Design

Version 3.0 | March 13, 2014 | CS 486

The Aviators

Charles Chavez

Dillon Postage

Mark Malone

Contents

Introduction

In this document we are going to address our sponsor's problem, provide a solution, and describe our implementation plan. Our sponsor's goal is that they want to create a business by teaching students to operate aircraft. The current problem is that our sponsor has no way to tell students if they are doing well or not. Our solution to this problem is the SOAR web application.

The SOAR web application interfaces with the flight simulation software called Prepar3d. The user takes a course in Prepar3d and is given a progress file. The SOAR web application takes this progress file and turns it into easy-to-understand charts, tables, and graphs. This information allows students to understand how they are doing. Additionally, to facilitate student improvement, we will provide resources to those who are doing poorly. Conversely, the system will encourage users who are doing well to continue their training.

The web application will provide a dashboard to students so they may navigate through the SOAR system. The dashboard will implement components based on the current user's role in the system. We will discuss the components within the dashboard in our architectural overview and our component descriptions.

Finally, our system's implementation plan is derived from our system architecture. We will first implement the database that will manage the system's users, institutions, and course. Then we will move into developing the front-end of our system, which contains all the components that directly interact with the user. Last, we will develop our back-end scripts that process the information from the front-end components.

System Assumption

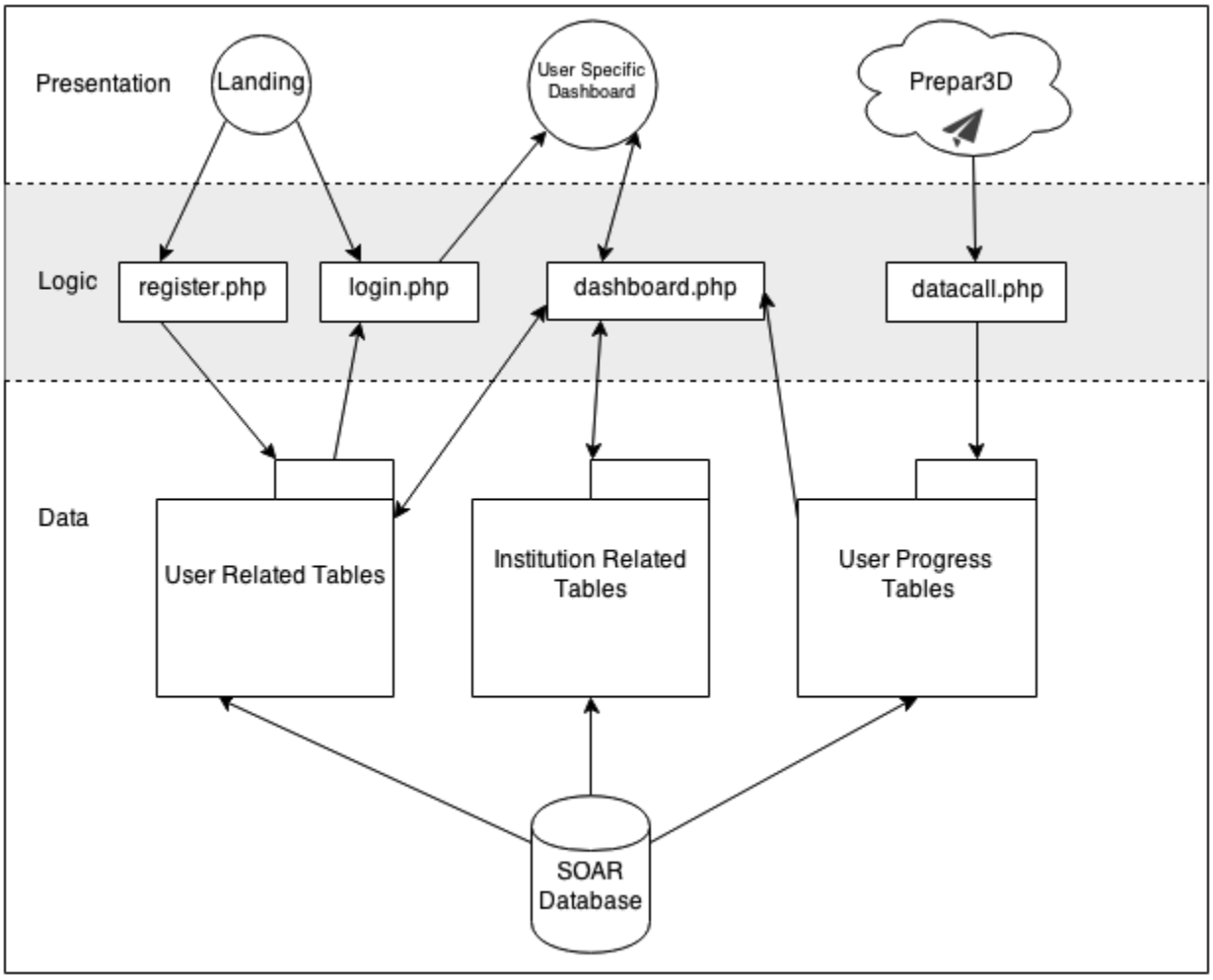
Because there is a lack of communication between the project sponsor and our team, we have had to assume some business rules of the system that are not related to software development.

