WIPRO
Applying Thought

DETAILED DESIGN

Project Code:	SRS-01
Project Name:	Ship Reservation System

Revision History

Version (x.yy)	Date of Revision	Description of Change	Reason for Change	Affected Sections	Approved By
1.0	16/04/2013	Initial Draft			
1.10	May 2013	Revision			
2.10	July 2013	Revision			
2.20	Sept-2013	Revision	Mapping with CPC		
2.3	Dec-2013	Revision	Mapping with UCF		

Affected Groups

List of Reference Documents

Name	Version No.
1. RS_SRS	2.20
2. FS_SRS	2.20
3.	
4.	

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

Table of Contents

1. Introduction

Background

XYZ Sea Travels Ltd provides sea travel services to users (customers) across the globe.

Purpose

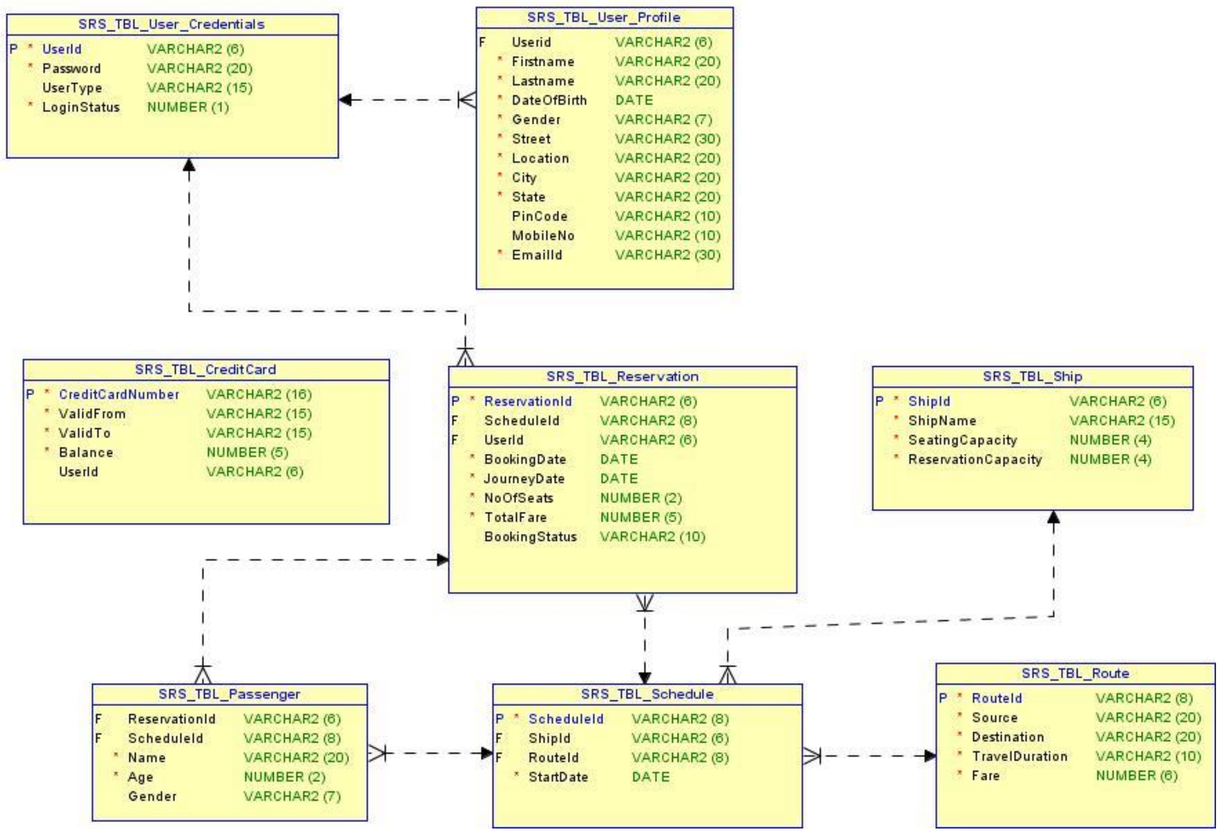
XYZ Sea Travels Ltd plans to develop “Ship Reservation System” - an application where users (customers) can reserve ship tickets and manage their reservations.

Scope

The scope of the Ship Reservation System [SRS] will be to provide the functionality as described in Functional Requirements document. The system will be developed on a Windows operating system using Java/J2EE and Oracle database.

