**New Zealand Diploma in Information Systems**

**HTCS5607 IS Application Project**

**TECHNICAL REPORT TEMPLATE**

**Project Name: NZ HORSE RACING**

|  |  |  |  |
| --- | --- | --- | --- |
| **Project Team Name(s)** | **Student ID** | **Email** | **Phone** |
| ***Project Manager***  *Sascha Williams* | 1536995 | [Sjawilliams92@gmail.com](mailto:Sjawilliams92@gmail.com) | 02041697977 |
| ***Deputy Project Manager***  *[Name]* |  |  |  |
| ***Team Members***  *[Name (s)]* |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Client Stakeholders**

|  |  |  |
| --- | --- | --- |
| **Client Stakeholders** | **Full name and title** | **Contact details** |
| **Project Sponsor(s)** | *Lei Song, Simon Dacey* | *[phone, email]* |
| ***[add other roles as appropriate]*** | *[full name and title]* | *[phone, email]* |

**DATE OF SUBMISSION**

*25/11/2021*

Table of Contents

[1. Document Control 4](#_Toc80799791)

[1.1 Version History 4](#_Toc80799792)

[1.2 Contribution to Report sections 4](#_Toc80799793)

[1.3 Glossary 4](#_Toc80799794)

[2. Executive Summary 5](#_Toc80799795)

[3. Introduction 6](#_Toc80799796)

[4. Technology Review 7](#_Toc80799797)

[5. IT Methodology 8](#_Toc80799798)

[6. Project Management 9](#_Toc80799799)

[6.1 Project Management Narrative 9](#_Toc80799800)

[6.2 Project Plan with Milestones 9](#_Toc80799801)

[6.3 Project Governance Responsibilities 9](#_Toc80799802)

[6.4 Project Meetings 9](#_Toc80799803)

[6.5 Project Reports 9](#_Toc80799804)

[6.6 Project Risk and Issue Analysis 9](#_Toc80799805)

[7. Requirements Analysis 10](#_Toc80799806)

[7.1 Introduction 10](#_Toc80799807)

[7.2 Use Case Diagram 10](#_Toc80799808)

[7.3 Business Use Case Narratives (Descriptions) 10](#_Toc80799809)

[7.4 Activity Diagrams 10](#_Toc80799810)

[7.5 Overall Class Diagram 10](#_Toc80799811)

[8. Project Design 11](#_Toc80799812)

[8.1 Introduction 11](#_Toc80799813)

[8.2 Software List 11](#_Toc80799814)

[8.3 Version Control Software 11](#_Toc80799815)

[8.4 Design Use Case Narratives (Descriptions) 11](#_Toc80799816)

[8.5 Sequence Diagrams 11](#_Toc80799817)

[8.6 Deployment Diagram 11](#_Toc80799818)

[8.7 Database Design 11](#_Toc80799819)

[8.8 Annotated User Interface Designs 11](#_Toc80799820)

[8.9 Test Plan 11](#_Toc80799821)

[9. Project Training 12](#_Toc80799822)

[9.1 End User Background and Training Objectives 12](#_Toc80799823)

[9.2 Training Materials 12](#_Toc80799824)

[9.3 Training Deliverables 12](#_Toc80799825)

[9.4 Evaluation 12](#_Toc80799826)

[10. Conclusion & Lessons Learned 13](#_Toc80799827)

[References 14](#_Toc80799828)

[Appendices 15](#_Toc80799829)

# 1. Document Control

## 1.1 Version History

This document has had the following revisions:

| **Version** | **Date** | **Author** | **Description of Change** |
| --- | --- | --- | --- |
| 0.1 |  |  | Initial draft |

## 1.2 Contribution to Report sections

| **Project Team Member name** | **Student ID** | **Report Section** |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

## 1.3 Glossary

To provide clarity, terms and acronyms used in this document are defined as follows:

| **Term / Abbreviation** | **Definition** |
| --- | --- |
| Supervisor | Technical Advisor |
|  |  |

# 2. Executive Summary

This is a summary about NZ HORSE RACING and their ambitions. NZ HORSE RACING is an agency that specializes in setting up horse race meetings. With intensive research, there are not many agencies that have the sort of setup that NZ HORSE RACING are aiming to produce. The goal that NZ HORSE RACING want, is a program that ministers their bookings and meetings. They want a easy to use program for their staff and also for non - technical people who are new to the job and can easily be taught on how to use this program. This program will go through several stages of development before its finally produced.

# 3. Introduction

NZ HORSE RACING is a horse racing agency and specializes in setting up and manages horse racing meetings. The administrative team at NZ HORSE RACING want a program built that manages their horse racing meetings and bookings. The program they want also has to be easy to use for non-technical users.

# 4. Technology Review

I have reviewed some sites that are based around HORSE RACING. Some technologies that are used on some websites are targeted to horse betting and Horse management. There are more generic applications such as race management, venue management and event (including sports) management.

<http://racemanager.com/> - Racemanager uses management software for their users.

<https://www.equinegenie.com/> - Equinegenie uses management software, training software, event software, conducts business operations and more. Some similarities to NZ HORSE RACING is probably the management software.

<https://www.pace-view.net/> -. Pace-view created a software that gives stats about the races and horses and sends those stats through emails and newsletters. Very similar to NZ HORSE RACING that print owner reports, racing reports and so on through its services.

<https://hello.raklet.com/> - Hello.raklet uses a variety of software. This includes management software, creating databases to store contacts, monthly updated news letters and reminders sent to your email and orgranising events.

# 5. IT Methodology

The methodology that’s been picked to use for the project is the waterfall methodology. The project manager chose to use this because of the simplicity that it offers, easy to keep on track and also makes every task easy.

# 6. Project Management

## 6.1 Project Management Narrative

*Details with evidence how the development of the project followed the selected systems development lifecycle*

*SDLC*

*1 Plan – We created a Gantt chart for marking progress, created a Technical Review as well as covering issues.*

*2 Analysis – In analysis, we developed diagrams as well as situational descriptions for the project.*

*3 Design – In design, we developed the plans for the database as well as annotated interface plans for coding.*

*4 Implementation – With implementation, we created the code following the plans that were written in the design phase of the project.*

*5 Maintain – With maintenance, the application will be looked at once every month or so just to see if It’s functionality is working properly and doing its job.*

## 6.2 Project Plan with Milestones

*Include an overall plan here and attach a detailed GANTT chart to the appendices*

## 6.3 Project Governance Responsibilities

*The project was handled by Sascha Williams and project itself was broken down into pieces and was allocated between the team members.*

## 6.4 Project Meetings

*Include a schedule of your meetings (date, duration, participants, and type) and attach the minutes of each meeting to the appendices*

## 6.5 Project Reports

*Discuss the project status reports and attach your project status reports to the appendices*

## 6.6 Project Risk and Issue Analysis

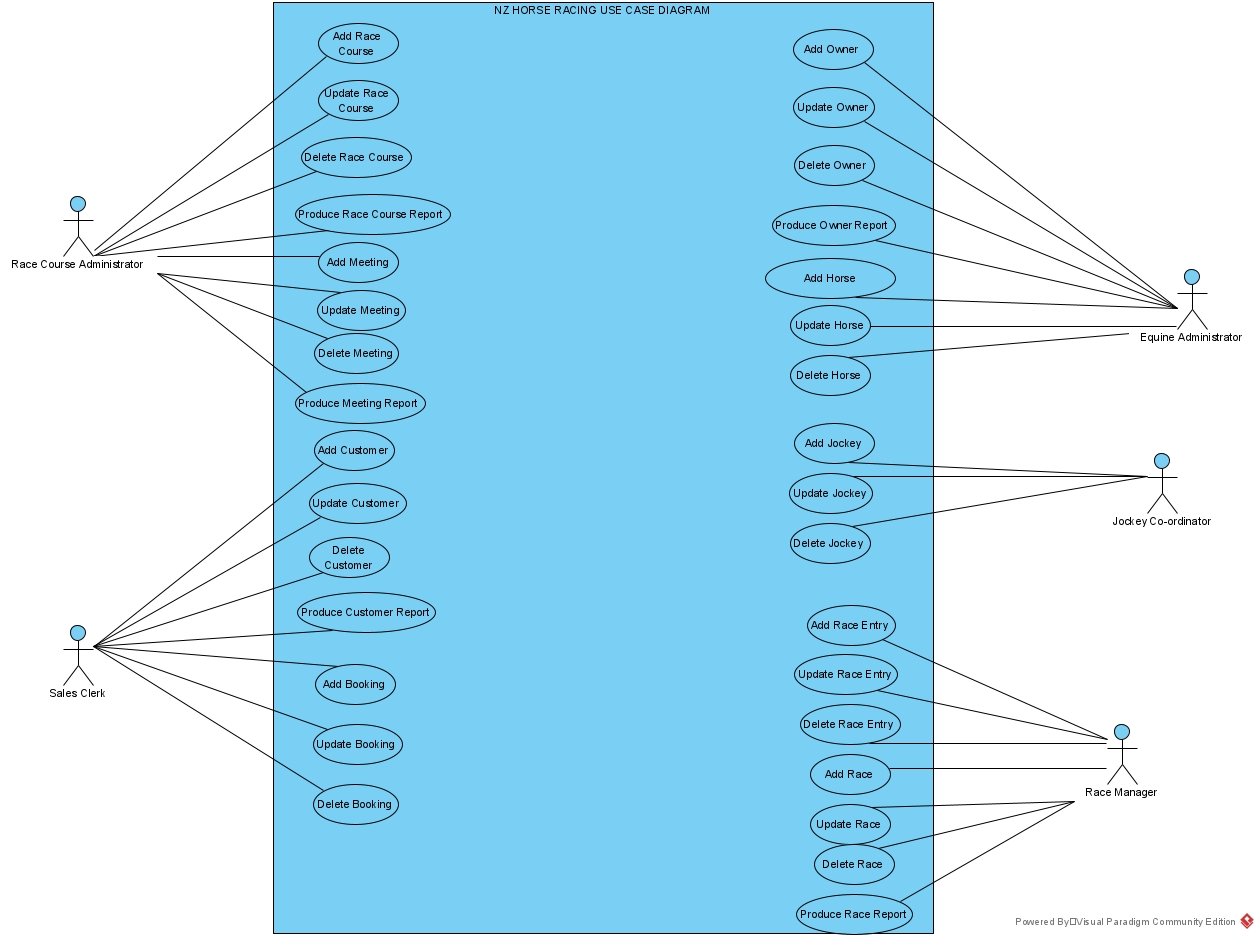
*Discuss project risks and issues and attach your project risk and issue register to the appendices*

# 7. Requirements Analysis

## 7.1 Introduction

*In the Requirement analysis we’ve created use case diagram which shows the staff relationship between them and the role they’re in within the agency, situational use case narratives as well as activity diagrams which show how system works and a overall class diagram or Entity Relationship Diagram, which shows the relationships and their attributes between entities.*

## 7.2 Use Case Diagram



## 7.3 Business Use Case Narratives (Descriptions)

|  |  |  |
| --- | --- | --- |
| **USE CASE NAME:** | Add Booking | **USE CASE TYPE** |
| **USE CASE ID:** | 16 | **Business Requirements: 🗹** |
| **PRIORITY:** | High |  |
| **PRIMARY BUSINESS ACTOR:** | Sales Clerk | |
| **OTHER PARTICIPATING ACTORS:** |  | |
| **DESCRIPTION:** | Enables the Sales Clerk to add a booking for a customer for a selected meeting. | |
| **PRE-CONDITIONS:** | The Sales Clerk has logged onto the system | |
| **TYPICAL COURSE** | 1. The Sales Clerk selects the “Add Booking” function 2. The system displays the “Add Booking” form with all fields blank. 3. The system displays a list of customers (customer id, last name, and first name) with a valid credit status. 4. The Sales Clerk selects a customer. 5. The system displays a list of meetings (meeting id and meeting name). 6. The Sales Clerk enters the booking’s details (booking id (1 to 9999999999) quantity (1 to 4 inclusive), and status (pending or paid)). 7. The system either confirms that the details are filled in correctly or identifies that the details are missing or incorrect and prompts for the completion of the details. 8. The Sales Clerk selects add the booking. 9. The system saves the booking’s details (booking id, customer id, meeting id, booking date: today’s date, quantity, and status). 10. The system displays the “Booking added successfully” message. 11. The system displays the “Add another booking?” prompt. 12. The Sales Clerk clicks on Return Button. 13. Add Booking Form closes. | |
| **OF EVENTS:** |
|  |
|  |
|  |
|  |
| **ALTERNATE COURSES:** | 8a1. Sales Clerk selects Return  8a2. Form closes | |
| 11a1. Sales Clerk selects “Add another booking”  11a2. Goes to Step 2. | |
| **POST CONDITIONS:** |  | |
| **ASSUMPTIONS:** |  | |

|  |  |  |
| --- | --- | --- |
| **USE CASE NAME:** | Add Horse | **USE CASE TYPE** |
| **USE CASE ID:** | 5 | **Business Requirements: 🗹** |
| **PRIORITY:** | High |  |
| **PRIMARY BUSINESS ACTOR:** | Equine Administrator | |
| **OTHER PARTICIPATING ACTORS:** |  | |
| **DESCRIPTION:** | Enables the Equine Administrator to add a Horse and its details. | |
| **PRE-CONDITIONS:** | The Equine Administrator has logged onto the system | |
| **TYPICAL COURSE** | 1. The Equine Administrator selects the “Add Horse” function. 2. The system displays the “Add Horse” form with all fields blank. 3. The Equine Administrator enters the horse’s details (horse id (1 to 999999 inclusive), horse name, gender (male, female or gelding), and date of birth). 4. The system validates that the details are filled in correctly or identifies that the details are missing or incorrect and prompts for the completion of the details. 5. The system displays a list of owners (owner id, last name, and first name). 6. The Equine Administrator selects an owner. 7. The Equine Administrator selects Add Horse 8. The system Saves the horses details (horse id, horse name, gender, date of birth, and owner id). 9. The system displays the “Horse added successfully” message. 10. The system displays the “Add another horse?” prompt. 11. The Equine Administrator clicks on Return Button 12. Add Horse Form closes. | |
| **OF EVENTS:** |
|  |
|  |
|  |
|  |
| **ALTERNATE COURSES:** | 7a1. Equine Administrator select Return  7a2. Form closes | |
| 10a1. Equine Administrator selects Add another Horse  10a2. Goes back to step 2 | |
| **POST CONDITIONS:** |  | |
| **ASSUMPTIONS:** |  | |

|  |  |  |
| --- | --- | --- |
| **USE CASE NAME:** | Add Race Entry | **USE CASE TYPE** |
| **USE CASE ID:** | 26 | **Business Requirements: 🗹** |
| **PRIORITY:** | High |  |
| **PRIMARY BUSINESS ACTOR:** | Race Manager | |
| **OTHER PARTICIPATING ACTORS:** |  | |
| **DESCRIPTION:** | Enables Race Manager to add a race entry to a selected race | |
| **PRE-CONDITIONS:** | The Race Manager has logged onto the system | |
| **TYPICAL COURSE** | 1. The Race Manager selects the “Add Race Entry” function. 2. The system displays the “Add Race Entry” form. 3. The system displays a list of all the scheduled races (race id and race name). 4. The Race Manager selects the race to which the entry will be added. 5. The system displays the race’s details (race id, race name, and meeting name). 6. The system displays the list of the race entries (horse name, jockey last name, and jockey first name) already added to the selected race. 7. The system displays a list of the horses (horse id and horse name). 8. The Race Manager selects a horse. 9. The system displays the error message “The horse has already entered this race” if the selected horse is already entered into the selected race and then displays a list of the horses (horse id and horse name) 10. The system displays a list of the jockeys (jockey id, last name, and first name) 11. The Race Manager selects a jockey. 12. The system displays the error message “The jockey has already entered this race on another horse” if the selected jockey is already entered into the selected race and then displays a list of the jockeys (jockey id, last name, and first name) 13. The Race Manager enters the status (pending, disqualified, or confirmed) of the race entry for the race. 14. The Race Manager selects add the race entry to the race 15. The system generated a unique race entry id (1-9999999999) for the race entry. 16. The system saves the race entry details (race entry id, race id, horse id, jockey id, status, and race entry time (set to zero)). 17. The system displays the “Race entry added to race successfully” message. 18. The system displays the “Add another race entry?” prompt. 19. The Race Manager clicks on Return Button 20. Add Race Entry Form closes. | |
| **OF EVENTS:** |
|  |
|  |
|  |
|  |
| **ALTERNATE COURSES:** | 4a1. Race manager selects Return  4a2. Form closes | |
| 14a1. Race Manager selects Return  14a2. Form closes | |
|  | 18a1. Race Manager selects to add another race  18a2. System goes back to step 2 | |
| **POST CONDITIONS:** |  | |
| **ASSUMPTIONS:** |  | |

|  |  |  |
| --- | --- | --- |
| **USE CASE NAME:** | Delete Booking | **USE CASE TYPE** |
| **USE CASE ID:** | 19 | **Business Requirements: 🗹** |
| **PRIORITY:** | High |  |
| **PRIMARY BUSINESS ACTOR:** | Sales Clerk | |
| **OTHER PARTICIPATING ACTORS:** |  | |
| **DESCRIPTION:** | Enables Sales Clerk to delete a selected bookings details. | |
| **PRE-CONDITIONS:** | The Sales Clerk has logged onto the system | |
| **TYPICAL COURSE** | 1. The Sales Clerk selects the “Delete Booking” function. 2. The system displays the “Delete Booking” form. 3. The system displays a list of all the bookings (booking id and booking date). 4. Sales Clerk selects the booking that requires deleting. 5. The system displays the booking’s details (booking id, booking date, quantity, status, customer id, customer last & first name, and meeting name). 6. Sales Clerk selects Deleting the booking. 7. The system displays the “Are you sure you want to delete this booking?” prompt. 8. Sales Clerk confirms deletion 9. The system deletes the booking’s details. 10. The system displays the “Booking deleted successfully” message. 11. The system displays the “Delete another booking?” prompt. 12. The Sales Clerk clicks on Return Button 13. Delete Booking Form closes. | |
| **OF EVENTS:** |
|  |
|  |
|  |
|  |
| **ALTERNATE COURSES:** | 4a1. Sales Clerk select Return  4a2. Form closes. | |
| 6a1. Sales Clerk select Return  6a2. Form closes. | |
| 8a1. Sales Clerk select another booking  8a2. Goes back to step 3 | |
| 12a1. Sales Clerk select to delete another booking  12a2 Goes back to step 3 | |
| **POST CONDITIONS:** |  | |
| **ASSUMPTIONS:** |  | |

