user is logged in, as defined in the SRS.
- messenger.php - Functions for interacting with the Message table of the DB and generating HTML for the messaging system interface.
- messenger.js - Ajax functions for calling functions in Messenger.php.
- home.js - JavaScript for adjusting appearance of homepage based on user interactions (pagination, login/registration popup, show search results div and append search results), as well as Ajax functions for logging in and retrieving sessions info from DB through user.php.
- layoutManager.js - Chooses between desktop layout and mobile layout when user navigates to the site.

5. Data design

5.1 Database schema

ChangeLog (date, time, type, username, description)
Message (senderID, recipientID, subject, content)
User (username, password, firstName, lastName, email, organization, role)
Site (name, address)
Session (sessionID, site, date, time, subject, gradeLevel, tutorUsername, tutorName)
WillAttend (username, sessionID)

5.2 Database description

- The Site table will contain all locations at which Project STEP-UP tutoring sessions occur.
- The User table will contain all tutors, students, and CCEL officials registered in the system. The roles of users will be specified in the “role” column, and their unique usernames will serve as the primary key. The values in the “role” column will be restricted to the set {“attender”, “tutor”, “site leader”, “supervisor”, “host”}.
- The Session table, containing all tutoring sessions scheduled to occur on or after the current date, will have a primary key consisting of a unique session number (sessionID) and the name of a tutor teaching at the session. In this way, a session with multiple tutors attending can be represented in the database.
- Session attendees will be able to indicate their intention to attend a certain session. This intention will be stored in the WillAttend table.

- A simple messaging system will be implemented. Messages will be stored in and retrieved from the Message table.
- All changes to the database will be noted in the ChangeLog table. The values in the “type” column will be restricted to the set {“message”, “registration”, “siteAdd”, “willAttend”, “sessionCreate”, “sessionChange”, “sessionCancel”}.

6. **Human interface design**

6.1 **Overview**

The home page will consist of a centered, paginated, complete list of all scheduled tutoring sessions with a search bar at the top. The sessions will be displayed as a stack of wide rectangular divs containing all information about each respective session. There will be a header at the top of the page matching that of the rest of CCEL’s web pages. Above the searchable list of sessions will be a short welcome message, with a link to basic instructions on the purpose and usage of the system.

In the upper right will be a “Log in or Register” button. Upon being clicked, JavaScript will gray out the web page and display a div with fields to enter the user’s username and password, as well as a link to begin the registration process.

Upon logging in, the user will be returned to the home page described above, except with the addition of a navigation bar under the header. This navigation bar will include links to a list of sessions the user plans to attend, a link to the user’s personal information, and a link to display a div containing the messaging system. Furthermore, once a user is logged in, the name of any tutor appearing within the context of a session’s information will then be a clickable link. Clicking that link will automatically open the messaging interface with the address field populated with the clicked user’s username.

7. **Requirements matrix**

	user.php	attender.php	tutor.php	supervisor.php	home.js	messenger.js	messenger.php	layoutManager.js
FR-01	■							
FR-02					■			
FR-03	■			■				
FR-04	■							
FR-05			■	■				
FR-06		■						
FR-07							■	

FR-08							■	
FR-09							■	
NR-01					■			
NR-02	░							
NR-03	■	■	■	■				
NR-04	■							
NR-05								

8. **Document change history**