

RugBot – Final Deliverable

ITSP300 - 2018



**W.P. RUGBY
ACADEMY**

RugBot Development Team

Group 2

Table of Contents

1	Project Proposal.....	1
1.1	Group & Customer Information	1
1.2	Project Introduction.....	2
1.2.1	Background, Purpose & Scope	2
1.2.2	Aim & Objectives	3
1.3	High-level Requirements.....	4
1.3.1	Functional Requirements	4
1.3.2	Non-functional Requirements.....	5
1.3.3	Technical Requirements	6
1.4	Schedule.....	7
1.5	Risk Management.....	9
2	Methodology, Development Plan & Schedule	23
2.1	Methodology	23
2.1.1	Agile Approach	23
2.1.2	Incremental Approach.....	24
2.1.3	Iterative Approach.....	24
2.2	Development Plan.....	25
2.3	Schedule.....	26
2.4	User Requirements Document.....	29
2.4.1	Introduction	29
2.4.2	System Perspective	33
2.4.3	Functional Requirements	33
2.4.4	Non-functional Requirements.....	36
2.4.5	Technical Requirements	38
2.4.6	User Characteristics.....	39
2.4.7	Operational Environment	39
3	Information System Design	40
3.1	Logical Design	40

3.1.1	Use Case Diagrams.....	40
3.1.2	Class Diagram	43
3.1.3	Activity Diagrams	44
3.1.4	Sequence Diagrams	47
3.2	Physical Design	50
3.2.1	Technologies	50
3.2.2	System Testing	51
4	Testing Results	66
4.1	System Interfaces	66
4.1.1	First Iteration Prototypes.....	66
4.1.2	Second Iteration Prototypes.....	70
4.2	Test Results.....	77
5	User Manual	89
5.1	Introduction	89
5.2	Getting Started.....	89
5.3	How to Install	89
5.4	Instructions	90
5.4.1	Register	90
5.4.2	Login	91
5.5	Users	92
5.5.1	Coach	92
5.5.2	Physiotherapist	110
5.5.3	Coach & Physiotherapist.....	115
5.5.4	All Users	118
6	Evaluation	129
6.1	The final system & Customer Requirements	129
6.2	List of Functional Requirements.....	130
6.3	Iterative & Incremental Development in Agile.....	132
6.4	Customer Evaluation.....	133

6.5	Group Dynamics and Team Collaboration	134
6.6	Time Management.....	134
6.7	Lessons Learnt	135
6.7.1	Stefanus Buys.....	135
6.7.2	Tyla Grey	135
6.7.3	Abongile Mdeleleni	136
6.7.4	Matthew Van der Bijl.....	136
7	Bibliography	137

Table of Figures

Figure 1 Gantt Chart showing Deliverable 1	8
Figure 2 Gannt Chart part 1 of 3	26
Figure 3 Gannt Chart part 2 of 3	27
Figure 4 Gannt Chart part 3 of 3	28
Figure 5 Use-case Diagram for RugBot	35
Figure 6 Coach User-case diagram.....	40
Figure 7 Physiotherapist Use-case diagram.....	41
Figure 8 Player Use-case diagram.....	42
Figure 9 Class Diagram of RugBot System	43
Figure 10 Login activity diagram	44
Figure 11 Injured player activity diagram.....	45
Figure 12 Registration activity diagram	46
Figure 13 Registration sequence diagram.....	47
Figure 14 Login sequence diagram	48
Figure 15 Injured player activity diagram.....	49
Figure 16 Prototype welcome page.....	66
Figure 17 Prototype registration page	67
Figure 18 Prototype login page	68
Figure 19 Prototype weekly planner	69
Figure 20 Login screen	70
Figure 21 Login screen validate	71
Figure 22 Login screen modal.....	72
Figure 23 Registration screen	73
Figure 24 Coach list	74
Figure 25 Physio list.....	75
Figure 26 Physio form	76
Figure 27 Register	90
Figure 28 Login	91
Figure 29 Register Link	91
Figure 30 Coach Home Page.....	92
Figure 31 Menu Icon	93
Figure 32 Menu bar.....	93
Figure 33 Coach Players Page	94
Figure 34: Date	95
Figure 35 Date picker.....	95

Figure 36 Thumbs up icon	96
Figure 37 Clicked upon Thumbs Up icon.....	96
Figure 38 Level of Injury.....	97
Figure 39 Injury Information	97
Figure 40 Close Button	98
Figure 41 Back Arrow.....	98
Figure 42 Attendance History Page.....	99
Figure 43 Attendance History Button.....	99
Figure 44: Player attendance history.....	100
Figure 45 Back Arrow.....	100
Figure 46 Calendar Page	101
Figure 47 Real-time date.....	101
Figure 48 Date required	102
Figure 49 Add Icon.....	102
Figure 50 Add Event Page	103
Figure 51 Add Title.....	103
Figure 52 Start Date Picker	103
Figure 53 Date Picker	104
Figure 54 End Date Picker	104
Figure 55 Date Picker	105
Figure 56 All Day Event	105
Figure 57 Add Event button.....	105
Figure 58 Cancel icon	105
Figure 59 Event saved into date.....	106
Figure 60 View Events on Date.....	106
Figure 61 Back Arrow icon	107
Figure 62 Match day page link	107
Figure 63 Match Day Page.....	108
Figure 64 Jersey Number button.....	108
Figure 65 Back arrow icon	109
Figure 66 Home Page Physio	110
Figure 67 Menu Icon	110
Figure 68 Menu bar.....	111
Figure 69 Player list	112
Figure 70: Level of Injury button.....	113
Figure 71 Injury edit details modal.....	113
Figure 72: Edit type of injury.....	113

Figure 73 Recovery date picker	114
Figure 74 Quick note.....	114
Figure 75 Users button.....	115
Figure 76 User List page.....	115
Figure 77: Coaches tab.....	116
Figure 78: Physios tab	116
Figure 79 Payer Details button.....	116
Figure 80 Edit Players Details page	117
Figure 81 Back arrow icon	117
Figure 82 About button.....	118
Figure 83 Home Page Player	118
Figure 84 Menu icon	119
Figure 85 About page.....	120
Figure 86 Back arrow icon	120
Figure 87 Logout button	121
Figure 88 Exit button	121
Figure 89 Calendar	122
Figure 90 Match Day event	122
Figure 91 Match Day Page.....	123
Figure 92 Access Denied	124
Figure 93 User List.....	125
Figure 94 User Details page.....	126
Figure 95 First Name text field	127
Figure 96 Surname text field	127
Figure 97 Email text field.....	127
Figure 98 Edit Details button.....	127
Figure 99 Contact button.....	127
Figure 100 Delete button.....	128
Figure 101 Back arrow icon.....	128

Table of Tables

Table 1 Functional Requirements.....	4
Table 2 Non-functional Requirements	5
Table 3 Technical Requirements.....	6
Table 4 Risk register for RugBot	11
Table 5 Development plan for RugBot	25
Table 6 Table of Users	29
Table 7 Definitions	32
Table 8 Acronyms and abbreviations	32
Table 9 Functional Requirements.....	33
Table 10 Non-functional Requirements	36
Table 11 Technical requirements	38
Table 12 User characteristics.....	39
Table 13 Testing types.....	51
Table 14 Testing template.....	53
Table 15 Sign in testing plan.....	54
Table 16 Data validation sign in test.....	55
Table 17 Home page test plan	56
Table 18 Calendar page test plan	57
Table 19 Attendance page test plan.....	58
Table 20 Injury page test plan	59
Table 21 Game admin page test plan	60
Table 22 Homepage test plan for physio.....	61
Table 23 Injury page test plan for physio.....	62
Table 24 Homepage test plan for players.....	63
Table 25 Calendar page test plan for players.....	64
Table 26 Test plan for match day page for players.....	65
Table 27 Sign in testing plan.....	77
Table 28 Data validation sign in test.....	78
Table 29 Home page test plan	79
Table 30 Calendar page test plan	80
Table 31 Attendance page test plan.....	81
Table 32 Injury page test plan.....	82
Table 33 Game admin page test plan	83
Table 34 Homepage test plan for physio.....	84
Table 35 Injury page test plan for physio.....	85

Table 36 Homepage test plan for players.....	86
Table 37 Calendar page test plan for players.....	87
Table 38 Test plan for match day page for players.....	88
Table 39 Functional Requirements.....	130

1 Project Proposal

1.1 Group & Customer Information

The table below outlines, the group number, name and the members that make it up.

Group number:	2
Group name:	RugBot Development Team
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
------------------	---

1.2 Project Introduction

We, the RugBot development team, have been tasked with the creation of a new and innovative system for Western Province Rugby Academy. This project will be delivered in eight deliverables. The document that follows outlines the Western Province Rugby Academy and the proposed system.

1.2.1 Background, Purpose & Scope

The sole purpose of the Western Province Rugby Academy is to facilitate the growth of young athlete so that they may reach their full potential. The Western Province Rugby Academy provides their athletes with an athlete development and high-performance rugby program to ensure that each individual athlete has the resources to complete successfully.

The rugby academy makes use of a functional strength and conditioning program with the purpose of transforming the young rugby boys into professional rugby men. This program which is implemented in their training is supported by a diet and an expert lifestyle management process.

The role of the Academy is to work on all athlete's skills, fitness progression, discipline on the field, coping with pressure and decision making. This is done to ensure that each individual athlete understands their role in the team as well as to develop them as well-rounded individuals.

The rugby academy works with Headstrong Consulting who provides their athletes with a sports psychology program which recognises the importance of providing structure, education and professional guidance.

The purpose of this project is to develop a new management application for the Western Province Rugby Academy. The system will include the creation of an application which will support mobile devices for management purposes. The application will allow the rugby coaches full control and easy communication over the day-to-day activities, training sessions, record keeping and management capabilities. The application system will include a section for the rugby players to view their weekly timetable, match fixtures and the teams which will be playing in the matches.

It is clear that the rugby academy would benefit from the development of the new management system. The new system would allow the coaches and management to apply full attention the athletes rather than struggling with an outdated paper-based system.

1.2.2 Aim & Objectives

The aim of this project is to develop and implement a Management System to support record keeping, access control, communication and management control. The primary objectives of the new system are:

1. Allow the user to manage their day-to-day administration;
2. Meet all user requirements; and
3. Allow all users to create, read, update and delete records where appropriate.

It is vital that the users find the final system easy to use, useful and provides with a holistic experience.

RugBot has the potential to change how the Western Province Rugby Academy functions on a fundamental level.

1.3 High-level Requirements

According to Sommerville (2001), high-level requirements specify what the system must do, but does not provide detailed explanations on how implementation should be done. System requirements should seek to describe the behaviour a system in the simplest manor possible as well as outline the system's operational restraints (Sommerville, 2001).

1.3.1 Functional Requirements

According to Sommerville (2001), functional requirements state the services that the system must provide and how the system should react to specific inputs and situations.

Table 1 Functional Requirements

Identifier	Requirement Description
FR01	Users must use a one-time login to log in to the application for authorization purposes.
FR02	Coaches must be able to take an attendance list of students at practice.
FR03	Coaches must be able to view a backlog of student's attendance for past dates.
FR04	Coaches must be able to view a list of all their students and their availability for practise sessions and matches.
FR05	Coaches and students must have a calendar with a practise match dates and times.
FR06	The physiotherapist must be able to mark a student as injured and not able to practise or play matches.
FR07	The physiotherapist must be able to add an estimated date of when a student will be able to practise again.
FR08	If a student missed more than three practise sessions, the coach must receive a notification of the student's absence.
FR09	The coach must be able to see the total of boys at practice.
FR10	The coach must be able to assign jersey numbers to players on match dates.

It is key that users are able the create, read, update and delete all data that they insert into the program.

1.3.2 Non-functional Requirements

According to Sommerville (2001), non-functional requirements do not directly describe what the system must do. Non-functional requirements define the properties that a system must have, for instance, performance, security etc.

Table 2 Non-functional Requirements

Identifier	Requirement Description
Performance	
NFR01	Database response times must be very quick.
NR02	Quick response times in applications.
NFR03	GUI must be quick and responsive.
Design	
NR04	The GUI design must be minimalist and simple.
NFR05	Navigation of the application must be sensible and straight-forward.
Security	
NFR06	The database must only be accessible by authenticated users.
NR07	A user must be able to access only data specific to their authorization level.
Reliability	
NR08	The application should never crash and be bug-free.
NR09	The application must be able to operate even when connected to the database is lost.
NR10	The database must be able to have multiple users access it at the same time.
Scalability	
NR11	The system must be able to grow in terms of active users.

It is vital that the project conforms to the modern design and usability principles. Ultimately, the users need to be provided with the best experience possible.

1.3.3 Technical Requirements

The technical requirements specify what technologies will be used during the development of the system. This will include software that is used in development; languages and frameworks that will be used; and what platform the system will be developed for and tested on.

Table 3 Technical Requirements

Identifier	Requirement Description
TR01	Users will need a mobile device running either Android or iOS to use the application.
TR02	Users will need an internet connection to connect to the database.
TR03	Firebase will be used for the database needs.
TR04	The application will be developed on the Ionic framework.
TR05	HTML, Sass and TypeScript (a superset of JavaScript) are the languages that will be used for development.
TR06	Developers will need Android and iOS devices for testing.
TR07	WebStorm or any modern code editor with TypeScript support will be used for writing and editing code.

The project's technical requirements should adapt to new technology and market change. It is vital that the client is presented with a truly modern system.

1.4 Schedule

The Gantt charts below shows an outline for the first deliverable, their task and dependencies as well as task duration and allocated resources. Both charts were made using Microsoft Project and follow precedents set by Schwalbe (2012).

The project the project initiation date is the 5th of February 2018. The deliverables due dates are as indicated below:

1. Deliverable 1: 23/02/2018;
2. Deliverable 2: 11/04/2018;
3. Deliverable 3: 10/07/2018;
4. Deliverable 4: 07/09/2018;
5. Deliverable 5 (User manual): 12/10/018;
6. Deliverable 5 (Evaluation Report): 19/10/2018;
7. Deliverable 6: 02/11/2018; and
8. Demonstration: 09/11/2018.

Each deliverable needs to be completed, reviewed and submitted before the submission dates. It is key that all components are completed on time. The Gantt chart should be updated for each submission.

The Gantt chart below presents the proposed schedule for the first deliverable of this project.

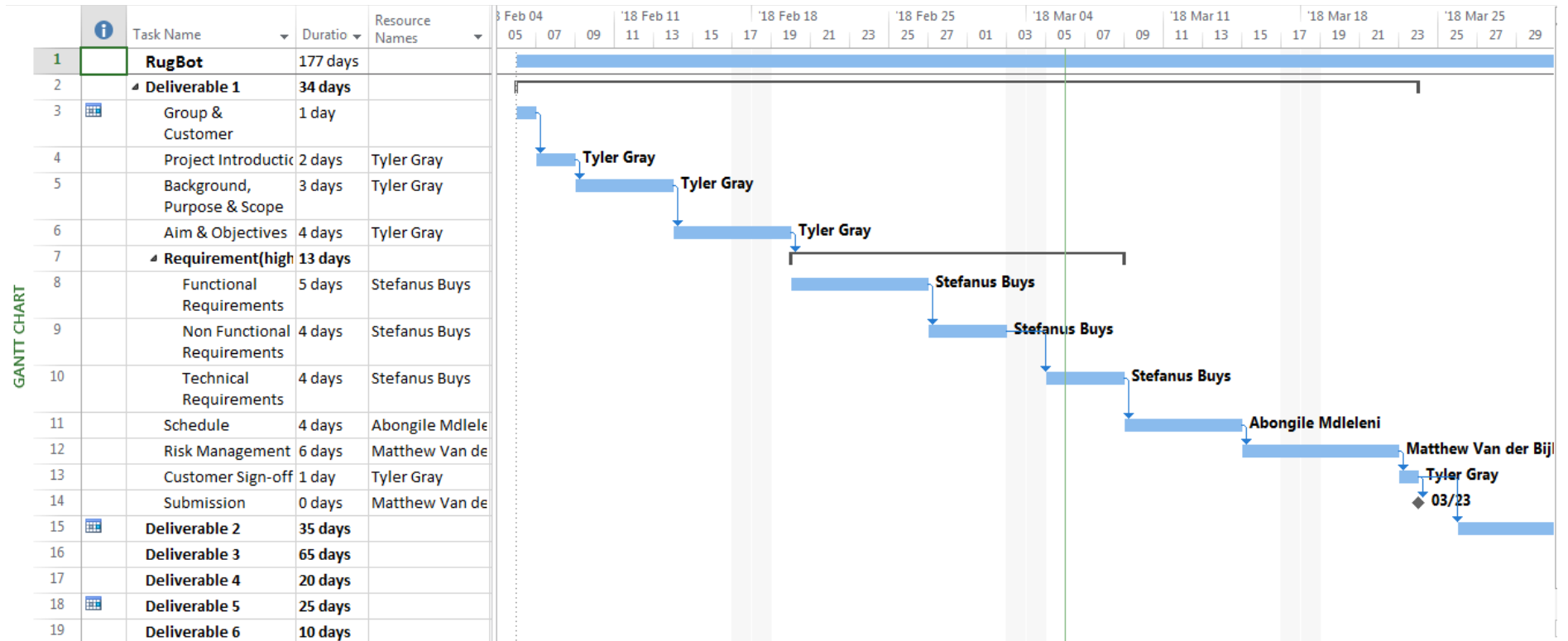


Figure 1 Gantt Chart showing Deliverable 1

As seen above, the project has been broken down into several logical chunks each with time allocated to them.

1.5 Risk Management

Risk concerns future happenings (Pressman & Maxim, 2015). Schwalbe (2012) defined a risk as the possibility for a future loss or injury to occur. The effective management of risks is paramount to the success of a project as highlighted by the appearance of risk management in the nine knowledge areas of project management proposed by Project Management Institute (2013). According to Buttrick (2009), actively monitoring risks is vital to good project management.

Project risk management is the process of identifying, analysing and accounting for risks during a project's lifecycle to ultimately ensure that the goals of the project are met (Schwalbe, 2012). Schwalbe (2012) states that the true importance of project risk management is often misunderstood. It is vital that project managers understand the nine knowledge areas to increase project success (Schwalbe, 2012). Unresolved risks may lead to late system delivery, budget depletion and other project problems (Sommerville, 2001).

However, it is important to note that Project Management Institute (2013) states that if a risk occurs it may have a positive impact on a given project. Botha and Musengi (2012) note that the ability to calculate and take risks may lead to greater financial success. As stated by Drucker (1975), though it is an act of futility to try and eliminate all risks it is essential that efforts are made to help mitigate them.