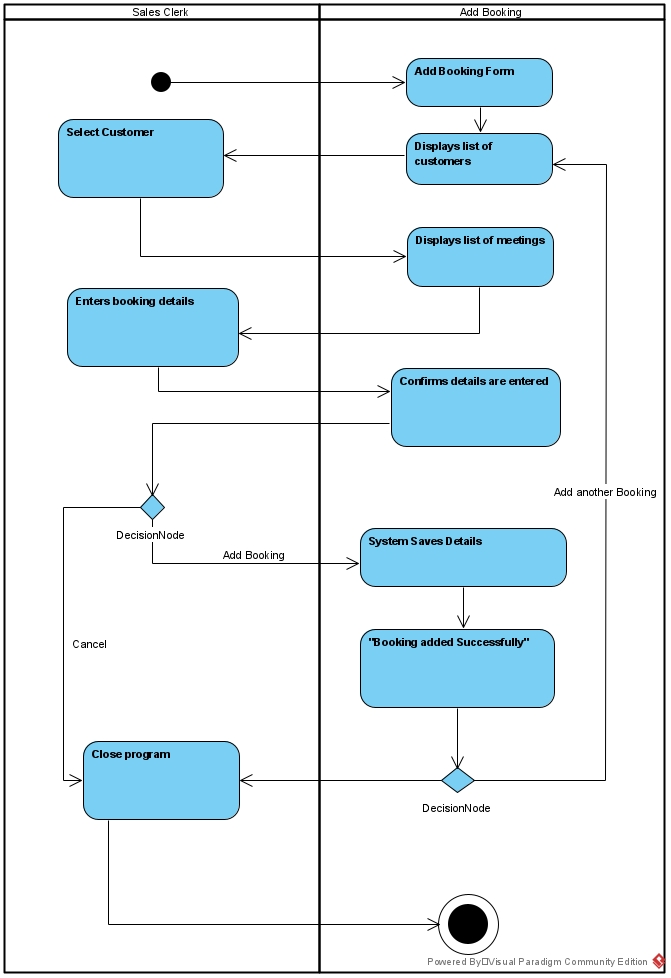
|  |  |  |
| --- | --- | --- |
| **USE CASE NAME:** | Delete Horse | **USE CASE TYPE** |
| **USE CASE ID:** | 22 | **Business Requirements: 🗹** |
| **PRIORITY:** | High |  |
| **PRIMARY BUSINESS ACTOR:** | Equine Administrator | |
| **OTHER PARTICIPATING ACTORS:** |  | |
| **DESCRIPTION:** | Enables Equine Administrator to delete a selected horse’s details | |
| **PRE-CONDITIONS:** | The Equine Administrator has logged onto the system | |
| **TYPICAL COURSE** | 1. The equine administrator selects the “Delete Horse” function. 2. The system displays the “Delete Horse” form. 3. The system displays a list of all the horses (horse id and horse name) that have no race entries. 4. Equine Administrator selects horse 5. The system displays the horse’s details (horse id, horse name, date of birth, and owner last name and first name). 6. Equine Administrator selects delete Horse 7. The system displays the “Are you sure you want to delete this horse?” prompt. 8. Equine Administrator confirms deletion 9. The system deletes all of the race details associated with the horse. 10. The system deletes the horse’s details. 11. The system displays the “Horse deleted successfully” message. 12. The system displays the “Delete another horse?” prompt. 13. Equine Administrator clicks on Return Button 14. Delete Horse Form closes. | |
| **OF EVENTS:** |
|  |
|  |
|  |
|  |
| **ALTERNATE COURSES:** | 4a1. Equine Administrator selects Return  4a2. Form closes | |
| 6a1. Equine Administrator selects Return  6a2. Form closes | |
|  | 8a1. Equine Administrator can Select a Different Horse  8a2. Goes back to step 3 | |
|  | 12a1. Equine Administrator can Select another Horse to delete  12a2. Goes back to step 3 | |
| **POST CONDITIONS:** |  | |
| **ASSUMPTIONS:** |  | |

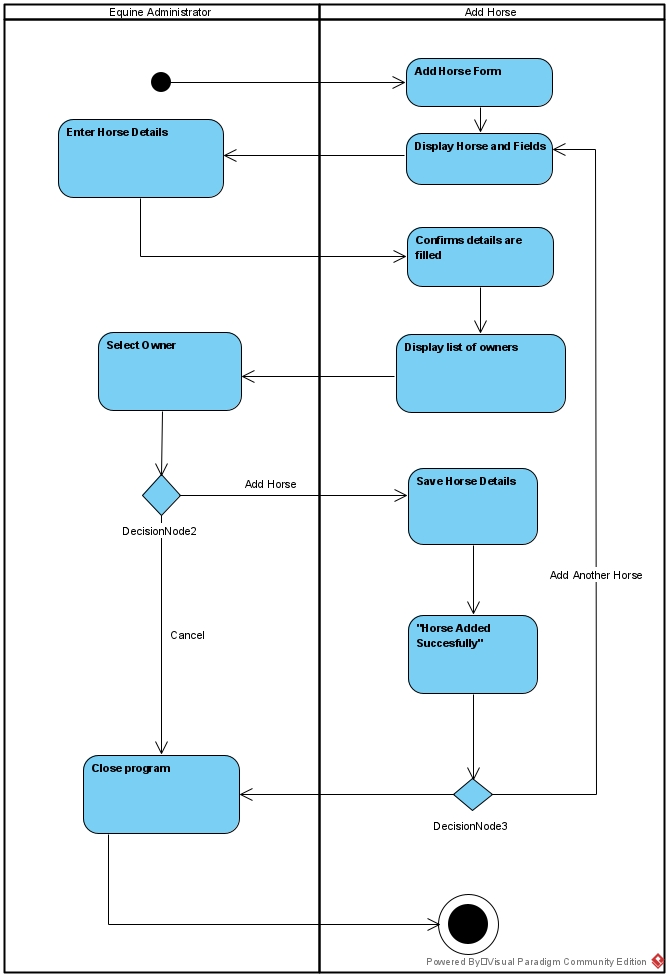
|  |  |  |
| --- | --- | --- |
| **USE CASE NAME:** | Delete Race Entry | **USE CASE TYPE** |
| **USE CASE ID:** | 28 | **Business Requirements: 🗹** |
| **PRIORITY:** | High |  |
| **PRIMARY BUSINESS ACTOR:** | Race Manager | |
| **OTHER PARTICIPATING ACTORS:** |  | |
| **DESCRIPTION:** | Enables Race Manager to delete a selected race | |
| **PRE-CONDITIONS:** | Race Manager has logged onto the system | |
| **TYPICAL COURSE** | 1. The Race Manager selects the “Delete Race Entry” function. 2. The system displays the “Delete Race Entry” form 3. The system displays a list of all the scheduled races (race id and race name) that have race entries. 4. The Race Manager selects the race with the race entry which is being deleted. 5. The system displays the race’s details (race id, race name, and status). 6. The system displays the list of the race entries (race entry id, horse name, jockey last name, and jockey first name) associated with the selected race. 7. The Race Manager selects a race entry. 8. The system displays the details of the race entry (race entry id, horse name, jockey last name, jockey first name, status, and race entry time). 9. The Race Manager selects to delete the race entry 10. The system deletes race entry’s details. 11. The system displays the “Race entry deleted successfully” message. 12. The system displays the “Delete another race entry?” prompt. 13. The Race Manager clicks on Return Button 14. Delete Race Entry Form closes. | |
| **OF EVENTS:** |
|  |
|  |
|  |
|  |
| **ALTERNATE COURSES:** | 4a1. Race Manager selects Return  4a2. Form closes | |
| 9a1. Race Manager selects Return  9a2. Form closes | |
|  | 12a1. Race Manager selects to delete another race entry  12a2. System goes to step 2. | |
| **POST CONDITIONS:** |  | |
| **ASSUMPTIONS:** |  | |

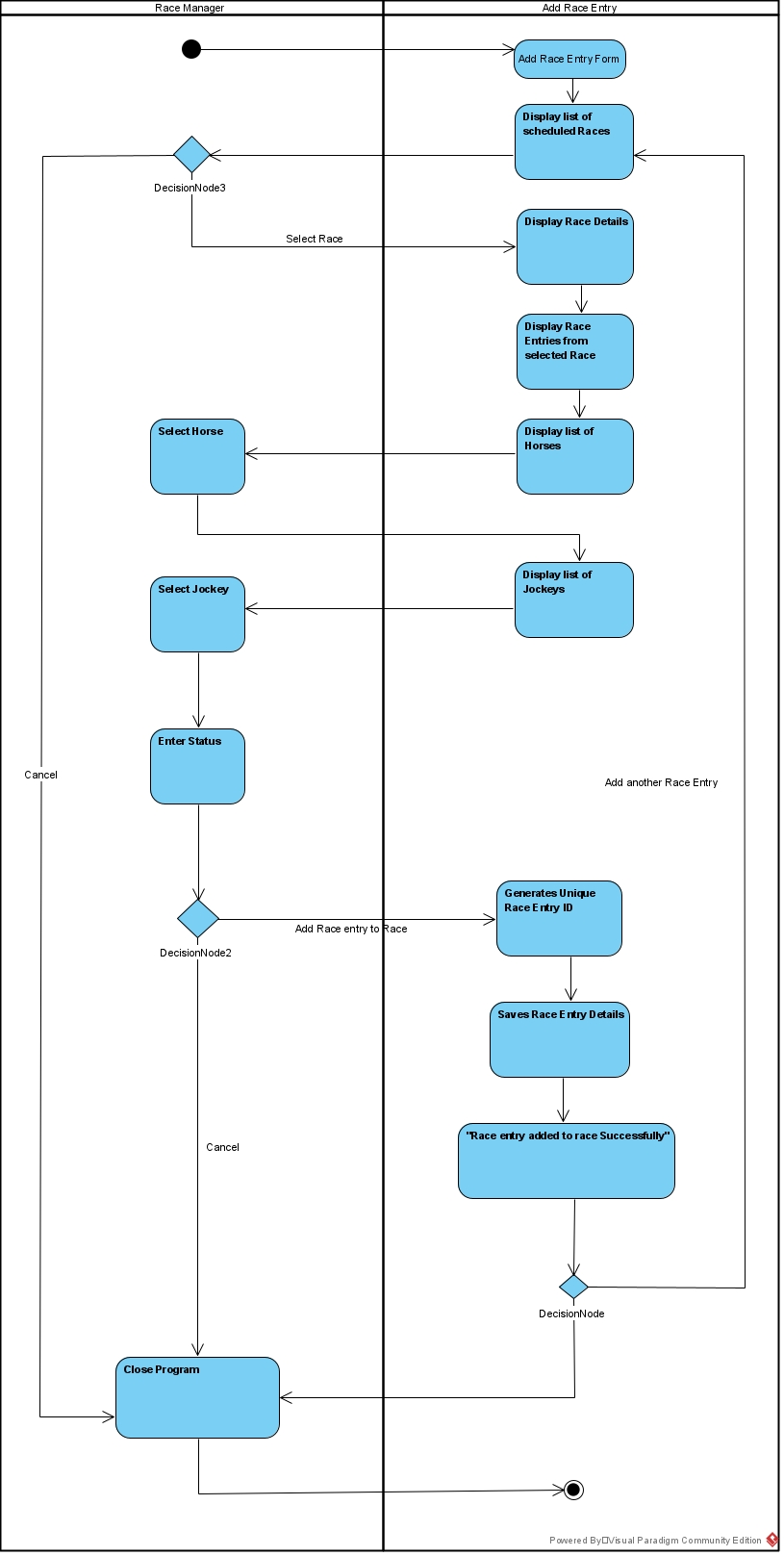
|  |  |  |
| --- | --- | --- |
| **USE CASE NAME:** | Update Booking | **USE CASE TYPE** |
| **USE CASE ID:** | 18 | **Business Requirements: 🗹** |
| **PRIORITY:** | High |  |
| **PRIMARY BUSINESS ACTOR:** | Sales Clerk | |
| **OTHER PARTICIPATING ACTORS:** |  | |
| **DESCRIPTION:** | Enables the Sales Clerk to update a selected bookings details. | |
| **PRE-CONDITIONS:** | The Sales Clerk has logged onto the system | |
| **TYPICAL COURSE** | 1. The Sales Clerk selects the “Update Booking” function. 2. The system displays the “Update Booking” form 3. The system displays a list of all the pending bookings (booking id and booking date). 4. The Sales Clerk selects a booking that has details that needs updating. 5. The system displays the booking’s details (booking id, booking date, quantity, status, customer id, customer last & first name, and meeting name). 6. The Sales Clerk updates the relevant details (quantity and status only). 7. The system validates the entries in the fields and either prompts for confirmation to change the booking’s details or identifies missing or incorrect fields and prompts for completion of the entry. 8. The Sales Clerk confirms the change of details. 9. The system saves the booking’s details. 10. The system displays the “Booking updated successfully” message. 11. The system displays the “Update another booking?” prompt. 12. The Sales Clerk clicks on Return Button 13. Update Booking Form closes. | |
| **OF EVENTS:** |
|  |
|  |
|  |
|  |
| **ALTERNATE COURSES:** | 4a. Sale Clerk selects Return  4a1. Form closes | |
| 8a1. Sale Clerk cancels the changes  8a2. Goes to step 12  8a3. Close Program | |
| 11a. Sale Clerk selects to update another booking  11a1. Goes back to step 2. | |
| **POST CONDITIONS:** |  | |
| **ASSUMPTIONS:** |  | |

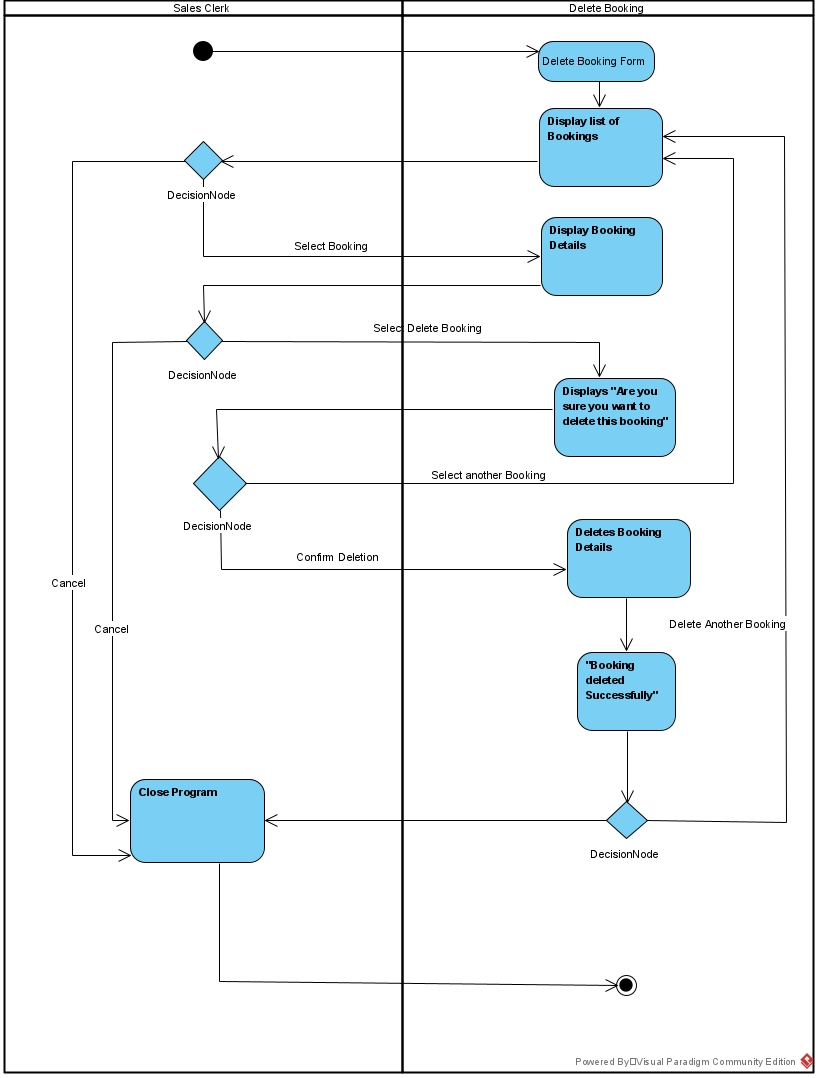
|  |  |  |
| --- | --- | --- |
| **USE CASE NAME:** | Update Horse | **USE CASE TYPE** |
| **USE CASE ID:** | 21 | **Business Requirements: 🗹** |
| **PRIORITY:** | High |  |
| **PRIMARY BUSINESS ACTOR:** | Equine Administrator | |
| **OTHER PARTICIPATING ACTORS:** |  | |
| **DESCRIPTION:** | Enables the Equine Administrator to update a selected horse’s details | |
| **PRE-CONDITIONS:** | The Equine Administrator has logged onto the system | |
| **TYPICAL COURSE** | 1. The Equine Administrator selects the “Update Horse” function. 2. The system displays the “Update Horse” form 3. The system displays a list of all the horses (horse id and horse name). 4. Equine Administrator selects horse that needs it details updated. 5. The system displays the horse’s details (horse id, horse name, gender, date of birth, and owner last name and first name). 6. The Equine Administrator updates the relevant details (horse name, horse date, gender, and date of birth only). 7. The system checks if fields are filled in if not the system prompts for completion of the entry. 8. The Equine Administrator confirms change of details 9. The system saves the horse’s details. 10. The system displays the “Horse updated successfully” message. 11. The system displays the “Update another horse?” prompt. 12. Equine Administrator clicks on Return Button 13. Update Horse Form closes. | |
| **OF EVENTS:** |
|  |
|  |
|  |
|  |
| **ALTERNATE COURSES:** | 4a. Equine Administrator selects Return  4a1. Form closes | |
| 8a. Equine Administrator selects Return  8a1. Form closes | |
| 11a. Equine Administrator selects update another horse.  11a1. Goes back to step 2 | |
| **POST CONDITIONS:** |  | |
| **ASSUMPTIONS:** |  | |

