**Air Ticket Reservation Management System**

**ATRMS**

***Software Development Progress Report – Part 1***

**Abstract**

This document keeps track of the progress of the initial system development phase (Part 1). It includes project objectives and overview, user’s requirements definition, existing systems study, ERD, potential screen designs and, finally, client’s comment and suggestions sections.

**Air Ticket Reservation Management System (ATRMS)**

**Project Objective and Overview**

Project objective is to produce a complete working ***Air Ticket Reservation Management System (ATRMS)***. The system is a kind of **CRS** **(Computerised Reservation System)**, an information system which promotes ticket sales and provides fast and reliable information of bookings, prices and availability of services and products in tourism industry. CRS is later upgraded into **GDS (Global Distribution System)** and largely used by travel agencies for selling tickets from multiple airlines.

In general, the system should include the following functions and features.

The system should have a login form to differentiate the types of user, i.e. administrator or ordinary users.

**User’s Point of View**

- should be able to search for desired flight, time and destination information.

- should be able to book the ticket if the desired flight, time and destination are found.

- should be able to complete the passenger registration information form.

**Administrator’s Point of View**

- should be able to create, update, search or delete flight, route tables.

- should be able to check passenger’s information and flight transactions table and be able to correct if any error is found.

**User’s Requirements Definition**

User’s requirements are defined according to the following possible scenarios.

**Note:** the underlined words will probably be the entities for the system.

**Scenario A (Book the ticket)**

A passenger wants to travel from origin (A) to destination (B). Therefore, he searches for the flight information on the website whether there is a flight available from A to B at a given date and time (departure and arrival). If he finds the flight information at a desired date and time, he will book the flight for that date and time. (The system should display either the eco or business class has chosen and show the available seats in the specified class.) If the seat is available, he will book the seat and fill up the form for passenger details. And proceed to payment details form and pay the amount due. After the payment is confirmed, a printable receipt copy will be generated and sent to passenger’s email address for record. (As cancellation is still possible before the travel date, a given allowance period for cancellation prior to the actual departure date should be identified here).

**Scenario B (Cancel the ticket)**

If the passenger wants to cancel the ticket within the allowance period, the system should check the ticket details and accept the cancellation, i.e. delete the reservation and move it into cancellation table, if possible, to know how many has been cancelled at a later period).

**Scenario C (Administrator create the various entries)**

Administrator performs CRUD for the flight information, can be able to view and correct errors in passenger information, reservation and cancellation details.

**Existing Systems Study**

The following list is some examples of existing system that I take as a reference.

<https://1000projects.org/dfd-er-class-diagram-for-airline-booking-system-project.html>

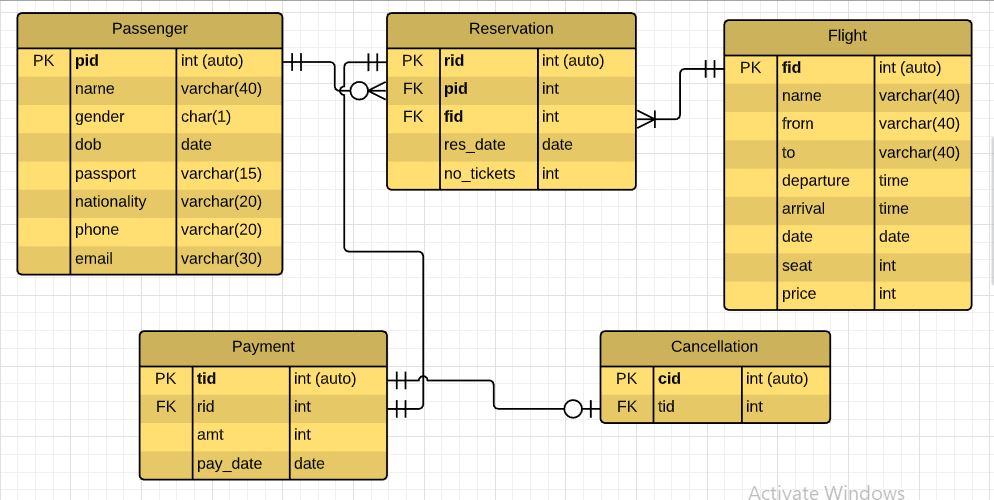
<https://1000projects.org/ms-dissertation-topic-on-airline-reservation-system.html>

