SOAR Design Review III

The Aviators

3/27/2014

Charles Chavez

Mark Malone

Dillon Postgate

Our sponsor

○ Our Sponsor is Arjomand Kalayeh

○ He is the CEO of True Course Aviations

○ What does our sponsor do?

- Teach people how to operate aircraft

- Use flight simulator software

The Flight Simulator

○ Prepar3D

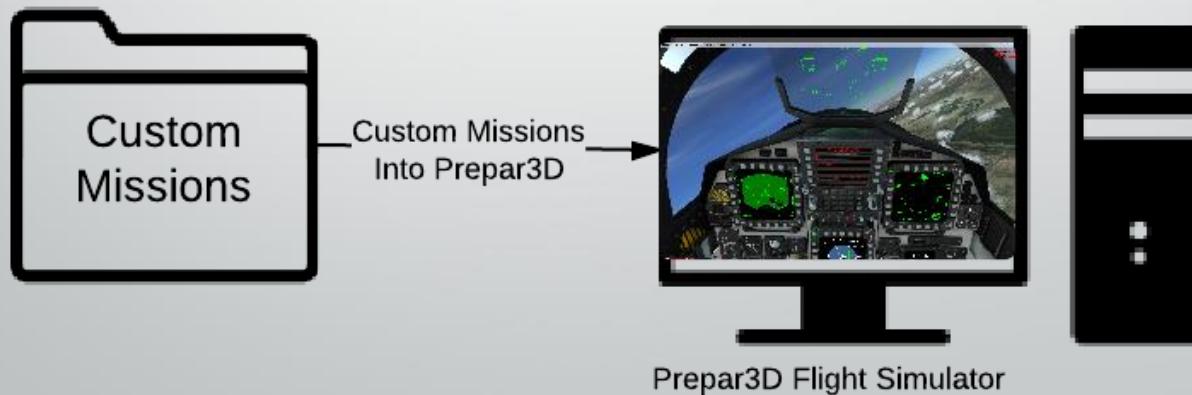- Developed by Lockheed Martin

- Open to development

Sponsor's goal

- Our sponsor wants to monetize pilot training using Prepar3D

- To accomplish this, a system called SOAR is being developed

The current system

○ Custom missions

○ Flight Simulator

- Aggregates user mission data

Custom Missions

Custom Missions Into Prepar3D

Prepar3D Flight Simulator

How is the data handled?

○ There is no current way to process user data

○ Without this component

- Users **will not** receive feedback

- Sponsor **will not** make a profit

Prepar3D Flight Simulator

Key Requirements

Based on sponsor request

Key Requirements - Overview

- User Management

- Data Analysis

- Dashboard

Key Requirements - Users

- User levels

  - System administrator

  - Institution administrator

  - Instructors

  - Student

- Login

- Register

- Session control

Key Requirements - Data

○ Users upload an encrypted XML file

○ Parsed and save to database

○ Determines student statistics

Key Requirements - Dashboard

○ Course selection

○ Upload progress

○ View progress

Our proposed solution

○ The SOAR Web Application

- Will receive data from Prepar3D

- Users will receive feedback

- Sponsor will have the opportunity to profit

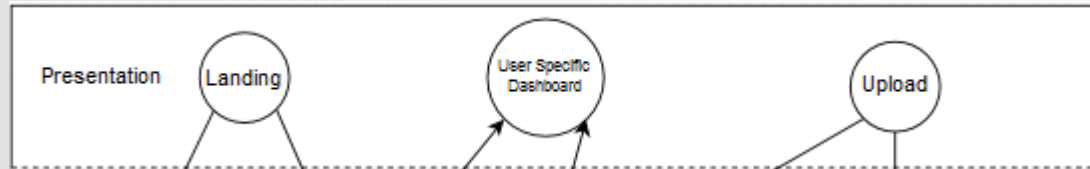Custom Missions → Custom Missions Into Prepar3D → Prepar3D Flight Simulator → User Data → SOAR Web Application