Business Model

We have developed a business model for the SOAR system to specifically address how we will manage access to the application. As such, we are proposing a pricing model that targets institutions of the system, students of institutions, and unaffiliated system users. This model allows us to develop a system that is based on a one time activation rather than a continuous subscription model.

Targeted User	Price	Benefits
Institution	$100,000 for life	● Unlimited Institution Administrators ● Unlimited Teachers
Institution Students	$100 for life	● Access To System
Unaffiliated Students	$120 for life	● Access To System

Architectural Overview



We are implementing a three-tiered architecture into our system. The three tiers separate our components into presentation, logic, and data. This architecture allows all user levels to connect to the server through the same interface.

The presentation tier is comprised of three different components: the landing information, the SOAR dashboard, and the aviation training using Prepar3d. Landing information is what the users will see when they first visit the web site. The SOAR dashboard is used by many different user types and, based on the user level, provides capabilities that range from management of the system for higher privileged users to reviewing course progress for students.

The logic layer has a register method, a way to log into the system, a customized dashboard for users, and data processing to gather analytics about user course progress. The register method provides an easy way to gain access to the SOAR system through a signup page. The login method allows users a secure method to authenticate into the system. The dashboard controls what each user can do within the SOAR system. Finally, the data call logic is how the Prepar3D program sends information about progress of a course to the user progress tables.

The last layer is the data layer that saves information for users, institutions, and records information. The user related tables have information about all the users of the system, keeps track of their sessions, and what role they are. The institution related tables are a way to know which institutions have purchased licenses, which students are signed up for which institution through a roster, and what courses each student can take. Lastly, the user progress tables are where the Prepar3D program saves information, and uses it on the dashboard to display statistics to the students.

Module and Interface Descriptions

For our component description we are going to separate our analysis into the three parts of our architecture. We will first focus on the presentation tier components. We will then move into our logic tier components. Finally, we will finish with our data tier components. For each of our tiers we will outline how they interact with each other.

Presentation Tier

Our presentation tier encapsulates all the components that have immediate interaction with users. This tier is made up from the front-end of the web application which includes every component that outputs information to a user, and every component that takes information from a user. The presentation tier directly interacts with the logic tier through procedural calls that transfer data.

- **Landing**

  This component is the first thing users see when they navigate to the web application. The responsibility of this component is to provide a path for a user to go from a guest to an authenticated user, and to inform the visitor what the SOAR application is and how it works.
- **Institution Inquiry**

  The institution contact page is where schools can contact SOAR to purchase the institution plan. Since it is a higher cost, there will be financial representatives to help integrate institutions into using the SOAR system.

- **Register**

  This is the page where new users will go to begin their SOAR experience. It will be a simple form that collects the user's email, name, and password that they can use later to login, after verifying their identity.

