# Simple Airline Management System (SAMS)

**CS 4400: Introduction to Database Systems**
**Course Project: Spring 2025 Semester**

## Project Purpose

In this project you will analyze, specify, design, implement, document, and demonstrate an online system. 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 also cannot use any other software like Access or SQLite. Ask the professors or TAs if you have questions.

## Project Phases

*Inputs (we provide to you…)*

| Phase I | Phase II | Phase III | Phase IV *(optional)* |
|---|---|---|---|
| • Scenario description<br>• Sample data elements | • Enhance ERD (EERD)<br>• Initial Data Set (in a non-normalized format) | • Physical database schema with initial data set<br>• View & stored procedure shells | • User interface specification |
| • Enhanced entity relationship diagram (EERD)<br>• List of assumptions (optional) | • Relational schema<br>• Physical database schema with initial data set<br>• Unhandled exceptions list | • Implemented views & stored procedures<br>• Any supporting views and related structures | • Fully functional application integrated with database system<br>• 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 that provides the same functional capabilities as described in the Project Description and Project Phase 3 Instructions.

Phase 4 is completely optional and can be used to replace the lowest grade received on a previous project phase. You will keep the three highest grades across all 4 project phases if you elect to complete Phase 4. If you do Phase 4, then all members of your team must be available for the live demo.

You will have the ability to form new groups for this phase. As a result, you will need to self-assign yourself to a team on Canvas. We will not be forming groups for this phase, but we do have a Team Formation Megathread on Ed Discussion to help facilitate team formation.

## Timeline

1. Teams must be formed in Canvas by **April 11th**.
2. You must submit your project to the Phase 4 assignment on Canvas by **April 22nd**.
3. Project demonstrations will generally take place in 45-minute time slots during the Final Exam Period from April 24th to May 1st. You will need to sign up for one demo slot with a TA on Canvas by **April 16th**. The online slots may appear to be of varying lengths due to the constraints of the scheduling system. All appointments are one hour long, and the start time is rounded down to the nearest half hour. EX: if there is a minute-long appointment for 3:32 PM, the demo is the hour from 3:30-4:30 PM. We will post specific demonstration slots as soon as possible.

## Demo Instructions

- Demos will be virtual and you are permitted to be anywhere in the world when completing them. They will also be recorded.
- Have all team members in attendance on time. **No credit will be given to absent members, and 15 points will be deducted for 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 will 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.
- Object Relational Mapping-based (ORM) solutions – or similar approaches that automatically generate all the functional code for you – are NOT permitted.
- 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 example screens document, 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. The commands shown below are “generic” and not related to any specific project, and the **+X/Y/Z** values represent scores.

**Create a new <entity> with the designated attribute values:**

**+X** for successfully creating the new entity, **+Y** for ensuring that all the new values are correct; or (when appropriate) **+Z** for not creating the entity when one or more system constraints would be violated.

**Modify the attribute values for the following <entity> with the designated values:**

**+X** for successfully modifying the entity’s attribute values; or (when appropriate) **+Z** for not modifying the values when one or more system constraints would be violated.

**Perform an operation that modifies the state of the system (e.g., one or more entities):**

**+X** for successfully modifying the state of the system; or (when appropriate) **+Z** for not modifying the state of the system when one or more system constraints would be violated.

**Remove an existing <entity> from the system:**

**+X** for successfully removing the new entity; or (when appropriate) **+Z** for not removing the entity when one or more system constraints would be violated.

**Display the current state of the system in accordance with a designated view:**

**+X** for successfully displaying the state of the system.

## Submission Instructions

1. You must submit either a .zip file or a link to a **public** GitHub repository 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.

## Recommendations for Getting Started…

This phase of the project provides a lot of freedom in terms of tech stack used, design of the GUI, etc. Such freedom inevitably means that the Instructors and TAs cannot provide as much assistance as previous phases, and especially given the wide variety of systems that might be selected by the teams.

**Tech Stack** – The choice of frontend technology used should be decided based on your familiarity:

- **Frontend** [***Options:*** React, Vue.js, AngularJS, plain HTML + JavaScript, …]
  All these options require basic knowledge in JavaScript. If you are unfamiliar with any of these, you could choose the one you think will be most useful in the future.
- **Backend** [***Options:*** Node.js (JavaScript), Flask (Python), Django (Python), Express (JavaScript), …]
  Any of these can connect to your MySQL server. Choose based on your familiarity with Python and JavaScript. If you are familiar with Python from a previous course, Flask is very easy to set up and use.

**Resources** – YouTube has a lot of tutorials on building a database management dashboard. You can also follow the official documentation of the framework you intend to use.

**Start Early** (!) – If you are unfamiliar with any of these technologies, it can take a lot of time to learn and implement your project. *Starting early provides the opportunity to address any challenges or uncertainties that may arise during the learning and implementation phases.*

**Tips**

- You will not be graded on the interface visual "look and feel" of your GUI. Any reasonable approach will be acceptable.
- Focus on your GUI's functional capabilities and its ability to run queries on your local MySQL server. The bulk of the points for Phase 4 will be based on the correctness of your system's updates to the database state, and its display of the correct query result sets. Balance your time between developing the front end, implementing the connection to the backend, and ensuring that the backend is operating correctly.
- It's your option to work on Phase 4; however, if you have little or no experience with any of the frontend and/or backend technologies listed here, we recommend that you really consider the pros and cons of this approach. Learning and deploying these technologies for the first time can be daunting and time consuming – please take this into account with your decision.

## Version History

| Version | Dates | Notes |
|---|---|---|
| 0 | April 4th, 2025 | Initial Release |