RugBot – Testing Plan

Deliverable 4 – ITSP300 – 2018

W.P. RUGBY ACADEMY

RugBot Development Team
Group 2

Group number:	2
Group name:	RugBot
Members:	**Student number:** XQ9X3WV31 **Name:** Matthew **Surname:** Van der Bijl
	Student number: MB2015-0023 **Name:** Tyler **Surname:** Gray
	Student number: PXHTJDCN5 **Name:** Stefanus **Surname:** Buys
	Student number: MB2014-0695 **Name:** Abongile **Surname:** Mdleleni

Customer:	**Full Name:** Angelo Nelson **Company:** WP Rugby Academy **Industry:** Sport Science

Table of Contents

1. Information System Design 1
1.1 Logical Design 1
Use Case Diagrams 1
Class Diagram 4
Activity Diagrams 5
Sequence Diagrams 8
1.2 Physical Design 11
1.2.1 Technologies 11
1.2.2 System Testing 12
Testing Types 12
Testing Template 14
Test Plan 15
Bibliography 27

Table of Figures

Figure 1 Coach User-case diagram 1
Figure 2 Physiotherapist Use-case diagram 2
Figure 3 Player Use-case diagram 3
Figure 4 Class Diagram of RugBot System 4
Figure 5 Login activity diagram 5
Figure 6 Injured player activity diagram 6
Figure 7 Registration activity diagram 7
Figure 8 Registration sequence diagram 8
Figure 9 Login sequence diagram 9
Figure 10 Injured player activity diagram 10

Tables of Tables

Table 1 Testing types 12
Table 2 Testing template 14
Table 3 Sign in testing plan 15
Table 4 Data validation sign in test 16
Table 5 Home page test plan 17
Table 6 Calendar page test plan 18
Table 7 Attendance page test plan 19
Table 8 Injury page test plan 20
Table 9 Game admin page test plan 21
Table 10 Homepage test plan for physio 22
Table 11 Injury page test plan for physio 23
Table 12 Homepage test plan for players 24
Table 13 Calendar page test plan for players 25
Table 14 Test plan for match day page for players 26

1. Information System Design

1.1 Logical Design

The following three use-case diagrams model user various users will interact with the system. There is one use-case diagram for each user of the system, coach, physiotherapist and player respectively.

Use Case Diagrams

Coaches will use the system to track player attendance and view the status of injured players. Coach is able to take attendance with the app. Only coaches and physios will be able to view a list of all players. The use-case diagram below was created using a format suggested by Bennet, et al. (2010).

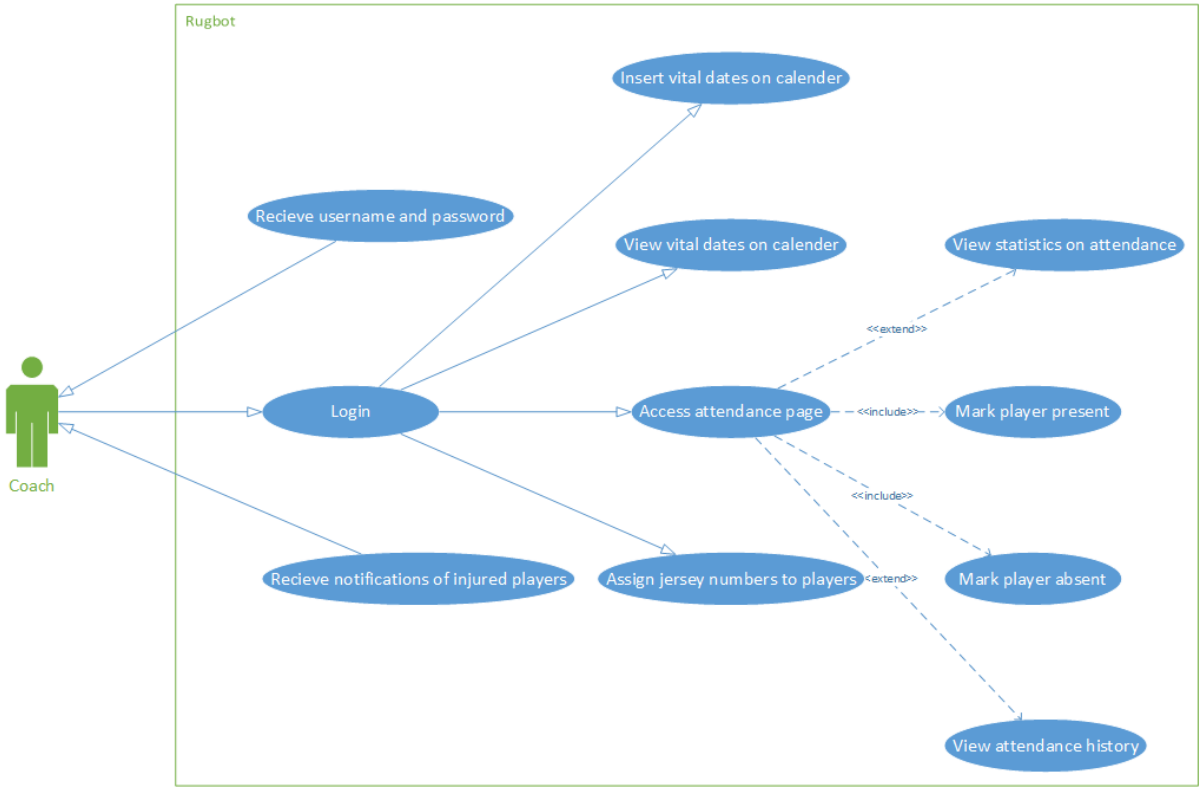

Figure 1 Coach User-case diagram

As seen above, once a coach registers they are able to login into the system. Coaches should be notified when a player's injury status is altered.

