FUNCTIONAL SPECIFICATION

Project Code:	ATA-01
Project Name:	Automation of Travel Agency

Revision History

Version (x.yy)	Date of Revision	Description of Change	Reason for Change	Affected Sections	Approved By
1.00	20-Sep-2011	Initial Draft			
2.10	Sept-2013	Revision	Mapping with CPC Tool		
2.20	Dec-2013	Revision	Aligning with UCF		

Affected Groups

Development Engineering
Quality Assurance
XYZ Travels Ltd.

List of Reference Documents

Name	Version No.
1. Request For Proposal	1.2
2.	
3.	
4.	

Prepared by/Date **Reviewed by/Date** **Approved by/Date**

Table of Contents

1. INTRODUCTION 3
2. SYSTEM OVERVIEW 3
3. SUB-SYSTEM DETAILS 4
4. DATA ORGANIZATION 5
5. ASSUMPTIONS 6
6. EXPECTATIONS 6
7. ACCEPTANCE CRITERIA 6
8. TRACEABILITY TO REQUIREMENTS 6
9. ACRONYMS AND GLOSSARY 7

1 Introduction

XYZ Travels Ltd. provides vehicle booking facilities to users across many cities.

XYZ Travels Ltd. plans to develop "Automation of Travel Agency" - standalone/web application [Core Java Batches - Swing Application; J2EE Batches - Web Application], where users (Customers) can reserve vehicles and manage their reservations.

Scope and Overview:

The scope of the Automation of Travel Agency (ATA) will be to provide the functionality as described below. The system will be developed on a Windows operating system using Java/J2EE.

2 System Overview

The Automation of Travel Agency should support basic functionalities (explained in section 2.1) for all below listed users.

- Administrator (A)
- Customer (C)

2.1 Authentication & Authorization

2.1.1 Authentication:

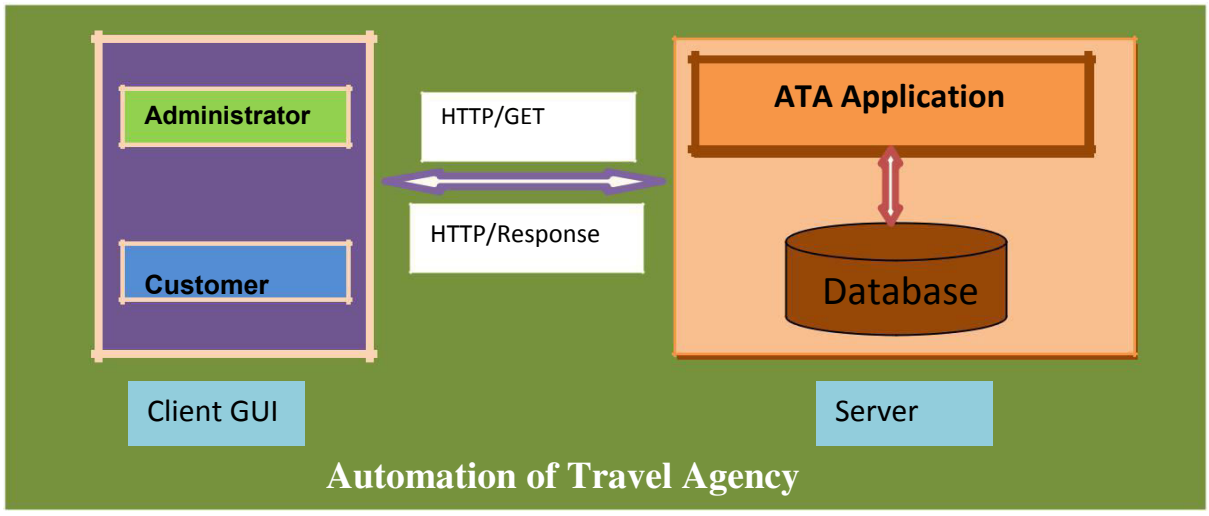
Any end-user should be authenticated using a unique login ID and password.

2.1.2 Authorization

The operations supported and allowed would be based on the user type. For example, Administrator has the rights to add/modify/delete and view driver, vehicle and route details. He can also view booking details traveling on a particular date/route.

2.2 Functional Flow

The functional flow of the messages across different application components is shown below.
Ex. - Web Application.



2.3 Environment

The system will be developed on a Windows XP machine using J2EE, JSP/HTML, and JDBC.

- Intel hardware machine (PC P4-2.26 GHz, 512 MB RAM, 40 GB HDD)
- Server – Apache Tomcat 6 or higher
- Database – Oracle 9i or higher
- JRE
- Eclipse IDE

3 Sub-system Details

The Automation of Travel Agency (ATA) is defined with two types of users (Administrator & Customer), wherein all users need to login successfully before performing any of their respective operations.

Find below (section 3.1 & 3.2) tables that provides functionality descriptions for each type of user / sub-system. Against each requirement, indicative data is listed in column 'Data to include'. Further, suggested to add/modify more details wherever required with an approval from customer/faculty.

3.1 Administrator

The administrator as a user is defined to perform below listed operations after successful login.

