Software Requirements Specification

for

Java Air

Prepared by Guoyu Qi, Steve Jia, Yuwei Cao

Team: Avian Limited

Author: Guoyu Qi, Steve Jia, Yuwei Cao

Table of Contents

Revision history

Name	Date	Changes Made	Version
Yuwei Cao, Guoyu Qi	2016-09-14	Initial draft; copied a template and modified its index to imitate the IEEE format.	1.0
Steve Jia	2016-09-23	Updated the table of contents to IEEE-830-1998 format; Rewrite sections 1.1 and 1.2; Added contents to section 1.3 and 1.4; Rewrite section 1.5 and section 2's intro; Rewrite section 2.1's intro; Added contents for section 2.1.3; Rewrite contents for sections 2.1.6 and 2.1.7; Added contents for section 2.1.8; Composed all contents for sections 2.2.1 to 2.2.15; Rewrite sections 2.3, 2.4, 2.5, and 2.6;	2.0
Steve Jia	2016-09-25	Updated section 3 titles to Object-Oriented IEEE format; Rewrite sections 3.1.1 and 3.1.2 Rewrite section 3.2 to organize specifications by classes/objects; Added contents to section 3.2 to include all forms and core classes needed for the project, contents are written as specific as possible; Revise section 3.3; Rewrite section 3.4; Removed template content from sections 3.5.1 and 3.5.4; added correct content to those two sections; Added content to section 3.5.2; Rewrite section 3.5.3;	3.0
Yuwei Cao, Guoyu Qi	2016-09-28	Added "Bags" class to section 3.2.8; Specified "Registered" and "Guest" Customers Entities in section 3.2.2.2; Added "One-way", "Round-trip" or "Multiple-cities" as Reservations Entities in section 3.2.4.2; Added "Customer Service", "Maintenance" and "Executive" as Employees Entities in section 3.2.5.2; Specified "Primary" and "Secondary" Passengers Entities in section 3.2.7.2; Revised format and spelling mistakes; Updated table of contents;	3.1

Yuwei Cao, Guoyu Qi	2016-10-01	Removed redundant use cases, i.e. "User Account Management" and "Customer Rewards Program" in section 2.2; Added step-by-step illustration to each use case in section 2.2; Added over-all use case figure in section 2.2;	3.2
Yuwei Cao, Guoyu Qi	2016-10-06	Reorganized attributes of each class in section 3; Revised *Flight Reservation* use case in section 2.2.6; Revised guest customer *Look-up Reservation* use case in section 2.2.7; Added *Reservation Status* as a Form entity in section 3.2.1.2.14; Revised *Reservation Cancellation* use case in section 2.2.11;	3.3
Yuwei Cao, Guoyu Qi	2016-10-15	Revised *Flight Reservation* use case steps in section 2.2.6, *Reservation Look-up* Use Case in section 2.2.7 and *Reservation Cancellation* Use Case 2.2.11; Revised attributes for *Reservations* class in section 3.2.4.1 and *Flight Status* attribute of *Flights* class in section 3.2.3.1; Added *Check Delay Time* function for *Forms* class in section 3.2.1.3.1; Revised definition of *Forms* entity *Customer Account Management: View Existing Reservations* in section 3.2.1.2.6 and *Forms* entity *Flight Reservation: Purchase Flights* in section 3.2.1.2.10; Revised definition of *Passengers* entity *Primary Passenger* in section 3.2.7.2.1; Removed "PIN" as an attribute of *Customers* class in section 3.2.2.1, and accordingly revised related use case including 2.2.3 *Password Reset* and 2.2.4 *Update Personal Information*; Removed *Bags* class, and accordingly revised related use case, i.e. *Employee Use Case of Flight Check-In* in section 2.2.8; Revised font, capitalization and writing pattern of form names;	3.4
Yuwei Cao, Guoyu Qi	2016-10-19	1. Recomposed section 2.1.3 to 2.1.5, revised other parts in section *2.1 Product Perspective*; 2. Added extensions under main success scenario in each use case in section *2.2 Product Functions*; 3. Specified how system treats locally stored data when shutoff access in *2.2.2 User Account Logout Use Case* and *2.2.10 Employee Logout Use Case*; 4. Revised *3.2.1.2.13 Flight Status* Form entity; 5. Revised *3.2.1.4.6 Cancelling a Reservation* Form event and accordingly added *Reservation Status* attribute in 3.2.4.1; 6. Specified format and accumulating rules of Reward-point in section *3.2.2.1 Attributes of Customers*; 7. Revised section *3.3. Performance Requirements*; 8. Added new section *3.4 Logical Database Requirements*;	4.0

		9. Removed ambiguity, inconsistence and incorrectness in this document.	
Yuwei Cao, Guoyu Qi	2016-10-27	1. Revised *3.6.2 Availability*. 2. Revised *3.4 Logical Database Requirements.* 3. Added Data Flow Diagrams and Database Diagram in *Appendix*. 4. Revised *Figure 2.2 User Account Registration use case for Java Air*.	4.1
Yuwei Cao, Guoyu Qi	2016-10-28	1. Updated Data Flow Diagrams and Database Diagram in *Appendix*.	4.2
Yuwei Cao, Guoyu Qi	2016-12-03	1. Final check, fixed some format and spelling mistakes;	4.2.2
Yuwei Cao, Guoyu Qi	2016-12-05	1. Updated test case nomenclature.	4.2.3

1. Introduction

1.1. Purpose

The main purpose of this document is to describe the end-user requirements for Java Air customer and employee systems. This document will also be used to guide the team members' development, the project, and communications. The document can also preliminary determine whether the project meets the requirement, and can be the start point of the software architect to design the software as well as the test engineers to design tests. This document will be subject to change, if more requirements are added to the project. This document is mainly prepared to set stage for the design phase of the project.

1.2. Scope

This document covers the requirements for release version 0.0.1 of the Java Air software product. Mention will be made throughout this document of selected probable features of future releases. The purpose of this is to guide the software architect in selecting a design that will be able to accommodate a full-scale application.

1.3.Definitions, Acronyms, and Abbreviations

CI = Configuration Item
CMMI = Capability Maturity Model Integration
IEEE = Institute of Electrical and Electronics Engineers
QA = Quality Assurance
SEI = Software Engineering Institute
SCMP = Software Configuration Management Plan
SPMP = Software Project Management Plan
SRS = Software Requirements Specification (this document)
SDD = Software Design Document
SQAP = Software Quality Assurance Plan
SVVP = Software Verification and Validation Plan
STP = Software Test Plan
UD = User Documentation
WBS = Work Breakdown Structure
U/PD = User/Product Director
PM = Project Manager
RE = Requirement Engineer
SA = Software Architect
IE = Integration Engineer
TE = Testing Engineer
CD = Code Developer
NYI = Not Yet Implemented

1.4. References

Software Engineering Modern Approaches, 2nd ed. By Eric J. Braude, Michael E. Bernstein.

Software Configuration Management Plan (SCMP) for Java Air version 1.0

Software Design Description (SDD) for Java Air version 1.0

Software Project Management Plan (SPMP) for Java Air version 1.0

Software Quality Assurance Plan (SQAP) for Java Air version 1.0

Software User Documentation Plan (SUDP) for Java Air version 1.0

Software Test Document (STD) for Java Air version 1.0

Software Verification and Validation Plan (SVVP) for Java Air version 1.0

1.5. Overview

Java Air software application is a full software product designed and developed for Java Air's customers and employees to simulate an airline's software system. For customers, the core functionality is to provide a flight reservation system. For employees, the core functionality is to provide customer service. The software will also provide other benefits to the customers and additional functionalities for the employees.

2. Overall Description

The Java Air software application will be designed for the use of Java Air customers and Java Air employees to simulate a full-scale airline software system. Customers will be able to reserve flights, manage accounts, check flight status, check into flights, and manage their Java Air rewards. Java Air customer service employees will be able to support the customers by doing all of the functions above, as well as additional functions. Java Air maintenance employees will be able to view a list of tasks they need to perform before and after each flight.

In the early versions of this software application, there will only be interfaces and functionalities for Java Air customers. Eventually, the application will feature the Java Air rewards program, and functionalities and features for Java Air employees.

The design should support expansion of a more-featured software application, including advanced flight-status systems and more functionalities for Java Air management employees.

2.1. Product Perspective

The Java Air software application is designed for a wide range of users, and is intended to optimize customer experience for flying by offering an easy to use interface and robust customer service functions.

2.1.1. System Interfaces

None.

2.1.2. User Interfaces

The user interface of Java Air shall have an appearance very roughly like that shown in Figure 2.1.

