Systems Analysis and design

Ray’s Rentals

Faran Azadi, Aamer Atique, Yusof Bandar, Naim Ahmed and Pritam Sangani

Table of Contents

[1.0 Introduction 2](#_Toc468477001)

[2.0 Use Case Diagrams 3](#_Toc468477002)

[2.1 Use Case Diagram Commentary 4](#_Toc468477003)

[3.0 Use Case Specifications 5](#_Toc468477004)

[3.1 Hire Bike 5](#_Toc468477005)

[3.1.1 Hire Bike ERD 5](#_Toc468477006)

[3.2 Update Maintenance Record 6](#_Toc468477007)

[3.2.1 Update Maintenance Record ERD 6](#_Toc468477008)

[3.3 Update Bike Record 7](#_Toc468477009)

[3.3.1 Update Bike Record ERD 7](#_Toc468477010)

[3.4 Order Parts 8](#_Toc468477011)

[3.4.1 Order Parts ERD 8](#_Toc468477012)

[3.5 Update Rental Record 9](#_Toc468477013)

[3.5.1 Update Rental Record ERD 9](#_Toc468477014)

[4.0 Top-Down ERD 10](#_Toc468477015)

[5.0 Bottom-Up ERD 11](#_Toc468477016)

[5.1 RDAs 11](#_Toc468477017)

[5.1.1 Maintenance and Dealer RDA 11](#_Toc468477018)

[5.1.2 Rental and Bike Record RDA 11](#_Toc468477019)

[5.1.3 Merged RDA 12](#_Toc468477020)

[5.2 Bottom-Up ERD from Merged RDA 13](#_Toc468477021)

[6.0 Final Merged ERD 14](#_Toc468477022)

[6.1 ERD Commentary 15](#_Toc468477023)

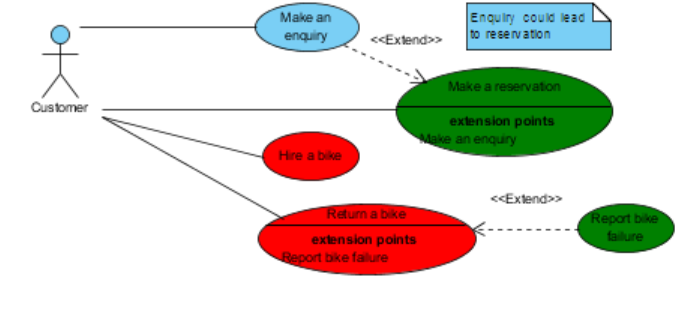
[7.0 Appendix 16](#_Toc468477024)

[7.1 Log Book 16](#_Toc468477025)

# 1.0 Introduction

For this report, our group had to design an entity relationship diagram (ERD) and a use case diagram (UCD) using the “MoSCoW” system of prioritisation. This had to be done in the context of the case study of “Ray’s Rentals”, which is a bike rental store. A finalised group ERD was created and each group member completed their own UCD and specification. A commentary is also provided to outline what has been learned and how obstacles were overcome while working on this project. This report consists of the findings that have been made during this part of the project.