Architecture

Architecture - Overview

Three-Tiered architecture

- ○ Presentation

  - • Web site display
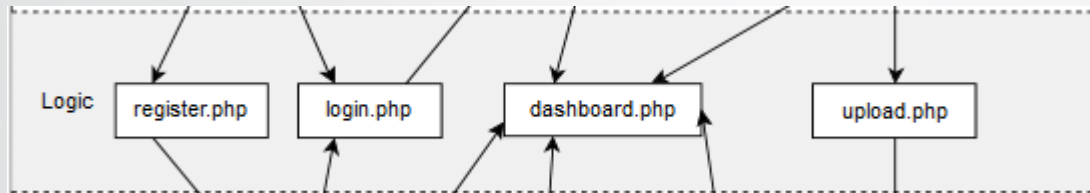
- ○ Logic

  - • PHP scripts

- ○ Data

  - • Tables in database
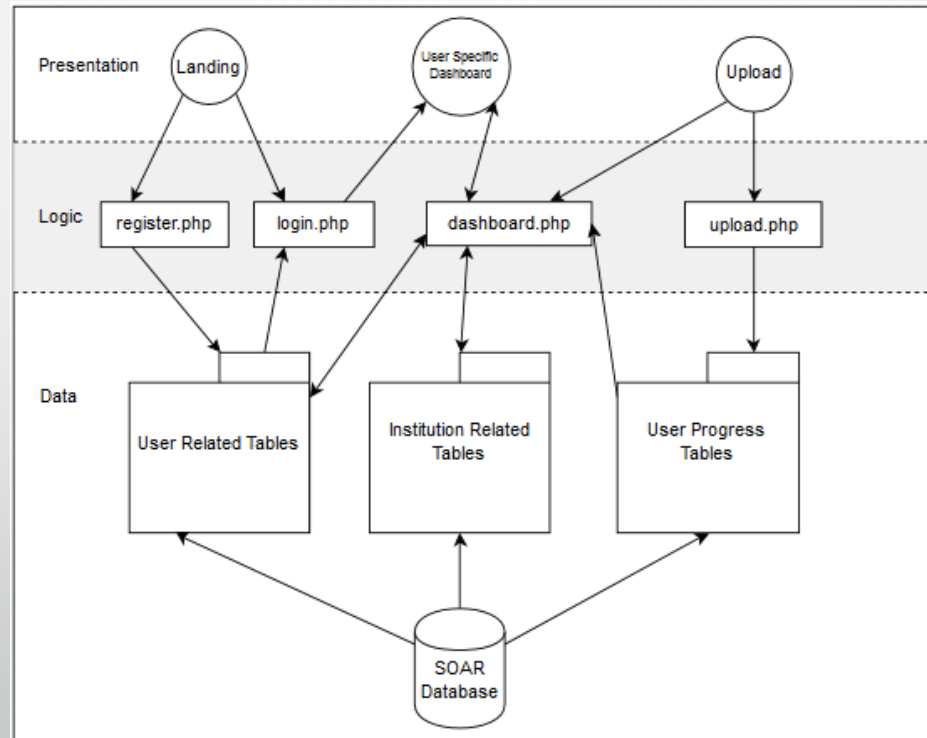
Architecture - Presentation Tier

○ Landing

- Description, institution contact, login, register

○ Dashboard

- Based on user-level

- Management functionality

○ Upload

- Students upload progress

Architecture - Logic Tier

- Register

  - Adds to database

- Login

  - Checks database, adds new session

- Dashboard

  - User-specific functionality

- Upload

  - Parses XML

Architecture - Data Tier

○ User Tables
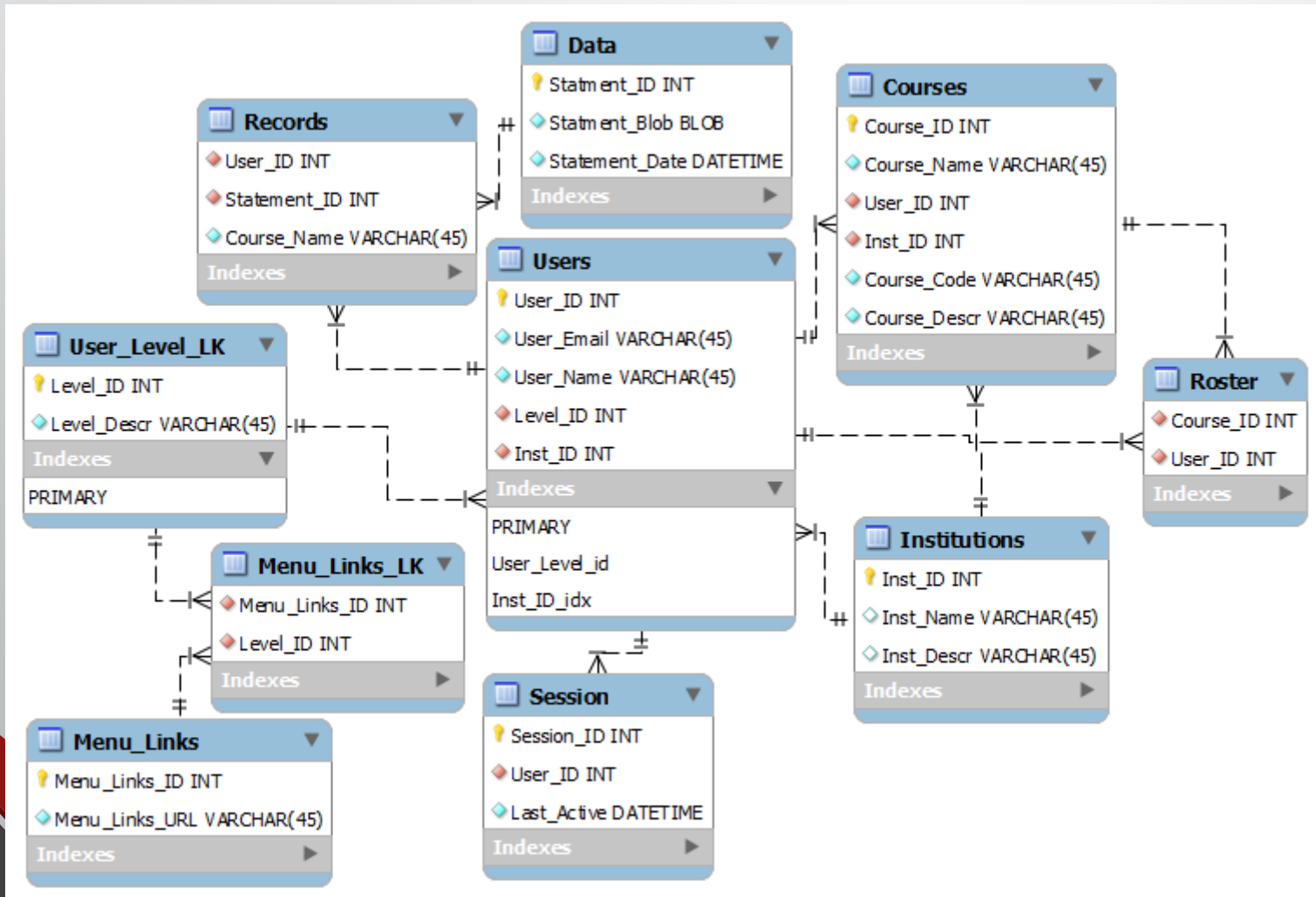
- Stores user and session information