2. Global Data Structures and Shared Data Functions

This section describes the structure of 8 tables to be used for the implementation of requirements as stated in the specification.



SRS_TBL_User_Credentials

P	*	UserId	VARCHAR2 (6)
	*	Password	VARCHAR2 (20)
		UserType	VARCHAR2 (15)
	*	LoginStatus	NUMBER (1)

SRS_TBL_User_Profile

F		Userid	VARCHAR2 (6)
	*	Firstname	VARCHAR2 (20)
	*	Lastname	VARCHAR2 (20)
	*	DateOfBirth	DATE
	*	Gender	VARCHAR2 (7)
	*	Street	VARCHAR2 (30)
	*	Location	VARCHAR2 (20)
	*	City	VARCHAR2 (20)
	*	State	VARCHAR2 (20)
		PinCode	VARCHAR2 (10)
		MobileNo	VARCHAR2 (10)
	*	EmailId	VARCHAR2 (30)

SRS_TBL_CreditCard

P	*	CreditCardNumber	VARCHAR2 (16)
	*	ValidFrom	VARCHAR2 (15)
	*	ValidTo	VARCHAR2 (15)
	*	Balance	NUMBER (5)
		UserId	VARCHAR2 (6)

SRS_TBL_Reservation

P	*	ReservationId	VARCHAR2 (6)
F		ScheduleId	VARCHAR2 (8)
F		UserId	VARCHAR2 (6)
	*	BookingDate	DATE
	*	JourneyDate	DATE
	*	NoOfSeats	NUMBER (2)
	*	TotalFare	NUMBER (5)
		BookingStatus	VARCHAR2 (10)

SRS_TBL_Ship

P	*	ShipId	VARCHAR2 (6)
	*	ShipName	VARCHAR2 (15)
	*	SeatingCapacity	NUMBER (4)
	*	ReservationCapacity	NUMBER (4)

SRS_TBL_Passenger

F		ReservationId	VARCHAR2 (6)
F		ScheduleId	VARCHAR2 (8)
	*	Name	VARCHAR2 (20)
	*	Age	NUMBER (2)
		Gender	VARCHAR2 (7)

SRS_TBL_Schedule

P	*	ScheduleId	VARCHAR2 (8)
F		ShipId	VARCHAR2 (6)
F		RouteId	VARCHAR2 (8)
	*	StartDate	DATE

SRS_TBL_Route

P	*	RouteId	VARCHAR2 (8)
	*	Source	VARCHAR2 (20)
	*	Destination	VARCHAR2 (20)
	*	TravelDuration	VARCHAR2 (10)
	*	Fare	NUMBER (6)

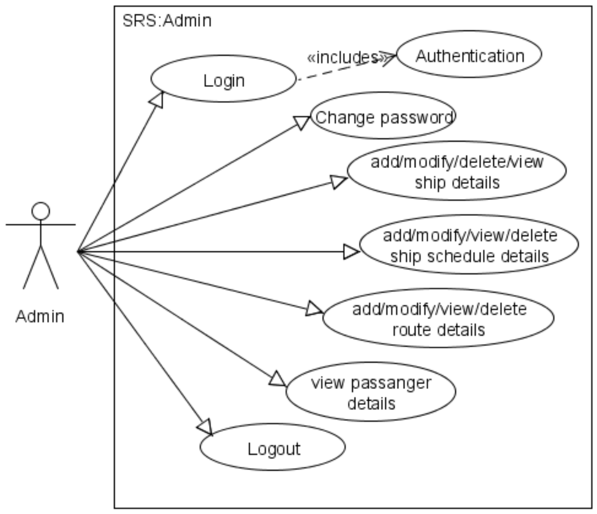
3. High Level Design

This section describes the high level design diagrams User case diagram with Use Case definition, Sequence Diagram and Class Diagram which provides a visual representation of the requirements , logical flow and their class representations.

