**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***  *Dan Zhang* | 1543359 | Zhangd55@myunitec.ac.nz |  |
| ***Deputy Project Manager***  *[Name]* |  |  |  |
| ***Team Members***  *Dan Zhang* | 1543359 | Zhangd55@myunitec.ac.nz |  |
|  |  |  |  |
|  |  |  |  |

**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](#_Toc88740610)

[1.1 Version History 4](#_Toc88740611)

[1.2 Contribution to Report sections 4](#_Toc88740612)

[1.3 Glossary 4](#_Toc88740613)

[2. Executive Summary 5](#_Toc88740614)

[3. Introduction 6](#_Toc88740615)

[4. Technology Review 7](#_Toc88740616)

[5. IT Methodology 9](#_Toc88740617)

[6. Project Management 10](#_Toc88740618)

[6.1 Project Management Narrative 10](#_Toc88740619)

[6.2 Project Plan with Milestones 11](#_Toc88740620)

[6.3 Project Governance Responsibilities 13](#_Toc88740621)

[6.4 Project Meetings 13](#_Toc88740622)

[6.5 Project Reports 13](#_Toc88740623)

[6.6 Project Risk and Issue Analysis 13](#_Toc88740624)

[7. Requirements Analysis 14](#_Toc88740625)

[7.1 Introduction 14](#_Toc88740626)

[7.2 Use Case Diagram 15](#_Toc88740627)

[7.3 Business Use Case Narratives (Descriptions) 16](#_Toc88740628)

[7.4 Activity Diagrams 24](#_Toc88740629)

[7.5 Overall Class Diagram 32](#_Toc88740630)

[8. Project Design 33](#_Toc88740631)

[8.1 Introduction 33](#_Toc88740632)

[8.2 Software List 33](#_Toc88740633)

[8.3 Version Control Software 33](#_Toc88740634)

[8.4 Design Use Case Narratives (Descriptions) 34](#_Toc88740635)

[8.5 Sequence Diagrams 42](#_Toc88740636)

[8.6 Deployment Diagram 50](#_Toc88740637)

[8.7 Database Design 51](#_Toc88740638)

[8.7.1 Logical ERD 51](#_Toc88740639)

[8.7.2 Dictionary 52](#_Toc88740640)

[8.8 Annotated User Interface Designs 55](#_Toc88740641)

[8.9 Test Plan 59](#_Toc88740642)

[9. Project Training 60](#_Toc88740643)

[9.1 End User Background and Training Objectives 60](#_Toc88740644)

[9.2 Training Materials 60](#_Toc88740645)

[9.3 Training Deliverables 60](#_Toc88740646)

[9.4 Evaluation 60](#_Toc88740647)

[10. Conclusion & Lessons Learned 61](#_Toc88740648)

[References 62](#_Toc88740649)

[Appendices 63](#_Toc88740650)

# 1. Document Control

## 1.1 Version History

This document has had the following revisions:

| **Version** | **Date** | **Author** | **Description of Change** |
| --- | --- | --- | --- |
| 0.1 | 25/11/2021 | Dan Zhang | Initial draft |

## 1.2 Contribution to Report sections

| **Project Team Member name** | **Student ID** | **Report Section** |
| --- | --- | --- |
| Dan Zhang | 1543359 | all |
|  |  |  |
|  |  |  |
|  |  |  |

## 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 report contacts 10 main part and summary as below:

#### Introduction

Briefly introduce the implementation purpose and basic background of this project

#### Technology Review

Through the investigation and analysis of current technologies and similar project, select the technical suitable for the project。

#### IT Methodology

Select systems development lifecycle and mapping of project phases to selected SDLC

#### Project Management

Develop a project management plan suitable for this project.

#### Requirement analysis

Analysis of project requirements and finish use case diagram, overall class diagram, business use case descriptors (narratives), and use case activity diagrams

#### Project Design

Design the model for the solutions(s) to the project based on the project requirements and methodology.

#### Project Training

Identify the end user and provide training materials

#### Conclusion and lessons learned

Overview of the project, show whether the project is successful or a failure and recommendations for next steps as a result of this project.

#### References

List the references of this report

#### Appendices

Supplementary materials relevant to the project

# 3. Introduction

This project is to integrate a single computer technology through a specific project. Through this project, project planning and management skills can be used to design and monitor information system solutions. The technical skills are applied to the information system project, and the document skills are finally applied to generate technical reports for the information system project. At the same time learn to apply communication and problem-solving skills in a professional way to obtain information system results. (Simon,D.,2021)

This project is based on the NZ horse racing company, it is set up the horse racing meetings. And they want a system to manage the meetings and bookings. Their requirement for this system is easy to use for no technical people and could handle and recover any data lost face to the event of a failure.

This project will go through planning, requirement analysis, project design, application development and testing, and project training stages based on the IT system development lifecycle. Finally, deliverable the application and reports are generated.

# 4. Technology Review

According to statistics from the American job search website indeed. Python, Java, and C# are the three most recognized languages ​​in the market. These languages ​​are currently being widely used, but they also have their own advantages and disadvantages and applicable scenarios. **Python** due to easy to learn and its [simple syntax](https://www.codingdojo.com/blog/top-7-programming-languages), a [large library of standards and toolkits](https://towardsdatascience.com/top-10-in-demand-programming-languages-to-learn-in-2020-4462eb7d8d3e). Currently widely used, its limitation is that it cannot be used for mobile program development. **Java** is the programming language that is easiest to integrate with client-server applications and is currently widely used by major international companies. But it is not suitable for running on the cloud, and its owner will charge a license fee. (11 Most In-Demand Programming Languages in 2021, 2021)**C#** It is fully integrated with Microsoft's .NET software framework and supports the development of applications for Windows, browser plug-ins and mobile devices. But it is not easy to learn, especially when it comes to solving errors. (Eastwood, 2020)

Currently the more popular IDEs are Visio studio and PyCharm. **Microsoft's** **Visual Studio** is very lightweight and faster than PyCharm. Microsoft's IDE has a wide range of extensions, add-ons, and other libraries. **PyCharm** is tailor-made for the developer of python, it can manually add a virtual environment and set its interpretation as the default value. (Yang, 2021)

For database access methods, there are online method and local method can choose. The most widely used are AJAX for online and Microsoft Access for local. **AJAC** is a web development technology that uses the network to store and process data. It is a positive help for large projects that require teamwork and can improve the efficiency of project integration. **Microsoft Office Access** is a database management system that Microsoft combines the graphical user interface of the database engine with software development tools. Access has powerful data processing and data analysis functions and supports the development of simple enterprise management projects. Easy to learn, software development can be done without knowing any programming language.

The computer technology used by the two companies currently operating is listed below：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Company Name | Web site | Web Framework | Programming Language | JavaScript Frameworks |
| Love Racing.NZ | https://loveracing.nz | Bootstrap | ASP.NET 4.0.30319 | jQuery 1.7.1  jQuery UI 1.9.1  Modernizr |
| Hawke’s Bay Racing | https://hawkesbayracing.co.nz | Nginx 1.14.0 | PHP 7.1.0 | jQuery 1  Spin JS |

(Loveracing.nz, 2021)

(hawkesbayracing.co.nz, 2021)

Based on the above information, the summary analysis is as follows: Each technology has its advantages and disadvantages and usage scenarios. The specific technology to be used should be comprehensively considered based on the size of the client's company, the way of personnel collaboration, the configuration of hardware and network equipment, and the needs of the client.

Regarding NZ horse racing agency situation, there are multiple roles, and the permissions vary based on their roles. The staff need to collaborate with each other to get the jobs done by following specific workflows. An application / system with all those rules built in will make daily based jobs much easier and more accurate to improve their customer experience. It will also make the organisation more efficient and in turn save running costs.

Back to the project design and implementation, web application works better for this multiple-user scenario. More benefits it brings is ease of deployment and accessibility on any smart device without annoying installation process. Considering it is a typical CRUD (Create, Insert, Update and Delete) project, Python and Flask can cover all the requirements.

# 5. IT Methodology

In the systems development lifecycle it consists of planning, requirement analysis, design, implementation and testing stages. (Systems development life cycle, 2021)The different arrangement way to organize the stages in systems development lifecycle can be consider 3 main model in IT methodology, they are Agile, Waterfall and Asnan. (Asmo, 2018)

Base the requirement form customer and after analysis each methodology. Waterfall methodology is much more match with this project. (Content, 2021)Here are the reasons in blow:

* This project is a small teamwork. There are only 4 people in one team for this project.
* Phases are processed and completed one at a time.
* A schedule can be set with deadlines for each stage of development. (Lei, S., 2021).

# 6. Project Management

## 6.1 Project Management Narrative

*This project follows blow outline. and it* *followed the selected systems development lifecycle*

* *Planning*

*Do the technology review, choose IT methodology and make plan of project management.*

* *Analysis*

*Analysis the customer requirement, finish use case diagram, overall class diagram, business use case descriptors (narratives), and use case activity diagrams for this project.*

* *Design*

*According to the customer requirement, finish design-level use case descriptors, sequence diagrams, deployment diagram with descriptions, database design, annotated interface designs, test plan (place in appendices)*

* *Coding*

*Finish the application development.*

* *Testing*

*Test the application whether archive the design result and match the customer requirement.*

* *Training Materials*

*Make the training video.*

* *Technical Report*

*Finish technical report.*

## 6.2 Project Plan with Milestones

### Week1:

***Project Initialization:***

* Information gathering
* Read all the use case materials

***Planning***

* Technology review
* IT Methodology

### Week2:

***Requirement Analysis***

* Use case diagram
* Overall class diagram
* Use case diagram
* Activity diagrams
* Create business use case narratives

### Week3 & week4:

***Project Design***

* Software list
* Design-level use case descriptions
* Sequence diagram
* Deployment diagram
* Database design
* Annotated interface designs
* Test plan

### Week5 & week6:

***Development***

* Implementation (coding) & Testing

### Week7:

***Deployment***

* hardware configuration
* network configuration
* software configuration

***Launch***

* User acceptance testing
* Project Training

**The Gantt chart can be found in Appendices folder.**

## 6.3 Project Governance Responsibilities

Dan Zhang was responsible for project management and quality assurance. Because this is a single person project.

## 6.4 Project Meetings

**The Meeting Minutes can be found in Appendices folder**.

*Meetings Schedule*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *Week* | *Date* | Duration | Participants | Type |
| *3* | 20/10/2021 | 15minutes | Simon Dacey and Dan Zhang | online meeting |
| *4* | 27/10/2021 | 15minutes | Lei Song and Dan Zhang | online meeting |
| *5* | 28/10/2021 | 15minutes | Simon Dacey and Dan Zhang | online meeting |
| *6* | 11/11/2021 | 15minutes | Lei Song and Dan Zhang | online meeting |
| *7* | 18/11/2021 | 15minutes | Simon Dacey and Dan Zhang | online meeting |

## 6.5 Project Reports

The project status report is help for track and record the status of each stage. Weather has risk or issue while the project progress. For this project, each stage finish on time and smooth execution of each stage. Did not happen expected risk.