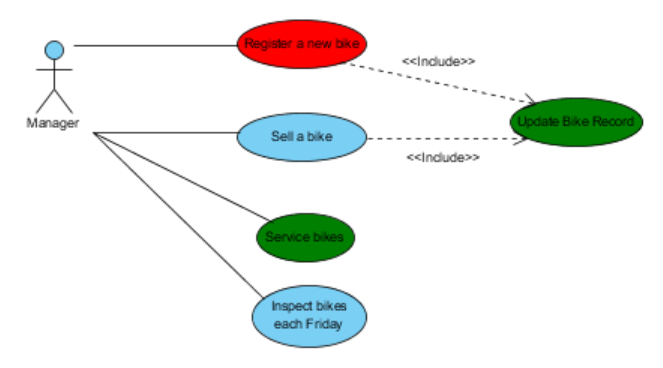
# 2.0 Use Case Diagrams

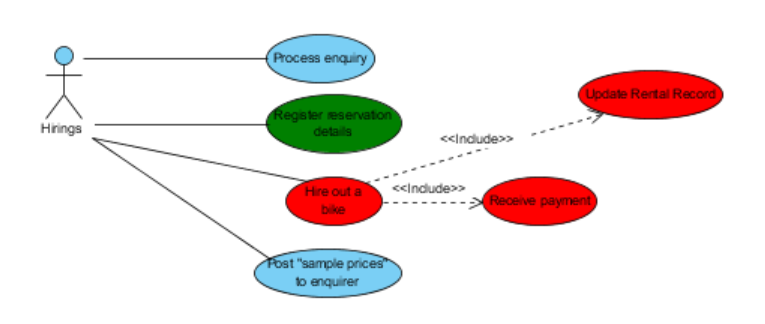


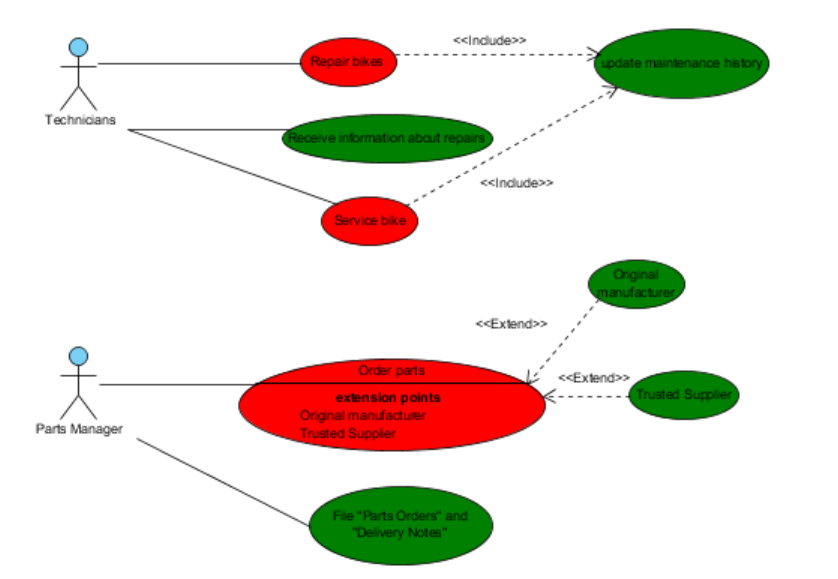
Must Have

Should Have

Could Have







## 2.1 Use Case Diagram Commentary

Initially, four different use cases were created; one by each member, merging the smaller use cases into one large one. We decided that the use cases must include "include"s and "extend"s. For example, we decided that extend was needed for the enquiries use case as it could lead to a reservation. We also decided that we needed an extra actor as none of the use cases included the technicians. Furthermore, we also changed the actors’ names to be less specific. Instead of specific names, such as Paul, the Parts Manager, we changed them to more general roles, such as "Manager" or "Hirings Department" .

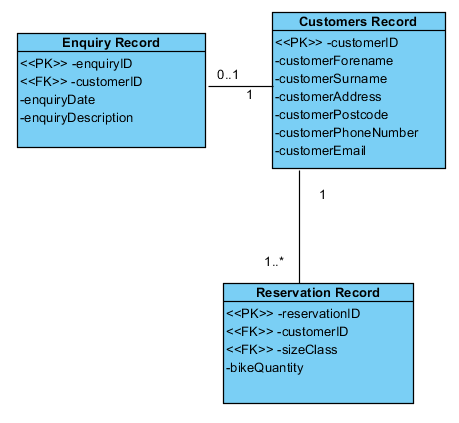
Merging the four use cases into one, developed some issues. The first issue noticed was that some of the use cases weren’t actions. We learnt that each use case must be an action, not just a statement, such as "update rental record", and so writing "rental record" was incorrect. A second issue noticed was some excludes had been drawn incorrectly; the arrows pointed towards the wrong action.