○ Institution Tables

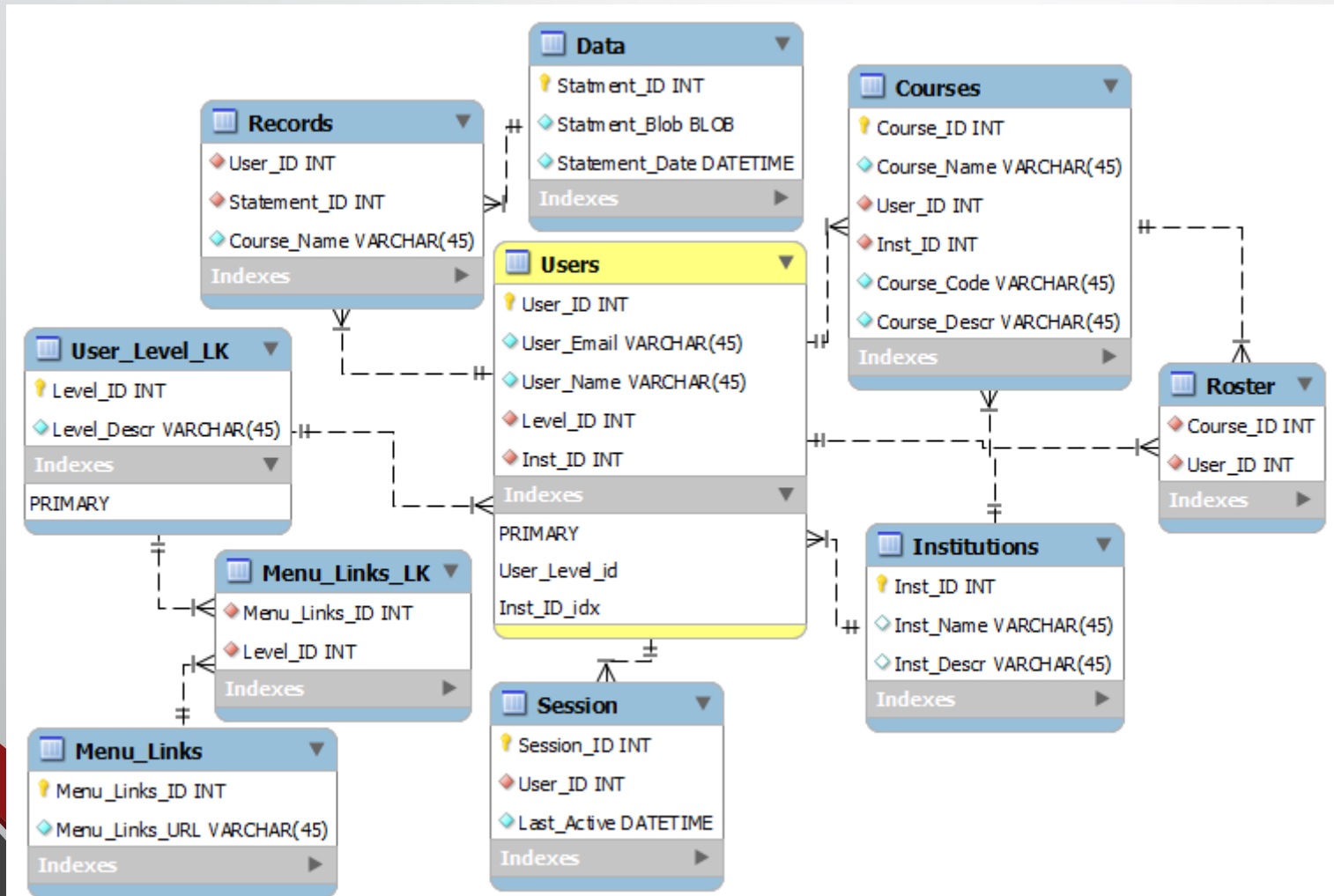- Covers courses, rosters, and institutions

