CS3002 Problem Specification

# Project description & Requirements.

The aim of this document is to provide a problem specification and a high level overview of my final year project. The project, which involves development of a mobile app, has been proposed and will be supervised by Kainos, the software development company based in Belfast.

The goal of the app is to facilitate the matching up of car sharing participants. Using their handheld Android devices, users must be able to register and log in in order to be able to post and search for car share listings. The application must provide users with the ability to specify dates, start and end destinations as well as multiple other criteria.

From the user’s perspective, the application’s user interface must be intuitive and allow for smooth transition from one activity screen onto another. On-screen controls and the screen activity’s layout must be designed carefully due to varying screen sizes as well as a generally limited amount of space available on the screen.

The functional requirements for this project are set out using the MoSCoW methodology and are listed below:

**Must haves:**

* User registration/login
* A web based admin panel to allow Administrator to log in and manage the system.
* Searching for car share listings by start & end locations as well as date.
* Ability for users to post new car share listings and specify dates, locations and fee.
* Save a list of user trips.
* Exchange messages with other users via the application.
* Implementation of Google Maps API to find cities and plot routes on the map.
* Quick search facility with locations and dates based on user defined search criteria.

**Should haves:**

* Advanced search options such as: women only, smokers, type of vehicle, fee, and number of seats.
* Rating system based on passengers experience with leader board.

**Could haves:**

* Service in the background with notifications when a car share becomes available.
* Specify the main stops in the journey.
* Allow users to specify a radius from the start and end locations in miles.
* Allow users to specify a city region from the start and end locations in miles.
* Instant messaging feature to allow app users exchange messages in real time.
* Search for car share listings with the help of GPS to find the one with nearest start location.

**Would like to haves:**

* Live driver tracking using GPS.
* Possible integration with Facebook/Google+

# High-level application overview.

The project will follow the N-Tier architecture and is going to be based client-server model. The client will be developed in JAVA and deployed as a mobile application for the Android Operating System. The client will communicate with a .NET WCF web service written in C# which in turn will communicate with a SQL database through Entity Framework. For more details, please see the diagram below.

To enable separation of concerns and ensure loose coupling between different layers of the system, the entire application will be divided into a number of layers. Please see figure 1 below for more details.

**Google Maps API**

* Location services.
* Plotting routes.
* Device tracking.

JAVA

**Client**

C#

Figure Application architecture overview.

**Database**

* SQL Server.

**Entity Framework**

* Code first.
* Database first.

**Data Access Layer (DAL)**

* IUnitOfWork.
* IRepository.
* ApplicationContext.

**Business Logic Layer**

* Return list of car share listings.
* Process new car share listing.

**WCF Web Service**

* Multiple endpoints.
* Exchanging JSON messages.
* RESTful services.

Removing as much business logic as possible from the client and exposing it via the WCF service instead will allow the application to be deployed more easily across multiple platforms i.e. Desktop, iOS etc. because the client will simply invoke methods exposed by the WCF service. This is extremely important to ensure code modularity and future scalability and expandability of the application.

**Hardware constraints:**

Due to the nature of the project, the application will require at least two separate hardware devices to be deployed on in order to function properly. It will be possible for all of the server-side functionality to reside on the same machine whilst client application will connect to it from an Android device and consume the WCF web service. Please see below for a summary of hardware requirements:

Server side:

* A Microsoft Windows Operating System with IIS and SQL Server Express installed.
* 4GB RAM.
* Dual Core AMD/Intel CPU.
* 20GB Free Hard Drive space.
* Full Internet connectivity.

Client side:

* Android device with full Internet connectivity and OS version 4.1 or higher.

**Development environment:**

Vast majority of the application will be developed in Microsoft Visual Studio 2012 and Android Studio for client side and user interface. For a more detailed breakdown, please see the list below.

Microsoft Visual Studio 2012

* WCF Service.
* Business logic layer.
* Entity Framework.
* Data access layer (DAL).
* Web Service unit tests.

Android Studio

* Client UI.
* Google Maps integration.

Source Control

* Private GitHub repository.

**Development methodology:**

Development will be carried out using the agile methodology. Each two week long sprint will be followed by a sprint retrospective and review. Sprints will be documented with user stories and burn down charts as proof of progress being made.

# Gantt chart