**The project status report can be found in Appendices folder.**

## 6.6 Project Risk and Issue Analysis

To ensure the smooth progress of the project, a risk estimate was made for the project. During the execution of the project, no estimated risks occurred, and no issue occurred.

**The project Risk and Issue table can be found in Appendices folder.**

# 7. Requirements Analysis

## 7.1 Introduction

This project was initiated by NZ horsing racing company, the company set up the horse racing meetings. They need a system to management the meetings and bookings. Their requirement for the system is suit for no-technology person to use and can handle and recover any data lost face to the event of a failure.

There are 5 position people use this system. The main user of this project contact salesclerks, racecourse administrator, race manager, equine administrator, and jockey coordinator.

The object of system operation contact racecourse, meeting, customer, owner, booking, horse, jockey, race entry and race. and the main function is to add, update and delete specific items and produce specific items reports.

## 7.2 Use Case Diagram

Diagram

Description automatically generated

## 7.3 Business Use Case Narratives (Descriptions)

### 7.3.1 Add Jockey use case

|  |  |  |
| --- | --- | --- |
| **USE CASE NAME:** | Add Jockey | **USE CASE TYPE** |
| **USE CASE ID:** | 23 | **Business Requirements: 🗹** |
| **PRIORITY:** | High |  |
| **PRIMARY BUSINESS ACTOR:** | Jockey coordinator | |
| **OTHER PARTICIPATING ACTORS:** | None | |
| **DESCRIPTION:** | This use case enables the jockey coordinator to add a jockey’s details | |
| **PRE-CONDITIONS:** | The jockey coordinator has logged onto the system | |
| **TYPICAL COURSE** | Step 1. The jockey coordinator selects the “Add Jockey” function.  Step 2. The system displays the “Add Jockey” form with all fields blank.  Step 3. The jockey coordinator enters the Jockey id, Last Name, First Name, Email Address, Weight and Date of birth.  Step 4. The system confirm that all necessary fields are filled in correctly.  Step 5. The system saves the jockey’s details.  Step 6. The system displays the “Jockey added successfully” message.  Step 7. The system displays the “Add another jockey?” prompt.  Step 8. The jockey coordinator selects to end the use case.  Step 9. The system closes the form. | |
| **OF EVENTS:** |
|  |
|  |
|  |
|  |
| **ALTERNATE COURSES:** | Step 5a.1 The system having identified missing or incorrect fields and prompts for the completion of all fields.  Step 5a.2 The system returns to step 3. | |
| Step 8a.1 The jockey coordinator elects to add another jockey.  Step 8a.2 The system returns to step 2. | |
| **POST CONDITIONS:** | None | |
| **ASSUMPTIONS:** | None | |

### 7.3.2 Update Jockey use case

|  |  |  |
| --- | --- | --- |
| **USE CASE NAME:** | Update Jockey | USE CASE TYPE |
| **USE CASE ID:** | 24 | Business Requirements: 🗹 |
| **PRIORITY:** | High |  |
| **PRIMARY BUSINESS ACTOR:** | Jockey coordinator | |
| **OTHER PARTICIPATING ACTORS:** | None | |
| **DESCRIPTION:** | This use case enables the jockey coordinator to update a selected jockey’s details | |
| **PRE-CONDITIONS:** | The jockey coordinator has logged onto the system | |
| **TYPICAL COURSE** | Step 1. The jockey coordinator selects the “Update Jockey” function.  Step 2. The system displays the “Update Jockey” form.  Step 3. The system displays a list of all the jockeys (jockey id, last name, and first name).  Step 4. The jockey coordinator selects the jockey whose details need updating.  Step 5. The system displays the jockey’s details (jockey’s id, last name, first name, email address, date of birth, and weight).  Step 6. The jockey coordinator updates the relevant details (last name, first name, email address, date of birth, and weight only)  Step 7. The jockey coordinator elects to update the jockey.  Step 8. The system validates the entries in the fields  Step 9. The jockey coordinator confirms the change of details.  Step 10. The system saves the jockey’s details.  Step 11. The system displays the “Jockey updated successfully” message.  Step 12. The system displays the “Update another jockey?” prompt.  Step 13. The jockey coordinator elects to end the use case.  Step 14. The system closes the form. | |
| **OF EVENTS:** |
|  |
|  |
|  |
|  |
| **ALTERNATE COURSES:** | Step 4a.1. The jockey coordinator elects to cancel the operation.  Step 4a.2. The system goes to step 14. | |
| Step 8a.1 The system having identified missing or incorrect fields and prompts for the completion of all fields.  Step 8a.2 The system returns to step 5. | |
| Step 10a.1 The jockey coordinator elects to cancel the change.  Step 10a.2 The system goes to step 14. | |
|  | Step 13a.1 The jockey coordinator elects to update another jockey.  Step 13a.2 The system goes to step 3. | |
| **POST CONDITIONS:** | None | |
| **ASSUMPTIONS:** | None | |

### 7.3.3 Delete Jockey use case

|  |  |  |
| --- | --- | --- |
| **USE CASE NAME:** | Delete Jockey | USE CASE TYPE |
| **USE CASE ID:** | 25 | Business Requirements: 🗹 |
| **PRIORITY:** | High |  |
| **PRIMARY BUSINESS ACTOR:** | Jockey coordinator | |
| **OTHER PARTICIPATING ACTORS:** | None | |
| **DESCRIPTION:** | This use case enables the jockey coordinator to delete a selected jockey’s details | |
| **PRE-CONDITIONS:** | The jockey coordinator has logged onto the system | |
| **TYPICAL COURSE** | Step 1. The jockey coordinator selects the “Delete Jockey” function.  Step 2. The system displays the “Delete Jockey” form.  Step 3. The system displays a list of all the jockeys (jockey id, last name and first name) who are not allocated to any race entries.  Step 4. The jockey coordinator selects the jockey who requires deleting Step 5. The system displays the jockey’s details (jockey ID, last name, first name, weight, and date of birth).  Step 6. The jockey coordinator elects to delete the jockey.  Step 7. The system displays the “Are you sure you want to delete this jockey” prompt.  Step 8. The jockey coordinator confirms the deletion.  Step 7. The system deletes the jockey.  Step 9. The system displays the” Jockey deleted successfully” message.  Step 10. The system displays the “Delete another jockey?” prompt.  Step 11. The jockey coordinator selects to end the use case.  Step 12. The system closes the form. | |
| **OF EVENTS:** |
|  |
|  |
|  |
|  |
| **ALTERNATE COURSES:** | Step 5a.1 The jockey coordinator elects to cancel the operation.  Step 5a.2 The system goes to step 11. | |
| Step 6a.1 The jockey coordinator elects to cancel the operation.  Step 6a.2 The system goes to step 11. | |
|  | Step 9a.1 The jockey coordinator elects to choose a different jockey.  Step 9a.2 The system returns to step 3. | |
|  | Step 10a.1 The jockey coordinator elects to delete another jockey.  Step 10a.2 The system returns to step 3. | |
| **POST CONDITIONS:** | None | |
| **ASSUMPTIONS:** | None | |

### 7.3.4 Update Race Entry use case

|  |  |  |
| --- | --- | --- |
| **USE CASE NAME:** | Update Race Entry | **USE CASE TYPE** |
| **USE CASE ID:** | 27 | **Business Requirements: 🗹** |
| **PRIORITY:** | high |  |
| **PRIMARY BUSINESS ACTOR:** | Race manager | |
| **OTHER PARTICIPATING ACTORS:** | None | |
| **DESCRIPTION:** | This use case enables the race manager to update a selected race entry | |
| **PRE-CONDITIONS:** | The race manager logged onto the system | |
| **TYPICAL COURSE** | Step 1. The race manager selects the “Update Race Entry” function.  Step 2. The system displays the “Update Race Entry” form.  Step 3. The system displays a list of all the scheduled races (race id and race name) that have race entries.  Step 4. The race manager selects the race with the race entry which is need updating.  Step 5. The system displays the race’s details (race id, race name, and status).  Step 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.  Step 7. The race manager selects a race entry.  Step 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).  Step 9. The race manager updates the race entry’s status and race time.  Step 10. The race manager elects to update the race entry.  Step 11. The system updates race entry’s details.  Step 12. The system displays the “Race entry updated successfully” message.  Step 13. The system displays the “Update another race entry?” prompt.  Step 14. The race manager selects to end the use case.  Step 15. The system closes the form. | |
| **OF EVENTS:** |
|  |
|  |
|  |
|  |
| **ALTERNATE COURSES:** | Step 4a.1 The race manager elects to cancel the operation.  Step 4a.2 The system goes to step 15. | |
| Step 10a.1 The race manager elects to cancel the operation.  Step 10a.2 The system goes to step 15. | |
| Step 14a.1 The race manager elects to update another race entry,  Step 14b.2 The system goes to step 3. | |
| **POST CONDITIONS:** | None | |
| **ASSUMPTIONS:** | None | |

### 7.3.5 Add Race use case

|  |  |  |
| --- | --- | --- |
| **USE CASE NAME:** | Add Race | **USE CASE TYPE** |
| **USE CASE ID:** | 29 | **Business Requirements: 🗹** |
| **PRIORITY:** | High |  |
| **PRIMARY BUSINESS ACTOR:** | Race manager | |
| **OTHER PARTICIPATING ACTORS:** | None | |
| **DESCRIPTION:** | This use case enables a race manager to add a race’s details | |
| **PRE-CONDITIONS:** | The race manager logged onto the system | |
| **TYPICAL COURSE** | Step 1. The race manager selects the “Add Race” function.  Step 2. The system displays the “Add Race” form with all fields blank.  Step 3. The race manager enters the race’s details (race id, race name, race time, status, and race type).  Step 4. The system displays a list of meetings (meeting id and meeting name).  Step 5. The race manager selects a meeting.  Step 6. The race manager selects to add the race.  Step 7. The system confirm that all necessary fields are filled in correctly.  Step 8. The system saves the race’s details.  Step 9. The system displays the “Race added successfully” message.  Step 10. The system displays the “Add another race?” prompt.  Step 11. The race manager selects to end the use case.  Step 12. The system closes the form. | |
| **OF EVENTS:** |
|  |
|  |
|  |
|  |
| **ALTERNATE COURSES:** | Step 7a.1 The race manager elects to cancel the operation.  Step 7a.2 The system goes to step 12. | |
| Step 8a.1 The system having identified missing or incorrect fields and prompts for the completion of all fields.  Step 8a.2 The system returns to step 3. | |
| Step 11a.1 The race manager elects to add another race.  Step 11a.2 The system returns to step 2. | |
| **POST CONDITIONS:** | None | |
| **ASSUMPTIONS:** | None | |