# 3.0 Use Case Specifications

## 3.1 Hire Bike

|  |
| --- |
| **Use Case: Hire Bike** |
| **Owner**: Customer |
| **Pre-Conditions** |
| 1. Existing Customer Record 2. Bike must be in stock 3. Enquire about prices, availability of bikes etc. |
| **Post-Conditions** |
| 1. Return bike after use |
| **Primary Path** |
| 1. Make a reservation for a bike(s) (by Class, Size) for a specific date and quantity of bikes required 2. Go into the store and tell staff in Hirings Department that you have a reservation. 3. Select your bike(s) 4. Take out bike(s) |
| **Alternate Path** |
| 1. Go into the store and ask if there are any bikes available to hire 2. If there are, tell staff in Hirings Department which bike class and size you want and how many you want 3. Select your bike(s) 4. Take out bike(s) |
| **Notes** |
| Pritam Sangani |

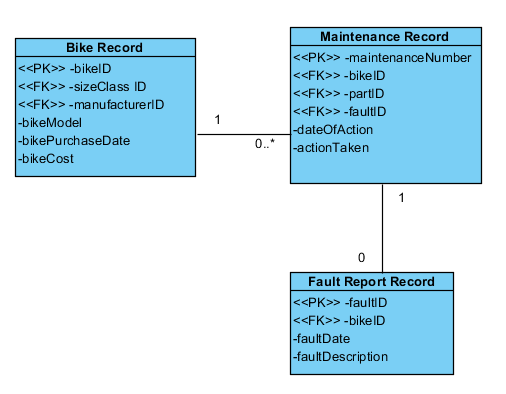
### 3.1.1 Hire Bike ERD



## 3.2 Update Maintenance Record

|  |
| --- |
| **Use Case: Update Maintenance Record** |
| Owner: Manager |
| **Pre-Conditions** |
| * Customer returns bike reporting a fault. * Manager services the bike finding a fault. |
| **Post-Conditions** |
| * After the bike fault(s) have been fixed by the technicians, the maintenance record is updated and the bike is ready to be hired out again. |
| **Primary Path** |
| * Customer returns bike reporting a fault with the bike. * The staff report back the fault to the technicians and make the bike unavailable. * The bike is then added to the maintenance record adding the fault with the bike. * The maintenance record is updated again once the bike has been fixed. |
| **Alternate Path** |
| * No fault is found, by the customer or the manger, with the bike. |
| **Notes** |
| Yusof Bandar |

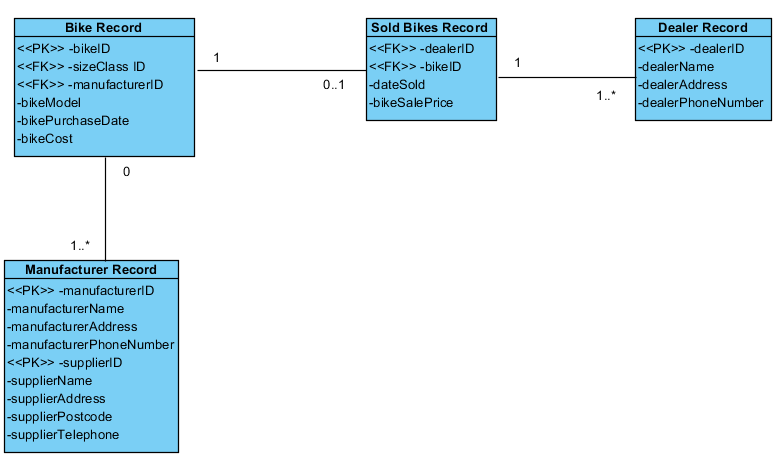
### 3.2.1 Update Maintenance Record ERD



## 3.3 Update Bike Record

|  |
| --- |
| **Use Case: Update Bike Record** |
| Owner: Manager |
| **Pre-Conditions** |
| 1. Existing bike record |
| **Post-Conditions** |
| N/A |
| **Primary Path** |
| 1. New bike is bought by the shop 2. Enter bikes details to the database 3. Update bike record 4. Sell bike 5. Update bike record |
| **Alternate Path** |
| 1. No new bikes registered 2. No bikes sold |
| **Notes** |
| Naim Ahmed |