Figure 2.1 *Java Air GUI*

2.1.3. Hardware Interfaces

Java Air requires PC with 1 gigahertz (GHz) or faster 32-bit (x86) or 64-bit (x64) processor, 1 GB RAM (32-bit) or 2 GB RAM (64-bit), 16 GB available hard disk space (32-bit) or 20 GB (64-bit), DirectX 9 graphics device with WDDM 1.0 or higher driver, 2.0 or greater USB driver, a compatible optical drive, and Internet access.

A mouse and a keyboard are also required to interact with the Java Air software application.

2.1.4. Software Interfaces

Requires operating system Windows 7 or above with java installed.

2.1.5. Communication Interfaces

Requires an access to local database and public Internet, so as to enable java version update.

2.1.6. Memory Constraints

Java Air shall require no more than 120 MB of RAM and 500 MB of secondary storage (see test plan *TCS_JA_Storage_v1.0*).

2.1.7. Operations

Customer users will be required to supply their username and password in order to access their account information.

The employee users will be required to supply their employee ID number in order to access the employee features.

2.1.8. Site Adaptation Requirements

In earlier versions of the software application, Java must be installed on a computer before the application can be started on that computer. The development team will compile the software application into an executable file (.exe) so that the application can be run easily on Windows computers.

2.2. Product Functions

This section specifies the required overall functionality of the application, but is not intended to provide the complete specification. Section 3 of this document provides the requirements in complete detail.

2.2.1. New User Account Registration Use Case

Actor: Java Air Customer

Description: This use case is used for new customer to register on Java Air application.

Use Case: Figure 2.2 gives the text of the new user account registration use case.

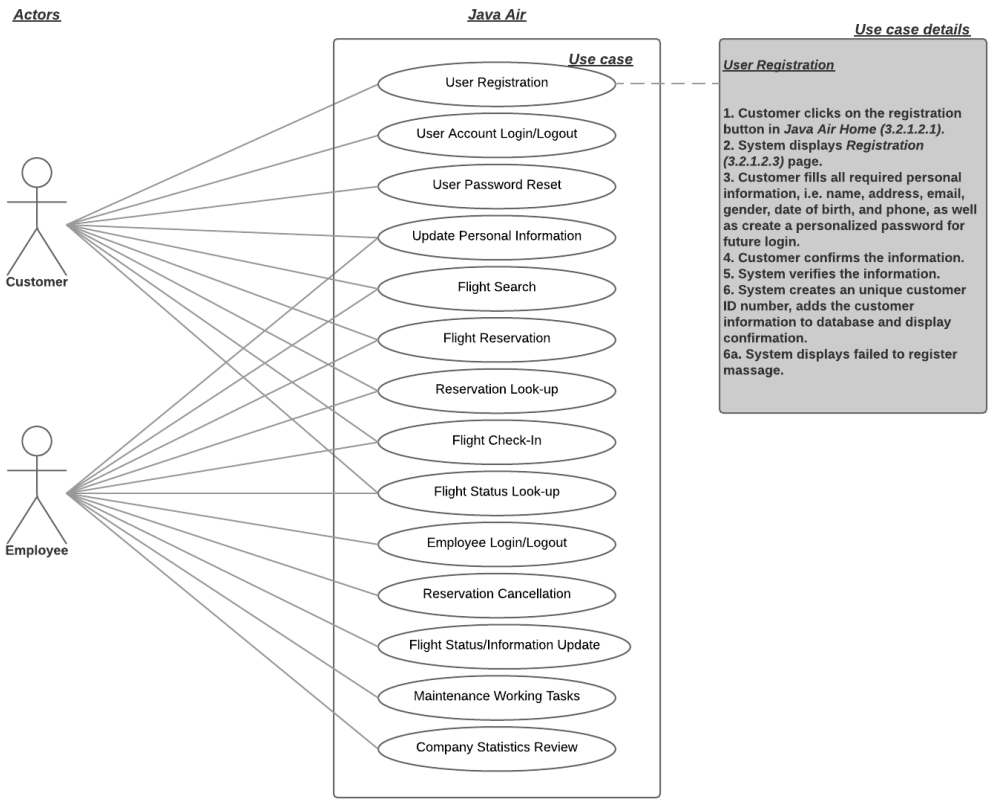


Figure 2.2 *User Account Registration use case for Java Air*

This use case corresponds to test < *TCS_JA_Registration_v1.1*> in the Software Test Documentation.

2.2.2. User Account Login/Logout Use Case

Actor: Java Air Customer

Description: A Java Air customer user must go to the *Landing (3.2.1.2.2)* page before they provide their username (e-mail address) and their password to access their account information. Once logged in, the user can then click on the "logout" button on the menu bar to close their account page. Once logged out, the user must perform the login steps again.

Login Use Case:
1. Customer provides username, i.e. e-mail address and password in *Landing (3.2.1.2.2)* page.
2. System verifies the username and password.
3. System displays *Welcome (3.2.1.2.4)* page.
3a.System displays failed to login message.

Logout Use Case:
1. Customer clicks on "logout" under login status

2. System shuts off access to customer's account.
3. System resets the locally stored date to default.
4. System returns to *Landing (3.2.1.2.2)* page.

This use case corresponds to test < *TCS_JA_User_Login_Logout_v1.1*> in the Software Test Documentation.

2.2.3. User Password Reset Use Case

Actor: Java Air Customer

Description: The registered user can reset his password in this use case.

Use Case:
1. Customer clicks on "reset password" button.
2. System displays *Reset Password (3.2.1.2.8)* page.
3. User input e-mail address, date of birth, customer ID, new password and confirm new password, then clicks on "Submit" button.
4. System updates the password and displays confirmation.
4a. System displays failed to reset password message.
5. System returns to the *Landing (3.2.1.2.2)* page.

This use case corresponds to test < *TCS_JA_PassReset_v1.1*> in the Software Test Documentation.

2.2.4. Update Personal Information

Actor: Java Air Customer / Customer Service Employee

Description: A registered user can look-up his personal information in his account. A customer service employee can also look up the personal information with certain pre-requisites.

Use Case of Customer:
1. Customer logs in.
2. Customer accesses to *Update Personal Information (3.2.1.2.5)* page.
3. System displays all the personal information, which are editable.
4. User modifies personal information then clicks on "update" button.
5. System verifies the information, updates database and displays confirmation.
5a. System displays failed to update message.
6. System then displays updated information.

Use Case of Employee:
1. Employee logs in.
2. Employee clicks on "Update Customer Information" button in *Landing Form (3.2.1.2.16)* page.
3. Employee enters the customer's identification information, i.e. name, date of birth and customer ID to access customer's information.
4. Employee updates account information and clicks on "Submit" button.

5. System verifies the information, update information to database and display the result.
5a. System displays failed to update message.

This use case corresponds to test < *TCS_JA_User_Update_v1.0*> in the Software Test Documentation.

2.2.5. Flight Search Use Case

Actor: Java Air Customer / Customer Service Employee

Description: User supplies all required information, i.e. trip type, destination/origin, departure/arrival date, and number of passengers before clicking on the "search" button to view a list of flights matching their search requirements.

Use Case of Customer:
1. Customer access to *Java Air Home (3.2.1.2.1)* page or *Search Flights (3.2.1.2.9)* page.
2. Customer fills in required information, i.e. trip type, destination/origin, departure/arrival/returning date, and number of passengers in flight-search input fields and click on "search".
3. System displays a list of matching flights.
3a. System displays failed search message.

Use Case of Employee:
Same to *Use Case of Customer*.

This use case corresponds to test < *TCS_JA_FlightSearch_v1.1*> in the Software Test Documentation.

2.2.6. Flight Reservation Use Case

Actor: Java Air Customer / Customer Service Employee

Description: After user has selected a flight (flights), he can book the flight. The user will need to provide passengers' information and payment information. To purchase a flight, the user must enter credit card info, billing info and (or) redeem their rewards. Amount paid by credit card will add reward points to the user's account. For guest customer, system will automatically collect registration information, creates him an account, and displays registration confirmation along with reservation confirmation.

Use Case of Customer:
1. Customer selects in flight-search outcomes departing flight(s) and/or returning flight and clicks on the "Purchase" button.
2. System displays *Purchase Flights (3.2.1.2.10)* page.

3. a) Guest customer provides passengers' information, i.e. name, gender and date of birth for each passenger and payment information, i.e. payment type, amount of points to redeem, credit card info, and billing info (including email, phone and address);
b) Registered customer under log-in status will has his information automatically fills the form;
c) Registered customer under log-out status can choose to log in using the quick log-in bar at the top of *Purchase Flights (3.2.1.2.10)* page to get automatically fill, or continue as a guest customer;
4. Customer clicks on "Confirm Purchase" button.
5. System verifies all the information, updates the database and displays a *Purchase Confirmation (3.2.1.2.11)* page, system also collects from the guest customer's payment info required registration info, creates an account for guest customer and displays a reservation confirmation with default password on it.
5a. System displays failed reservation message.

