S2G4 – MyID90

Thomas Bioren, Carson Holscher, Chris Kim

Table of Contents

1. Executive Summary

This document describes our project and the problem it will solve. The document introduces the problem, provides a short description of the solution, outlines the project's scope, and gives some primary success criteria.

Some background knowledge is required to understand the problem MyID90 aims to solve. One benefit of working for an airline is that employees are allowed to purchase tickets on commercial flights at a 90% discount off the standard airfare. This is commonly referred to in the industry as an ID90 ticket. However, the employee is not guaranteed a seat on the flight with a purchased ID90 ticket; instead, the employee has purchased the right to fly standby on the flight, meaning that the employee will only be able to travel on the flight if there is an available seat. In order to determine whether there are available seats on any given flight, the employee has to "check the loads" (i.e., the employee checks to see how many seats are still available on the flight). An employee "checks the loads" on a flight by accessing an internal company database with real-time flight loads. Employees can easily "check the loads" on flights operated by their employer because they have access to its internal database. However, as stated above, airline employees can also use ID90 tickets for flights that are not operated by their employer. This is the problem. Because these ID90 tickets are still only available on a standby basis, the employee needs to know what the loads (i.e. seat availability) is on the flight but is not able to access the internal database that provides that information because the employee does not work for that airline. As a result, the employee has to purchase the ID90 ticket and simply hope that there will be seats available on the flight (something they will not know until the rest of the regular paying passengers board the plane) or try to contact an employee of the airline they are hoping to fly on and ask that employee to check the airline's internal database for them. This is an unreliable and convoluted process. MyID90 will solve this problem.

2. Introduction

This document exists to help us plan what we will do in future weeks. This document is the first of many. There will be a sample data source, periodic reports, a security and data integrity analysis, a front-end prototype demonstration, and a final report. Currently, we have our Elevator Statement, Primary Success Criteria, Key Stakeholders, Expected Functionality, Entity-Relationship (ER) Diagram, Relational Schema, Indices, and Glossary. These components will show how the prospective database will be created, managed, and evaluated to determine its success. A final presentation of this database will be created and presented by the end of the quarter.

3. High Level Problem Summary

3.1. Elevator Statement

MyID90 aims to create a simple application for airline employees to use to find the loads on flights. Rather than relying on people's good faith, MyID90 creates a "currency" for airline employees. Employees can earn "tokens" by looking up loads for other employees and can spend tokens by requesting flight loads.

3.2. Primary Success Criteria

S2G4 aims to have a minimum viable product (MVP) by the end of Spring Quarter 2024. An MVP must: 1) allow users to "purchase" load requests with tokens, an arbitrary currency that forces users to fulfill other user's load requests, 2) allow users to earn tokens by fulfilling load requests, 3) allow users to build trips with individual flights.

3.3. Scope

3.3.1. Within Scope:

- Airline employees
- Airline employees' dependents
- Trips
- Load requests
- Airlines
- Flights
- Destinations (airports)

3.3.2. Outside of Scope:

- Searching for trips (no search algorithm for finding trips)