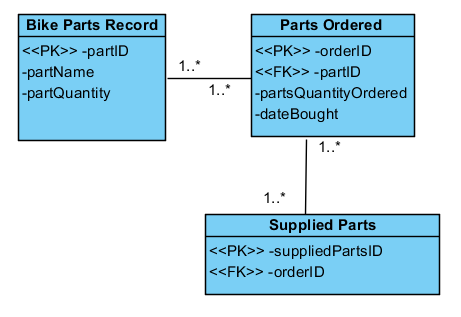
### 3.3.1 Update Bike Record ERD



## 3.4 Order Parts

|  |
| --- |
| **Use Case: Order Parts** |
| Owner: Parts Manager |
| **Pre-Conditions** |
| 1. Parts not in stock or low in stock 2. Create parts list 3. Find supplier of parts |
| **Post-Conditions** |
| 1. Delivery notice 2. Delivery invoice 3. Stock check |
| **Primary Path** |
| 1. Stock check what parts are low or out of stock 2. Produce a list of parts which need to be ordered 3. Order parts 4. Check all correct parts have arrived using invoice 5. Check for damaged parts |
| **Alternate Path** |
| 1. No parts needed |
| **Notes** |
| Aamer Atique |

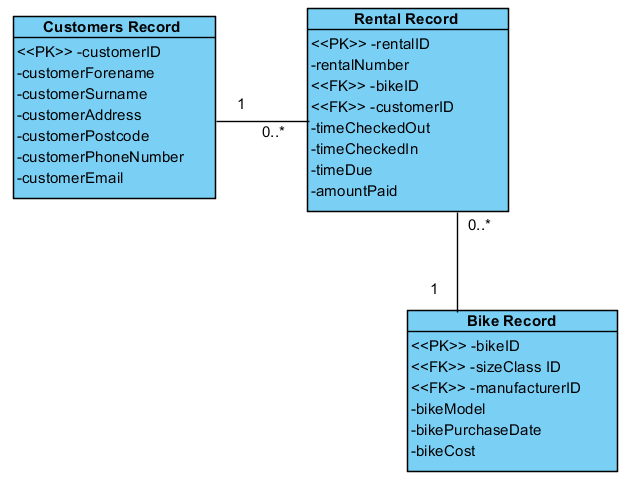
### 3.4.1 Order Parts ERD



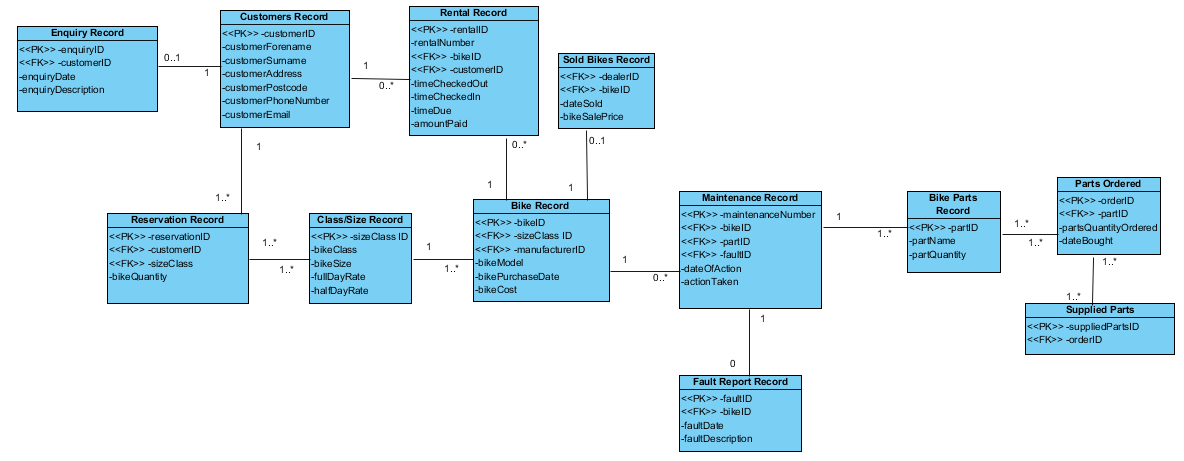
## 3.5 Update Rental Record

|  |
| --- |
| **Use Case: Update Rental Record** |