Use Case of Employee:
Same to *Use Case of Customer*.

This use case corresponds to test < *TCS_JA_FlightReservation_v1.1*> in the Software Test Documentation.

2.2.7. Reservation Look-up Use Case

Actor: Java Air Customer / Customer Service Employee

Description: Customers can look-up their flight reservation information as registered customer, guests, or a customer service employee can look up reservation information with certain pre-requisites. The reservation information will contain airport information, trip information, passengers' information, etc.

Use Case of Registered Customer:
1. Registered customer logs in and clicks on "Reservations" in the menu list of *Customer Account Management* pages.
2. System access to *View Existing Reservations (3.2.1.2.6)* page and displays existing reservations in a list.
3. Registered Customer clicks on specific reservation on the list.
4. System displays unfolds detail reservation information in *View Existing Reservations (3.2.1.2.6)* page.

Use Case of Guest Customer (Optional):
1. Guest Customer clicks on "reservation status" button in *Java Air Home (3.2.1.2.1)* page.
2. System displays *Reservation Status (3.2.1.2.14)* page.
3. Guest Customer fills in the required information for search, i.e. a combination of reservation number and passenger name then clicks on "search" button.
4. System displays detail reservation information.
4a. System displays failed search message.

Use Case of Employee:
Same to use case of guest customer or *Reservation Cancellation Use Case (2.2.11)* step1 to step4.

This use case corresponds to test < *TCS_JA_ReservationLookup_v1.0*> in the Software Test Documentation.

2.2.8. Flight Check-In Use Case

Actor: Java Air Customer / Customer Service Employee

Description: A user can check into a flight either from the customer interfaces or have a customer service user to check in for you. For guest customer users and customer service users, name and reservation number must be provided. For registered users, he must log into his account first. Customer service users will also have the ability to check bags.

Use Case of Customer:
1. Customer clicks on "check in" button in *Java Air Home (3.2.1.2.1)* page.
2. System checks log-in status and displays *View Existing Reservations (3.2.1.2.6)* page for logged-in registered customer, and displays *Check-In (3.2.1.2.12)* page for logged-out registered customer and guest customer.
3. a) Logged-in registered customer selects reservation and clicks on "Check-in" button; b) logged-out registered customer and guest customer fills in required information, i.e. reservation number, first name, and last name and clicks on "Check-in" button.
4. System verifies information.
5. System updates reservation status and displays confirmation.
5a. System displays failed operation message.

Use Case of Employee:
1. Employee logs in.
2. Employee click on "Customer Check-In" button in *Landing Form (3.2.1.2.16)* page.
3. System displays *Customer Check-In (3.2.1.2.20)* page.
4. Employees fills in required information, i.e. reservation number, first name, last name and number of bags, then clicks on "Check-in" button.
5. System verifies information and updates reservation status.
6. System displays confirmation page.

This use case corresponds to test < *TCS_JA_Flight_Checkin_v1.0*> in the Software Test Documentation.

2.2.9. Flight Status Look-up Use Case

Actor: Java Air Customer / Customer Service Employee

Description: A user can look up information regarding certain flights by accessing to Flight Status (3.2.1.2.13) page, on which flights information of the day, including flight ID, schedule and status of all flights are displayed.

Use Case of Customer:
1. Customer clicks on "flight status" button in *Java Air Home (3.2.1.2.1)* page.
2. System displays *Flight Status (3.2.1.2.13)* page.

Use Case of Employee:
Same to *Use Case of Customer*.

This use case corresponds to test < *TCS_JA_Status_v1.0*> in the Software Test Documentation.

2.2.10. Employee Login/Logout Use Case

Actor: Java Air Employee User

Description: A Java Air employee user can view interfaces of his function by entering his employee ID number at the employee login screen. A logout button will be used to log the user out, and the user will have to repeat the login process before granting access to employee functions.

Login Use Case:
1. Employee accesses to *Login (3.2.1.2.15)* page.
2. Employee inputs identification information, i.e. employee ID number then clicks on "login" button.
3. System verifies information and displays *Landing Form (3.2.1.2.16)* page, or *Maintenance Work List (3.2.1.2.22)* page, or *Executive/Accounting (3.2.1.2.23)* page depending on account type.
3a. System displays failed to login message.

Logout Use Case:
1. Employee clicks on "log out" button.
2. System shuts off access to employee's account.
3. System resets the locally stored date to default.
4. System returns to *Login (3.2.1.2.15)* page.

This use case corresponds to test < *TCS_JA_Employee_Login_Logout_v1.0*> in the Software Test Documentation.

2.2.11. Reservation Cancellation Use Case

Actor: Java Air Customer Service Employee

Description: A Java Air customer service employee can cancel reservations at the request of a customer. The customer must provide passenger name(s) and reservation number(s) for customer service employee to operate. The customer service employee will need to confirm the cancellation before a refund, of dollar value and/or miles, can be issued to the customer.

Use Case:
1. Employee clicks on "Cancel Existing Reservations" button in *Landing (3.2.1.2.16)* page.
2. System displays *Cancel Existing Reservations (3.2.1.2.19)* page.
3. Employee fills in required information, i.e. reservation number and primary passenger's name, then clicks on "search".
4. System displays reservation information.
5. Employee view reservation information and clicks on "cancel this reservation" button.
6. System verifies information.
7. System updates reservation status, refunds miles and dollar payment displays confirmation.
7a. System displays failed to update message.

This use case corresponds to test < *TCS_JA_ReservationCancel_v1.0*> in the Software Test Documentation.

2.2.12. Flight Status/Information Update Use Case

Actor: Java Air Customer Service Employee

Description: A Java Air customer service user can update flights information (on-time status, delays, cancellations, etc.). The updated information will show up on a customer user's reservation status interfaces. The customer service user will need to confirm the updates before the information is applied/stored.

Use Case:
1. Employee clicks on "Update Flight Status" button in *Landing Form (3.2.1.2.16)* page.
2. System displays *Update Flight Status (3.2.1.2.21)* page.
3. Employee fills in required information, i.e. flight number, departure date and time, delay-status (on-time, early, delayed, or cancelled) then clicks on "update status" button.
4. System verifies information.
5. System updates flight status in database displays confirmation.
5a. System displays failed to update message.

This use case corresponds to test < *TCS_JA_FlightUpdate_v1.0*> in the Software Test Documentation.

2.2.13. Maintenance Working Tasks Use Case

Actor: Java Air Flight Maintenance Employee

Description: A Java Air flight maintenance user can access a user interface to view a list of tasks that needs to be performed. Tasks will include but not limited to, fuel the aircraft, load/unload the luggage, check aircraft logs for issues, etc. Descriptions of the tasks will be provided on their user interface, and the user must complete all tasks before a flight can take off.

Use Case:
1. Employee logs in.
2. System verifies account type and displays *Maintenance Work List (3.2.1.2.22)* page.
3. Employee marks "performed" for each work item listed under specific flight then clicks on "confirm".
4. System verifies information.
5. System updates flight status in database, and displays confirmation.
5a. System displays failed to update message.

This use case corresponds to test < *TCS_JA_MaintenanceWork_v1.0*> in the Software Test Documentation.

2.2.14. Company Statistics Review Use Case

Actor: Java Air Executive Employee

This feature is optional. If the development team have enough time, then this feature will be implemented in a later version of the Java Air software application.

Description: A Java Air executive employee can view summary statistics regarding the company (total number of flights, number of users, how much money made, etc.)

Use Case:
1. Employee logs in.
2. System verifies account type and displays *Executive/Accounting (3.2.1.2.23)* page.

2.3. User characteristics

Users of the Java Air software application are general users who have basic understandings of computer operations and graphical user interface navigations. Users of this software application must be able to remember his username passwords, for customer users, or Java Air employee ID numbers, for employee users. Users should use this software application on Windows PCs and have Java installed on said PCs. Users should also have the ability to read a user manual written in English if needed.

2.4.Constraints

Java Air shall operate on PCs running Windows 7 or above version at a minimum speed of 1 GHz. Java shall be the implementation language.

2.5. Assumptions and Dependencies

The software application depends on Java Run-time Environment.

2.6.Apportioning of Requirements

The requirements described in Sections 1 and 2 of this document are referred to as "customer requirements" or "high-level requirements", those in Section 3 are referred to as "detailed requirements."

The requirements will be implemented in the following order: first the customer user functionalities will be implemented, this includes graphical interfaces, and then the employee user functionalities will be implemented. Essential requirements (flight reservation, customer accounts, rewards program) will be given a higher priority, and therefore be implemented first, and then rest of the requirements (listed in Section 2) will be implemented.

