SOAR Web Application

Charles Chavez, Mark Malone, Dillon Postgate

Department of Computer Science, Northern Arizona University

NORTHERN ARIZONA UNIVERSITY

College of Engineering, Forestry & Natural Sciences

Motivation

Our sponsor is True Course Aviation.

True Course Aviation currently has a system called SOAR which is comprised of:

- Custom Aviation Missions
- Prepar3D Flight Simulator

There is no way to store the data from users taking missions in the flight simulator.

There is also no way for users to track progress in their courses.

The SOAR web application addresses both of these problems.

Key Features

The SOAR web application contains multiple key features:

- Institution Management

  Allows the sponsor to add, remove, or edit institutions. This feature is important because it allows for institutions to purchase access to the system.

- User Management System

  Separates functionality for the four user levels: students, teachers, institutions, and SOAR administrators.

- Data Upload

  Feature which allows students to upload their progress reports.

- Data Parsing

  Obtains the information contained in the progress report and stores it into the database.

- Data Analysis

  Generates graphs based on the parsed progress reports and provides feedback to student based on their performance.

Solution Overview

The SOAR web application creates an interface for institutions that wish to teach students in the field of aviation. Our solution creates an end point for the data that the Prepar3D flight simulator outputs, which is necessary to provide feedback to students using the application. The data received from the flight simulator is parsed and analyzed by the application and then displayed to the student through easy-to-read graphs.

Below is a sample use case for a typical user of the SOAR system. A student can run through a course on the flight simulator, upload his progress report, and then see a graph of how they did.

Prepar3D Flight Simulator

Architecture – Three Tier

Presentation tier:

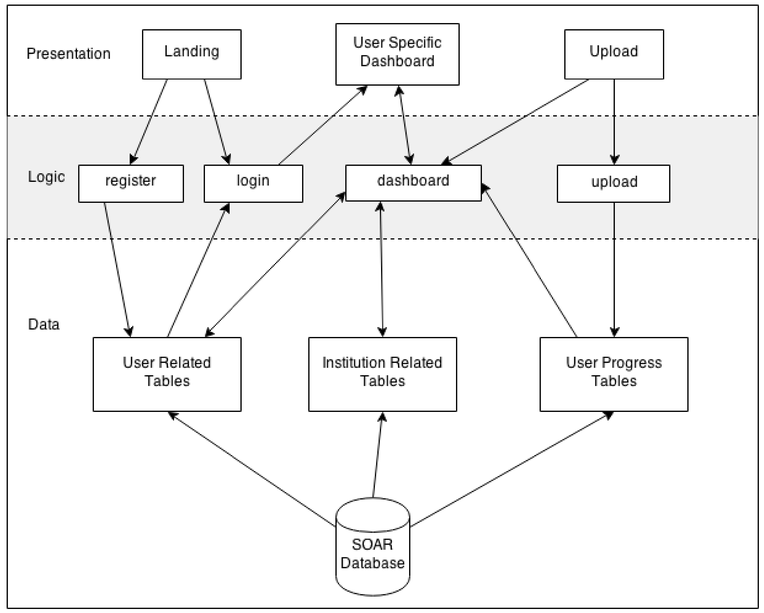
Encapsulates all the components that have immediate interaction with users. This tier is makes up the front-end of the web application which includes every component that outputs information to a user and every component that takes information from a user.

Logic tier:

Acts as the intermediary between the presentation tier and the data tier. Information gathered from the presentation tier is transformed 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.

Data tier:

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.



Challenges and Risks

- Lack of access to flight simulator
- No knowledge of aviation principles
- Co-development of products

Risk	Likelihood	Severity	Mitigation
Data loss	Low/Moderate	High	Daily backups
Bad XML file	Low	Moderate	Capture error and report to developer
Inaccurate reporting	Low/Moderate	High	Unit testing

Technologies

Github
Code collaboration and management

PHP
Server-side web scripting language

MySQL
Database system management

jQuery
Dynamic web page manipulation

Testing

Unit Testing:

This strategy is necessary to ensure that key components of the web application work on their own before we can ensure that the system works as a whole. Specifically, we used this strategy on the parsing and data display components .

Integration Testing:

Integration testing ensures that components communicate properly. Once unit testing is completed, integration testing is used to secure the integration between the parsing and display components.

Usability Testing:

We used three testing strategies to ensure that our system was as "user friendly" as possible: use cases, user surveys, and user task analysis. Use cases involved analyzing expected functionality against actual functionality. User surveys are simple survey questions we asked typical users to see how they feel about the web application. Finally, user task analysis involved timing users when they accomplish specific tasks.