**Miles Acquisition System**

**Objective**

Miles Acquisition System (MAS) is a Web based application for a particular Airlines company. This application facilitates registering, updating, and utilization of miles for a Frequent Flier of its flights. The frequent flier should be able to login and key in all the details of his travel in their airlines. The application is accessible for the frequent fliers from the existing website of the airlines. This project mainly designed for frequent flyer belongs to particular airline service.

**Introduction**

Miles Acquisition System (MAS) is a Web based application for a particular Airlines company. And also this provide the information how earn points and a how to redeem points and how to utilize the points. This provides the information about flights and hotels of the particular airline services.

A **frequent flyer program** (FFP) is a [loyalty program](http://en.wikipedia.org/wiki/Loyalty_program) offered by many [airlines](http://en.wikipedia.org/wiki/Airlines). Typically, airline customers enrolled in the program accumulate frequent flyer miles (kilometers, points, segments) corresponding to the distance flown on that airline. Acquired miles can be redeemed for free air travel; for other goods or services.

**How to calculate Frequent Flyer points:**

The actual monetary value is typically 2 cents per frequent flyer mile (reportedly the airline industry average). For example, if you want to redeem 25,000 miles for a free ticket, the number of miles multiplied by 2 cents per mile is $500. As some industry researchers have noted, it may be smarter to save your miles and purchase a lower cost ticket to your destination.

You should always have your frequent flyer number handy when speaking with the [airlines](http://travel.howstuffworks.com/airline.htm). You purchase a ticket with miles the same way you do with [cash](http://money.howstuffworks.com/currency.htm). They will use your number to access your account and subtract the appropriate amount of miles as you purchase the ticket. Tickets are available in paper form via regular mail or shipping carriers, and, of course, as an **e-ticket**.

**Project Category**

Normally Project are having different types of categories

Like OOPS, Networking, Multimedia, Artificial Inteligence, RDBMS.

Social**-**networking project comes under OOPS category. In this project we are using purely object oriented programming language JAVA. Here we are using Servlets, jsp, JDBC, core java concepts to build this project.

The purpose of the Object-Oriented Programming Support (OOPS) Project is to provide a library as a framework for object-oriented programming in C++, java .

**Design Principles Behind OOPS**

The design of the OOPS Library is partly inspired from the Smalltalk and Java class hierarchies, and is based on the following general principles.

**Complexity:**

Classes should use the template mechanism, in order to overcome the duplication of data types with distinct parameters.

**Modularity:**

No class should depend on the internal details of any other class, in order to avoid using the friend mechanism.

**Factoring:**

Each independent object in the OOPS Library should appear in only one place.

**Classification:**

The OOPS Library must provide a means for adding new classes on equal footing with the classes of the library.

**SOFTWARE REQUIREMENTS**

Operating System : Windows XP/2003 or Linux

User Interface : HTML, CSS

Client-side Scripting : JavaScript

Programming Language : Java

Web Applications : JDBC, Servlets, JSP

IDE/Workbench : My Eclipse 8.6

Database : Oracle 10g

Server Deployment : Tomcat 6.x

**HARDWARE REQUIREMENTS**

Processor : core2duo

Hard Disk : 160GB

RAM : 1GB or more

**Problem Definition:**

Through the existing system process which is purely manual process and time taking process. There are chances to maintain separate profile for frequent flyers. They can’t provide information about points and how to use that points.

**About Existing System**

**Problem of the Current System**

* Existing system does not provide the secure registration.
* There is no chance to maintain separate profile for Frequent Flyers.
* Existing system can’t provide booking ticket directly in this system.
* Some airlines only provide this frequent flyer program.
* The existing system can’t provide information about how to get points.

The existing system can’t provide information about how to use points.

**Proposed System:**

* Proposed system provides the secure registration for the frequent flyer.
* And also the proposed system provides the facility to book a ticket directly no need to go other websites.
* Proposed system provides the information like how to get the points and how to use points.

The frequent flyer can ask any queries to admin regarding to using point.

