**App Development Journal**

Week 1:

**Splitwise Logic**  
Wrote the basic logic for a text-based program to determine how to split up bills between multiple people while accounting for tax, tips, etc. Wrote functions to determine the grand price (compute\_price()), the individual price for everyone to pay (split\_my\_bill()), rounding up (round\_up()), removing a person (remove\_person()), printing cleanly (print\_clean()), and doing everything (do\_everything()).

**Carmax Logic**

Wrote the basic logic for a text-based program to determine how to determine the optimal car for a customer. Wrote MakeModelYear class to keep track of a basic car, print\_clean() to print, a class titled “Carmax\_Dexc” tp hold the Carmax description of a car, user\_prefs\_basic() to determine whether the car would meet a basic selection of user criteria, user\_prefs\_full() to output a score that determines whether the car meets every criteria for the user, and tested everything. All structures also have init() functions as needed.

Week 2:

**Tic Tac Toe Logic**

Wrote the basic logic for a text-based function to evaluate a Tic Tac Toe game. Included a board struct to keep track of the board, isValidMove() to determine if a move is actually valid, a Game struct to keep track of the overall game and watch the board, a move function that can mutate the game in order to fully process a move, and switchTurn(), a function called after every move. Additionally, there is a reset function, a checkWin() function, and init functions for everything.

**4 Queens Logic**

This assignment has logic for a solution to the 4 queens problem. The basis of the function is VERY very similar to the tic tac toe functions, but is modified to make sure that the checkWin() function only outputs TRUE when the 4 queens problem has been solved. This required a very slight modification to the total grid present in the UI and substantial hangs to the win condition to ensure that we were oly declaring that a game has been one when the conditions were met.

Week 3:

**CarMax Front End**

This assignment has a VERY basic frontend design for a Carmax app that involves enumerations designed to hold examples of body type, defect count, owner count, etc. Information is displayed using VSTacks and an array of .padding(), Pickers, and Focus variables are used to space things out properly, ensure that items can be selected from the stacks, and that they keyboard is displayed on the iOS platform whenever necessary.

**Basic Project: Task Manager**

This assignment has a very basic implementation of a task manager using enums, structs, classes, etc. Though this only contains a text-based implementation, the logic can easily be transferred to a Service function in order to create a more functional UI of the manager with functional logic. Notable functions include deleteTask, addTask, change priority, etc.\

Week 4:

**Basic Project: SmartHome**

This assignment has an implementation of a smart home controller, in which a user can enter devices and track energy consumption through the text-based interface. There are individual functions designed to minimize the user’s interaction with the program for the sake of time efficiency, including functions that encapsulate the calculation of energy consumption for an hour, a day, a week, etc.

**Basic project: Library Management**

This assignment has an implementation of a library management system that is able to store books by ISBN, as well as track the books being borrowed by individual borrowers. By using classes, enums, and structs to represent the individuals, systems, classifications (overdue, checked-out, available, etc.) and other information, I was able to encapsulate a large amount of the functionality through a text-based software.

Week 5:

**Tic Tac Toe Visualization (done in code comments)**

The visualization of the app was done in the Code Comments for each of the four files submitted under this assignment. I made sure to point out how references to external pages (e.g. GameService from StartView) were affecting the logic in the functions to ensure that the visual representation of the application’s dependency on GameService was shown throughout.

**Basic project: Music Playlist Manager**

This assignment includes a basic, text-based implementation of a music playlist manager. Examples of this in the real would would include Spotify, Apple Music, Pandora, etc. Notable functions within this assignment include functions to shuffle, sort, and filter the songs by genre.

Week 6:

**Basic Project: Advanced Restaurant Management System**

This assignment was the most difficult one that I attempted, which is why it took the whole week. For this assignment, I had to encompass an entire restaurant’s schedule in code and account for menu pricing, ordering, and staff scheduling through a very complicated set of data structures, enumerations, and functions.