Real-Time Online Scoreboard

Project Proposal

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Team Members: | Hayley-Belle Cleverdon  | Vinicius Alves Ferreira | Karanjit Gahunia  | Seung-Kyu Jin  | Alex Lu |  | Client:  |Dr. Robin Hankin  Supervisor:  |Dr. Stephen Thorpe |  | 28th March 2017  Version 1.0 |

Table of Contents

1. [Terms of Reference 2](#_Toc478503969)

[Project Objective](#_Toc478503970)

[Project Opportunity](#_Toc478503971)

1. [Rationale 3](#_Toc478503972)

[Rationale for the Project](#_Toc478503973)

1. [Scope and Objectives 4](#_Toc478503974)

[Project Scope](#_Toc478503975)

[Project Objectives](#_Toc478503976)

[Stakeholders](#_Toc478503977)

1. [Project Approach 8](#_Toc478503978)

[Methodology](#_Toc478503979)

[Methodology Rationale](#_Toc478503980)

[Communications Management Plan](#_Toc478503981)

1. [Project Plan 15](#_Toc478503982)

[Work Breakdown Structure](#_Toc478503983)

Provisional Milestone Plan

[Network Diagram](#_Toc478503986)

[Gantt Chart](#_Toc478503987)

[Justification](#_Toc478503988)

Project Phases

[Risk Management Plan](#_Toc478503985)

1. [Skills and Knowledge Involved 21](#_Toc478503989)

[Skill Matrix](#_Toc478503990)

1. [Estimated Costs 23](#_Toc478503991)
2. [Disclaimer 24](#_Toc478503992)
3. [References 24](#_Toc478503993)
4. [Appendix 25](#_Toc478503994)

[Appendix A](#_Toc478503995)[\_Toc478503998](#_Toc478503998)

[Appendix B](#_Toc478503999)

[Appendix C](#_Toc478504000)

[Appendix D](#_Toc478504001)

[Team Agreement](#_Toc478503984)

# Terms of Reference

**Organisation**

The Auckland Mathematical Association (AMA) is an organisation whose goal is to promote, support and improve the mathematical education in the Auckland region. Every year AMA holds a mathematics competition called Casio MATHEX. The event is aimed at year 7-10 students in which about 100 teams of 4 students compete for prizes. The event is usually hosted at ASB Stadium, Kohimarama and is very well attended.

**Client**

Dr Robin Hankin is a senior lecturer in Mathematical Sciences at the Auckland University of Technology (AUT) and is the client for this project. He has requested for us to build a new scoreboard system for MATHEX. He is expecting this project to promote and associate the AUT brand with Mathematical education within Auckland.

**Problem**

Currently, the scoring system follows the traditional use of pen-and-paper by judges to keep track of the scores for each team. To display scores to the spectators at these events, there is a paper attached to the wall behind the judges for each team. As the competition progresses, these papers are updated to reflect each team's score. Due to this, it is difficult to locate any particular team's score and compare it to the rest of the competition. This affects the experience that the participants and spectators have at these MATHEX events negatively.

## Project Objective

The objective of this project is to create a digital, online scoring system and display board suitable for MATHEX competitions. The system should be easily viewable for any person with an internet capable device such as a smartphone, tablet, or laptop.

The system will consist of 3 major components. The first being the backend, a database that hosts all of the information. Secondly, an interface that approximately 100 judges can use to update the scores within the database. And lastly, an interface for all users to view the scoreboard, search and track for their teams and any other information they need to know.

## Project Opportunity

The current situation of the scoring system had several drawbacks which created the interest for a real-time scoreboard system.  A switch to a digital online scoring system and display board would effectively improve the audience experience, while aiding the organizers in keeping the tournament smooth and pleasant. Also, a real-time online scoring system would allow the judges to inform the results on-time, so that the competitors can also stay updated throughout the tournament, which we envisage will generate an exciting and competitive atmosphere.

# Rationale

## Rationale for the Project

Currently, a paper based system is used to track and display scores during MATHEX events. Using a paper based system does not create any issues for the judges, as it is reliable and efficient. However, participants and spectators have significant difficulty in viewing any particular team’s score, as well as comparing teams overall to see who is winning. This limits the viewing experience for the spectators which consequently makes MATHEX events less fun for both the spectators and the participants.

By developing a real-time scoreboard system that follows a Bring-Your-Own-Device (BYOD) model, these issues could be solved. A simple application for judges to use alongside their existing paper judging system would allow spectators to follow the event closely as well as providing contestants a way to compare their progress with the rest of the competition.

Implementing this in MATHEX events would provide opportunities such as:

* Promoting MATHEX and mathematics education within Auckland.
* Promoting the brand of Auckland University of Technology.
* Promoting our own abilities as developers, for future career opportunities.

Making the competition more engaging for spectators and fun for competitors would create more interest in MATHEX competitions, as well as mathematics in general. This could lead to larger MATHEX competitions with more schools participating, more frequently held competitions, or potentially a national MATHEX competition with teams from different parts of New Zealand that qualify through their regionally held MATHEX events.

# Scope and Objectives

## Project Scope

The scope of this project includes but is not limited to a feasibility study for the product in proposal. The feasibility study should state whether the implementation of proposed software, including infrastructure required, is achievable. The feasibility study should address scope, time, and costs.

The product to be proposed has, as main goals, to track the judges’ responses and display the scores online for audience members.

The scope includes two major sections:

1. **Project evaluation:**

* A roadmap for the project which details what functionality the proposed system will have, how the system will be implemented (in terms of both software and hardware), and what procedures will be followed to implement the system. The roadmap should be easily understood and highly detailed so if the project is handed off to another team in the future, there will be sufficient information for them to be able to smoothly transition into it.
* Provide options to the solution domain.

1. **Project Deliverables:**

* A real-time scoreboard application/website (Software) which can be run on a wide variety of commonly used portable devices such as smartphones, tablet, and laptops.
* Hardware which can set up a local Wi-Fi network to connect a variety of devices to a database. The range of this box should be able to encompass the whole of the ASB Stadium sports venue and connect to at least 400 devices at once.
* Feature list to be extracted from the requirements. Please refer to the project objective.

If the feasibility study reveals that the project cannot be completed as per the client's requirements, our team will create a software prototype to deliver. This prototype will endeavour to showcase as many of the client's required features as possible. The prototype will serve as a framework, or at the least a design example for any future teams who undertake this project.

We expect to develop a web-based application that communicates with a database for this prototype. The application will contain the core 3 components, but may not be able to support requests from as many as 500 users.

*\**Note: The project scope may be altered through the evaluation of the project feasibility study. The client, Dr. Robin Hankin, will decide based on the feasibility study.