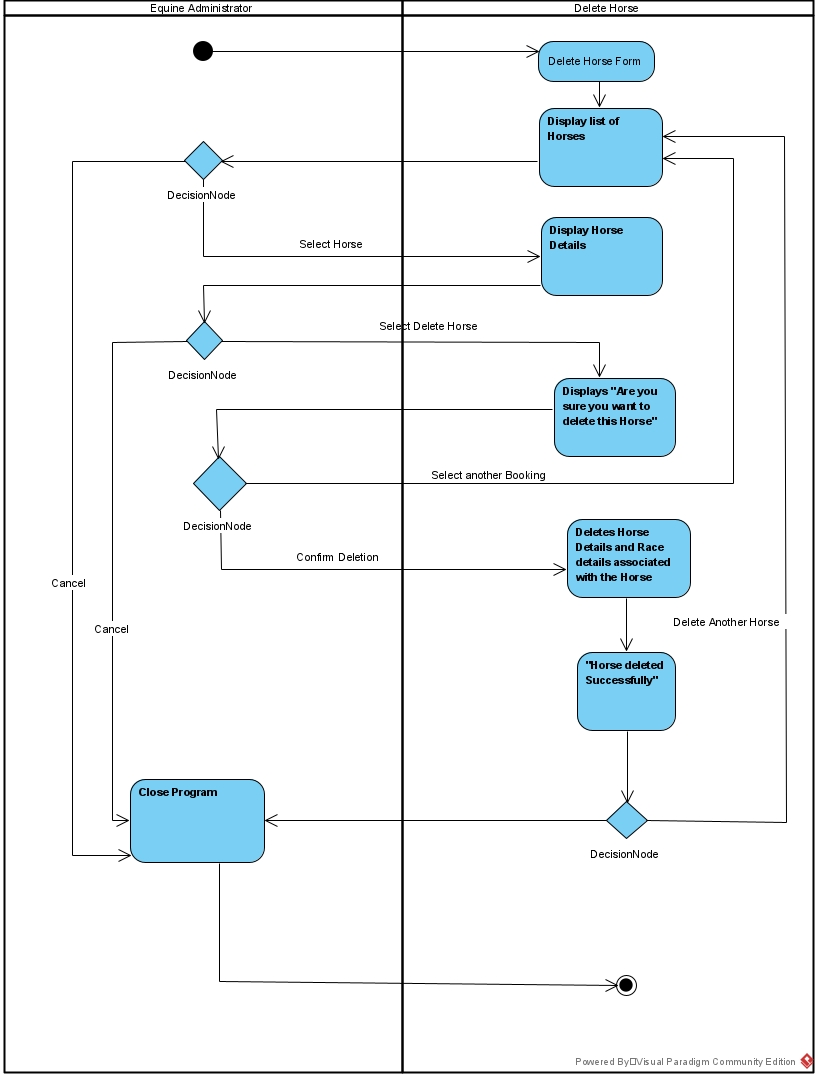
## 7.4 Activity Diagrams

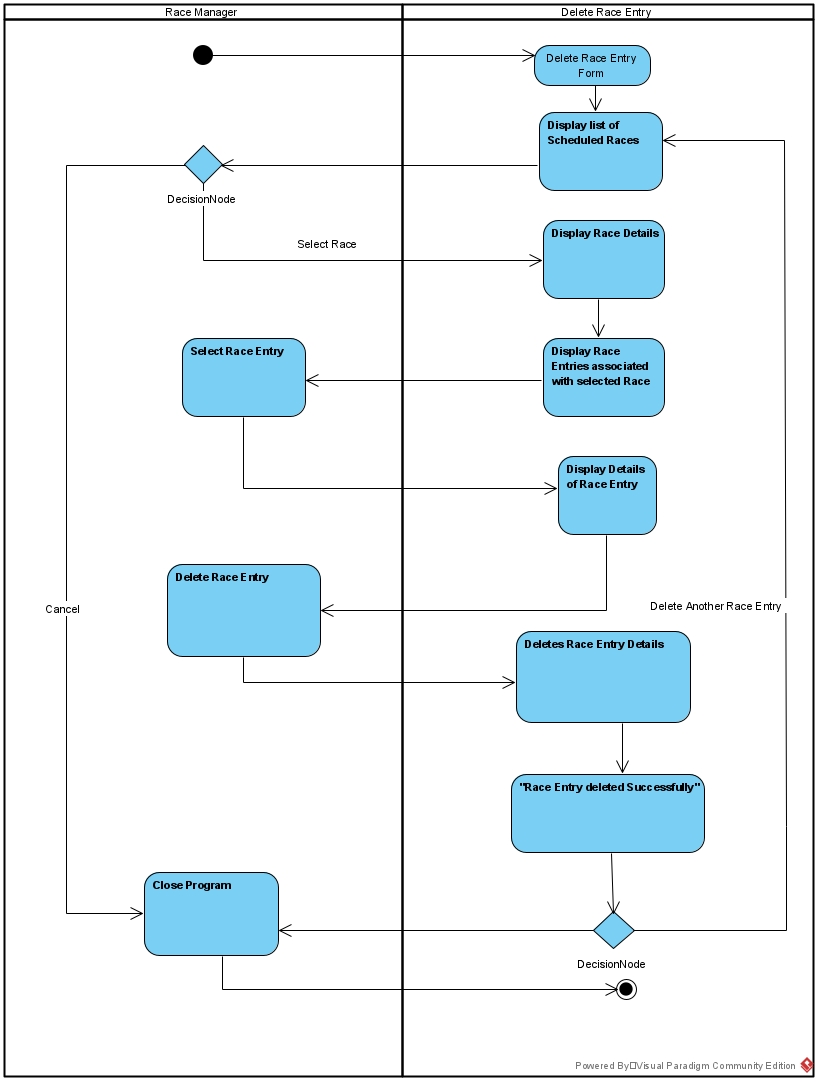
**

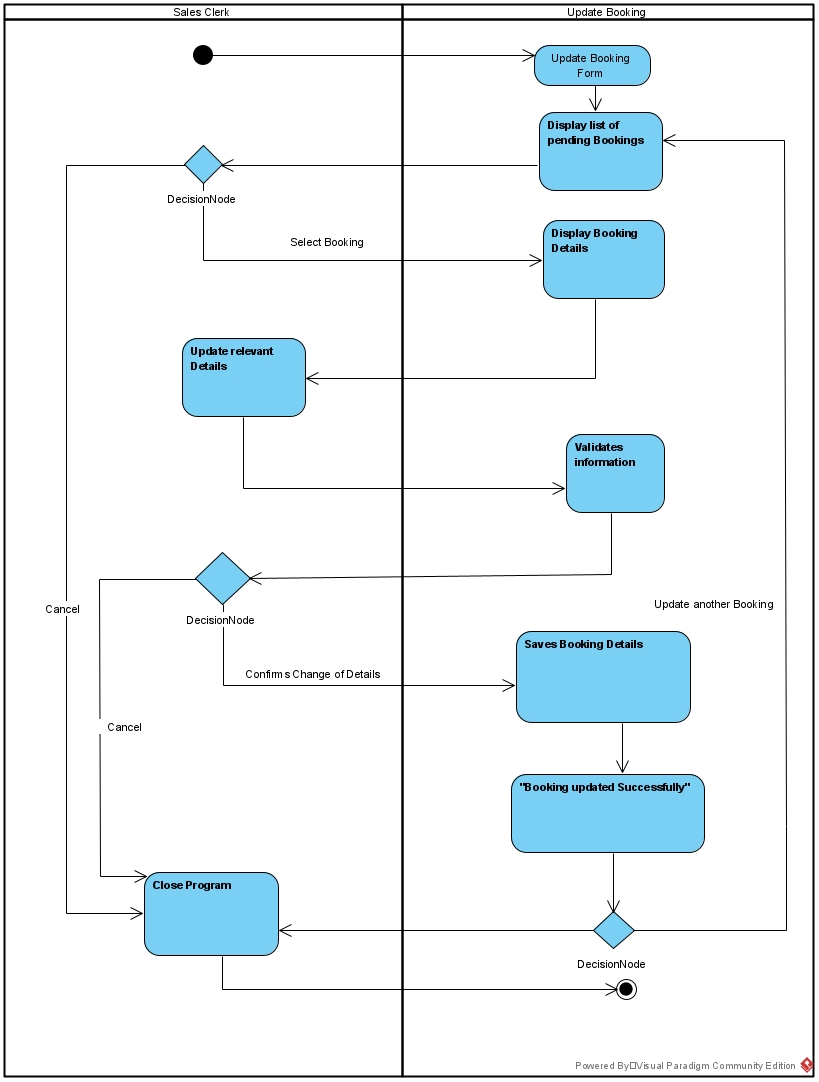
**

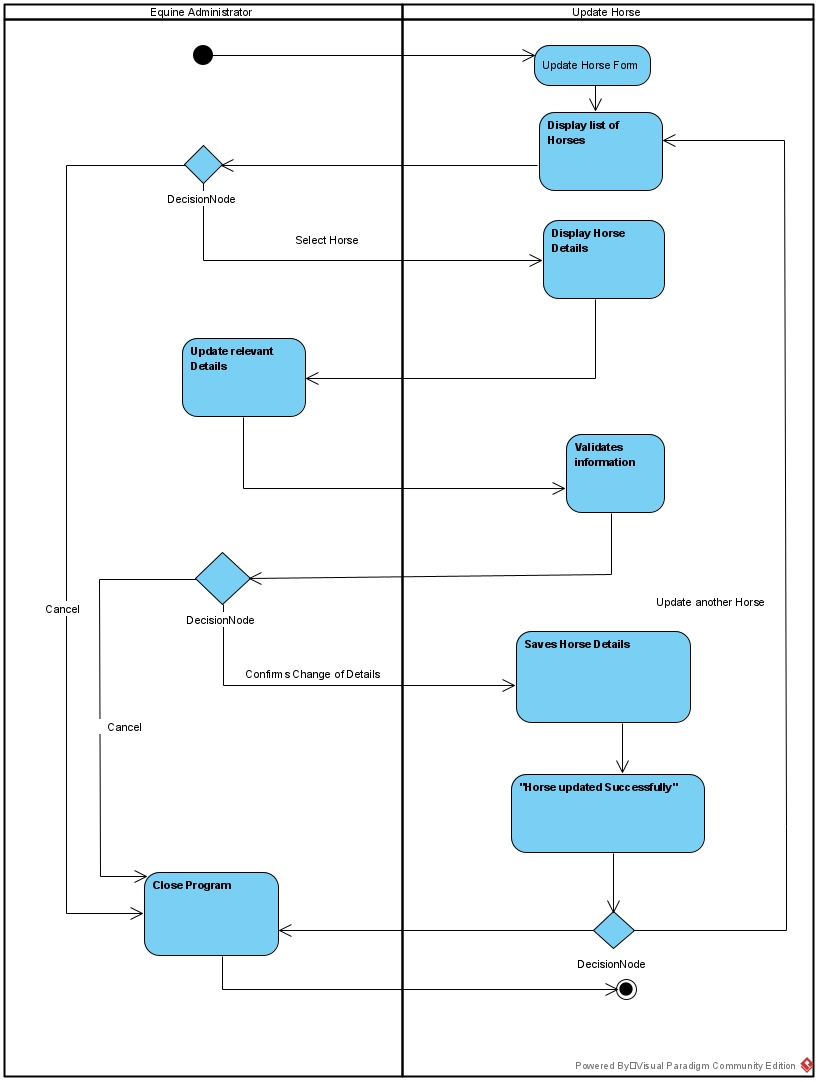
**

**

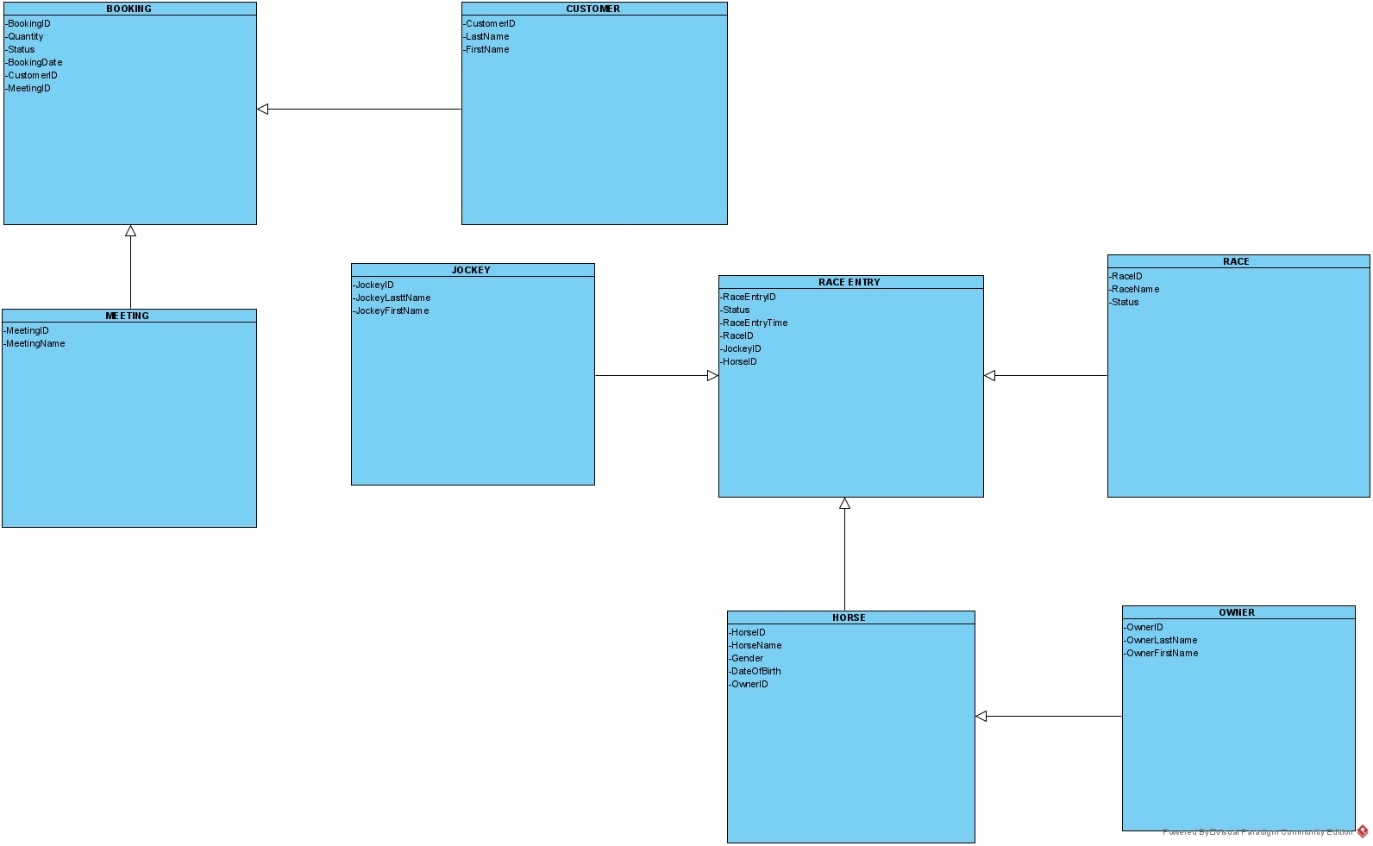
**

**

**

**

## 7.5 Overall Class Diagram



# 8. Project Design

## 8.1 Introduction

*The project design phase is where we started putting all the parts together ready for implementation.*

## 8.2 Software List

- MS Access

- MS Excel

- Visual Studio 2019

## 8.3 Version Control Software

*- Git Hub*

## 8.4 Design Use Case Narratives (Descriptions)

|  |  |  |
| --- | --- | --- |
| **USE CASE NAME:** | Add Booking | **USE CASE TYPE** |
| **USE CASE ID:** | 16 | **Design Requirements: 🗹** |
| **PRIORITY:** | High |  |
| **PRIMARY BUSINESS ACTOR:** | Sales Clerk | |
| **OTHER PARTICIPATING ACTORS:** |  | |
| **DESCRIPTION:** | Enables the Sales Clerk to add a booking for a customer for a selected meeting. | |
| **PRE-CONDITIONS:** | The Sales Clerk has logged onto the system | |
| **TYPICAL COURSE** | 1. The Sales Clerk selects the “Add Booking” function 2. The system displays the “Add Booking” form with all fields blank. 3. System gets Customer Details for display 4. The system displays a list of customers (customer id, last name, and first name) with a valid credit status. 5. The Sales Clerk selects a customer. 6. System gets Meeting Details for display 7. The system displays a list of meetings (meeting id and meeting name). 8. The Sales Clerk enters the booking’s details quantity (1 to 4 inclusive), and status (pending or paid)). 9. The Sales Clerk selects add the booking. 10. The system verify details 11. The system generates Booking ID 12. The system saves the booking’s details (booking id, customer id, meeting id, booking date: today’s date, quantity, and status). 13. The system displays the “Booking added successfully” message. 14. The system displays the “Add another booking?” prompt. 15. The Sales Clerk clicks on no Button. 16. Add Booking Form closes. | |
| **OF EVENTS:** |
|  |
|  |
|  |
|  |
| **ALTERNATE COURSES:** | 8a1. Sales Clerk selects Return  8a2. Form closes | |
| 11a1. Error message  11a2. Go back to step 2. | |
|  | 14a1. Sales Clerk selects “Add another booking”  14a2. Go to Step 2. | |
| **POST CONDITIONS:** |  | |
| **ASSUMPTIONS:** |  | |

|  |  |  |
| --- | --- | --- |
| **USE CASE NAME:** | Add Horse | **USE CASE TYPE** |
| **USE CASE ID:** | 5 | **Design Requirements: 🗹** |
| **PRIORITY:** | High |  |
| **PRIMARY BUSINESS ACTOR:** | Equine Administrator | |
| **OTHER PARTICIPATING ACTORS:** |  | |
| **DESCRIPTION:** | Enables the Equine Administrator to add a Horse and its details. | |
| **PRE-CONDITIONS:** | The Equine Administrator has logged onto the system | |
| **TYPICAL COURSE** | 1. The Equine Administrator selects the “Add Horse” function. 2. The system displays the “Add Horse” form with all fields blank. 3. The Equine Administrator enters the horse’s details (horse id (1 to 999999 inclusive), horse name, gender (male, female or gelding), and date of birth). 4. The system validates that the details are filled in correctly or identifies that the details are missing or incorrect and prompts for the completion of the details. 5. The system displays a list of owners (owner id, last name, and first name). 6. The Equine Administrator selects an owner. 7. The Equine Administrator selects Add Horse 8. The system Saves the horses details (horse id, horse name, gender, date of birth, and owner id). 9. The system displays the “Horse added successfully” message. 10. The system displays the “Add another horse?” prompt. 11. The Equine Administrator clicks on Return Button 12. Add Horse Form closes. | |
| **OF EVENTS:** |
|  |
|  |
|  |
|  |
| **ALTERNATE COURSES:** | 7a1. Equine Administrator select Return  7a2. Form closes | |
| 10a1. Equine Administrator selects Add another Horse  10a2. Goes back to step 2 | |
| **POST CONDITIONS:** |  | |
| **ASSUMPTIONS:** |  | |

|  |  |  |
| --- | --- | --- |
| **USE CASE NAME:** | Add Race Entry | **USE CASE TYPE** |
| **USE CASE ID:** | 26 | **Design Requirements: 🗹** |
| **PRIORITY:** | High |  |
| **PRIMARY BUSINESS ACTOR:** | Race Manager | |
| **OTHER PARTICIPATING ACTORS:** |  | |
| **DESCRIPTION:** | Enables Race Manager to add a race entry to a selected race | |
| **PRE-CONDITIONS:** | The Race Manager has logged onto the system | |
| **TYPICAL COURSE** | 1. The Race Manager selects the “Add Race Entry” function. 2. The system displays the “Add Race Entry” form. 3. System gets Scheduled race details for display 4. The system displays a list of all the scheduled races (race id and race name). 5. The Race Manager selects the race to which the entry will be added. 6. System gets race details for display 7. The system displays the race’s details (race id, race name, and meeting name). 8. System gets Race Entry details for display 9. The system displays the list of the race entries (horse name, jockey last name, and jockey first name) already added to the selected race. 10. System gets Horse details for display 11. The system displays a list of the horses (horse id and horse name). 12. The Race Manager selects a horse. 13. The system displays the error message “The horse has already entered this race” if the selected horse is already entered into the selected race and then displays a list of the horses (horse id and horse name) 14. System gets Jockey details for display 15. The system displays a list of the jockeys (jockey id, last name, and first name) 16. The Race Manager selects a jockey. 17. The system displays the error message “The jockey has already entered this race on another horse” if the selected jockey is already entered into the selected race and then displays a list of the jockeys (jockey id, last name, and first name) 18. The Race Manager enters the status (pending, disqualified, or confirmed) of the race entry for the race. 19. The Race Manager selects add the race entry to the race 20. The system generated a unique race entry id (1-9999999999) for the race entry. 21. The system saves the race entry details (race entry id, race id, horse id, jockey id, status, and race entry time (set to zero)). 22. The system displays the “Race entry added to race successfully” message. 23. The system displays the “Add another race entry?” prompt. 24. The Race Manager clicks on Return Button 25. Add Race Entry Form closes. | |
| **OF EVENTS:** |
|  |
|  |
|  |
|  |
| **ALTERNATE COURSES:** | 4a1. Race manager selects Return  4a2. Form closes | |
| 14a1. Race Manager selects Return  14a2. Form closes | |
|  | 18a1. Race Manager selects to add another race  18a2. System goes back to step 2 | |
| **POST CONDITIONS:** |  | |
| **ASSUMPTIONS:** |  | |

|  |  |  |
| --- | --- | --- |
| **USE CASE NAME:** | Delete Booking | **USE CASE TYPE** |
| **USE CASE ID:** | 19 | **Design Requirements: 🗹** |
| **PRIORITY:** | High |  |
| **PRIMARY BUSINESS ACTOR:** | Sales Clerk | |
| **OTHER PARTICIPATING ACTORS:** |  | |
| **DESCRIPTION:** | Enables Sales Clerk to delete a selected bookings details. | |
| **PRE-CONDITIONS:** | The Sales Clerk has logged onto the system | |
| **TYPICAL COURSE** | 1. The Sales Clerk selects the “Delete Booking” function. 2. The system displays the “Delete Booking” form. 3. System gets Booking details for display 4. The system displays a list of all the bookings (booking id and booking date). 5. Sales Clerk selects the booking that requires deleting. 6. System gets selected Booking details for display 7. The system displays the booking’s details (booking id, booking date, quantity, status, customer id, customer last & first name, and meeting name). 8. Sales Clerk selects Deleting the booking. 9. The system displays the “Are you sure you want to delete this booking?” prompt. 10. Sales Clerk confirms deletion 11. The system deletes the booking’s details. 12. The system displays the “Booking deleted successfully” message. 13. The system displays the “Delete another booking?” prompt. 14. The Sales Clerk clicks on Return Button 15. Delete Booking Form closes. | |
| **OF EVENTS:** |
|  |
|  |
|  |
|  |
| **ALTERNATE COURSES:** | 4a1. Sales Clerk select Return  4a2. Form closes. | |
| 6a1. Sales Clerk select Return  6a2. Form closes. | |
|  | 8a1. Sales Clerk select another booking  8a2. Goes back to step 3 | |
|  | 12a1. Sales Clerk select to delete another booking  12a2 Goes back to step 3 | |
| **POST CONDITIONS:** |  | |
| **ASSUMPTIONS:** |  | |