- **Login**

  The login page is where the user puts their registered email and password in to gain access to the SOAR system, and subsequently the dashboard.
- **Dashboard**

  This module provides an authenticated user with options to navigate through the web application. The data displayed will vary based on the user's level. For administrators, who work

for SOAR, there will be options to add institutions. For Institutions (administrators) there will be options to add other institutions, and teachers that can manage their subset of students. Teachers can add courses which will manage students. The students will be able to view information based on how they did using the Prepar3D program. All users will be able to change simple options like what their name or password are.

- **Prepar3D**

  This module is here to display the link between the modeling functionality of the system and the web management functionality of the system. The responsibility of this module is to allow users to take lessons in aviation, and to send the data from lessons to the SOAR application.

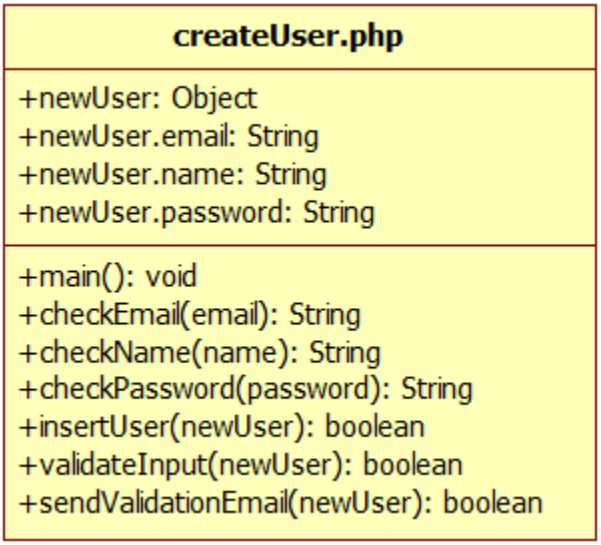
Logic Tier

The logic tier acts as the intermediary between the presentation tier and the data tier. Information gathered from the presentation tier is transformed in the logic tier into a query that is passed along to the data tier. The data tier provides results to the logic tier that is used to decide what the user will see.

- **register.php**

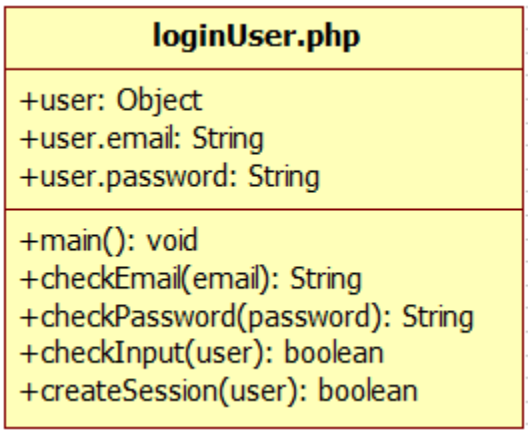
  Description This method allows students to sign up to the SOAR system. It will take information from the register presentation page and validate it through defined business rules.

  Interface This method will interface with the register presentation module and the user table.



- **login.php**

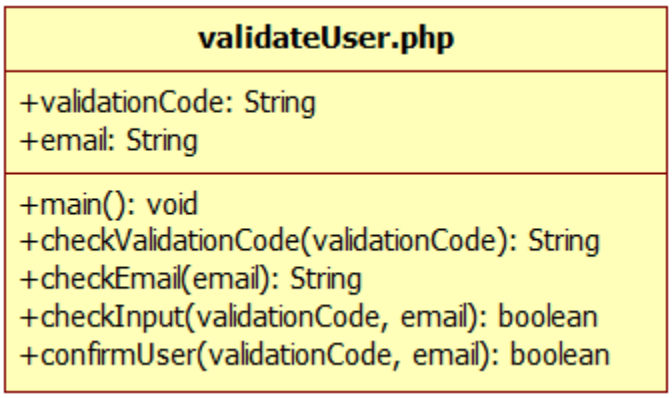
  Description This component takes information from the login presentation method and authenticates a user. It will then create a session if the user has provided the correct credentials. Finally, it will redirect the user to the dashboard presentation page.