○ User Progress
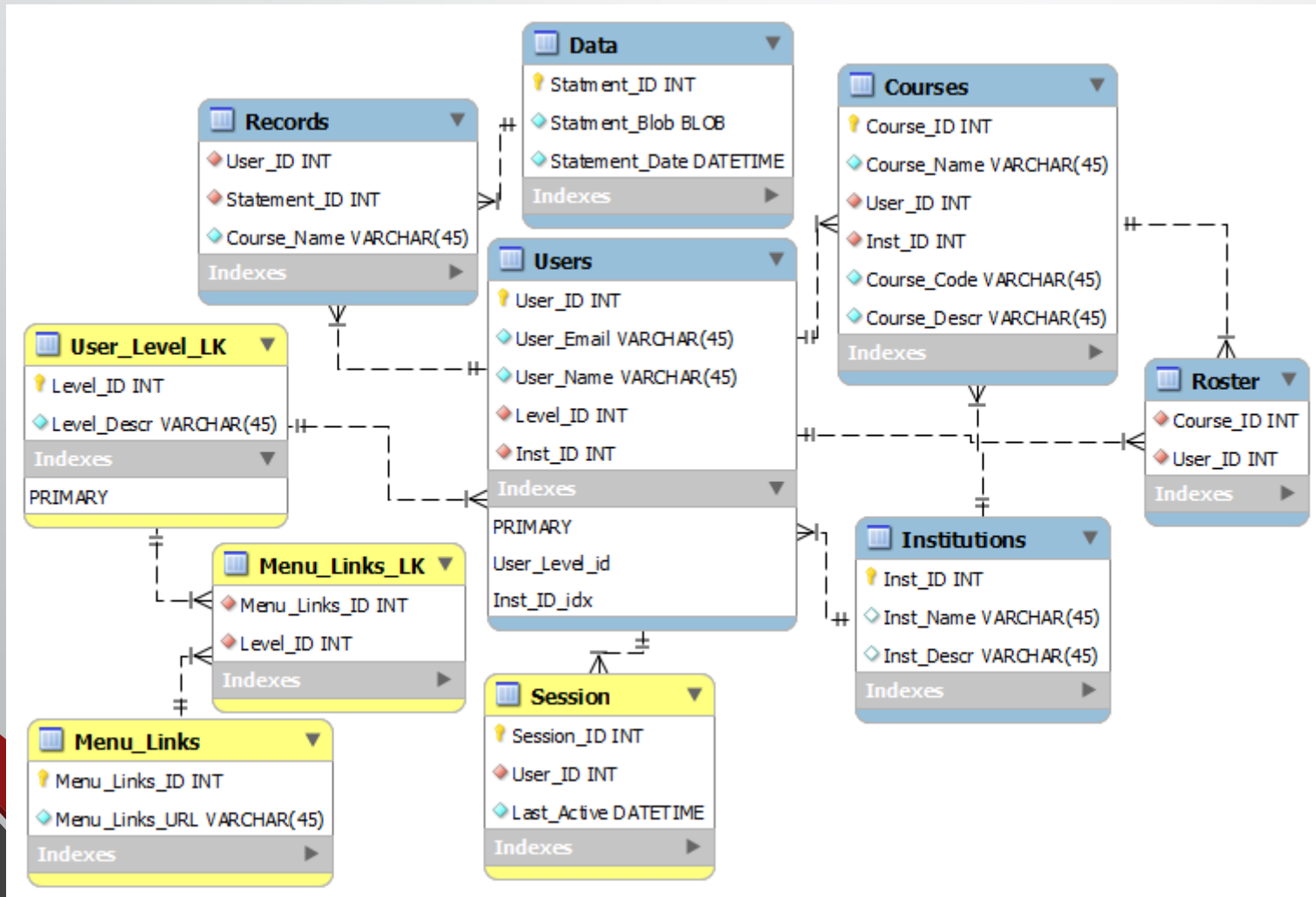
- Stores data from user progress
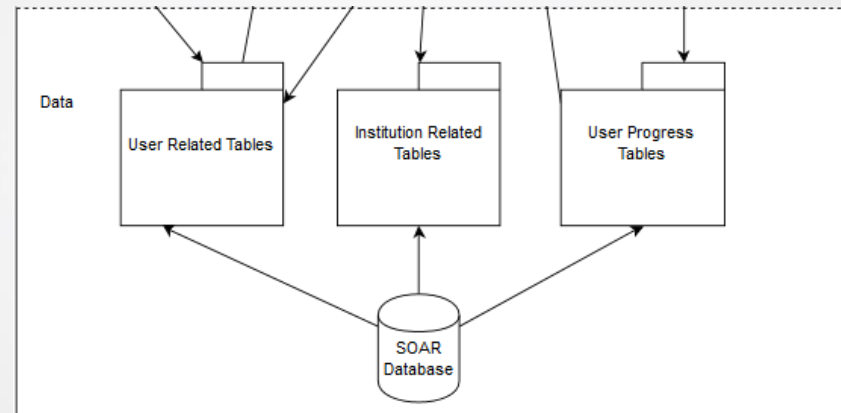