### 7.3.6 Update Race use case

|  |  |  |
| --- | --- | --- |
| **USE CASE NAME:** | Update Race | USE CASE TYPE |
| **USE CASE ID:** | 30 | Business Requirements: 🗹 |
| **PRIORITY:** | High |  |
| **PRIMARY BUSINESS ACTOR:** | Race manager | |
| **OTHER PARTICIPATING ACTORS:** | None | |
| **DESCRIPTION:** | This use case enables a race manager to update a selected race’s details | |
| **PRE-CONDITIONS:** | The race manager logged onto the system | |
| **TYPICAL COURSE** | Step 1. The race manager selects the “Update Race” function.  Step 2. The system displays the “Update Race” form.  Step 3. The system displays a list of all the scheduled races (race id and race name).  Step 4. The race manager selects the race that has details that needs updating.  Step 5. The system displays the race’s details (race id, race name, race time, status, race type, and meeting name).  Step 6. The race manager updates the relevant details (race name, race time, status, and race type only).  Step 7. The system validates the entries in the fields.  Step 8. The system prompts for confirmation to change the race’s details.  Step 9. The system saves the race’s details.  Step 10. The system displays the “Race updated successfully” message.  Step 11. The system displays the “Update another race?” prompt.  Step 12. The race manager elects to end the use case.  Step 13. the system closes the form. | |
| **OF EVENTS:** |
|  |
|  |
|  |
|  |
| **ALTERNATE COURSES:** | Step 5a.1 The race manager elects to cancel the operation.  Step 5a.2 The system goes to step 13. | |
| Step 8a.1 the system having identified missing or incorrect fields, prompts for completion of the entry.  Step 8a.2.the system return to step 6. | |
| Step 9a.1 The race manager elects to cancel the operation.  Step 9a.2 The system goes to step 13. | |
| Step 12a.1 The race manager elects to update another race.  Step 12a.2 The system goes to step 3. | |
| **POST CONDITIONS:** | None | |
| **ASSUMPTIONS:** | None | |

### 7.3.7 Delete Race use case

|  |  |  |
| --- | --- | --- |
| **USE CASE NAME:** | Delete Race | USE CASE TYPE |
| **USE CASE ID:** | 31 | Business Requirements: 🗹 |
| **PRIORITY:** | High |  |
| **PRIMARY BUSINESS ACTOR:** | Race manager | |
| **OTHER PARTICIPATING ACTORS:** | None | |
| **DESCRIPTION:** | This use case enables a race manager to delete a selected race’s details | |
| **PRE-CONDITIONS:** | The race manager logged onto the system | |
| **TYPICAL COURSE** | Step 1. The race manager selects the “Delete Race” function.  Step 2. The system displays the “Delete Race” form.  Step 3. The system displays a list of all the finished races (race id and race name).  Step 4. The race manager selects the race that requires deleting.  Step 5. The system displays the race’s details (race id, race name, and meeting name).  Step 6. The race manager elects to delete the race.  Step 7. The system display “Are you sure you want to delete this race?” prompt.  Step 8. The race manager confirms the deletion.  Step 9. The system deletes all the details of race entries associated with the race.  Step 10. The system deletes the race’s details.  Step 11. The system displays the “Race deleted successfully” message.  Step 12. The system displays the “Delete another race?” prompt.  Step 13. The race manager elects to end the use case.  Step 14. The system closes the form. | |
| **OF EVENTS:** |
|  |
|  |
|  |
|  |
| **ALTERNATE COURSES:** | Step 5a.1 The race manager elects to cancel the operation.  Step5a.2 The system goes to step 14. | |
| Step 7a.1 The race manager elects to cancel the operation.  Step 7a.2 The system goes to step 14. | |
| Step 9a.1 The race manager elects to choose a different race to delete.  Step 9a.2 The system goes to step 3. | |
|  | Step 13a.1 The race manager elects to delete another race.  Step 13a.2 The system goes to step 3. | |
| **POST CONDITIONS:** | None | |
| **ASSUMPTIONS:** | None | |

### 7.3.8 Produce Race Report use case

|  |  |  |
| --- | --- | --- |
| **USE CASE NAME:** | Produce Races Report | **USE CASE TYPE** |
| **USE CASE ID:** | 32 | **Business Requirements: 🗹** |
| **PRIORITY:** | High |  |
| **PRIMARY BUSINESS ACTOR:** | Race manager | |
| **OTHER PARTICIPATING ACTORS:** | None | |
| **DESCRIPTION:** | This use case enables a race manager to produce the races report | |
| **PRE-CONDITIONS:** | The race manager logged onto the system | |
| **TYPICAL COURSE** | Step 1. The race manager selects the “Races Report” function.  Step 2. The system displays the “Races Report” form.  Step 3. The race manager selects the option to generate the report.  Step 4. The system gets the details (race ID, race name, race time, status race type, and meeting name) of each race.  Step 5. The system gets the details (horse name, status, jockey last name and jockey first name) of each race entry for each race.  Step 6. The system then generates the race report (race ID, race name, race time, status, race type, meeting name, horse name, status, jockey last name and jockey first name) with the races sorted by race name.  Step 7. The system closes the form. | |
| **OF EVENTS:** |
|  |
|  |
|  |
|  |
| **ALTERNATE COURSES:** | Step 3a.1 The race manager elects to cancel without generating the report.  Step 3a.2 The system goes to step 7. | |
| **POST CONDITIONS:** | None | |
| **ASSUMPTIONS:** | None | |

## 7.4 Activity Diagrams

### 7.4.1 Add Jockey use case

Diagram

Description automatically generated

### 7.4.2 Update Jockey use case

Diagram

Description automatically generated

### 7.4.3 Delete Jockey use case

Diagram

Description automatically generated

### 7.4.4 Update Race Entry use case

Diagram

Description automatically generated

### 7.4.5 Add Race use case

Diagram

Description automatically generated

### 7.4.6 Update Race use case

Diagram

Description automatically generated

### 7.4.7 Delete Race use case

Diagram

Description automatically generated

### 7.4.8 Produce Race Report use case

Diagram

Description automatically generated

## 7.5 Overall Class Diagram

Diagram

Description automatically generated

# 8. Project Design

## 8.1 Introduction

*According to the analysis of the company structure, user types and customer needs, it is necessary to design a solution model for the project. The solution needs to be specific to the system and design the model for the solutions to the project based on the project requirements and methodology.*

*This includes:*

* Software list
* Version control software
* Design-level use case descriptors
* Sequence diagrams
* Deployment diagram with descriptions
* Database design
* Annotated interface designs
* Test plan design

## 8.2 Software List

* Database design: Microsoft Access
* Development software: Visual studio 2019
* Development Language: C#
* User interface design: Microsoft Power Point

## 8.3 Version Control Software

Use GitHub to manage files for data integration at all stages.

## 8.4 Design Use Case Narratives (Descriptions)

### 8.4.1 Add Jockey use case

|  |  |  |
| --- | --- | --- |
| **USE CASE NAME:** | Add Jockey | **USE CASE TYPE** |
| **USE CASE ID:** | 23 | **Design Requirements: 🗹** |
| **PRIORITY:** | High |  |
| **PRIMARY BUSINESS ACTOR:** | Jockey Coordinator | |
| **OTHER PARTICIPATING ACTORS:** | None | |
| **DESCRIPTION:** | This use case enables the jockey coordinator to add a jockey’s details. | |
| **PRE-CONDITIONS:** | The jockey coordinator logged into the system | |
| **TYPICAL COURSE** | Step 1. The jockey coordinator selects the “Add Jockey” function.  Step 2. The system displays the “Add Jockey” form (Last Name, First Name, Email Address, Weight and Date of birth) with all fields blank.  Step 3. The jockey coordinator enters the fields.  Step 4. The jockey coordinator clicks on the “Add Jockey” button  Step 5. The system validates the data to confirm that all necessary fields are filled in correctly.  Step 6. The system gets a unique jockey id for the new jockey.  Step 7. The system saves the jockey’s details (Jockey ID, Last Name, First Name, Email Address, Weight and Date of birth).  Step 8. The system displays the “Jockey added successfully” message.  Step 9. The jockey coordinator clicks the “Return” button.  Step 11. The system closes the form to end the use case. | |
| **OF EVENTS:** |
|  |
|  |
|  |
|  |
| **ALTERNATE COURSES:** | Step 6a. The system having identified missing or incorrect fields and display the “Please fill in all fields correctly” message.  Step 6a. The system returns to step3. | |
| Step 9a1. The jockey coordinator elects to enter another jockey  Step 9a2. The system returns to step2. | |
| **POST CONDITIONS:** | None | |
| **ASSUMPTIONS:** | None | |

### 8.4.2 Update Jockey use case

|  |  |  |
| --- | --- | --- |
| USE CASE NAME: | Update Jockey | USE CASE TYPE |
| USE CASE ID: | 24 | Design Requirements: 🗹 |
| PRIORITY: | High |  |
| PRIMARY BUSINESS ACTOR: | Jockey Coordinator | |
| OTHER PARTICIPATING ACTORS: | None | |
| DESCRIPTION: | This use case enables the jockey coordinator to update a selected jockey’s details. | |
| PRE-CONDITIONS: | The jockey coordinator logged into the system | |
| TYPICAL COURSE | Step 1. The jockey coordinator selects the “Update Jockey” function.  Step 2. The system displays the “Update Jockey” form.  Step 3. The system gets all the jockeys.  Step 4. The system displays the list of all the jockeys (jockey id, last name, and first name) in a list box.  Step 5. The jockey coordinator selects the jockey whose details need updating.  Step 6. The system gets the details of the selected jockey.  Step 7. The system displays the jockey’s details (jockey’s id, last name, first name, email address, date of birth, and weight).  Step 8. The jockey coordinator updates the relevant details (last name, first name, email address, date of birth, and weight only)  Step 9. The jockey coordinator clicks “Update Jockey” button.  Step 10. The system validates the entries in the fields.  Step 11. The system asks for confirmation to change the jockey’s details.  Step 12. The jockey coordinator confirms the change of details  Step 13. The system saves the jockey’s details.  Step 14. The system displays the “Jockey updated successfully” message.  Step 15. The jockey coordinator clicks the “Return” button.  Step 16. The system closes the form to end the use case. | |
| OF EVENTS: |
|  |
|  |
|  |
|  |
| ALTERNATE COURSES: | Step 11a1. The system having identified missing or incorrect fields and display the “Please fill in all fields correctly” message.  Step 11a2. The system returns to step 3. | |
| Step 12a1. The jockey coordinator clicks “Cancel” button.  Step 12a2.The system goes to step 16. | |
| Step 15a1.The jockey coordinator elects to update another jockey,  Step 15a2.The system returns to step 3. | |
| POST CONDITIONS: | None | |
| ASSUMPTIONS: | None | |