| Owner: Hirings Department |
| **Pre-Conditions** |
| 1. An existing Customer hires an existing bike(s) through the Hirings Department |
| **Post-Conditions** |
| 1. Return bike(s) after use |
| **Primary Path** |
| 1. Hirings Department finds the bike being hired using bikeID 2. Customer takes the bike 3. Rental Record is updated |
| **Alternate Path** |
| N/A. |
| **Notes** |
| Faran Azadi |

### 3.5.1 Update Rental Record ERD



# 4.0 Top-Down ERD



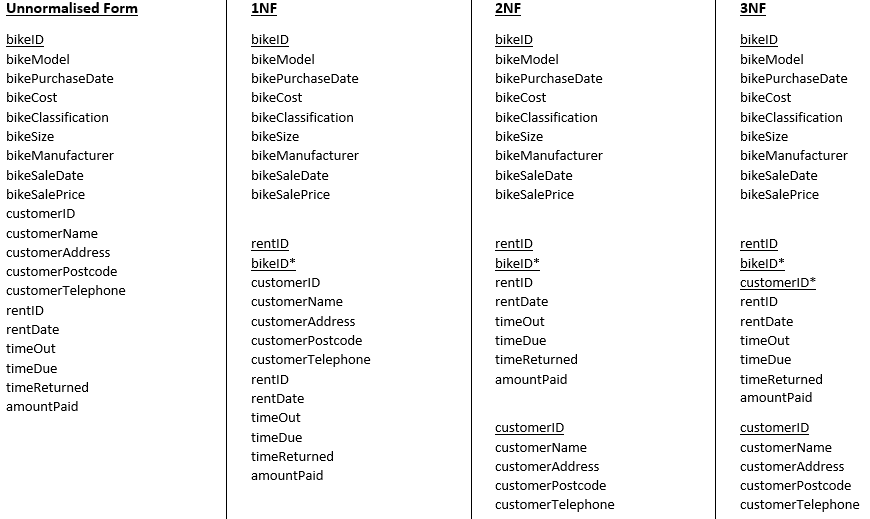
# 5.0 Bottom-Up ERD

## 5.1 RDAs

### 5.1.1 Maintenance and Dealer RDA



### 5.1.2 Rental and Bike Record RDA



### 5.1.3 Merged RDA

bikeID  
bikeModel  
bikePurchaseDate  
bikeCost  
bikeClassification  
bikeSize  
bikeSaleDate  
bikeSalePrice  
bikeManufacturer

rentID  
bikeID\*  
customerID\*  
rentDate  
timeOut  
timeDue  
timeReturned  
amountPaid