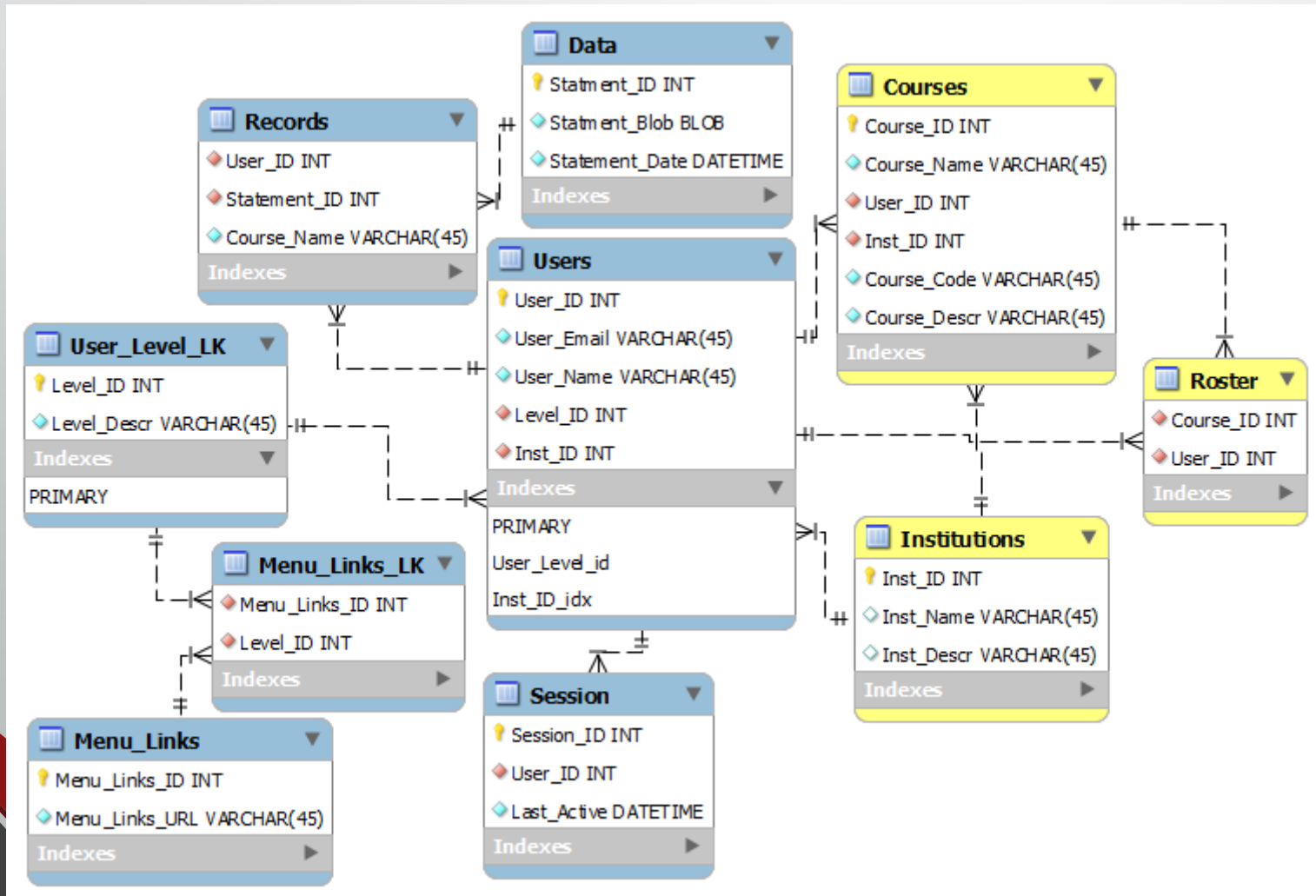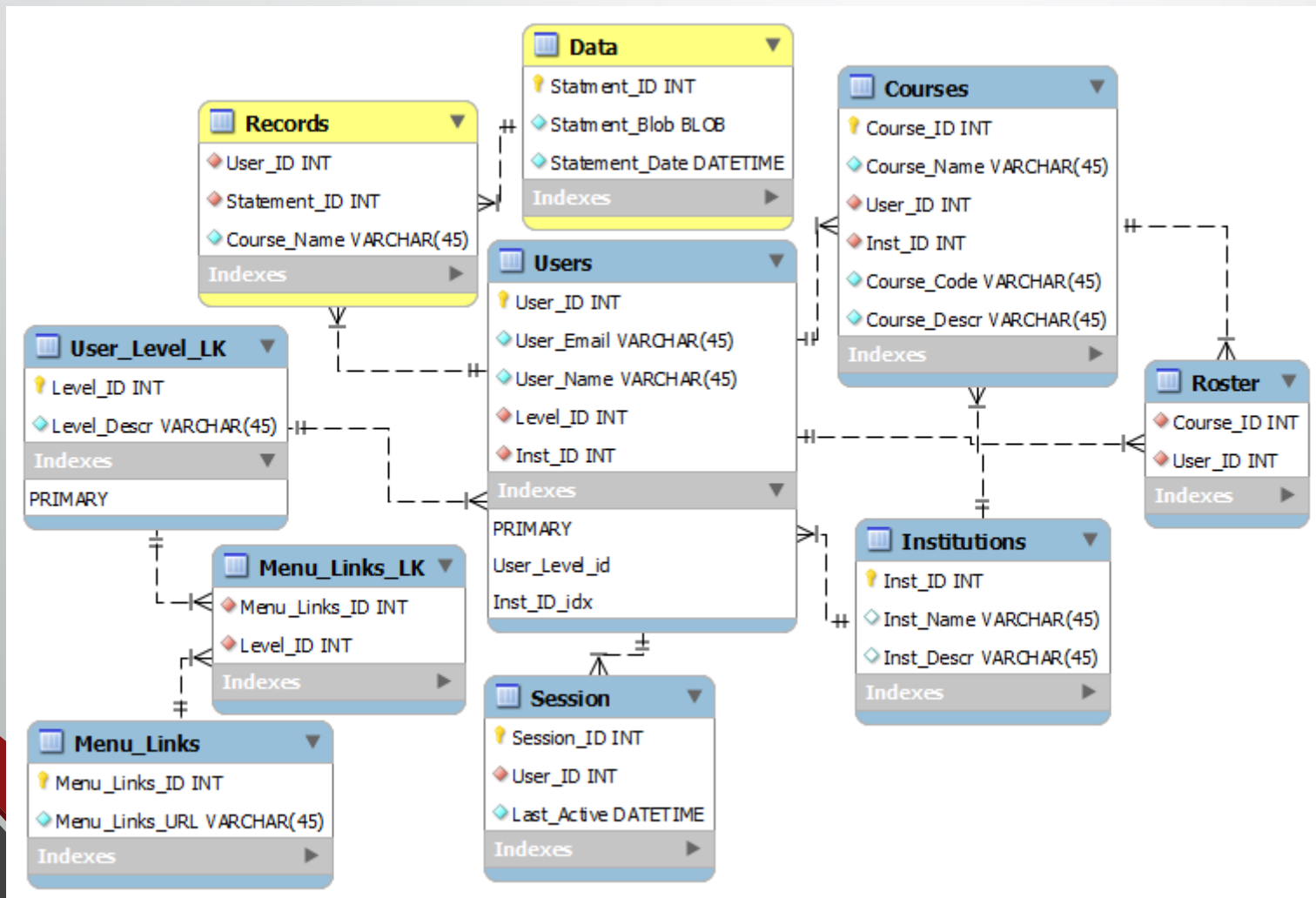
Database Diagram