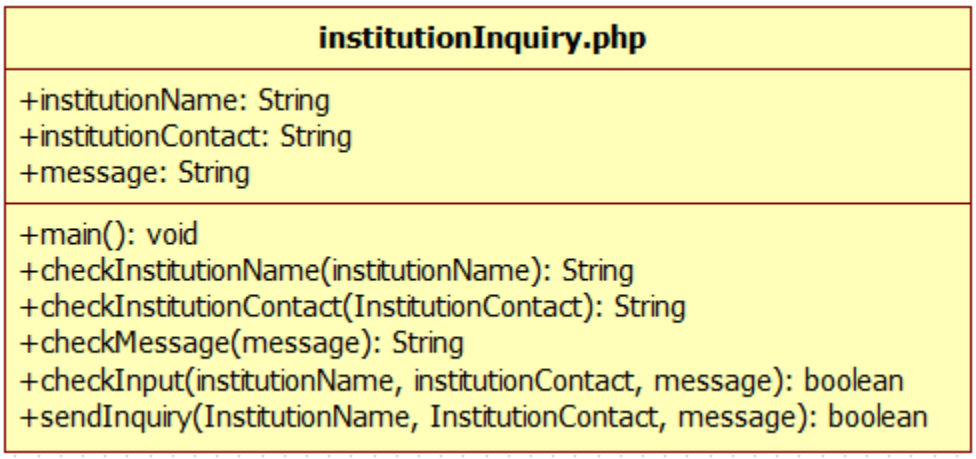
  Interface This method interfaces with the user database table, the login presentation module, and the dashboard module.

- **validateUser.php**

  Description This method validates users who have finished a successful registration. It will take in a validation code and a valid email address and validate a user. This means that a user without a valid email cannot get into the SOAR system.



  Interface This method interacts with the user table by removing the validation hold from a user.



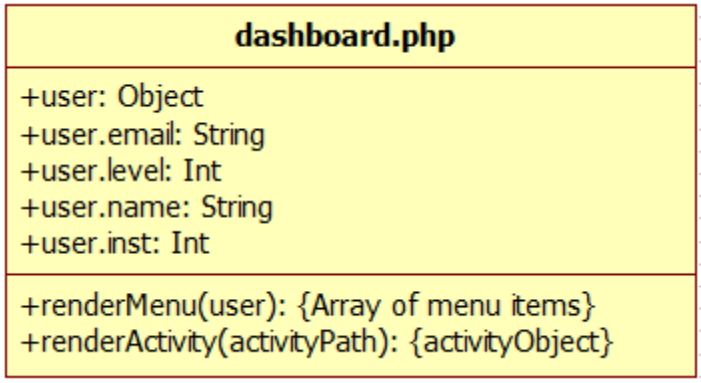
- **institutionInquiry.php**

  Description This method manages inquiries made by institutions to SOAR. It takes in the institution's name, a contact for the institution, and a message and creates an email that is sent to the SOAR sale's team.

  Interface This method interacts directly with the institution inquiry method in the presentation tier.

- **dashboard.php**

  Description This method manages the different activities users can perform. The dashboard will render a menu based on the user's level, and it will render the main body's content based on the provided path.



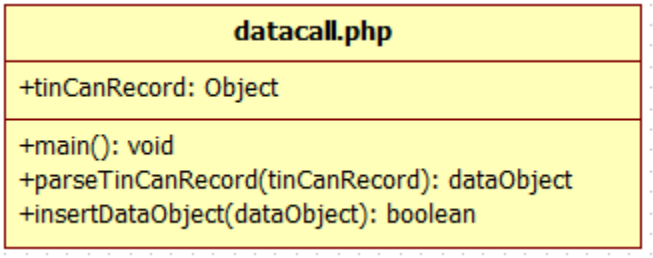
  Interface This method interfaces with the user tables, the institution tables, and the user progress tables. This method provides all the information to the dashboard page in the presentation tier.