### 8.4.2 Delete Jockey use case

|  |  |  |
| --- | --- | --- |
| USE CASE NAME: | Delete Jockey | USE CASE TYPE |
| USE CASE ID: | 25 | Design Requirements: 🗹 |
| PRIORITY: | High |  |
| PRIMARY BUSINESS ACTOR: | Jockey coordinator | |
| OTHER PARTICIPATING ACTORS: | None | |
| DESCRIPTION: | This use case enables the jockey coordinator to delete a selected jockey’s details | |
| PRE-CONDITIONS: | The jockey coordinator has logged onto the system | |
| TYPICAL COURSE | Step 1. The jockey coordinator selects the “Delete Jockey” function.  Step 2. The system displays the “Delete Jockey” form.  Step 3. The system gets all the jockeys who are not allocated to any race.  Step 4. The system displays the list of all the jockeys (jockey id, last name and first name) who are not allocated to any race entries in a list box.  Step 5. The jockey coordinator selects the jockey who requires deleting.  Step 6. The system gets the details of selected jockey.  Step 7. The system displays the jockey’s details (jockey ID, last name, first name, weight, and date of birth) in read-only text boxes.  Step 8. The jockey coordinator clicks on the “Delete Jockey” button.  Step 9. The system displays the “Are you sure you want to delete this jockey?” prompt.  Step 10. The jockey coordinator confirms the deletion.  Step 11. The system deletes the jockey’s details.  Step 12. The system displays the “Jockey deleted successfully” message.  Step 13. The jockey coordinator clicks on the “Return” button.  Step 14. The system closes the form to end the use case. | |
| OF EVENTS: |
|  |
|  |
|  |
|  |
| ALTERNATE COURSES: | Step 10a.1 The jockey coordinator clicks on the “Cancel” button.  Step 10a.2 The system returns to step 3. | |
| Step 10b.1 The jockey coordinator clicks on the “No” button to choose a different jockey.  Step 10b.2 The system returns to step 3. | |
|  | Step 13a.1 The jockey coordinator elects to delete another jockey.  Step 13a.2 The system returns to step 3. | |
| POST CONDITIONS: | None | |
| ASSUMPTIONS: | None | |

### 8.4.2 Update Race Entry use case

|  |  |  |
| --- | --- | --- |
| USE CASE NAME: | Update Race Entry | USE CASE TYPE |
| USE CASE ID: | 27 | Design Requirements: 🗹 |
| PRIORITY: | high |  |
| PRIMARY BUSINESS ACTOR: | Race manager | |
| OTHER PARTICIPATING ACTORS: | None | |
| DESCRIPTION: | This use case enables the race manager to update a selected race entry | |
| PRE-CONDITIONS: | The race manager logged onto the system | |
| TYPICAL COURSE | Step 1. The race manager selects the “Update Race Entry” function.  Step 2. The system displays the “Update Race Entry” form.  Step 3. The system gets all the scheduled races.  Step 4. The system displays the list of all the scheduled races (race id and race name) that have race entries in the list box.  Step 5 The race manager selects the race with the race entry which is need updating.  Step 6. The system gets the details of the selected race.  Step 7. The system displays the race’s details (race id, race name, and status).  Step 8. The system gets all the race entries associated with the selected race.  Step 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 in the list box.  Step 10. The race manager selects a race entry.  Step 11. The system gets the details of the selected race entry.  Step 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).  Step 13. The race manager updates the race entry’s status and race time.  Step 14. The race manager clicks” Update race entry” button.  Step 15. The system validates the entries in the fields.  Step 16. The system asks for confirmation to change the race entry’s details.  Step 17. The race manager confirms the change of details  Step 18. The system updates race entry’s details.  Step 19. The system displays the “Race entry updated successfully” message.  Step 20. The race manager clicks on the “Return” button.  Step 21. The system closes the form to end the use case. | |
| OF EVENTS: |
|  |
|  |
|  |
|  |
| ALTERNATE COURSES: | Step 16a.1 The system having identified missing or incorrect fields and display the “Please fill in all fields correctly” message.  Step 16a.2 The system returns to step 3. | |
| Step 17a.1 The race manager clicks “Cancel” button.  Step 17a.2 The system goes to step 20. | |
| Step 20a.1 The race manager elects to update another race entry,  Step 20b.2 The system goes to step 3. | |
| POST CONDITIONS: | None | |
| ASSUMPTIONS: | None | |

### 8.4.2 Add Race use case

|  |  |  |
| --- | --- | --- |
| **USE CASE NAME:** | Add Race | **USE CASE TYPE** |
| **USE CASE ID:** | 29 | **Design Requirements: 🗹** |
| **PRIORITY:** | High |  |
| **PRIMARY BUSINESS ACTOR:** | Race manager | |
| **OTHER PARTICIPATING ACTORS:** | None | |
| **DESCRIPTION:** | This use case enables a race manager to add a race’s details | |
| **PRE-CONDITIONS:** | The race manager logged onto the system | |
| **TYPICAL COURSE** | Step 1. The race manager selects the “Add Race” function.  Step 2. The system displays the “Add Race” form with all fields (race name, race time, status, and race type) blank.  Step 3. The race manager enters the race’s details (race name, race time, status, and race type).  Step 4. The system gets all meetings.  Step 5. The system displays a list of meetings (meeting id and meeting name) in the list box.  Step 6. The race manager selects a meeting.  Step 7. The race manager clicks on the “Add Race” button.  Step 8. The system validates the data to confirm that all necessary fields are filled in correctly.  Step 9. The system gets a unique race id for the new race.  Step 10. The system saves the race’s details (race id, race name, race time, status, race type, and meeting id).  Step 11. The system displays the “Race added successfully” message.  Step 12. The race manager clicks on the “Return” button.  Step 13. The system closes the form. | |
| **OF EVENTS:** |
|  |
|  |
|  |
|  |
| **ALTERNATE COURSES:** | Step 9a1. The system having identified missing or incorrect fields and display the “Please fill in all fields correctly” message.  Step 9a2. The system returns to step3. | |
| Step12a1. The race manager elects to enter another race.  Step12a2. The system returns to step2. | |
| **POST CONDITIONS:** | None | |
| **ASSUMPTIONS:** | None | |

### 8.4.2 Update Race use case

|  |  |  |
| --- | --- | --- |
| **USE CASE NAME:** | Update Race | **USE CASE TYPE** |
| **USE CASE ID:** | 30 | **Design Requirements: 🗹** |
| **PRIORITY:** | High |  |
| **PRIMARY BUSINESS ACTOR:** | Race manager | |
| **OTHER PARTICIPATING ACTORS:** | None | |
| **DESCRIPTION:** | This use case enables a race manager to update a selected race’s details | |
| **PRE-CONDITIONS:** | The race manager logged onto the system | |
| **TYPICAL COURSE** | Step 1. The race manager selects the “Update Race” function.  Step 2. The system displays the “Update Race” form.  Step 3. The system gets all the scheduled races.  Step 4. The system displays the list of all the scheduled races (race id and race name) in a list box.  Step 5. The race manager selects the race that has details that needs updating.  Step 6. The system gets the detail of selected race.  Step 7. The system displays the race’s details (race id, race name, race time, status, race type, and meeting name).  Step 8. The race manager updates the relevant details (race name, race time, status, and race type only).  Step 9. The race manager clicks on the “Update Race” button.  Step 10. The system validates the entries in the fields.  Step 11. The system prompts for confirmation to change the race’s details.  Step 12. The race manager confirms the change of details.  Step 13. The system saves the race’s details.  Step 14. The system displays the “Race updated successfully” message.  Step 15. The race manager clicks the “Return” button.  Step 16. The system closes the form to end the use case. | |
| **OF EVENTS:** |
|  |
|  |
|  |
|  |
| **ALTERNATE COURSES:** | Step 11a.1 The system having identified missing or incorrect fields and display the “Please fill in all fields correctly” message.  Step 11a.2.the system return to step 7. | |
| Step 12a.1 The race manager clicks on the “Cancel” button.  Step 12a.2 The system goes to step 15. | |
|  | Step 15a.1 The race manager elects to update another race.  Step 15a.2 The system goes to step 2. | |
| **POST CONDITIONS:** | None | |
| **ASSUMPTIONS:** | None | |

### 8.4.2 Delete Race use case

|  |  |  |
| --- | --- | --- |
| **USE CASE NAME:** | Delete Race | USE CASE TYPE |
| **USE CASE ID:** | 31 | Design Requirements: 🗹 |
| **PRIORITY:** | High |  |
| **PRIMARY BUSINESS ACTOR:** | Race manager | |
| **OTHER PARTICIPATING ACTORS:** | None | |
| **DESCRIPTION:** | This use case enables a race manager to delete a selected race’s details | |
| **PRE-CONDITIONS:** | The race manager logged onto the system | |
| **TYPICAL COURSE** | Step 1. The race manager selects the “Delete Race” function.  Step 2. The system displays the “Delete Race” form.  Step 3. The system gets all the finished races.  Step 4. The system displays the list of all the finished races (race id and race name) in a list box.  Step 5. The race manager selects the race that requires deleting.  Step 6. The system gets the details of the selected race.  Step 7. The system displays the race’s details (race id, race name, and meeting name).  Step 8. The race manager clicks on the “Delete Customer” button.  Step 9. The system display “Are you sure you want to delete this race?” prompt.  Step 10. The race manager confirms the deletion.  Step 11. The system deletes all the details of race entries associated with the race.  Step 12. The system deletes the race’s details.  Step 13. The system displays the “Race deleted successfully” message.  Step 14. The race manager clicks on the “Return” button.  Step 15. The system closes the form. | |
| **OF EVENTS:** |
|  |
|  |
|  |
|  |
| **ALTERNATE COURSES:** | Step 10a.1 The race manager clicks on the “Cancel” button.  Step 10a.2 The system goes to step 15. | |
| Step 10b.1 The race manager clicks on the “No” button to choose a different race to delete.  Step 11b.2 The system goes to step 3. | |
| Step 14a.1 The race manager elects to delete another race.  Step 14a.2 The system goes to step 3. | |
| **POST CONDITIONS:** | None | |
| **ASSUMPTIONS:** | None | |