3. Specific Requirements

3.1. External Interface Requirements

3.1.1. User Interfaces

The Java Air software application has two main forms that takes up a section of the monitor screen. One form is for customer users and the other form is for employee users. Figure 2.1 shows a typical screen shot of the customer user interface. The initial start form, "home", is intended primarily for customer users, but the employee users will also use the home form to access their user interfaces.

For each main form, there will be a consistent title area and a menu area. The main forms will have a container area where other forms of specific functionality are displayed, one at a time. The entire set of interfaces is as follows:

a. A main form with a title bar and a menu bar with the rest of the forms blank.
b. One or more user interface for each user function, specified in *3.2.1.2 Form Entities*.
c. For the customer-account-management use case, an option bar and another container form will be displayed tiled.
d. For the flight-search use case, a search-input form and a search-result form will be displayed tiled.
e. A user login form for customer users. The form size is much smaller than the main form.
f. A user login form for employee users. The form size is much smaller than the main form.

One of the main forms will always be present on the monitor. When a user navigates to different area/functions, interfaces of type b, c, or d will reside inside of type a. When users perform the login use case, interfaces of type e or f and interface of type a through d are superimposed.

3.1.2. Hardware Interfaces

See *2.1.3 Hardware Interfaces*.

3.1.3. Software Interfaces

See *2.1.4 Software Interfaces*.

3.1.4. Communications Interfaces

See *2.1.5 Communication Interfaces*.

3.2. Classes / Objects

Categories for the Java Air software application sufficient for expressing the requirements are Forms, Customers, Reservations, Flights, Employees, Aircrafts and Passengers.

3.2.1 Forms

A form is an interface viewable on the monitor. All activities of Java Air take place in forms. Home, flight-search, user-account are examples of forms.

3.2.1.1 Attributes of Forms

Attributes	Format & value description
Form Name	Every Java Air application form will have a unique name consisting of 1 to 50 characters. Acceptable characters shall consist of a through z, and A through Z only.

Test plan < *TCS_JA_Forms_v1.0*>.

3.2.1.2 Form Entities

Every form should have a title information and the menu bar displayed. A login/logout button should be visible at all times.

3.2.1.2.1 Java Air Home (Essential)

This is the initial display for a new instance of the Java Air software application. This form will show the title text/picture, the menu bar, as well as a set of welcome information and shortcuts. This includes flight-search inputs, link to flight-status, reservation-status (optional) and check-in forms. Optionally, if the customer user is logged in, it will display the account's rewards information, or else it will display a "signup for rewards" button that links to the user registration form (*3.2.1.2.3 Registration*).

3.2.1.2.2 Customer Account: Landing (Essential)

This form shall be the landing form for user account access. By clicking on the "account" button on the menu bar, the user will be taken to this page. There will be two sections tiled on this form. Left-hand side will be the "log in" section, where customer users can enter their usernames (e-mails) and passwords. The other section, on the right-hand side, will be a new account registration button.

3.2.1.2.3 Customer Account: Registration (Essential)

This form shall be where the customer can sign up for a new account. To sign up for a new account, the user must enter first name, last name, address, gender, birthday, phone number, e-mail address, and password. After entering the required information, the user will click on the "submit" button on the form to send the information to the database.

3.2.1.2.4 Customer Account Management: Welcome (Essential)

This form should be displayed after a customer user has logged in. The welcome form should display a menu list, "option list", that shows "personal information", "reservations", and "rewards". The welcome form should be simple and shows a welcome message for the user by displaying "Hello

<name>". Optionally, the welcome page can also display upcoming reservation(s) and rewards summary information.

3.2.1.2.5 Customer Account Management: Update Personal Information (Essential)

This form should be displayed after a customer user has clicked on the "personal information" button on the "account management" option list. The option list should still be visible after form-load. The customer user's personal information should be displayed and editable on this form. At the bottom of this form, there should be an update button and confirm with the user regarding the information updates.

3.2.1.2.6 Customer Account Management: View Existing Reservations (Essential)

This form should be displayed after a customer user has clicked on the "reservations" button on the option list. The option list should still be visible after form-load. The customer user's reservations shall be displayed on the form in a list, a click on specific reservation on the list will unfolds information including reservation number, flight number(s), origin/destination, departure/arrival date and time, passenger detail, and an option for check-in.

3.2.1.2.7 Customer Account Management: View Rewards Program Information (Essential)

This form should be displayed after a customer user has clicked on the "rewards" button on the option list. The option list should still be visible after form-load. The customer user's current reward-point count and rewards-level shall be displayed on this form. Optionally, the rewards-level information is presented as a graph or image, and the customer user has the ability to view their miles earned breakdown report.

3.2.1.2.8 Customer Account Management: Reset Password (Essential)

This form is for customer users who forgot their password. Users must enter his e-mail address, date of birth, customer ID in order to type in a new password and to update it.

3.2.1.2.9 Flight Reservation: Search Flights (Essential)

This form is where a customer can view a list of flights. There should be two sections of this form. The left-hand side contains the search-input functions, and the right-hand side contains the search result. The user must enter trip-type, origin, destination, departure date, and number of passengers into the search-input fields before clicking on the search button. Returning date shall be required if the customer chose "round-trip" as his trip-type. After the search button is clicked, a list of matching flights will be displayed on the right-hand side of the form. Then the user can select a departing flight and/or returning flight and click the "purchase" button.

3.2.1.2.10 Flight Reservation: Purchase Flights (Essential)

This form is for purchasing flights. This form contains three parts, from top to the bottom are, “quick log-in / welcome” bar, passenger info field, and payment info field.
About “quick log-in / welcome” bar, for logged in customer, it shows welcome message plus brief account information, and his information will automatically fills passenger info field and payment info field; Otherwise, it shows welcome message plus account & password collection field, through which registered customer under log-out status can log-in to get an automatically fill.
About passenger info field, the customer is allowed to add more than 1 passengers in a single reservation and shall provide required passenger information, i.e. name, gender, and date of birth for each passenger.
About payment info field, the customer can chose to pay wholly or partially by redeem reword-point (see *Amount of points to redeem* in 3.2.4.1 for constrains), and shall provide credit card info, email, phone and billing address when credit card payment is involved.

3.2.1.2.11 Flight Reservation: Purchase Confirmation (Essential)

This form is a confirmation form intended to be displayed after a successful reservation purchase. The confirmation form should display the flight details, passenger information, purchase dollar or points amount, remaining reward point and a unique reservation number.

3.2.1.2.12 Flight Reservation: Check-In (Essential)

This form is intended for guest customers to check into their flight(s). The form requires reservation ID, first name, and last name to be entered before passengers are checked-in.

3.2.1.2.13 Flight Status (Essential)

This form is intended for users to view flight status information (on-time, delayed, cancelled, actual depart time, actual arrival time, etc.). This form shall display flights information of the day, including flight ID, schedule and status of all flights.

3.2.1.2.14 Reservation Status (Optional)

This form is intended for guest customers to view reservation information and status (valid, cancelled, etc.). The form requires a combination of reservation ID and passenger name in order to display the search result.

3.2.1.2.15 Employee: Login (Essential)

This form will have a title of “Java Air Employee Login”. It will show an input box where an employee user can enter his employee ID number and click the “login” button to show a specific employee form (section 3.2.1.2.14, 3.2.1.2.20, or 3.2.1.2.21).

3.2.1.2.16 Customer Service: Landing Form (Essential)

This form is the initial display for a Java Air customer service user. This form will show a button for each customer service form. All the buttons will be laid out in a grid. The user can click on different buttons to access different forms with the ability to go back to this landing form.

3.2.1.2.17 Customer Service: Update Customer Information (Essential)

This form is for a customer service user to use to update account information at the request of a customer user. The customer service user must enter a customer's information (name, date of birth, and customer ID) before he can update it. This form will be very similar to that of Section 3.2.1.2.5.

3.2.1.2.18 Customer Service: Make Reservations for Customers (Essential)

This form is for a customer service user to use to make reservations at the request of a customer user. This form will be exactly the same to that of Sections 3.2.1.2.8, .9 and .10;

3.2.1.2.19 Customer Service: Cancel Existing Reservations (Essential)

This form is for customer service users to cancel an existing reservation at the request of a customer user. The customer service user must have the reservation number and primary passenger's name before cancelling a reservation. The customer user will not have this function. Only the customer service user can access this form.

3.2.1.2.20 Customer Service: Customer Check-In (Essential, NYI)

This form is for customer service users to check in passengers of an existing reservation at the requests of customer users. The form will be similar to that of Section 3.2.1.2.11. An additional function that the customer service user will have on this form is to check in bags and specify the number of checked-bags for the reservation that's being checked-in.