## Project Objectives

Below are a set of core, expected system requirements, translated from the description that the client had given us in our first meeting.

|  |  |
| --- | --- |
| **Functional requirements:** |  |
|  | **Overall System:** |
| 1.1 | The system infrastructure must run on a Windows PC. |
| 1.2 | The system must be accessible to users through most mobile devices. |
| 1.3 | The system software must be released on GPL or another open-source license. |
| 1.4 | The system must be able to handle requests from 500 or more users at one time. |
| 1.5 | The system must have a database. |
| 1.6 | The system must have a page with higher user privileges for judges. |
| 1.7 | The system must have secure login credentials for judges. |
| 1.8 | The system must have a page with view only privileges for audience members. |
| 1.9 | The system must run in real-time. |
|  |  |
|  | **Database Functionality:** |
| 2.1 | The system must store the list of teams. |
| 2.2 | The system must be able to store input from a judge. |
| 2.3 | The system must be able to update correct answer scores for each team. |
| 2.4 | The system must be able to update incorrect answer scores for each team. |
| 2.5 | The system must be able to update number of passes for each team. |
| 2.6 | The system must organise teams by their correct answer scores, minus incorrect answer scores, numerically. |
| 2.7 | The system must organise teams alphabetically by name. |
| 2.8 | The system must be able to search for a specified team and display the correct scores for that team. |
| 2.9 | The system must be able to access all teams and their scores simultaneously. |
| 2.10 | The system must store login IDs for judges. |
| 2.11 | The system must store passwords for judges. |
| 2.12 | The system must give edit privileges to judges when they input the correct login ID and password combination. |
|  |  |
|  | **Interface Functionality:** |
| 3.1 | The system must have text box inputs for: |
| 3.1.1 | Judge login ID. |
| 3.1.2 | Judge login password. |
| 3.1.3 | Search for team. |
| 3.2 | The system must have a submit button that sends the data from the text boxes to the database. |
| 3.3 | The system must return a searched team when the team name is entered correctly. |
| 3.3.1 | The system must be able to search without case-sensitivity. |
| 3.3.2 | The system must display an error message under the text box when the input returns no results. |
| 3.4 | The system must display the following upon returning a team: |
| 3.4.1 | The team name. |
| 3.4.2 | The team’s numerical position on the leaderboard (i.e. 1st, 2nd etc). |
| 3.4.3 | The team’s scores. |
| 3.5 | The system must have input buttons for judges as follows: |
| 3.5.1 | A ‘correct’ button. |
| 3.5.2 | An ‘incorrect’ button. |
| 3.5.3 | A ‘pass’ button. |
| 3.5.4 | Buttons must be time-delayed so the judge must hold the button down for at least 2 seconds before input is sent to the database. |
| 3.5.5 | Buttons must display a confirmation when data is sent to the database. |
| 3.5.6 | Buttons must deactivate when data is sent. |
| 3.5.7 | Buttons must reactivate when the user lifts their finger/cursor off the button. |
| 3.6 | The system must display all of the teams’ scores on a leaderboard. |
| 3.6.1 | The system must show the scores in a graphical format (bars). |
|  |  |
|  | **Non-functional requirements:** |
| 1. | The system must be robust and maintainable. |
| 2. | The system must be well documented, to allow another team to continue working on it if necessary. |
| 3. | The system must be easy to use. |
| 4. | The system must be scalable. |
| 5. | The system must display the AUT logo on the interface at all times where possible. |

## Stakeholders

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Position** | **Internal/External** | **Project Role** | **Contact Information** |
| Robin Hankin | Senior Lecturer in Mathematical Sciences at AUT | Internal | Client | robin.hankin@aut.ac.nz |
| Kerri Spooner | Senior Lecturer in Mathematical Sciences at AUT | Internal | Client | kspooner@aut.ac.nz |
| MATHEX Spectators | Spectators at MATHEX competitions | External | Product Users | N/A |
| MATHEX Contestants | Students competing in MATHEX competitions | External | Product Users | N/A |
| MATHEX Judges | Judges for MATHEX competitions | External | Product Users | N/A |
| Nikola Kasabov | Professor of Computer Science and Director KEDRI | Internal | Project Supervisor | nkasabov@aut.ac.nz |
| Akshay Raj Gollahalli | Teaching Assistant at Auckland University of Technology | Internal | Technical Advisor | akshay.gollahalli@aut.ac.nz |
| Stephen Thorpe | Head of External Relations and Development for the School of Computer and Mathematical Sciences at AUT. | Internal | Project Moderator | sthorpe@aut.ac.nz |
| Auckland Mathematics Association | Professional Association for Auckland Mathematics Teachers. | External | Sponsor | 027 203 4479 |
| AUT | Auckland University of Technology. | Internal | Sponsor | 09 921 9999 |
| Alex Lu | AUT BCIS Student | Internal | Team Member | xvh7419@aut.ac.nz |
| Hayley Cleverdon | AUT BCIS Student | Internal | Team Member | hayleycleverdon@gmail.com |
| Vinicius Alves Ferreira | AUT BCIS Student | Internal | Team Member | Vini.vaf92@gmail.com |
| Seung-Kyu Jin | AUT BCIS Student | Internal | Team Member | sk.jin@live.com |
| Karanjit Gahunia | AUT BCIS Student | Internal | Team Member | karanjit.gahunia@gmail.com |

# Project Approach

## Methodology

**Feature Driven Development (FDD)** is an agile and adaptive methodology that covers the design and build phases of the software development process. It is highly iterative and is tailored towards projects that expect rapid change during development and that are focused on what the client values. FDD consists of iterations that take requested features individually, then plan, design and build them.

**Feature Driven Development**

Feature Driven Development consists of 5 steps, the final two steps are repeated for the lifetime of the system’s design and build phase, the steps are listed below.

**Develop an Overall Model:**This step assumes the scope, context and requirements of the system are known, and therefore some documented requirements should already exist. During this step the team will “walkthrough” the system with a high-level description and begin to break down the project into domains to be modelled. (Abrahamsson, Salo, Ronkainen & Warsta, 2002) We will also complete a UML diagram of the expected system so that classes are identified. In our case, these will likely be individual scripts. During this stage a rough concept of the visual design will be completed.

**Build a Features List:**Once the domains are modelled, we will be able to build a comprehensive list of features that are valuable to the client (Abrahamsson et al., 2002). These will be Minimal Marketable features. Any major features that will take longer than 2 weeks to complete are further broken down into smaller features or feature sets, and written up as user stories. And the end of this step we will have the client and our supervisor review these for validity and completeness.

**Plan by Feature:** With a completed features list, we will then organise the features into a sequence per their dependencies and priority. The classes identified in the model will be assigned to team members and milestones for feature completion will be set so that we can report these to the client (Abrahamsson et al., 2002).  We also want to adopt Kanban’s workload control, by setting a limit on how many features may be in progress at any one time, to ensure that a balanced workflow is achieved (J. Buchan, personal communication, April, 2016).

**Design and Build by Feature:**A small number of features is selected to be designed, any features that rely on visual design will be fully designed and clear, so that it is ready for the build process. Each feature will have a high-level description of how it should work, so that during coding, clear steps may be followed.