Risk identification is the ongoing process of spotting and documenting potential risks to a project (Project Management Institute, 2013; Pressman & Maxim, 2015). According to Schwalbe (2012) risks can be placed in one of five categories, namely:

1. People;
2. Technological;
3. Market;
4. Financial; and
5. Structural.

Buttrick (2009) states that risks need to be identified and evaluated in a consistent manner throughout the project's lifecycle. Pressman and Maxim (2015) suggest the construction of a risk management plan. All members of a project's team, including stakeholders, need to actively participate in risk management (Pressman & Maxim, 2015).

Once a potential risk has been identified a response needs to be formulated. Project Management Institute (2013) defined risk response as the process of developing actions and

options to combat a risk. Schwalbe (2012) outlined four basic responses to negative risks, namely:

1. Acceptance;
2. Mitigation;
3. Avoidance; and
4. Transference.

Risk management is needed from the onset of any project (Buttrick, 2009). It is important to focus on the risks with the greatest probability of occurring and those with the greatest impact (Buttrick, 2009).

With effective risk management, any project can be completed successfully. The ongoing identification of risks ensures that the entire project team is aware of the project's status. The effective management of risks is project management. Ultimately, risk management ensures that the projects are delivered.

Risk management refers to the identification, analysis, and prevent potential issues that may occur with a project.

Table 4 Risk register for RugBot

No.	Rank	Risk	Description	Category	Root Cause	Triggers	Potential Responses	Risk owner	Probability	Impact	Status
R13	1	Poor project control	Poor project control may lead to late project delivery or failure to deliver any system.	People	Poor team management.	Failing to monitor the overall status of the project.	Meet with the client. Consult objectives, requirements and scope.	RugBot Team	High	High	RugBot Team members will continue to monitor and control the project.
R14	2	Impossible targets	The deadlines of deliverables and milestones are unattainable.	People	Poorly established requirements.	Failing to understand what the client requires.	Reevaluate project targets.	RugBot Team	High	High	RugBot Team will need to evaluate the project and set reasonable goals.

No.	Rank	Risk	Description	Category	Root Cause	Triggers	Potential Responses	Risk owner	Probability	Impact	Status
R0D	3	Scope creep	The client requests additional features, to increase the scope after the scope has been established and development has begun.	People	Poorly established requirements.	Failing to manage the project and understand what the client requires.	Meet with the client. Consult objectives, requirements and scope.	RugBot Team	High	Medium	RugBot Team will need to consult the requirements of the project.
R0E	4	Unreliable operation	The system does not operate as intended due to logical errors and bugs in the system.	Technological	Poorly constructed system.	Failing to effectively test the project.	Perform thought project testing and rectify any errors that occur.	RugBot Team	High	Medium	The projects need to be thoroughly tested and bugs corrected.

No.	Rank	Risk	Description	Category	Root Cause	Triggers	Potential Responses	Risk owner	Probability	Impact	Status
R11	5	Operational issues	The system does not operate as intended and thus does not meet the client's business needs.	Technological	Poorly constructed system.	Falling to effectively test the project.	Meet with the client. Consult objectives, requirements and scope.	RugBot Team	High	Medium	Team members will communicate with the client and make corrects where necessary.
R12	6	Poor response time		People	Poor team coordination .	Failing to operate together as a team.	Perform thought project testing and rectify any errors that occur as they occur.	RugBot Team	High	Low	Team members are meeting regularly.

No.	Rank	Risk	Description	Category	Root Cause	Triggers	Potential Responses	Risk owner	Probability	Impact	Status
R03	7	Overall quality of the project is not up to standard	The quality of the final product to the client is not acceptable resulting the client refusing to use the product.	People	Poorly constructed system.	Falling to understand and effective test the project.	Perform thorough usability testing and make corrections where needed.	RugBot Team	Medium	High	Team members are working hard to ensure that their development skills are second to none.
R05	8	No clear vision of final project	Due to poor requirement analysis, poor scope establishment or the complete misunderstanding the client's needs to final product delivered does not meet the client's needs.	People	Poorly established requirements.	Failing to establish project requirements and understand what the client requires.	Meet with the client. Consult objectives, requirements and scope.	RugBot Team	Medium	High	Team members are meeting regularly. The client is being consulted on a regular basis.

No.	Rank	Risk	Description	Category	Root Cause	Triggers	Potential Responses	Risk owner	Probability	Impact	Status
R08	9	Risks are ignored	Ignoring critical risk will lead to the failure of the project.	People	Poor team coordination.	Failing to manage the project and project team.	Review the project and team structure. Make rectifications where needed.	RugBot Team	Medium	High	The status of the project is being continuously evaluated and any risks identified are being resolved.
R09	10	The client does not accept the final product	If the final product does not meet the client's needs the client won't accept it. Ultimately, the project will fail.	People	Poorly established requirements and poorly constructed system.	Failing to manage the project and understand what the client requires.	Meet with the client. Consult objectives, requirements and scope.	RugBot Team	Medium	High	The client is being consulted on a regular basis.

No.	Rank	Risk	Description	Category	Root Cause	Triggers	Potential Responses	Risk owner	Probability	Impact	Status
R0A	11	Poor interface design	Poorly design interfaces will have a detrimental effect on the use of the product.	Technological	Poorly constructed system.	Failure to effectively develops and testing the system.	Meet with client and evaluate prototypes. Refer to design principles proposed by Preece, <i>et. al.</i> (2015)	RugBo t Team	Medium	High	The project is being continuously evaluated
R0C	12	Unfeasibly implementation	The product required by the client cannot be created because of insurmountable technical challenges.	Technological	Poorly established requirements.	Failing to manage the project and understand what the client requires.	Reevaluate the project and make necessary corrections. The client may need to be involved in the process.	RugBo t Team	Medium	High	The project is being continuously evaluated

No.	Rank	Risk	Description	Category	Root Cause	Triggers	Potential Responses	Risk owner	Probability	Impact	Status
R10	13	Users change their mind	The client decides that the system is no longer needed. This may be caused by changes in the external environment.	People	Poor client management.	Failure to understand what the client requires.	Meet with the client. Consult objectives, requirements and scope.	RugBo t Team	Medium	High	The client is being consulted on a regular basis.

No.	Rank	Risk	Description	Category	Root Cause	Triggers	Potential Responses	Risk owner	Probability	Impact	Status
R04	14	Unresolved personal disagreements	Intergroup conflict which affects the productivity of the group. This may lead to the late delivery of the project, have a detrimental impact on the quality of the project and ultimately lead to the failure of the project.	People	Poor team coordination	Failing to manage the project and project team.	Elect a member of the group to mediate conflict during a group meeting. Go for some coffee.	RugBo t Team	Medium	Medium	Team members are meeting regularly.

No.	Rank	Risk	Description	Category	Root Cause	Triggers	Potential Responses	Risk owner	Probability	Impact	Status
R06	15	Poor team coordination	Poor team communication and coordination will have a detrimental effect the productivity of the team.	People	Poor team coordination	Failing to manage the project and project team.	Meet as a team and work.	RugBo t Team	Medium	Medium	Team members are meeting regularly.
R07	16	Poor integration management	Poor integration management of final product may lead to the client rejecting it.	Operation	Poor client management.	Failure to understand what the client requires.	Consult the client and make correction where needed.	RugBo t Team	Medium	Medium	Team members are meeting regularly.

No.	Rank	Risk	Description	Category	Root Cause	Triggers	Potential Responses	Risk owner	Probability	Impact	Status
R00x	17	Incorrect requirements analyzed	The user's requirements are poorly analyzed or misunderstood. This may result in the wrong problem being addressed	People	Poorly established requirements.	Failure to understand what the client requires.	Review project objectives and reanalyze project objectives. The scope of the project may need to be adjusted.	RugBot Team	Low	High	Team members are meeting regularly. The client is being consulted on a regular basis.
R0B	18	Failure to deliver the system	Due to unforeseen consequences or the inevitable failure of the project, the project gets terminated before the delivery of the final product.	People	Poor team coordination	Failure to understand what the client requires as well as failing to manage the project and project team.	Consult the client and make correction where needed to ensure that the final system is delivered.	RugBot Team	Low	High	Team members are meeting regularly. The client is being consulted on a regular basis.

No.	Rank	Risk	Description	Category	Root Cause	Triggers	Potential Responses	Risk owner	Probability	Impact	Status
R00	19	Poorly defined scope	The scope of the project is poorly established resulting in the wrong problem being addressed.	People	Poor team coordination and client management.	Failing to manage the project and project team.	Review project objectives, reanalyze requirements and amend project scope.	RugBot Team	Low	Medium	Team members are meeting regularly. The client is being consulted on a regular basis.
R01	20	Poorly defined scheduled	The deadlines for key deliverables and project milestones are poorly established. This may result in late deliverable delivery.	Schedule	Poorly established requirements.	Failing to manage the project and project team.	Review project objectives and scope. Reconstruct schedule accordingly.	RugBot Team	Low	Medium	Team members are meeting regularly. The client is being consulted on a regular basis.

No.	Rank	Risk	Description	Category	Root Cause	Triggers	Potential Responses	Risk owner	Probability	Impact	Status
R02	21	Poorly estimated budget	The overall budget for the project is poorly established or misunderstood. This may lead to the misappropriation of funds.	Financial	Poorly established requirements.	Failing to manage the project and project team.	Review project objectives, requirements and scope. Reconstruct budget accordingly.	RugBot Team	Low	Low	Team members are meeting regularly. The client is being consulted on a regular basis.
R0F	22	Poor maintenance of documentation	Poor documentation is developed and maintained for the duration of the project's lifecycle.	People	Poor team coordination.	Failing to manage the project and project team.	Meet as a team and get the work done.	RugBot Team	Low	Low	Team members are meeting regularly to work on the documentation.

As seen in the table above, though there are many risks the RugBot the impact of all risks can be mitigated through effective risk management.

2 Methodology, Development Plan & Schedule

2.1 Methodology

Each element of the project is designed and constructed separately and delivered to the client for an overview. This technique allows the client to view individual components and makes room for feedback if any user requirements have changed. This is a benefit as it decreases the time of development, where an issue can be resolved quickly and efficiently.

2.1.1 Agile Approach

An Agile methodology is an approach to software development which supports all unpredictable mishaps during the development process. Unlike the waterfall methodology, the agile methodology makes use of an incremental approach. Ultimately, this means that the model is designed, implemented and thereafter tested as each element is added to the project. Ultimately, it involves both processes of development and maintenance. The iterative approach is focused on conducting development in cycles rather than a sequential step-by-step process. It allows developers to create a working prototype which is compiled against user requirements and offers users the ability to add features while the cycle of development is occurring (Ghahrai, 2018).

2.1.1.1 Advantages of Agile Methodology:

Using an agile methodology will have the following advantages:

1. Supports the change of user requirements to be inserted into the project during development;
2. Easy to manage and manipulate features in the system;
3. After each deliverable is completed, agile allows the client to have an overview of the system and therefore make changes if necessary;
4. Testing is done after each deliverable and therefore makes it easy to identify bugs in a specific section of the system; and;
5. Agile ensures that the schedule of the project is kept and therefore is more reliable to meet the due date.

As clearly shown, there are clear advantages in using an agile methodology.

2.1.2 Incremental Approach

The development phases of this approach are done in a linear format. A developer can only start the next phase of developing the project once the initial phase has been completed. This is since each phase is dependent on the next phase. The figure below is a diagrammatic representation of the incremental model as defined by Ghahrai (2017).

2.1.2.1 *Advantages of Incremental*

The usage of an incremental model has many advantages.

1. Allows working software to be developed early in the life cycle;
2. Scope and requirements changes will not have a huge cost impact;
3. Risk identification occurs early in development; and;
4. Testing and debugging happen during each iteration, therefore, saving time to debug and test the entire project once it is completed (Ghahrai, 2018).

As clearly show that is various advantages to an incremental approach.

2.1.3 Iterative Approach

The iterative approach does not contain any set number of phases and development process which a developer must follow to achieve a successful project. Instead, the approach deals with individual cycles which do not necessarily depend on information or development done in the previous cycle. All development and information are individually processed.

2.1.3.1 *Advantages of an iterative approach*

There numerous advantages associated with an iterative approach

1. Changes in user requirements can be facilitated;
2. Risks are identified during each iteration;
3. More time is spent on designing and developing rather than documentation; and;
4. Prototypes are created during each iteration (Ghahrai, 2018).

By initially creating prototypes for the project, the client can thereafter manage and manipulate the features on the system. Ultimately creating more development time to ensure the completion of a successful project. the

In conclusion, this project adopts a hybrid version of agile methodology which incorporates aspects of both incremental and iterative approaches.

2.2 Development Plan

The Development Plan shows the increments during which certain requirements will be implemented. The increments each have a timeframe during which it must be completed, and the development plan shows during which deliverable it will be completed. In this development, plan sprints have been called increments and are generally each 3-weeks long.

Table 5 Development plan for RugBot

D	I	Purpose	Requirements	Timeframe
	1	Create database and user profile data.		5 days
	2	Allow coaches to add or remove players; functionality for coaches to do roll call; functionality for the physiotherapist to mark players as unavailable for practise or matches and when the player will be back in action.	FR02, FR03, FR06, FR07, FR21	29 days
	3	Allows coaches to view player availability for practice and matches; allow all users to update or delete personal information.	FR04, FR18, FR19, FR20	21 days
	4	Supplies players and coaches with a calendar with match dates and times; functionality for coaches to assign jersey numbers to players for matches; coaches can schedule matches; coaches and players can view match teams and match times.	FR05, FR10, FR13, FR14	26 days
	5	Login screen; authorization and access rights; functionality for a physiotherapist to specify player injuries; coaches can view injured player details.	FR01, FR11, FR12, FR22	21 days
	6	Player notifications about practice, lunch and match information; coach notifications about upcoming matches, player injuries and player absence; coaches can view total boys at a practice session	FR08, FR09, FR15, FR16, FR17	24 days

Key:

D – Deliverable

I - Increment

2.3 Schedule

The Gantt chart below was constructed by following examples presented by Buttrick (2009) and Schwalbe (2012). Deliverable 2 is due on the 11th of May 2018.