Through the rugby academy, players have access to a physiotherapist. These physiotherapists can mark players as injured if they are unable to endure in matches. The use-case diagram below was created using a format suggested by Bennet, et al. (2010).

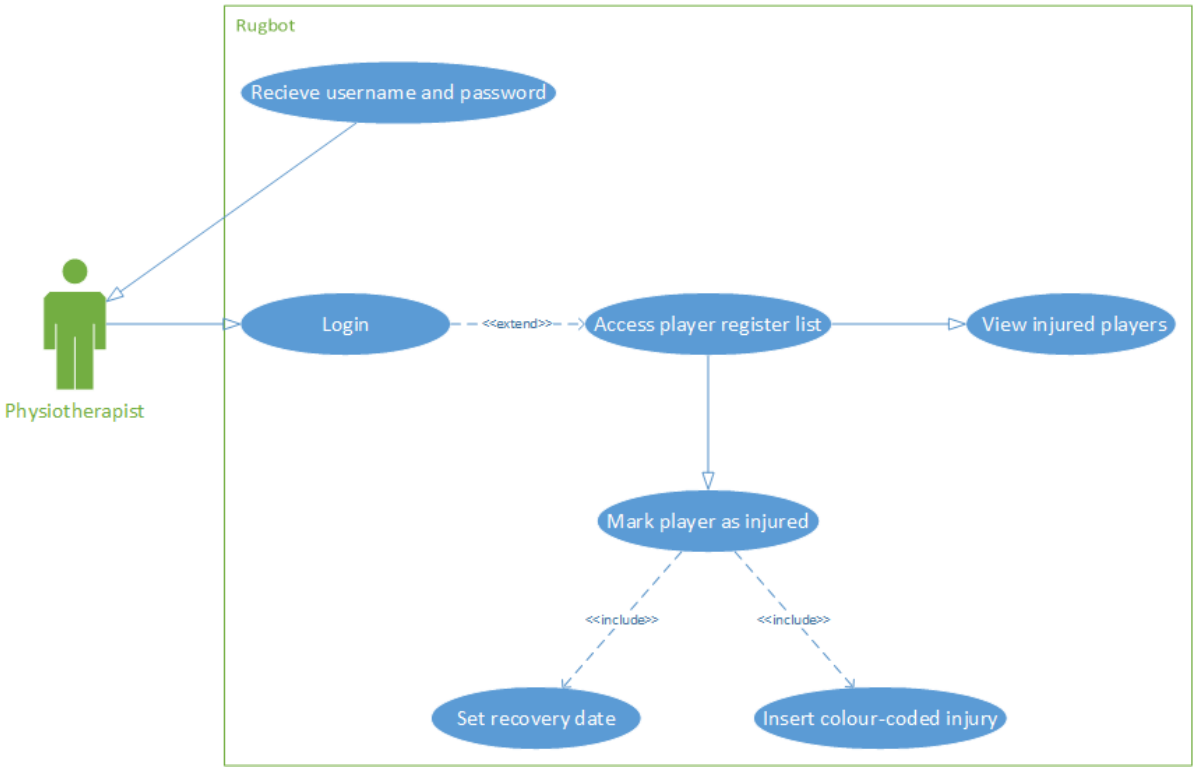

Figure 2 Physiotherapist Use-case diagram

Physiotherapists have access to a list of all registered players. They can mark a player as injured and set an estimated recovery date. Once a player has been marked as injured they are barred from competing.

Of all users, players have the least functionality. Once registers, players are able to log in and see all the information that is relative to them. Once registered, the player should receive their login details. The use-case diagram below was created using a format suggested by Bennet, et al. (2010).