Revolves Around Users

Session, Levels, Menus

Courses and Institutions
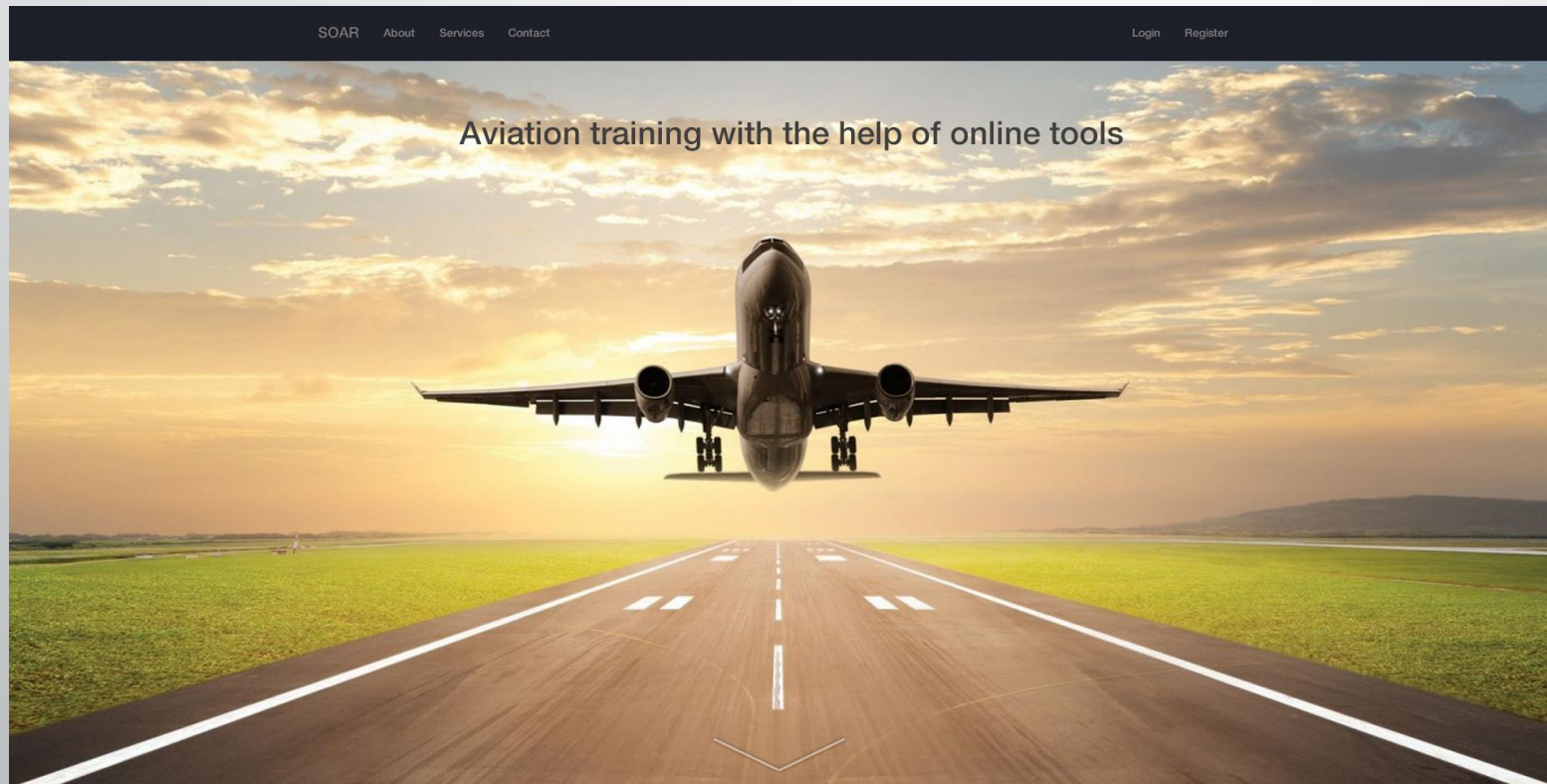
Data

Implementation

Implementation - Progress

○ Slightly behind schedule

○ Completed work:

- Landing

- User Management

- Data Upload

Implementation - Overview

○ Main Landing Page

○ All User Level Dashboards

○ Student Upload (XML File)

Implementation – Landing

Implementation - Menus

Administrators

SOAR Dashboard