4. Detailed Problem Statement

4.1. Key Stakeholders

Airline Employees: Users

Dependents of Airline Employees: Users

Airlines: Source of our data

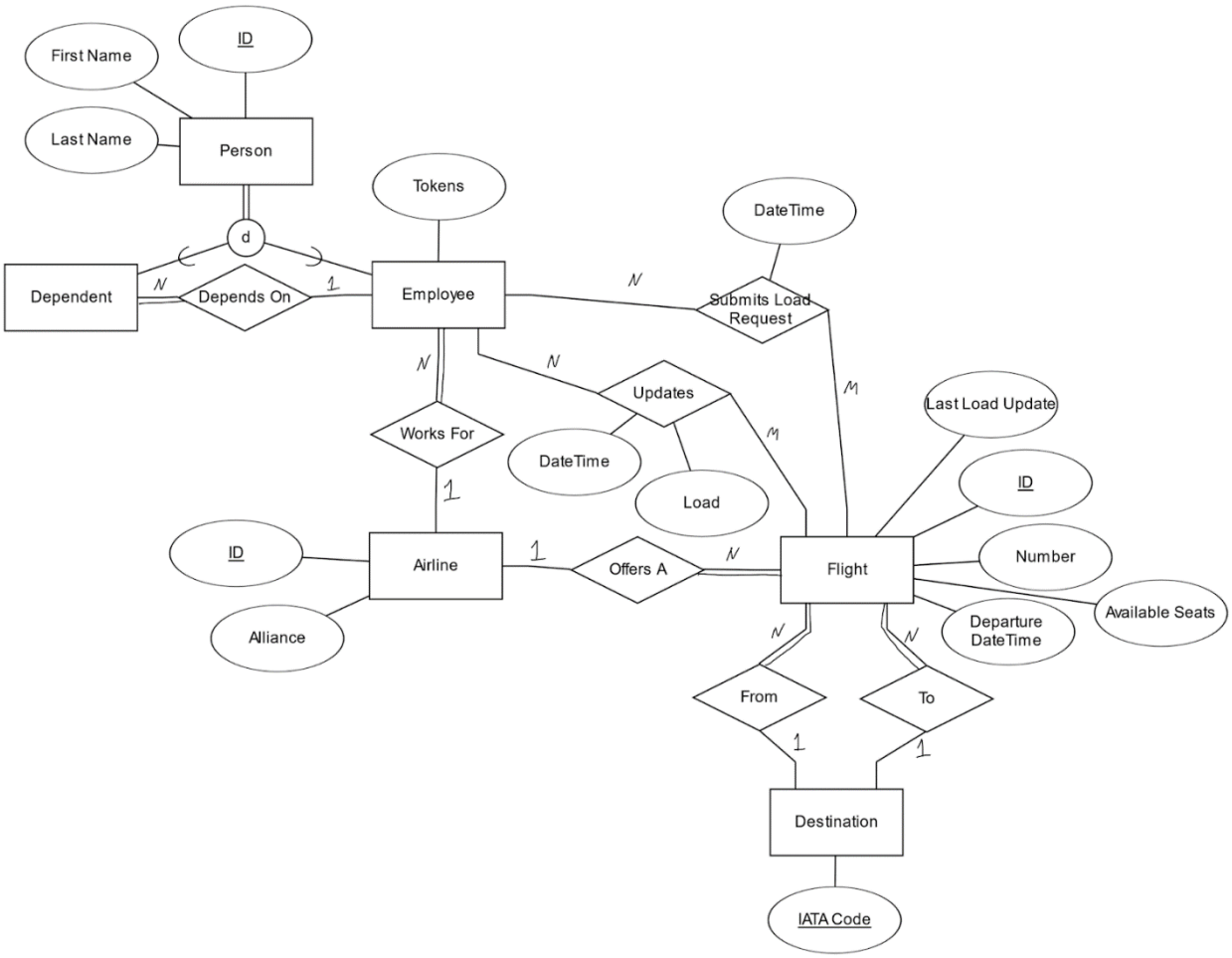
4.2. Expected Functionality

- Users can request data about another airline’s flights at the cost of tokens. It will cost 1 token to submit a request.
- Users should be able to create a plan for a trip from individual flights.
- Other users can fulfill requests for data about their airline’s flights to receive tokens. A user will receive 1 token to fulfill a request.

5. References

6. Appendix

6.1. ER Diagram



6.2. Relational Schema

Relations:
Employee: (ID, FirstName, LastName, Tokens, AirlineID)
Dependent: (ID, FirstName, LastName, EmployeeID)
Submits Load Request: (EmployeeID, FlightID, DateTime, Token)
Updates: (EmployeeID, FlightID, DateTime, Load, Token)
Flight: (ID, LastLoadUpdate, Number, Load, DepartureDateTime, FromCode, ToCode, AirlineID)

Airline: (ID, Alliance)
Destination: (IATACode)

Foreign Keys:
Employee(AirlineID) -> Airline(ID)
Dependent(EmployeeID) -> Employee(ID)
Submits Load Request(EmployeeID) -> Employee(ID)
Submits Load Request(FlightID) -> Flight(ID)
Updates(EmployeeID) -> Employee(ID)
Updates(FlightID) -> Flight(ID)
Flight(FromCode) -> Destination(IATA Code)
Flight(ToCode) -> Destination(IATA Code)
Flight(AirlineID) -> Airline(ID)

7. Index

8. Glossary

ID90 – Industry Discount 90; a 90% discount for airline employees to fly standby on any airline.

Load – How full a flight is.

Token – A currency that force users to fulfill load requests for other users

Load Request – A request made to the users of the database for the load on a given flight.