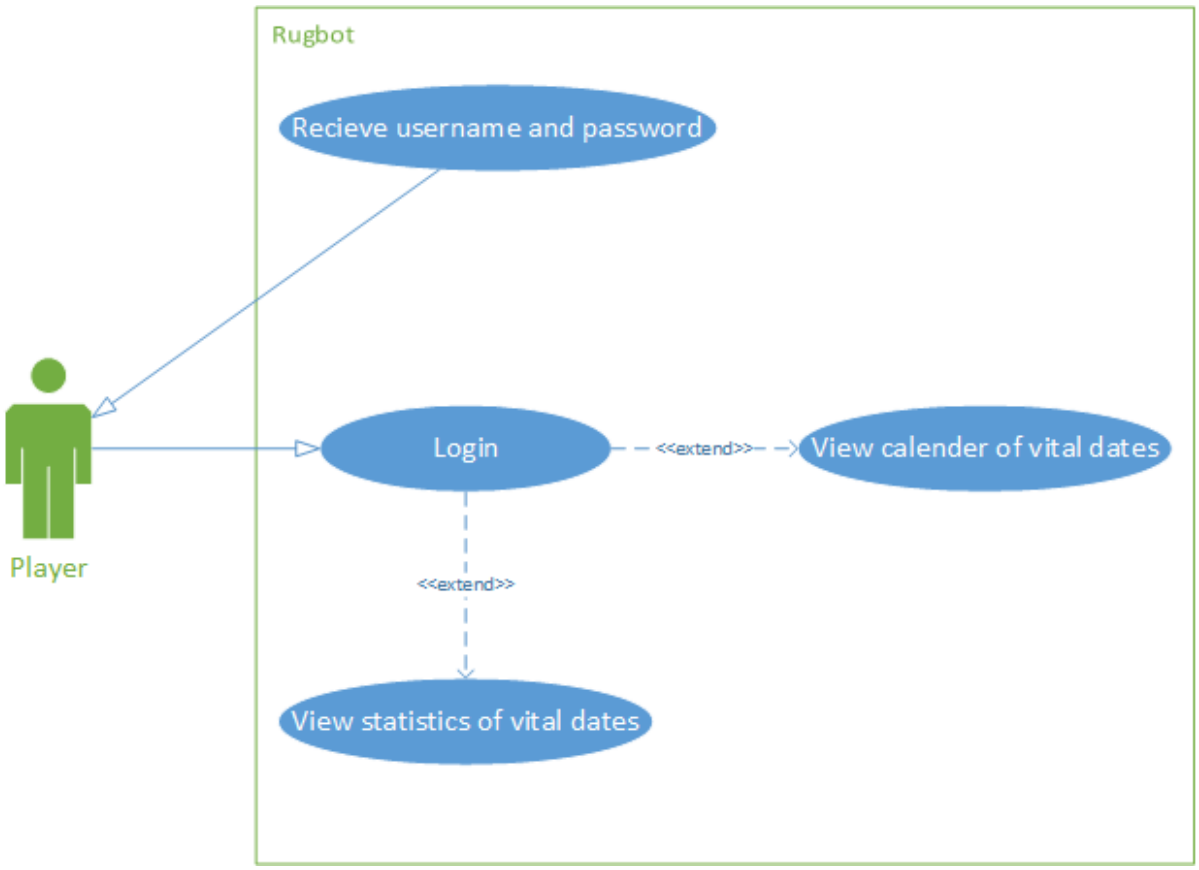

Figure 3 Player Use-case diagram

As seen above, once a player registers they are able to login into the system. Players should be notified when their injury status is altered. Players should be able to view a calendar of all upcoming match fixtures that they are participating in.

Class Diagram

The class diagram below depicts all classes that exist within the RugBot system was created using a format suggested by Pretorius and Erasmus (2012).

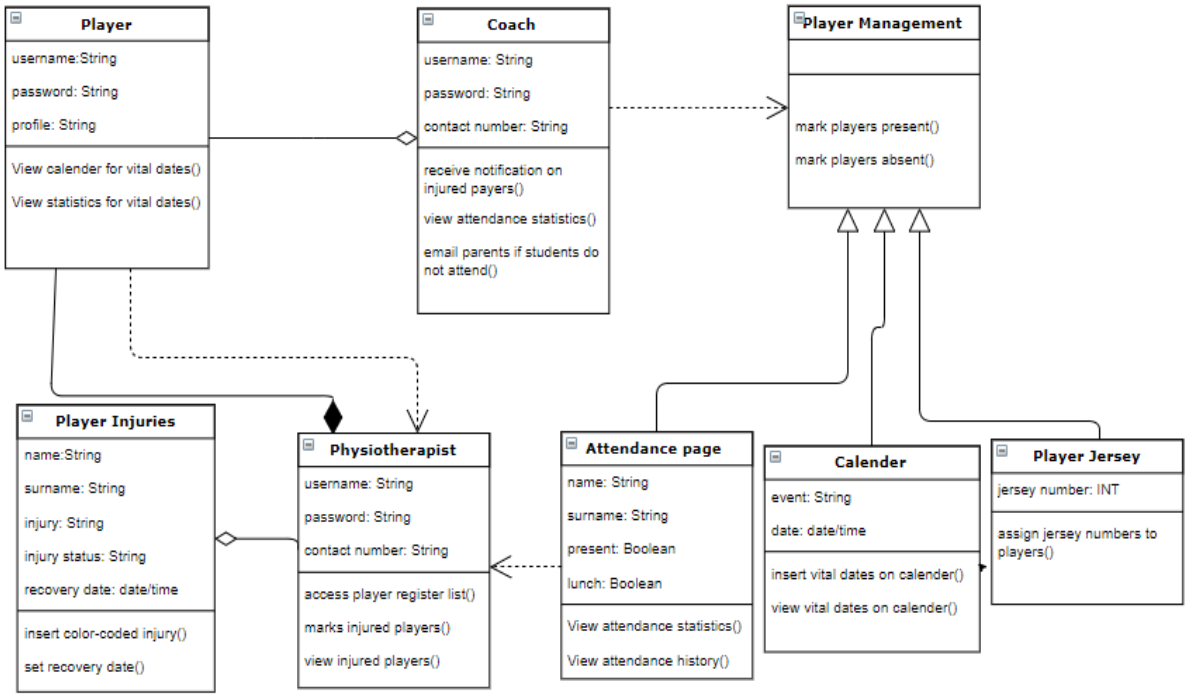

Figure 4 Class Diagram of RugBot System

The classes may change while developing the project.