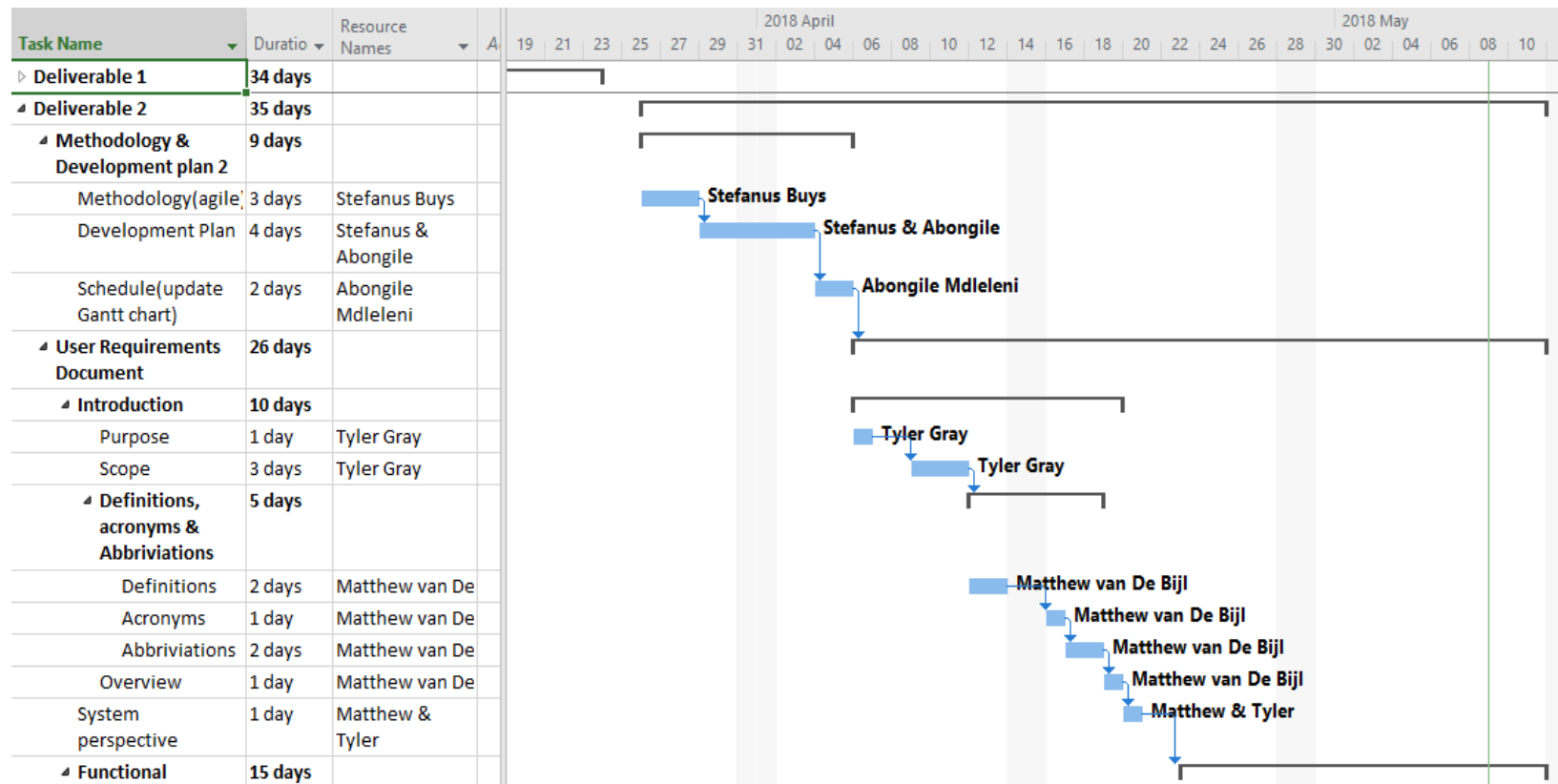


Figure 2 Gantt Chart part 1 of 3

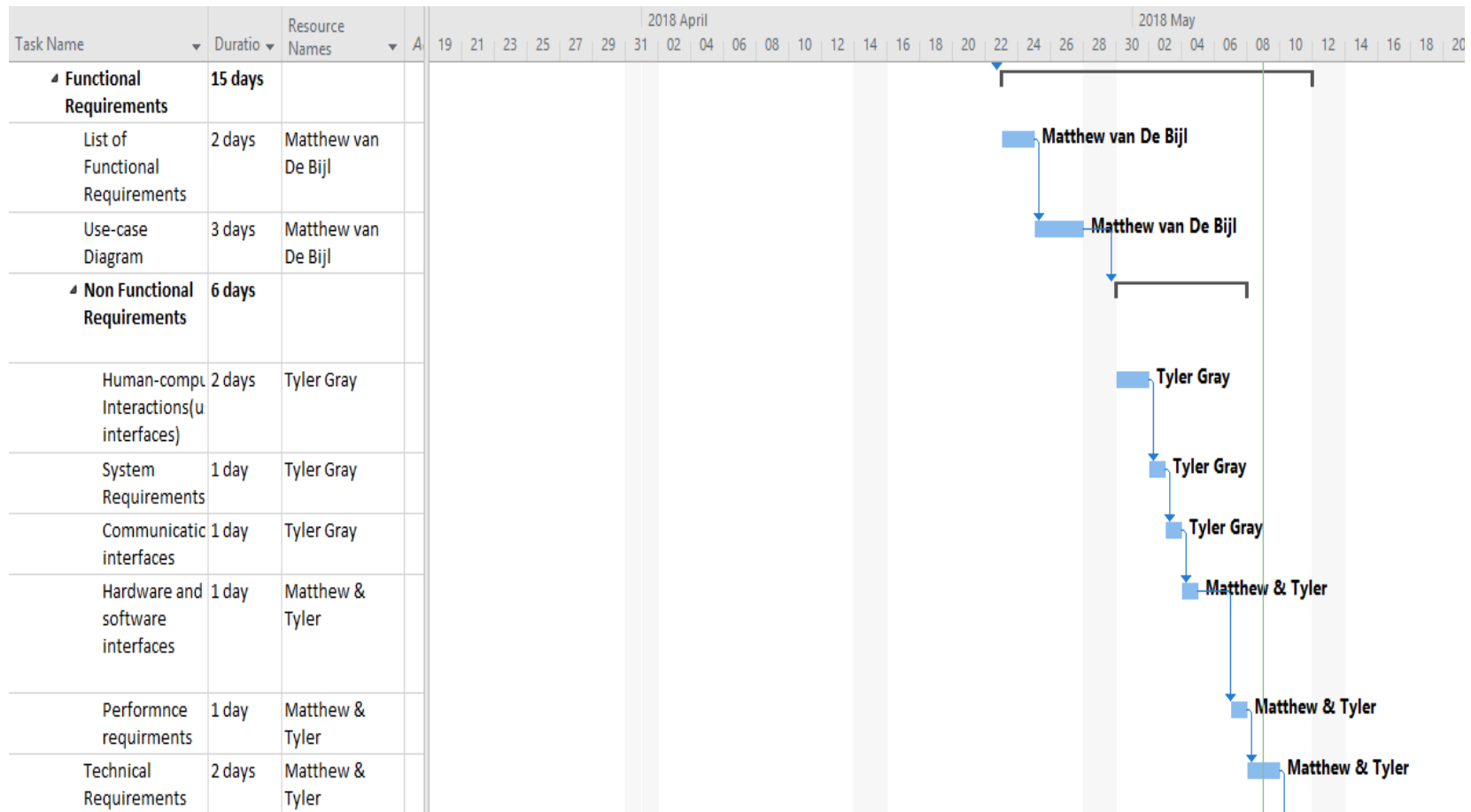


Figure 3 Gantt Chart part 2 of 3

Human-compu Interactions(u interfaces)	2 days	Tyler Gray	
System Requirements	1 day	Tyler Gray	
Communicatic interfaces	1 day	Tyler Gray	
Hardware and software interfaces	1 day	Matthew & Tyler	
Performnce requirments	1 day	Matthew & Tyler	
Technical Requirements	2 days	Matthew & Tyler	
User characteristics	1 day	Matthew & Tyler	
Operation enviromnent	1 day	Matthew & Tyler	
Submission	0 days	Matthew van De	
Deliverable 3	65 days		
Deliverable 4	20 days		
Deliverable 5	25 days		
Deliverable 6	10 days		

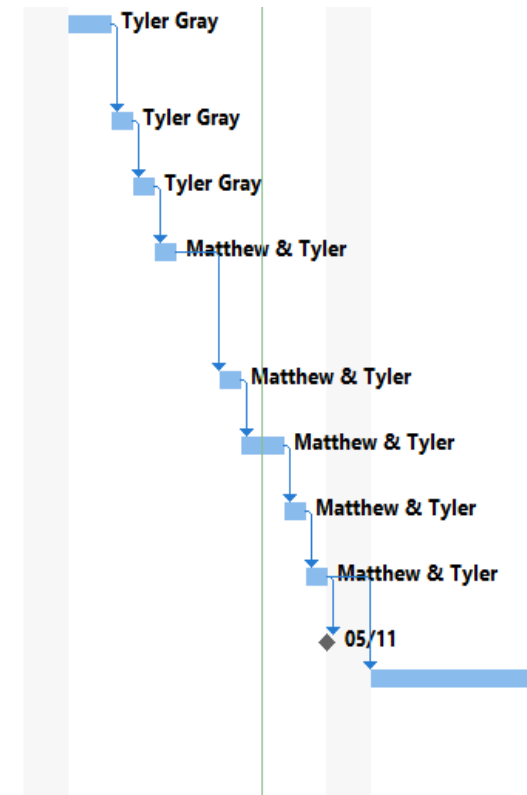


Figure 4 Gantt Chart part 3 of 3

As previously stated, each deliverable needs to be completed, reviewed and submitted before the submission dates. It is key that all components are completed on time. The Gantt chart should be updated for each submission. Add in here about Agile sprints vs Increments and Deliverables

2.4 User Requirements Document

2.4.1 Introduction

The following document will include necessary information concerning all user requirements needed for the system being produced.

2.4.1.1 Purpose

The type of system being produced is an application for mobile devices. The system will be supported by a database to hold all necessary information. A connection between the application and database will be established to retrieve and send information to the database. There will be three several types of readers for the system, as indicated in the table below.

Table 6 Table of Users

User	Description
Coaches	Coaches will have administrative privileges on the system.
Player	Players will only be able to see information displayed for them by the coaches.
Physiotherapist	The Physiotherapist will have certain administrative privileges.

All users will need to be able to interfaces with the system. All users will need to provide their own mobile devices.

2.4.1.2 Scope

RugBot is an application-based system. Rugbot will be supported on android and apple devices. A database will be included in the system to store and retrieve information. The main purpose of producing RugBot is to supply the Western Province Rugby Academy with an easy to use management system. The system has three parts which support the different administrative privileges for the intended users.

Coaches and physiotherapists would be provided with a username and password to login to their account. This is a once off sign in and Rugbot would not ask them for this information again unless the application is uninstalled.

Firstly, the coaches will have full administrative privileges on the system. They will be able to insert, delete and update all information stored in the database. They will also be granted access to retrieve information which is already stored in the database. The main issue which the Rugby Academy is currently dealing with will be doing admin duties in an easy manner. Rugbot will display a register of all the rugby players and allow the coaches the ability to click on a player's name if they are present at a match or a training session. This method of doing admin will provide efficient time management for the coaches. As soon as a coach clicks on a player name it will be saved into the database as being present on that specific day. Once the register has been completed, the coach will be provided with a "Done" button and thereafter all players name's which haven't been selected will be saved into the database as not present.

Another issue which the coaches are having is the ability to keep track of the jerseys being assigned to players on game day. Missing jerseys and nobody taking ownership of the responsibility has been a frequent matter. Rugbot will allow the coaches to assign the relative jersey to the player and store that information into the database until it is time to retrieve the information. This functionality will provide the coaches with concrete information on all jersey assigning to the players.

The weekly schedule of the players is not a fixed schedule. Therefore, the coaches would constantly need to provide a copy of the schedule to each of the players. This can be a time-consuming process and there may be players who do not receive the schedule. Rugbot will allow the coaches to update the weekly schedule on the mobile application and it will thereafter appear on the player's interface of the application. This functionality allows for a clear communication path between coaches and players.

Match fixtures and match results have also been a strain in the admin apartment. Assigning players to the different teams (first team, second team and third team) to a wide range number of matches has been a time-consuming effort on the coaches' part. For example; a player can

participate for the first time in an upcoming match but might not participate in that same team the following match. Assigning positions to players for games has also been a lengthy process. For example; a player might not play in the same position for every match. This can be due to the coach's decision or an injury occurrence. Therefore, RugBot will allow the coaches to update all match fixtures and results and it will be sent over to the player's end of the application. The positions of every player in every match will also be displayed for the player and therefore eliminates all confusion on game day.

Secondly, physiotherapists are a very important role in the rugby academy. They tend to the player's physical needs and ensures their capability out on the rugby field. Rugby is a contact sport and therefore injuries will occur. It is important that coaches and physiotherapists have a clear and open communication path. The issue at hand is that information about a player's health has not been forwarded to the coaches immediately which leaves a result of constant miscommunication. However, Rugbot will provide therapists with the ability to click on a player's name and insert any injuries or health issues which has occurred. They will have the option of stating whether and when the player can resume normal training and match games or if it is just a minor issue. They can also add any notes which will be relevant for the coaches to see. All this information will be displayed on the coaches' side of the application as soon as it has been inserted into the application.

Thirdly, the players will not have any administrative privileges on the application. Their side of Rugbot will only be viewing the weekly schedule, matches fixtures, match results and any other essential information which the coaches display.

2.4.1.3 Definitions, Acronyms & Abbreviations

The tables that follow outline all definitions, acronyms and abbreviations used in this document.

2.4.1.3.1 Definitions

Table 7 Definitions

Definition	Description
Gantt chart	A Gantt chart is a diagram used to display a project's schedule (Schwalbe, 2012).
Coach	Member of staff who instructs player.
Player	Rugby student.
Western Province Ruby Academy	Western Province Rugby Academy is to facilitate the growth of young athlete so that they may reach their full potential.
Nielsen's Heuristics	A high-level list of non-functional requirements presented by Nielsen (1994).

2.4.1.3.2 Acronyms & Abbreviations

Table 8 Acronyms and abbreviations

Term	Expansion
App	Application
API	Application Programme Interface
IDE	Integrated Development Environment
UI	User Interface
UX	User Experience
DBMS	Database Management System
JS	JavaScript
FR	Functional Requirements
NFR	Non-functional Requirements
GUI	Graphical User Interface

2.4.2 System Perspective

The system will take the form of a mobile application and a remote database. Users will interact with the database through the application. In order to achieve this configuration, the system will make use of a client-server model.

2.4.3 Functional Requirements

2.4.3.1 List of Functional Requirements

The table below outlines functional requirements that the final system needs to meet.

Table 9 Functional Requirements

Identifier	Requirement Description	Priority	Source
FR01	Users must use a one-time login to log in to the application for authorisation purposes.	High	RugBot Development Team
FR02	Coaches must be able to take an attendance list of students at practice.	High	Coaches
FR03	Coaches must be able to view a backlog of student's attendance for past dates.	High	Coaches
FR04	Coaches must be able to view a list of all their students and their availability for practise sessions and matches.	High	Coaches
FR05	Coaches and students must have a calendar with a practise match dates and times.	High	Coaches
FR06	The physiotherapist must be able to mark a student as injured and not able to practise or play matches.	High	Coaches
FR07	The physiotherapist must be able to add an estimated date of when a student will be able to practise again.	High	Coaches
FR10	The coach must be able to assign jersey numbers to players on match dates.	High	Coaches
FR11	All users need to be able to see the injury status of a player.	High	Coaches
FR18	Players must be able to update their personal information	High	RugBot Development Team
FR13	Coaches need to be able to schedule a match.	High	Coaches
FR14	Coaches and students need to be able to see match teams.	High	Coaches

Identifier	Requirement Description	Priority	Source
FR08	If a student missed more than three practise sessions, the coach must receive a notification of the student's absence.	Medium	Coaches
FR09	The coach must be able to see the total of boys at practice.	Medium	Coaches
FR12	Coaches and physio must be able to view the medical history of a player. A player must only be able to see their own data.	Medium	RugBot Development
FR15	Players must be notified when they are playing games.	Low	RugBot Development Team
FR16	Players must be notified when they have a physio appointment.	Low	RugBot Development Team
FR17	Players must be notified when they have missed a practice.	Low	RugBot Development Team

The table has been sorted by priority. All high priority requirements need to be completed before the delivery of the final system.

2.4.3.1.1 Use-case Diagram

Use-case diagram creation was informed by Bennet, et al. (2010).

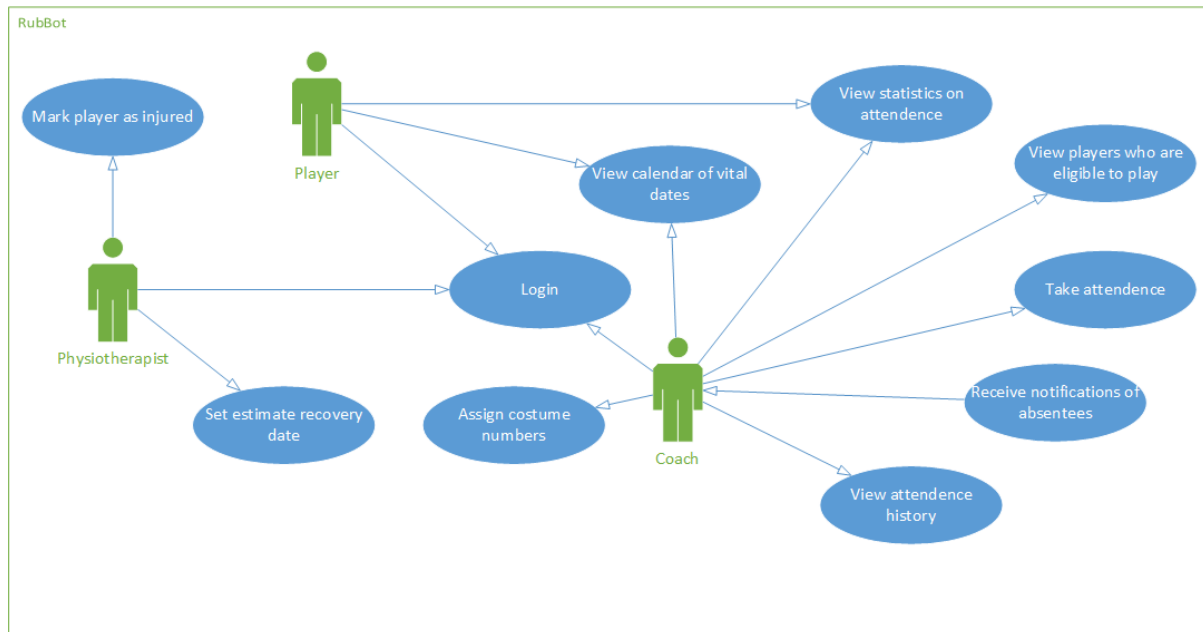


Figure 5 Use-case Diagram for RugBot

The system will consist of three users, or actors, namely: player, coach and physiotherapist. All users will need to be able to login to the system.

2.4.4 Non-functional Requirements

According to Chung, et al. (1996), non-functional requirements are also known as quality requirements. As previously stated, non-functional requirements do not directly describe what the system must do (Sommerville, 2001).

Table 10 Non-functional Requirements

Identifier	Requirement Description	Priority	Source
Performance			
NFR01	Database response times must be very quick.	High	RugBot Development Team
NR02	Quick response times in applications.	Medium	RugBot Development Team
NFR03	GUI must be quick and responsive.	Medium	RugBot Development Team
Design			
NR04	The GUI design must be minimalist and simple.	Medium	RugBot Development Team
NFR05	Navigation of the application must be sensible and straight-forward.	High	RugBot Development Team
Security			
NFR06	The database must only be accessible by authenticated users.	High	RugBot Development Team
NR07	A user must be able to access only data specific to their authorization level.	High	RugBot Development Team
Reliability			
NR08	The application should never crash and be bug-free.	High	RugBot Development Team
NR09	The application must be able to operate even when connected to the database is lost.	Low	RugBot Development Team
NR10	The database must be able to have multiple users access it at the same time.	High	RugBot Development Team
Scalability			
NR11	The system must be able to grow in terms of active users.	Low	RugBot Development Team

In conclusion, User data needs to be kept confidential, as suggested by Chung, et al. (1996). The system must not waste the users time, as suggested by Chung, et al. (1996).

2.4.4.1 Human-Computer Interactions

All system interfaces should comply with design principles to provide users with a holistic user experience. All users facing interfaces should strictly comply with Nielsen's Usability Heuristics as suggested by Nielsen (1994). These include:

1. Users should be provided with feedback;
2. The system must comply with the user's model of reality;
3. The system's interfaces must remain consistent is designed;
4. Users must always be able to exit an unwanted state;
5. The system must protect against error-prone conditions;
6. Users should always be able to recognise what they need to do, rather than recall what they need to do.
7. The system needs to be efficient;
8. The GUI design must be minimalist and simple
9. The system must assist the user in overcoming errors; and;
10. Documentation, such as a user manual, must be available to the user (Nielsen, 1994).

The guidelines listed above should be a high-level set of non-functional (Preece, et al., 2015). Stair and Reynolds (2016) suggests that all user-facing interfaces should be tested against these guidelines.

2.4.4.2 System Requirements

The final system delivered will require end users to own their own mobile devices as well as a remote server for the database. A connection will need to be made between user devices and the database users will need to provide their own mobiles devices.

2.4.4.3 Communication Interfaces

The app will need to facilitate easy communication.

2.4.4.4 Hardware & Software Interfaces

Users will interact with the system through mobile devices. Users will need to provide their own mobile devices for the app to be installed on. The system will require a remote server. This server will need to host the database that mobile devices interact with.

2.4.4.5 Performance Requirements

The system will need to maintain a high-level of performance throughout the entire system. The performance of a system has a direct impact on the user experience.

2.4.5 Technical Requirements

The table below outlines the technical requirements of the system.

Table 11 Technical requirements

Identifier	Requirement Description	Priority	Source
TR01	Users will need a mobile device running either Android or iOS to use the application.	High	Coaches
TR02	Users will need an internet connection to connect to the database.	High	Coaches
TR03	Firebase will be used for the database needs.	Low	RugBot Development Team
TR04	The application will be developed on the Ionic framework.	Low	RugBot Development Team
TR05	HTML, Sass and TypeScript (a superset of JavaScript) are the languages that will be used for development.	High	RugBot Development Team
TR06	Developers will need Android and iOS devices for testing.	High	RugBot Development Team
TR07	WebStorm or any modern code editor with TypeScript support will be used for writing and editing code.	High	RugBot Development Team

As previously stated, the project's technical requirements should adapt to recent technology and market change. It is vital that the client is presented with a truly modern system.

2.4.6 User Characteristics

The table below outlines all users of the system.

Table 12 User characteristics

User Role	Level of computer literacy	Level of Experience	language	Technical Skills Required
Developer	Expert	Expert	English	Expert
Coach	Proficient	Novice	English	Literate
Player	Literate	Novice	English	Literate
Physio	Proficient	Novice	English	Literate

2.4.7 Operational Environment

The system will need to exist in two operational environments, a development environment and practical environment. The development environment will be used for developing and testing the system.