3.2.1.2.21 Customer Service: Update Flight Status (Optional; NYI)

This form is for customer service users to update delay information regarding a particular flight. The user will enter the flight number and departure date and the delay-status (on-time, early, delayed, or cancelled).

3.2.1.2.22 Maintenance Work List (Essential)

This form is for Java Air maintenance workers to use before a flight-takeoff or after a flight has landed. The form will show a list of tasks to be "performed", and the user must check-off a task for it to be "completed/finished".

3.2.1.2.23 Executive/Accounting Form (Optional; NYI)

This form is for Java executive/accounting users to view summary information regarding Java Air. Information displayed on this form can be total number of aircrafts, airports serviced, total money made, etc.

3.2.1.3 Form Functionality

3.2.1.3.1 Check Delay Time (Optional; NYI)

System will check Maintenance Work List (3.2.1.2.22) at Actual Departure Time (3.2.3.1), if Maintenance Work List (3.2.1.2.22) hasn't been finished by then, extra delayed time will be generated and added to Actual Departure Time (3.2.3.1), thus may cause a change on Flight Status (3.2.3.1). In this case, Actual Departure Time (3.2.3.1) will be same to the time when Maintenance Work List (3.2.1.2.22) been finished.

3.2.1.4 Events Pertaining to Forms

3.2.1.4.1 Record New User Information

When the user clicks on the register button, database queries will update the database by inserting a new record in the "customers" database table. The database will also generate a unique 6-digit account number and initialize the reward point count for this new account.

3.2.1.4.2 Update Existing User Information

When the user clicks on the update button, database queries will update the "customers" database table with information from the form to the correct record associated with the account number.

3.2.1.4.3 Obtaining a List of Flights

When the user clicks on the search button, database queries and code-base will return a list of "flight" objects matching the user's search criteria.

3.2.1.4.4 Making a New Reservation

When the user purchases flight(s), database queries will update the "reservations" database table by inserting a new record with the proper information. The database will generate a new unique reservation number to associate with this new reservation.

3.2.1.4.5 Update User's Reward Points

After a user successfully made a reservation, database queries will update the "customers" database table by adding or removing values to the "current reward points" field and adding values to the "total reward points" field is the purchase is made with a credit card.

3.2.1.4.6 Cancelling a Reservation

After customer service user clicks on the "cancel reservation" button, database queries will update the "reservation" database table by setting the reservation status to "canceled" and update reward points by giving the customer user a refund. Database queries will not remove the record from the "reservations" database table.

3.2.1.4.7 Updating a Flight's Status

After a user clicks on the update button, database queries will update the "flights" table by updating the "status" field with the corresponding flight-number and departure date.

3.2.1.4.8 Completing a Maintenance Task

After a user clicks on the complete button, database queries will update the "tasks" database table by updating the "status" field with the corresponding flight number and departure date.

3.2.2 Customers

3.2.2.1 Attributes of Customers

All Java Air customers will have the same set of attributes. Each attribute will be a string value consists of a combination or combinations of a through z, A through Z, 0 through 9, and symbols.

Attributes	**Format & value description**
Customer ID	Every customer of Java Air must have a unique customer ID. The ID shall contains a fixed prefix "JAC", followed by a 4-digit number.
Customer Name	Every customer of Java Air must have a name. A name shall consist of a first name and a last name. Names should only consist of a through z, A through Z, and blank spaces.
Address	Including address line 1(to specify street address, P.O. box, company name, c/o), address line 2(to specify apartment, suite, unit, building, floor, etc.), city, state/province/region, ZIP, and country
Gender	"male" or "female", Default as "male"
Date of Birth	"YYYY-MM-DD", for example, 2016-09-25.
E-mail Address	A combination of a through z, A through Z, 0 through 9, and ".", "_", "@".
Account Password	At least 6 characters.
Phone	10-digit number.
Current Reward-point Count	Reward-point shall be a 6-digit number equivalent to currency, initialized to 0.00 for new customers. Each dollar payment will add reward-point to customer's account at the rate of 100:1, for example, a $300 credit card payment will add to customer's account 3.00 reward-point. Reward-point earned from a reservation can be used for redeem only after this reservation is fulfilled, and will be automatically
Total Reward-point Count	

	deducted from customer's account if this reservation is cancelled.

Test plan < *TCS_JA_Attr_Cust_v1.0*>.

3.2.2.2 Customer Entities

A Customer is a user of Java Air front-end functionalities. A Customer do not necessarily have to be a passenger and vice versa. Java Air customers can be a registered customer or a guest customer.

3.2.2.2.1 Registered Customer (Essential)

A *Registered Customer* is a customer who has a Java Air account and with overall personal information recorded in Java Air database. A Registered Customer has a personalized e-mail and password combination used for identification when accessing *User Account Management, User Password Reset, Customer Rewards Program,* and *Flight Reservation* function.

3.2.2.2.2 Guest Customer (Essential)

A *Guest Customer* is a customer without a personalized e-mail and password combination. A guest customer cannot access to functions including *User Account Management, User Password Reset* and *Customer Rewards Program,* while will be required to provide personal information and automatically converted to a *Registered Customer* when using Flight Reservation function.

3.2.2.3 Functionality of Customers

All customer related functions must be performed through different entities of the Forms.

3.2.3 Flights

3.2.3.1 Attributes of Flights

All Java Air flights will have the same set of attributes. Each attribute will be a string value consists of a combination or combinations of a through z, A through Z, 0 through 9, and symbols.

A combination of flight-name, origin airport, and scheduled departure date shall generate a unique flight.

Attributes	**Format & value description**
Flight Number	Every flight of Java Air must have a unique ID, with fixed prefix "JAF", followed by a unique 4-digit number consist of 0 through 9.
Aircraft Name	See *3.2.6.1 Attributes of Aircrafts.*
Origin Airport	All airports will have a format of "City name, State name abbreviation (IATA code)", for example, Chicago, IL (ORD)
Destination Airport	
Scheduled Departure Date	All dates will have a format of "YYYY-MM-

Scheduled Departure Time Scheduled Arrival Date Scheduled Arrival Time Actual Departure Date Actual Departure Time Actual Arrival Date Actual Arrival Time	DD", for example, 2016-09-25. All times will have a 24-hour format of "hh:mm", for example, 17:30. Departure date and time shall be automatically generated for recurring flights, and arrival date shall be automatically generated by dividing the flight mileage by a pre-defined speeds of aircrafts.
Trip Mileage	A 6-digit number consist of 0 through 9, with "miles" as units.
Flight Status	A comparison between *Scheduled Departure Time* and *Actual Departure Time*, indicated by "Early", "On-time", "Delayed" or "Canceled".
Number of Checked Bags	Number of checked bags for this flight. Initialized to 0 and may be updated during customer check-in.

Test plan < *TCS_JA_Attr_Flight_v1.0*>.

3.2.3.2 Flights Entities

There will be only one type of flights.

3.2.3.3 Functionality of Flights

3.2.3.3.1 Generating Travel Time Adjustments

A random travel time adjustment of -30 to 120 minutes will be generated for each flight during departure and during in-flight. When a non-zero adjustment occurs, actual departure date time, actual arrival date time, and flight status will be updated in the database.

3.2.4 Reservations

3.2.4.1 Attributes of Reservations

All Java Air reservations will have the same set of attributes. Each attribute will be a string value consists of a combination or combinations of a through z, A through Z, 0 through 9, and symbols.

Attributes	**Format & value description**
Reservation Number	Every Java Air reservation must have a unique ID, with a fixed prefix "JAR", followed by a unique 8-digit number consist of 0 through 9.
List of Passengers	Same format as customer name. Each reservation shall have one Primary Passenger and 0 to 9 secondary passengers, see 3.2.7.2
Departure Flight	Same format as flight ID.

Returning Flight	Same format as flight ID. Returning flight can be optional if the trip is one-way.
Number of Checked Bags	Number of checked bags for this reservation. Initialized to 0 and may be updated during customer check-in.
Payment Type	"Credit card" or "Reward-point".
Amount of Points to Redeem	Same format as specified in *3.2.2.1 Attributes of Customers*. This number shall be no larger than customer's current reward-point count or the price of the ticket(s).
Amount Paid	A 6-digit number consist of 0 through 9 with a"$" symbol as the prefix.
Billing Information	Including email, same format as *Customer E-mail Address (3.2.2.1)*, phone, same format as *Customer Phone (3.2.2.1)* and billing address, same format as *Customer Address (3.2.2.1)*.
Reservation Status	Normal/Canceled

Test plan < *TCS_JA_Attr_Reservation_v1.0*>.

3.2.4.2 Reservations Entities

Java Air reservations can be a One-way Reservation, Round-trip Reservation or Multiple-cities Reservation.

3.2.4.2.1 One-way Reservation (Essential)

A One-way Reservation do not has a returning flight.