Activity Diagrams

Login Activity Diagram

The activity diagram below illustrates how users will log in the system. The use-case diagram below was created using a format suggested by Bennet, et al. (2010).

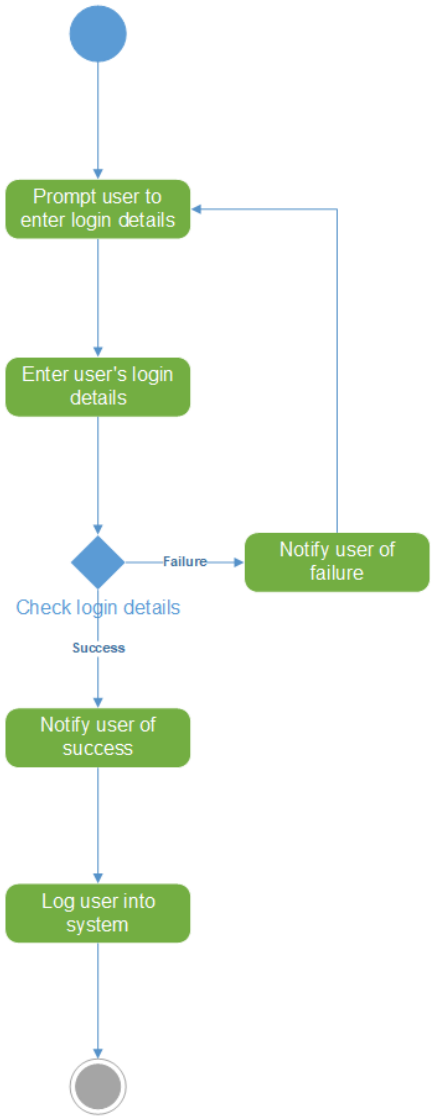

Figure 5 Login activity diagram

As seen above, four primary steps will be taken. Once the activities outlined above have been performed the user should be logged into the system.

Physiotherapist Marking Player as Injured Activity Diagram

The activity diagram below illustrates how physio will mark a player as injured using the system. The use-case diagram below was created using a format suggested by Bennet, et al. (2010).

Figure 6 Injured player activity diagram

As seen above, five primary steps will be taken. Once the activities outlined above have been performed a player should be marked as injured.

Registration Activity Diagram

The activity diagram below illustrates how users will be registered into the system. The use-case diagram below was created using a format suggested by Bennet, et al. (2010).

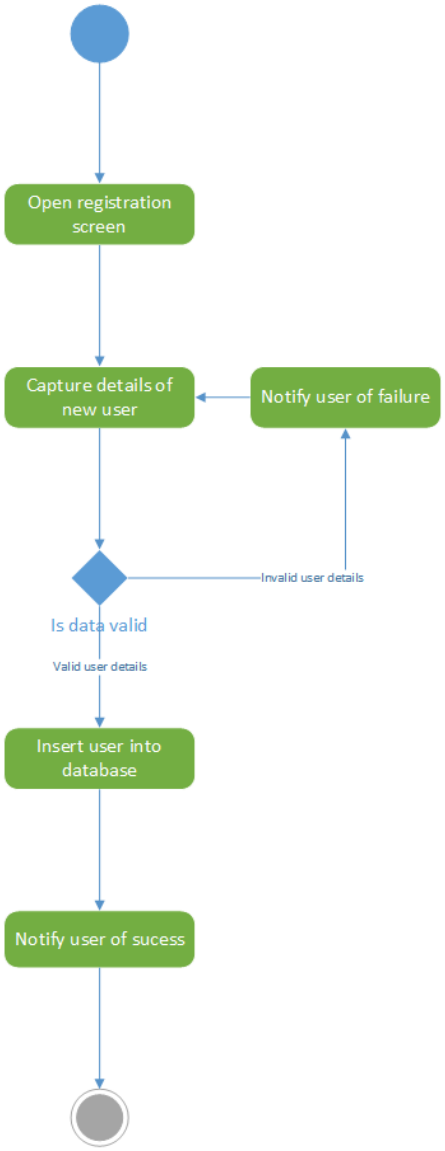

Figure 7 Registration activity diagram

As seen above, four main steps need to be taken. Once the activities illustrated above have been performed by a user, the user should be login into the system.

Sequence Diagrams

User Registration Sequence Diagram

The use-case diagram below was created using a format suggested by Bennet, et al. (2010).