As RugBot is a mobile application, the final system will be deployed to mobile devices. The system will need to make use of a database. The database will need to reside on a remote server. The database will need to be secured to ensure that user data remain confidential.

3 Information System Design

3.1 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.

3.1.1 Use Case Diagrams

Coaches will use the system to track player attendance and view the status of injured players. A coach is able to take a register 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).

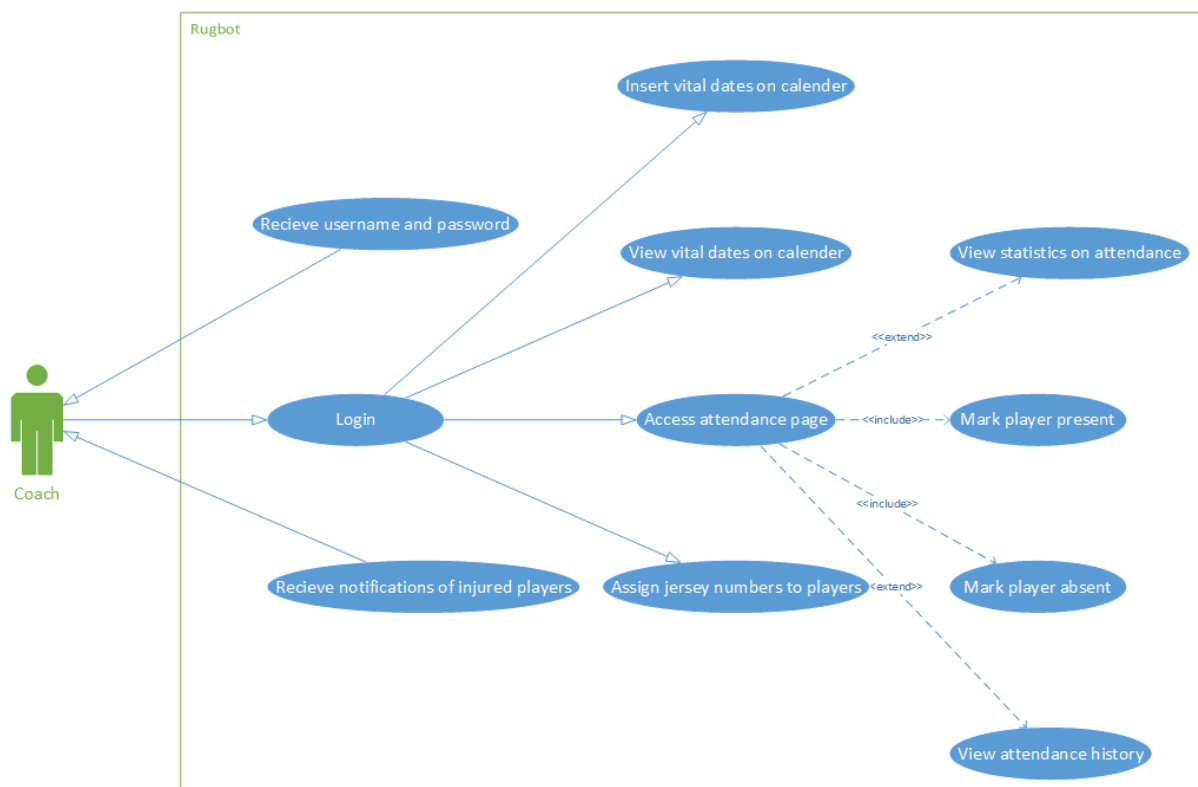


Figure 6 Coach User-case diagram

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

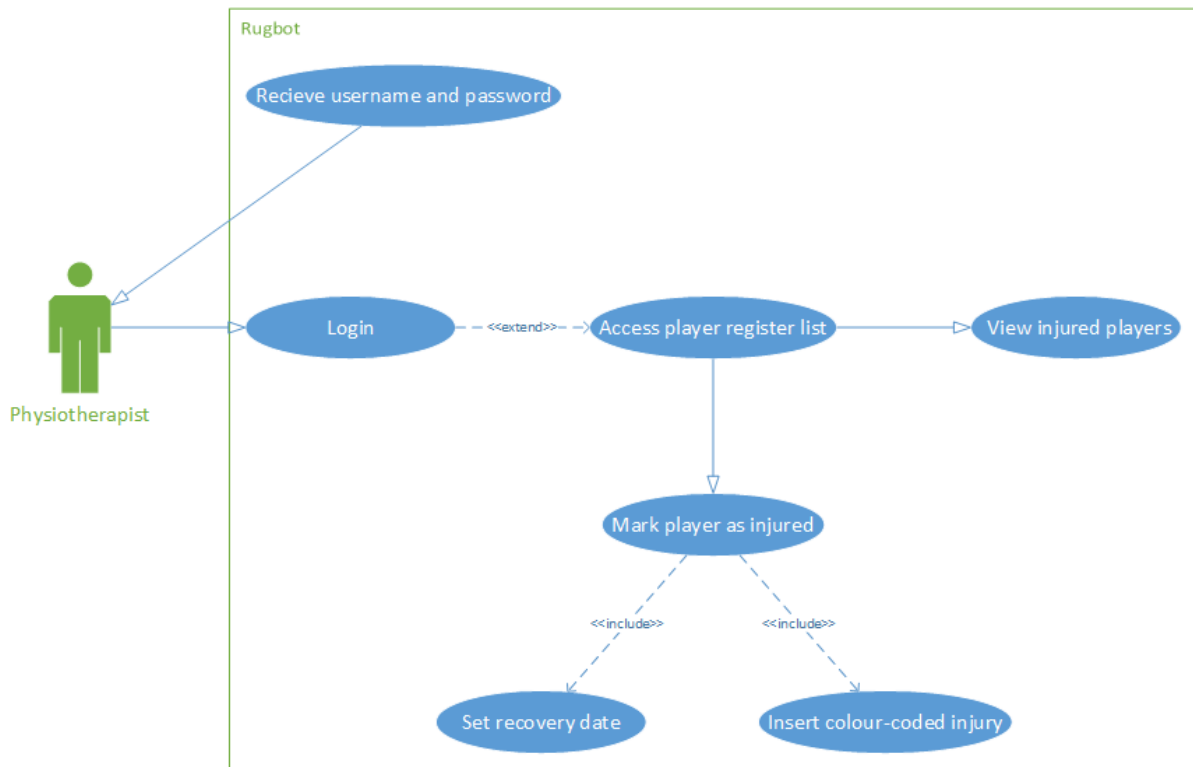


Figure 7 Physiotherapist Use-case diagram

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 the users, players have the least functionality. Once registered, players are able to log in and view 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).

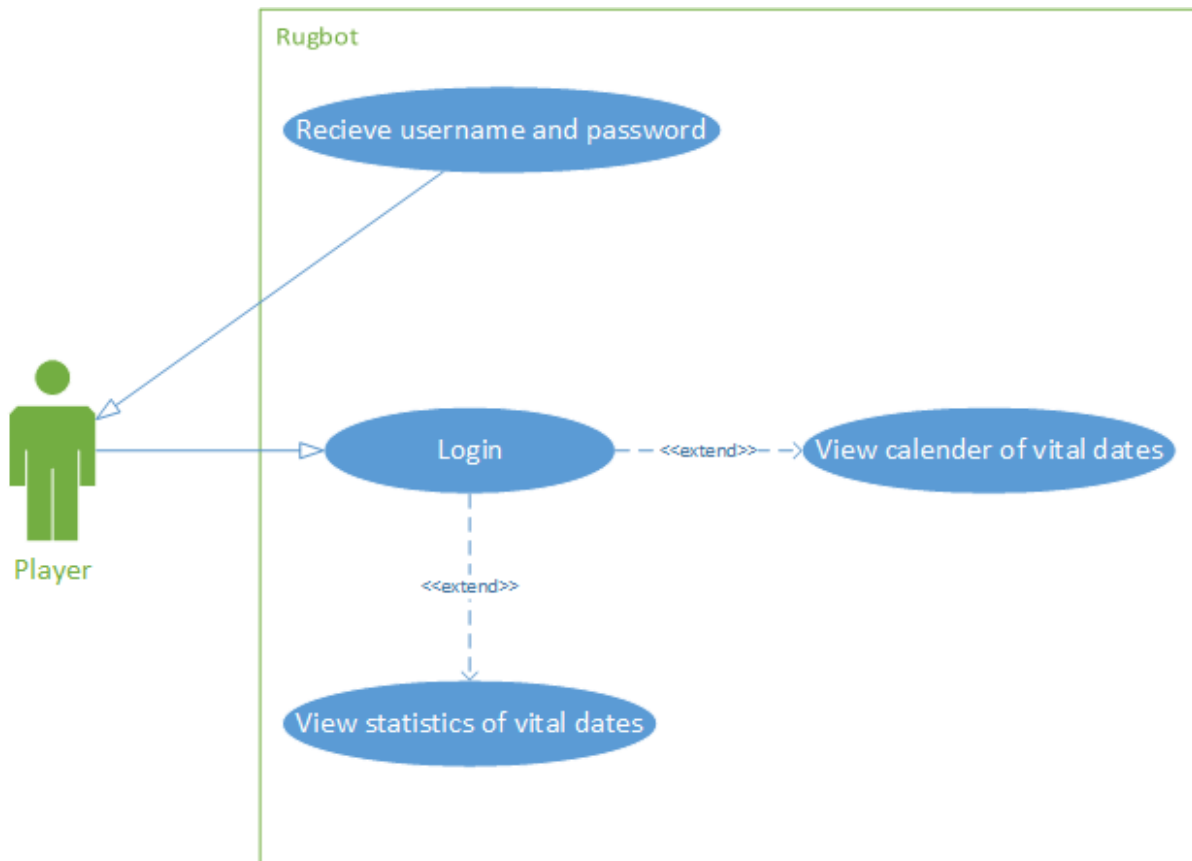


Figure 8 Player Use-case diagram

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.

3.1.2 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).

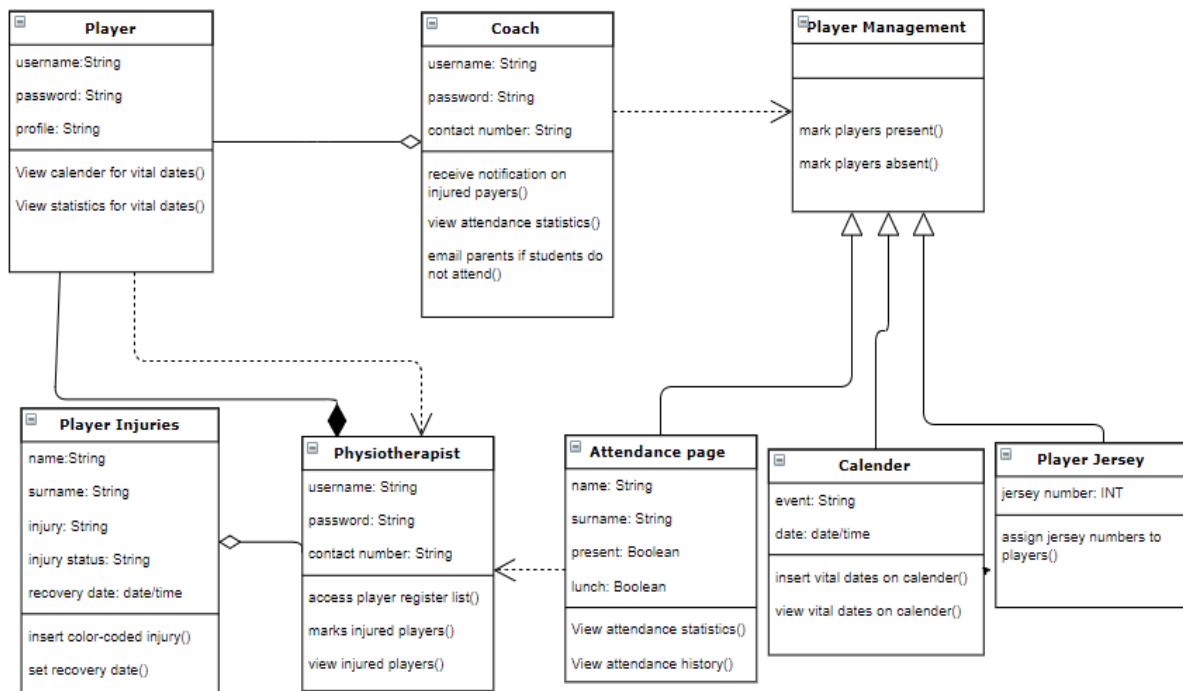


Figure 9 Class Diagram of RugBot System

The classes may change while developing the project.

3.1.3 Activity Diagrams

3.1.3.1 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).

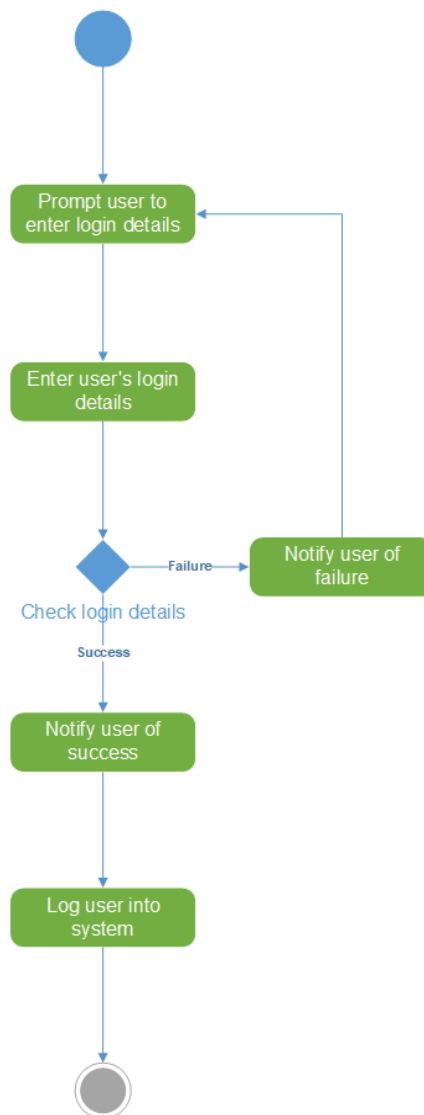


Figure 10 Login activity diagram

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.

3.1.3.2 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 11 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.

3.1.3.3 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).

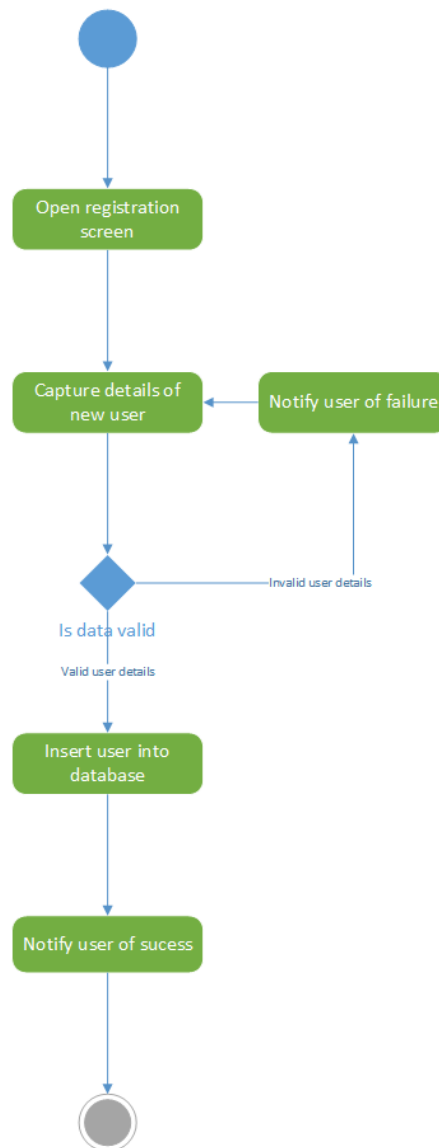


Figure 12 Registration activity diagram

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.

3.1.4 Sequence Diagrams

3.1.4.1 User Registration Sequence Diagram

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

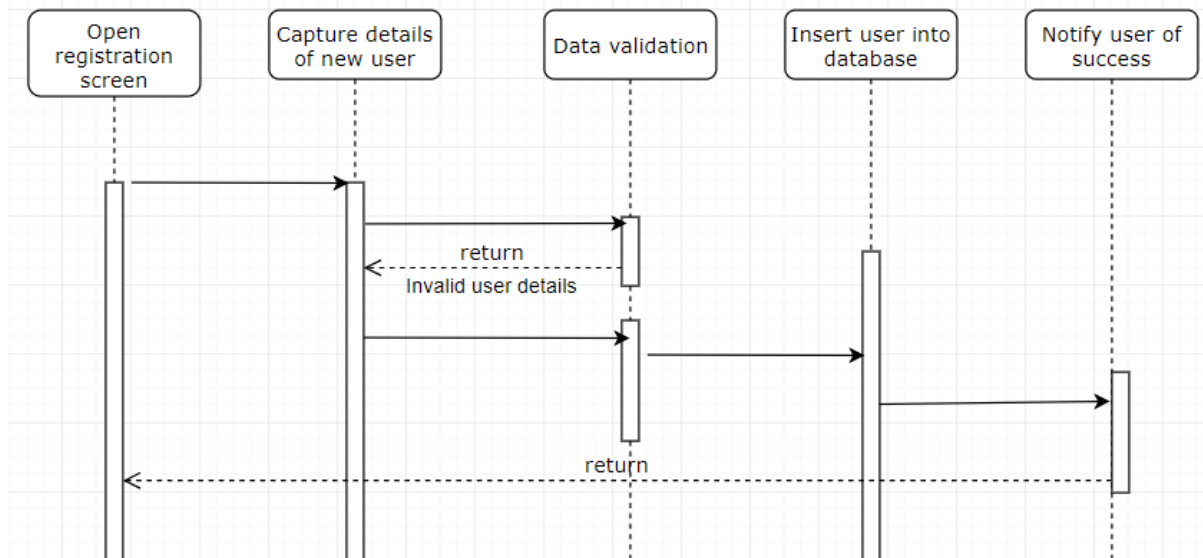


Figure 13 Registration sequence diagram

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.

3.1.4.2 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).