<https://www.academia.edu/33078817/Air_Ticket_Reservation_System.docx>

<https://1000projects.org/airline-ticket-reservation-system-project-in-java.html/comment-page-1#comments>

<https://www.slideshare.net/AngelinaNjegus/lesson-3-from-computer-reservation-systems-to-global-distribution-systems>

**Entity Relationship Diagram (ERD)**



*Fig. Entity Relationship Diagram for ATRMS*

**Foreign Keys and Relationships Constraints**

**Note:** primary keys in all tables use auto-numbering system.

The relationship constraints for the tables are as follows.

**Passenger - Reservation**

One passenger may book zero or many reservations. But one reservation is made by one passenger only.

**Flight - Reservation**

One reservation can have only one flight. But a flight has one or many reservations. \*(Assume all the flights are running well and have at least one reservation for each one.)

**Payment – Reservation**

One reservation has only one payment and similarly, one payment can be made for only one reservation.

**Payment – Cancellation**

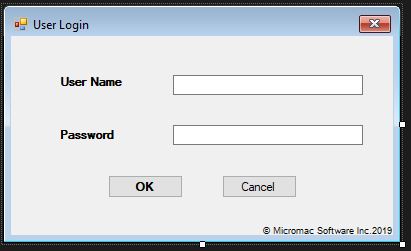
Zero or one payment should be cancelled, but only one cancellation is for one payment.

**Note:** When cancellation is made, the cancelled ticket will be moved to cancellation table for recording purpose and the original record in the “Payment” table will be deleted.

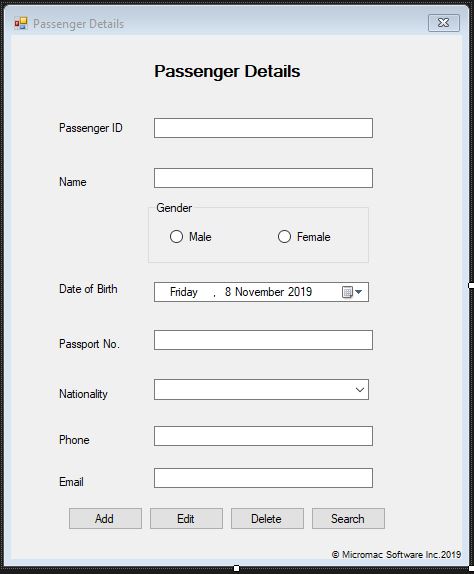
**Potential Screen Designs**

These screen designs are created in Visual Studio. This is one of Rapid Application Development (RAD) techniques called “Throw Away Prototyping”. As the actual system will be a web-based system, these screen designs will be thrown away upon user’s agreement on the proposed designs. When user’s requirements are vague, this is a good way to present to user whether both parties share the same idea or not.

