# Guidelines to Group Project

MIS 768, Advanced Software Concepts, Spring 2024

## Project Overview

In this group project, you are required to analyze, design and implement a prototype system in Java, using object-oriented (OO) and Model–View–Controller (MVC) programming techniques. The project will be executed and completed by a self-selected group consisted of three members. A *group grade* will be assigned on the basis of the quality of the system design, the implemented prototype system, and the documentation. The grade for each individual member will then be determined, using the group as a basis and taking into account the group participation and team member evaluation results.

**Note:** Project teams are encouraged to meet with the instructor to review the resulting deliverable(s) together.

## Grading Criteria

| Deliverable | Points | Due Date |
|---|---|---|
| Proposal and Design | 40 Points | March 4 |
| Project Design Presentation | 5 Points | March 4 |
| Final System | 35 points | April 29 |
| Project Report | 15 Points | April 29 |
| Final System Presentation | 5 points | April 29 |

## Main Deliverables

1. *Proposal and design*

   The document should include the following items:

   (a) Background and the needs: Identify the problem to be solved and justify the value of this system.

   (b) Function list: Detailed description of each system function.

(c) User interface (UI) design: Provide a graphical representation of each function. The UI design section should show a mock-up flow of using the system from start to finish.

(d) Class diagram: Provide the class design for the system, including the model and the control classes. Include a description that details the design behind the diagram.

(e) File and database design: Data dictionary for database tables and non-database files. Both relational database and plain text file(s) should be used.

(f) Expectations of project fulfillment: Describe how this system can demonstrate your mastery of the topics required for the system (see Section 2 below).

2. *System*

This system should demonstrate your mastery of the following topics:

(a) Define at least five model classes and instantiate objects of the model classes, including the demonstration of inheritance hierarchies.

(b) Define controller classes corresponding to the views.

(c) Design GUI applications (JavaFX-based or Web-based).

(d) Use array or Arraylist.

(e) Include exception handling.

(f) Both database operations and file handling operations; including the operations of Create, Read, Update, Delete (CRUD). Data files and databases should be populated with at least 25 records in each table and each file.

(g) Documentation of the program using in-code comments and annotations and JavaDoc generated based on the comments.

3. *Project Report*

The documentation should include the following items:

(a)-(e) Updated proposal and design document that is consistent with the implemented system.

(f) Summary of project fulfillment: Describe how the project's final implementation demonstrates your mastery of the topics required for the system (see Section 2 above).

(g) Manual: Instructions on how to use the system, including the username and password.

(h) Response document: Summarize how the comments and suggestions from the peer are addressed.

4. *Presentation*

- Prepare a 7-minute presentation (with face and voice), using slide presentation to cover the system design details. All group members are required to participate in the presentation.
- Prepare a 15-minute presentation (with face and voice), including slide presentation to cover the project overview, the design, and a system demonstration of the functions. All group members are required to participate in the presentation.

**Example Project Topics**

- Alumni Connect: Develop a system that maintains the personal data of the alumni, including the job placement and contact information. The administrators of the system can search, browse, and export data as needed.
- Vegas Strong Resiliency Center: Develop a system that organizes information about the resources available to survivors and those impacted by the 1 October mass shooting. Examples of resources are mental health providers and alternative therapeutic services. The managers would like to be able to note their specialty areas of practice, business hours, contact information, and some other details. They can utilize the system to help match services with the needs of those they serve.
- Board Game: Write a board game of your choice (or your invention). The game should involve multiple players (or computer players).
- Chess Game: Write a program for a chess game between two players. The game starts with a coin toss that determines who starts with White. The players can move the chess pieces on the chessboard. Each of the six different kinds of pieces moves according to the rules. The game continues until 1 player wins or both draw. Ways of winning include: Mating (checkmating) the king; one player resigns if there is no way to win; one player has run out of time and is noticed by the other player.

- Office Scheduler: Write a program handling doctors' schedule. A doctor works from 9 am to 6 pm on weekdays. The user can enter a date to look up the doctor's schedule, and enter new appointments at the doctor's availability. However, a doctor can handle only up to 16 patients per day. The user can also retrieve the weekly and monthly schedule with the starting indicated.

- Customer and Order History: Create a program for managing the customer information, as well as the transactions made by the customers. The managers can create a new customer record and edit the details (e.g., contact information, preference, etc.) later. For each order entered, the order number, transaction date, transaction details (e.g., part number, quantity, and color) should be recorded. When an order is fulfilled, the operator should update the order information. The manager can retrieve a list that shows all the customers and a summary of orders in a certain period.

- Bet Buddies Application: Develop an application where you can propose wagers on anything with your friends. The application will allow anyone to propose a wager with specific details, an expiration date, price (amount to wager/win/lose), etc. Every user can have a friend list, and can create any number of wagers. When all the participants agree on the wager details, the wager would become active. When it reaches the expiration date, the participants should report and agree the result; the wager then would be closed. A user should be able to see the history of all his/her participating wagers.