### 8.4.2 Produce Race Report use case

|  |  |  |
| --- | --- | --- |
| **USE CASE NAME:** | Produce Races Report | USE CASE TYPE |
| **USE CASE ID:** | 32 | Design Requirements: 🗹 |
| **PRIORITY:** | High |  |
| **PRIMARY BUSINESS ACTOR:** | Race manager | |
| **OTHER PARTICIPATING ACTORS:** | None | |
| **DESCRIPTION:** | This use case enables a race manager to produce the races report | |
| **PRE-CONDITIONS:** | The race manager logged onto the system | |
| **TYPICAL COURSE** | Step 1. The race manager selects the “Races Report” function.  Step 2. The system displays the “Races Report” form.  Step 3. The race manager clicks on the “Races Report” button.  Step 4. The system gets the details (race ID, race name, race time, status race type, and meeting name) of each race.  Step 5. The system gets the details (horse name, status, jockey last name and jockey first name) of each race entry for each race.  Step 6. The system then displays the race report (race ID, race name, race time, status, race type, meeting name, horse name, status, jockey last name and jockey first name) with the races sorted by race name.  Step 7. The race manager clicks on the “Return” button.  Step 8. The system closes the form. | |
| **OF EVENTS:** |
|  |
|  |
|  |
|  |
| **ALTERNATE COURSES:** | Step 3a.1 The race manager elects to cancel without generating the report.  Step 3a.2 The system goes to step 7. | |
| **POST CONDITIONS:** | None | |
| **ASSUMPTIONS:** | None | |

## 8.5 Sequence Diagrams

### 8.5.1 Add Jockey use case

Timeline

Description automatically generated with medium confidence

### 8.5.2 Update Jockey use case

Timeline

Description automatically generated

### 8.5.3 Delete Jockey use case

Diagram, timeline

Description automatically generated

### 8.5.4 Update Race Entry use case

Timeline

Description automatically generated

### 8.5.5 Add Race use case

Timeline

Description automatically generated with medium confidence

### 8.5.6 Update Race use case

Timeline

Description automatically generated

### 8.5.7 Delete Race use case

Diagram, timeline

Description automatically generated

### 8.5.8 Produce Race Report

Diagram

Description automatically generated with medium confidence

## 8.6 Deployment Diagram

Graphical user interface, application, PowerPoint

Description automatically generated

## 8.7 Database Design

### 8.7.1 Logical ERD

**Diagram

Description automatically generated**

### 8.7.2 Dictionary

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Race Course** | | | | | |
| Fields Name | Required | Data Type | Maximum Length | Default | Range/List |
| RaceCourseID | Yes | Auto-number | 3 |  | Primary Key |
| RaceCourseName | Yes | Short Text | 30 |  |  |
| StreetAddress | Yes | Short Text | 60 |  |  |
| Suburb | Yes | Short Text | 20 |  |  |
| City | Yes | Short Text | 20 |  |  |
| PhoneNumber | Yes | Short Text | 16 |  | (999)000-0000 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Meeting** | | | | | |
| Fields Name | Required | Data Type | Maximum Length | Default | Range/List |
| MeetingID | Yes | Auto-number | 6 |  | Primary Key |
| MeetingName | Yes | Short Text | 60 |  |  |
| MeetingDate | Yes | Date/Time | 10 |  | DD/MM/YYYY |
| TicketPrice | Yes | Currency | 6 | 40 | $40-$1500 inclusive |
| Capacity | Yes | Number | 6 | 500 | 500-250000 inclusive |
| RaceCourseID | Yes | Foreign Key  (RACE COURSE) | 3 |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Customer** | | | | | |
| Fields Name | Required | Data Type | Maximum Length | Default | Range/List |
| CustomerID | Yes | Auto-number | 8 |  | Primary Key |
| LastName | Yes | Short Text | 30 |  |  |
| FirstName | Yes | Short Text | 30 |  |  |
| StreetAddress | Yes | Short Text | 60 |  |  |
| Suburb | Yes | Short Text | 20 |  |  |
| City | Yes | Short Text | 20 |  |  |
| EmailAddress | Yes | Short Text | 30 |  |  |
| PhoneNumber | Yes | Short Text | 16 |  | (999)000-0000 |
| CreditStatus | Yes | Short Text | 7 | Valid | Valid or Invalid |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Booking** | | | | | |
| Fields Name | Required | Data Type | Maximum Length | Default | Range/List |
| BookingID | Yes | Auto-number | 10 |  | Primary Key |
| Quantity | Yes | Number | 2 | 1 | 1-4 inclusive |
| Status | Yes | Short Text | 7 | Paid | Pending or Paid |
| BookingDate | Yes | Date/Time | 10 |  | DD/MM/YYYY |
| CustomerID | Yes | Foreign Key(CUSTOMER) | 8 |  |  |
| MeetingID | Yes | Foreign Key(MEETING) | 6 |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Owner** | | | | | |
| Fields Name | Required | Data Type | Maximum Length | Default | Range/List |
| OwnerID | Yes | Auto-number | 4 |  | Primary Key |
| LastName | Yes | Short Text | 30 |  |  |
| FirstName | Yes | Short Text | 30 |  |  |
| StreetAddress | Yes | Short Text | 60 |  |  |
| Suburb | Yes | Short Text | 20 |  |  |
| City | Yes | Short Text | 20 |  |  |
| EmailAddress | Yes | Short Text | 30 |  |  |
| PhoneNumber | Yes | Short Text | 16 |  | (999)000-0000 |
| NZRAMember | Yes | Yes/No | 3 | Yes | Yes or No |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Horse** | | | | | |
| Fields Name | Required | Data Type | Maximum Length | Default | Range/List |
| HorseID | Yes | Auto-number | 6 |  | Primary Key |
| HorseName | Yes | Short Text | 50 |  |  |
| Gender | Yes | Short Text | 7 | Male | Male, Female or Gelding |
| DateOfBirth | Yes | Date/Time | 10 |  | DD/MM/YYYY |
| OwnerID | Yes | Foreign Key(OWNER) | 4 |  |  |

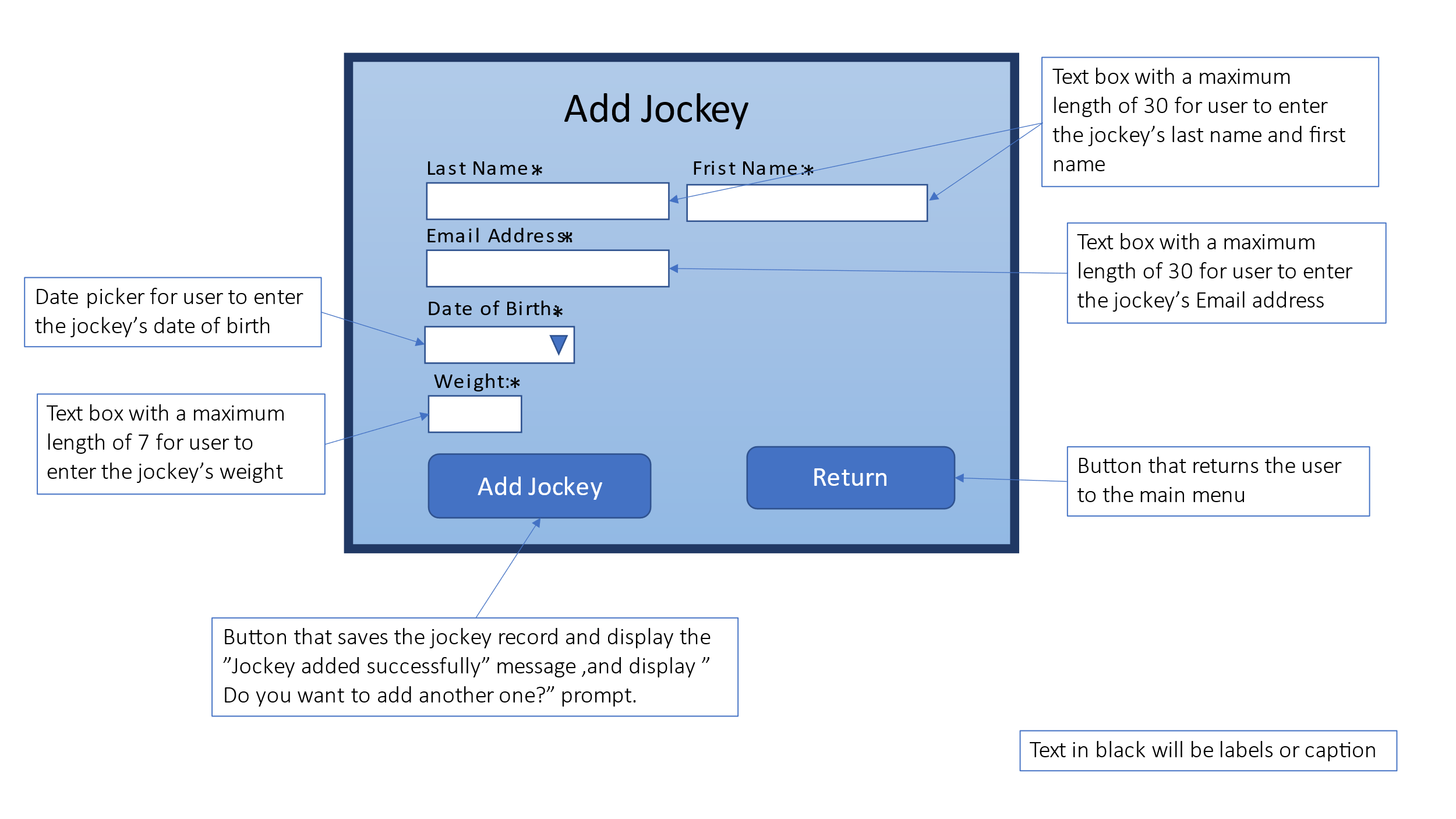
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Jockey** | | | | | |
| Fields Name | Required | Data Type | Maximum Length | Default | Range/List |
| JockeyID | Yes | Auto-number | 6 |  | Primary Key |
| LastName | Yes | Short Text | 30 |  |  |
| FirstName | Yes | Short Text | 30 |  |  |
| EmailAddress | Yes | Short Text | 30 |  |  |
| Weight | Yes | Short Text | 7 | 44 | between 44 and 55 inclusive |
| DateOfBirth | Yes | Date/Time | 10 |  | DD/MM/YYYY |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Race Entry** | | | | | |
| Fields Name | Required | Data Type | Maximum Length | Default | Range/List |
| RaceEntryID | Yes | Auto-number | 10 |  | Primary Key |
| Status | Yes | Short Text | 12 | Confirmed | Pending, Disqualified or Confirmed |
| RaceEntryTime | Yes | Short Text | 4 | 0 |  |
| RaceID | Yes | Foreign Key(RACE) | 8 |  |  |
| HorseID | Yes | Foreign Key(HORSE) | 6 |  |  |
| JockeyID | Yes | Foreign Key(JOCKEY) | 6 |  |  |