Add Institution

Delete Institution

Add System Administrator

Delete System Administrator

Institution Administrators

SOAR Dashboard

Update Institution

Manage Teachers

Manage Courses

Teachers

SOAR Dashboard

View Courses

View Students

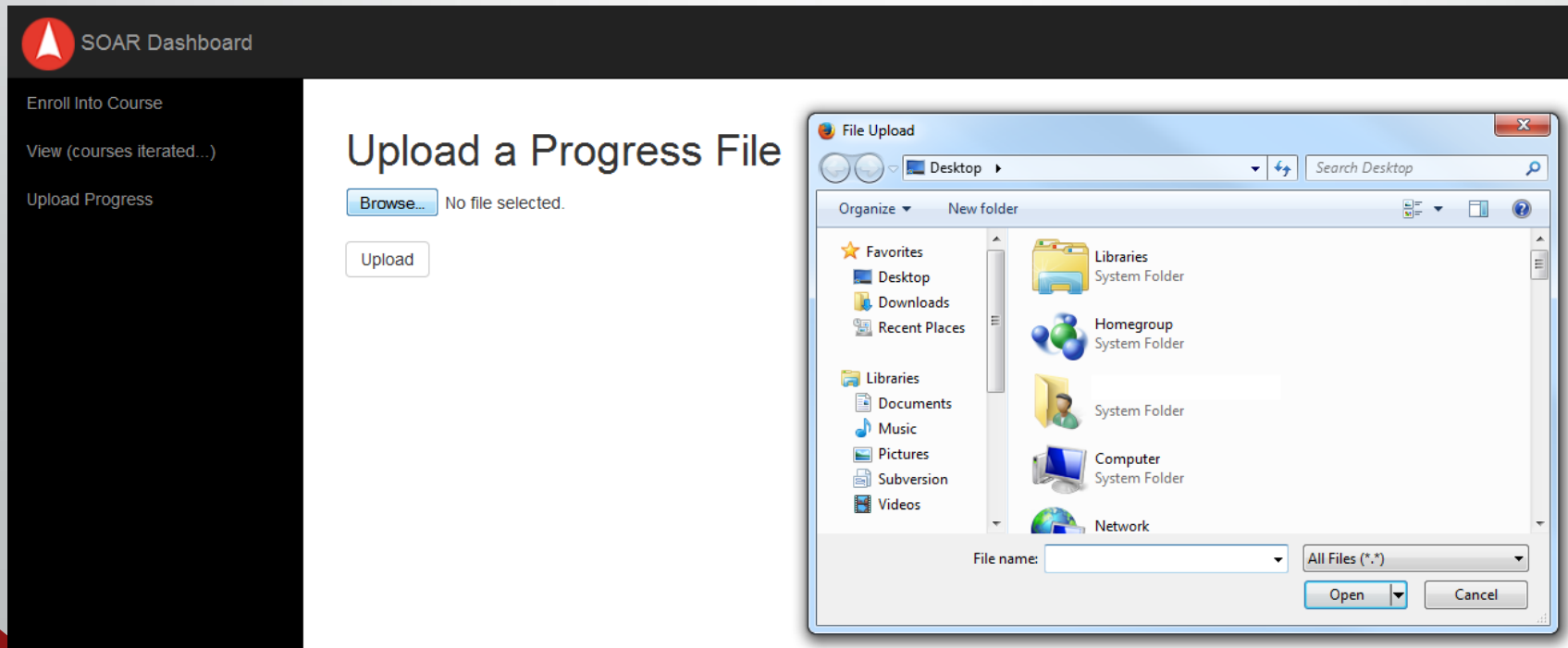**Students Pre-pay**

SOAR Dashboard

Purchase Subscription

Students Post-pay

SOAR Dashboard

Enroll Into Course

View (courses iterated...)

