PRCS252 – Travel Management System Software:

Initial Planning Document

Overview:

This document is intended to provide a brief description of the travel management system that is to be created for the integrating project. The system will be for buses which will factor in a database to communicate information between mobile, website and desktop clients through an API.

Based off the specification we will be using buses as our mode of transport to base the system around so that a bus company can manage their buses and routes.

Team Information:

1. **William Butler** – Team member, database developer and C# programmer
2. **Andrew Bellas** – Team member, web developer, HCI and C# programmer
3. **Goel Biju** – Team member, database developer and C#/Java programmer
4. **Vincent Castellani** – Team member, database developer and HCI

**•** All team members will contribute to various aspects of the project.

Technologies and Tools:

* **Oracle SQL Developer**; used to develop the database and all associated database objects: triggers, views, constraints.
* **.NET API**; will be used to transfer data to and from the database and clients, this middleware will be hosted on an Xserve server.
* **Java** (NetBeans); for development of the desktop application for staff.
* **Android** (Java); development of mobile application for customers.
* **HTML/CSS/JavaScript**; used to develop website for admin to manage the system.
* **GitHub**; repository for version control and project management through the project board.

Functional requirements

Customer (mobile) application:

* Customer will be able to register an account through the mobile application:
* Create booking (minimum age of the passenger must be 15 in order to make an independent booking).
* Pay for a booking through integrated payment system – PayPal.
* View timetables for routes.
* View information regarding a service or route.
* View history of prior journeys.
* Customer can login to the mobile application to book new journeys/manage account.
* Customer can book a new journey from a prior journey.
* Customer can view booked journeys which has booking information e.g. booking reference code.
* Customer can request account termination.
* Customer can update account details.
* Customer’s mobile application should update based on new information from the system.

Driver (desktop) application:

* Start a service for a valid route.
* Complete a service for a valid route.
* View current stock (buses).
* View route.
* Log into the system – log in details are stored until the driver signs out; a session is created upon logging in.
* Accept booking reference number/ticket number in order to validate passenger’s right of travel (from a list of all bookings made for that journey).
* View bus service information: bus capacity, remaining seats, bookings made for the journey, locations to pick up booked passenger from.

Admin application:

* Create records of what buses are in the depot and their status
* Update records of what buses are in the depot and their status
* Retrieve records of what buses are in the depot and their status
* Delete/archive what buses are in the depot and their status
* Create records of the journeys of the buses
* Update records of the journeys of buses
* Retrieve records of the journeys of the buses
* Delete/archive records of the journeys of the buses
* Create records of the starting and ending points of the bus journeys
* Update records of the starting and ending points of the bus journeys
* Retrieve records of the starting and ending points of the bus journeys
* Delete/archive records of the starting and ending points of the bus journeys
* Create records of the timetables of buses
* Update records of the timetables of buses
* Retrieve records of the timetables of the buses
* Delete/archive records of the timetables of the buses
* Deploy replacement services in the event of a bus breakdown and therefore assist passengers.