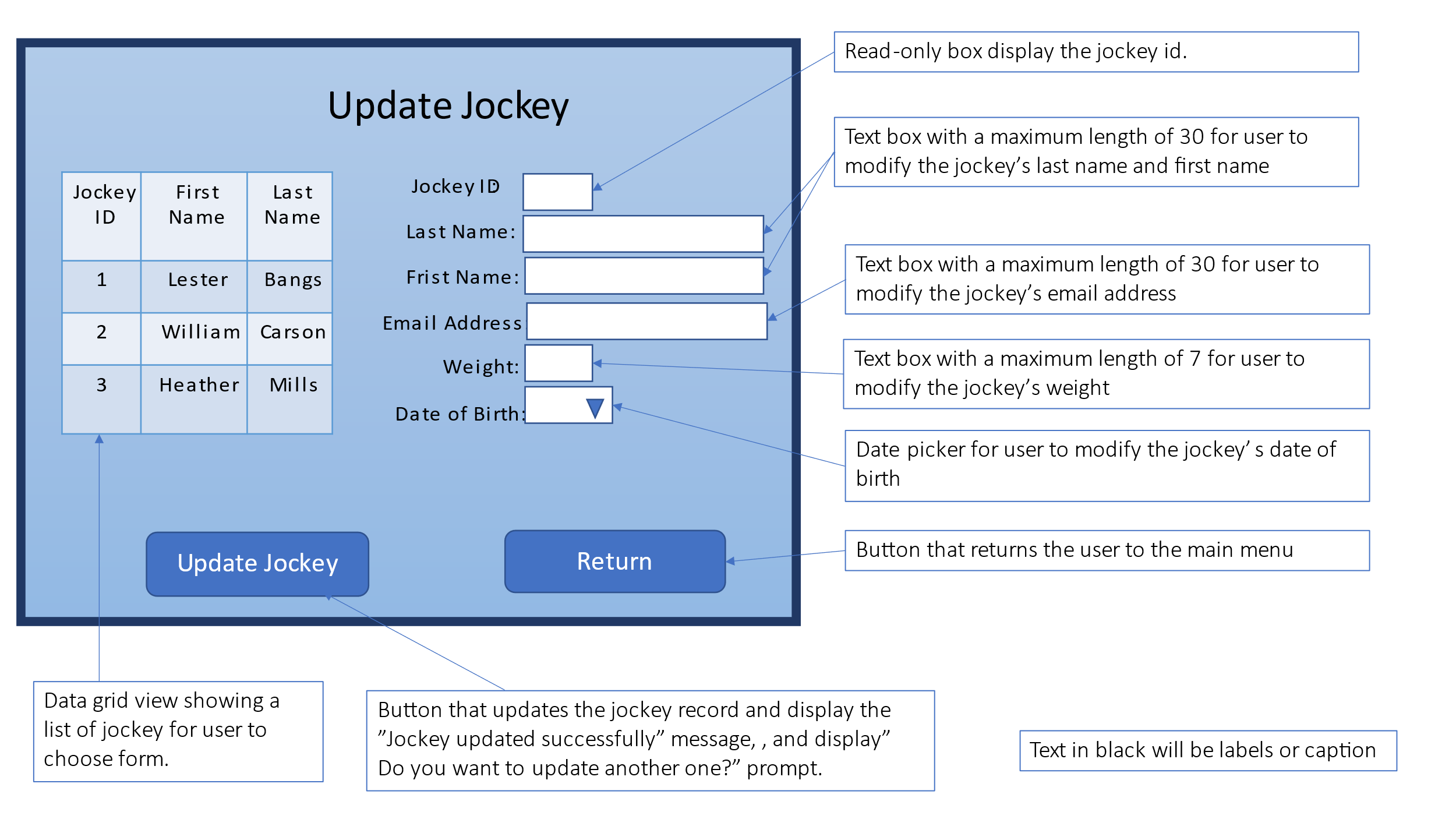
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Race** | | | | | |
| Fields Name | Required | Data Type | Maximum Length | Default | Range/List |
| RaceID | Yes | Auto-number | 8 |  | Primary Key |
| RaceName | Yes | Short Text | 20 |  |  |
| RaceTime | Yes | Time/Date | 6 |  | Format: HH:MM, range 9:00 to 19:00 inclusive |
| Status | Yes | Short Text | 9 | Finished | Scheduled or Finished |
| RaceType | Yes | Short Text | 8 | Flat | Hurdles, Chase, Handicap or Flat |
| MeetingID | Yes | Foreign Key(MEETING) | 6 |  |  |