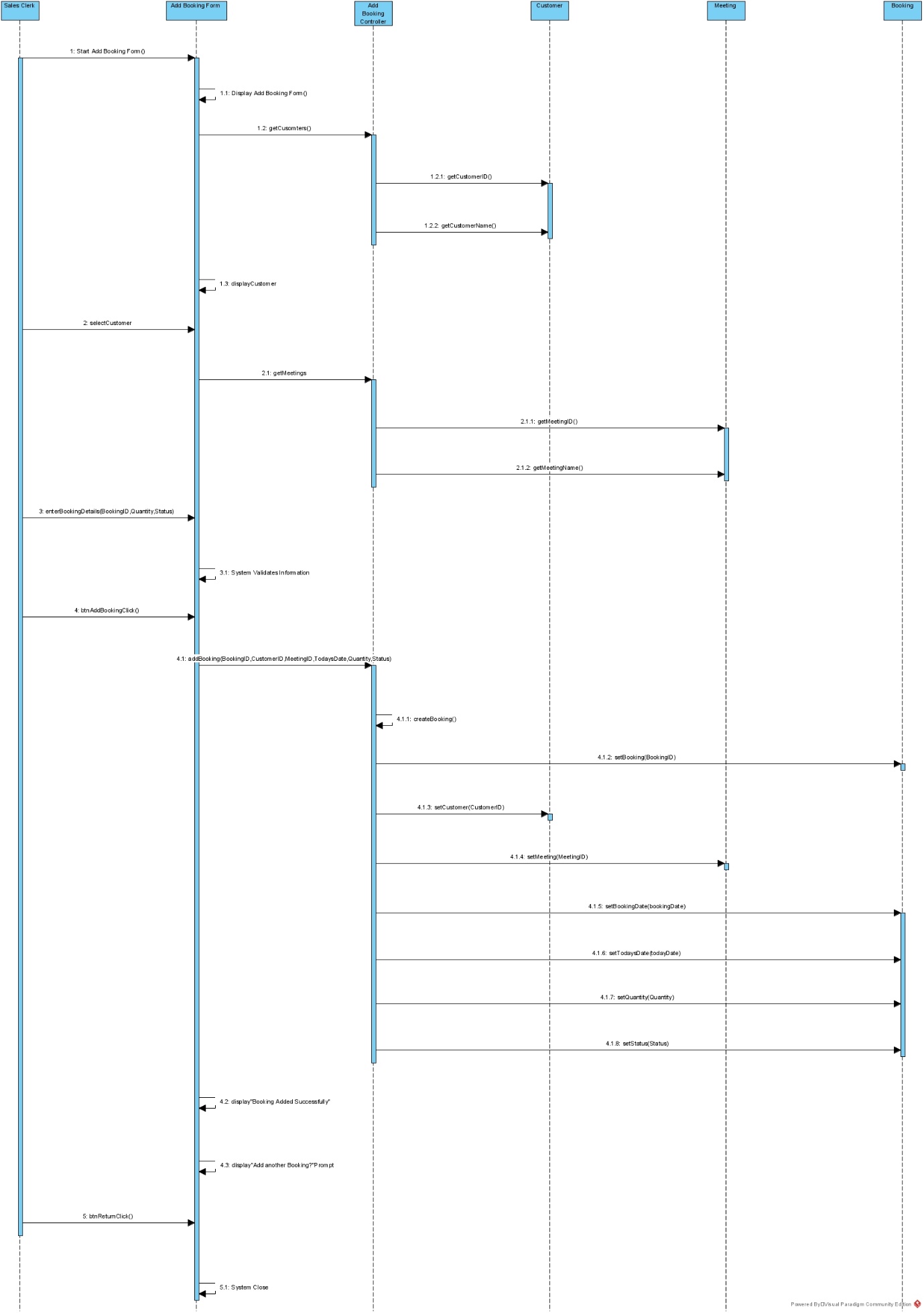
|  |  |  |
| --- | --- | --- |
| **USE CASE NAME:** | Delete Horse | **USE CASE TYPE** |
| **USE CASE ID:** | 22 | **Design Requirements: 🗹** |
| **PRIORITY:** | High |  |
| **PRIMARY BUSINESS ACTOR:** | Equine Administrator | |
| **OTHER PARTICIPATING ACTORS:** |  | |
| **DESCRIPTION:** | Enables Equine Administrator to delete a selected horse’s details | |
| **PRE-CONDITIONS:** | The Equine Administrator has logged onto the system | |
| **TYPICAL COURSE** | 1. The equine administrator selects the “Delete Horse” function. 2. The system displays the “Delete Horse” form. 3. System gets Horse details for display 4. The system displays a list of all the horses (horse id and horse name) that have no race entries. 5. Equine Administrator selects horse 6. System get selected Horse details for display 7. The system displays the horse’s details (horse id, horse name, date of birth, and owner last name and first name). 8. Equine Administrator selects delete Horse 9. The system displays the “Are you sure you want to delete this horse?” prompt. 10. Equine Administrator confirms deletion 11. The system deletes the horse’s details. 12. The system displays the “Horse deleted successfully” message. 13. The system displays the “Delete another horse?” prompt. 14. Equine Administrator clicks on Return Button 15. Delete Horse Form closes. | |
| **OF EVENTS:** |
|  |
|  |
|  |
|  |
| **ALTERNATE COURSES:** | 4a1. Equine Administrator selects Return  4a2. Form closes | |
| 6a1. Equine Administrator selects Return  6a2. Form closes | |
| 8a1. Equine Administrator can Select a Different Horse  8a2. Goes back to step 3 | |
| 12a1. Equine Administrator can Select another Horse to delete  12a2. Goes back to step 3 | |
| **POST CONDITIONS:** |  | |
| **ASSUMPTIONS:** |  | |

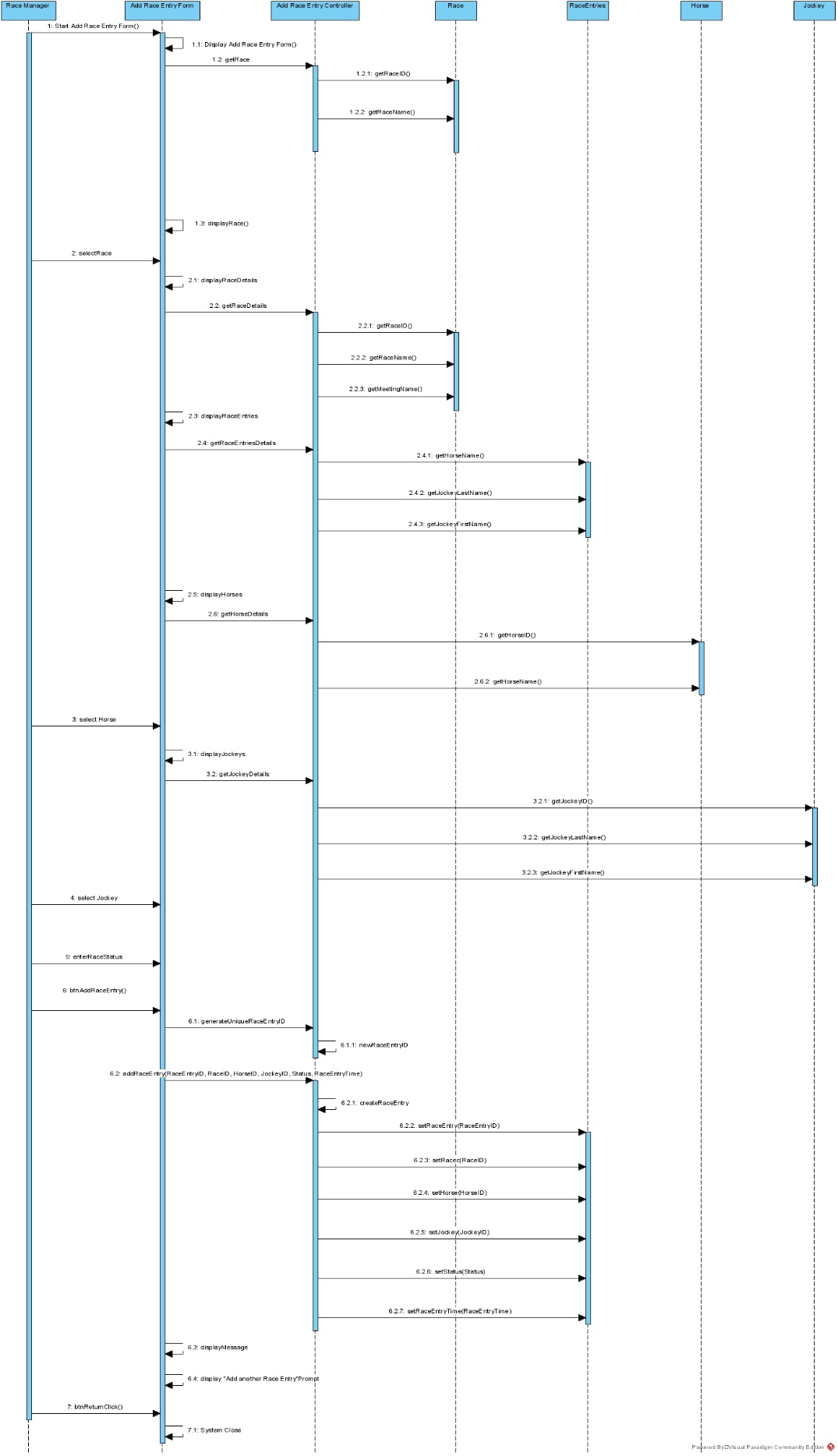
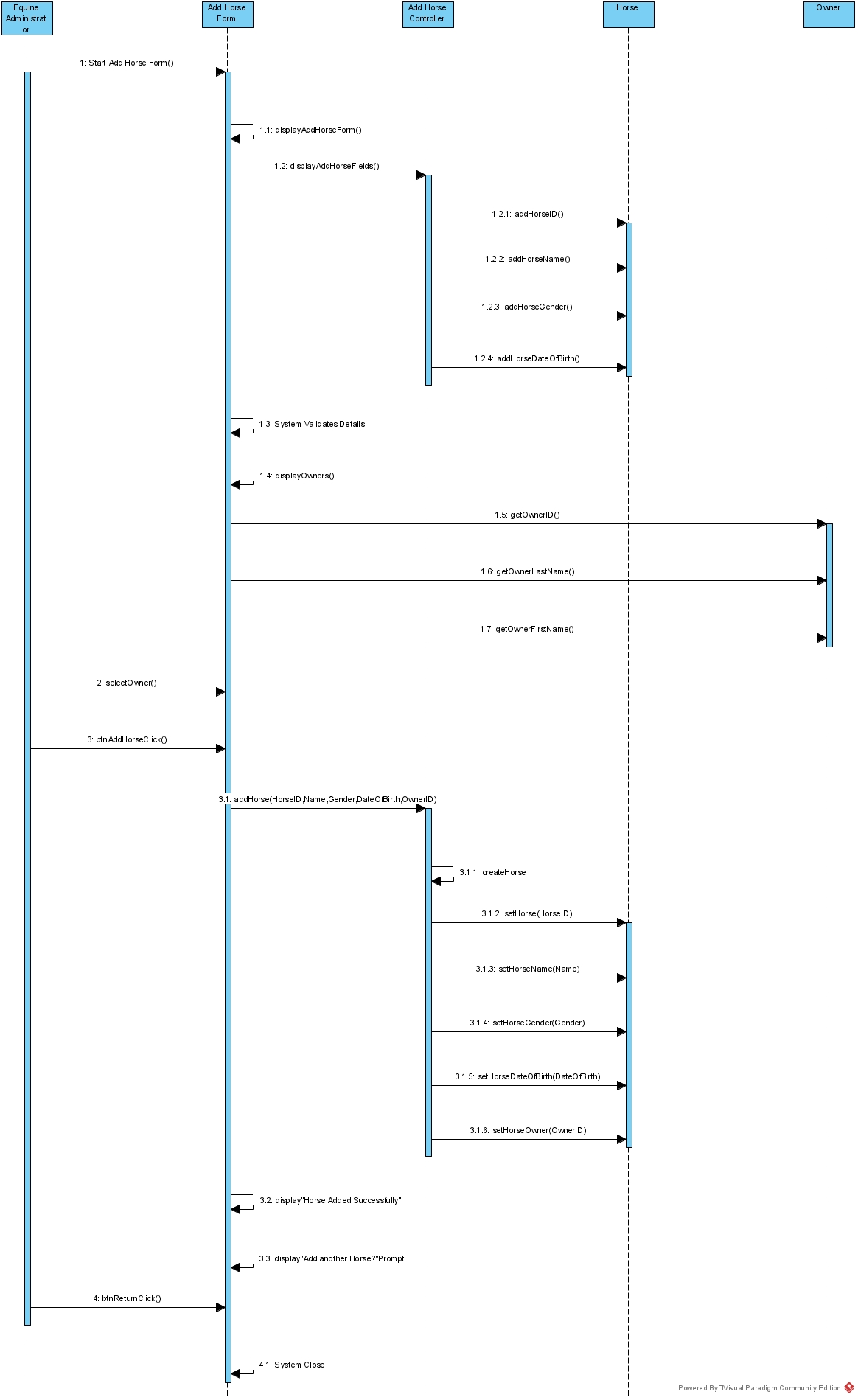
|  |  |  |
| --- | --- | --- |
| **USE CASE NAME:** | Delete Race Entry | **USE CASE TYPE** |
| **USE CASE ID:** | 28 | **Design Requirements: 🗹** |
| **PRIORITY:** | High |  |
| **PRIMARY BUSINESS ACTOR:** | Race Manager | |
| **OTHER PARTICIPATING ACTORS:** |  | |
| **DESCRIPTION:** | Enables Race Manager to delete a selected race | |
| **PRE-CONDITIONS:** | Race Manager has logged onto the system | |
| **TYPICAL COURSE** | 1. The Race Manager selects the “Delete Race Entry” function. 2. The system displays the “Delete Race Entry” form 3. System gets details for scheduled races list 4. The system displays a list of all the scheduled races (race id and race name) that have race entries. 5. The Race Manager selects the race with the race entry which is being deleted. 6. System gets race details for display 7. The system displays the race’s details (race id, race name, and status). 8. System gets Associated Race Entry details for selected Race 9. The system displays the list of the race entries (race entry id, horse name, jockey last name, and jockey first name) associated with the selected race. 10. The Race Manager selects a race entry. 11. System gets race entry details for display 12. The system displays the details of the race entry (race entry id, horse name, jockey last name, jockey first name, status, and race entry time). 13. The Race Manager selects to delete the race entry 14. The system deletes race entry’s details. 15. The system displays the “Race entry deleted successfully” message. 16. The system displays the “Delete another race entry?” prompt. 17. The Race Manager clicks on Return Button 18. Delete Race Entry Form closes. | |
| **OF EVENTS:** |
|  |
|  |
|  |
|  |
| **ALTERNATE COURSES:** | 4a1. Race Manager selects Return  4a2. Form closes | |
| 9a1. Race Manager selects Return  9a2. Form closes | |
|  | 12a1. Race Manager selects to delete another race entry  12a2. System goes to step 2. | |
| **POST CONDITIONS:** |  | |
| **ASSUMPTIONS:** |  | |

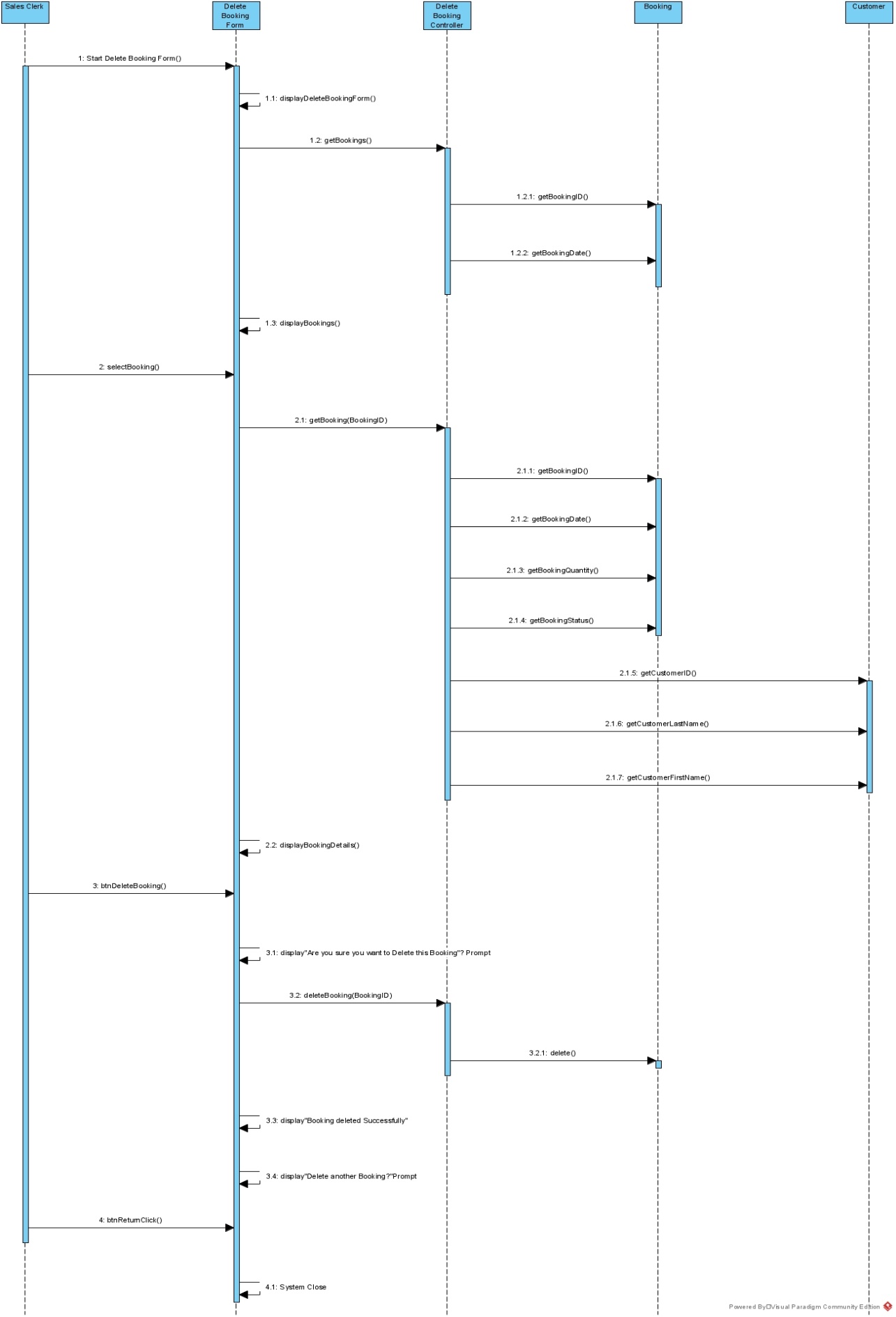
|  |  |  |
| --- | --- | --- |
| **USE CASE NAME:** | Update Booking | **USE CASE TYPE** |
| **USE CASE ID:** | 18 | **Design Requirements: 🗹** |
| **PRIORITY:** | High |  |
| **PRIMARY BUSINESS ACTOR:** | Sales Clerk | |
| **OTHER PARTICIPATING ACTORS:** |  | |
| **DESCRIPTION:** | Enables the Sales Clerk to update a selected bookings details. | |
| **PRE-CONDITIONS:** | The Sales Clerk has logged onto the system | |
| **TYPICAL COURSE** | 1. The Sales Clerk selects the “Update Booking” function. 2. The system displays the “Update Booking” form 3. System gets details for Booking list 4. The system displays a list of all the pending bookings (booking id and booking date). 5. The Sales Clerk selects a booking that has details that needs updating. 6. System gets details to display for Booking 7. The system displays the booking’s details (booking id, booking date, quantity, status, customer id, customer last & first name, and meeting name). 8. The Sales Clerk updates the relevant details (quantity and status only). 9. The system validates the entries in the fields and either prompts for confirmation to change the booking’s details or identifies missing or incorrect fields and prompts for completion of the entry. 10. The Sales Clerk confirms the change of details. 11. The system saves the booking’s details. 12. The system displays the “Booking updated successfully” message. 13. The system displays the “Update another booking?” prompt. 14. The Sales Clerk clicks on Return Button 15. Update Booking Form closes. | |
| **OF EVENTS:** |
|  |
|  |
|  |
|  |
| **ALTERNATE COURSES:** | 4a1. Sale Clerk selects Return  4a2. Form closes | |
| 8a1. Sale Clerk cancels the changes  8a2. Goes to step 12  8a3. Close Program | |
| 11a1. Sale Clerk selects to update another booking  11a2. Goes back to step 2. | |
| **POST CONDITIONS:** |  | |
| **ASSUMPTIONS:** |  | |