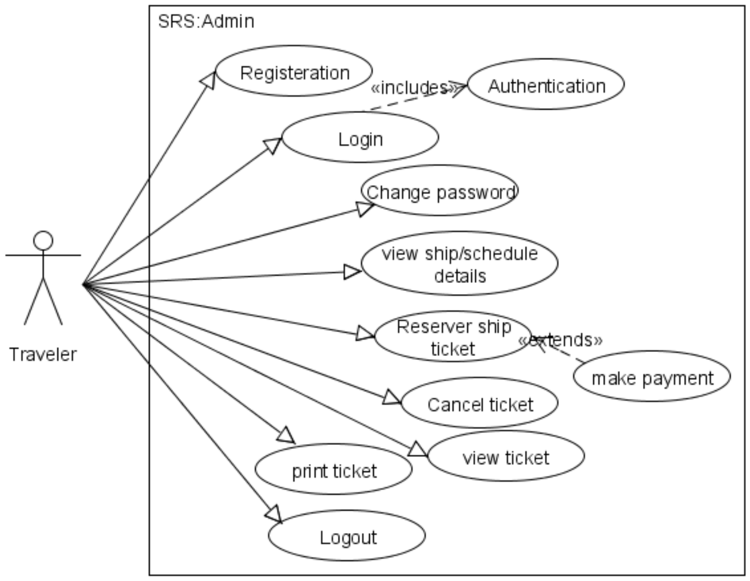
3.1 Use Case Diagrams

The requirements of a system can be represented using a use case model in the Use Case Diagram. The use case diagram for the actors of this case study is given as below.

3.1.1 Use Case Diagram for Administrator


SRS:Admin
Admin
Login
«includes»
Authentication
Change password
add/modify/delete/view ship details
add/modify/view/delete ship schedule details
add/modify/view/delete route details
view passanger details
Logout

3.1.2 Use Case Diagram for Customer



3.2 Use case Definition

Generally, in a design document, Use case definitions should be written for all the *Requirements* of the system.

Note: Participants are expected to document use case definitions for all requirements. However, for few requirements documented below for reference.

Below table explains 'Use Case' definition for requirement "AA-001" - Login operation for all users.

3.2.1 Login

All Users should be logged in to perform there respective functionalites.

USE CASE #	AA-001 Login	
Goal	All users logging into the system should be authenticated using a unique login-id and password (operations to be supported based on type of user)	
Preconditions	Credentials of the respective user should be present in srs_tbl_user_credentials table.	
Success End Condition	If the user type is 'Admin', he/she should be redirected to the Admin home page. If the user type is 'Customer', he/she should be redirected to the Customer home page.	
Failed End Condition	The end user is redirected to a Login Page with proper message, and is asked to re-enter login credentials.	
Primary, Secondary Actors	Admin, Customer	
Trigger	Login button	
DESCRIPTION	**Step**	**Action**
	1	Provide valid Login credentials
	2	Click on Login button
	Step	**Branching Action**
	1	If (User Type is Admin) then Redirect to Admin Page
	2	If (User Type is Customer) then Redirect to Customer Page
Related Information/Use cases		
Priority	P1	
Performance	2 seconds (excluding user input)	
Frequency	10 per minute	
Assumptions	No registration for Admin and login credentials are known to Admin	

3.2.2 Delete ship

Admin performs Delete ship functionality

USE CASE #	AD-002 Delete Ship	
Goal	Admin should delete ship details.	
Preconditions	Details of the ship should be present in database	
Success End Condition	Ship delete successfully should be displayed to the Admin	
Failed End Condition	Not able to delete ship details.	
Primary, Secondary Actors	Admin	
Trigger	Delete Ship button	
DESCRIPTION	**Step**	**Action**
	1	Provide valid shipId or relevant details
	2	Click on "Delete Ship" button
	Step	**Branching Action**
	1	If not able to delete, should display proper message to the Admin and redirection to delete ship page
Related Information/Use cases		
Priority	P1	
Performance	2 seconds (excluding user input)	
Frequency	3/month	
Assumptions	Ship not reserved for any schedule.	

3.3 Class Diagram

The class diagram is a very basic concept in object-oriented world. Class diagrams demonstrate a model, describing what attributes and behavior it has rather than describing the methods for accomplishing operations. Class diagrams are very useful in representing relationships between classes and interfaces.