Once the design for a feature is complete, the assigned developer builds it and tests it for functionality. It can be removed from the workflow once it passes the user story requirement.

These processes are iterative, and may be repeated on the same feature once it is built, should changes be needed or requested (Abrahamsson et al., 2002).

FDD has a set of specific roles, we have elected to use a selection of these roles, customising for our project’s needs regarding complexity and considering the size of our team. The roles, adapted from Palmer and Felsing’s (2002) list are below.

**Project Manager:**Administrative leader of the project. Aids the team by shielding them from outside distractions and provide the working conditions. The Project Manager is responsible for arranging meetings (time and location), the team’s budget, as well as communicating with people who hold no stakes in the project, should there be a situation in which information is required. This project manager differs slightly from the traditional FDD manager, as we do not have control over all project costs or the schedule.

**Chief Architect:**Lead designer, is responsible for the overall design of the system and meeting with the team to discuss design decisions. Any design issues are the final say of the Chief Architect.

**Development Manager:** Leads daily development activities, delegates tasks when needed and resolves team conflicts. Additionally, the Development Manager is responsible for any resourcing issues.

**Build Engineer:**Sets up, maintains and runs the build process. The Build Engineer also manages version control and publishing of documentation.

**Programmer:**Is not specific to FDD, but is rather an amalgamation of several roles that may or may not be needed. All team members may be programmers, if the feature list allows for it, or there may only be one, quite simply, a programmer builds the system in code.

**Tester:**Verifies the system is meeting the requirements outlined, this task will be taken on by all programmers.

**Technical Writer:**FDD calls for user documentation, but any edits to planning documentation, meeting summaries or any required documentation not including the assessment artefacts are the responsibility of the Technical Writer. Many members may be technical writers, but one lead writer should be forwarded all documents to be submitted to the project manager.

Feature Driven Development has a few core practices; considering the experience and size of the team, as well as the complexity of the project we have adapted the practices list as follows:

**Domain Object Modelling:**Problem exploration and identification, which leads to building a framework on which to build the features list.

**Developing by Feature:**Progress is made through developing and tracking a list of small features and feature sets that are valued by the client.

**Inspection:** The team will use their best-known defect-detection mechanisms. This will involve rigorous testing and frequent communication.

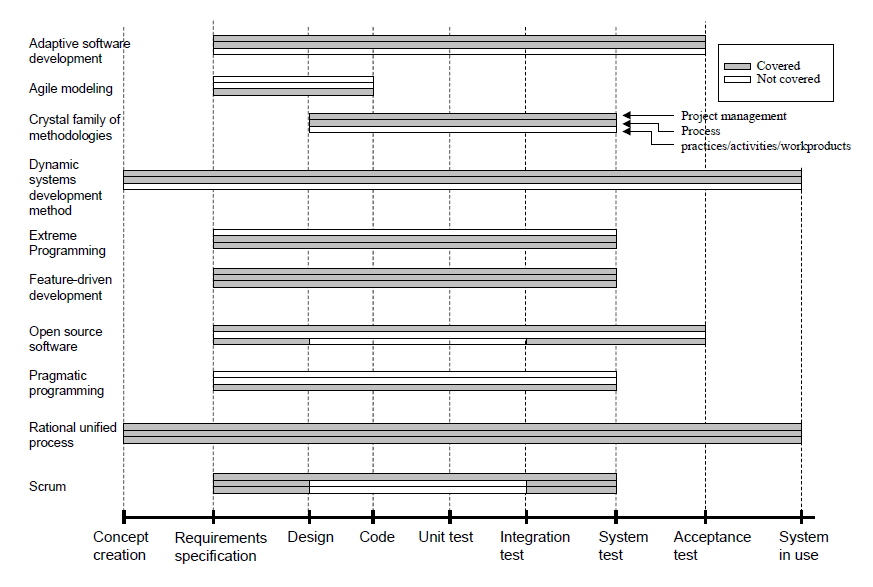
**Regular Builds:**Once the first build is running, future versions will always be demonstrable. If it does not run and meet requirements, it is not eligible as a build.

**Configuration Management:**Detailed documentation of changes will be recorded, and historical tracking will be available for each completed source code file.

**Progress Reporting:**Regular team communication to report back to the project manager, and milestone reports to the client and supervisor.

This development methodology not clearly state practices for communication and reflection, so as a team we have decided on our own additional practices to cover these. They can be found outlined in the team contract.

Due to being a small team and each having several other commitments during the project, we have also elected to adopt some development processes from Kanban. Time management is a huge factor, and FDD lacks control over workload amounts, and methods for a clear, visual workflow. As a team, we will set a realistic limit to features in progress and track this progress on a Kanban board in GitHub. This should aid us in avoiding; building features we do not need, designing more features than we can code, writing more code than we can test and testing more code than we can deploy (J. Buchan, personal communication, April, 2016).

Software Development Life-Cycle support (Abrahamsson et al., 2002). See below for header definitions.

**Concept Creation:** This phase is where the need and viability of the system is evaluated, and a feasibility study is conducted to determine whether development should go ahead.

**Requirements Specification:** The business requirements are gathered from stakeholders and analyzed for validity, complexity and importance.

**Design:** Documents and prototypes are created to prepare a guideline for implementation.

**Code:** The work put into the system, where features are programmed and implemented.

**Unit Testing:** Tests created whilst coding so that programmed behaviours are validated to work as expected.

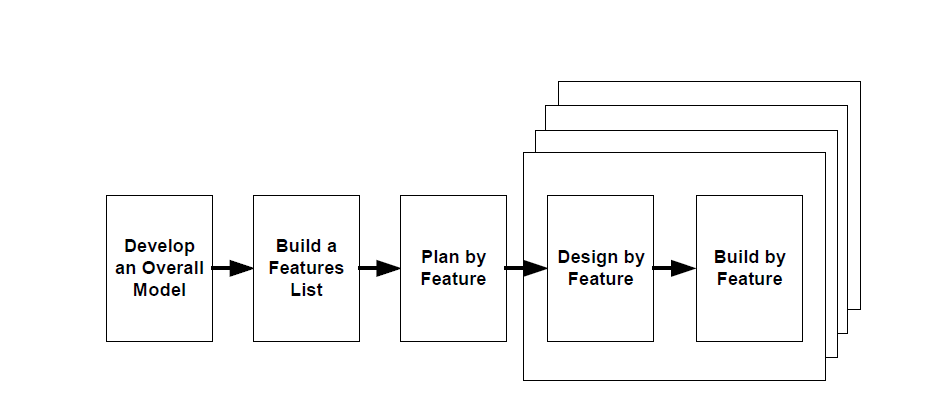
**Integration Testing:** Tests created to ensure that each component of the system communicates with the others as expected.