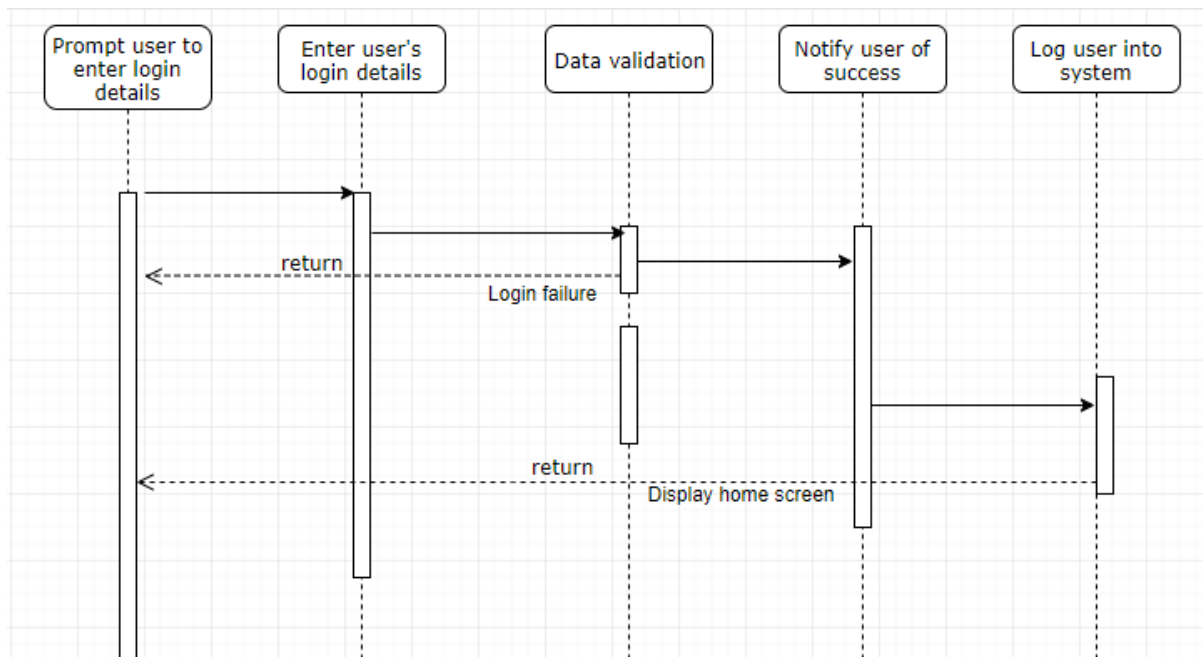


Figure 14 Login sequence diagram

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.

3.1.4.3 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).

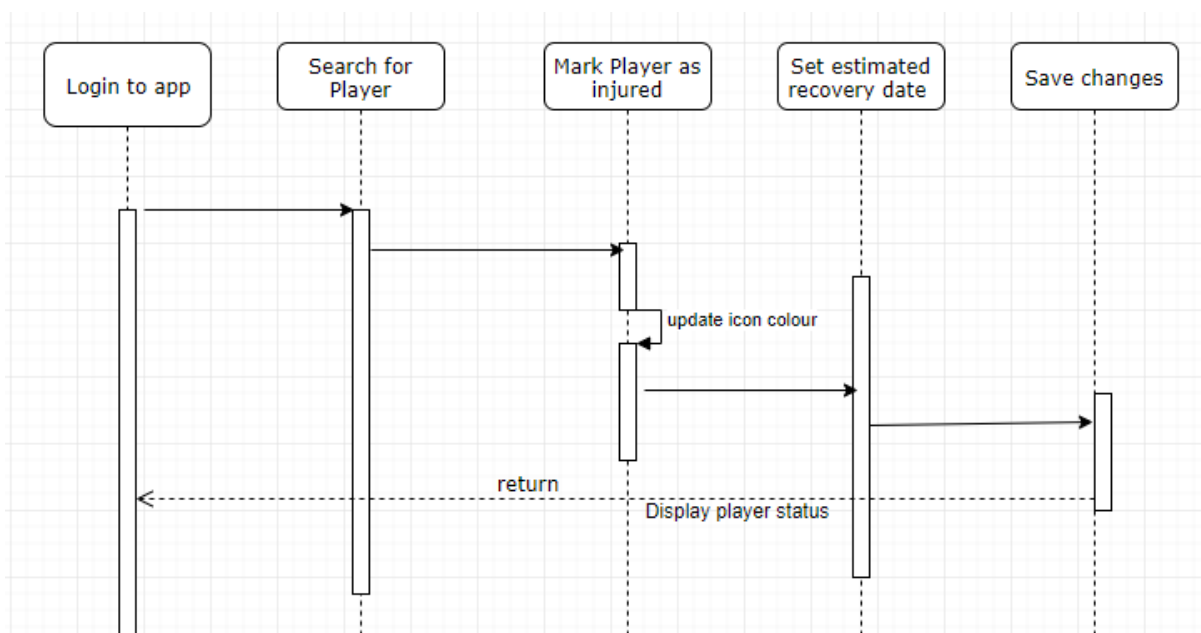


Figure 15 Injured player activity diagram

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

3.2 Physical Design

3.2.1 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).

3.2.2 System Testing

3.2.2.1 Testing Types

There are several types of tests that need to be performed (Deitel & Deitel, 2012). Testing is vital in the development of any information system (Whitman, et al., 2012). 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, regression 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 13 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 <i>real-world operating</i> 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 individual 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 (Weisfeld, 2013). It is for this reason that the clients and users should be involved in the testing process (Valacich, et al., 2015).

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.

3.2.2.2 Testing Template

All test plan has been created using the following testing template. How successful a test is should be measured using a four-point Likert scale.

Page:

Test Date:

Description:

Test for Type of User:

Tested By:

Type of Test:

Signature:

Table 14 Testing template

Home Page USER			
Test ID	Requirement	Successful*	Comments

* Use a Likert scale

Additional Comments:

3.2.2.3 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 15 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.		

* Use a Likert scale

Additional Comments:

Sign in Test continued:

Table 16 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.		

* Use a Likert scale

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:
Functionality

Type of Test:

Signature:

Table 17 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.		

* Use Likert scale

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 18 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.		

* Use Likert scale

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 19 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.		

* Use a Likert scale

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 20 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.		

* Use a Likert scale

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 21 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		

* Use a Likert scale

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: Physiotherapist

Tested By:

Type of Test: Functionality

Signature:

Table 22 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		

* Use a Likert scale

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 23 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: <ul style="list-style-type: none">• 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.		

* Use a Likert scale

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 24 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.		

* Use a Likert scale

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 25 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.		

* Use a Likert scale

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 26 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.		

* Use a Likert scale

Additional Comments:

4 Testing Results

4.1 System Interfaces

Various prototypes have been created throughout the development of the project. All prototypes have been created by following guidelines suggested by Nielsen (1994), Preece, et al. (2015) and Bennet, et al. (2010). Please note that the prototypes presented will differ from the designs that are present in the final deliverable. As suggested by Schwalbe (2012) Pressman and Maxim (2015) and Sommerville (2001) design and development will be done in iterations.

4.1.1 First Iteration Prototypes

The prototypes were created using a variety of programmes. Please refer to the caption of each image to see what it is.



Figure 16 Prototype welcome page

The image above is of a potential welcome page. The welcome page would be shown open. This feature may be removed in the final deliverable of the system.

The image below is of a potential registration frame. Users would use the registrations frame to create new users.



Figure 17 Prototype registration page

This feature may be removed in the final deliverable of the system.

The two images below are prototype login pages. Users would use the login page to open login in to the system.



Figure 18 Prototype login page

All users should be shown the same login page.

Both coaches and players need to be able to view the weekly planner.

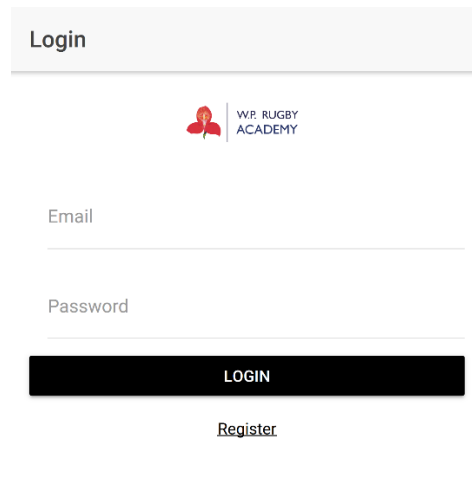


4.1.2 Second Iteration Prototypes

The following screens should bear a closer resemblance to the final product.

4.1.2.1 Login screen

Before entering the application use they are required to enter their login credentials.

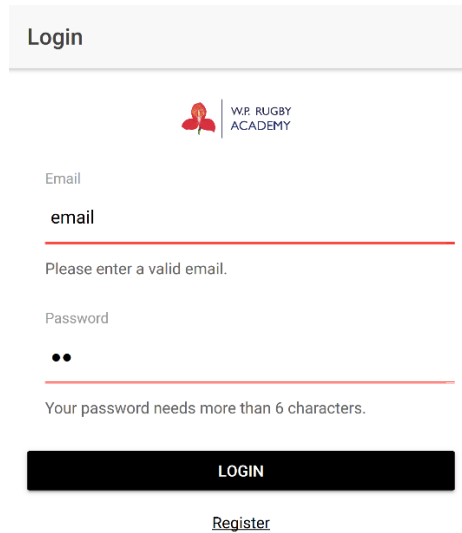


The login screen features a light gray header with the word "Login" in a dark gray font. Below the header is the W.P. Rugby Academy logo, which consists of a red stylized figure and the text "W.P. RUGBY ACADEMY". Underneath the logo are two input fields: "Email" and "Password", each with a light gray border and a small red eye icon for password visibility. Below the input fields is a black button with the word "LOGIN" in white capital letters. At the bottom of the screen is a link labeled "Register" in a dark gray font.

Figure 20 Login screen

Once users have entered their login credentials they should proceed to the home page. If a user does not have an account, by pressing the register button they are directed to the registration page.

Users are required to enter valid inputs. If a user enters data that is of an incorrect format the application will notify them.



The screenshot shows a login form for the W.P. Rugby Academy. The form has two input fields: 'Email' and 'Password'. The 'Email' field contains the text 'email' and has a red error message below it: 'Please enter a valid email.' The 'Password' field contains two dots and has a red error message below it: 'Your password needs more than 6 characters.' Below the password field is a black 'LOGIN' button. At the bottom of the form is a link that says 'Register'.

Login

W.P. RUGBY
ACADEMY

Email

email

Please enter a valid email.

Password

••

Your password needs more than 6 characters.

LOGIN

[Register](#)

Figure 21 Login screen validate

As seen above, all user input is validated before they can proceed.

If the login button is pressed while invalid data is entered the users will be shown the following error message.

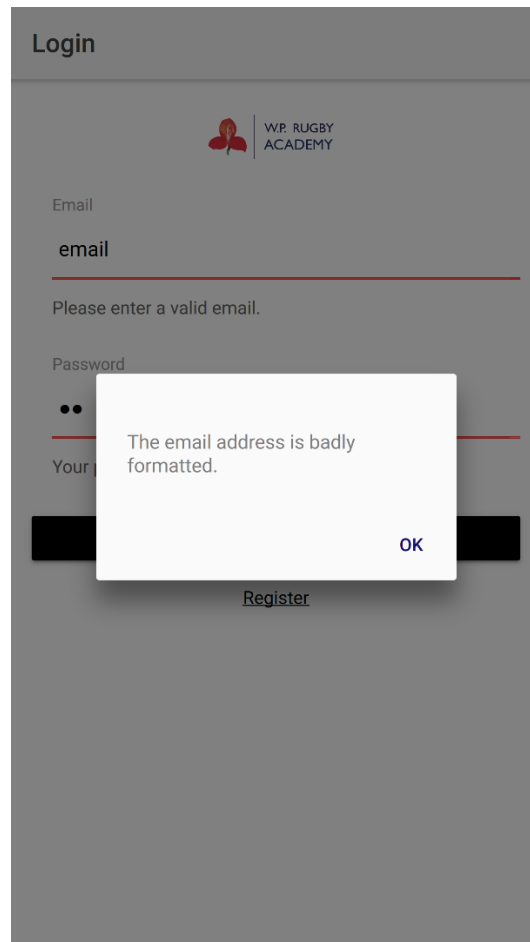
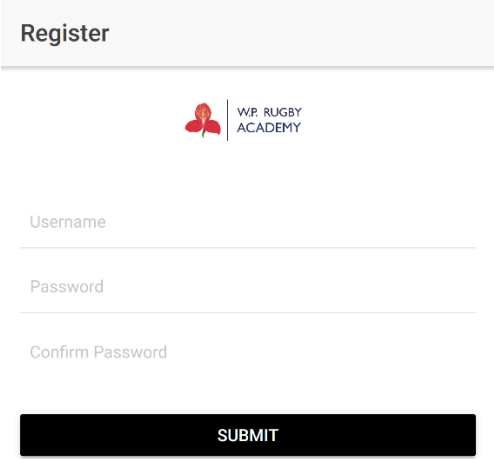


Figure 22 Login screen modal

Users are required to enter valid data before they proceed.

4.1.2.2 Registration screen

New users are able to register from the registration page. Users are prompted to enter their details. All inputs are validated.



The registration screen features a light gray header with the word "Register" in bold. Below the header is the W.P. Rugby Academy logo, which consists of a red stylized flower icon and the text "W.P. RUGBY ACADEMY". The form contains three input fields: "Username", "Password", and "Confirm Password", each with a light gray border and placeholder text. At the bottom of the form is a black "SUBMIT" button with white text.

Figure 23 Registration screen

Once registered, new users are able to login. The registration process is completed by pressing the 'submit button'.

4.1.2.3 Coach list screen

Coaches should be presented with the coach page. From this page coaches are able to view and edit the status of players and their marches.

Rugby Players

Search

First Name	Last Name	Action
Jill	Smith	Edit
Eve	Jackson	Edit
Adam	Johnson	Edit

Figure 24 Coach list

Coaches are able to search for players. Once found, a coach should be able to view a player's information.

4.1.2.4 Physio list screen

The physio list is similar to the coach list. Physios are able mark players as injured

Rugby Players			
<input type="text" value="Search"/>			
First Name	Last Name	Status	Action
Jill	Smith	Recieve from physio	View
Eve	Jackson	Recieve from physio	View
Adam	Johnson	Recieve from physio	View



Figure 25 Physio list

Like to coach list, physios are able to search for players.

4.1.2.5 Physio form screen

Physios should use the following form to mark players as injured. All inputs are validated.

The image displays two versions of the 'Injury Details' form side-by-side. Both forms have a header with the 'W.P. RUGBY ACADEMY' logo. The left form is a wireframe with placeholder text 'get from database' for the First Name, Last Name, and Injury Status fields. The right form is a functional mockup with a blue 'Submit' button at the bottom. The right form also includes a 'Date seen' field with the placeholder 'Enter Date Seen' and a 'Date To Resume Training' field with the placeholder 'Enter Date To Resume Training'. The 'Injury Status' field in the right form is a dropdown menu with 'Red(actual colour)' selected. The 'Comments' field in the right form is a text area with the placeholder 'Comment..'.

Injury Details

W.P. RUGBY ACADEMY

First Name
get from database

Last Name
get from database

Injury Status
Red(actual colour)

Date seen
Enter Date Seen

Date To Resume Training
Enter Date To Resume Training

Comments
Comment..

Submit

Figure 26 Physio form

Once a player is marked as injured the change needs to be reflected on the coaches list.

4.2 Test Results

Page: Sign in

Test Date: 11/09/2018

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: Group 2

Type of Test: Functionality

Signature:

Table 27 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.	4	None
SN002	A user will receive a validated username and password.	4	None
SN003	A user can insert the username in the username text field.	4	None
SN004	A user can insert the password in the password text field.	4	None
SN005	A user can click on the Sign In button to Sign in.	4	None

* Use a Likert scale

Additional Comments:

Sign in Test continued:

Table 28 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.	4	None
SN007	The text field Password is highlighted if an incorrect password is inserted.	4	None
SN008	The Username and Password field will be required to be filled in before Signing In.	4	None

* Use a Likert scale

Additional Comments:

Page: Home Page

Test Date: 11/09/2018

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

Test for Type of User: Coach

Tested By: Group 2

Type of Test: Functionality

Signature:

Table 29 Home page test plan

Home Page Coach			
Test ID	Requirement	Successful*	Comments
HPC001	A user can click on the menu button.	0	Incomplete
HPC002	A user can access the following pages from the menu: 8. Calendar; 9. Attendance; 10. Injured Players and 11. Gameday administration.	0	Incomplete

* Use Likert scale

Additional Comments:

Page: Calendar Page

Test Date: 11/09/2018

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

Test for Type of User: Coach

Tested By: Group 2

Type of Test: Functionality

Signature:

Table 30 Calendar page test plan

Calendar Page Coach			
Test ID	Requirement	Successful*	Comments
CPC001	A user can access the Calendar Page.	1	Crashes on start
CPC002	A user can view a calendar.	1	Crashes on start
CPC003	A user can select a date on the calendar.	1	Crashes on start
CPC004	A user can insert information on selected dates on the calendar.	1	Crashes on start
CPC005	A user can submit information which was inserted into the calendar	1	Crashes on start
CPC006	A user can modify information in the calendar.	1	Crashes on start

* Use Likert scale

Additional Comments:

Page: Attendance Page

Test Date: 11/09/2018

Description: This page will allow coaches to take attendance.

Test for Type of User: Coach

Tested By: Group 2

Type of Test: Functionality

Signature:

Table 31 Attendance page test plan

Attendance Page Coach			
Test ID	Requirement	Successful*	Comments
APC001	A user can access the Attendance Page.	4	
APC002	A user can view a register of all the players.	4	
APC003	A user can click on "✓" icon if a player is present.	0	Incomplete
APC004	A user can add a comment next to a player's name.	0	Incomplete
APC005	A user can access the search bar.	4	
APC006	A user can search for a player by first or last name.		1Put in code
APC007	A user can view the attendance history	1	In progress. Make new page.
APC008	A user can click on the submit button to save information inserted.	0	Incomplete

* Use a Likert scale

Additional Comments:

Page: Injury Page

Test Date: 11/09/2018

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: Group 2

Type of Test: Functionality

Signature:

Table 32 Injury page test plan

Injury Page Coach			
Test ID	Requirement	Successful*	Comments
IPC001	A user can access the Injury Page.	0	
IPC002	A user can view all players who are injured.	0	
IPC003	A user can view comments about a player's injury.	0	Put in search status.
IPC004	A user can view when a player can resume training.	0	
IPC005	A user can view when a player can resume participating in matches.	0	

* Use a Likert scale

Additional Comments:

Basically, view button receiving information from the physio form.