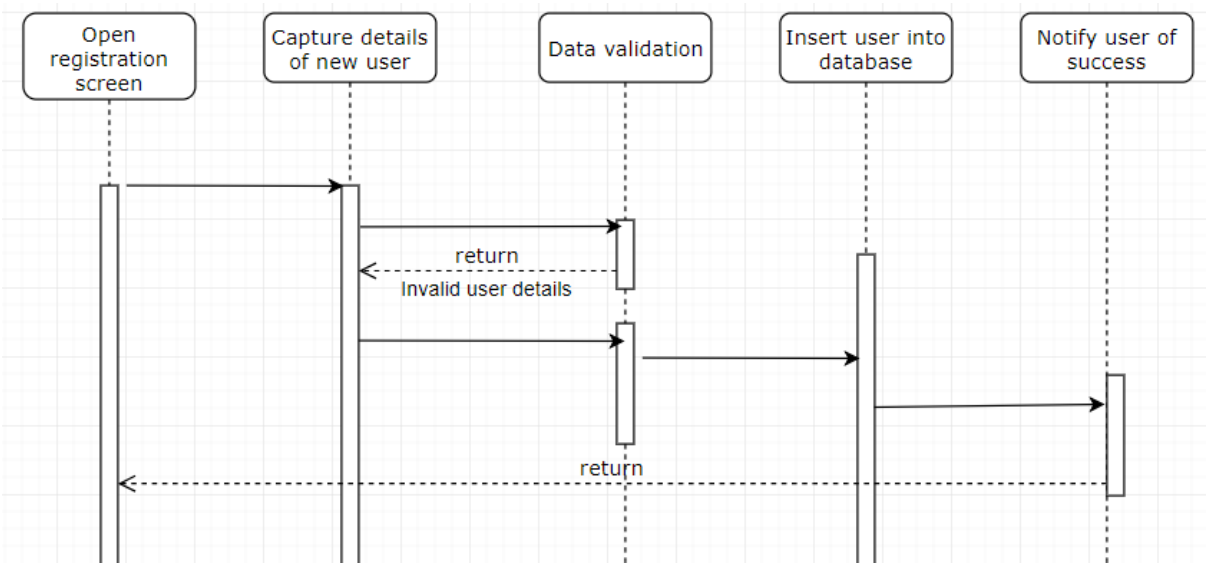

Figure 8 Registration sequence diagram

As seen above, five primary steps will be taken. Once the activities outlined above have been performed the user should be logged into the system.

User Login Sequence Diagram

The sequence diagram below illustrates how users will log in the system. The use-case diagram below was created using a format suggested by Bennet, et al. (2010).

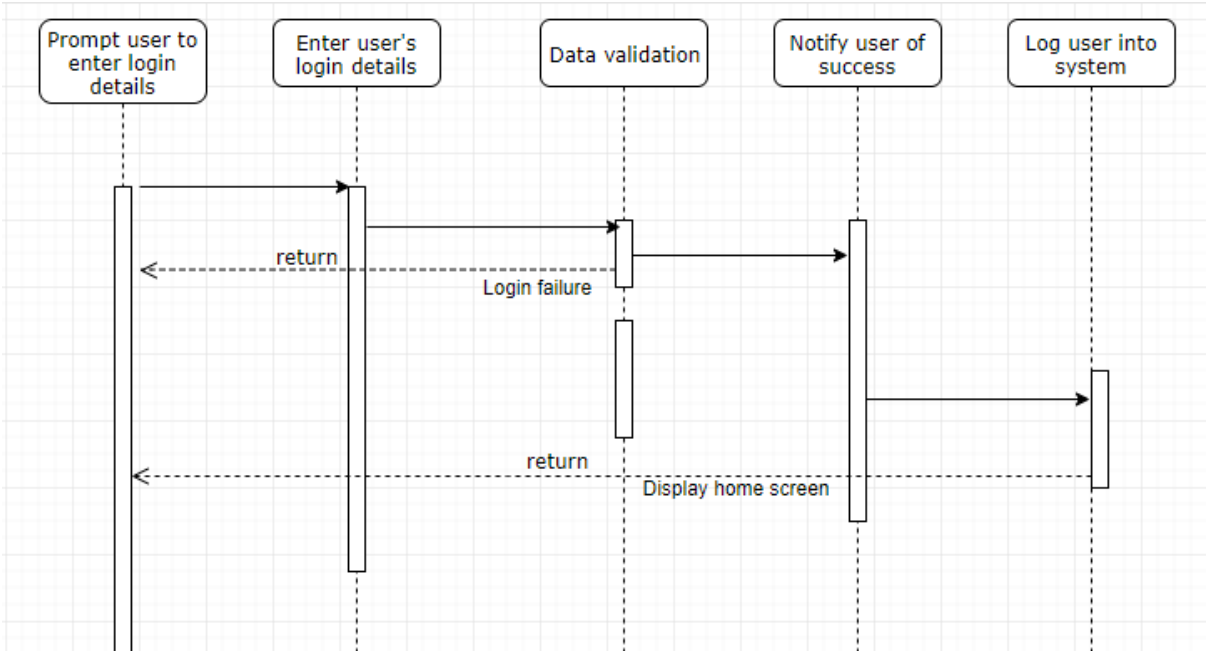

Figure 9 Login sequence diagram

As seen above, five primary steps will be taken. Once the activities outlined above have been performed the user should be logged into the system.