Upload Progress

Uploading Functionality

○ Saved as BLOB type in database

Testing

Testing – Current Progress

○ Front-end web development

- Limited amount of testing

- Compatibility testing

- Current Issues with mobile browsers

○ Back-end development

- Use case testing

Testing - Use Cases

Type	Description	Expected	Pass?	Notes
Register	User chooses to register with no entries	Inform user of missing fields	Yes/No	Only informs user of first fill-in that is empty
Register	User chooses to register with a single missing entry	Inform user of missing fields	Yes	N/A
Register	User chooses to register with incorrect field parameters	Inform user the fields that are incorrect	Yes/No	Will only tell user of the first case of bad input
Register	User chooses to register with a single incorrect parameter field	Inform user the field that is incorrect	Yes	N/A
Register	User inputs non-matching passwords	Inform user that passwords do not match	Yes/No	Doesn't allow, but resets all fields
Register	User chooses to register with everything correct	Successfully add that user to the database verification email sent	Yes	N/A
Login	User attempts to login without verifying their email	System says no, and informs the user of no verification	Yes	N/A
Login	User attempts login with incorrect password	Inform user that password is wrong	Yes	N/A
Login	User attempts to login with incorrect username	Informs user of nonexistent username or incorrect password	Yes/No	Can possibly tell the user their password is wrong even if they just made a typo in username

Testing - Plan

○ Unit Testing

- Parsing XML

- Displaying data

○ Integration Testing

- Interaction between data display and parsed XML

○ Usability Testing

- Surveys to ensure "user friendliness"

Sponsor Challenges

○ Constantly changing requirements

○ Developer communication

○ Lack of access to flight simulator

Risks

Risk	Mitigation
Data loss	Daily backups
Bad XML file	- Format testing - Capture error and report to developer
Inaccurate reporting	Unit testing
Lack of aviation knowledge	Study aviation terms and aviation training practices

Schedule
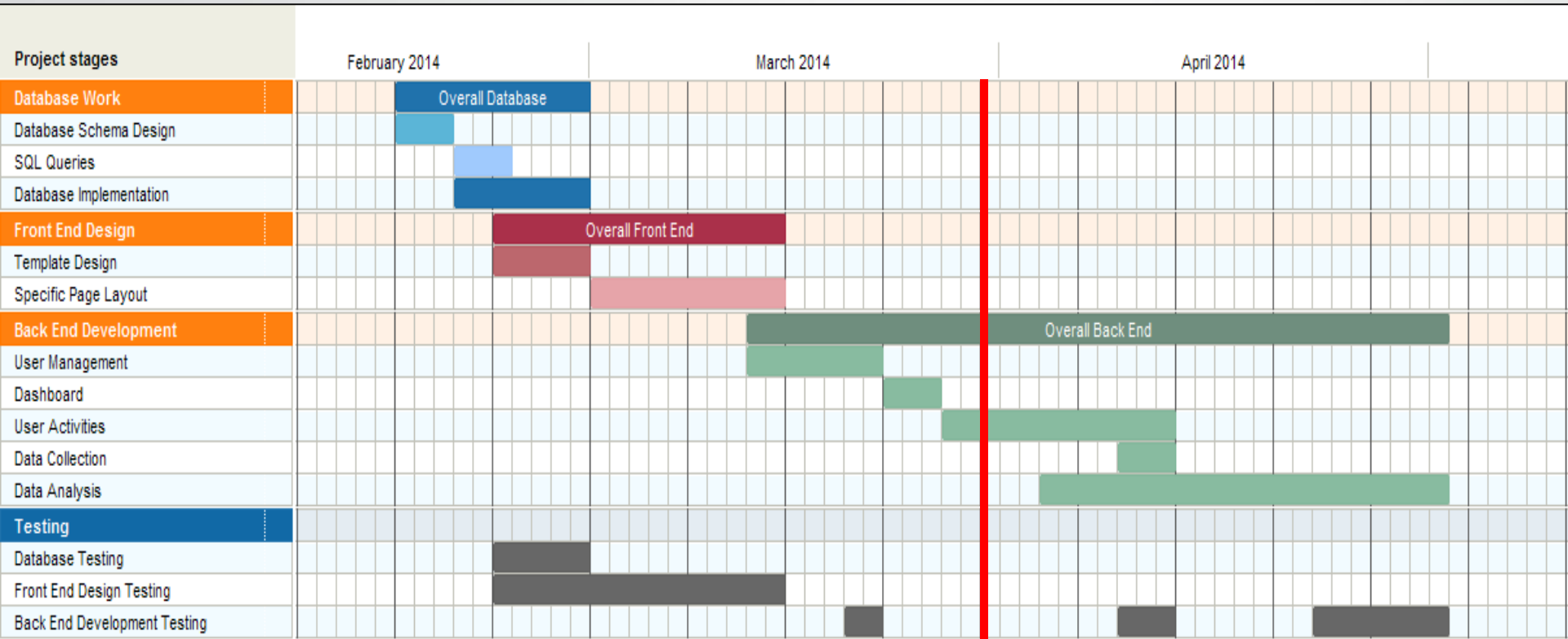
On Schedule

Conclusion

○ Sponsor wants to monetize aviation training

○ We create the bridge between training and improvement based on analysis

○ Empowering students to succeed in aviation

Thank you

Questions and Comments