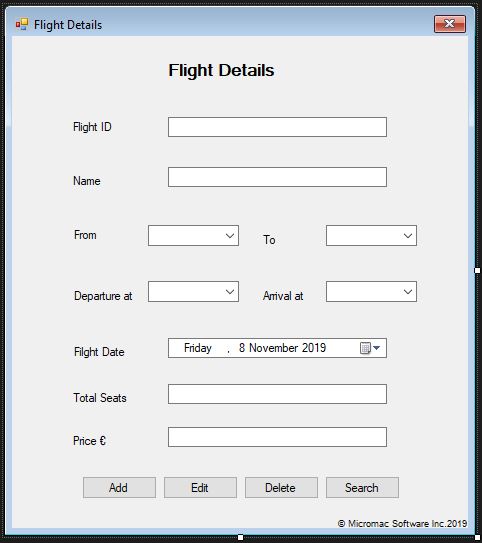
**Login Form & Add New User Form**



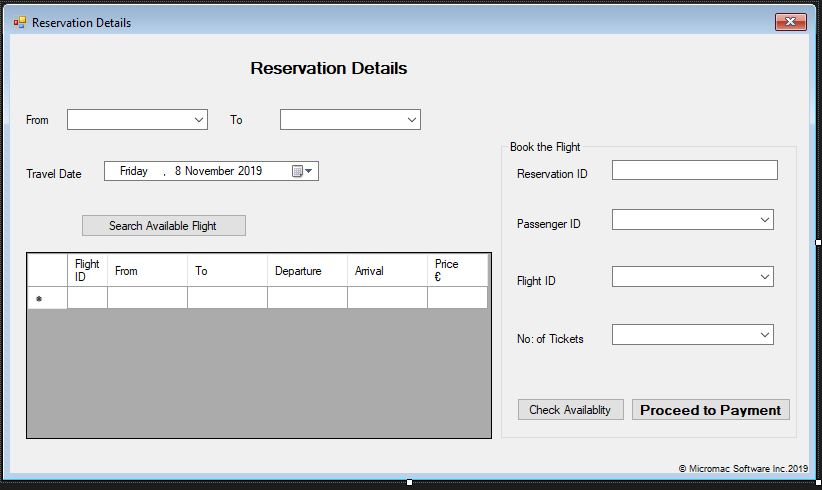
**Passenger Details**



**Flight Details**



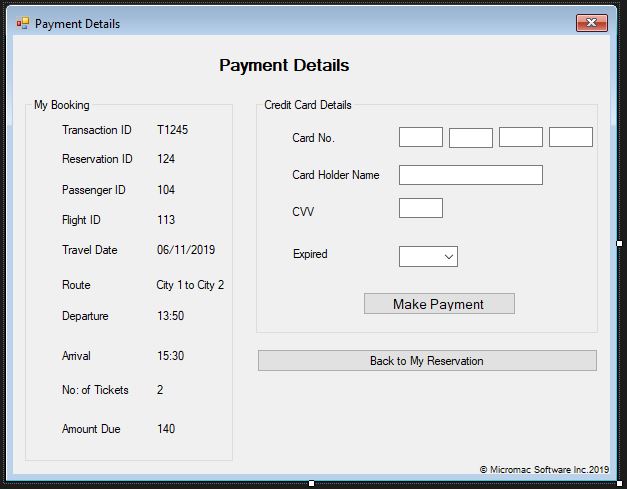
**Reservation Details**



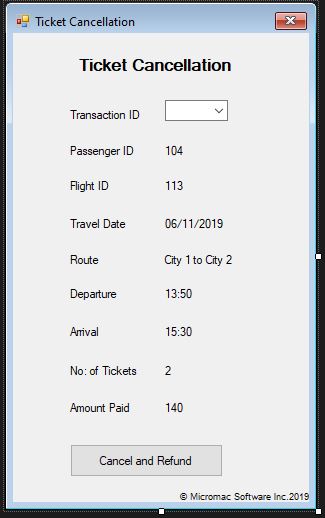
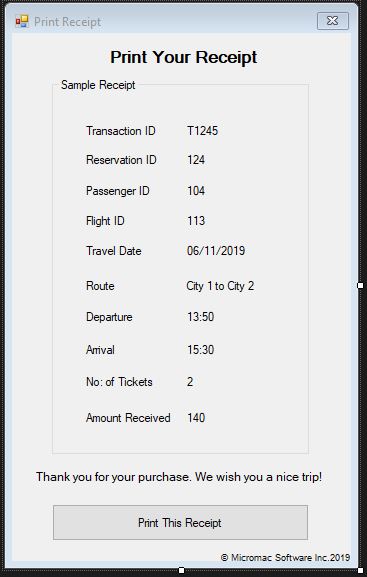
**View All Data**



**Payment Form**



**Print Receipt & Cancellation Form**



**Client’s Comments and Suggestions Form**

Dear Client,

Please fill up the comments for each section and, suggestions or ideas (if you have any) for each section.

**Project Objective and Overview**

Comment:

Suggestion:

**User’s Requirements Definition**

Comment:

Suggestion:

**Entity Relationship Diagram**

Comment:

Suggestion:

**Potential Screen Designs**

Comment:

Suggestion:

Thank you very much for your support on this application development.

Yours Sincerely,

***Myo Thet Tun***

Application Developer

**Micromac Software Inc.**

**Date: 09/11/19**

**Our motto: Software quality is satisfaction of requirements, not goodness.**