Simple Airline Management System (SAMS)

CS 4400: Introduction to Database Systems Course

Project: Spring 2023 Semester

Project Purpose

In this project you will analyze, specify, design, implement, and document an online system based on the provided scenario description. You are required to use the classical methodology for relational database development. The system will be implemented using a relational DBMS that supports standard SQL queries. You will use your localhost MySQL Server (Version 8.0 or above) to implement your database and the application. You cannot use any other software like Access or SQLite. Please ask the Instructors and TAs if you have questions.

Project Phases

Inputs (we provide to you…)

• Scenario description • Sample data elements	• Enhance ERD (EERD) • Initial Data Set (in a non-normalized format)	• Physical database schema with initial data set • View & stored procedure shells	• User interface specification
Phase I	**Phase II**	**Phase III**	**Phase IV** *(optional)*
• Enhanced entity relationship diagram (EERD) • List of assumptions (optional)	• Relational schema • Physical database schema with initial data set • Unhandled exceptions list	• Implemented views & stored procedures • Any supporting views and related structures	• Fully functional application integrated with database system • Application source code

Outputs (…you turn in to us)

Phase IV Directions

In this phase, your team will implement a full-fledged, stand-alone application for the Airline Management as described in the project description.

Phase IV vs Final Exam

Phase 4 is completely optional: you can either take the Final Exam, do Phase 4, or do neither, but you cannot do both. If you do Phase 4, then all members of your team must be available for the live demo.

- For those in Section A (TR 3:30-4:45): your final exam is Thursday April 27th from 2:40 to 5:30 PM
- For those in Section B (MWF 2:00-2:50): your final exam is Friday April 28th from 2:40 to 5:30 PM

The biggest "challenge" that we've seen over the years is when one or more members of a team decides, very close to the Due Date (e.g., less than 24 hours in some cases), that they no longer want to do Phase 4, and won't complete their portions of the project, which leaves the remainder of the team in a very bad predicament. We encourage everyone to be very honest about how much you're committed to completing Phase 4. And thinking proactively, we encourage you to setup a realistic development schedule with checkpoints established well before the Phase 4 Due Date deadline to ensure that all team members are on track with their planned implementation efforts and results.

Timeline

1. Project Demos will take place in 45 minute time slots between April 27th at 12:00 PM and May 4th at 5:30 PM. The TAs will be adding their demo slots during the last week of classes. Should any scheduling conflicts arise, email ryantobin77@gatech.edu ASAP.
2. All demo timeslots will be posted by April 24th and you must sign up for your demo timeslot by April 26th @ noon.
3. Because some team members may choose to do phase 4 while others may not, we will allow you to form new teams specific for phase 4. If you decide that you want to do phase 4, you must go into Canvas > People > Phase_4 Teams and move your name from “Unassigned Students” to a team. If you leave your name in the “Unassigned” section, this means you are taking the final exam or are not completing a “final event” assignment (i.e. you are happy with your current grade). If you place your name on a

team, this means you are doing phase 4. Even if everyone on your team decides to do phase 4, you must all add your names to a phase 4 team in Canvas groups. You must do this by April 25th @ midnight. All decisions will be final at this point.

Demo Instructions

- Demos will be held virtually. You sign up for them via the Canvas calendar. Demo slots will be published by 4/24 and you must sign up for your time slot by 4/26 @ noon.
- Have all team members in attendance on time. **No credit will be given to absent members, and 15 points will be deducted from tardy (up to 10 minutes) members**.
- The TA will go through a script of user stories and ask you to demonstrate a comprehensive set of application functionalities
- The TA may ask questions to assess your understanding of the application as well as your participation within the team
- The TA may ask to see your database to ensure changes are persisted there
- The TA won't run your application on their personal computer. A team member (or multiple) will run the application on their computer and screenshare.
- The TA won't try to break your application via SQL injections or some nefarious edge case. However, anything that's listed or depicted in the description is fair game.
- Remember to be respectful of the TA. They are trying to assess your application in a fair and consistent way. They are also in the middle of their own final exams and projects. Be kind to them, and they'll be kind to you.
- You will have **exactly 45 minutes** to complete your demo. We cannot give you more time, so you must come prepared.
- You will not receive your grade directly after the demo. Don't ask for it, as the TA is not allowed to tell you.

Restrictions

- You must use a database. It does not have to be MySQL, but you must use some database to persist data that is not just an in-memory data structure.
- Your code should not be public and should only be shared with your team.
- Your screens should generally follow what we've shown in the description, but you are free to present the UI however you see fit as long as the functionality is met.

During Demo Repairs

As mentioned above, you will have up **to 45 minutes** to complete all the steps. If you encounter any problems during the demonstration process where your queries (or application capabilities) are not working correctly, then we will offer you the opportunity to perform minor "on-the-spot" repairs.

You should weigh this offer very carefully:

- If you choose to "make some repairs", then the clock will continue to tick during your efforts, and you are still responsible for completing as much of the testing script as possible. Steps from the testing script that are left uncompleted will count against your final score.
- If you choose to accept/ignore the errors and continue with the script, then you will likely lose some points because of the errors. On the other hand, this might still result in a better overall score than stopping to make repairs.

Ultimately, this choice is your call to make as a team. The TAs are allowed to let you know where you are in the testing script (e.g. "You've completed 9 of the 15 steps so far..."), and can give you some very general sense of how severe the error is compared to the expected result, but they will not troubleshoot the error for you, nor will they determine the likely impact of the error on the remaining steps of the testing script. We recommend that you discuss this as a team before the demonstration, so that you have a general strategy in advance - time is precious during the demo.

Note that we do expect the demo script to take most of the demo time, so do not submit code on Sunday night intending to make fixes during the demo. These “on-the-spot” repairs are mainly for minor fixes you don’t find out about until the demo.

Demo Script Sample

The below sample is provided to give you a sense of what sorts of tasks the TA will ask you to perform as well as how you will be earning points (point values are hidden below). Note the full demo script is a comprehensive walkthrough of your entire application; below is just a sample.

Go to the Add Airplane screen. Try adding a new airplane with these parameters: [Failure case parameters].

> **+X** for failing to add a new airplane with invalid params.

Add an airplane with these parameters: [Success case parameters].

> **+X** for successfully adding the airplane

Make another addition with those same parameters. [Same parameters as above]

+X for failing to make duplicate plane addition.

Here's another example:

Go to the Flight Landing screen. Try landing a flight with these parameters: [Failure case parameters].

+X for failing to land a flight with invalid params.

Add an airplane with these parameters: [Success case parameters].

+X for successfully landing the specified flight.

Make another addition with those same parameters. [Same parameters as above]

+X for failing to land the same flight again.

Go to the Flights On The Ground View and check to see if flight you just landed is there.

+X for updated **View Flights On The Ground** screen.

Submission Instructions

1. You should submit a zip file including the following:
    a. All code required to setup and run your application
    b. A readme including:
        i. Instructions to setup your app
        ii. Instructions to run your app
        iii. **Brief** explanation of what technologies you used and how you accomplished your application (don't spend too much time on this)
        iv. Explanation of how work was distributed among the team members
2. To be clear, **your grade is almost entirely based on your demo**. The submission serves to ensure you are code complete by the deadline and serves as a deliverable for your efforts.

Version History

Version	Date	Notes
0	April 19th, 2023	Initial Release