Page: Game Administration Page

Test Date: 11/09/2018

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

Test for Type of User: Coach

Tested By: Group 2

Type of Test: Functionality

Signature:

Table 33 Game admin page test plan

Game Administration Page Coach			
Test ID	Requirement	Successful*	Comments
GPC001	A user can access the game administration page.	0	Incomplete
GPC002	A user can access an attendance form.	0	Incomplete
GPC003	A user can allocate a jersey to a relative player.	0	Incomplete
GPC004	A user can insert and allocate a player's position.	0	Incomplete
GPC005	A user can save inserted information by clicking on the Save button	0	Incomplete

* Use a Likert scale

Additional Comments:

List of jersey name

- ➔ Click on field
- ➔ Open list of players (modal with search bar)

Page: Home Page

Test Date: 11/09/2018

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

Test for Type of User: Physiotherapist

Tested By: Group 2

Type of Test: Functionality

Signature:

Table 34 Homepage test plan for physio

Home Page Physiotherapist			
Test ID	Requirement	Successful*	Comments
HPP001	A user can click on the menu button.	2	In progress
HPP002	A user can access the following pages from the menu: 12. Injuries	2	In progress

* Use a Likert scale

Additional Comments:

Page: Injuries Page

Test Date: 11/09/2018

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: Group 2

Type of Test: Functionality

Signature:

Table 35 Injury page test plan for physio

Injuries Page Physiotherapist			
Test ID	Requirement	Successful*	Comments
IPP001	A user can access the Injuries Page.	4	
IPP002	A user can access the player's register.	4	
IPP003	A user can insert the type of injury.	4	
IPP004	A user can insert the seriousness of an injury by colour-coding a player's name with either: <ul style="list-style-type: none">• Red or• Orange.	0	Automatic resume training colour.
IPP005	A user can insert the colour-code green when a player has been cleared of all injuries.	4	
IPP006	A user can insert the date when a player can return to training.	4	
IPP007	A user can insert the date when a player can participate in games.	4	
IPP008	A user can insert any additional comments about the injury.	4	
IPP009	A user can save all inserted information by clicking on the Save button.	4	Submit button

* Use a Likert scale

Additional Comments:

Edit command.

Page: Home Page

Test Date: 11/09/2018

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

Test for Type of User: Players

Tested By: Group 2

Type of Test: Functionality

Signature:

Table 36 Homepage test plan for players

Home Page Players			
Test ID	Requirement	Successful*	Comments
HPL001	A user can click on the menu button.	0	Incomplete
HPL002	A user can access the following pages from the menu: 13. Calendar and 14. Game Day.	0	Incomplete

* Use a Likert scale

Additional Comments:

Page: Calendar Page

Test Date: 11/09/2018

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: Group 2

Type of Test: Functionality

Signature:

Table 37 Calendar page test plan for players

Calendar Page Players			
Players			
Test ID	Requirement	Successful*	Comments
CPL001	A user can access the Calendar Page.	0	Incomplete
CPL002	A user can view all important dates on the calendar.	0	Incomplete

* Use a Likert scale

Additional Comments:

Page: Match Day Page

Test Date: 11/09/2018

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: Group 2

Type of Test: Functionality

Signature:

Table 38 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.	0	Incomplete
MDP002	A user can view all information on the page.	0	Incomplete

* Use a Likert scale

Additional Comments:

5 User Manual

5.1 Introduction

The RugBot development team was tasked with the creation of the RugBot mobile application. The aim of the application was to develop a system to manage and support the record keeping of the Western Province Rugby Academy.

The final outcome of the system was what the RugBot development intended it to be. The system has met all customer requirements in an efficient manner.

Concerning the design point of view, the RugBot application is user-friendly with a clean minimalistic design for ease of application. The application is designed in such a way that every intended user has the ability to make use of the application without any difficulty or confusion.

5.2 Getting Started

Concerning the design point of view, the RugBot application is user-friendly with a clean minimalistic design for ease of application. The application is designed in such a way that every intended user has the ability to make use of the application without any difficulty or confusion.

5.3 How to Install

Users will first need to register with the rugby academy. Once registered users will be required to visit the appropriate app store. The mobile device should handle the installation of the application. Users should be provided with a link to the necessary download page upon registration with the academy.

Users are then required to register an account. This process is covered later in this document.

5.4 Instructions

5.4.1 Register



← Register

W.F. RUGBY ACADEMY

Email

Name

Surname

Type

Password

Confirm Password

REGISTER

Figure 27 Register

A user will need to register before they can access the RugBot mobile application. A user will need to:

- Enter Email;
- Enter Name;
- Enter Surname;
- Type;
- Password;
- Confirm Password and
- Click on the “Register” button.

Once the user has been successfully registered, the user will be redirected to the Login Page.

5.4.2 Login



Figure 28 Login

Once a user has registered successfully, the user will be required to login into the RugBot application.

The user will need to provide the following information:

1. Email and
2. Password.

To login into the application a user is required to click on the “login” button.

If a user has not yet registered on the RugBot mobile application, the register button can be clicked to do so.

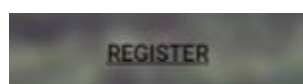


Figure 29 Register Link

5.5 Users

5.5.1 Coach

5.5.1.1 Home Page



Figure 30 Coach Home Page

The landing page for the user coach is displayed as the home page.

A coach has three button options to click upon to navigate to the required page:

1. Players;
2. Attendance and
3. Calendar.

To view other pages from the Home page:

1. Click on the Menu Toggle Icon.
- 2.



Figure 31 Menu Icon

Once the menu icon has been clicked, the menu will appear by sliding across from the left side of the mobile application.

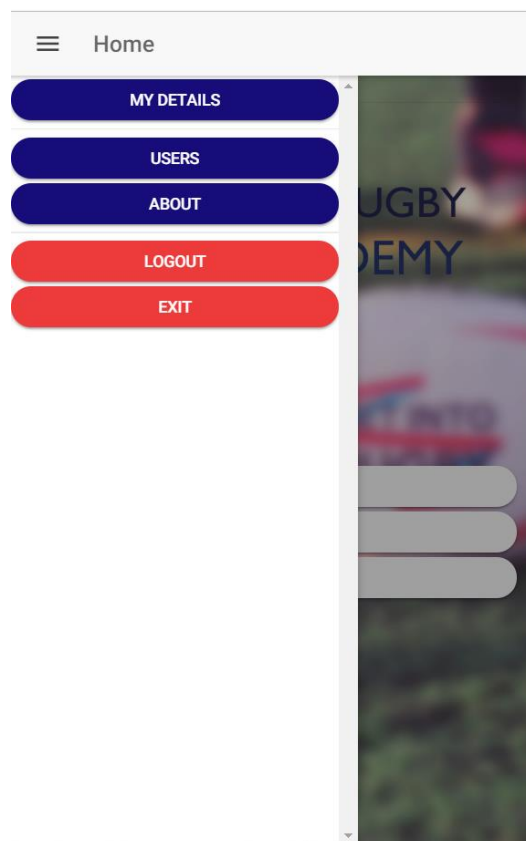


Figure 32 Menu bar

A coach can click on either button:

3. My Details;
4. Users;
5. About;
6. Logout; and
7. Exit.

5.5.1.2 *Players Page*

The purpose of the “Players” page is for coaches to conduct a quick and easy register attendance before a match day or practice day.










← Rugby Players		
Date	02-11-2018	
	Name	Level of Injury
	Coetzee, Dirk	NO PLAY
	Grobler, Brandon	NO PLAY
	Williams, Deangelo	INJURED
	Mutua, Andrew	OKAY
	Maguire, Grayson	INJURED
	Staples, Francois	INJURED
	Lourens, Dian	OKAY
	Van Der Loo, Chad	OKAY
	Coetzee, Alexander	INJURED

Figure 33 Coach Players Page

To select the date, click on the “Date” word or the current date displayed.

Date

02-11-2018

Figure 34: Date

Once the date has being clicked, a date picker will appear.

The screenshot shows a mobile application interface. At the top, there is a header bar with a back arrow and the text 'Rugby Players'. Below this, there is a section for 'Date' with the value '02-11-2018'. Underneath, there is a table of rugby players with their names and injury levels. A date picker is open at the bottom of the screen, showing the current date '02-11-2018' in blue text, indicating it is selected. The date picker has three columns: day, month, and year. The day column shows '09', '01', '02', '03', '04'. The month column shows '10', '11', '12'. The year column shows '2018', '2017', '2016'. At the bottom right of the date picker, there are 'CANCEL' and 'DONE' buttons.

	Name	Level of Injury
👍	Coetzee, Dirk	NO PLAY
👍	Grobler, Brandon	NO PLAY
👍	Williams, Deangelo	INJURED
👍	Mutua, Andrew	OKAY
👍	Maguire, Grayson	INJURED

09
01 10
02 11 2018
03 12 2017
04 2016

CANCEL DONE

Figure 35 Date picker

The current date of the date picker being open will appear. However, if a different date is required a coach can:

1. Scroll through the day, month and year.

To finish this action, a coach can click on the:

2. “Cancel” button to discard the action or
3. “Done” button to save the action.

A coach will click on the “thumbs up” icon to register a player for attending on the said day.



Figure 36 Thumbs up icon

Once the icon has been clicked, it will turn green to indicate that the information has been saved.



Figure 37 Clicked upon Thumbs Up icon

If a player has sustained an injury, the injury will be displayed on a button and will be colour-coded against the level of seriousness of the injury.

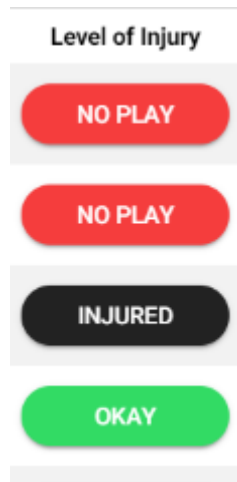


Figure 38 Level of Injury

If a coach wishes to view the type of injury sustained and the estimated date for recovery, the coach will click on the button to bring up a modal displaying the required information.

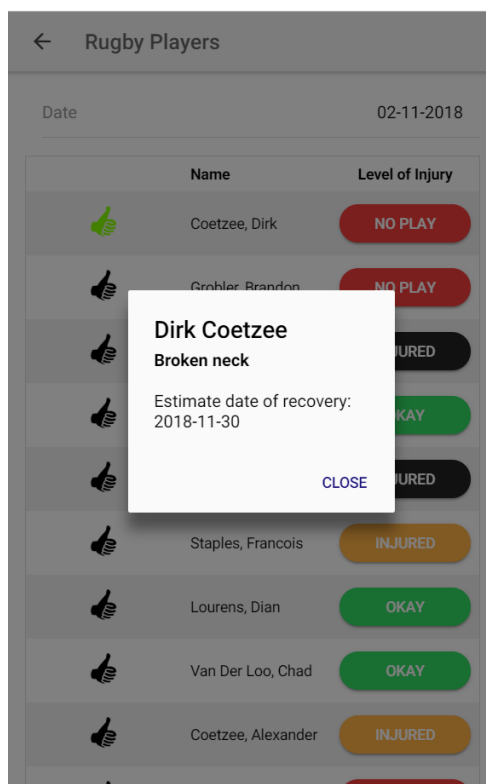


Figure 39 Injury Information

To close the modal, click on the “Close” button.



Figure 40 Close Button

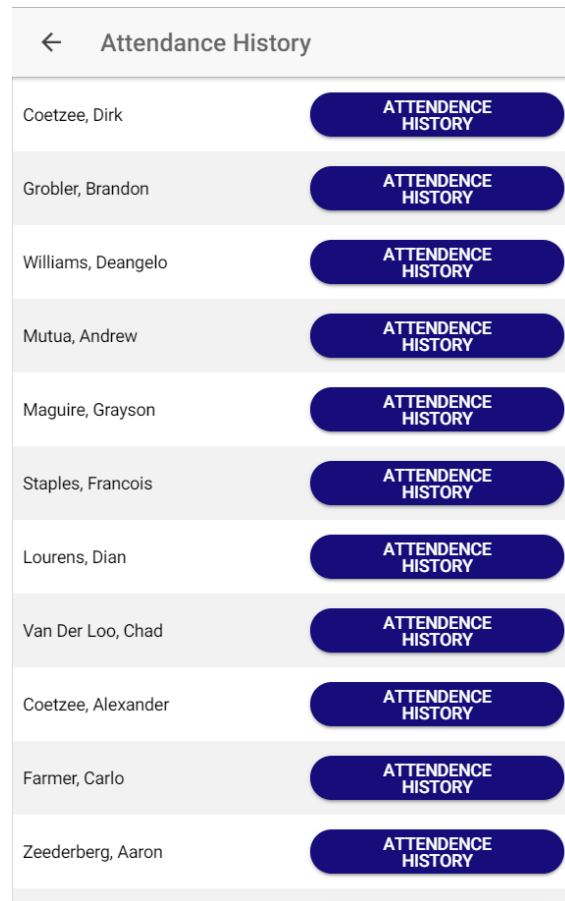
To return to the Home page, click on the back-arrow icon.



Figure 41 Back Arrow

5.5.1.3 Attendance Page

The purpose of the Attendance page is to display a list of players and all the dates that they have attendant activities.



← Attendance History	
Coetzee, Dirk	ATTENDANCE HISTORY
Grobler, Brandon	ATTENDANCE HISTORY
Williams, Deangelo	ATTENDANCE HISTORY
Mutua, Andrew	ATTENDANCE HISTORY
Maguire, Grayson	ATTENDANCE HISTORY
Staples, Francois	ATTENDANCE HISTORY
Lourens, Dian	ATTENDANCE HISTORY
Van Der Loo, Chad	ATTENDANCE HISTORY
Coetzee, Alexander	ATTENDANCE HISTORY
Farmer, Carlo	ATTENDANCE HISTORY
Zeederberg, Aaron	ATTENDANCE HISTORY

Figure 42 Attendance History Page

To view the attendance history to a player:

1. Click on the “Attendance History” button.



Figure 43 Attendance History Button

When the “Attendance History” button has been clicked, it will bring up the dates that the said player has attended activities.



Figure 44: Player attendance history

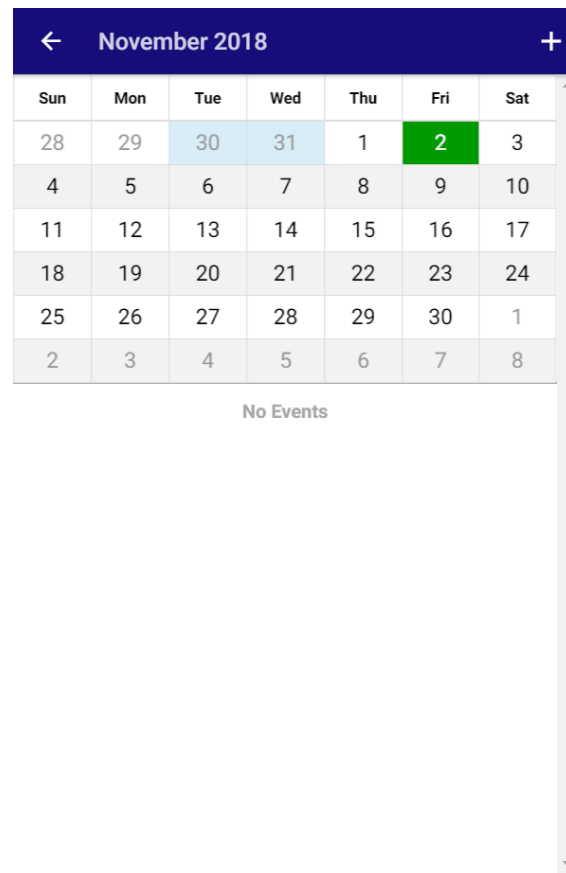
To return to the Home page, click on the back-arrow icon.



Figure 45 Back Arrow

5.5.1.4 Calendar Page:

The purpose of this page is to allow a coach to insert the schedule of the day for the viewing of the rugby player.



November 2018						
Sun	Mon	Tue	Wed	Thu	Fri	Sat
28	29	30	31	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	1
2	3	4	5	6	7	8

No Events

Figure 46 Calendar Page

The date of the calendar being opened will automatically appear.



Figure 47 Real-time date

To add an event to a day:

2. Click on the date required.



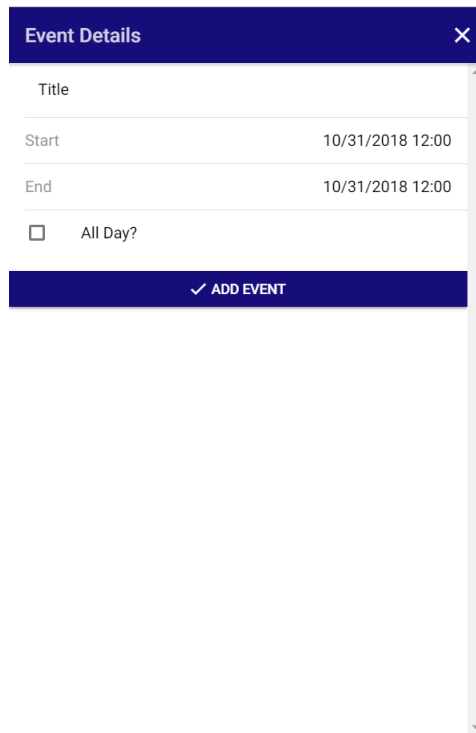
Figure 48 Date required

Thereafter, click on the “add” icon.



Figure 49 Add Icon

Once the “add” icon has been clicked, a coach will be redirected to the “Events Details” page.



The image shows a mobile application interface for adding an event. It features a dark blue header with the text 'Event Details' and a close button (X). Below the header is a form with the following fields: 'Title' (a text input field), 'Start' (a date and time field showing '10/31/2018 12:00'), 'End' (a date and time field showing '10/31/2018 12:00'), and an 'All Day?' checkbox. At the bottom of the form is a dark blue button with a white checkmark and the text 'ADD EVENT'.

Figure 50 Add Event Page

To add an event:

3. Type the title or name of the event;



The image shows a single text input field with the placeholder text 'Title' in a light gray font.

Figure 51 Add Title

4. Select the start date;



The image shows a date and time picker interface. It has a light gray label 'Start' on the left and a date and time '10/31/2018 12:00' on the right.

Figure 52 Start Date Picker

When the date has been clicked upon, a date picker will appear.

			CANCEL	DONE
			09	
01	10			
02	11	2018		
03	12	2017		
04		2016		

Figure 53 Date Picker

1. To save the date click on the “Done” button and
 2. To cancel the date, click on the “Cancel” button.
5. Select the end date and time;

End	10/31/2018 12:00
-----	------------------

Figure 54 End Date Picker

When the date has been clicked upon, a date picker will appear.