**Functional Requirements:**

* Should provide a common platform where people of India can
  + Voice out violations, injustice, inhumanity, corruption happening in their vicinity
  + Endorse someone else’s concern and augment with more proofs, details etc.
  + Call for an online debate or discussion on certain topics of broad applicability
* Should be highly dynamic, with minimal static content as framework and maximum content created by site participants
* Should have the ability to tag social discipline violators using their UID
* The site can provide various measures for fellow citizens to rate, control and monitor the social responsibility of a given individual. For example, if you find your classmate did a good job at the college function arrangements, you can increase his service-orientation score by +1(rating). If you find your neighbor smoking in public in front of children, you can decrement his responsibility score by 1. When everyone assess & reward a person for his social actions this way, one can see a true scale of where he stands vis-à-vis where he thinks he really is. A traffic police giving ticket is no more just monetary. He can (as a user of this website) affect you driving sense score. If a police cop demanded for bribe, he may not be dismissed in system, but this website could show his corruption level as utterly bad.
* Authenticity for adding users is utmost important for such a website. Definitely one should not be allowed to have more than one profile.
* There should be a placeholder for dealing with disputing situations for the site as it is quite possible for recipients of negative reward to resist. The dispute resolution mechanism should not be like traditional court, it should again be based on social forum like multiple opinions, group actions etc.

**Scope of the Solution**

Miles Acquisition System (MAS) is a Web based application for a particular Airlines company. This application facilitates registering, updating, and utilization of miles for a Frequent Flier of its flights. The frequent flier should be able to login and key in all the details of his travel in their airlines. The application is accessible for the frequent fliers from the existing website of the airlines.

And also this provide the information how earn points and a how to redeem points and how to utilize the points. This provides the information about flights and hotels of the particular airline services.

**Data Flow Diagrams**

A graphical tool used to describe and analyze the moment of data through a system manual or automated including the process, stores of data, and delays in the system. Data Flow Diagrams are the central tool and the basis from which other components are developed. The transformation of data from input to output, through processes, may be described logically and independently of the physical components associated with the system. The DFD is also know as a data flow graph or a bubble chart.

DFDs are the model of the proposed system. They clearly should show the requirements on which the new system should be built. Later during design activity this is taken as the basis for drawing the system’s structure charts. The Basic Notation used to create a DFD’s are as follows:

**1. Dataflow:** Data move in a specific direction from an origin to a destination.

**2. Process:** People, procedures, or devices that use or produce (Transform) Data. The physical component is not identified.