**System Testing:** Testing the behavior of the entire solution, as defined by the project scope.

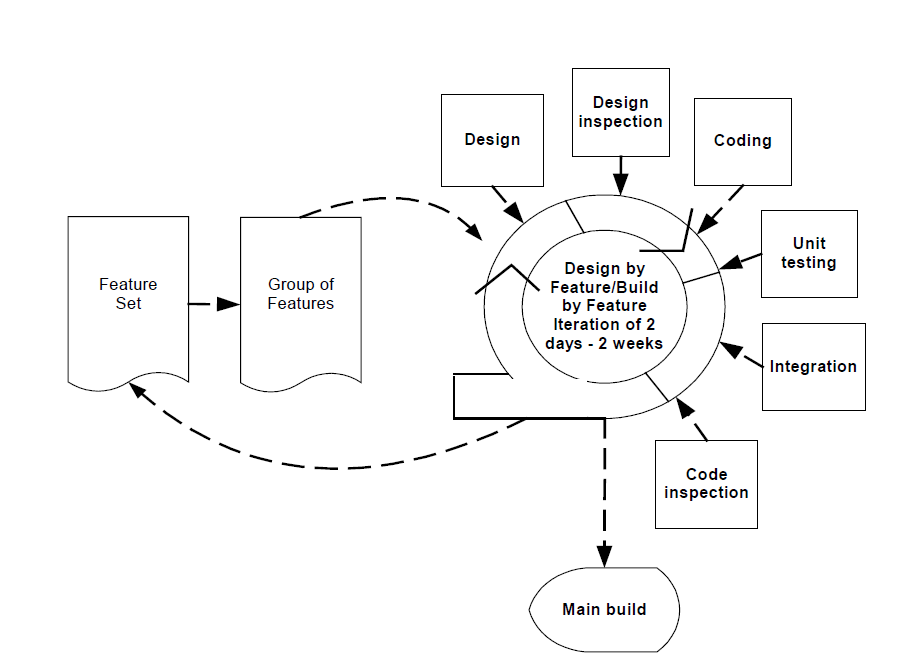
**Acceptance Testing:** Tests done by users to ensure all functional requirements and non-functional requirements are met.

**System in Use:** After testing the product is deployed and maintained, bug fixes and required changes are addressed.

(What are the Software Development Life Cycle (SDLC) phases?, n.d.).



Processes of FDD (Palmer & Felsing, 2002)



Design and Build by feature processes of FDD. (Abrahamsson et al., 2002)

## Methodology Rationale

FDD mainly focuses on the design and build of a system. The project we have been tasked with relies heavily on these stages, as our client’s top priority is to have a clear roadmap of the development so that the project may continue after we complete the Research and Development paper.

Following this reasoning, we found it would not be suitable to follow methodology like Scrum, as the length of the tasks are still unknown, and the skills necessary to implement the features may be more advanced than we have anticipated. Sprints would be ineffective, as a task may take less or more time than we expect, the rules around sprints would lead us to using our time inefficiently. It would be difficult for us to fully plan when we expect each part of the project to complete, so we needed a methodology that was more adaptive than Scrum.

Additionally, we opted to use a more lightweight methodology, as we preferred to tailor one to our own experiences. Each team member has already been a part of different project groups over the last two or more years, and therefore has gained knowledge on various methodologies and how well they worked with each group. Out-of-the-box methodologies like Scrum can be quite intimidating and stressful, as there are strict practices to follow; in failing those the project can quickly fall apart.

Also, because we are working with a non-technical client, and have been tasked with a project in a concept phase, methodologies such as Waterfall and Spiral were out of the question. It can be inferred that the requirements and design will absolutely change during development, and each feature is likely to cycle through more than one iteration.

FDD for us feels like a good choice, as it has a well-defined set of roles. Considering that our group has 5 members, and the prospectus called for a group size of 2-3 members, we need to ensure that each team member is able to make a worthy contribution to the project. All the while feeling like they are using their unique skillset to the best of their ability. In particular, the programmer role, as stated above, will be an amalgamation of several roles, which allows each member to be an expert of their own domain within the project, giving each member the opportunity to take on a leadership role.

Our initial meeting with the client revealed that there are few stakeholders in the project, and as a busy person, he will not have constant input into the project, so he only wishes to hear from us as we hit milestones and regarding design decisions.  In this respect, we searched for a methodology that put more responsibility in the developers, than in the client or stakeholders. We found FDD is very self-directed, that is to say, that development can move forward without the need to check in constantly with a client at the end of sprints or cycles. Quality control relies on the developers and testers themselves, so they can ensure their work is up to standard.

FDD covers the phases from requirements specification, up to system testing, in the aspects of project management, processes and practices. This suits this project, as the earlier stages of project conception were covered by the client, so we can focus on the requirements and rely on the client to provide feedback as the product owner.

## Communications Management Plan

**Introduction**

The Communications Management plan will be used to set the communications structure for this project. This will serve as a guide for communications throughout the life of the project and will be updated as communications need to be changed. In this plan the identities and the roles of the persons involved in this project will be defined.

It also includes a communications matrix which maps the communication requirements of this project. An in-depth guide for conducting meetings details both the communications rules and how the meetings will be conducted, ensuring successful meetings. A project team directory is included to provide contact information for all stakeholders directly involved in the project.

**Communication Approach**

Vinicius, the Project Manager for this R&D project will ensure effective communications in this project by recording meeting minutes, guiding meetings and allocating project tasks. The communications requirements are documented in the Communications Matrix presented in this document.

As with many projects, updates or changes may be necessary as the project progresses.

Changes or updates may be required due to changes in personnel, scope, budget, or other reasons.

These changes should be approved by stakeholders (Client and Supervisor), updated in the documentation and redistributed to the team and all stakeholders. 

**Communication Constraints**

All project communication activities will occur within the project’s approved budget, schedule, and resource allocations. The project manager is responsible for ensuring that communication activities are performed by the project team, and without external resources which will result in exceeding the authorized budget. Communication activities will occur in accordance with the frequencies detailed in the Communication Matrix in order to ensure the project adheres to schedule constraints.

**Stakeholder Communication Requirements**

As part of identifying all project stakeholders, the project group will communicate with each stakeholder in order to determine their preferred frequency and method of communication. This feedback will be maintained by the project manager in the project’s Stakeholder Register. Standard project communications will occur in accordance with the Communication Matrix.

Once all stakeholders have been identified and communication requirements are established, the project team will maintain this information in the project’s Stakeholder Register and use this, along with the project communication matrix as the basis for all communications.