Physiotherapist Updating Player Injury Sequence Diagram

The sequence diagram below outlines how physio will mark a player as injured using the system. The use-case diagram below was created using a format suggested by Bennet, et al. (2010).

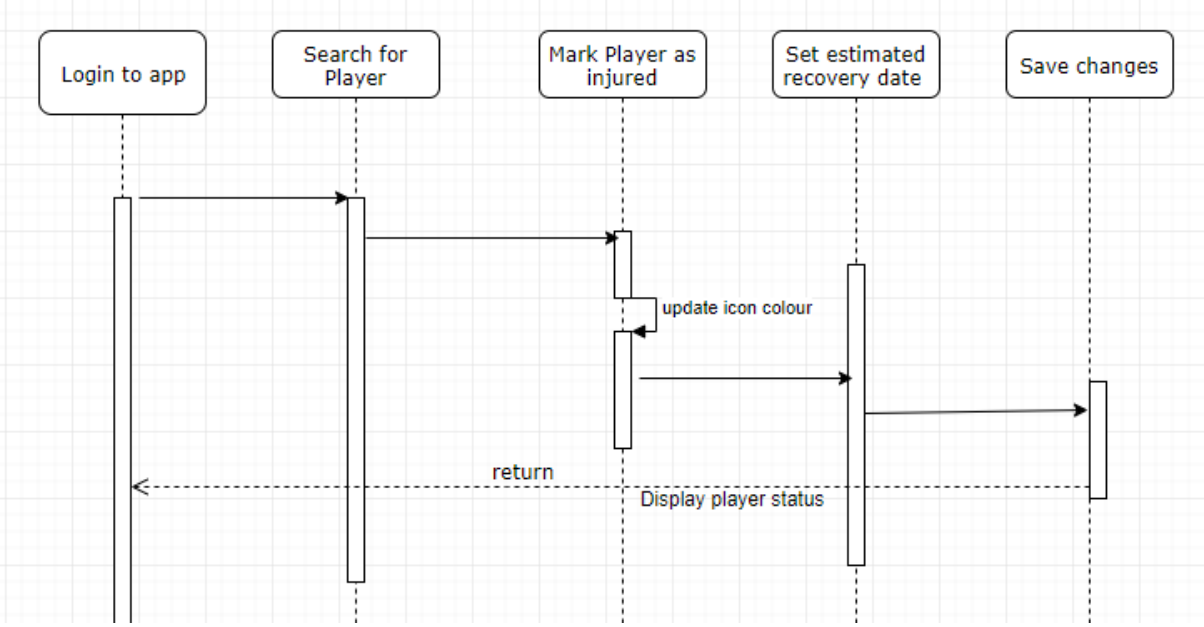

Figure 10 Injured player activity diagram

As seen above, five primary steps will be taken. Once the activities outlined above have been performed a player should be marked as injured.

1.2 Physical Design

1.2.1 Technologies

The RugBot application will be developed in different environments. The developers will use either Windows 10 or Linux for the development of the application. The operating system has no effect on the development as all the tools used for the development is available for Windows and Linux environments.

The application is developed using web technologies. The framework that the application is built on is Ionic and uses Cordova to deploy natively to any mobile operating system, or runs as a progressive web application, says Ionic (2018). Ionic is used to create hybrid mobile applications. According to Korf and Oksman (2016), a hybrid application is one that is built using web technologies that is wrapped in a thin native container. Ionic is built on Angular, which uses TypeScript as the scripting language. Ionic uses HTML5 and Sass for content and styling and uses Cordova plugins to use native APIs to run as a native web application on any operating system.

Since development is done with web technologies, any modern text editor with plugins can be used for development. The specific editor used by a developer will be by personal preference. Editors used include Visual Studio Code, Brackets, Atom, Sublime 3 or WebStorm which is an Integrated Development Environment (IDE) for creating web applications. All the above-mentioned text editors, not including WebStorm, allows for the installation of third-party plugins to assist in development. Plugins include functionality for code highlighting, debugging and version control.

The developers will make use of WhatsApp, Discord and email to communicate with each other during the development phase. GitHub is used as the version control system during development. The database that will be used is Firebase. According to Google (2018), Firebase is a cloud-hosted no-SQL (non-relational) database. This means that the application will need an internet connection to read or write data to or from the database. This allows for easy sync between different users of the same data without having to run a privately hosted web server. Firebase has real-time syncing across devices and is backed by Google (Google, 2018).

1.2.2 System Testing

Testing Types

There are several types of tests that need to be performed. Testing is vital in the development of any information system. Testing helps to ensure that a system operates as expected and without any bugs or crashes (Connolly & Begg, 2015). requirements. There are several types of testing including, unit testing, system testing, user acceptance testing, usability testing, regressing testing specification-based testing and white-box testing and black-box testing (Testing Excellence, 2018).