- **User Activities**

Administrators	**Institution Administrators**	**Teachers**	**Students**
Manage Administrators -- An administrator can use this activity to add or remove other administrators of the SOAR system. The first user of the SOAR system is undeletable.	Manage Institution Administrators -- An institution administrator can add other institution administrators. The initial account cannot be deleted, but a SOAR administrator reserves the right to delete institution administrators.	Manage Students -- Here a teacher can manage students in bulk and see their progress through the courses defined. This is where a teacher would gather grade information for their institutions grading system.	Manage Courses -- Here students can add courses which will allow teachers to view their progress. They can drop from courses as well.
Manage Institutions -- Administrators can add or remove institutions from the SOAR system.	Manage Teachers -- Here institution administrators can add or remove teachers from their institution.	Manage Courses -- Teachers can add or remove courses here. The courses will automatically purge after 1 year after course begin date.	View Progress -- This is the main function of the SOAR program that allows students to see what they have accomplished and what they need improvement on.
Emulate -- The emulate function follows a hierarchy rule set of user levels and temporarily replaces the user ID of the user logged to be that of the user they want to emulate.	Emulate -- Same as administrator emulate, but cannot emulate administrators.	Emulate -- Same as institution administrators emulate, but cannot emulate institution administrators.	// -- No emulate function for students.

- **datacall.php**

  Description This method takes in a HTTP request from the Lockheed Martin Prepar3D software and parses the data into a data object that can be inserted into the user progress tables.



  Interface This method interfaces with the Prepar3D software and the user progress tables.

Data Tier

The data tier is the lowest tier in our system, and it stores state information for our system. The data tier allows the logic tier to make decisions by adding information to the request of the user. There is no direct interaction between the data tier and the presentation tier. All of the following tables are within the "SOAR" database.

- **Users Table**

  Description: This table stores information for all the users of the SOAR system. This is major link between all other information in the database. The information saved is the email, name, and password.

  Interface: The users table is written to when a user signs up on the register page. Depending on if the user signed up on the site, or if they are given an account by an administrator, they will be assigned a specific user level. It is used by the login page to validate a user. Record progress information is shown on the dashboard based on the user. When a user selects an institution, the table will update with the referenced key from the institution table.

- **Users_Level_LK Table**

  Description: This table defines what level a user is identified as. There are four levels: administrators, institution administrators, teachers, and students. The table will be written before the system is launched.

Interface: An account sign up will default for a student user level, but an account made by an administrator is determined by what the administrator is creating. When a user goes to the dashboard, their menu options are going to be determined from the user level.

- **Session Table**

  Description This table keeps track of user session data.

  Interface If the user is not active on the web application after 30 minutes, then the table is written to and the user is signed out of their current session. They will need to log in again to use the system.

- **Institutions Table**

  Description When an institution pays their fee by contacting a SOAR associate, an administrator adds them through the administrator dashboard and the table is written to.

  Interface When a new user signs up, they will use their institution appointed email. If there is no match to the domain part of the institutions table, then the user is assigned the default SOAR institution. When a course is made, the institution is referenced by this table.

- **Courses Table**

  Description Teachers add courses through their dashboard. They define how long the course will go for, but cannot select an end date longer than a year. After a student has registered and logged into their dashboard, they have the option to manage their courses. Here they can enter a course code provided by the teacher of the institution and allow that teacher to track their progress.

Interface The course's name is defined by what the Prepar3D program identifies the course as in the data call. The course is also listed with a user's ID to identify the roster of students enrolled in the course.

- **Roster Table**

  Description This table keeps track of the students enrolled in a particular course.

  Interface When a student enrolls in a course, the roster table is written to with that student's ID and the course they are signing up for.

- **Data Table**

  Description The data table stores the call information from Prepar3D. There is a key for referencing the call, a blob type variable of the information sent, and a date that the call was made.

  Interface Parts of the blob data are saved to the records table to save user and course information.

- **Records Table**

  Description The records table grabs the course name from the data table blob information along with the user information from the blob and immediately writes it to the records table. This saves progress information for specific courses.

  Interface The charts and graphs are going to be based on the count of record rows. A user will have a specific amount of record rows that will correlate with their progress in the course they are signed up for.