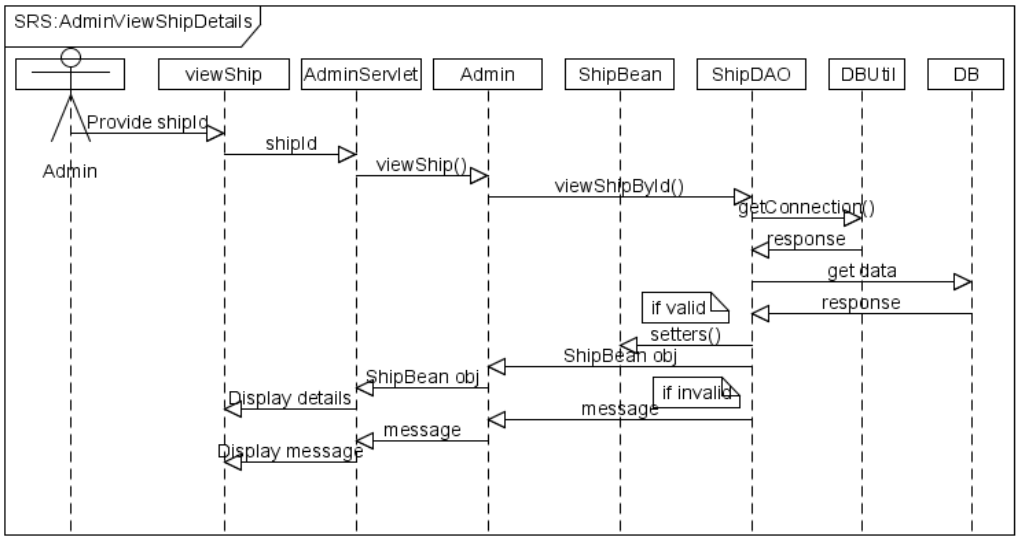
<to be designed by the participant………..>

3.4 Sequence Diagram

A graphical representation of a module's function invoking functions of other modules in order to achieve a task (specific user requirement) is called a sequence diagram. A sequence diagram for the authentication process is given below for reference. The below example is for a Web Application using jsp/servlets.

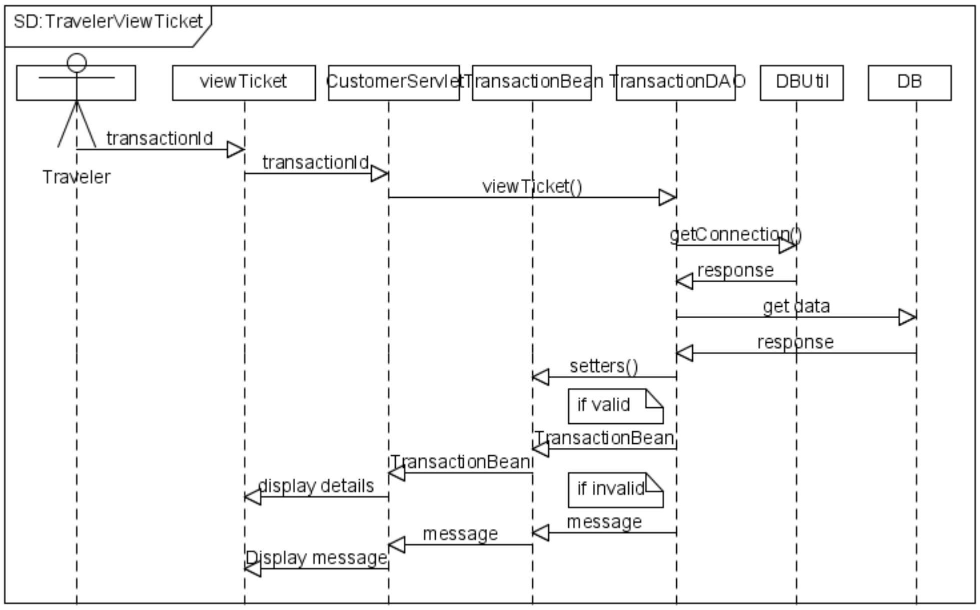
3.4.1 Add Ship

Admin performing view ship functionality



3.4.2 View Ticket

Customer performing view ticket operation



3.5 Packages / Interface / Classes

This section provides a brief outlook on the packaging hierarchy along with the respective classes to be used for the implementation.

The 4 packages mentioned below are for both GUI and Web Application.