**3. Source:** External sources or destination of data, which may be People, programs, organizations or other entities.

**4. Data Store:** Here data are stored or referenced by a process in the System.

**Context Level Data Flow Diagram**



**Level1 Data Flow Diagram For Frequent Flyer:**



**Level1 Data Flow Diagram for Administrator:**



**Level1 Data Flow Diagram for General User:**



**Authentication Data Flow Diagram:**



**Admin:**

**Context Diagram Level Data Flow Diagram**



**Frequent Flyer Context Level Data Flow Diagram Diagram**

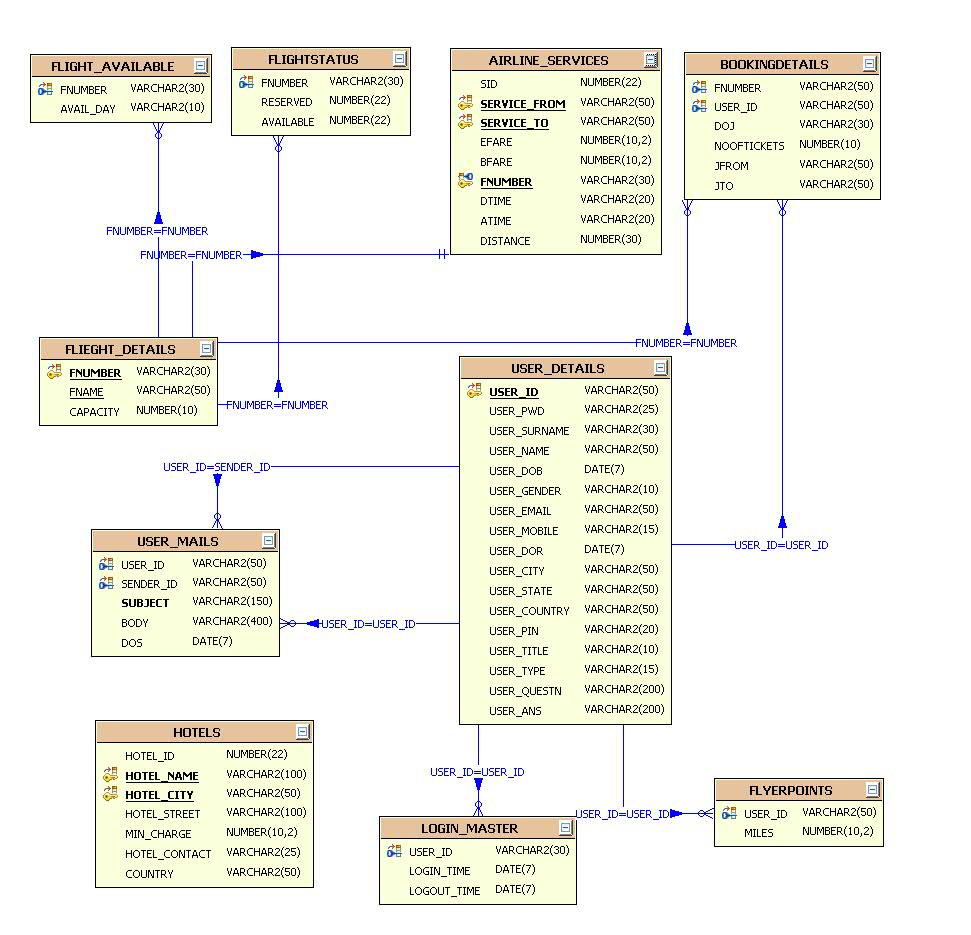


**Level2 Data Flow Diagram:**

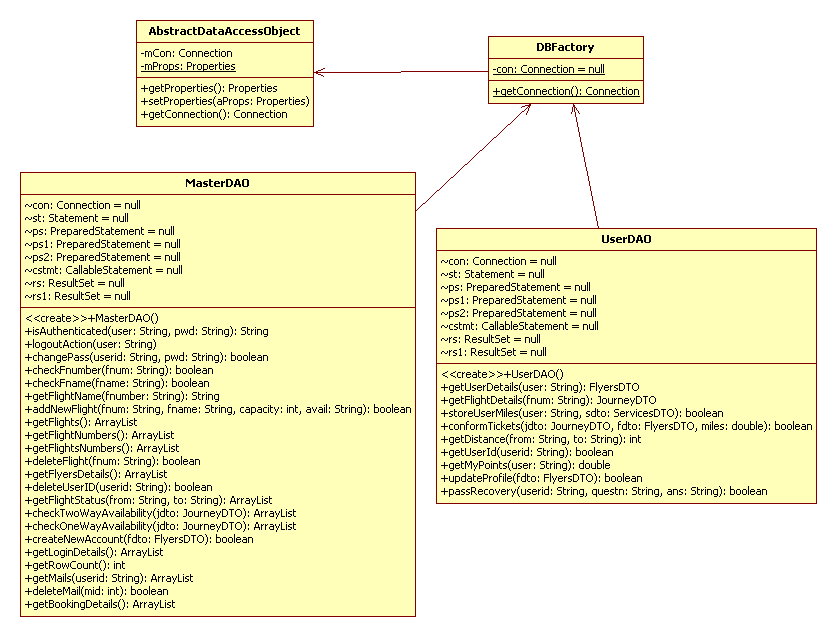


**Level3 Data Flow Diagram:**



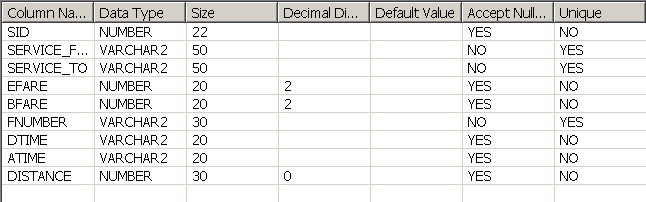
**ER Diagram**

CLASS DIAGRAM:

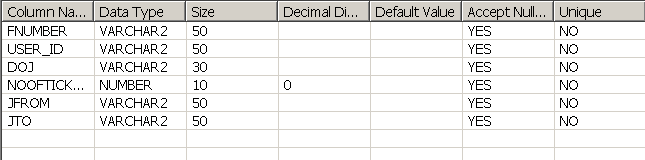


**Database Tables**

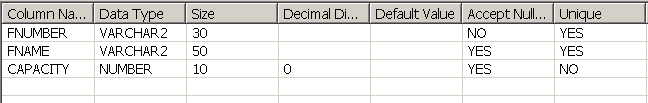
AIRLINE\_SERVICES



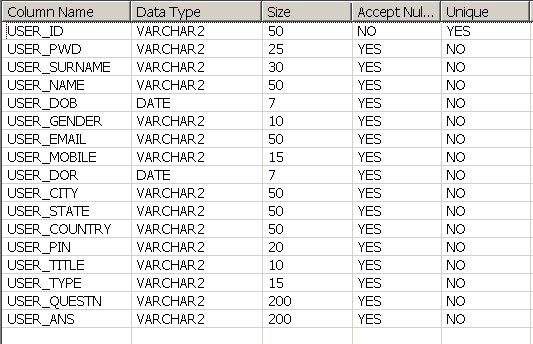
BOOKINGDETAILS



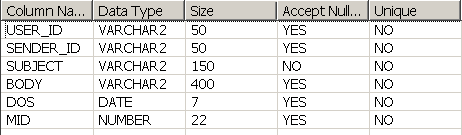
FLIEGHTDETAILS



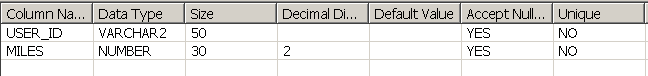
USERDETAILS



USERMAILS

****

FLYERPOINTS



**Modules**

The system after careful analysis has been identified to be presented with the following modules:

The Modules involved are

1. Authentication and Security Module
2. Admin Module
3. Flyer Module.
4. Flight Management Module.
5. Service Management Module.
6. Reports Module.

**MODULES DESCRIPTION**

Module description for each module mention on top

**Admin Module:**

In this administration module, admin should login to the system to access this system. After he logged Admin can add flight details and also he/she can modify the flight details belong to particular airline services. And also he/she has able to add services like hotels in different countries.

And admin can also monitor the users login and logout details.

And also he can suggest to the frequent flyer through mails about the points and services. The admin can communicate with the users of this system.

**Flight Management Module:**

In this module admin can add flight details and also he/she can modify and delete the flight details. This module serves the user like on which dates flights available and how many seats are available to book a ticket and also this module will help the frequent flyer to know the how many he/she needs to travel from one point to another point with free ticket.

**Frequent Flyer Module:**

In this module the flyer should register to access this system. Frequent flyer means, the person who travel frequently by air. The frequent flyer should login to the system, to utilize the benefits of the frequent flyer offered by particular airline services. This system maintains the complete details of the registered user and also its facilitate to update and modify the details of the user. In this module the user can book a ticket and see the status of the flight between two points. This system facilitates to know how many points the user need to travel from one place to another place with free ticket.

The frequent flyer of this system can send queries to the admin via mails and he can also ask about the accumulate miles.

**Security and authentication module:**

In this module this system stores the details of the registered user with safe and secure manner. And it can allow only authentication user to access the system. This system prevents the unauthorized accessing and malfunctioning.

**Service Management Module:**

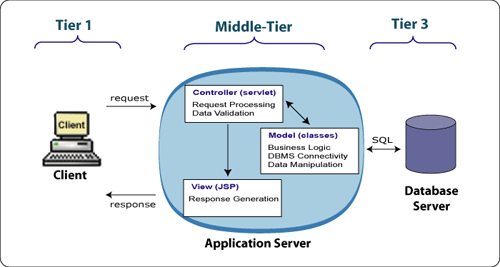
In this module the administration can add the services provided by the airline service and he can also modify the services .He has to add service from source to destination and distance between two points.

**Report Module:**

In this module the data from the database will be shown in tabular format as report to the users of this system.

**ARCHITECTURE**

USER VIEW CONTROLLER MODEL



1. **THE PRESENTATION LAYER**

Also called as the client layer comprises of components that are dedicated to presenting the data to the user. For example: Windows/Web Forms and buttons, edit boxes, Text boxes, labels, grids, etc.