**Communication Matrix**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Communication Type | Objective | Medium | Frequency | Audience | Deliverable | Format |
| Kickoff Meeting | Introduce team members | Face-to-Face | Once | Team | N/A | N/A |
| Team Meetings | Review progress of project and work to be done | Face-to-Face | Weekly | Team | Meeting Minutes | Copy archived in OneNote on the cloud |
| Supervisor Meetings | Get answers to questions and guidance for the project | Face-to-Face | When necessary | Nikola Kasabov (Project Supervisor) | Project Status Report | Copy archived in OneNote on the cloud |
| Client Meetings | Report on status of project to client and get feedback | Face-To-Face | When necessary | Robin Hankin (Client), Nikola Kasabov (Project supervisor) | Project Status Report | Copy archived in OneNote on the cloud |
| Project Status Reports | Report project status including activities, progress, costs and problems | E-mail | Monthly | Robin Hankin (Client), Stephen Thorpe (Project Moderator) Nikola Kasabov (Project Supervisor) | Project Status Report | E-mail |

# Project Plan

## Work Breakdown Structure

1. Initiation
   1. Project Proposal
   2. Project Proposal Presentation
2. Planning
   1. Feasibility Evaluation
      1. Operational feasibility
      2. Resource feasibility
      3. Technical feasibility
      4. Schedule feasibility
      5. Financial and Legal feasibility
   2. Project Process Analysis and Review
      1. Presentation to Client
      2. Mid Project Progress Review
   3. Major Upskilling
3. Executing
   1. Iteration 0
      1. Develop an Overall Model of system
      2. Build a Features List
      3. Plan by Feature
   2. Iteration 1 – X
      1. Design by Feature
      2. Build by Feature
      3. Review of Object Model for changes (optional step)
      4. Minor Upskilling (where necessary)
   3. Client Feedback
   4. Supervisor Feedback
4. Closing
   1. Poster
   2. Reflective Report
   3. Portfolio
   4. Final Release

## Provisional Milestone Plan

|  |  |  |
| --- | --- | --- |
| Provisional Milestones | Semester | Finish |
| Project Proposal | 1 | 30/03/17 |
| Project Proposal Presentation | 31/03/17 |
| Feasibility Evaluation | 26/05/17 |
| Project process analysis and review | 26/05/17 |
| Presentation to Client | 2/06/17 |
| Mid project progress review | 2/06/17 |
| Major Upskilling | 14/07/17 |
| Holiday | ---------- | 14/07/17 |
| Stage 0 | 2 | 28/07/17 |
| Plan - design of project model complete | 28/07/17 |
| Stage 1 | 11/08/17 |
| Framework and Database setup complete | 11/08/17 |
| Stage 2 | 25/08/17 |
| Website development complete | 25/08/17 |
| Stage 3 | 22/09/17 |
| Website integration with database completed | 22/09/17 |
| Stage 4 | 06/10/17 |
| Online Real Time Scoreboard implementation complete | 06/10/17 |
| Stage 5 | 20/10/17 |
| Website UX\UI improvement complete | 20/10/17 |
| Handover Complete | 27/10/17 |
| Client feedback | 29/09/17 |
| Supervisor feedback | 20/10/17 |
| Poster | 3/11/17 |
| Reflective Report | 3/11/17 |
| Portfolio and Final Product | 3/11/17 |

Legend:

Red text indicates Project Deliverable

## Network Diagram

Please refer to Appendix A

## Gantt Chart

Please refer to Appendix B

## Justification

Project Plan was designed to address the client's main request and satisfy our team needs when entered in the developing phase. Since the project deliverable does not limit to the software development attribute only, a significant emphasis is given to research and analysis of the project requirements and necessary infrastructure to deliver the product. A structured view of the milestones may be seen in the work breakdown structure(WBS) and a visual representation in the Gantt Chart and Network diagram.

The feasibility study will be broken down into the following segments:

• Technical Feasibility: Investigate different implementations of the proposed system, the hardware, software and infrastructure required for each implementation, and any existing solutions that may be utilized.

• Operational Feasibility: Investigate if the proposed system suits the stakeholders involved (judges in particular) and how well the proposed system solves the project problem.

• Resource Feasibility: Investigate resources such how much time is available to build the system, when it can be built, and what hardware and software are available.

• Schedule Feasibility: Investigate how long it will take to develop the system (software and hardware).

• Financial Feasibility: Investigate the total estimated cost of the project.

• Legal Feasibility: Investigate whether the proposed system will conflict with any legal requirements of Auckland University of Technology, Auckland Mathematics Association, and ASB Stadium.

Once the feasibility study is finished and a report is constructed, we will look to present our findings to our client Robin Hankin. If the project is deemed feasible, we will also recommend a implementation of the proposed system to the client that best suits our needs. It is important for us to make sure that our implementation meets the client's requirements and this meeting will allow us to discuss the implementation in detail.

Our project may differ from others, as there will be a large amount of time spent on planning as a major deliverable is the feasibility study. This is at the request of the client, as he will need to pass on this roadmap and feasibility study to the Auckland Mathematical Association to review. Dr. Hankin is not directly in control of whether the system will be implemented, he has asked for the aid of AUT students to promote AUT and find a cost-effective way to create technology as well.

Therefore, we will be spending a large portion of the project ensuring that we meet the client’s needs before we begin development. However, no matter the outcome of the feasibility study, we intend to create a piece of software that serves the functions the client has requested.

**Project phases**

**Initiating**

In this stage, there is an emphasis on developing a team work environment and understand the project to be proposed. The team will be introduced to the client and few possible stakeholders in order to gather information and requirements about the project in discussion. These steps will lead to the development of a project proposal in which aims to outline the rationale, scope, methodology approach, plan and the skills required to complete the project.

**Planning**

In this stage, we target a detailed project feasibility study that will assists the client deciding if the project should take place or not and will contain enough information to assists future team undertaking this project. A greater in-depth investigation of the project requirements will be undertaken, followed by an extensive research that will highlight the project constraints and stablish the structure required to the commence of the product development. In addition, more the one solution to the client\organization problem will be introduced

**Executing**

At conclusion of the planning phase, there will be enough information to start the product development where the use of our chosen methodology **Feature Driven Development (FDD)** will apply; this phase will be broken down into iterations.

The first iteration, Iteration0, will take 2 weeks and will construct our project model that will be used throughout the followed iterations. In this iteration, we will model the project with diagrams, followed by breaking the project into features and grouping them by dependencies and priority.

The following iterations are consisted on the design, construction, and test of a set of features gathered in iteration0. When feature is concluded the team will quickly review project model to ensure no changes have been made to it. Each iteration may or may not require minor upskilling.

There may be as many iterations as required, as it is outlined in the chosen methodology, until either the client is happy with the product or the time allocated is concluded.

**Closing**

Project closing will include the finalization of any pending documentation as well as final client meeting to inform client of overall project status. A final presentation will be arranged.

Project will be then concluded and team will focus on our paper deliverables.

\*\*Note: There may be required changes to the project plan depending on the outcome of the planning phase or any further alterations requested by client.