3.2.4.2.2 Round-trip Reservation (Essential)

A Round-trip Reservation has one departure flight and one returning flight, the origin and destination airport of the returning flight shall be exchanged ones of the departure flight, while the scheduled departure time of returning flight shall be no early than scheduled arrival time of the departure flight.

3.2.4.2.3 Multiple-cities Reservation (Optional; NYI)

A Multiple-cities Reservation has a series of departure flight and no returning flight, all departure flights shall be in a sequence both in space and in time.

3.2.4.3 Functionality of Reservations

None

3.2.5 Employees

3.2.5.1 Attributes of Employees

All Java Air employees will have the same set of attributes. Each attribute will be a string value consists of a combination or combinations of a through z, A through Z, 0 through 9, and symbols.

Attributes	Format & value description
Employee Name	Every Java Air employee must have a name. Same format as customer name.
Employee ID	Every employee will have a fixed prefix "JAE", followed by a unique 4-digit number consist of 0 through 9.
Role	"Customer service" or "Maintenance" or "Executive".
Starting Date	Same format as flight date.
Salary	A 6-digit number consist of 0 through 9 with a"$" symbol as the prefix.
Address	Same format as customer address.
Date of Birth	Same format as customer date of birth.
Gender	Same format as customer gender.
E-mail	Same format as customer e-mail.

3.2.5.2 Employees Entities

Employee entities will be defined by their role. Roles for Java Air employees are Customer Service, Maintenance and Executive. Each role has different job functions and can access different forms.

3.2.5.2.1 Customer Service Employee (Essential)

A Customer Service Employee can access to Employee Login/Logout, User Account Management, Flight Search, Flight Reservation, Customer Rewards Program, Reservation Look-up, Flight Check-In, Flight Status Look-up, Reservation Cancellation, and Flight Status/Information Update function.

3.2.5.2.2 Maintenance Employee (Essential)

A Maintenance Employee can access to Employee Login/Logout and Maintenance Working Tasks function.

3.2.5.2.3 Executive Employee (Essential)

An Executive Employee can access to Employee Login/Logout and Company Statistics Review function.

3.2.5.3 Functionality of Employees

None

3.2.6 Aircrafts

3.2.6.1 Attributes of Aircrafts

All Java Air aircrafts will have the same set of attributes. Each attribute will be a string value consists of a combination or combinations of a through z, A through Z, 0 through 9, and symbols.

Attributes	Format & value description
Aircraft	Every Java Air aircraft will have a unique

Name	name, with a fixed prefix "JAC" followed by a 4-digit number (0 through 9).
Passenger Capacity	Large aircraft will have a passenger capacity of 200; medium aircraft 120; small craft 60;
Flight Range	Large aircraft will have a flight-range of 6000 miles; medium aircraft 2500 miles; small craft 800 miles;
Speed	Large aircraft will have a cruising speed of 500 miles/hour; medium aircraft 200 miles/hour; small craft 120 miles/hour;
Number in Fleet	A 3-digit number.
Purchase Price	Large aircraft $50,000,000; medium aircraft $18,000,000; small craft $4,000,000;

Test plan < *TCS_JA_Attr_Aircraft_v1.0*>.

3.2.6.2 Aircraft Entities

There shall be three types of Java Air aircrafts.

3.2.6.2.1 Large Aircrafts (Essential)

Large aircrafts will have a passenger capacity of 200, a flight-range of 6000 miles, and a cruising speed of 500 miles per hour. The cost of a large aircraft is $50 million.

3.2.6.2.2 Medium Aircrafts (Essential)

Medium aircrafts will have a passenger capacity of 120, a flight-range of 2500 miles, and a cruising speed of 200 miles per hour. The cost of a medium aircraft is $18 million.

3.2.6.2.3 Small Aircrafts (Essential)

Small aircrafts will have a passenger capacity of 60, a flight-range of 800 miles, and a cruising speed of 150 miles per hour. The cost of a small aircraft is $4 million.

3.2.6.3 Functionality of Aircrafts

None

3.2.7 Passengers

3.2.7.1 Attributes of Passengers

All Java Air passengers will have the same set of attributes. Each attribute will be a string value consists of a combination or combinations of a through z, A through Z, 0 through 9, and symbols.

Attributes	**Format & value description**
Passenger Name	Every Java Air passenger must have a name. Same format as customer name.
Date of Birth	Same format as customer date of birth.

Gender	Same format as customer gender.
Phone	Same format as customer phone.
Primary Passenger Status	"Yes" or "No".

Test plan < *TCS_JA_Attr_Passenger_v1.0*>.

3.2.7.2 Passenger Entities

Java Air passengers can be a Primary Passenger or Secondary Passenger. The purpose of having a primary passenger is for customer service employee to conveniently look-up reservation(s).

3.2.7.2.1 Primary Passenger (Essential)

A primary passenger will be the first name to appear on the reservation. Typically, the account holder will be set as primary passenger by the system during flight booking. In the case when the account holder is not a passenger, system shall choose the first passenger on the *list of passengers (3.2.4.1)* as primary passenger. If the primary passenger's reservation is canceled, system shall choose and set a new primary passenger according to rules above.

3.2.7.2.2 Secondary Passenger (Essential)

Passenger who shares the same reservation ID with the primary passenger are considered as secondary passenger.

3.2.7.3 Functionality of Passengers

None

3.3.Performance Requirements

Java Air software application instance shall load and display the "home" form in less than 10 seconds. The loading and display of an initial employee function form shall take less than 10 seconds.

Responses to queries shall take no longer than 10 seconds to load onto the screen or to update the database tables after the user submits the query.

The system shall display confirmation messages to users within 4 seconds after the user submits information to the system.

3.4. Logical Database Requirements

The logical requirements for Java Air database placement are specified from following aspects:

a) Types of information used by various functions: Specified in sections 3.2;

b) Frequency of use: Functional depends. The database will be used whenever required by any function.
c) Accessing capabilities: Both customer and employee accessibilities are limited to predefined functions, see section *2.2 Product Functions* for specification.
d) Data entities and their relationships: Specified in *Appendix ii Database Diagram*;
e) Integrity constraints: Primary keys in the database including CustomerID, ReservationID, EmployeeID, FlightID and AirportID;
f) Data retention requirements: Data are stored locally and permanently unless menu intervention.

3.5. Design Constraints

Java Air shall be designed using UML and object-oriented design. It shall be implemented in Java. The software shall run as a Java application on Windows 7 or newer operating systems.

3.6. Software System Attributes

3.6.1. Reliability

The Java Air software application must meet the following reliability requirements:
a. Customer registration shall fail not more than once in every 100 registrations.
b. Customer account updates shall fail not more than once in every 1000 updates.
c. Flight searching shall fail not more than once in every 1000 searches. Test
d. Flight reservation shall fail not more than once in every 500 reservations. Test

3.6.2. Availability

Java Air shall be available to run on any PC running Windows 7, with other applications running simultaneously.

3.6.3. Security

Java Air shall require user log-in information before presenting account-specific information. The application must obtain a user's correct identification information, i.e. email, DOB, customer ID before changing the account's password.

Java Air must have a correct employee ID number before presenting the employee-specific information.

3.6.4. Maintainability

3.6.4.1 Changing Forms (Essential)

Developer must have the latest version of the code-base and the team-specified IDE before changing interfaces and forms. Modifying the forms shall be straightforward.

3.6.4.2 Adding and Removing Flights (Optional)

Developer must have access to the database before adding or removing flights. Modifying the list of flights shall be easy.