Packages	
Package	**Description**
com.wipro.srs.service	This package contains all the Service classes
com.wipro.srs.bean	This package contains all the Bean classes
com.wipro.srs.dao	This package contains all the DAO functionality classes
com.wipro.srs.util	This package contains all the generic functionality classes

This package is used only for a GUI application.

com.wipro.srs.ui	This package contains all the UI related classes [For Core Java]

The package for the controller class should be used as below based on the type of application

com.wipro.srs.listener	**listener -** core java
or	
com.wipro.srs.servlet	**servlet -** Web Applications
or	
com.wiro.srs.action	**action -** Struts
or	
com.wipro.frs.controller	**controller -** Spring

Package com.wipro.srs.bean

Class Name	Attributes	Data Type
CredentialsBean	userID	String
	password	String
	userType	String
	loginStatus	int
ProfileBean	userID	String
	firstName	String
	lastName	String
	dateOfBirth	Date
	gender	String
	street	String
	location	String
	city	String
	state	String
	pincode	String
	mobileNo	String
	emailID	String
	password	String

ShipBean	shipID	String
	shipName	String
	seatingCapacity	int
	reservationCapacity	int
ScheduleBean	scheduleID	String
	shipID	String
	routeID	String
	startDate	Date
ReservationBean	reservationID	String
	scheduleID	String
	userID	String
	bookingDate	Date
	journeyDate	Date
	noOfSeats	int
	totalFare	double
	bookingStatus	String
PassengerBean	reservationID	String
	scheduleID	String
	name	String
	age	int
	gender	String
RouteBean	routeID	String
	source	String
	destination	String
	travelDuration	String
	fare	double

Package com.wipro.srs.service

Interface Summary	
Interface	**Description**
Administrator	Entity interface for Administrator dealing with the Administrator process functionalities.

Method Summary	
String	**addShip**(ShipBean shipbean) Return value must be either: "SUCCESS", "FAIL", ERROR”
boolean	**modifyShip**(ShipBean Shipbean)
int	**removeShip**(ArrayList<String> ShipId)
String	**addSchedule**(ScheduleBean schedulebean) Return value must be either: "SUCCESS", "FAIL", “ERROR”
boolean	**modifySchedule**(ScheduleBean schedulebean)
int	**removeSchedule**(ArrayList<String> scheduleid)
String	**addRoute**(RouteBean routebean) Return value must be either: "SUCCESS", "FAIL", “ERROR”
boolean	**modifyRoute**(RouteBean routebean)
int	**removeRoute**(String routeid)
ShipBean	**viewByShipId**(String ShipId)
RouteBean	**viewByRouteId**(String routeid)
ArrayList<ShipBean>	**viewByAllShips**()
ArrayList<RouteBean>	**viewByAllRoute**()
ArrayList<ScheduleBean>	**viewByAllSchedule**()
ScheduleBean	**viewByScheduleId**(String scheduleid)
ArrayList<PassengerBean>	**viewPasengersByShip**(String scheduleid)

Interface	Description
Customer	Entity interface for Customer dealing with the customer process functionalities.

Method Summary	
ArrayList<ScheduleBean>	**viewScheduleByRoute** (String source, String destination, Date date)
String	**reserveTicket**(ReservationBean reservationBean, ArrayList<PassengerBean> passengerBean) Return value must be either: "SUCCESS", "FAIL"
boolean	**cancelTicket**(String reservationId)
Map<ReservationBean, PassengerBean>	**viewTicket**(String reservationId)

	Map<ReservationBean, PassengerBean>	**printTicket**(String reservationId)

Package com.wipro.srs.dao

Find below the suggestive approach for CRUD operations [method naming & signature] for the DAO classes. Create the necessary DAO interface/classes.

Interface Summary	
Interface	**Description**
xyzDAO	DAO interface to deal with operations related to the specific table.
	Method Summary
	String createXYZ(BeanObject)
	int deleteXYZ(ArrayList<String>)
	boolean updateXYZ(BeanObject)
	BeanObject findByID(String)
	ArrayList<BeanObject> findAll()

- If required, additional find methods can be created.

Package com.wipro.srs.util

Interface Summary	
Interface	**Description**
Authentication	This interface is responsible for performing the Authentication and Authorization process.
	Methods boolean **authenticate**(CredentialsBean credentialsBean) String **authorize**(String userId) boolean **changeLoginStatus**(CredentialsBean credentialsBean, int loginStatus)
User	interface for handling different types of users
	Methods String **login**(CredentialsBean credentialsBean) Return value must be either: "A", "C", "FAIL", “INVALID” A->Admin, C->Customer **Wrong username/password should return INVALID.** boolean **logout**(String userId) String **changePassword**(CredentialsBean credentialsBean, String newPassword) Return value must be either: "SUCCESS", "FAIL", “INVALID” String **register**(ProfileBean profileBean) Return value must be either: <userId of lenght 6>, "FAIL" Note: userId-> first 2 letter of first name followed by 4 digit auto generated number