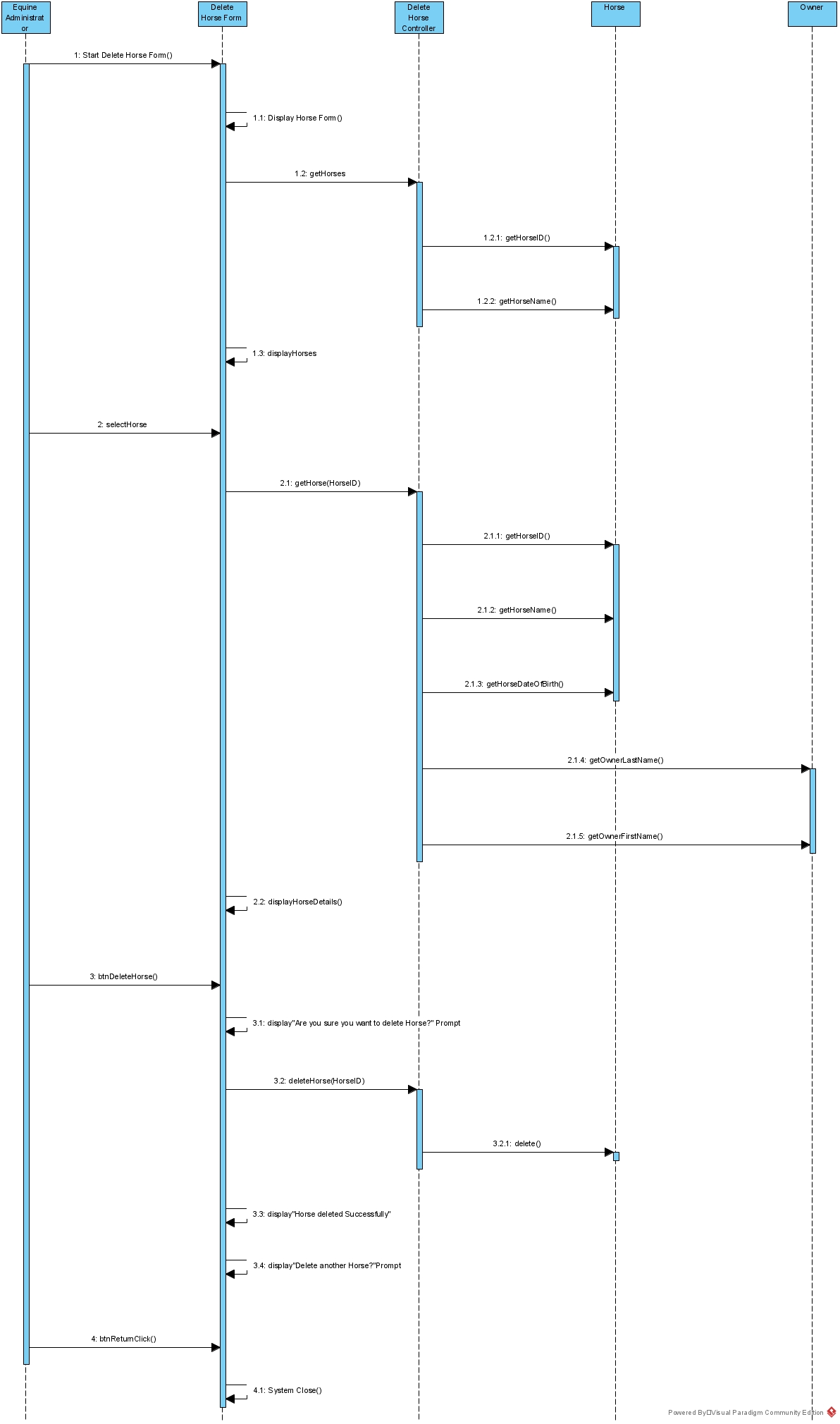
|  |  |  |
| --- | --- | --- |
| **USE CASE NAME:** | Update Horse | **USE CASE TYPE** |
| **USE CASE ID:** | 21 | **Design Requirements: 🗹** |
| **PRIORITY:** | High |  |
| **PRIMARY BUSINESS ACTOR:** | Equine Administrator | |
| **OTHER PARTICIPATING ACTORS:** |  | |
| **DESCRIPTION:** | Enables the Equine Administrator to update a selected horse’s details | |
| **PRE-CONDITIONS:** | The Equine Administrator has logged onto the system | |
| **TYPICAL COURSE** | 1. The Equine Administrator selects the “Update Horse” function. 2. The system displays the “Update Horse” form 3. System gets Horse details for Horse list 4. The system displays a list of all the horses (horse id and horse name). 5. Equine Administrator selects horse that needs it details updated. 6. System gets Horse details for selected Horse 7. The system displays the horse’s details (horse id, horse name, gender, date of birth, and owner last name and first name). 8. The Equine Administrator updates the relevant details (horse name, gender, and date of birth only). 9. The system checks if fields are filled in if not the system prompts for completion of the entry. 10. The Equine Administrator confirms change of details 11. The system saves the horse’s details. 12. The system displays the “Horse updated successfully” message. 13. Equine Administrator clicks on Return Button 14. Update Horse Form closes. | |
| **OF EVENTS:** |
|  |
|  |
|  |
|  |
| **ALTERNATE COURSES:** | 4a1. Equine Administrator selects Return  4a2. Form closes | |
| 8a1. Equine Administrator selects Return  8a2. Form closes | |
| 11a1. Equine Administrator selects update another horse.  11a2. Goes back to step 2 | |
| **POST CONDITIONS:** |  | |
| **ASSUMPTIONS:** |  | |

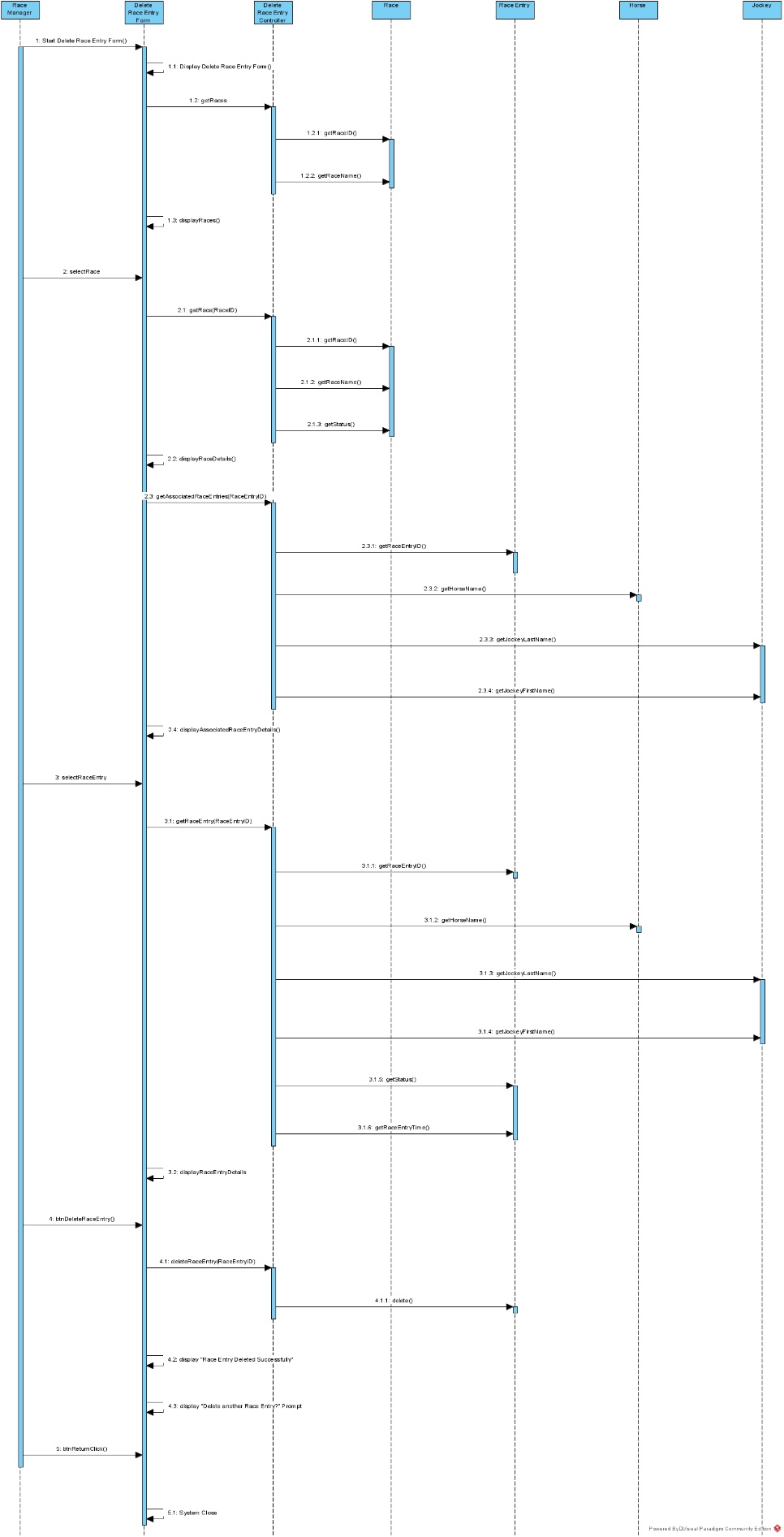
## 8.5 Sequence Diagrams

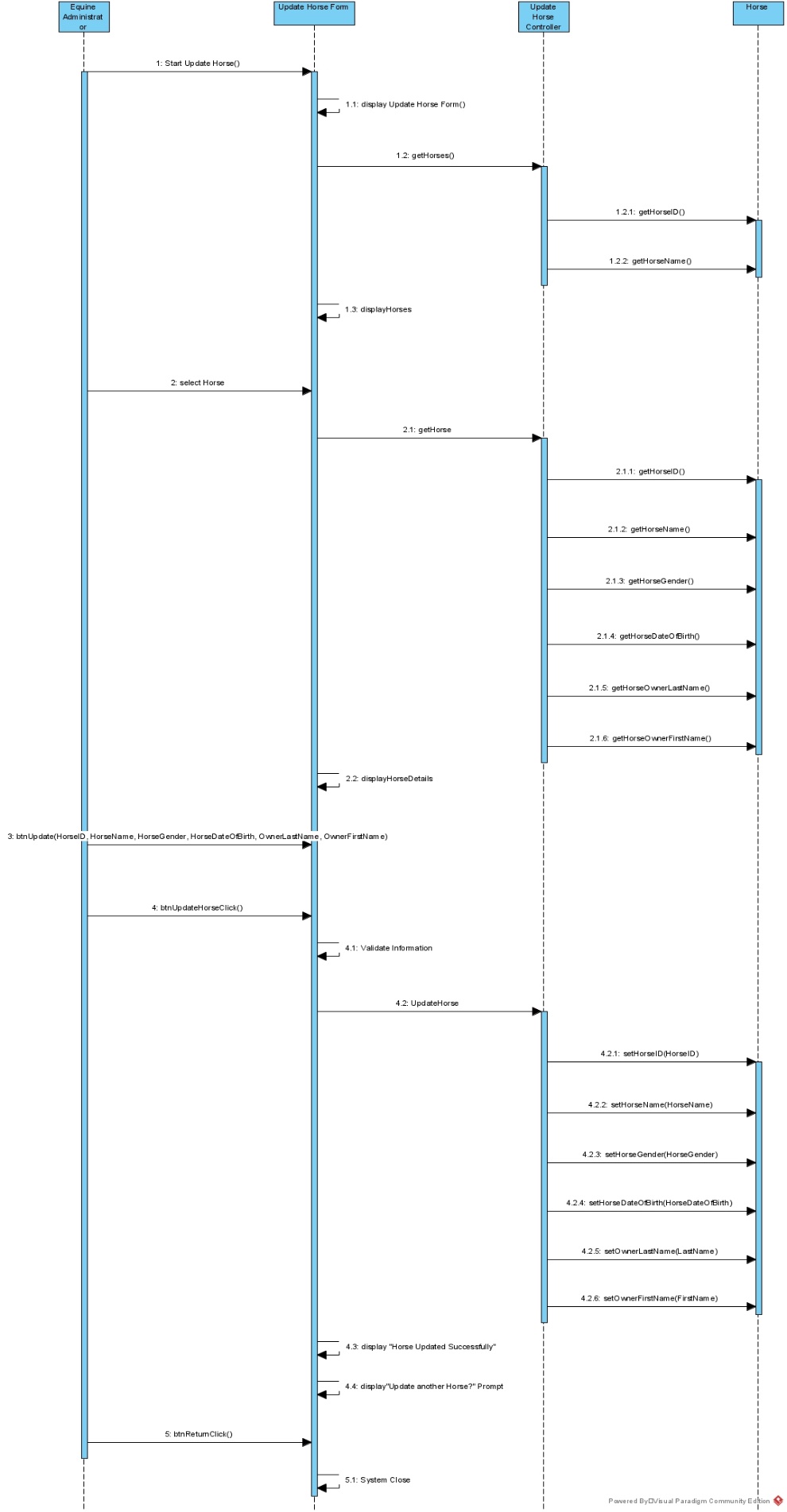
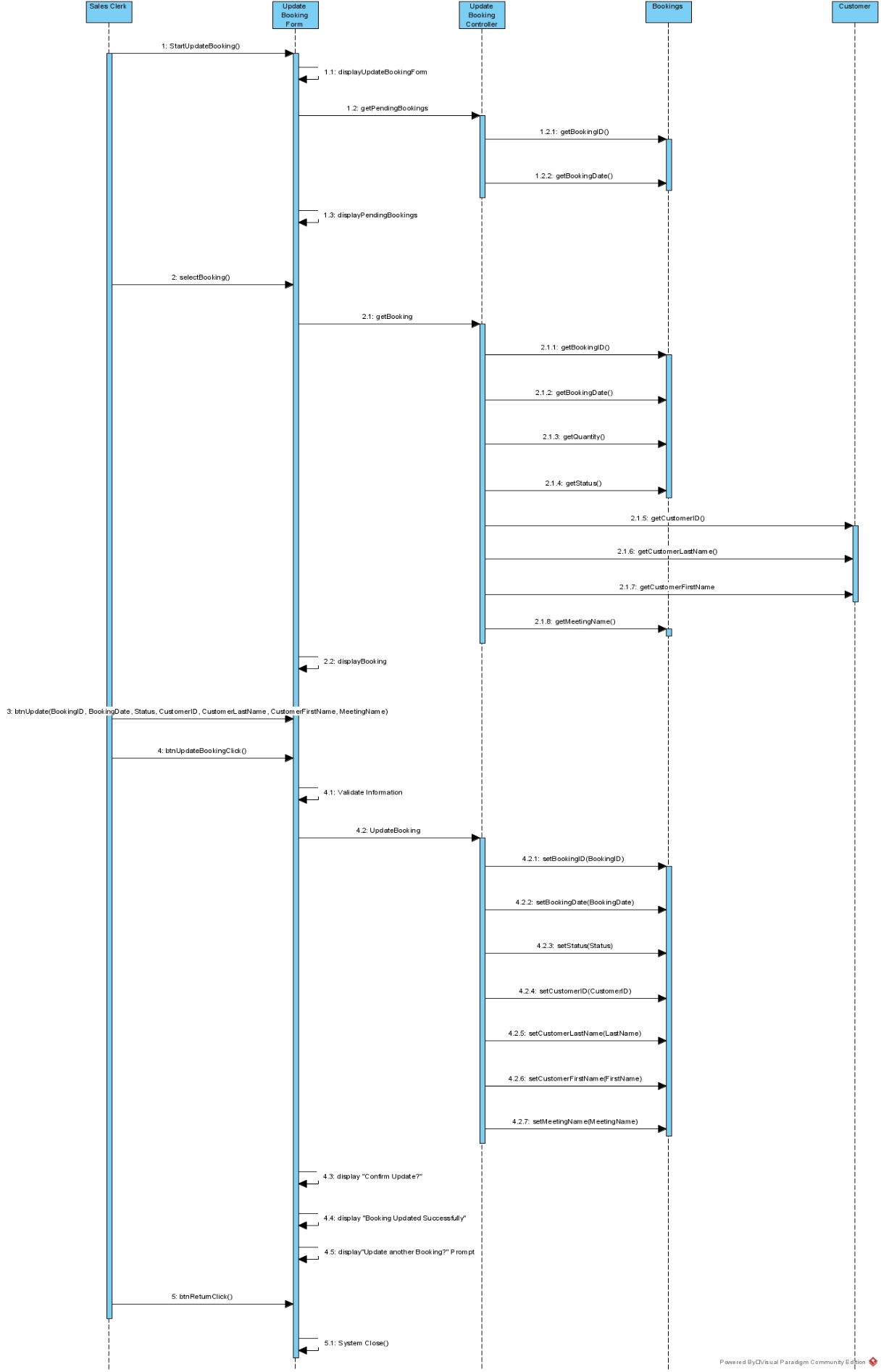
**