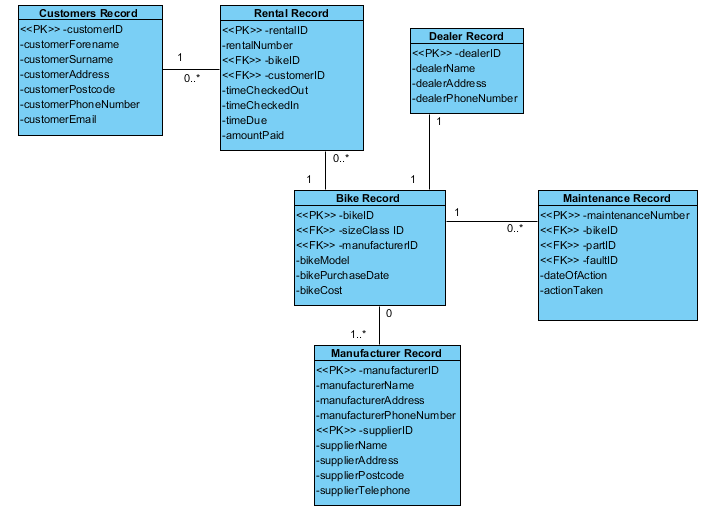
customerID  
customerName  
customerAddress  
customerPostcode  
customerTelephone

dealerID  
dealerName  
dealerAddress  
dealerTelephone

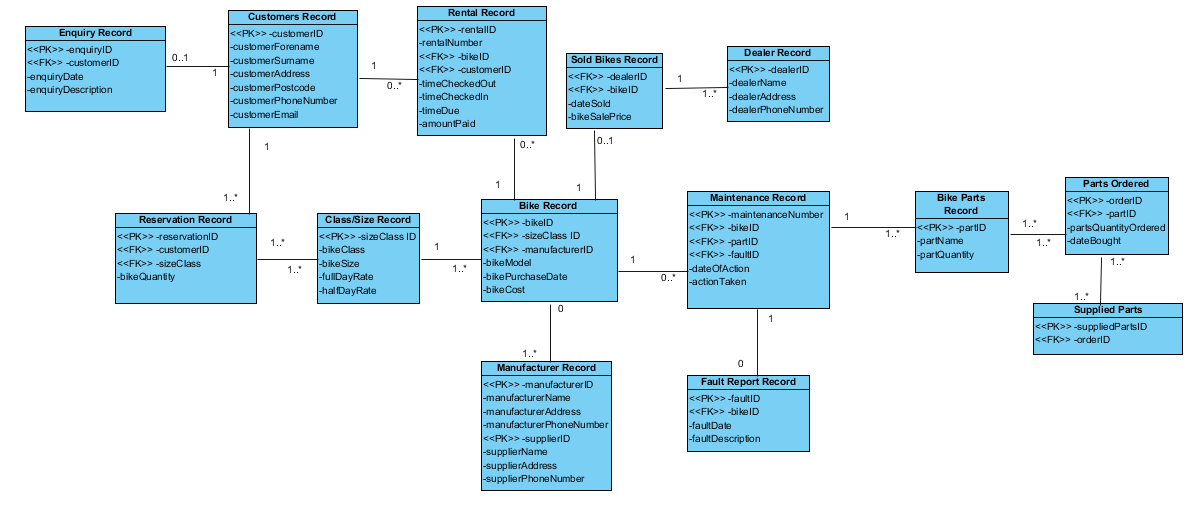
bikeID\*  
maintenanceID  
faultDetails  
faultDate  
actionTaken  
actionDate

manufacturerID  
manufacturerName  
manufacturerAddress  
manufacturerPostcode  
manufacturerTelephone

## 5.2 Bottom-Up ERD from Merged RDA



# 6.0 Final Merged ERD



## 6.1 ERD Commentary

There were many decisions which were made to create our final ERD. For example, for the enquiries record, we assumed that some customers may not enquire about bikes and may just come in and rent the bikes straight from the shop or may just make a reservation without enquiring. For the relationships between the entities, we went through different scenarios which could happen and decided which relationship would make sense. Going through many examples of different ERDs allowed us to get a better understanding of how they work and the relationships between the entities. For many to many relationships, we had to create new weak entities to split the relationships.

# 7.0 Appendix

## 7.1 Log Book