Payment	Interface for handling payment related information String creditCardNumber, validFrom, validTo int balance
	Methods boolean **findByCardNumber**(String userid, String cardnumber) String **process**(Payment payment)
DBUtil	This interface is responsible for performing the Database connectivity. **Method Summary** static Connection getDBConnection(String driverType)

3.6 UI Templates

3.5.1 UI Principle

The UI [Presentation Layer] should be designed with the below mentioned principles which helps easy interaction by the user to the application.

3.5.2 UI controls and Usage Principle

UI Type	Controls	Description
Direct Entry	Text Box, Text Area	Any input that cannot be predicted and needs the user to key in. e.g Name, Address, contact no etc.
Static Selection	Option Button, Check Box, Drop Down	Should be used where the input can be predefined. e.g gender, month [Jan – Dec] etc. If number of items is more, drop down is preferred.
Dynamic Selection	Drop Down	The items for the drop down should be retrieved from a stored data. e.g Displaying Districts in a drop down from places table.
Automation	Label Text Field [Read Only]	Data’s that are calculative or an output of a function. e.g : Displaying system date, showing total amount etc.
Decision Control	Button	Operations like submit, save, clear should be executed only upon clicking respective buttons.

3.5.3 UI Template

This section contains the design template for the website home page [Fig. 1] that will be displayed at the time of opening this web application and Actor specific home page [Fig. 2].

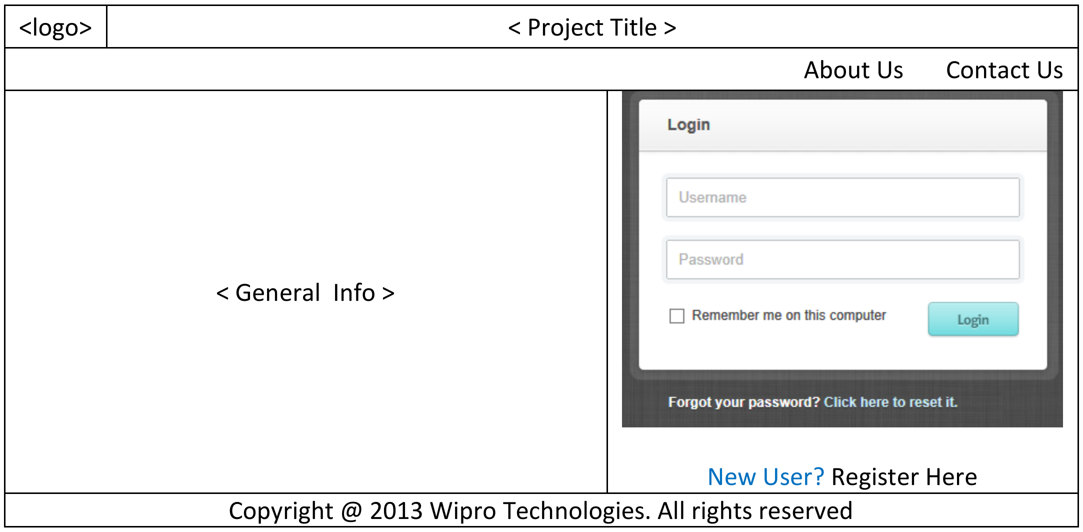

<logo>
< Project Title >
About Us
Contact Us
< General Info >
Login
Username
Password
Remember me on this computer
Login
Forgot your password? Click here to reset it.
New User? Register Here
Copyright @ 2013 Wipro Technologies. All rights reserved

Fig. 1 - Main Page [First Page to open]