**

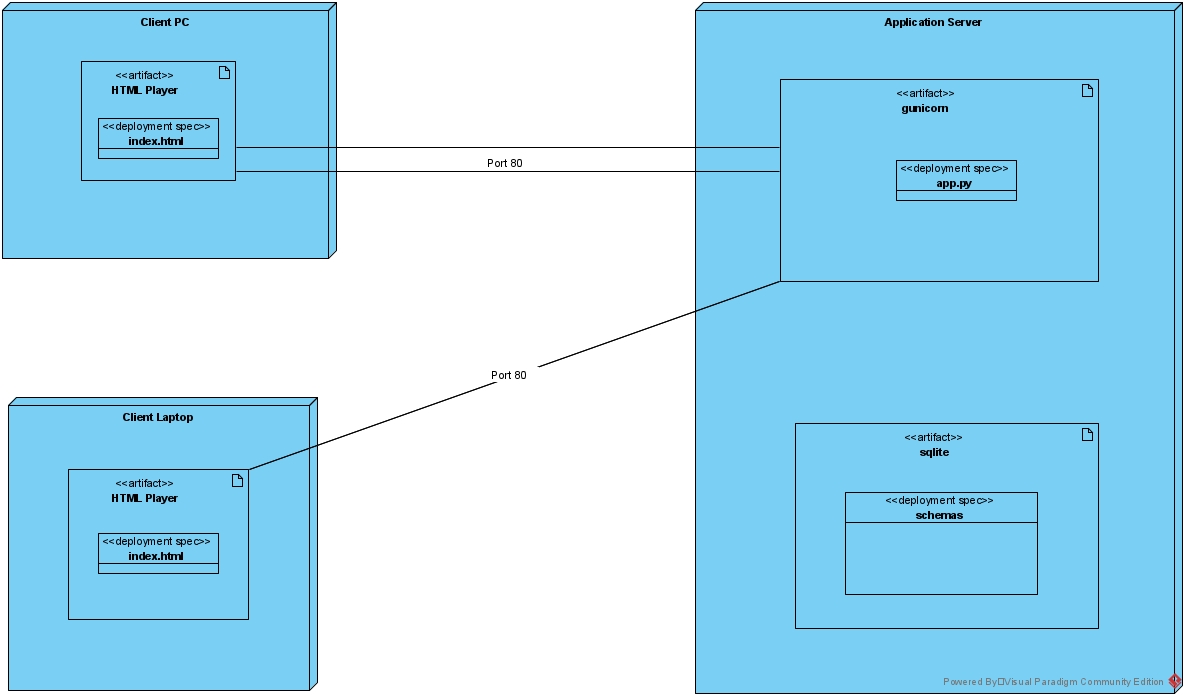
**

**

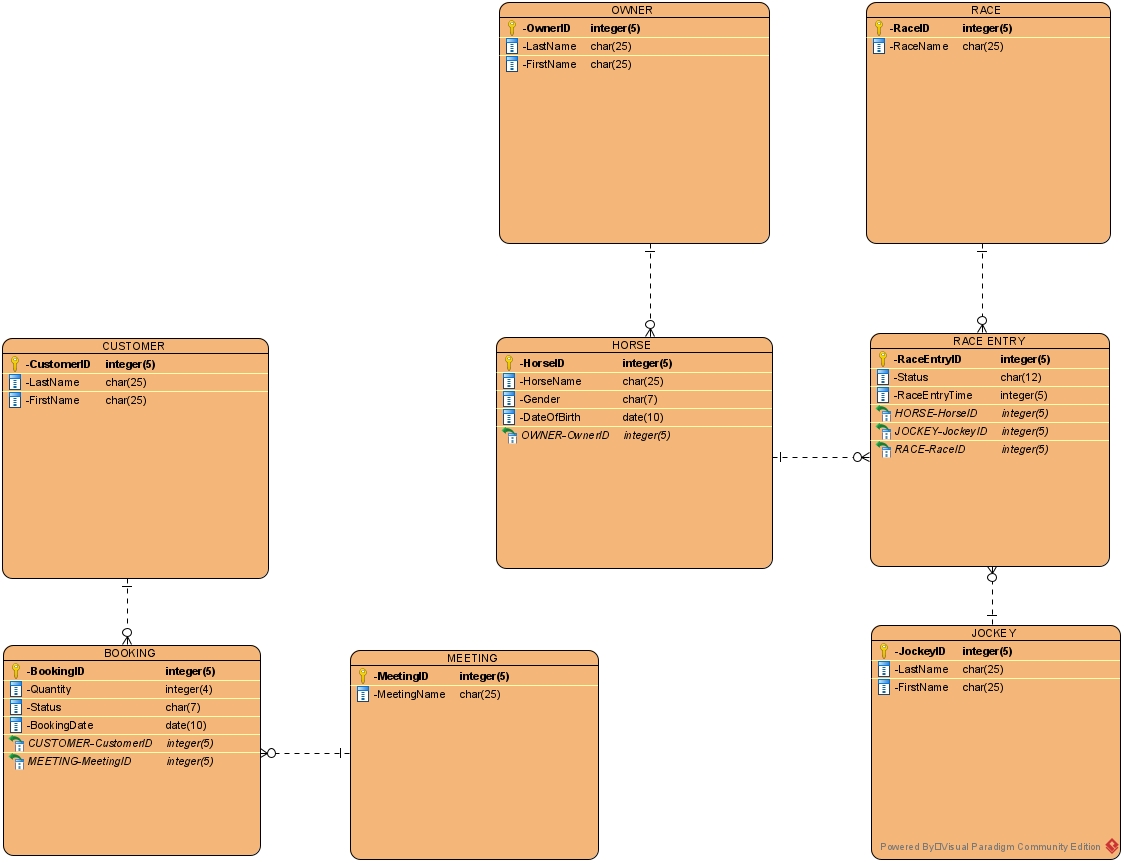
**

**

## 8.6 Deployment Diagram



## 8.7 Database Design



**BOOKING**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Required** | **Data Type** | **Maximum Length** | **Range/List** |
| **BookingID** | **Yes** | **Auto-number** | **5** | **Primary Key 1 - 99999** |
| **Quantity** | **Yes** | **Text** | **1** | **1 – 4 inclusive** |
| **Status** | **Yes** | **Text** | **7** | **Pending, Paid** |
| **BookingDate** | **Yes** | **Date** | **10** | **DD/MM/YYYY** |
| **CustomerID** | **Yes** | **Foreign Key(CUSTOMER)** | **5** | **-** |
| **MeetingID** | **Yes** | **Foreign Key(MEETING)** | **5** | **-** |

**HORSE**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Required** | **Data Type** | **Maximum Length** | **Range/List** |
| **HorseID** | **Yes** | **Auto-number** | **5** | **Primary Key 1- 99999** |
| **HorseName** | **Yes** | **Text** | **25** | **-** |
| **Gender** | **Yes** | **Text** | **7** | **Male, Female, Gelding** |
| **DateOfBirth** | **Yes** | **Date** | **10** | **DD/MM/YYYY** |
| **OwnerID** | **Yes** | **Foreign Key(OWNER)** | **5** | **-** |

**OWNER**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Required** | **Data Type** | **Maximum Length** | **Range/List** |
| **OwnerID** | **Yes** | **Auto-number** | **5** | **Primary Key** |
| **LastName** | **Yes** | **Text** | **25** | **-** |
| **FirstName** | **Yes** | **Text** | **25** | **-** |

**CUSTOMER**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Required** | **Data Type** | **Maximum Length** | **Range/List** |
| **CustomerID** | **Yes** | **Auto-number** | **5** | **Primary Key** |
| **LastName** | **Yes** | **Text** | **25** | **-** |
| **FirstName** | **Yes** | **Text** | **25** | **-** |
| **CreditStatus** | **Yes** | **Text** | **7** | **Valid, Invalid** |

**MEETING**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Required** | **Data Type** | **Maximum Length** | **Range/List** |
| **MeetingID** | **Yes** | **Auto-number** | **5** | **Primary Key** |
| **MeetingName** | **Yes** | **Text** | **25** | **-** |

**RACE ENTRY**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Required** | **Data Type** | **Maximum Length** | **Range/List** |
| **RaceEntryID** | **Yes** | **Auto-number** | **5** | **Primary Key, 1 - 99999** |
| **RaceID** | **Yes** | **Foreign Key(RACE)** | **5** | **-** |
| **HorseID** | **Yes** | **Foreign Key(HORSE)** | **5** | **-** |
| **JockeyID** | **Yes** | **Foreign Key(JOCKEY)** | **5** | **-** |
| **Status** | **Yes** | **Text** | **12** | **Pending, Disqualified, Confirmed** |
| **Race Entry Time** | **Yes** | **Text** | **5** | **Set to 0** |

**RACE**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Required** | **Data Type** | **Maximum Length** | **Range/List** |
| **RaceID** | **Yes** | **Auto-number** | **5** | **Primary Key** |
| **RaceName** | **Yes** | **Text** | **25** | **-** |
| **Status** | **Yes** | **Text** | **9** | **Scheduled, Finished** |

**JOCKEY**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Required** | **Data Type** | **Maximum Length** | **Range/List** |
| **JockeyID** | **Yes** | **Auto-number** | **5** | **Primary Key** |
| **LastName** | **Yes** | **Text** | **25** | **-** |
| **FirstName** | **Yes** | **Text** | **25** | **-** |

## 8.8 Annotated User Interface Designs

Graphical user interface

Description automatically generated

Button that saves the Booking and displays the message “Booking added Successfully”

Button that returns User to Main Menu

Booking Date, Calendar Box selection. Max 10 Characters.

Status Drop Down Box list. Max 7 Characters.

Quantity Text Box, Max of 1 Character.

Auto Number, Max 5 Characters. Read-Only Text Box

Meeting Name is a Text Box. Max 25 Characters.

Auto Number, Max 5 Characters, Read-Only text box

Auto Number, Max 5 Characters. Read-Only text box

Last Name and First Name are Text Box. Max 25 Characters.

Graphical user interface, application

Description automatically generated

Button that saves the Horse and displays a message “Horse Added Successfully”

Button that returns User to Main Menu

Last Name and First Name are Text Box. Max 25 Characters.

Auto Number, Max 5 Characters, Read-Only text box

Date Of Birth Calendar Box selection. Max 10 Characters.

Gender Drop Down Box list. Max 7 Characters.

Horse Name Text Box. Max 25 Characters.

Auto Number, Max 5 Characters. Read-Only text box



Button that saves the Race Entry and displays a message “Race Entry added to Race Successfully”.

Horse Drop Down Box list. Max 25 Characters

Jockey Drop Down Box list. Max 25 Characters.

Button returns user to Main Menu

Status Drop Down Box list. Max 12 Characters.

Race and Meeting Name Max 25 Characters. Text Box

Auto Number, 5 Max Characters. Read-Only text box

Race Entries List Box.

Race Drop Down Box list. Max 25 Characters.

Graphical user interface, application

Description automatically generated

Button that deletes Booking Record and displays a message “Booking Deleted Successfully”.

Button returns user to Main Menu

Customer Last Name, First Name and Meeting Name Max 25 Characters. Read-Only Text Box

Auto Number, 5 Max Characters. Read-Only text box

Status Box. Max 7 Characters. Read-Only text box

Quantity Box. Max 1 Character. Read-Only text box

Booking Date. Max 10 Characters. Read-Only text box

Auto Number, 5 Max Characters. Read-Only text box

Booking Drop Down Box list. Max 25 Characters.

Graphical user interface, application

Description automatically generated

Horse Name Max 25 Characters. Read-Only Text Box

Date of Birth Max 10 Characters. Read-Only Text Box

Owner Last Name and First Name Max 25 Characters. Read-Only Text Box

Auto Number, 5 Max Characters. Read-Only text box

Horse Drop Down Box list. Max 25 Characters.

Button that deletes Horse Record and associated race details and displays message “Horse Deleted Successfully”

Button that returns user to Main Menu

Graphical user interface, application

Description automatically generated

Race Entry Time box. Max 5 Characters. Read-Only Text Box

Button that deletes Race Entry record and displays message “Race entry Deleted Successfully”

Button that returns user to Main Menu

Status Box Max 12 Characters. Read-Only Text Box

Status Box Max 12 Characters. Read-Only Text Box

Horse Name, Jockey First and Last Name 25 Max Characters. Read-Only Text Box

Auto Number, 5 Max Characters. Read-Only Text Box

Race Entry Select list button type. Max 25 Characters.

Race Name Max 25 Characters. Read-Only Text Box

Auto Number, 5 Max Characters. Read-Only text box

Race Drop Down Box list. Max 25 Characters.

Graphical user interface, application

Description automatically generatedGraphical user interface, application

Description automatically generated

Button that updates Booking Record and displays message “Booking Updated Successfully”

Button returns user to Main Menu

Customer First Name, Last Name and Meeting Name Max 25 Characters.

Auto Number, Max 5 Characters. Read-Only Text Box

Status Box. Max 7 Characters. Drop down selection

Quantity Box. Max 1 Character

Booking Date, Calendar selection. Max 10 characters

Auto Number, Max 5 Characters long. Read-Only text box

Booking Drop Down Box list. Max 25 Characters.

Button that updates Horse Record and displays message “Horse Updated Successfully”

Button that returns user to Main Menu

Owner Last Name and First Name Max 25 Characters

Date Of Birth, Calendar selection. Max 10 Characters

Gender box. Max 7 Characters. Drop down selection

Horse Name Box. Max 25 Characters

Auto Number. Max 5 Characters long. Read-Only Text Box

Horse Drop Down Box list. Max 25 Characters.

## 8.9 Test Plan

Data Base Test Plans

Booking

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Table Test ID** | |  | **Table Test Description** | | Booking | | | | | |
| **Created By** | |  | **Reviewed By** | |  | | **Version** | | 1.0 | |
|  |  |  |  |  |  |  |  |  |  |  |
| **QA Tester’s Log** | |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
| **Tester's Name** | | Sascha | **Date Tested** | |  | | **Test Case (Pass/Fail/Not Executed)** | | Pass | |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
| **Field Name** | **Item** | | **Expected Results** | | **Actual Results** | | | **Pass / Fail / Not executed / Suspended** | | |
|  |
| BookingID | AutoNumber | | Unique primary key | | Generates new number | | | Pass | | |  |
| Quantity | Number | | Can enter number from 1 - 4 | | Entered number between 1 and 4 | | | Pass | | |  |
| Status | Look Up | | "Pending", "Paid" | | Selected "Pending" | | | Pass | | |  |
| Booking Date | Date / Time | | Selected Calendar date | | Date works | | | Pass | | |  |
| Customer ID | Number | | Customer ID from Customer Table | | Viewed CustomerID assigned to Booking | | | Pass | | |  |
| Meeting ID | Number | | Meeting ID from Meeting Table | | Viewed MeetingID assigned to Booking | | | Pass | | |  |

Horse

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Table Test ID** | |  | **Table Test Description** | | Horse | | | | | |
| **Created By** | |  | **Reviewed By** | |  | | **Version** | | 1.0 | |
|  |  |  |  |  |  |  |  |  |  |  |
| **QA Tester’s Log** | |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
| **Tester's Name** | | Sascha | **Date Tested** | |  | | **Test Case (Pass/Fail/Not Executed)** | | Pass | |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
| **Field Name** | **Item** | | **Expected Results** | | **Actual Results** | | | **Pass / Fail / Not executed / Suspended** | | |
|  |
| HorseID | AutoNumber | | Unique primary key | | Generates new number | | | Pass | | |  |
| HorseName | Short Text | | 25 Characters Only | | Did not pass 25 | | | Pass | | |  |
| Gender | Look Up | | "Male", "Female", "Gelding | | Selected "Female" | | | Pass | | |  |
| DateOfBirth | Date / Time | | Selected Calendar date | | Date works | | | Pass | | |  |
| OwnerID | Number | | Owner ID from Owner Table | | Viewed OwnerID assigned to Horse | | | Pass | | |  |

Race Entry

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Table Test ID** | |  | **Table Test Description** | | RaceEntry | | | | | |
| **Created By** | |  | **Reviewed By** | |  | | **Version** | | 1.0 | |
|  |  |  |  |  |  |  |  |  |  |  |
| **QA Tester’s Log** | |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
| **Tester's Name** | | Sascha | **Date Tested** | |  | | **Test Case (Pass/Fail/Not Executed)** | | Pass | |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
| **Field Name** | **Item** | | **Expected Results** | | **Actual Results** | | | **Pass / Fail / Not executed / Suspended** | | |
|  |
| RaceEntryID | AutoNumber | | Unique primary key | | Generates new number | | | Pass | | |  |
| RaceEntryTime | Number | | Enter Number as Time | | Entered Number | | | Pass | | |  |
| Status | Look Up | | "Pending", "Disqualified", "Confirmed" | | Selected "Pending" | | | Pass | | |  |
| RaceID | Number | | RaceID from Race Table | | Viewed RaceID assigned to RaceEntry | | | Pass | | |  |
| HorseID | Number | | HorseID from Horse Table | | Viewed HorseID assigned to RaceEntry | | | Pass | | |  |
| JockeyID | Number | | JockeyID from Jockey Table | | Viewed JockeyID assigned to RaceEntry | | | Pass | | |  |

Customer

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Table Test ID** | |  | **Table Test Description** | | Customer | | | | | |
| **Created By** | |  | **Reviewed By** | |  | | **Version** | | 1.0 | |
|  |  |  |  |  |  |  |  |  |  |  |
| **QA Tester’s Log** | |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
| **Tester's Name** | | Sascha | **Date Tested** | |  | | **Test Case (Pass/Fail/Not Executed)** | | Pass | |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
| **Field Name** | **Item** | | **Expected Results** | | **Actual Results** | | | **Pass / Fail / Not executed / Suspended** | | |
|  |
| CustomerID | AutoNumber | | Unique primary key | | Generates new number | | | Pass | | |  |
| LastName | Short Text | | 25 Characters Only | | Did not pass 25 | | | Pass | | |  |
| FirstName | Short Text | | 25 Characters Only | | Did not pass 25 | | | Pass | | |  |

Race

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Table Test ID** | |  | **Table Test Description** | | Race | | | | | |
| **Created By** | |  | **Reviewed By** | |  | | **Version** | | 1.0 | |
|  |  |  |  |  |  |  |  |  |  |  |
| **QA Tester’s Log** | |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
| **Tester's Name** | | Sascha | **Date Tested** | |  | | **Test Case (Pass/Fail/Not Executed)** | | Pass | |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
| **Field Name** | **Item** | | **Expected Results** | | **Actual Results** | | | **Pass / Fail / Not executed / Suspended** | | |
|  |
| RaceID | AutoNumber | | Unique primary key | | Generates new number | | | Pass | | |  |
| RaceName | Short Text | | 25 Characters Only | | Did not pass 25 | | | Pass | | |  |

Meeting

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Table Test ID** | |  | **Table Test Description** | | Meeting | | | | | |
| **Created By** | |  | **Reviewed By** | |  | | **Version** | | 1.0 | |
|  |  |  |  |  |  |  |  |  |  |  |
| **QA Tester’s Log** | |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
| **Tester's Name** | | Sascha | **Date Tested** | |  | | **Test Case (Pass/Fail/Not Executed)** | | Pass | |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
| **Field Name** | **Item** | | **Expected Results** | | **Actual Results** | | | **Pass / Fail / Not executed / Suspended** | | |
|  |
| MeetingID | AutoNumber | | Unique primary key | | Generates new number | | | Pass | | |  |
| MeetingName | Short Text | | 25 Characters Only | | Did not pass 25 | | | Pass | | |  |