- **Menu_Links Table**

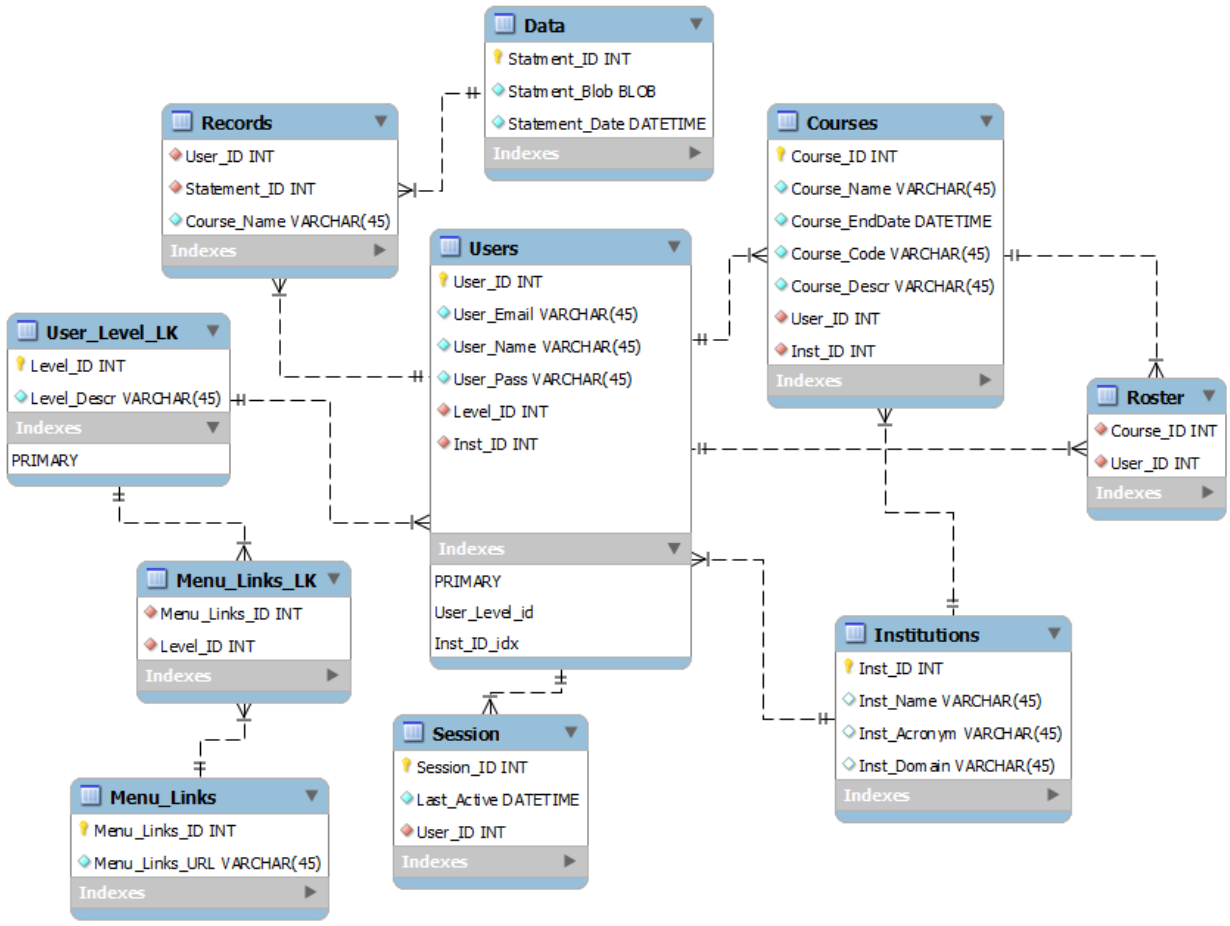
  Description This table remembers what links each user level type is allowed to access on the dashboard.

  Interface The table interfaces with the menu links lookup table that defines what pages the menu links will point to.

- **Menu_Links_LK Table**

  Description This table is a lookup table to outline what links are available.

  Interface This table is used by menu links to find out the overall link information in the menu of each user's dashboard.


