<Company Name>
<Company Name>

<Car Renting Application>
Supplementary Specification

Version <1.0>

Revision History

Date	Version	Description	Author
<dd/mmm/yy>	<x.x>	<details>	<name>

Table of Contents

Supplementary Specification

1. Introduction

The system is a client-server application for renting cars which can be used by customers and by the renting company's employees. The application will allow a customer to search and view the available cars (detailed car information and pictures) for the selected date interval and from the specified source city. To rent a car, the customer must fill its personal data.

After the customer has rented a car, an email is sent to him with the renting details. When a customer that has rented a car comes to the renting company to pick up the car, an employee creates a contract which is then printed so that the client can sign it. When a car is brought back by a customer, the employee marks the car as returned.

The application must have also an administrator component for CRUD operations over cars, customers and employees.

2. Non-functional Requirements

- **Source of stimulus**: Human interaction
- **Stimulus**: Human entering different data and expecting specific outcome
- **Environment**: the system "idle" state is either the login page or one of it's user specific page
- **Artifact**: the whole system reacts to stimulus but only the user interface provides results and acquire data from user
- **Response**: the system either introduces data to database or display some information to UI
- **Response measure**: the number of records extracted from database and displayed

2.1 Availability

The application in its current state it is available as a desktop app, but because of its client-server architecture it can be extended.

2.2 Performance

Running the application takes just a few seconds and retrieving/uploading information about 0.4 seconds.

2.3 Security

Users information are stored in a database. Because the passwords are all protected by converting and storing them using a hash function, these information are safely stored.

2.4 Usability

Because of its friendly user interface the application is easy to use even for a non-experienced user. The application's features are described using labels, buttons and text fields.

3. Design Constraints

- The code for this application was written using C# programming language.
- Compiler Software – Visual Studio 2013
- Architectural Constrains – Client-Server Application
- Different Form Application for each type of user
- Using a database to store information and pictures
- Storing passwords in a safe manner