Jockey

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Table Test ID** | |  | **Table Test Description** | | Jockey | | | | | |
| **Created By** | |  | **Reviewed By** | |  | | **Version** | | 1.0 | |
|  |  |  |  |  |  |  |  |  |  |  |
| **QA Tester’s Log** | |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
| **Tester's Name** | | Sascha | **Date Tested** | |  | | **Test Case (Pass/Fail/Not Executed)** | | Pass | |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
| **Field Name** | **Item** | | **Expected Results** | | **Actual Results** | | | **Pass / Fail / Not executed / Suspended** | | |
|  |
| JockeyID | AutoNumber | | Unique primary key | | Generates new number | | | Pass | | |  |
| LastName | Short Text | | 25 Characters Only | | Did not pass 25 | | | Pass | | |  |
| FirstName | Short Text | | 25 Characters Only | | Did not pass 25 | | | Pass | | |  |

Owner

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Table Test ID** | |  | **Table Test Description** | | Owner | | | | | |
| **Created By** | |  | **Reviewed By** | |  | | **Version** | | 1.0 | |
|  |  |  |  |  |  |  |  |  |  |  |
| **QA Tester’s Log** | |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
| **Tester's Name** | | Sascha | **Date Tested** | |  | | **Test Case (Pass/Fail/Not Executed)** | | Pass | |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
| **Field Name** | **Item** | | **Expected Results** | | **Actual Results** | | | **Pass / Fail / Not executed / Suspended** | | |
|  |
| OwnerID | AutoNumber | | Unique primary key | | Generates new number | | | Pass | | |  |
| LastName | Short Text | | 25 Characters Only | | Did not pass 25 | | | Pass | | |  |
| FirstName | Short Text | | 25 Characters Only | | Did not pass 25 | | | Pass | | |  |

Coding Test Plans

Add Booking

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | | 16 | **Test Case Description** | | Enables the Sales Clerk to add a booking for a customer for a selected meeting. | | | | | |
| **Created By** | | Sascha | **Reviewed By** | | Bill | | **Version** | | 2.1 | |
|  |  |  |  |  |  |  |  |  |  |  |
| **QA Tester’s Log** | | Review comments from Bill incorprate in version 2.1 | | | |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
| **Tester's Name** | | Sascha | **Date Tested** | | November 17, 2021 | | **Test Case (Pass/Fail/Not Executed)** | | Pass | |
|  |  |  |  |  |  |  |  |  |  |  |
| **S #** | **Prerequisites:** | | |  | **S #** | **Test Data** | | | | |
| 1 | The Sales Clerk has logged onto the system | | |  | 1 | Customer | | | | |
| 2 |  | | |  | 2 | Meeting | | | | |
| 3 |  | | |  | 3 | Quantity | | | | |
| 4 |  | | |  | 4 | Status | | | | |
|  |  |  |  |  |  |  |  |  |  |  |
| **Test Scenario** | Input valid data, system saves new booking. | | | |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
| **Step #** | **Step Details** | | **Expected Results** | | **Actual Results** | | | **Pass / Fail / Not executed / Suspended** | | |
|  |
| 1 | Click Add Booking Button | | Form should open. Customer Details should display | |  | | | Pass | | |  |
| 2 | Select Customer | | Meeting Details displayed | | As Expected | | | Pass | | |  |
| 3 | Enter Booking Details | | Allowed to enter | | As Expected | | | Pass | | |  |
| 4 | Click Add Booking | | Prompt message "Booking add successfully". Prompt message "Add another booking?" | |  | | |  | | |  |
| 5 | Click No | | Form Closed | |  | | |  | | |  |
| **S #** | **Prerequisites:** | | |  | **S #** | **Test Data** | | | | |  |
| 1 | The Sales Clerk has logged onto the system | | |  | 1 |  | | | | |  |
| 2 |  | | |  | 2 |  | | | | |  |
| 3 |  | | |  | 3 |  | | | | |  |
| 4 |  | | |  | 4 |  | | | | |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
| **Test Scenario** | Click Return button from close without save. | | | |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
| **Step #** | **Step Details** | | **Expected Results** | | **Actual Results** | | | **Pass / Fail / Not executed / Suspended** | | |  |
|  |
| 1 | Click Add Booking Button | | Form should open. Customer Details should display | |  | | | Pass | | |  |
| 2 | Select Customer | | Meeting Details displayed | | As Expected | | | Pass | | |  |
| 3 | Click Return | | Form Closed | | As Expected | | | Pass | | |  |

Add Horse

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | | 5 | **Test Case Description** | | Enables the Equine Administrator to add a Horse and its details. | | | | | |
| **Created By** | | Sascha | **Reviewed By** | | Bill | | **Version** | | 2.1 | |
|  |  |  |  |  |  |  |  |  |  |  |
| **QA Tester’s Log** | | Review comments from Bill incorprate in version 2.1 | | | |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
| **Tester's Name** | | Sascha | **Date Tested** | | November 17, 2021 | | **Test Case (Pass/Fail/Not Executed)** | | Pass | |
|  |  |  |  |  |  |  |  |  |  |  |
| **S #** | **Prerequisites:** | | |  | **S #** | **Test Data** | | | | |
| 1 | The Equine Administrator has logged onto the system | | |  | 1 | Horse | | | | |
| 2 |  | | |  | 2 | Owner | | | | |
| 3 |  | | |  | 3 |  | | | | |
| 4 |  | | |  | 4 |  | | | | |
|  |  |  |  |  |  |  |  |  |  |  |
| **Test Scenario** | Input valid data, system saves new horse. | | | |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
| **Step #** | **Step Details** | | **Expected Results** | | **Actual Results** | | | **Pass / Fail / Not executed / Suspended** | | |
|  |
| 1 | Click Add Horse Button | | Form should open. Horse Details should display | | As Expected | | | Pass | | |  |
| 2 | Enter Horse Details | | Horse Details displayed | | As Expected | | | Pass | | |  |
| 3 | System displays Owner | | Display owner Details | | As Expected | | | Pass | | |  |
| 4 | Select Owner | | Select Owner for Horse | | As Expected | | | Pass | | |  |
| 5 | Click Add Horse Button | | Prompt message "Horse add successfully". Prompt message "Add another Horse?" | | As Expected | | | Pass | | |  |
| 5 | Click Return | | Form Closes | | As Expected | | | Pass | | |  |
| **S #** | **Prerequisites:** | | |  | **S #** | **Test Data** | | | | |  |
| 1 | The Equine Administrator has logged onto the system | | |  | 1 |  | | | | |  |
| 2 |  | | |  | 2 |  | | | | |  |
| 3 |  | | |  | 3 |  | | | | |  |
| 4 |  | | |  | 4 |  | | | | |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
| **Test Scenario** | Click Return button, form closes | | |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
| **Step #** | **Step Details** | | **Expected Results** | | **Actual Results** | | | **Pass / Fail / Not executed / Suspended** | | |  |
|  |
| 1 | Click Add Horse Button | | Form should open. Horse Details should display | |  | | | Pass | | |  |
| 2 | Select Owner | | Add Horse | | As Expected | | | Pass | | |  |
| 3 | Click Return | | Form Closed | | As Expected | | | Pass | | |  |
| **S #** | **Prerequisites:** | | |  | **S #** | **Test Data** | | | | |  |
| 1 | The Equine Administrator has logged onto the system | | |  | 1 |  | | | | |  |
| 2 |  | | |  | 2 |  | | | | |  |
| 3 |  | | |  | 3 |  | | | | |  |
| 4 |  | | |  | 4 |  | | | | |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
| **Test Scenario** | Add another Horse. Return to step 2 | | |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
| **Step #** | **Step Details** | | **Expected Results** | | **Actual Results** | | | **Pass / Fail / Not executed / Suspended** | | |  |
|  |
| 1 | Click Add Horse Button | | Form should open. Horse Details should display | |  | | | Pass | | |  |
| 2 | Select Owner | | Add Horse | | As Expected | | | Pass | | |  |
| 3 | Add another horse | | Return to step 2 | | As Expected | | | Pass | | |  |

Add Race Entry

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | | 26 | **Test Case Description** | | Enables Race Manager to add a race entry to a selected race | | | | | |
| **Created By** | | Sascha | **Reviewed By** | | Bill | | **Version** | | 2.1 | |
|  |  |  |  |  |  |  |  |  |  |  |
| **QA Tester’s Log** | | Review comments from Bill incorprate in version 2.1 | | | |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
| **Tester's Name** | | Sascha | **Date Tested** | | November 17, 2021 | | **Test Case (Pass/Fail/Not Executed)** | | Pass | |
|  |  |  |  |  |  |  |  |  |  |  |
| **S #** | **Prerequisites:** | | |  | **S #** | **Test Data** | | | | |
| 1 | The Race Manager has logged onto the system | | |  | 1 | Race | | | | |
| 2 |  | | |  | 2 | Race Entry | | | | |
| 3 |  | | |  | 3 | Horse | | | | |
| 4 |  | | |  | 4 | Jockey | | | | |
|  |  |  |  |  |  |  |  |  |  |  |
| **Test Scenario** | Input valid data, system saves new Race Entry. | | | |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
| **Step #** | **Step Details** | | **Expected Results** | | **Actual Results** | | | **Pass / Fail / Not executed / Suspended** | | |
|  |
| 1 | Click Add Race Entry Button | | Form should open. Race Details should display | | As Expected | | | Pass | | |  |
| 2 | Select Race | | System gets race details | | As Expected | | | Pass | | |  |
| 3 | System displays Race Entry Details | | Shows details added to race | | As Expected | | | Pass | | |  |
| 4 | System displays Horse Details | | Select Horse for Entry | | As Expected | | | Pass | | |  |
| 5 | System displays Jockey | | Select Jockey for Entry | | As Expected | | | Pass | | |  |
| 6 | Enter Status for Race Entry | | Race Entry Details Saved | | As Expected | | | Pass | | |  |
| 7 | Click Add Race Entry Button | | Message displayed "Race Entry added to race Successfully", "Add another Race Entry?" | | As Expected | | | Pass | | |  |
| 8 | Click Return button | | Form Closes | | As Expected | | | Pass | | |  |
| **S #** | **Prerequisites:** | | |  | **S #** | **Test Data** | | | | |  |
| 1 | The Race Manager has logged onto the system | | |  | 1 |  | | | | |  |
| 2 |  | | |  | 2 |  | | | | |  |
| 3 |  | | |  | 3 |  | | | | |  |
| 4 |  | | |  | 4 |  | | | | |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
| **Test Scenario** | Click Return button, form closes | | |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
| **Step #** | **Step Details** | | **Expected Results** | | **Actual Results** | | | **Pass / Fail / Not executed / Suspended** | | |  |
|  |
| 1 | Click Add Race Entry Button | | Form should open. Race Details should display | |  | | | Pass | | |  |
| 2 | Select Race | | Add Race Entry Details | | As Expected | | | Pass | | |  |
| 3 | Click Return | | Form Closed | | As Expected | | | Pass | | |  |
| **S #** | **Prerequisites:** | | |  | **S #** | **Test Data** | | | | |  |
| 1 | The Race Manager has logged onto the system | | |  | 1 |  | | | | |  |
| 2 |  | | |  | 2 |  | | | | |  |
| 3 |  | | |  | 3 |  | | | | |  |
| 4 |  | | |  | 4 |  | | | | |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
| **Test Scenario** | Click Return button, form closes | | |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
| **Step #** | **Step Details** | | **Expected Results** | | **Actual Results** | | | **Pass / Fail / Not executed / Suspended** | | |  |
|  |
| 1 | Click Add Race Entry Button | | Form should open. Race Entry Details should display | |  | | | Pass | | |  |
| 2 | Select Race | | Add Race Entry Details | | As Expected | | | Pass | | |  |
| 3 | Select Horse | | Horse selected for Race Entry | | As Expected | | | Pass | | |  |
| 4 | Click Return | | Form Closed | | As Expected | | | Pass | | |  |
| **S #** | **Prerequisites:** | | |  | **S #** | **Test Data** | | | | |  |
| 1 | The Race Manager has logged onto the system | | |  | 1 |  | | | | |  |
| 2 |  | | |  | 2 |  | | | | |  |
| 3 |  | | |  | 3 |  | | | | |  |
| 4 |  | | |  | 4 |  | | | | |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
| **Test Scenario** | Add another Race. Return to step 2 | | |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
| **Step #** | **Step Details** | | **Expected Results** | | **Actual Results** | | | **Pass / Fail / Not executed / Suspended** | | |  |
|  |
| 1 | Click Add Race Entry Button | | Form should open. Race Entry Details should display | |  | | | Pass | | |  |
| 2 | Select Race | | Add Race Entry Details | | As Expected | | | Pass | | |  |
| 3 | Select Horse | | Horse selected for Race Entry | | As Expected | | | Pass | | |  |
| 4 | Select Jockey | | Jockey selected for Race Entry | | As Expected | | | Pass | | |  |
| 5 | Select another Race | | Go back to step 2 | | As Expected | | | Pass | | |  |

Delete Booking

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | | 19 | **Test Case Description** | | Enables Sales Clerk to delete a selected bookings details. | | | | | |
| **Created By** | | Sascha | **Reviewed By** | | Bill | | **Version** | | 2.1 | |
|  |  |  |  |  |  |  |  |  |  |  |
| **QA Tester’s Log** | | Review comments from Bill incorprate in version 2.1 | | | |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
| **Tester's Name** | | Sascha | **Date Tested** | | November 17, 2021 | | **Test Case (Pass/Fail/Not Executed)** | | Pass | |
|  |  |  |  |  |  |  |  |  |  |  |
| **S #** | **Prerequisites:** | | |  | **S #** | **Test Data** | | | | |
| 1 | The Sales Clerk has logged onto the system | | |  | 1 | Booking | | | | |
| 2 |  | | |  | 2 | Customer | | | | |
| 3 |  | | |  | 3 | Meeting | | | | |
| 4 |  | | |  | 4 |  | | | | |
|  |  |  |  |  |  |  |  |  |  |  |
| **Test Scenario** | Select Booking for Deleting | | |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
| **Step #** | **Step Details** | | **Expected Results** | | **Actual Results** | | | **Pass / Fail / Not executed / Suspended** | | |
|  |
| 1 | Click Delete Booking Button | | Form should open. Booking details ready for display | | As Expected | | | Pass | | |  |
| 2 | Select Booking | | Display Booking Details | | As Expected | | | Pass | | |  |
| 3 | Click Delete Booking Button | | Message displayed "Booking deleted Successfully", "Delete another Booking?" | | As Expected | | | Pass | | |  |
| 4 | Click Return button | | Form Closes | | As Expected | | | Pass | | |  |
| **S #** | **Prerequisites:** | | |  | **S #** | **Test Data** | | | | |  |
| 1 | The Sales Clerk has logged onto the system | | |  | 1 |  | | | | |  |
| 2 |  | | |  | 2 |  | | | | |  |
| 3 |  | | |  | 3 |  | | | | |  |
| 4 |  | | |  | 4 |  | | | | |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
| **Test Scenario** | Click Return button, form closes | | |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
| **Step #** | **Step Details** | | **Expected Results** | | **Actual Results** | | | **Pass / Fail / Not executed / Suspended** | | |  |
|  |
| 1 | Click Delete Booking Button | | Form should open. Booking details ready for display | | As Expected | | | Pass | | |  |
| 2 | Select Booking | | Display Booking Details | | As Expected | | | Pass | | |  |
| 3 | Click Delete Booking Button | | Message displayed "Booking deleted Successfully", "Delete another Booking?" | | As Expected | | | Pass | | |  |
| 4 | Click Return | | Form Closes | | As Expected | | | Pass | | |  |
| **S #** | **Prerequisites:** | | |  | **S #** | **Test Data** | | | | |  |
| 1 | The Sales Clerk has logged onto the system | | |  | 1 |  | | | | |  |
| 2 |  | | |  | 2 |  | | | | |  |
| 3 |  | | |  | 3 |  | | | | |  |
| 4 |  | | |  | 4 |  | | | | |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
| **Test Scenario** | Click Return button, form closes | | |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
| **Step #** | **Step Details** | | **Expected Results** | | **Actual Results** | | | **Pass / Fail / Not executed / Suspended** | | |  |
|  |
| 1 | Click Delete Booking Button | | Form should open. Booking details ready for display | | As Expected | | | Pass | | |  |
| 2 | Select Booking | | Display Booking Details | | As Expected | | | Pass | | |  |
| 3 | Click Delete Booking Button | | Message displayed "Booking deleted Successfully", "Delete another Booking?" | | As Expected | | | Pass | | |  |
| 4 | Click Return | | Form Closes | | As Expected | | | Pass | | |  |
| **S #** | **Prerequisites:** | | |  | **S #** | **Test Data** | | | | |  |
| 1 | The Sales Clerk has logged onto the system | | |  | 1 |  | | | | |  |
| 2 |  | | |  | 2 |  | | | | |  |
| 3 |  | | |  | 3 |  | | | | |  |
| 4 |  | | |  | 4 |  | | | | |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
| **Test Scenario** | Delete another booking | |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
| **Step #** | **Step Details** | | **Expected Results** | | **Actual Results** | | | **Pass / Fail / Not executed / Suspended** | | |  |
|  |
| 1 | Click Delete Booking Button | | Form should open. Booking details ready for display | | As Expected | | | Pass | | |  |
| 2 | Select Booking | | Display Booking Details | | As Expected | | | Pass | | |  |
| 3 | Click Delete Booking Button | | Message displayed "Booking deleted Successfully", "Delete another Booking?" | | As Expected | | | Pass | | |  |
| 4 | Select another Booking | | Display Booking Details | | As Expected | | | Pass | | |  |
| **S #** | **Prerequisites:** | | |  | **S #** | **Test Data** | | | | |  |
| 1 | The Sales Clerk has logged onto the system | | |  | 1 |  | | | | |  |
| 2 |  | | |  | 2 |  | | | | |  |
| 3 |  | | |  | 3 |  | | | | |  |
| 4 |  | | |  | 4 |  | | | | |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
| **Test Scenario** | Delete another booking | |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
| **Step #** | **Step Details** | | **Expected Results** | | **Actual Results** | | | **Pass / Fail / Not executed / Suspended** | | |  |
|  |
| 1 | Click Delete Booking Button | | Form should open. Booking details ready for display | | As Expected | | | Pass | | |  |
| 2 | Select Booking | | Display Booking Details | | As Expected | | | Pass | | |  |
| 3 | Click Delete Booking Button | | Message displayed "Booking deleted Successfully", "Delete another Booking?" | | As Expected | | | Pass | | |  |
| 4 | Select delete another Booking | | Display Booking Details | | As Expected | | | Pass | | |  |