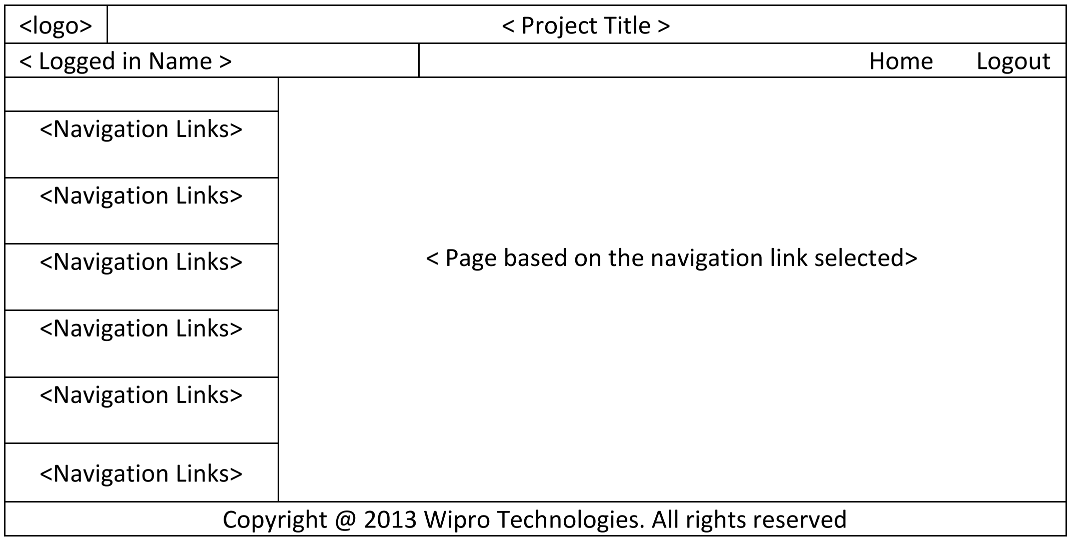
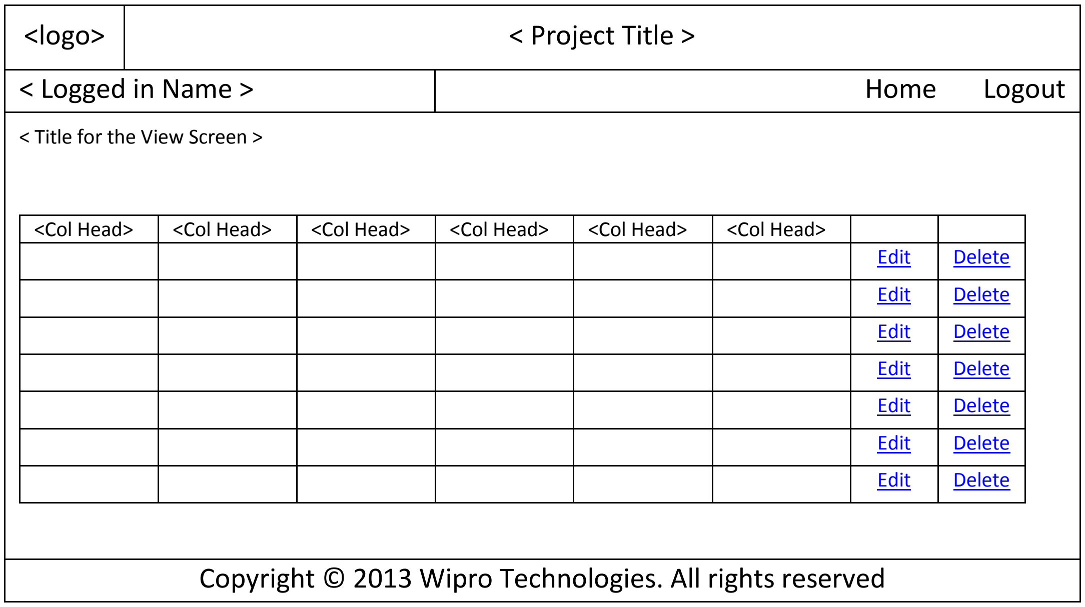


Fig. 2 - Home Page for Actor



<logo>	< Project Title >
< Logged in Name >	Home Logout

< Title for the View Screen >

<Col Head>	<Col Head>	<Col Head>	<Col Head>	<Col Head>	<Col Head>		
						Edit	Delete
						Edit	Delete
						Edit	Delete
						Edit	Delete
						Edit	Delete
						Edit	Delete
						Edit	Delete

Copyright © 2013 Wipro Technologies. All rights reserved

Fig. 3 – View Screen with Edit and Delete Functionality

4. Critical Functions and Focus for Testing

Authorization & Authentication are the critical functions need to be implemented before performing the tasks.