## Risk Management Plan

|  |  |  |
| --- | --- | --- |
| Risk | Risk Level | Control/Recommendation |
| Proposal presentation may be rejected. | High | Ensure that a draft of the proposal is sent to the supervisor before the presentation to ensure that it is up to standards. The presentation must be well planned and rehearsed. In case of rejection, the proposal must be reviewed until a satisfactory level is reached. |
| Project data is lost due to an unexpected accident. | High | Regularly backup all data for this project to online/cloud storage such as Google Drive and GitHub. |
| Supervisor is not assigned or is unavailable to work/contact with the group for an extended period of time. | Medium | Contact Steven Thorpe querying supervisor details and available options. If supervisor is unavailable for important actions then Steven may be treated as a temporary supervisor. Risk level may be escalated if this continues for an extremely long period of time. |
| Team member leaves the team unexpectedly. | Medium | That team member's role must be re-assigned and their workload distributed among the rest of the team. If possible, any work that team member was working should be retrieved. |
| Hardware costs exceed allocated funds. | Medium | There are several options.   1. Negotiate with Stakeholders to increase funding. 2. Find another vendor with a cheaper price. 3. Decrease the requirements for hardware and seek a cheaper alternative. |
| Stakeholders have an inaccurate expectation for the project. | Low | The feasibility report must clearly include the scope and specifications of the project as well as limitations. This feasibility report should be presented to stakeholders and feedback should be recorded to get an idea of what the stakeholder perceives. |

**Key:**

|  |  |  |
| --- | --- | --- |
| Risk Level | Risk Evaluation | Risk Mitigation |
| High | Risks which will have a severe impact on the completion of the project. These risks may cause critical and key requirements to fail. The impact and control of these risks may deplete an excessive amount of resources. | Risk must be monitored on a daily basis. The risk owner must document all avoidance and mitigation strategies and must be made readily available and easily accessible. These strategies must be put into action until the risk is diminished or is avoided. |
| Medium | Risks which will have a moderate impact on the completion of the project. These risks may cause requirements to be at a minimal acceptable level only. The impact and control of these risks may deplete a moderate amount of resources. | Risk must be monitored on a weekly basis. The risk owner will document all avoidance and mitigation strategies. These strategies will be put into action until the risk is diminished or is avoided. |
| Low | Risks which will have a minor impact on the completion of the project. These risks may cause some requirements to be below targeted goals but will still be satisfactory. The impact and control of these risks may deplete a minimal amount of resources. | Risk must be monitored on a bi-weekly basis. Avoidance and mitigation strategies can be documented and put into action but is not necessary unless it is possible for the risk to escalate levels. |

# Skills and Knowledge Involved

Skills and knowledge involved. Identify the skills and knowledge required to complete this project. List the IT-specific (specific to your degree) skills first and then add the full range of personal, professional and technical capabilities. Identify what skills and knowledge are not yet available in your team and how or when you plan to acquire them. Make sure time for this upskilling is also included in your plan.

**Personal Capabilities**

|  |  |  |  |
| --- | --- | --- | --- |
| **Skill** | **Description** | **Presence in team** | **Plan for upskilling** |
| Programming ability | Having the necessary programming skills to partake in this software development project. | Yes |  |
| Communication | Being able to effectively communicate thoughts, concerns and information with team members, supervisor and the client. | Yes |  |
| Research and Learning | To be able to independently learn skills that are needed for the project. | Yes |  |
| Problem Solving | To be able to find the most suitable solutions to problems using the tools and knowledge available. | Yes |  |

**Professional Capabilities**

|  |  |  |  |
| --- | --- | --- | --- |
| **Skill** | **Description** | **Presence in team** | **Plan for upskilling** |
| Teamwork | Working well with other team members by combining strengths and sharing ideas. | Yes | Worked on throughout the project. |
| Time Management | The ability to manage the hours spent on the project to maximize efficiency. | Yes | Worked on throughout the project. |
| Feature Driven Development | A software development process. Will help us to deliver tangible, working software in a timely manner. | Partially | Worked on throughout the project. |
| Planning and Organization | Setting goals and determining the steps necessary to complete them. | Yes | Worked on throughout the project. |

**Technical Capabilities**

|  |  |  |  |
| --- | --- | --- | --- |
| **Skill** | **Description** | **Presence in team** | **Plan for upskilling** |
| Trello | Web-based project management application. Will be used to organize tasks and keep track of the project. | Yes | Worked on throughout the project. |
| Java and JavaScript | This will be the back-end programming language that will be used for the Online Real time Scoreboard. | Yes | Use https://www.w3schools.com/ and other online resources to help us further develop our skills. |
| GitHub | A web-based version control repository. Will be used to share code for the Online Real time Scoreboard. | Partially | Worked on throughout the project. |
| HTML5 | A markup language. Will be used to structure the Online Real time Scoreboard. | Partially | Learning more HTML5 during the project through AUT's Web Development Paper.    Use https://www.w3schools.com/ and other online resources to help us further develop our skills. |
| OneNote | Online program used to share documentation for the project with all team members. | Yes | Worked on throughout the project. |

## Skill Matrix

Please refer to Appendix C

# Estimated Costs

The software will be open source and therefore will be free of additional costs.

The hardware's cost will be harder to estimate as it will be evaluated through the feasibility study and as the project goes on. Therefore, we will put the maximum amount of allocated funds for hardware as an estimate for now.

|  |  |
| --- | --- |
| Phases | Cost |
| Initiation | $6,875 |
| Planning | $15, 468.75 |
| Executing | $24,062.50 |
| Closing | $3,437.50 |
| Total Cost | $49,843.75 |
|  |  |

|  |  |  |
| --- | --- | --- |
| Project Roles | Members | Cost per hour |
| Project Manager | Vinicius | $45 |
| Chief Architect | Hayley | $50 |
| Development Manager | Karanjit | $40 |
| Build Engineer | Jin | $30 |
| Lead Tester | Alex | $30 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Costs | Team | GST | Supervisor | GST | Hardware | Total Costs |
| Cost per hour | $195 | $29.25 | $142 | $21.30 | N/A | $387.55 |
| Cost per day | $278.60 | $41.80 | $20.30 | $3.05 | N/A | $343.75 |
| Cost per week | $1,950 | $292.50 | $142 | $21.30 | N/A | $2,405.80 |
| Total Costs | $58,500 | $8,775 | $4,260 | $639 | $20,000 | $82,124 |

* All values are rounded up where appropriate.
* For overlapping roles, the highest cost is paid out.
* There are 5 team members.
* Each team member contributes 300 hours of work throughout the project.
* There are 30 weeks total for this project.
* Training costs are not included for now. This will be evaluated and added in a future revision.

# Disclaimer

Please refer to appendix D

# References

Abrahamsson, P., Salo, O., Ronkainen, J., & Warsta, J. (2002). *Agile software development methods: Review and analysis.*Espoo, Finland: VTT Publications.