## 8.8 Annotated User Interface Designs

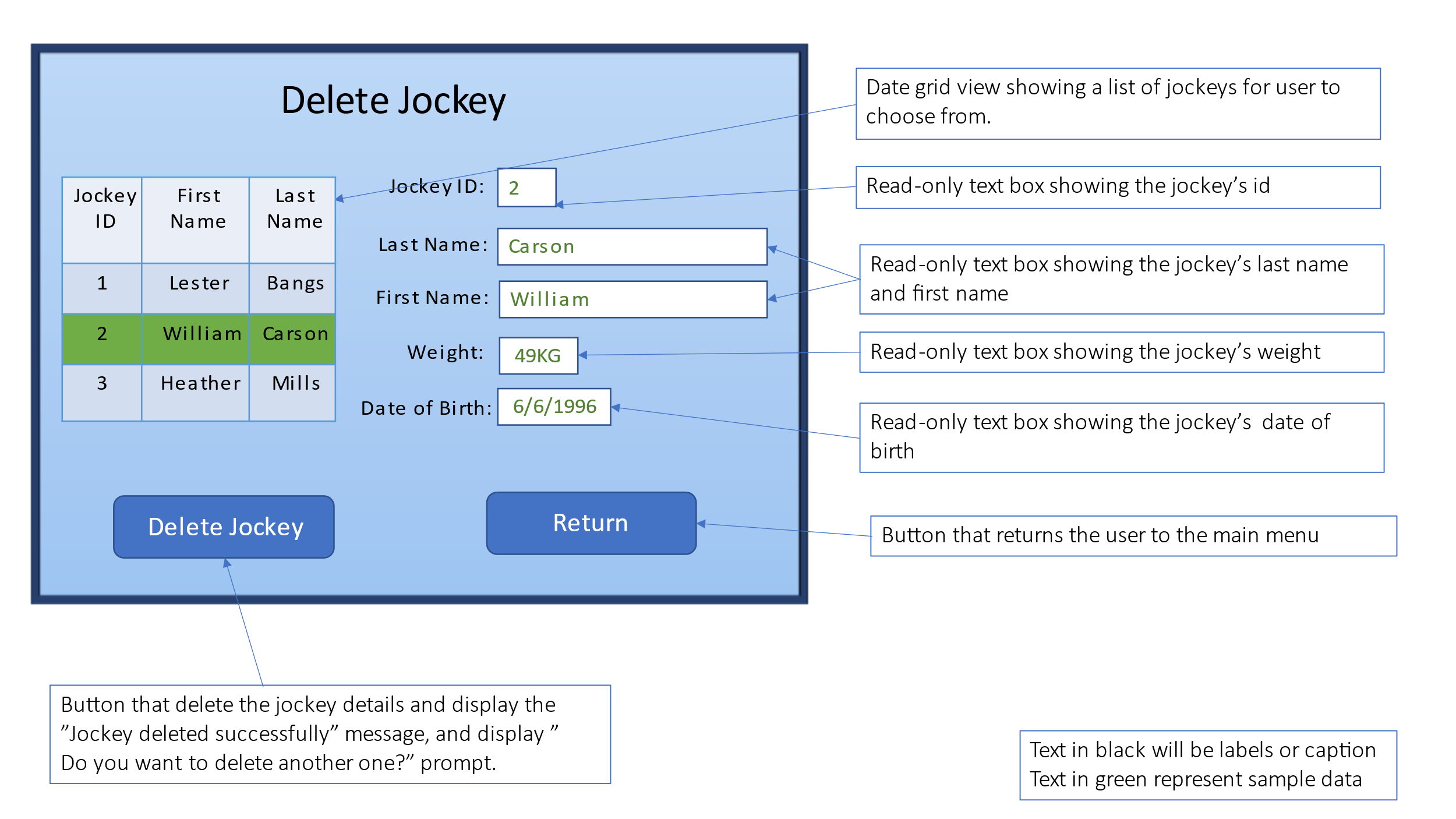
### 8.8.1 Add Jockey use case



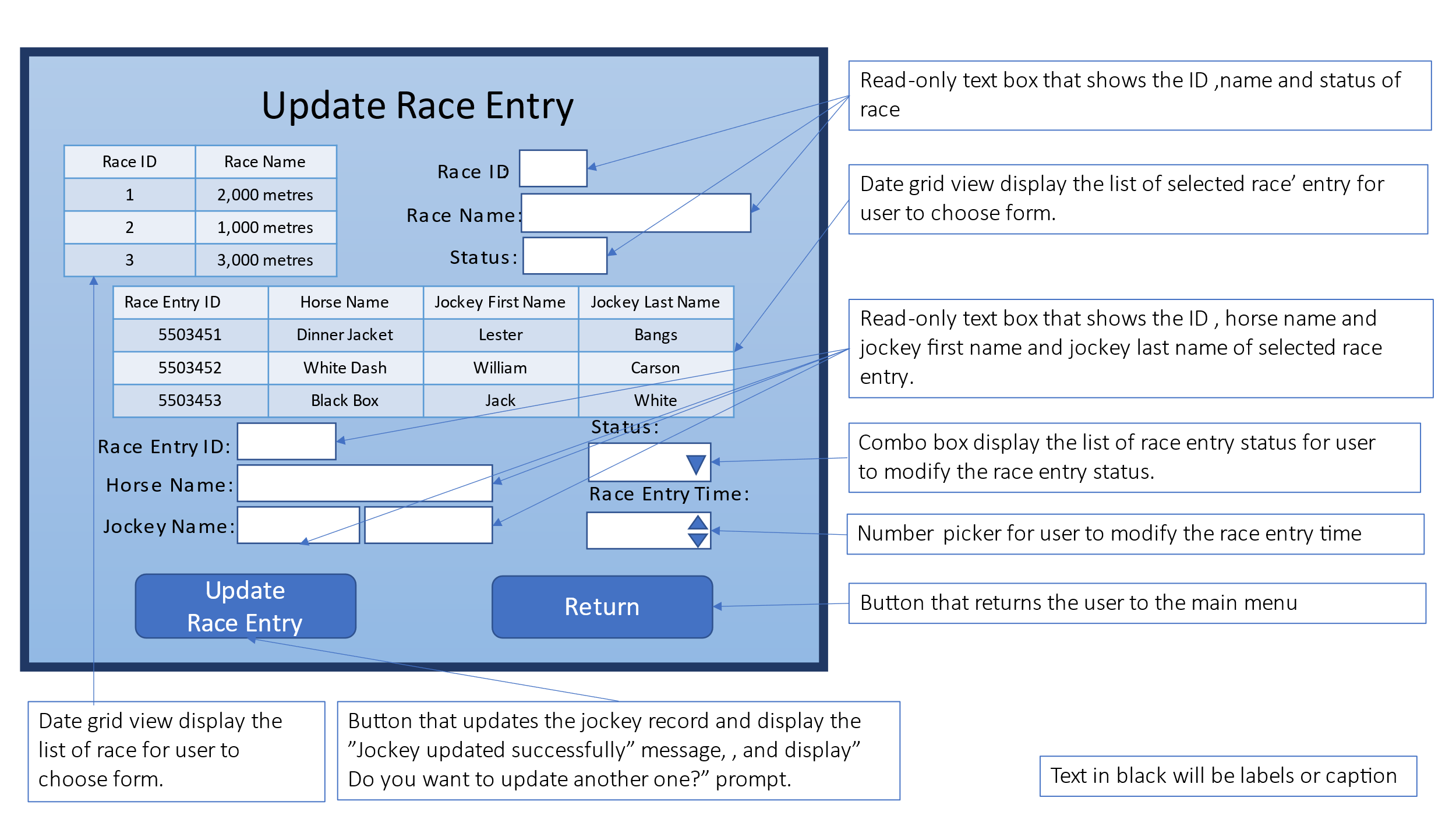
### 8.8.2 Update Jockey use case



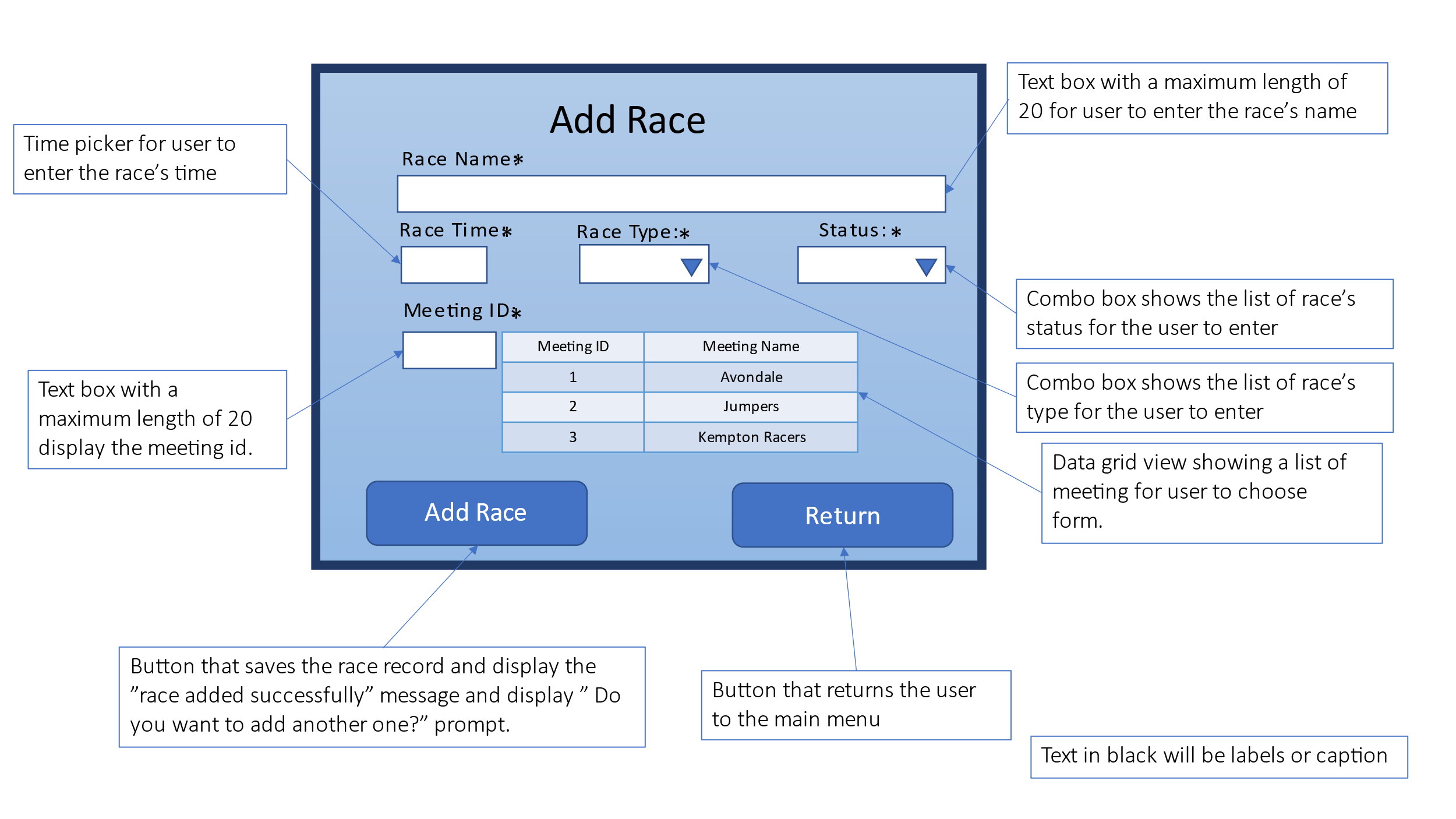
### 8.8.3 Delete Jockey use case



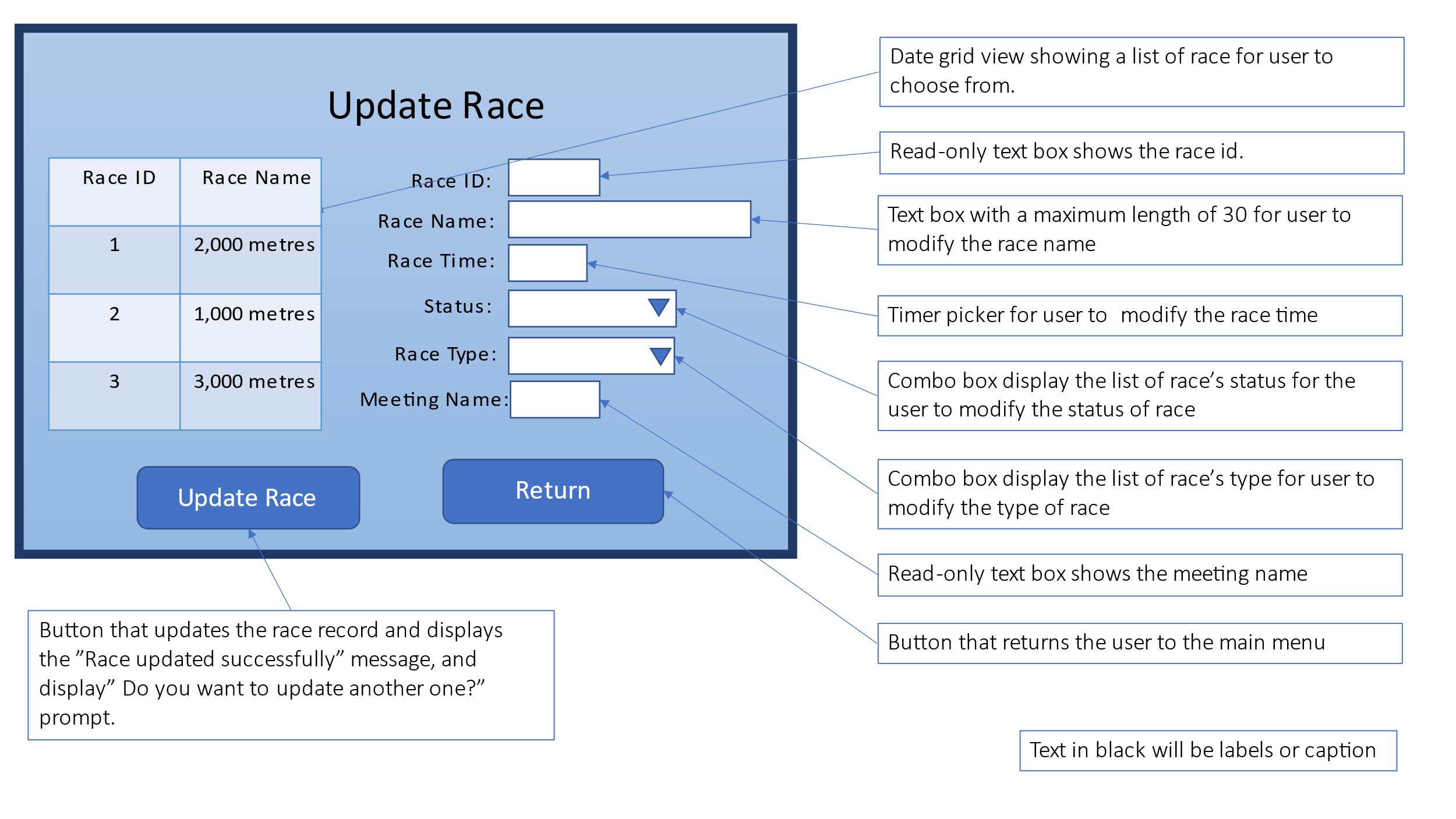
### 8.8.4 Update Race Entry use case



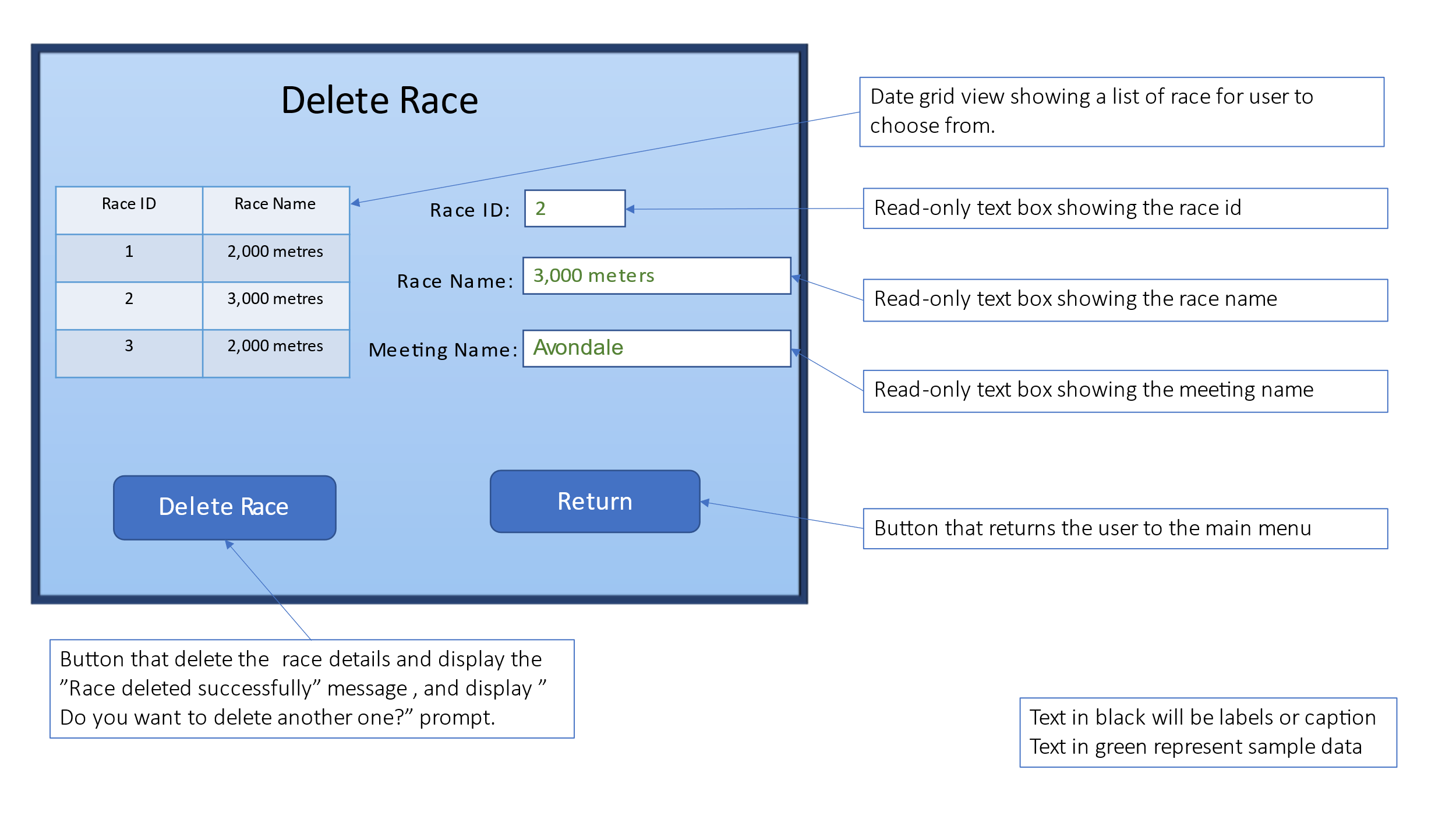
### 8.8.5 Add Race use case



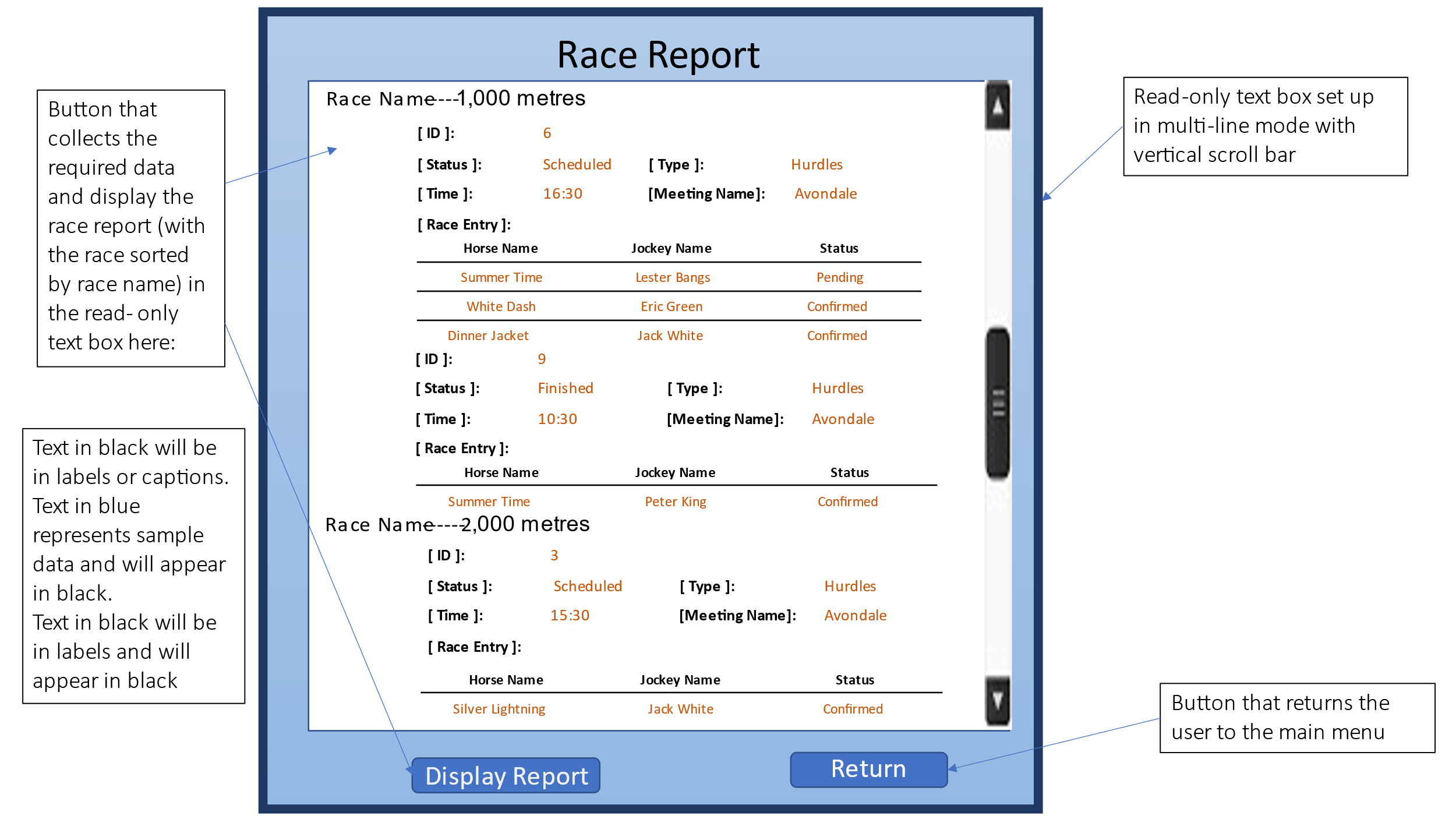
### 8.8.6 Update Race use case



### 8.8.7 Delete Race use case



### 8.8.8 Produce Race Report use case



## 8.9 Test Plan

### 8.9.1 Table test plan

**The table test plan table can be found in Appendices folder.**

### 8.9.2 System test plan

**The system test plan table can be found in Appendices folder.**

# 9. Project Training

## 9.1 End User Background and Training Objectives

The end user is the manager and staff that work in the NZ horse racing company. All the staff in the company need know this system. There are 5 poison staff is the directly user of this system. They are the racecourse administrator, the sale clerks, the race manager, the equine administrator, and the jockey coordinator. They need suppose giving a training of how to use the system.

## 9.2 Training Materials

For end users to understand and use the system faster and more effectively, the development team has produced training videos.

## 9.3 Training Deliverables

**The training video can be found in Appendices folder.**

## 9.4 Evaluation

# 10. Conclusion & Lessons Learned

Through this time hard working, the project was finally completed. During this time, we first conducted research on the market and similar products and specified our project plan and project management plan based on this. Through the analysis of customer needs, the application of project planning and management skills to design and monitor information system solution projects. After that, technical skills in database design and development and software development were applied to the project. Finally, the information system project generates a technical report.

This project integrated the individual technologies I learned before and gave me a more intuitive understanding of the IT systems development lifecycle. And can apply communication and problem-solving skills in a professional way to obtain information system results.

During the whole project, I can make a reasonable time plan to ensure that each stage is completed on time. But on the other hand, the functions that can be achieve in the application development process are not considered very well in the design phase. That is what I am not doing enough.

