Data Management Layer

Database Management System - Object Persistence Format

The Husky Air plane rental system should use an Object-Oriented Database Management System (OODBMS). An OODBMS is able to support both simple and complex data types and also directly supports object-orientation. The use of an OODBMS system can help Husky Air avoid impedance mismatches between objects in the software system and data stored in tables. Additionally, the Husky Air system may need to store complex data types such as images, videos or audio so an OODBMS should be used to support these needs.

Since the Husky Air system will enable customers and other users to make, update, and cancel plane and instructor reservations, generate invoices, and perform other functions, the system needs data storage formats that are well suited for continuous data updates from many users and respond quickly to queries from users. An OODBMS can support transaction-processing systems such as this.

The system needs to support random access files and operations such as finding and updating specific objects in an efficient manner. The application will utilize all different types of files including master files, lookup files, transaction files, audit files and history files. Examples of master files for the system include customer and instructor profile information and information about the Husky Air planes. The system will use lookup files in certain instances, for example if a pilot is renting a plane the system can verify that pilot is qualified to rent a specific plane by using lookup files to validate a pilot’s certifications and license type. Transaction files will be used to update the master files and to update plane and instructor schedules, customer reservations, and so on. Examples of history files in the system will include old customer and instructor data, information about past plane rentals and instructor lessons, and other past transactions.

Data Access and Manipulation Classes

There are several data access and manipulation classes are required for the system. The DAM classes include: **Customer-DAM, Instructor-DAM, Rental Order-DAM, Scheduled Flight-DAM, Plane-DAM, Schedule-DAM, and Bill-DAM**. These classes will need to be capable of at least reading and writing to the problem domain objects and tables.

Use Scenarios

Use Scenario: Existing Customer Reserves a Plane

1. Customer logs in to system to make a reservation (1)
2. Customer selects an available plane (2)
3. Customer selects desired date and time for plane rental (3)
4. Customer selects an available instructor if required (4)
5. System records rental reservation and sends confirmation

Use Scenario: Customer manages Existing Rental Reservation

1. Customer logs in to manage an existing rental reservation (1)
2. Customer selects option to manage reservations (2)
3. Customer selects option to reschedule an existing reservation (3)
4. Customer selects new date/time for reservation
5. System confirms new reservation and removes old reservation from system

Use Scenario: Admin Manages a Rental Billing

1. Customer submits beginning and ending Hobbs time to Husky Air Admin after a successful plane rental (1)
2. Admin records times to calculate total price for plane rental (2)
3. Admin adds instructor fee (if applicable)
4. Admin generates final invoice for customer (3)
5. Customer receives invoice (4)
6. Customer pays invoice (5)
7. Admin records invoice as paid

Use Scenario: Existing Customer Manages Customer Account

1. Customer logs in to system (1)
2. Customer is presented with account profile page (2)
3. Customer selects profile information to update (3)