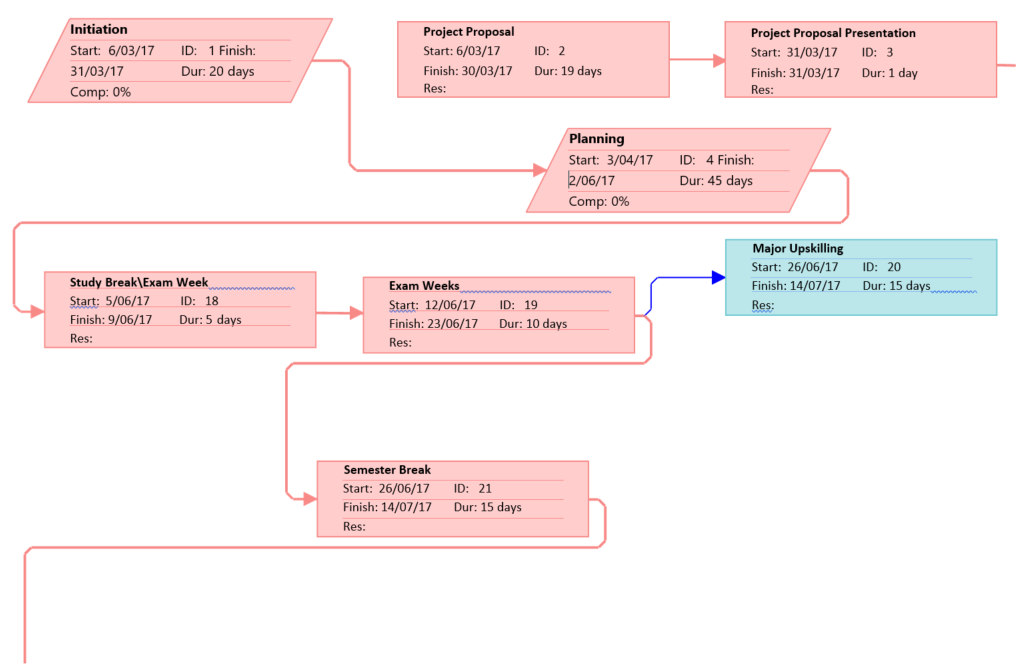
Buchan, J. (2016). *Lean Thinking and Kanban*[PowerPoint slides] Software Engineering ENSE701. Retrieved from Auckland University of Technology Blackboard website:  <https://blackboard.aut.ac.nz/>

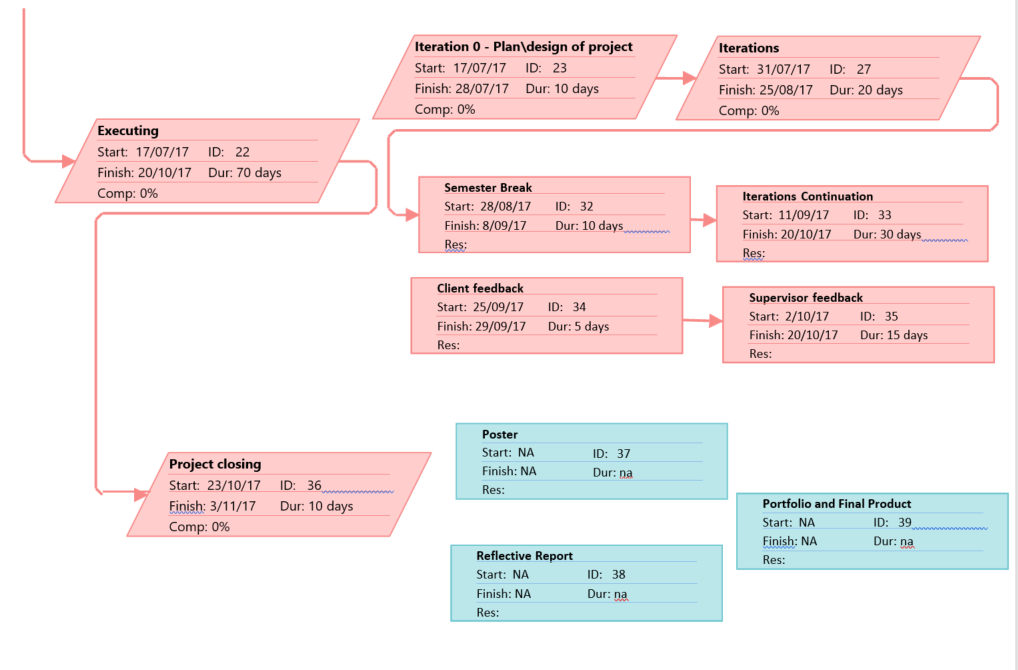
Palmer, S. R., & Felsing, J. M. (2002). A Practical Guide to Feature-Driven Development. Upper Saddle River, NK, Prentice-Hall.

*What are the Software Development Life Cycle (SDLC) phases?* (n.d.).Retrieved May 22, 2017, from <http://istqbexamcertification.com/what-are-the-software-development-life-cycle-sdlc-> phases/

# Appendix

## Appendix A



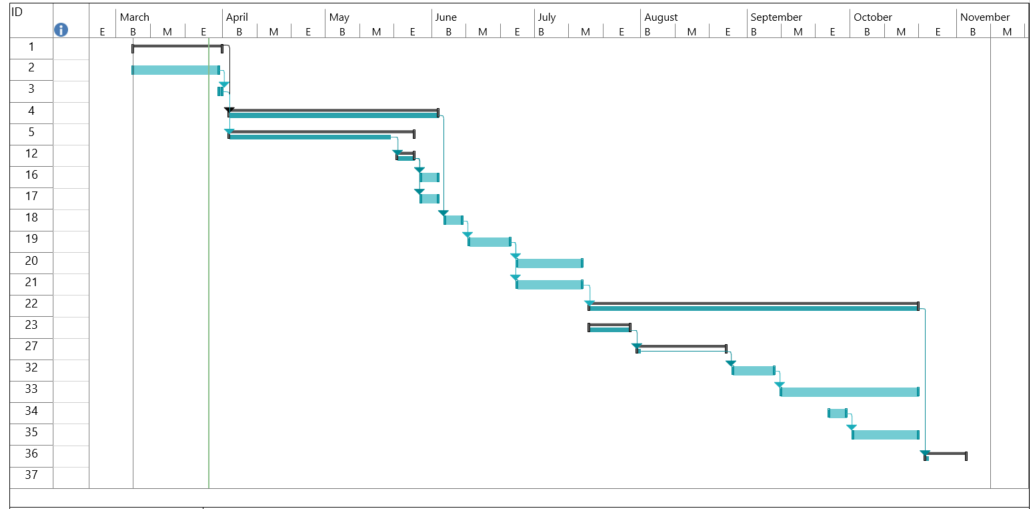
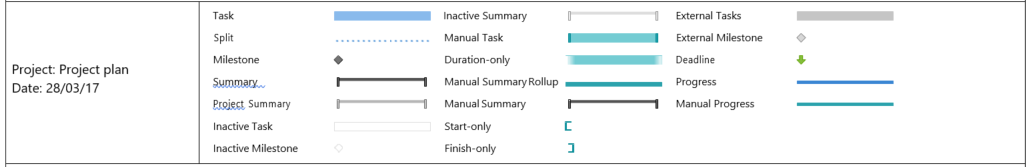