ID	Objects	Operations	Data to include	Remarks
AD-001 to AD-004	Vehicle	Add Delete View Modify	VehicleName, VehicleNumber, SeatingCapacity, DriverID, VehicleType, FarePerKm, etc	VehicleID should be auto generated. VehicleType: AC/NonAC
AD-005 to AD-008	Route	Add Delete View Modify	Source, Destination, Distance, Duration	RouteID should be auto generated
AD-009 to AD-012	Driver	Add Delete Modify Allotment	Name, Address, ContactNumber, LicenseNumber	ScheduleID should be auto generated
AD-013	Customer	View	VehicleID, BookingDate, JourneyDate, BoardingPoint, DropPoint, ContactNumber, Fare	

3.2 Customer

The customer as a user is defined to perform below listed operations after successful login.

ID	Objects	Operations	Data to include	Remarks
US-001	UserProfile	Register	Name, DOB, Gender, Address, Mobile Number, Email ID etc	UserID should be auto generated
US-002	Vehicle, Route	View vehicle/route details	Journey Date, Source, Destination, VehicleName	All unbooked vehicles should be visible
US-003	CreditCard	Book Vehicle	BookingDate, JourneyDate, Source, Destination, BoardingPoint, DropPoint, NoOfPassegers, etc	Booking should be confirmed after successful payment
US-004	Booking	Cancel Booking	BookingID	Booking can be cancelled only before journey time.
US-005	Booking	View booking status and print e-ticket	Booking information	

[Swing Application - Core Java]

* US-002 : Allow user to view current status of vehicle (vehicle list should be dynamically updated)
  - Hint: Use multithreading.

* US-005 : Allow user to generate ticket details in HTML format

[Web Application - J2EE]

* US-003 : Use | Create Web services for Payment process.
* US-005 : Allow user to generate ticket details in PDF format

NOTE:

* Total fare should be calculated on the basis of Route and VehicleType information.
  (Distance*FarePerKm)
* Vehicle status should be efficiently handled upon booking or cancellation.
* Fare for different types of vehicles are different.
* Bookings should be efficiently handled on deleting Vehicle details.

3.3 Login | Logout

[Swing Application - Core Java]

- Use System properties to enable the application to Startup with default/last user details for login.
- Enable the application to run from command prompt with user credentials.

[Web Application - J2EE]

- Implement Session tracking for all logged in users before allowing access to application features. Anonymous users should be checked, unless explicitly mentioned.

4 Data Organization

This section explains the data storage requirements of the Automation of Travel Agency and **indicative** data description along with suggested table (database) structure. The following section explains few of the tables (fields) with description. However in similar approach need to be considered for all other tables.

4.1 Table: UserProfile

The user specific details such as name, address, authentication and authorization / privileges should be kept in one or more tables, as necessary and applicable.

Field Name	Description
UserID	Customer ID is auto generated after registration and it is used as LoginID.
Name	Customer Name [first name & last name]
DOB	DOB of Customer
Gender	Gender of user [Male / Female]
PresentAddress	Present Address of Customer
PermanentAddress	Permanent Address of Customer
PhoneNumber	10 digit contact Number
Emailid	Email ID of the traveler

4.2 Table: UserCredentials

The table contains Authentication Information for Administrator and Customer

Field Name	Description
UserType	Administrator and Customer
UserID	User Identification, corresponding to UserProfile table
Password	Password
LoginStatus	Login status of the user

4.3 Table: CreditCardDetails

This table contains information related to the credit cards of the users used for making payments.

Field Name	Description
CreditCardNumber	Credit Card Number of the user
ValidFrom	Validity of Credit Card from
ValidTo	Validity of Credit Card to
Balance	Existing Credit Limit of the user

5 Assumptions

- User Interface: The type of client interface (front-end) to be supported– GUI based/Web based
- The scope of the application is limited to an agency which avails only cabs
- Fare per km for AC and NonAC will be fixed by admin
- The vehicle/driver assigned will be based on FIFO style depending on the availability and requirement
- Allotment of driver to a booking is performed by admin
- The customer can choose only vehicles and the route of journey

6 General Expectations

- The server should be a concurrent server servicing multiple clients
- Database can be implemented using Oracle 9i or above
- To begin with, the application should support at least 1 admin and 2 customers.
- Compilation and Build should be done using Eclipse IDE
- Source-code and all documents must be maintained (checked-in) in configuration management system (subversion)
- Int's coding standards (for Java) should be followed,
- Deliverables should include compiled and tested source code, Unit Test Code (Using IntUT), IntStyle report and System test-plan / report documents.

NOTE:

1. **Validation of user Data**[1]
   - ✓ Struts 2 validation via XML or annotations or Spring MVC using JSR-303 annotations
   - ✓ AJAX validation without forcing the page to reload (Wherever applicable)
   - ✓ JavaScript validation (if necessary)
   - ✓ In case of Swing applications, use 'ClassInputVerifier' for validation
2. **UI Design –(for Web Application) Use DIV/CSS to control the style and layout**
3. **Create at least one SQL DML-statement inside PL/SQL blocks**

7 Acceptance Criteria

All P1 requirements have to be mandatorily implemented

8 Traceability to Requirements

Appropriate requirements from RS and FS are mapped here.

Document Reference ID & Description: (Doc ID from which this document is derived)		
Sl. No.	**Reference document: RS Requirement/Feature (Section ID/Name)**	**Current document: FS Location (Section ID/Name)**
1.		
2.		

[1] Validations should be performed at all levels of application appropriately.

9 Acronyms and Glossary

Acronym and glossary for this document mentioned in the below table.

Abbreviation	Remark
ATA	Automation of Travel Agency
RS	Requirement Specification
FS	Functional Specification