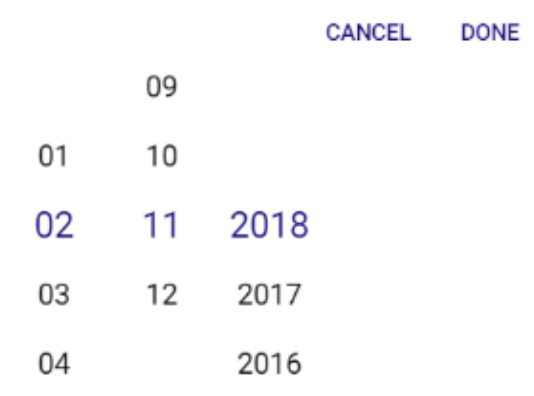


Figure 55 Date Picker

1. To save the date click on the “Done” button and
 2. To cancel the date, click on the “Cancel” button.
-
6. If it is an all-day event, check the “All Day” box and



Figure 56 All Day Event

7. Click the “Add Event” button.
- 8.



Figure 57 Add Event button

To return to the Calendar page and cancel the add event action, click on the “cross” icon.



Figure 58 Cancel icon

When an event has been added to the calendar, the date will turn a light blue colour to indicate to a coach that there is an event occurring on that specific date.



Figure 59 Event saved into date

To view all events occurring on a specific date:

9. Click on the required date.

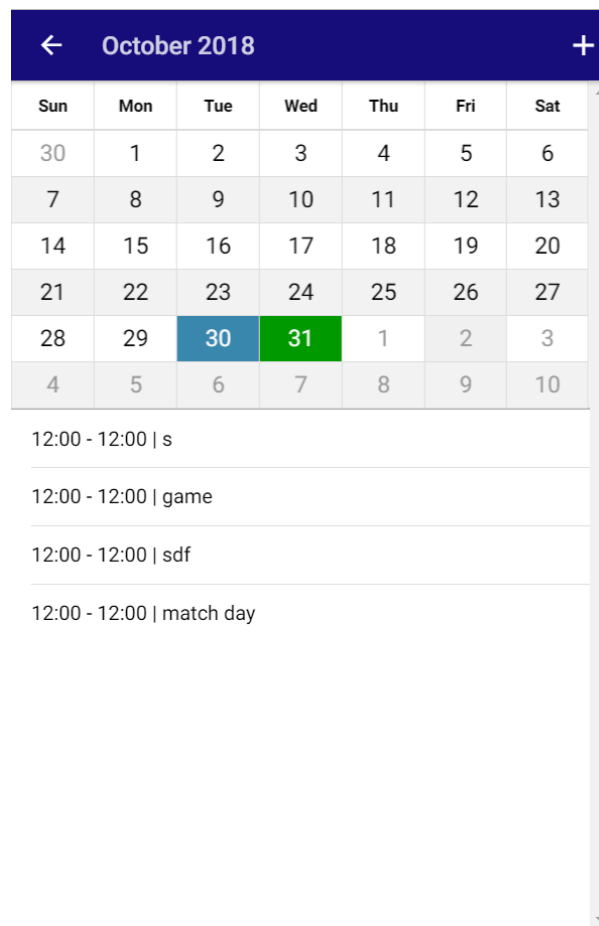


Figure 60 View Events on Date

To return to the Home page, click on the “back arrow” icon.



Figure 61 Back Arrow icon

To view the Match Day page:

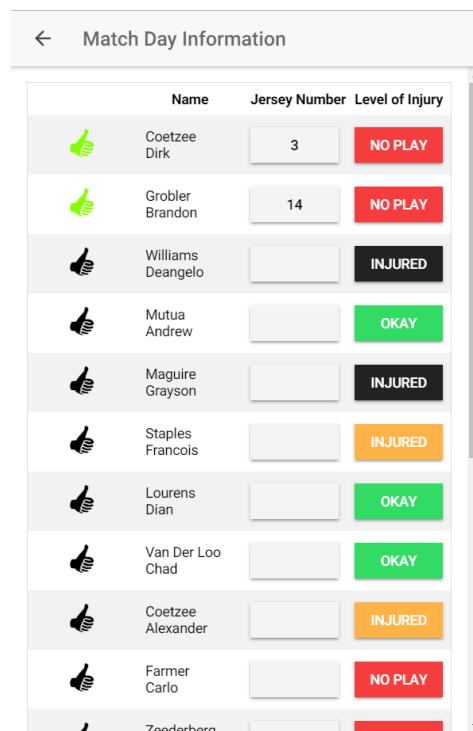
10. Click on the “Match Day” event in the calendar.

12:00 - 12:00 | match day

Figure 62 Match day page link

5.5.1.5 Match Day Page:

The purpose of this page is to insert the jersey numbers of the players for a match.



The screenshot shows a mobile application interface titled "Match Day Information". It contains a table with three columns: "Name", "Jersey Number", and "Level of Injury". Each row represents a player, with a thumbs-up icon to the left of the name. The "Jersey Number" column contains either a text input field or a button with a number. The "Level of Injury" column contains a colored button with text: "NO PLAY" (red), "INJURED" (black or orange), or "OKAY" (green).

	Name	Jersey Number	Level of Injury
👍	Coetzee Dirk	3	NO PLAY
👍	Grobler Brandon	14	NO PLAY
👍	Williams Deangelo		INJURED
👍	Mutua Andrew		OKAY
👍	Maguire Grayson		INJURED
👍	Staples Francois		INJURED
👍	Lourens Dian		OKAY
👍	Van Der Loo Chad		OKAY
👍	Coetzee Alexander		INJURED
👍	Farmer Carlo		NO PLAY
👍	Zeederberg		NO PLAY

Figure 63 Match Day Page

To insert the jersey number of a player:

11. Click on the "Jersey Number" button and insert the jersey number.

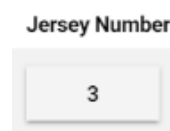


Figure 64 Jersey Number button

To return to the Home page, click on the “back arrow” icon.



Figure 65 Back arrow icon

5.5.2 Physiotherapist

5.5.2.1 Home Page

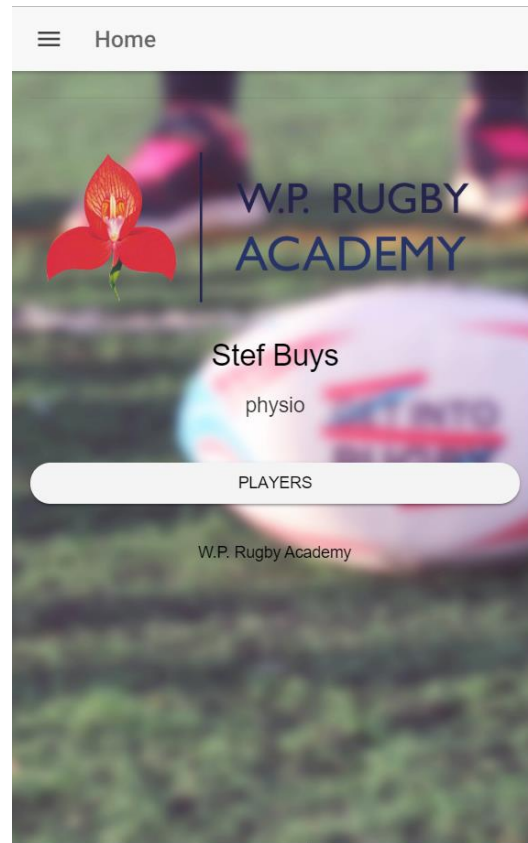


Figure 66 Home Page Physio

The Home Page for the physiotherapist is the landing page after being logged in.

The physiotherapist has an option to redirect to the “Players” page by:

12. Clicking on the “Players’ button.

To view other pages from the Home page:

8. Click on the Menu Toggle Icon.



Figure 67 Menu Icon

Once the menu icon has been clicked, the menu will appear by sliding across from the left side of the mobile application.

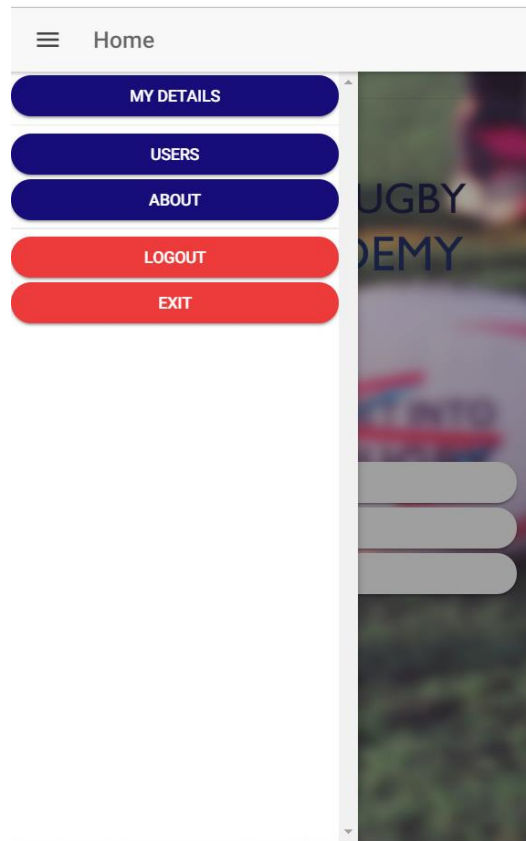


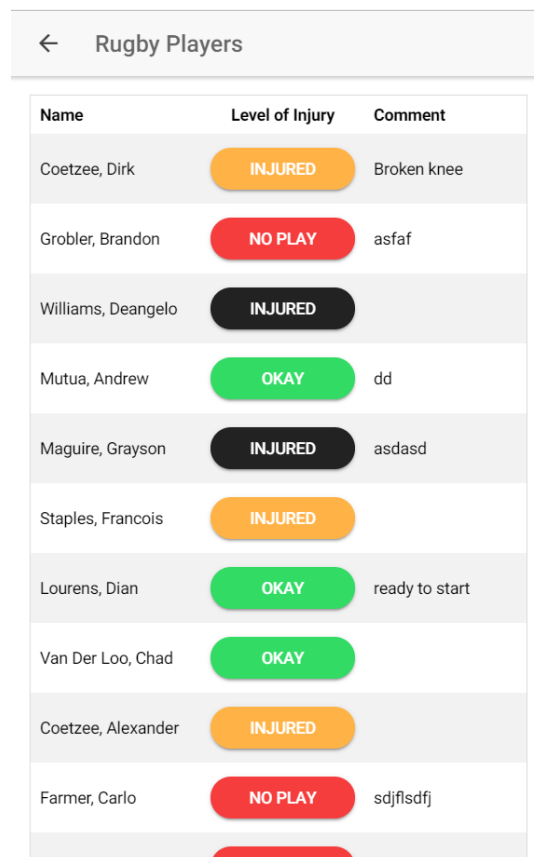
Figure 68 Menu bar

A physiotherapist can click on either button:

9. My Details;
10. Users;
11. About
12. Logout and
13. Exit.

5.5.2.2 Players Page

The purpose of this page is to view all players and insert an injury and recovery date where necessary.



The screenshot shows a mobile application interface for 'Rugby Players'. At the top, there is a back arrow and the title 'Rugby Players'. Below this is a table with three columns: 'Name', 'Level of Injury', and 'Comment'. The table lists ten players with their respective injury statuses and comments.

Name	Level of Injury	Comment
Coetzee, Dirk	INJURED	Broken knee
Grobler, Brandon	NO PLAY	asfaf
Williams, Deangelo	INJURED	
Mutua, Andrew	OKAY	dd
Maguire, Grayson	INJURED	asdasd
Staples, Francois	INJURED	
Lourens, Dian	OKAY	ready to start
Van Der Loo, Chad	OKAY	
Coetzee, Alexander	INJURED	
Farmer, Carlo	NO PLAY	sdjflsdfj

Figure 69 Player list

To insert the level of injury:

1. Click on the “Level of Injury” button.



Figure 70: Level of Injury button

Thereafter, a modal will appear.

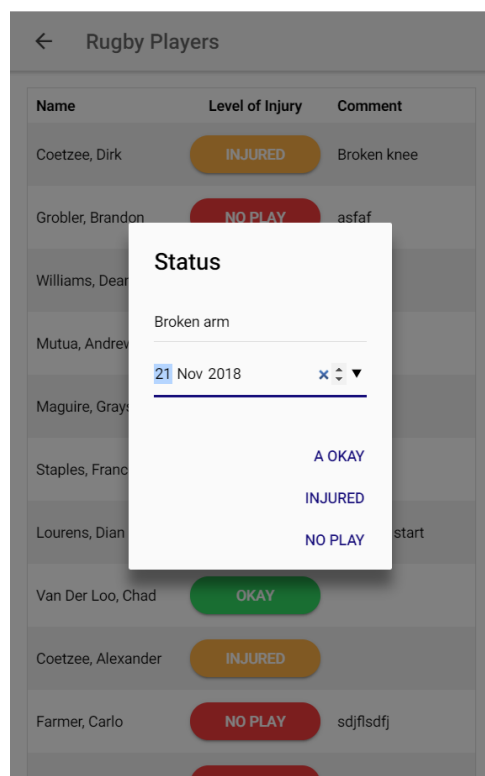


Figure 71 Injury edit details modal

To add a comment about the injury:

2. Click and type in the required text field.

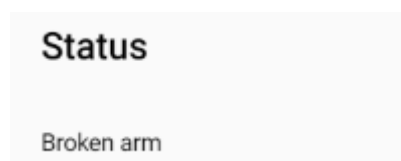


Figure 72: Edit type of injury

To insert the estimated recovery date:

3. Navigate the date arrows or
4. Click on the “down” arrow to display the date picker.



Figure 73 Recovery date picker

To add a quick note about the status of the injury click on either of the following buttons:

1. “A-okay”;
2. “Injured” or
3. “No Play”



Figure 74 Quick note

5.5.3 Coach & Physiotherapist

To view the “Users” page:

13. Click on the “User” button in the menu.



Figure 75 Users button

5.5.3.1 Users Page

The purpose of this page is to view all players, coaches and physiotherapists who are registered on the RugBot mobile application.

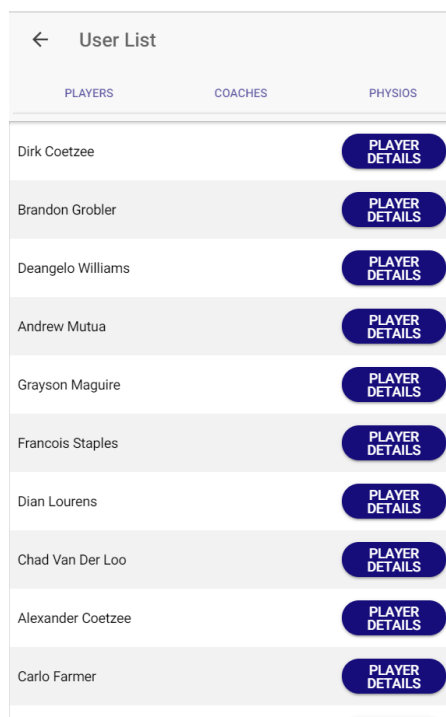


Figure 76 User List page

To view all coaches:

14. Click on the “Coaches” tab.



COACHES

Figure 77: Coaches tab

To view all physiotherapists:

15. Click on the “Physios” tab.



PHYSIOS

Figure 78: Physios tab

A coach has the ability to edit a player’s details.

To edit a player’s details:

16. Click on the “Player Details” button.



PLAYER
DETAILS

Figure 79 Payer Details button

5.5.3.2 Edit Player Details page

The purpose of this page is to allow a coach to edit the personal details of a player.



← User Details

 **W.P. RUGBY
ACADEMY**

W.P. Rugby Academy

First Name Dirk

Surname Coetzee

Email dcoetzee@gmail.com

EDIT DETAILS

CONTACT

DELETE

Figure 80 Edit Players Details page

The way in which you will edit your own details, is the same as editing a player's details. To return to the User list page, click on the "back arrow" icon.



Figure 81 Back arrow icon

5.5.4 All Users

To view the “About” page:

1. Click on the “About” button in the menu.



Figure 82 About button

5.5.4.1 Home Page



Figure 83 Home Page Player

The Home Page for the player is the landing page after being logged in. The player has an option to redirect to the “Calendar” page by:

2. Clicking on the “Calendar” button.

To view other pages from the Home page:

14. Click on the Menu Toggle Icon.



Figure 84 Menu icon

5.5.4.2 About Page

The purpose of this page is to read the background information on the Western Province Rugby Academy.

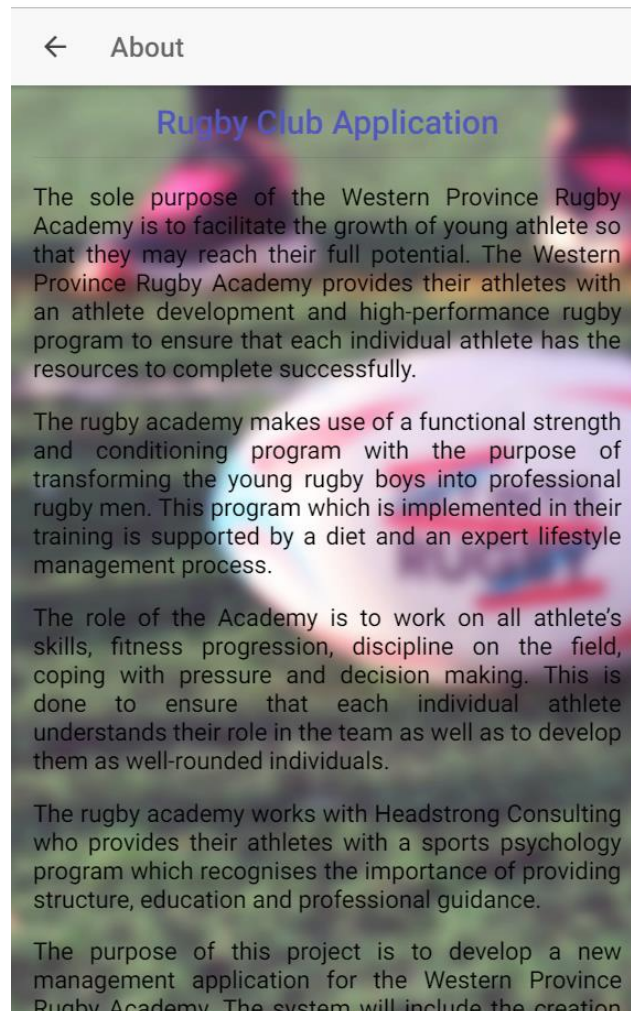


Figure 85 About page

To return to the menu bar, click on the “back arrow” icon.



Figure 86 Back arrow icon

To logout and redirect to the Login page:

3. Click on the “Logout” button in the menu.



Figure 87 Logout button

To exit the mobile application but not logout:

4. Click on the “Exit” button.



Figure 88 Exit button

5.5.4.3 Calendar Page

The purpose of this page is to allow a player to view the daily schedule and events.