3.6.4.3 Adding and Removing Airports (Optional)

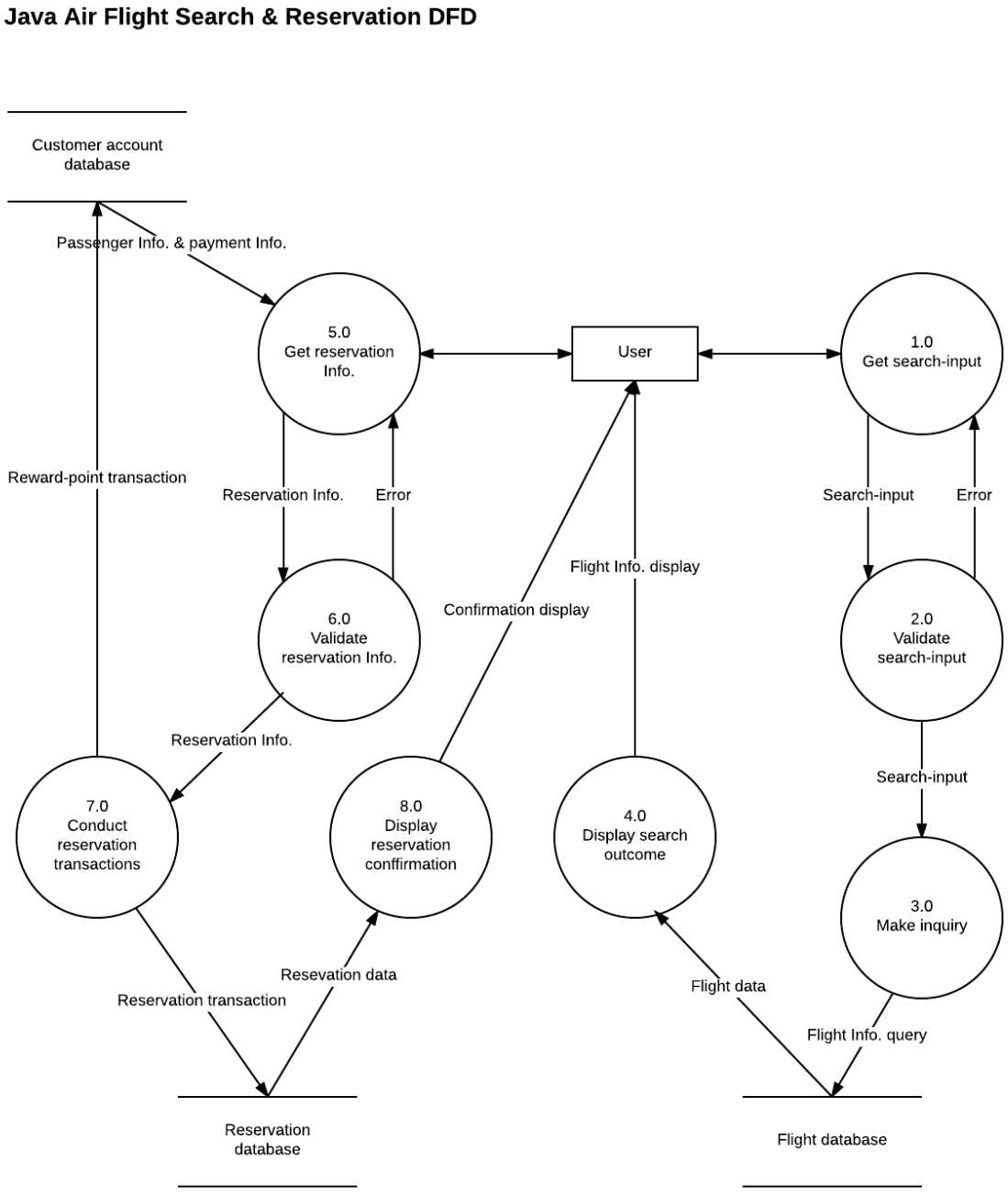
Developer must have access to the database before adding or removing airports. Modifying the list of airports shall be easy.