There are still some aspects of this project that need to be improved. For software development, it would be more reasonable if the "Race Report" form interface is added the "Print" function. For myself, through technology review, it is shown that python is more suitable for developing this project, which is also the development language I want to learn in the future.

# References

*11 Most In-Demand Programming Languages in 2021*. (2021). Retrieved from Berkeley Extension: https://bootcamp.berkeley.edu/blog/most-in-demand-programming-languages/

(2021). Retrieved from Loveracing.nz: https://loveracing.nz/Home.aspx

(2021). Retrieved from hawkesbayracing.co.nz: https://hawkesbayracing.co.nz

Asmo. (2018, March 9). *7 Popular Project Management Methodologies And What They’re Best Suited For*. Retrieved from Zenkit Blog: https://zenkit.com/en/blog/7-popular-project-management-methodologies-and-what-theyre-best-suited-for/

Content, L. (2021). *What the Waterfall Project Management Methodology Can (and Can’t) Do for You*. Retrieved from Lucidchart: https://www.lucidchart.com/blog/waterfall-project-management-methodology

Eastwood, B. (2020, june 18). *The 10 Most Popular Programming Languages to Learn in 2021*. Retrieved from Northeastern University: https://www.northeastern.edu/graduate/blog/most-popular-programming-languages/

*Systems development life cycle*. (2021). Retrieved from wikipedia.org: https://en.wikipedia.org/wiki/Systems\_development\_life\_cycle

Yang, D. (2021). *The 9 Best Programming Languages to Learn in 2021*. Retrieved from Fullstack Academy: https://www.fullstackacademy.com/blog/nine-best-programming-languages-to-learn

Lei, S. (2021).  [Project Management SlidesFile](https://moodle.unitec.ac.nz/mod/resource/view.php?id=606970) [Microsoft PowerPoint]. HTCS5607 IS Project. Retrieved from <https://moodle.unitec.ac.nz/>

Simon, D. (2021).  [Introduction to Course SlidesFile](https://moodle.unitec.ac.nz/mod/resource/view.php?id=606970) [Microsoft PowerPoint]. HTCS5607 IS Project. Retrieved from https://moodle.unitec.ac.nz/

# Appendices

* Gantt Chart

File name:

Gantt Chart.xlsx

* Meeting Minus

File name:

Meeting Minutes Week 3 Zhang, Dan.docx

Meeting Minutes Week 5 Zhang, Dan.docx

Meeting Minutes Week 7 Zhang, Dan.docx

* Project Status Report

File name:

Project Status Report-05112012 .doc

Project Status Report-19112012 .doc

Project Status Report-22102012 .doc

* Project Risks & Issues Register

File name:

Project Risks & Issues Register.xlsx

* Test plan

File name:

System test plan.xls

Table test plan.xls

* Training Materials

File name:

Training video.mp4