1. **THE BUSINESS RULES LAYER**

This layer encapsulates the Business rules or the business logic of the encapsulations. To have a separate layer for business logic is of a great advantage. This is because any changes in Business Rules can be easily handled in this layer. As long as the interface between the layers remains the same, any changes to the functionality/processing logic in this layer can be made without impacting the others. A lot of client-server apps failed to implement successfully as changing the business logic was a painful process

1. **THE DATA ACCESS LAYER**

This layer comprises of components that help in accessing the Database. If used in the right way, this layer provides a level of abstraction for the database structures. Simply put changes made to the database, tables, etc do not affect the rest of the application because of the Data Access layer. The different application layers send the data requests to this layer and receive the response from this layer.

1. **THE DATABASE LAYER**

This layer comprises of the Database Components such as DB Files, Tables, Views, etc. The Actual database could be created using SQL Server, Oracle, Flat files, etc.   
In an n-tier application, the entire application can be implemented in such a way that it is independent of the actual Database. For instance, you could change the Database Location with minimal changes to Data Access Layer. The rest of the Application should remain unaffected.

**SDLC:**

This document play a vital role in the development of life cycle (SDLC) as it describes the complete requirement of the system. It means for use by developers and will be the basic during testing phase. Any changes made to the requirements in the future will have to go through formal change approval process.

SPIRAL MODEL was defined by Barry Boehm in his 1988 article, “A spiral Model of Software Development and Enhancement. This model was not the first model to discuss iterative development, but it was the first model to explain why the iteration models.

As originally envisioned, the iterations were typically 6 months to 2 years long. Each phase starts with a design goal and ends with a client reviewing the progress thus far. Analysis and engineering efforts are applied at each phase of the project, with an eye toward the end goal of the project.

The steps for Spiral Model can be generalized as follows:

* The new system requirements are defined in as much details as possible. This usually involves interviewing a number of users representing all the external or internal users and other aspects of the existing system.
* A preliminary design is created for the new system.
* A first prototype of the new system is constructed from the preliminary design. This is usually a scaled-down system, and represents an approximation of the characteristics of the final product.
* A second prototype is evolved by a fourfold procedure:

1. Evaluating the first prototype in terms of its strengths, weakness, and risks.
2. Defining the requirements of the second prototype.
3. Planning an designing the second prototype.
4. Constructing and testing the second prototype.

* At the customer option, the entire project can be aborted if the risk is deemed too great. Risk factors might involved development cost overruns, operating-cost miscalculation, or any other factor that could, in the customer’s judgment, result in a less-than-satisfactory final product.
* The existing prototype is evaluated in the same manner as was the previous prototype, and if necessary, another prototype is developed from it according to the fourfold procedure outlined above.
* The preceding steps are iterated until the customer is satisfied that the refined prototype represents the final product desired.
* The final system is constructed, based on the refined prototype.
* The final system is thoroughly evaluated and tested. Routine maintenance is carried on a continuing basis to prevent large scale failures and to minimize down time.

**The following diagram shows how a spiral model acts like:**



**LIMITATIONS AND SCOPE FOR FUTURE ENHANCEMENTS:**

**Limitations of the system:**.

* + System works in all platforms and its compatible environments.
  + Advanced techniques are not used to check the authorization.

**Future Enhancements:**

It is not possible to develop a system that makes all the requirements of the user. User requirements keep changing as the system is being used. Some of the future enhancements that can be done to this system are:

* As the technology emerges, it is possible to upgrade the system and can be adaptable to desired environment.
* Because it is based on object-oriented design, any further changes can be easily adaptable.
* Based on the future security issues, security can be improved using emerging technologies.
* Attendance module can be added
* sub admin module can be added

(1) Java Complete Reference by Herbert Shield

(2) Database Programming with JDBC and Java by George Reese

(3) Java and XML By Brett McLaughlin

(4) Wikipedia, URL: <http://www.wikipedia.org>.

(5) Answers.com, Online Dictionary, Encyclopedia and much more, URL: <http://www.answers.com>

(6) Google, URL: <http://www.google.co.in>

(7)Project Management URL: http://www.startwright.com/project.htm