3.7. Other Requirements

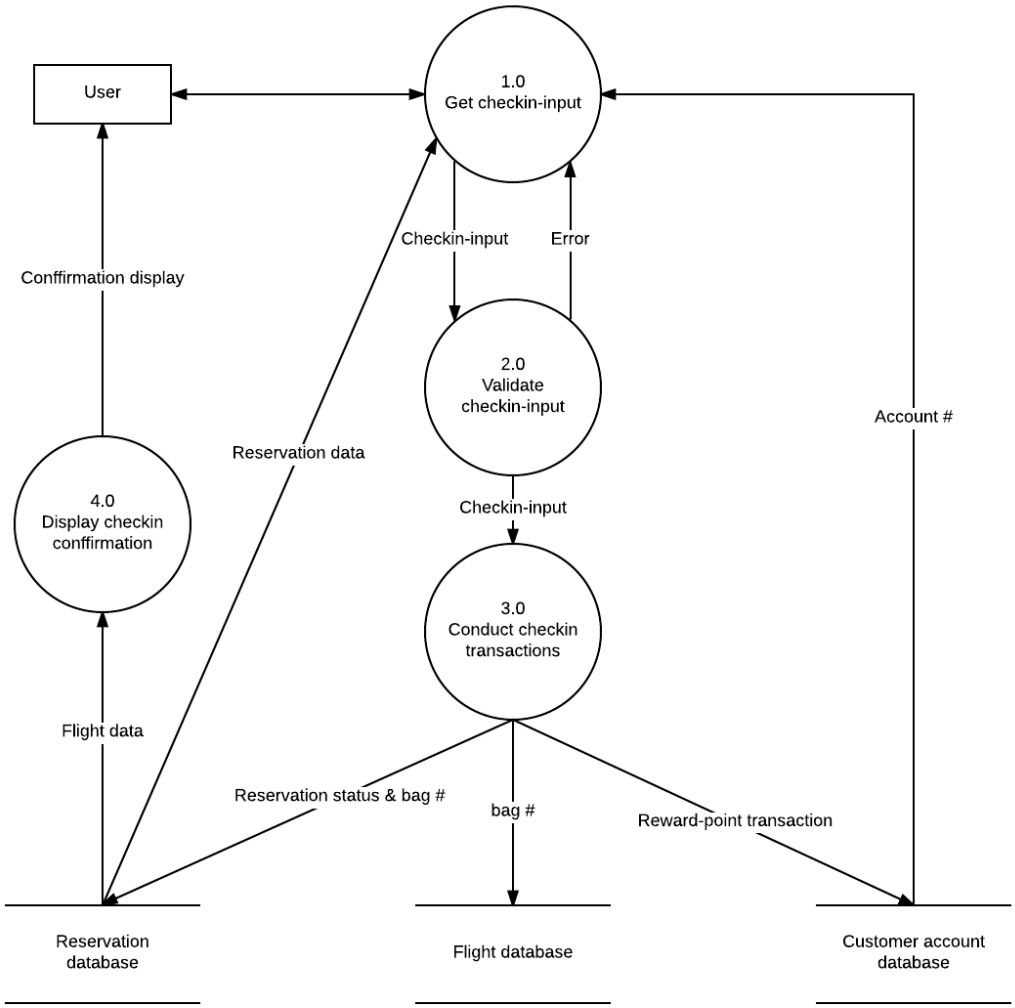
None

Appendix

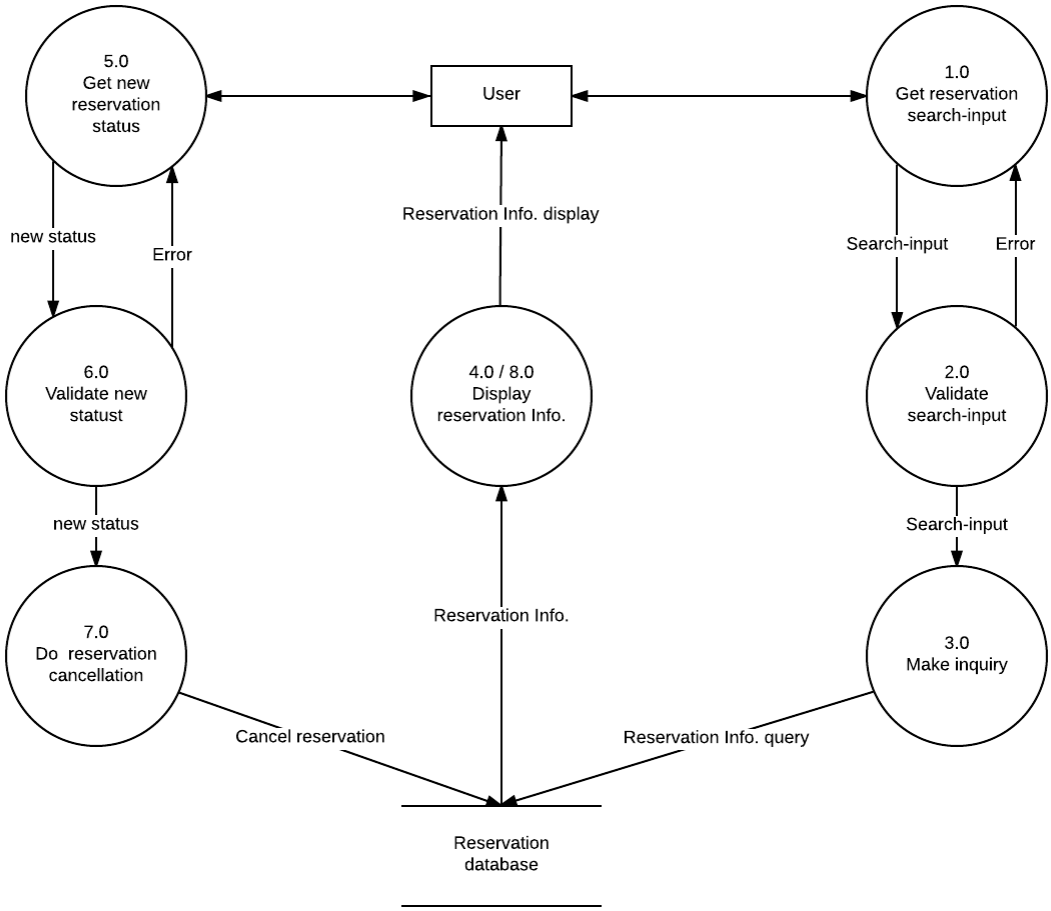
i. Data Flow Diagrams



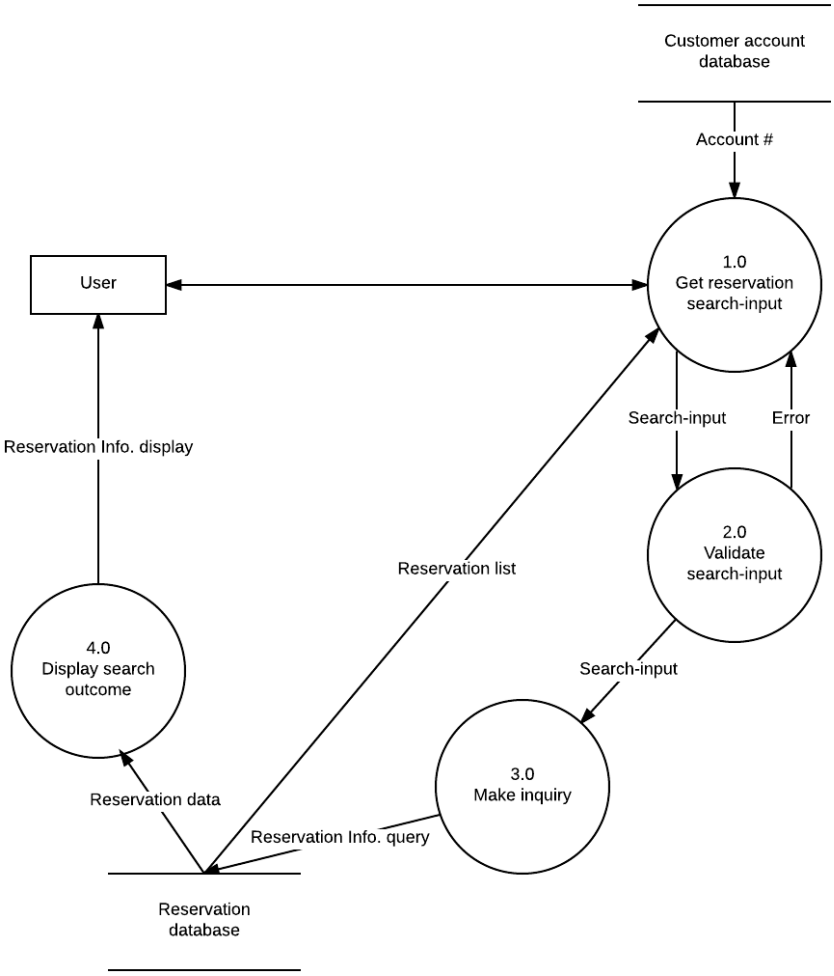
Java Air Flight Check-In DFD



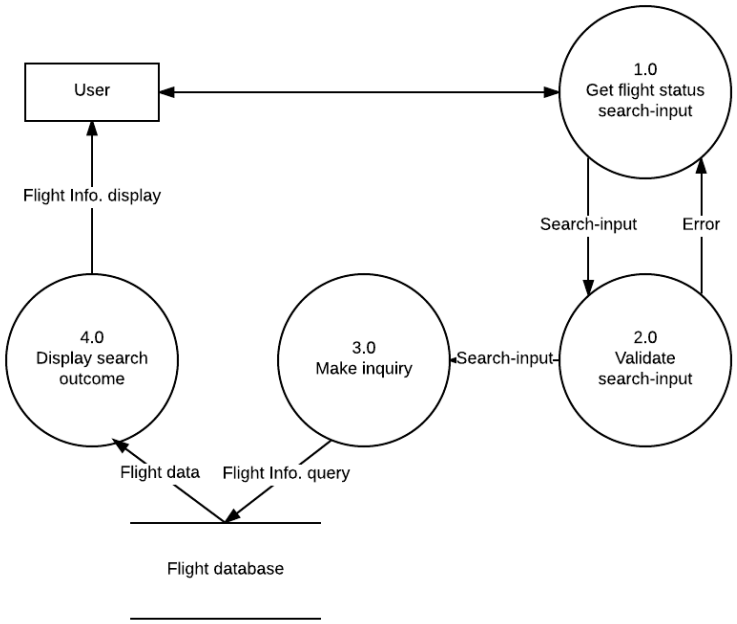
Java Air Reservation Cancellation DFD



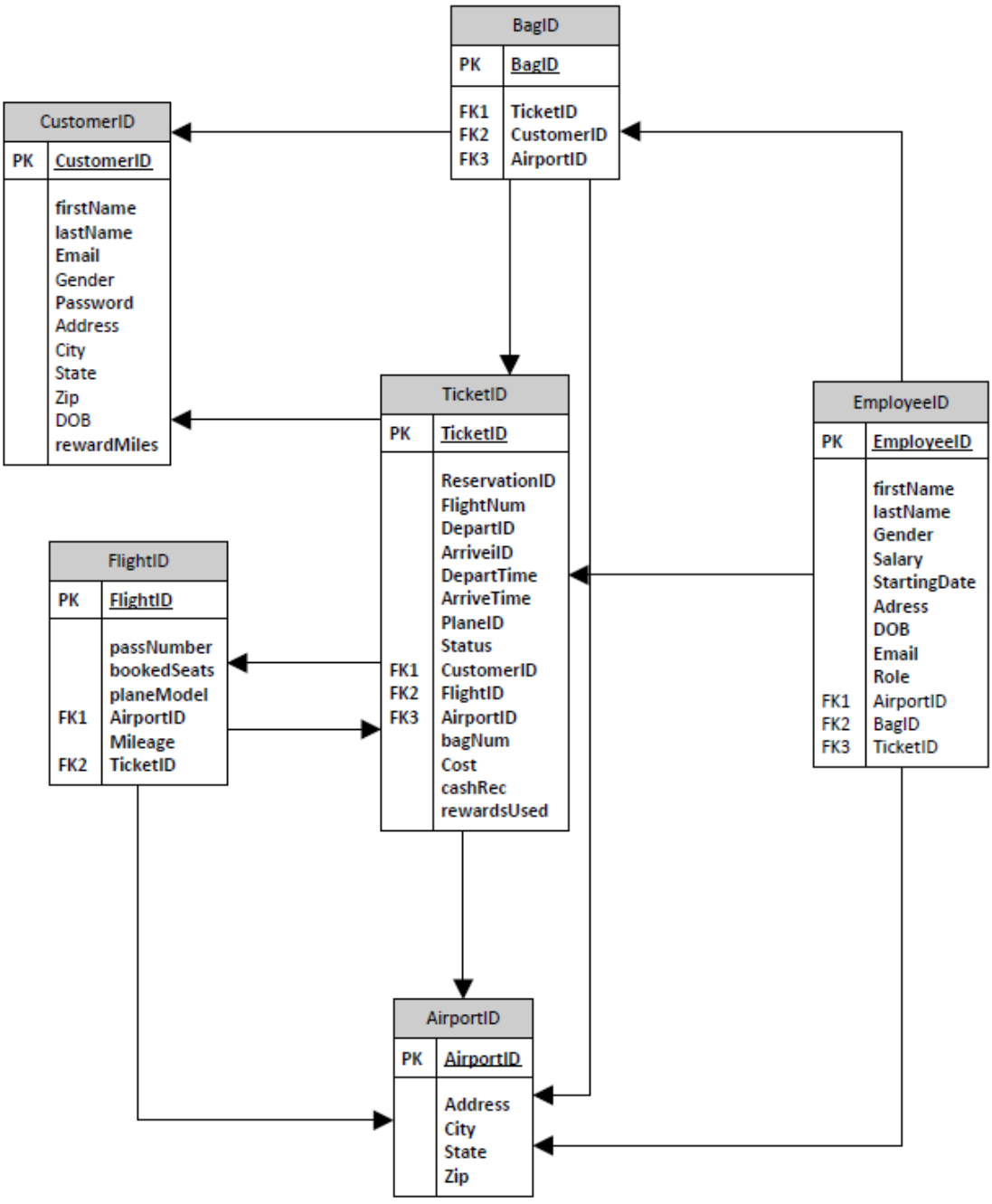
Java Air Reservation Look-up DFD



Java Air Flight Status Look-up DFD



ii. Database Diagram



Java Air Data Base Diagram 1.0