Data
Statment_ID INT
Statment_Blob BLOB
Statement_Date DATETIME
Indexes
Records
User_ID INT
Statement_ID INT
Course_Name VARCHAR(45)
Indexes
Courses
Course_ID INT
Course_Name VARCHAR(45)
Course_EndDate DATETIME
Course_Code VARCHAR(45)
Course_Descr VARCHAR(45)
User_ID INT
Inst_ID INT
Indexes
Users
User_ID INT
User_Email VARCHAR(45)
User_Name VARCHAR(45)
User_Pass VARCHAR(45)
Level_ID INT
Inst_ID INT
Indexes
PRIMARY
User_Level_id
Inst_ID_idx
User_Level_LK
Level_ID INT
Level_Descr VARCHAR(45)
Indexes
PRIMARY
Roster
Course_ID INT
User_ID INT
Indexes
Menu_Links_LK
Menu_Links_ID INT
Level_ID INT
Indexes
Institutions
Inst_ID INT
Inst_Name VARCHAR(45)
Inst_Acronym VARCHAR(45)
Inst_Domain VARCHAR(45)
Indexes
Session
Session_ID INT
Last_Active DATETIME
User_ID INT
Indexes
Menu_Links
Menu_Links_ID INT
Menu_Links_URL VARCHAR(45)
Indexes

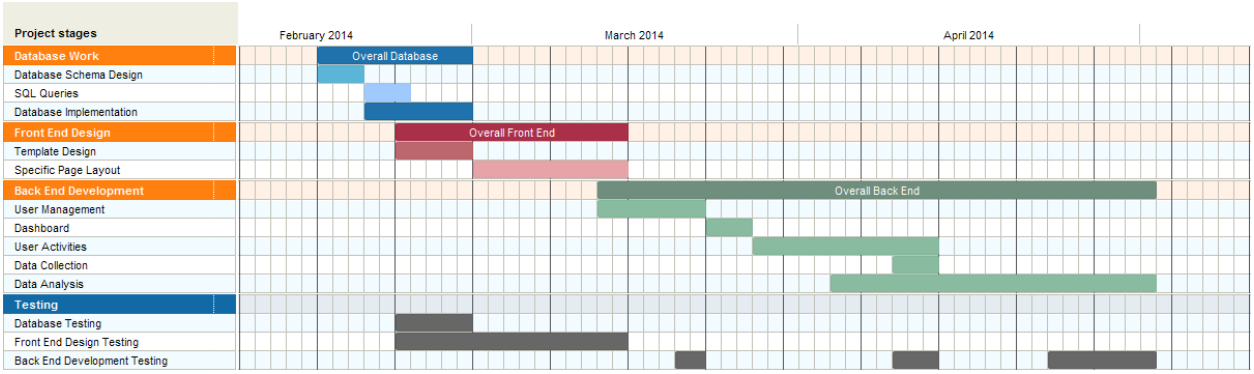
Implementation Plan

Our current implementation schedule is represented in the Gantt chart below. This Gantt chart is just a draft, and encapsulates how much time and effort each part of our project will take. The chart is broken up into the three tiers of our system's architecture.

The first tier shown is the databases, which are shown in blue. This is where we keep track of all our user data through appropriate data tables. We have allotted only a couple weeks for this section because it is applying our designed database schema to our database. We start off with database work because we cannot properly implement our system without a database to test it against.

The second tier is the front-end design, and it is shown in red. This section is all about what the user sees when they visit the SOAR web application. We plan to spend a little more time here because we want the website's design to be aesthetically pleasing, and efficient for a user. Once we get this working we can work on the logic.

The third tier is the back-end development, and it is our logic tier. The logic tier is colored in green. This section is where the majority of the work comes in. This is because we need to ensure the website is completely functional. Additionally, we have to work with our most difficult subsection, which is data analysis. This is where we will analyze the data the user gets from the Prepar3D program, and send the data to the front-end in order to display it to the user.


Project stages
February 2014
March 2014
April 2014
Database Work
Overall Database
Database Schema Design
SQL Queries
Database Implementation
Front End Design
Overall Front End
Template Design
Specific Page Layout
Back End Development
Overall Back End
User Management
Dashboard
User Activities
Data Collection
Data Analysis
Testing
Database Testing
Front End Design Testing
Back End Development Testing