Connolly and Begg (2015) provided the following table to help highlight the differences between testing types.

Table 1 Testing types

Testing Type	What is tested	Purpose	Tester
User Acceptance Testing	The entire, complete, system (Connolly & Begg, 2015).	According to Connolly and Begg (2015), the purpose of user acceptance testing is to test the *real-world operating* of the final system. User acceptance testing is designed to ensure that the systems meet the user's requirements (Connolly & Begg, 2015).	Users of the system.
Volume Testing	Testing the performance of the system when strained (Connolly & Begg, 2015).	Design for testing the performance of the system under an intense workload (Connolly & Begg, 2015). Volume testing should be done throughout the development process.	Development team.
System Testing	The entire system.	System testing is done to ensure that the system operates as expected (Connolly & Begg, 2015). System testing should be done throughout the development process.	Development team.

Testing Type	What is tested	Purpose	Tester
Integration Testing	All the Individual units that make up the system. (Connolly & Begg, 2015)	Integration testing is performed to ensure that each induvial module that makes up the system operates as expected when used in conjunction with other modules (Connolly & Begg, 2015).	Development team.
Unit Testing	The Individual units that make up the system (Connolly & Begg, 2015).	The purpose of unit testing is to ensure that each individual module that makes up the system operates as expected (Connolly & Begg, 2015).	Development team.

As clearly shown in the above table, various forms of the test need to be performed throughout the development process. According to Sommerville (2001), testing is used to ensure that the final system meets the customers. If the client does not accept the final system, the entire project will be a waste.

As clearly shown, there are several types of testing that need to be performed to ensure the overall quality of the system. Testing is vital in the development of any information system. The testing templates that follow further outline the process that will be taken in testing the system.

Testing Template

All test plan has been created using the following testing template.

Page: **Test Date**:

Description:

Test for Type of User:

Tested By: **Type of Test**:

Signature:

Table 2 Testing template

Home Page **USER**			
Test ID	**Requirement**	**Successful**	**Comments**

Additional Comments:

Test Plan

Page: Sign in **Test Date**:

Description: This is the landing page of the RugBot mobile application where users will be required to sign into the application to gain access to the whole application. There are three types of users for the RugBot application and therefore will be distinguished by their Sign in credentials.

Test for Type of User: Coach, Physiotherapist and Player

Tested By: **Type of Test**: Functionality

Signature:

Table 3 Sign in testing plan

Sign In			
Test ID	**Requirement**	**Successful**	**Comments**
SN001	A user is able to access the Sign In page as the landing page.		
SN002	A user will receive a validated username and password.		
SN003	A user can insert the username in the username text field.		
SN004	A user can insert the password in the password text field.		
SN005	A user can click on the Sign In button to Sign in.		

Additional Comments:

Sign in Test continued:

Table 4 Data validation sign in test

Data Validation			
Test ID	**Requirement**	**Successful**	**Comments**
SN006	The text field Username is highlighted if an incorrect username is inserted.		
SN007	The text field Password is highlighted if an incorrect password is inserted.		
SN008	The Username and Password field will be required to be filled in before Signing In.		

Additional Comments:

Page: Home Page **Test Date**:

Description: This is the landing page after the Signing in the process has been successfully completed.

Test for Type of User: Coach

Tested By: **Type of Test**:
Functionality

Signature:

Table 5 Home page test plan

Home Page Coach			
Test ID	**Requirement**	**Successful**	**Comments**
HPC001	A user can click on the menu button.		
HPC002	A user can access the following pages from the menu: 1. Calendar; 2. Attendance; 3. Injured Players and 4. Gameday administration.		

Additional Comments:

Page: Calendar Page **Test Date**:

Description: This page will allow coaches to insert important dates onto the calendar

Test for Type of User: Coach

Tested By: **Type of Test**: Functionality

Signature:

Table 6 Calendar page test plan

Calendar Page Coach			
Test ID	**Requirement**	**Successful**	**Comments**
CPC001	A user can access the Calendar Page.		
CPC002	A user can view a calendar.		
CPC003	A user can select a date on the calendar.		
CPC004	A user can insert information on selected dates on the calendar.		
CPC005	A user can submit information which was inserted into the calendar		
CPC006	A user can modify information in the calendar.		

Additional Comments:

Page: Attendance Page **Test Date**:

Description: This page will allow coaches to take attendance.

Test for Type of User: Coach

Tested By: **Type of Test**: Functionality

Signature:

Table 7 Attendance page test plan

Attendance Page Coach			
Test ID	**Requirement**	**Successful**	**Comments**
APC001	A user can access the Attendance Page.		
APC002	A user can view a register of all the players.		
APC003	A user can click on "✓" icon if a player is present.		
APC004	A user can add a comment next to a player's name.		
APC005	A user can access the search bar.		
APC006	A user can search for a player by first or last name.		
APC007	A user can view the attendance history		
APC008	A user can click on the submit button to save information inserted.		

Additional Comments:

Page: Injury Page **Test Date**:

Description: This page will indicate to coaches the players who are injured. All information on this page is provided by the physiotherapists.