5. Limitations

- The administrator can set the schedule for the ships on a monthly basis
- The scope of the application is limited to only one country
- The seats are assumed to be of Non-AC type

6. Appendix

1 Table : SRS_TBL_User_Credentials

This table contains Authentication Information for Administrator, Customer [Passenger]

Field Name	Data Type	Description
UserId	VARCHAR2(6)	Auto-generated, Primary Key*
Password	VARCHAR2(20)	Not Null
UserType	VARCHAR2(15)	Either ['A','C'] A->Admin, C->Customer
LoginStatus	Number(1)	Not Null

* First 2 letters of User first name followed by 4 digits auto generated number

2 Table : SRS_TBL_User _Profile

This table contains User specific details entered during User Registration.

Field Name	Data Type	Description
Userid	VARCHAR2(6)	Foreign Key
Firstname	VARCHAR2(20)	Not Null
Lastname	VARCHAR2(20)	Not Null
DateOfBirth	DATE	Not Null
Gender	VARCHAR2(7)	Not Null
Street	VARCHAR(30)	Not Null
Location	VARCHAR2(20)	Not Null
City	VARCHAR2(20)	Not Null
State	VARCHAR2(20)	Not Null
PinCode	VARCHAR2(10)	
MobileNo	VARCHAR(10)	Exact 10 digit only
EmailId	VARCHAR2(30)	Not Null

3 Table : SRS_TBL_Ship

This table contains ship related information.

Field Name	Data Type	Description
ShipId	VARCHAR2(6)	Auto-generated, Primary Key**
ShipName	VARCHAR2(15)	Not Null
SeatingCapacity	NUMBER(4)	Not Null
ReservationCapacity	NUMBER(4)	Not Null

* ShipId should be, first 2 letters of Ship name followed by 4 digits auto generated number

4 Table : SRS_TBL_Route

This table contains route related information.

Field Name	Data Type	Description
RouteId	VARCHAR2(8)	Auto-generated, Primary Key***
Source	VARCHAR2(20)	Not Null
Destination	VARCHAR2(20)	Not Null
TravelDuration	VARCHAR2(10)	Not Null
Fare	NUMBER	Not Null

*** RouteId should be, first 2 letters of source name followed by 2 letters of destination name followed by 4 digits auto generated number

5 Table : SRS_TBL_Schedule

This table contains ship schedule details which is utilized for booking ticket.

Field Name	Data Type	Description
ScheduleId	VARCHAR2(8)	Auto-generated, Primary Key
ShipId	VARCHAR2(6)	Foreign Key
RouteId	VARCHAR2(8)	Foreign Key
StartDate	Date	Not Null

* ScheduleId should be first 2 letters of source name followed by 2 letters of destination and auto generated 4 digits

6 Table : SRS_TBL_Reservation

This table contains booking related information performed by Customer.

Field Name	Data Type	Description
ReservationId	VARCHAR2(8)	Auto-generated, Primary Key
ScheduleId	VARCHAR2(8)	Foreign Key
UserId	VARCHAR2(6)	Foreign Key
BookingDate	DATE	Not Null
JourneyDate	DATE	Not Null
NoOfSeats	NUMBER	Not Null
TotalFare	NUMBER(5)	Not Null
BookingStatus	VARCHAR2(10)	

- ReservationId should be first 2 letters of source name followed by 2 letters of destination and auto generated 4 digits

7 Table : SRS_TBL_Passenger

This table contains Passenger/Passenger related information once booking has done.

Field Name	Data Type	Description
ReservationId	VARCHAR2(8)	Foreign Key
ScheduleId	VARCHAR2(8)	Foreign Key
Name	VARCHAR2(20)	Not Null
Age	NUMBER	Not Null
Gender	VARCHAR2(7)	

8 Table : SRS_TBL_CreditCard

This table contains credit card details for performing payment by the Customer.

Field Name	Data Type	Description
CreditCardNumber	VARCHAR2(16)	Primary Key
ValidFrom	VARCHAR2(15)	Not Null
ValidTo	VARCHAR2(15)	Not Null
Balance	NUMBER	Not Null
UserId	VARCHAR2(6)	

Database Sequences

Sequence Name	Purpose	Starts with
SRS_SEQ_USER_ID	User ID	1000
SRS_SEQ_ROUTE_ID	Route ID	1000
SRS_SEQ_SHIP_ID	ShipID	1000
SRS_SEQ_SCHEDULE_ID	Schedule ID	1000
SRS_SEQ_RESERVATION_ID	Reservation ID	1000