Delete Horse

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | | 22 | **Test Case Description** | | Enables Equine Administrator to delete a selected horse’s details | | | | | |
| **Created By** | | Sascha | **Reviewed By** | | Bill | | **Version** | | 2.1 | |
|  |  |  |  |  |  |  |  |  |  |  |
| **QA Tester’s Log** | | Review comments from Bill incorprate in version 2.1 | | | |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
| **Tester's Name** | | Sascha | **Date Tested** | | November 17, 2021 | | **Test Case (Pass/Fail/Not Executed)** | | Pass | |
|  |  |  |  |  |  |  |  |  |  |  |
| **S #** | **Prerequisites:** | | |  | **S #** | **Test Data** | | | | |
| 1 | The Equine Administrator has logged onto the system | | |  | 1 | Horse | | | | |
| 2 |  | | |  | 2 | Owner | | | | |
| 3 |  | | |  | 3 |  | | | | |
| 4 |  | | |  | 4 |  | | | | |
|  |  |  |  |  |  |  |  |  |  |  |
| **Test Scenario** | Select Horse for deleting | |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
| **Step #** | **Step Details** | | **Expected Results** | | **Actual Results** | | | **Pass / Fail / Not executed / Suspended** | | |
|  |
| 1 | Click Delete Horse Button | | Form should open. Horse details ready for display | | As Expected | | | Pass | | |  |
| 2 | Select Horse | | Display Horse details | | As Expected | | | Pass | | |  |
| 3 | Click Delete Horse Button | | Message displayed "Horse deleted Successfully", "Delete another Horse?" | | As Expected | | | Pass | | |  |
| 4 | Click Return Button | | Form Closes | | As Expected | | | Pass | | |  |
| **S #** | **Prerequisites:** | | |  | **S #** | **Test Data** | | | | |  |
| 1 | The Equine Administrator has logged onto the system | | |  | 1 |  | | | | |  |
| 2 |  | | |  | 2 |  | | | | |  |
| 3 |  | | |  | 3 |  | | | | |  |
| 4 |  | | |  | 4 |  | | | | |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
| **Test Scenario** | Click Return button, form closes | | |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
| **Step #** | **Step Details** | | **Expected Results** | | **Actual Results** | | | **Pass / Fail / Not executed / Suspended** | | |  |
|  |
| 1 | Click Delete Horse Button | | Form should open. Horse details ready for display | | As Expected | | | Pass | | |  |
| 2 | Select Horse | | Display Horse Details | | As Expected | | | Pass | | |  |
| 3 | Click Return | | Form Closes | | As Expected | | | Pass | | |  |
| **S #** | **Prerequisites:** | | |  | **S #** | **Test Data** | | | | |  |
| 1 | The Equine Administrator has logged onto the system | | |  | 1 |  | | | | |  |
| 2 |  | | |  | 2 |  | | | | |  |
| 3 |  | | |  | 3 |  | | | | |  |
| 4 |  | | |  | 4 |  | | | | |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
| **Test Scenario** | Click Return button, form closes | | |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
| **Step #** | **Step Details** | | **Expected Results** | | **Actual Results** | | | **Pass / Fail / Not executed / Suspended** | | |  |
|  |
| 1 | Click Delete Horse Button | | Form should open. Horse details ready for display | | As Expected | | | Pass | | |  |
| 2 | Select Horse | | Display Horse Details | | As Expected | | | Pass | | |  |
| 3 | Click Return | | Form Closes | | As Expected | | | Pass | | |  |
| **S #** | **Prerequisites:** | | |  | **S #** | **Test Data** | | | | |  |
| 1 | The Equine Administrator has logged onto the system | | |  | 1 |  | | | | |  |
| 2 |  | | |  | 2 |  | | | | |  |
| 3 |  | | |  | 3 |  | | | | |  |
| 4 |  | | |  | 4 |  | | | | |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
| **Test Scenario** | Select Different Horse | |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
| **Step #** | **Step Details** | | **Expected Results** | | **Actual Results** | | | **Pass / Fail / Not executed / Suspended** | | |  |
|  |
| 1 | Click Delete Horse Button | | Form should open. Horse details ready for display | | As Expected | | | Pass | | |  |
| 2 | Select Horse | | Display Horse Details | | As Expected | | | Pass | | |  |
| 3 | Click Delete Horse Button | | Select No | | As Expected | | | Pass | | |  |
| 4 | Select another Horse | | Display Horse Details | | As Expected | | | Pass | | |  |
| **S #** | **Prerequisites:** | | |  | **S #** | **Test Data** | | | | |  |
| 1 | The Equine Administrator has logged onto the system | | |  | 1 |  | | | | |  |
| 2 |  | | |  | 2 |  | | | | |  |
| 3 |  | | |  | 3 |  | | | | |  |
| 4 |  | | |  | 4 |  | | | | |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
| **Test Scenario** | Select another Horse for deleting | | |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
| **Step #** | **Step Details** | | **Expected Results** | | **Actual Results** | | | **Pass / Fail / Not executed / Suspended** | | |  |
|  |
| 1 | Click Delete Horse Button | | Form should open. Horse details ready for display | | As Expected | | | Pass | | |  |
| 2 | Select Horse | | Display Horse Details | | As Expected | | | Pass | | |  |
| 3 | Click Delete Horse Button | | Message displayed "Horse deleted Successfully", "Delete another Horse?" | | As Expected | | | Pass | | |  |
| 4 | Select Horse | | Display Horse Details | | As Expected | | | Pass | | |  |

Delete Race Entry

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | | 28 | **Test Case Description** | | Enables Race Manager to delete a selected race | | | | | |
| **Created By** | | Sascha | **Reviewed By** | | Bill | | **Version** | | 2.1 | |
|  |  |  |  |  |  |  |  |  |  |  |
| **QA Tester’s Log** | | Review comments from Bill incorprate in version 2.1 | | | |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
| **Tester's Name** | | Sascha | **Date Tested** | | November 17, 2021 | | **Test Case (Pass/Fail/Not Executed)** | | Pass | |
|  |  |  |  |  |  |  |  |  |  |  |
| **S #** | **Prerequisites:** | | |  | **S #** | **Test Data** | | | | |
| 1 | Race Manager has logged onto the system | | |  | 1 | Race Entry | | | | |
| 2 |  | | |  | 2 | Race | | | | |
| 3 |  | | |  | 3 | Jockey | | | | |
| 4 |  | | |  | 4 |  | | | | |
|  |  |  |  |  |  |  |  |  |  |  |
| **Test Scenario** | Select Race Entry for Deleting | | |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
| **Step #** | **Step Details** | | **Expected Results** | | **Actual Results** | | | **Pass / Fail / Not executed / Suspended** | | |
|  |
| 1 | Click Delete Race Entry Button | | Form should open.Race details ready for display | | As Expected | | | Pass | | |  |
| 2 | Select Race | | Display Race with Race Entries | | As Expected | | | Pass | | |  |
| 3 | Select Race Entry | | Display Race Entry Details | | As Expected | | | Pass | | |  |
| 4 | Click Delete Race Entry Button | | Message displayed "Race Entry deleted Successfully", "Delete another Race Entry?" | | As Expected | | | Pass | | |  |
| 5 | Click Return | | Form Closes | | As Expected | | | Pass | | |  |
| **S #** | **Prerequisites:** | | |  | **S #** | **Test Data** | | | | |  |
| 1 | Race Manager has logged onto the system | | |  | 1 |  | | | | |  |
| 2 |  | | |  | 2 |  | | | | |  |
| 3 |  | | |  | 3 |  | | | | |  |
| 4 |  | | |  | 4 |  | | | | |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
| **Test Scenario** | Click Return button, form closes | | |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
| **Step #** | **Step Details** | | **Expected Results** | | **Actual Results** | | | **Pass / Fail / Not executed / Suspended** | | |  |
|  |
| 1 | Click Delete Race Entry Button | | Form should open.Race details ready for display | | As Expected | | | Pass | | |  |
| 2 | Select Race | | Display Race with Race Entries | | As Expected | | | Pass | | |  |
| 3 | Click Return | | Form Closes | | As Expected | | | Pass | | |  |
| **S #** | **Prerequisites:** | | |  | **S #** | **Test Data** | | | | |  |
| 1 | Race Manager has logged onto the system | | |  | 1 |  | | | | |  |
| 2 |  | | |  | 2 |  | | | | |  |
| 3 |  | | |  | 3 |  | | | | |  |
| 4 |  | | |  | 4 |  | | | | |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
| **Test Scenario** | Click Return button, form closes | | |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
| **Step #** | **Step Details** | | **Expected Results** | | **Actual Results** | | | **Pass / Fail / Not executed / Suspended** | | |  |
|  |
| 1 | Click Delete Race Entry Button | | Form should open.Race details ready for display | | As Expected | | | Pass | | |  |
| 2 | Select Race | | Display Race with Race Entries | | As Expected | | | Pass | | |  |
| 3 | Click Return | | Form Closes | | As Expected | | | Pass | | |  |
| **S #** | **Prerequisites:** | | |  | **S #** | **Test Data** | | | | |  |
| 1 | Race Manager has logged onto the system | | |  | 1 |  | | | | |  |
| 2 |  | | |  | 2 |  | | | | |  |
| 3 |  | | |  | 3 |  | | | | |  |
| 4 |  | | |  | 4 |  | | | | |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
| **Test Scenario** | Delete another Race Entry | |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
| **Step #** | **Step Details** | | **Expected Results** | | **Actual Results** | | | **Pass / Fail / Not executed / Suspended** | | |  |
|  |
| 1 | Click Delete Race Entry Button | | Form should open.Race details ready for display | | As Expected | | | Pass | | |  |
| 2 | Select Race | | Display Race with Race Entries | | As Expected | | | Pass | | |  |
| 3 | Select Race Entry | | Display Race Entry Details | | As Expected | | | Pass | | |  |
| 4 | Click Delete Race Entry Button | | Message displayed "Race Entry deleted Successfully", "Delete another Race Entry?" | | As Expected | | | Pass | | |  |
| 5 | Select another Race | | Display Race with Race Entries | | As Expected | | | Pass | | |  |

Update Booking

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | | 18 | **Test Case Description** | | Enables the Sales Clerk to update a selected bookings details. | | | | | |
| **Created By** | | Sascha | **Reviewed By** | | Bill | | **Version** | | 2.1 | |
|  |  |  |  |  |  |  |  |  |  |  |
| **QA Tester’s Log** | | Review comments from Bill incorprate in version 2.1 | | | |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
| **Tester's Name** | | Sascha | **Date Tested** | | 1-Jan-2017 | | **Test Case (Pass/Fail/Not Executed)** | | Pass | |
|  |  |  |  |  |  |  |  |  |  |  |
| **S #** | **Prerequisites:** | | |  | **S #** | **Test Data** | | | | |
| 1 | The Sales Clerk has logged onto the system | | |  | 1 | Booking | | | | |
| 2 |  | | |  | 2 | Status | | | | |
| 3 |  | | |  | 3 | Quantity | | | | |
| 4 |  | | |  | 4 |  | | | | |
|  |  |  |  |  |  |  |  |  |  |  |
| **Test Scenario** | Update Booking | |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
| **Step #** | **Step Details** | | **Expected Results** | | **Actual Results** | | | **Pass / Fail / Not executed / Suspended** | | |
|  |
| 1 | Click Update Booking Button | | Form should open. Booking Details should display | | As Expected | | | Pass | | |  |
| 2 | Select Booking | | Booking Details displayed | | As Expected | | | Pass | | |  |
| 3 | Enter Relevant Details | | Enter Quantity and Status Details | | As Expected | | | Pass | | |  |
| 4 | Click Update Booking | | Prompt message "Booking Updated successfully". Prompt message "Update another booking?" | | As Expected | | | Pass | | |  |
| 5 | Click Return button | | Form Closes | | As Expected | | | Pass | | |  |
| **S #** | **Prerequisites:** | | |  | **S #** | **Test Data** | | | | |  |
| 1 | The Sales Clerk has logged onto the system | | |  | 1 |  | | | | |  |
| 2 |  | | |  | 2 |  | | | | |  |
| 3 |  | | |  | 3 |  | | | | |  |
| 4 |  | | |  | 4 |  | | | | |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
| **Test Scenario** | Click Return button, form closes | | |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
| **Step #** | **Step Details** | | **Expected Results** | | **Actual Results** | | | **Pass / Fail / Not executed / Suspended** | | |  |
|  |
| 1 | Click Update Booking Button | | Form should open. Booking Details should display | | As Expected | | | Pass | | |  |
| 2 | Select Booking | | Booking Details displayed | | As Expected | | | Pass | | |  |
| 3 | Click Return | | Form Closed | | As Expected | | | Pass | | |  |
| **S #** | **Prerequisites:** | | |  | **S #** | **Test Data** | | | | |  |
| 1 | The Sales Clerk has logged onto the system | | |  | 1 |  | | | | |  |
| 2 |  | | |  | 2 |  | | | | |  |
| 3 |  | | |  | 3 |  | | | | |  |
| 4 |  | | |  | 4 |  | | | | |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
| **Test Scenario** | Cancel Changes | |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
| **Step #** | **Step Details** | | **Expected Results** | | **Actual Results** | | | **Pass / Fail / Not executed / Suspended** | | |  |
|  |
| 1 | Click Update Booking Button | | Form should open. Booking Details should display | | As Expected | | | Pass | | |  |
| 2 | Select Booking | | Booking Details displayed | | As Expected | | | Pass | | |  |
| 3 | Enter Relevant Details | | Enter Quantity and Status | | As Expected | | | Pass | | |  |
| 4 | Cancel Changes | | Cancels data input | | As Expected | | | Pass | | |  |
| 5 | Click Return | | Form closes. | | As Expected | | | Pass | | |  |
| **S #** | **Prerequisites:** | | |  | **S #** | **Test Data** | | | | |  |
| 1 | The Sales Clerk has logged onto the system | | |  | 1 |  | | | | |  |
| 2 |  | | |  | 2 |  | | | | |  |
| 3 |  | | |  | 3 |  | | | | |  |
| 4 |  | | |  | 4 |  | | | | |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
| **Test Scenario** | Update another Booking | |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
| **Step #** | **Step Details** | | **Expected Results** | | **Actual Results** | | | **Pass / Fail / Not executed / Suspended** | | |  |
|  |
| 1 | Click Update Booking Button | | Form should open. Booking Details should display | | As Expected | | | Pass | | |  |
| 2 | Select Booking | | Booking Details displayed | | As Expected | | | Pass | | |  |
| 3 | Enter Relevant Details | | Enter Quantity and Status Details | | As Expected | | | Pass | | |  |
| 4 | Click Update Booking | | Prompt message "Booking Updated successfully". Prompt message "Update another booking?" | | As Expected | | | Pass | | |  |
| 5 | Update another booking | | Returns to Select another booking | | As Expected | | | Pass | | |  |

Update Horse

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | | 21 | **Test Case Description** | | Enables the Equine Administrator to update a selected horse’s details | | | | | |
| **Created By** | | Sascha | **Reviewed By** | | Bill | | **Version** | | 2.1 | |
|  |  |  |  |  |  |  |  |  |  |  |
| **QA Tester’s Log** | | Review comments from Bill incorprate in version 2.1 | | | |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
| **Tester's Name** | | Sascha | **Date Tested** | | November 17, 2021 | | **Test Case (Pass/Fail/Not Executed)** | | Pass | |
|  |  |  |  |  |  |  |  |  |  |  |
| **S #** | **Prerequisites:** | | |  | **S #** | **Test Data** | | | | |
| 1 | The Equine Administrator has logged onto the system | | |  | 1 | Horse | | | | |
| 2 |  | | |  | 2 | Gender | | | | |
| 3 |  | | |  | 3 | Date Of Birth | | | | |
| 4 |  | | |  | 4 |  | | | | |
|  |  |  |  |  |  |  |  |  |  |  |
| **Test Scenario** | Select Horse for deleting | |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
| **Step #** | **Step Details** | | **Expected Results** | | **Actual Results** | | | **Pass / Fail / Not executed / Suspended** | | |
|  |
| 1 | Click Update Horse Button | | Form should open. Horse details ready for display | | As Expected | | | Pass | | |  |
| 2 | Select Horse | | Display Horse details | | As Expected | | | Pass | | |  |
| 3 | User updates Relevant Details | | Update Horse Name, Gender, Date Of Birth | | As Expected | | | Pass | | |  |
| 4 | Click Update Horse Button | | Message displayed "Horse updated Successfully". | | As Expected | | | Pass | | |  |
| 5 | Click Return button | | Form Closes | | As Expected | | | Pass | | |  |
| **S #** | **Prerequisites:** | | |  | **S #** | **Test Data** | | | | |  |
| 1 | The Equine Administrator has logged onto the system | | |  | 1 |  | | | | |  |
| 2 |  | | |  | 2 |  | | | | |  |
| 3 |  | | |  | 3 |  | | | | |  |
| 4 |  | | |  | 4 |  | | | | |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
| **Test Scenario** | Click Return button, form closes | | |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
| **Step #** | **Step Details** | | **Expected Results** | | **Actual Results** | | | **Pass / Fail / Not executed / Suspended** | | |  |
|  |
| 1 | Click Update Horse Button | | Form should open. Horse details ready for display | | As Expected | | | Pass | | |  |
| 2 | Select Horse | | Display Horse Details | | As Expected | | | Pass | | |  |
| 3 | Click Return | | Form Closes | | As Expected | | | Pass | | |  |
| **S #** | **Prerequisites:** | | |  | **S #** | **Test Data** | | | | |  |
| 1 | The Equine Administrator has logged onto the system | | |  | 1 |  | | | | |  |
| 2 |  | | |  | 2 |  | | | | |  |
| 3 |  | | |  | 3 |  | | | | |  |
| 4 |  | | |  | 4 |  | | | | |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
| **Test Scenario** | Click Return button, form closes | | |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
| **Step #** | **Step Details** | | **Expected Results** | | **Actual Results** | | | **Pass / Fail / Not executed / Suspended** | | |  |
|  |
| 1 | Click Update Horse Button | | Form should open. Horse details ready for display | | As Expected | | | Pass | | |  |
| 2 | Select Horse | | Display Horse Details | | As Expected | | | Pass | | |  |
| 3 | Click Return | | Form Closes | | As Expected | | | Pass | | |  |
| **S #** | **Prerequisites:** | | |  | **S #** | **Test Data** | | | | |  |
| 1 | The Equine Administrator has logged onto the system | | |  | 1 |  | | | | |  |
| 2 |  | | |  | 2 |  | | | | |  |
| 3 |  | | |  | 3 |  | | | | |  |
| 4 |  | | |  | 4 |  | | | | |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
| **Test Scenario** | Update Another Horse | |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
| **Step #** | **Step Details** | | **Expected Results** | | **Actual Results** | | | **Pass / Fail / Not executed / Suspended** | | |  |
|  |
| 1 | Click Update Horse Button | | Form should open. Horse details ready for display | | As Expected | | | Pass | | |  |
| 2 | Select Horse | | Display Horse details | | As Expected | | | Pass | | |  |
| 3 | User updates Relevant Details | | Update Horse Name, Gender, Date Of Birth | | As Expected | | | Pass | | |  |
| 4 | Click Update Horse Button | | Message displayed "Horse updated Successfully". | | As Expected | | | Pass | | |  |
| 4 | Select another Horse to Update | | Display Horse details | | As Expected | | | Pass | | |  |

# 9. Project Training

## 9.1 End User Background and Training Objectives

*This program should be easy to use even with people who find using a computer hard.*

*- With the training video, it would teach you how to use this program efficiently. The video is easy to understand and the steps that are given within the video are also easy to understand.*

*The training video is to teach new users at all skill levels how to use the program.*

## 9.2 Training Materials

## 9.3 Training Deliverables

The team have made some training videos for new users. These videos should help with being able to use the program and eventually build confidence in using the program.

## 9.4 Evaluation

*We have put together some good training material that will help new users and existing users on how to confidently use the program. The program is fairly easy to use and the training material is easy to understand. We have put together a good program and our hours shall reflect off the program we have developed.*

# 10. Conclusion & Lessons Learned

This program and development of this program have been a big highlight within the course. Adding everything together that you’ve learnt throughout the year, you can see what you can do when you add everything into one big project. I’ve learnt that with the waterfall methodology, it doesn’t have much flexibility because you must keep moving. I had trouble with my data dictionary where I kept going back and forth adding new data from other tables I didn’t see till I asked one of the tutors. “Do it once, do it right” is probably the motto for waterfall methodology. I’ve still got a long way to learn coding in computing but I’m happy that I’ve learnt these skills and added it to my skill book. With the covid, it made it hard for me to learn things because I like to talk with the tutor person to person on how to do things and get hands on experience. The zoom sessions were good but that 1 to 1 session with the tutor made things easy to understand. Having like a 15 min break every half hour to a hour made things less stressful too and that helped me a lot just have that quick break. I’m still learning on how to process complicated English grammar but it’s better than before I started the course. I’d like to say, thanks to the tutors for their time and helping me progress on my path as a computer technician and yeah. Thanks team!! Sascha Williams

# References

# Appendices