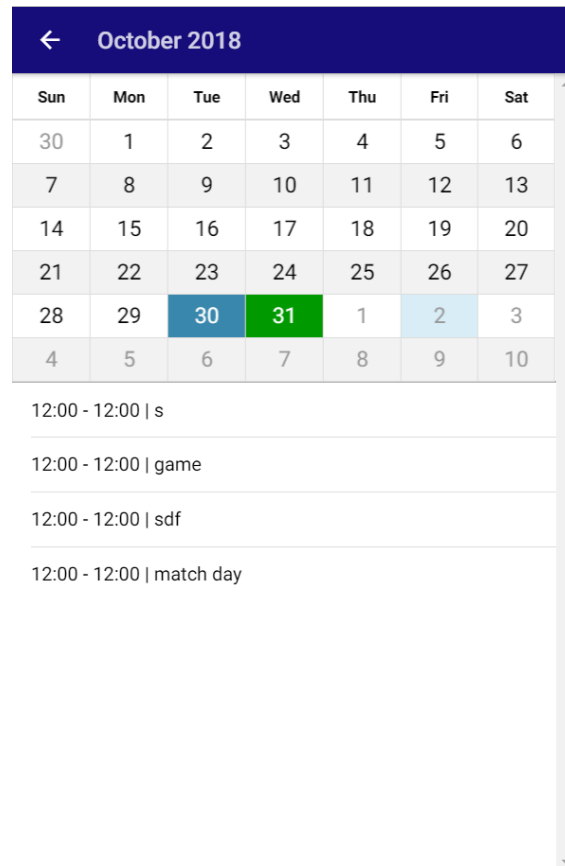


Figure 89 Calendar

To navigate to the “Match Day” page:

15. Click on the “match day” event.

12:00 - 12:00 | match day

Figure 90 Match Day event

Once the match day event has been clicked, the player will be redirected to the “match day” page.

5.5.4.4 Match Day Page

The purpose of this page is to allow a player to view their individual jersey numbers for a match.












← Match Day Information			
	Name	Jersey Number	Level of Injury
	Coetzee Dirk	3	NO PLAY
	Grobler Brandon	14	NO PLAY
	Williams Deangelo		INJURED
	Mutua Andrew		OKAY
	Maguire Grayson		INJURED
	Staples Francois		INJURED
	Lourens Dian		OKAY
	Van Der Loo Chad		OKAY
	Coetzee Alexander		INJURED
	Farmer Carlo		NO PLAY
	Zeederberg		NO PLAY

Figure 91 Match Day Page

A player will not be allowed to edit the Jersey Numbers.

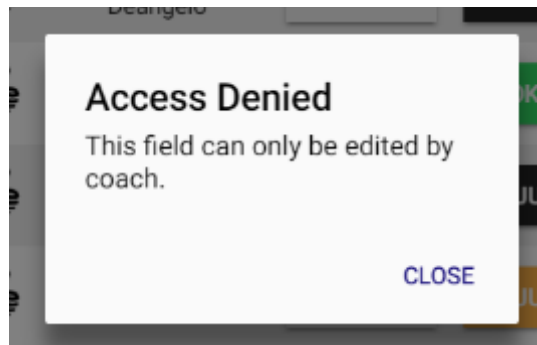


Figure 92 Access Denied

5.5.4.5 User List Page

The purpose of this page to allow a player to view all team mates, coaches and physiotherapists.

<div> <div>←</div> <div>User List</div> </div>	
PLAYERS	COACHES
PHYSIOS	
Dirk Coetzee	
Brandon Grobler	
Deangelo Williams	
Andrew Mutua	
Grayson Maguire	
Francois Staples	
Dian Lourens	
Chad Van Der Loo	
Alexander Coetzee	
Carlo Farmer	
Aaron Zeederberg	
Alex Retilainen	
Brendan Lotter	
Cameron Moodie	
Aden Luke Masoling	
Jayson Dyers	
Byron Botha	
Stef Buys	
Cameron Pope	

Figure 93 User List

5.5.4.6 User Details Page

←

User Details



W.P. RUGBY
ACADEMY

W.P. Rugby Academy

First Name

Stef

Surname

Buys

Email

stefbuys21@gmail.com

EDIT DETAILS

CONTACT

DELETE

Figure 94 User Details page

To edit personal details required:

17. Click and insert on the “First Name” text field;

A rectangular text input field with a blue border. The label "First Name" is on the left in blue text, and the value "Stef" is on the right in black text.

Figure 95 First Name text field

18. Click and insert on the “Surname” text field;

A rectangular text input field with a blue border. The label "Surname" is on the left in blue text, and the value "Buys" is on the right in black text.

Figure 96 Surname text field

19. Click and insert on the “Email” text field and

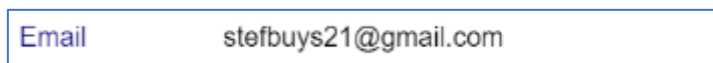
A rectangular text input field with a blue border. The label "Email" is on the left in blue text, and the value "stefbuys21@gmail.com" is on the right in black text.

Figure 97 Email text field

20. Click on the “Edit Details” button to save changes made.

A dark blue rectangular button with the text "EDIT DETAILS" in white, uppercase letters.

Figure 98 Edit Details button

To Contact the RugBot development team:

21. Click on the “Contact” button.

A dark blue rectangular button with the text "CONTACT" in white, uppercase letters.

Figure 99 Contact button

To delete the user account:

22. Click on the “Delete” button.



Figure 100 Delete button

To return to the menu bar, click on the “back arrow” icon.



Figure 101 Back arrow icon

6 Evaluation

The RugBot development team was tasked with the creation of the RugBot mobile application. The aim of the application was to develop a system to manage and support the record keeping of the Western Province Rugby Academy.

The following report will act as a final evaluation of the project. The report will contain an evaluation of the project, commentary of the design methodology, an evaluation of customer involvement, commentary of group involvement, time management review and personal thoughts of each group member.

6.1 The final system & Customer Requirements

The final outcome of the system was what the RugBot development intended it to be. The system has met all customer requirements in an efficient manner.

Concerning the design point of view, the RugBot application is user-friendly with a clean minimalistic design for ease of application. The application is designed in such a way that every intended user has the ability to make use of the application without any difficulty or confusion.

In relation with the function aspect of the system developed, RugBot has met all functional requirements intended for the application. All functionality operates the way it is intended to and operations occur without any delay time.

6.2 List of Functional Requirements

The table below displays functional requirements in order of priority.

Table 39 Functional Requirements

Identifier	Requirement Description	Priority	Source	Status
FR01	Users must use a one-time login to log in to the application for authorisation purposes.	High	RugBot Development Team	Met
FR02	Coaches must be able to take an attendance list of students at practice.	High	Coaches	Met
FR03	Coaches must be able to view a backlog of student's attendance for past dates.	High	Coaches	Met
FR04	Coaches must be able to view a list of all their students and their availability for practise sessions and matches.	High	Coaches	Met
FR05	Coaches and students must have a calendar with a practise match dates and times.	High	Coaches	Met
FR06	The physiotherapist must be able to mark a student as injured and not able to practise or play matches.	High	Physiotherapist	Met
FR07	The physiotherapist must be able to add an estimated date of when a student will be able to practise again.	High	Physiotherapist	Met
FR10	The coach must be able to assign jersey numbers to players on match dates.	High	Coaches	Met
FR11	All users need to be able to see the injury status of a player.	High	Coaches and Physiotherapist	Met
FR13	Coaches need to be able to schedule a match.	High	Coaches	Met
FR14	Coaches and students need to be able to see match teams.	High	Coaches and Player	Met

Identifier	Requirement Description	Priority	Source	Status
FR18	Players must be able to update their personal information	High	RugBot Development Team	Met
FR19	Coaches must be able to update their personal information	High	RugBot Development Team	Met
FR20	Physios must be able to update their personal information.	High	RugBot Development Team	Met
FR21	Coaches must be able to add and remove players.	High	RugBot Development Team	Met
FR09	The coach must be able to see the total of boys at practice.	Medium	Coaches	Met
FR12	Coaches and physio must be able to view the medical history of a player. A player must only be able to see their own data.	Medium	RugBot Development Team	Met
FR15	Players must be notified when they are playing games.	Low	RugBot Development Team	Met

6.3 Iterative & Incremental Development in Agile

The RugBot development made use of the Agile Methodology, making use of iteration and incremental development style to produce the RugBot application. When compared to more traditional development methodologies, like those suggested by Sommerville (2001) and Pressman & Maxim (2015), there are clear advantages in making use of an agile development methodology (Schwalbe, 2012). Agile supports four main principles for developers, namely:

1. Customer collaboration;
2. Response to client requirement changes;
3. Software development over documentation; and
4. Interactions over processes and tools (Sacolick, 2018).

The term incremental is defined as the ability to add new functionality in small parts. This allows developers to focus on perfecting sections of a system simultaneously. The advantage of this methodology is to allow developers to conduct testing after increments have been completed. This allows bugs to be fixed during the early stages of development (Ghahrai, 2018).

The term Iterative is defined as adding new functionality in a repetitive manner. This allows developers to stick to a constructive time management schedule and ensures that deliverables are completed when due (Ghahrai, 2018).

By making use of this methodology, developers are exposed to learn how to grow from the initial and early sections of the work which has been completed. Learning objectives could include better time management skills, communication amongst team members and work ethic in general. It allows developers to learn and evolve from their mistakes (Ghahrai, 2018).

The agile methodology was the best-suited method to mobile development (Pearlson & Saunders, 2013), such as the RugBot application. The application contains three users with various amount of different functions for each user. By making use of this methodology, it allowed the RugBot development team to work on each user separately but simultaneously and eventually cohesively bringing together the mobile application as one working software.

6.4 Customer Evaluation

The RugBot development team had an open and clear communication pathway with our customer. Angelo Nelson is a coach for the Western Province Rugby Academy and has been our correspondent during the development of the RugBot mobile application.

In the beginning stages of development, namely planning and gathering requirements, we had regular meetings with our customer to ensure that they knew exactly what they wanted the application to achieve. The reason for conducting regular meeting was to ensure that we as developers had a clear understanding of our customer and the business, to development exactly what they had envisioned.

The Western Province Rugby Academy is situated on the same premises as our development team's working space. We had constant access to our customer at all times. We also had our customer on the WhatsApp mobile application which made instant messaging frequent and easy for both parties. If there was a situation where any party was no sure about an aspect of the application being developed, it was resolved and cleared immediately.

During meetings, the RugBot team recorded what was verbally stated to ensure that there was no miscommunication between developers and the customer. The customer has expressed concerns where he did not initially have a clear idea of what was needed and wanted from the mobile application but the RugBot team provided ideas and concepts that allowed the customer to tell us exactly what was wanted.

Overall, the customer felt comfortable with the RugBot team and new that what he wanted was going to be delivered. Communication between the customer and the RugBot team was constantly open which resulted in a good customer relationship situation.

6.5 Group Dynamics and Team Collaboration

The RugBot development team performed well as a cohesive unit. Communication was always kept open during the development process of the RugBot application. A WhatsApp group was created to ensure instant messaging was possible. When it came to sharing information, documents and code, GitHub and email were used.

All group members had a chance to browse through every deliverable to ensure that all work was correctly conducted. There was always room for input and advice to be given.

6.6 Time Management

Time management is always a difficult task to accomplish successfully, especially when individuals are faced with numerous other tasks simultaneously with the one at hand. With that being said, during the development of the RugBot mobile application, every individual in the team had their own time to consider against other subjects and their own personal lives.

During the initial stages of development, the RugBot team was on top of time. All work is completed for deliverables was done well ahead of time. However, when it came to the actual development of the mobile application, time started to creep upon the team. One of the main reasons for this was the frustrating task of learning how to deal with the framework at hand namely; Ionic Framework.

Ionic was fairly new to all members of the group and therefore simple tasks seemed to take up more time than expected. You would assume a task would take a certain amount of time but at the end realising it has taken more time than expected because of the constant learning of how to use the framework.

Another factor that contributed to the time creep up was all the other work which needed to be done simultaneously to the development of the mobile application. It would be fair to say that the final semester of the final year in our studies has been the most workload every team member has dealt with.

Overall, the RugBot development team had fairly good time management when it came to development, communication and contribution. However, we lacked when identifying which tasks needed more concentration and therefore found ourselves scrunching for time to complete certain functionality.

6.7 Lessons Learnt

6.7.1 Stefanus Buys

Developing the Rugbot application has been a challenge, although mostly a good one. It was the first time our group members collaborated and getting to know each other has been interesting. I am quite introverted, and communication is not my strongest skill. I learnt the importance of keeping in contact with my team members during development. We had our differences, but we were successful in resolving personal issues. I learnt that having a leader and knowing one's place is important in the development process. There was quite a lot of technical skills that I picked up during the development of the application, but the most important lessons I learnt are about group dynamics and relationships within a professional environment.

6.7.2 Tyla Grey

The RugBot development team decided to make use of the Ionic framework to develop the mobile application. During the course of the year, I have learnt many new languages and frameworks that needed to complete certain modules. Ionic has been the most annoying and difficult to get comfortable with. I found myself constantly struggling to get functionality to work and design to view the way in which I wanted it to. However, this does not mean that I did not learn anything. Through the struggle, I have managed to get used to the Ionic framework and deliver what I needed to. It was frustrating but that is where patience was learnt to. Also, I learnt to ask for input and suggestion were needed to ensure that I understand every aspect of the mobile application.

Time management was not conducted in the way I had liked it to be. All work which my responsibility was completed when needed to be, but I would have liked it to be completed much early than required. This was due to the workload of the entire semester.

Working in a group is always difficult due to every individual's different work ethic. I have realised that individuals proceed with completing tasks in different ways that I would. Therefore, I have learnt that communication is a key component in group work. When confused about a specific task or statement; clarify what you are confused about, when needing to know something specific about anything during development; communicate by asking group members and when miscommunication occurs; communicate your thoughts and opinions to your team members.

Overall, I do believe that every team member had a lot of learning to accomplish when building this application and struggled through frustration. However, in the end, we managed to complete all tasks required with a satisfactory end product.

6.7.3 Abongile Mdleleni

In this group project, I have learnt quite a few things through the development of our App. It has been a rollercoaster of emotions and a great learning experience. I have learned to use Ionic Framework, ionic is a cross-platform development tool used to develop phone applications. In our app, we have used frameworks, plugin and hard-coded some of the functionality in the app. I have learned to code in TypeScript, we are using it as the main programming language in our development as well as HTML and SASS. Since the beginning of development, it has been an interesting journey facing multiple challenges. Debugging abilities have grown quite a bit, with each successful completion and testing we integrate a new feature which breaks everything. That requires countless hours to fix before moving on to the next task. This repetition of events has sharpened my debugging abilities.

I have been reminded of the importance of time management and teamwork, with everyone doing their part has allowed progress to be much quicker and swifter as we achieve more by working together. I have also learnt how to use Firebase, which is what we are using for our database. It has been quite an experience with some challenges as a first-time user being and being nominated as one of the database administrators for this project. I am grateful for the nomination, challenging it is but I have gained a new skill in the process.

6.7.4 Matthew Van der Bijl

Frameworks are more trouble than they are worth. This is something that I had heard of but never fully understood until I started to work on this project. Compared to last year's project where native Android was used, this year was a significant jump in complexity and energy required to complete simple tasks. I feel that the framework was not only not helping me develop the system but actively preventing me from doing so. Don't take this point as me not wanting to learn a new system only that the amount of learning required was disproportional to the task that needed to be completed.

As with all projects, time management could have been better. I am happy that I managed to stay on top of my other work which allowed to put more work into the projected when needed. I feel that the project will be completed within the allocated time and the client will be satisfied. In conclusion, I learnt a lot working on this project.

7 Bibliography

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

Botha, S. & Musengi, S., 2012. *Introduction to Business Management*. Cape Town: Pearson.

Buttrick, R., 2009. *The Project Workout*. 4th ed. Harlow: Prentice Hall.

Chung, L., Nixon, B. A. & Yu, E. S. K., 1996. Dealing with change: An approach using non-functional requirements. *Requirements Engineering*, 1(4), pp. 238-260.

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

Deitel, P. & Deitel, H., 2012. *Java : How to Program*. 9th ed. Boston: Prentice Hall.

Drucker, P., 1975. *The practice of management*. London: Heinemann.

Ghahrai, A., 2018. *Difference Between Iterative and Incremental Development in Agile*.

[Online]

Available at: <https://www.testingexcellence.com/iterative-incremental-development-agile/>

[Accessed 15 October 2018].

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](https://developer.salesforce.com/page/Native,_HTML5,_or_Hybrid:_Understanding_Your_Mobile_Application_Development_Options)

[Accessed 31 July 2018].

Nielsen, J., 1994. Heuristic evaluation. In: *Usability Inspection Methods*. New York: John Wiley & Sons, pp. 25-62.

Pearlson, K. E. & Saunders, C. S., 2013. *Managing & Using Information Systems A Strategic Approach*. 5th ed. New Jersey: John Wiley & Sons, Inc.

Preece, J., Rogers, Y. & Sharp, H., 2015. *Interaction Design*. 4th ed. Chichester: Wiley & Sons Ltd.

Pressman, R. S. & Maxim, B. R., 2015. *Software Engineering a Practitioner's Approach*. 8th ed. New York: McGraw-Hill Education.

Pretorius, C. M. & Erasmus, G. H., 2012. *Basic Programming Principles*. 2nd ed. Cape Town: Pearson Education South Africa.

Project Management Institute, 2013. *A Guide to the Project Management Book of Knowledge*. 5th ed. Pennsylvania: Project Management Institute.

Sacolick, I., 2018. *What is agile methodology? Modern software development explained*. [Online]

Available at: <https://www.infoworld.com/article/3237508/agile-development/what-is-agile-methodology-modern-software-development-explained.html>

[Accessed 15 October 2018].

Schwalbe, K., 2012. *Information Technology Project Management*. 7th ed. Boston: Cengage Learning.

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

Stair, R. M. & Reynolds, G. W., 2016. *Principles of Information Systems*. Boston: Cengage Learning.

Testing Excellence, 2018. *Types of Software Testing – Complete List*. [Online]

Available at: <https://www.testingexcellence.com/types-of-software-testingcomplete-list/>

[Accessed 20 July 2018].

Valacich, J. S., George, J. F. & Hoffer, J. A., 2015. *Essentials of Systems Analysis and Design*. Global Edition ed. New Jersey: Prentice Hall.

Weisfeld, M., 2013. *The Object-Oriented Thought Process*. 4th ed. Boston: Addison-Wesley.

Whitman, M., Coles, M. & Mattord, H., 2012. *Principles of information security*. 4th ed. London: Cengage Learning.