# Final Term Project Documentation

## Introduction

This document provides the required documentation for the Final Term Project in CIS 3270. It includes an overview of the project requirements and specifications, class diagrams, data models, and a detailed explanation of the program's functionality and flow.

## Requirements and Specifications

The project simulates an airline reservation system with functionalities similar to platforms like Expedia.com or Cheaptickets.com. Key features include user registration, login, password recovery, flight search, booking, cancellation, and admin-specific capabilities. The application must be functional, handle exceptions, and maintain data integrity.

## UML Class Diagram

The following is a placeholder for the UML Class Diagram. It illustrates the relationships and interactions between various classes used in the project.  
A screenshot of a computer

Description automatically generated

## Data Model

The data model defines the structure of the database used in the application. The following table outlines the primary keys (PKs) and foreign keys (FKs) for the database tables.

## A screenshot of a computer Description automatically generated Functionality and Program Flow

The program follows a structured flow to provide a seamless user experience. Below is a brief overview of the functionality and program flow:

• LandingPage

Serves as the entry point with options to log in as a customer, register, or log in as an admin.

Layout uses VBox for alignment and navigation buttons for each option.

• RegisterPage

Allows new users to create accounts by filling out personal details like name, email, SSN, and a recovery question.

Validates input and stores user data in the database upon successful registration.

Includes a "Back" button to return to the landing page.

• Login Pages

AdminLoginPage: Validates admin credentials, including a hardcoded admin code, and navigates to the admin panel.

CustomerLoginPage: Validates user credentials and navigates to the user page upon successful login.

• ForgotPasswordPage

Enables password recovery by validating a user’s response to their security question.

• AdminPage

Central hub for administrative actions: create, delete, update flights, or log out.

Buttons navigate to respective pages for managing flights.

• Flight Management Pages

CreateFlightPage: Provides input fields for flight details and adds new flights to the system.

DeleteFlightPage: Deletes flights based on flight number using a SQL DELETE query.

UpdateFlightPage: Updates existing flight details with input from the admin.

SearchFlightPage: Allows users to search for flights based on various criteria and displays results dynamically.

• ViewFlightPage

Displays a user’s booked flights. Lacks implementation for fetching flight data, requiring database query integration.

• UserPage

Offers options for logged-in users to search for flights, view booked flights, or log out.

• Page & PageManager

Page: Abstract class serving as a blueprint for all application pages.

PageManager: Manages page transitions and initializes all pages.

• GUI

Entry point of the application, responsible for initializing the PageManager and setting the initial scene.