# 1.jpg

**Assessment Cover Sheet**

**Assessment Title:**

**BACHELOR OF ICT**

**ECHNOLOGY**

**Programme Title:**

**Project**

**IT8415**

**Course No.:**

**Course Title:**

**Database Programming 2**

**Due Date: See table below**

***Requests for extensions should be made to the course coordinator 48 hours before the deadline.***

***You are only permitted one extension per subject per semester***

***Late Rule:***

1. If an assessment is submitted late the maximum result the student can achieve is 60%.
2. The cut off time for submitting an assessment will be 3 calendar days after the assessment is due. A student submitting after 3 calendar days will get 0%.
3. Extensions request should be made 48 hours before submission date, and for valid reasons only.

Contents

[ 1](#_Toc480281137)

[Database Programming 2 Project 3](#_Toc480281138)

[Due Dates 3](#_Toc480281139)

[Project Description 4](#_Toc480281140)

[Searching for available cars 4](#_Toc480281141)

[Creating a reservation 5](#_Toc480281142)

[Amending Reservation Details 5](#_Toc480281143)

[Administration System 6](#_Toc480281144)

[Administration Reports 6](#_Toc480281145)

[Other Requirements 7](#_Toc480281146)

[Deliverables – what you need to hand in 7](#_Toc480281147)

[Marking Scheme 7](#_Toc480281148)

# Database Programming 2 Group Project

**Due Date: See below**

**Weighting: 30%**

**Submission: Electronic via Moodle**

**Assessment: Group**

**Learning Outcomes**

The following learning outcomes will be assessed in this assessment:

1. Design, develop and test data driven, server-side applications.

2. Follow best practice, industry standards, design methodology, programming and documentation conventions during the development process

## 

## Due Dates

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Phase |  | Due Date | Type | Submission |
| Final Project Submission |  | 10/6/2018 | Group | Electronic |
|  |  |  |  |  |

# Project Description

**Online Car Reservation System Functional Requirements**

### 

### **Searching for available cars**

A car rental company has an online reservation system that allows customers to search for and to reserve cars.

When a customer wants to make a reservation the first step is to enter the preferred dates of reservation, and then the desired car from different car categories and the price range per day.

After the customer submits this information a list of possible cars is shown. The customer should be able to see all available cars on these days along with the car price and image.

At this point the customer can enter different search criteria to find alternative cars information.

If the customer is happy with the details they can select the required car(s), and proceed with the reservation.

### **Creating a reservation**

When the customer gets to the reservation page the car details are displayed - manufacturer, model, year and price, with any other additional details of the car.

If the customer does not wish to proceed they can cancel and return to the main page.

If the customer chooses to proceed they are presented with a screen where they can choose additional car accessories like (car seat, screens, navigation … etc.).

Then in the next page the customer should enter his details in order to proceed with the reservation.

At this stage the customer can still cancel and return to the main page otherwise the customer is now required to go to the checkout page where he/she can view a summary of his reservation with the total cost.

The customer now confirms the reservation. The reservation details are displayed.

If the user is satisfied with the reservation details they can confirm and proceed to the payment page.

On this page the customer enters the payment type, card details (number, expiry date, security number), the cardholders name, nationality and billing address.

On confirmation of these details the user will be shown a summary screen and treservation

### **Amending Reservation Details**

If, after making a reservation, a customer wishes to change their reservation details, the customer will be able to access their reservation details using the reservation number sent to them.

Customers will be permitted to change their reservation details up to 2 days before they are due to get the car but amending the reservation will incur a 10% charge.

If the user cancels the reservation the car available dates should be updated accordingly.

### **Administration System**

The system also requires an admin section where company staff can log in to add and amend details of cars

* Make
* Model
* Year

And any other details.

Admin staff should be able to search the existing cars by different criteria’s.

### **Administration Reports**

The admin system should include a management reporting subsystem showing the following reports:

* Most popular reserved cars.
* Monthly sales revenue – this should be displayed for a period between dates entered by the admin user.

### **Other Requirements**

**Navigation** – the system should apply a consistent and logical means of navigating the application.

**Look and feel** - the application should have a professional look and feel and should be user friendly. It should use appropriate windows controls should as drop-down lists, radio buttons and so on wherever possible.

**Validation of input** – it is important that data is validated before being added to the database to ensure system integrity. Any invalid input should result in an appropriate error message being displayed.

Other features you may wish to consider are multiple language and currency support and ensuring that the database is secure.

Additional marks will be awarded for extra features added that exceed the original requirements.

**Note**: it is important that your database contains sufficient data to allow it to be tested thoroughly so ensure that you have created an effective test bed.

## 

## Deliverables – what you need to hand in

* Your Netbeans project
* SQL for creating tables and data plus any functions, procedures or triggers
* A document outlining any additional features
* A test plan outlining how to test your system to demonstrate the functional requirements

## Marking Scheme

|  |  |  |
| --- | --- | --- |
| Section | Topic | Marks |
| 1 | Test Plan | **5** |
| 2 | Application functionality | **85** |
| 3 | Additional features | **10** |
|  | Total | **100** |