Test for Type of User: Coach

Tested By: **Type of Test**: Functionality

Signature:

Table 8 Injury page test plan

Injury Page Coach			
Test ID	**Requirement**	**Successful**	**Comments**
IPC001	A user can access the Injury Page.		
IPC002	A user can view all players who are injured.		
IPC003	A user can view comments about a player's injury.		
IPC004	A user can view when a player can resume training.		
IPC005	A user can view when a player can resume participating in matches.		

Additional Comments:

Page: Game Administration Page **Test Date**:

Description: This page will allow coaches to do necessary game day administration.

Test for Type of User: Coach

Tested By: **Type of Test**: Functionality

Signature:

Table 9 Game admin page test plan

Game Administration Page Coach			
Test ID	**Requirement**	**Successful**	**Comments**
GPC001	A user can access the game administration page.		
GPC002	A user can access an attendance form.		
GPC003	A user can allocate a jersey to a relative player.		
GPC004	A user can insert and allocate a player's position.		
GPC005	A user can save inserted information by clicking on the Save button		

Additional Comments:

Page: Home Page **Test Date**:

Description: This is the landing page after the Signing in process has been successfully completed.

Test for Type of User: Physiotherapist

Tested By: **Type of Test**: Functionality

Signature:

Table 10 Homepage test plan for physio

Home Page Physiotherapist			
Test ID	**Requirement**	**Successful**	**Comments**
HPP001	A user can click on the menu button.		
HPP002	A user can access the following pages from the menu: 5. Injuries		

Additional Comments:

Page: Injuries Page **Test Date**:

Description: This is the page where a user will be able to insert information about a player's injuries.

Test for Type of User: Physiotherapist

Tested By: **Type of Test**: Functionality

Signature:

Table 11 Injury page test plan for physio

Injuries Page Physiotherapist			
Test ID	**Requirement**	**Successful**	**Comments**
IPP001	A user can access the Injuries Page.		
IPP002	A user can access the player's register.		
IPP003	A user can insert the type of injury.		
IPP004	A user can insert the seriousness of an injury by colour-coding a player's name with either: • Red or • Orange.		
IPP005	A user can insert the colour-code green when a player has been cleared of all injuries.		
IPP006	A user can insert the date when a player can return to training.		
IPP007	A user can insert the date when a player can participate in games.		
IPP008	A user can insert any additional comments about the injury.		
IPP009	A user can save all inserted information by clicking on the Save button.		

Additional Comments:

Page: Home Page **Test Date**:

Description: This is the landing page after the Signing in the process has been successfully completed.

Test for Type of User: Players

Tested By: **Type of Test**: Functionality

Signature:

Table 12 Homepage test plan for players

Home Page Players			
Test ID	**Requirement**	**Successful**	**Comments**
HPL001	A user can click on the menu button.		
HPL002	A user can access the following pages from the menu: 6. Calendar and 7. Game Day.		

Additional Comments:

Page: Calendar Page **Test Date**:

Description: This page allows players to view important dates on the calendar which has been inserted by the coaches.

Test for Type of User: Players

Tested By: **Type of Test**: Functionality

Signature:

Table 13 Calendar page test plan for players

Calendar Page Players			
Players			
Test ID	**Requirement**	**Successful**	**Comments**
CPL001	A user can access the Calendar Page.		
CPL002	A user can view all important dates on the calendar.		

Additional Comments:

Page: Match Day Page **Test Date**:

Description: This page allows players to view important match information which has been inserted by the coaches.

Test for Type of User: Players

Tested By: **Type of Test**: Functionality

Signature:

Table 14 Test plan for match day page for players

Match Day Page Players			
Test ID	**Requirement**	**Successful**	**Comments**
MDP001	A user can access the Match Day Page.		
MDP002	A user can view all information on the page.		

Additional Comments:

Bibliography

Bennet, S., McRobb, S. & Farmer, R., 2010. *Object-Oriented Systems Analysis and Design Using UML.* 4th ed. London: McGraw Hill.

Connolly, T. M. & Begg, C. E., 2015. *Database Solutions: A step-by-step approach to building databases.* 6th ed. Harlow: Pearson Education Limited.

Google, 2018. *Firebase helps mobile app teams succeed..* [Online]
Available at: https://firebase.google.com/
[Accessed 31 July 2018].

Ionic, 2018. *The top open source framework for building amazing mobile apps..* [Online]
Available at: https://ionicframework.com/framework
[Accessed 31 July 2018].

Korf, M. & Oksman, E., 2016. *Native, HTML5, or Hybrid: Understanding Your Mobile Application Development Options.* [Online]
Available at:
https://developer.salesforce.com/page/Native,_HTML5,_or_Hybrid:_Understanding_Your_Mobile_Application_Development_Options
[Accessed 31 July 2018].

Sommerville, I., 2001. *Software Engineering.* 9th ed. Boston: Addison-Wesley.

Testing Excellence, 2018. *Types of Software Testing – Complete List.* [Online]
Available at: https://www.testingexcellence.com/types-of-software-testingcomplete-list/
[Accessed 20 07 2018].