## 

## 

## 

## Appendix B



## 

## Appendix C

**Key:   
Nu** = Never used; **Tried**= Used few times \ can get around it;

**OK** = Have some knowledge; **Conf** = Feels comfortable

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Skill Matrix** | | | | | | | | | | | | | | | | | | | | | |
| **Group** | **Skill Expertise** | **Vini** | | | | **Hayley** | | | | **Karanjit** | | | | **Jin** | | | | **Alex** | | | |
| **Nu** | **Tried** | **Ok** | **Conf** | **Nu** | **Tried** | **Ok** | **Conf** | **Nu** | **Tried** | **Ok** | **Conf** | **Nu** | **Tried** | **Ok** | **Conf** | **Nu** | **Tried** | **Ok** | **Conf** |
| **Application Server** | Apache |  | x |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | x |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | | | | | | | | | | | | | | | | | | | | |
| **Database** | Oracle 11g Database |  | x |  |  |  |  |  |  |  | x |  |  |  | x |  |  |  | x |  |  |
| MySQL |  |  |  |  |  |  |  |  |  | x |  |  |  |  |  |  |  | x |  |  |
|  | | | | | | | | | | | | | | | | | | | | |
| **Web Technology** | ASP.NET |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| HTML/XHTML 1.1/CSS 2.0 |  |  |  |  |  |  |  |  |  | x |  |  |  |  |  |  |  |  |  |  |
| HTML5\CSS3 |  |  | x |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| XML/XSLT |  | x |  |  |  |  |  |  |  | x |  |  |  |  |  |  |  |  |  |  |
|  | | | | | | | | | | | | | | | | | | | | |
| **Mobile Computing** | Android |  |  | x |  |  |  |  |  |  | x |  |  |  |  |  |  |  |  | x |  |
| IOS |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | x |  |  |
| Windows |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | | | | | | | | | | | | | | | | | | | | |
| **Languages / Development Tools** | C |  | x |  |  |  |  |  |  |  |  | x |  |  |  |  |  |  |  |  |  |
| C++ |  |  |  |  |  |  |  | x |  |  |  |  |  | **x** |  |  |  |  |  | x |
| JAVA |  |  | x |  |  |  |  | x |  |  |  | x |  |  | x |  |  |  |  | x |
| C# |  | x |  |  |  |  | x |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Python |  |  |  |  |  |  |  |  |  | x |  |  |  | x |  |  |  |  |  |  |
| PHP |  | x |  |  |  |  | x |  |  | x |  |  |  | x |  |  |  |  |  |  |
| Javascript |  | x |  |  |  | x |  |  |  | x |  |  |  |  |  |  |  |  |  |  |
| Ruby |  |  |  |  |  | x |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | | | | | | | | | | | | | | | | | | | | |
| **Project Management Tools** | MS PROJECT |  |  | x |  |  |  |  | x |  | x |  |  |  | **x** |  |  |  |  |  |  |
| Trello |  |  |  | x |  |  | x |  |  |  |  | x |  |  |  | x |  |  |  | x |
|  | | | | | | | | | | | | | | | | | | | | |
| **Multimedia** | ADOBE PHOTOSHOP |  |  |  |  |  |  |  | x |  |  | x |  |  |  |  |  |  | x |  |  |
| AFTER EFFECTS |  |  |  |  |  |  | x |  |  |  |  |  |  |  |  |  |  |  |  |  |
| ADOBE ILLUSTRATOR |  |  |  |  |  | x |  |  |  | x |  |  |  |  |  |  |  |  |  |  |
| PREMIER |  |  |  |  |  |  |  | x |  |  |  |  |  |  | x |  |  |  |  |  |
|  | | | | | | | | | | | | | | | | | | | | |
| **QA Expertise** | White Box Testing |  |  |  |  |  |  | x |  |  |  |  |  |  |  |  |  |  | **x** |  |  |
| Black Box Testing |  |  |  |  |  |  | x |  |  |  |  |  |  |  |  |  |  | **x** |  |  |
| Performance Testing |  |  |  |  |  | x |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

## Appendix D

**Auckland University of Technology**

**Bachelor of Computer & Information Sciences**

**Research & Development Project**

**Disclaimer:**

**Clients should note the general basis upon which the Auckland University of Technology undertakes its student projects on behalf of external sponsors:**

*While all due care and diligence will be expected to be taken by the students, (acting in software development, research or other IT professional capacities), and the Auckland University of Technology, and student efforts will be supervised by experienced AUT lecturers, it must be recognised that these projects are undertaken in the course of student instruction. There is therefore no guarantee that students will succeed in their efforts.*

*This inherently means that the client assumes a degree of risk. This is part of an arrangement, which is intended to be of mutual benefit. On completion of the project it is hoped that the client will receive a professionally documented and soundly constructed working software application, some part thereof, or other appropriate set of IT artefacts, while the students are exposed to live external environments and problems, in a realistic project and customer context.*

*In consequence of the above, the students, acting in their assigned professional capacities and the Auckland University of Technology, disclaim responsibility and offer no warranty in respect of the “technology solution” or services delivered, (e.g. a “software application” and its associated documentation),both in relation to their use and results from their use.*

## Team Agreement

**Code of Conduct:**We will:

* Plan out our work to avoid problems with conflicting work and prevent any possible issues
* Communicate openly with the other team members to keep them informed of their work regarding the project.
* Put in our fair share of work and be open to guiding or assisting others if needed.
* Be respectful to the work and ideas of fellow group members and provide constructive criticism.
* Work to attain a quality final product to present.

**Participation:**We will:

* Work to ensure we meet our personal and group deadlines efficiently.
* Encourage each other and provide useful feedback which can be applied.
* Inform the group of our current status with our work.
* Present ideas to the group in an understandable and sensible manner.
* Communicate if any interruptions will be made to the team.
* Mark cards on Trello that we are working on to avoid confusion.
* Move cards on Trello through different phases (In development, In testing, Done) to show progress.

**Communication:**We will:

* Include the entire team in all correspondence regarding the project, whether through email, Slack, GitHub, or other means.
* Primarily use Slack for communication within the team. Texting or calling is the first alternative to Slack.
* Use GitHub to show progress on the project.
* Use emails to communicate with people outside of our group (e.g. Supervisor, Client).
* Notify the team on Slack when a task is started and finished to keep everyone on the same page.
* Provide a response to communication in a reasonable timeframe (24 hours).
* Work towards finding a solution rather than making excuses.
* Ask others for help when things are not clear or may be misunderstood.
* Keep discussion regarding the project related to our work .

**Problem Solving:**We will:

* Work together as a unit to overcome our problems.
* Provide each other with fair constructive criticism.
* Use feedback to help on improving our overall product.

**Meeting Guidelines:**We will:

* Participate as a team regularly during our agreed meeting times.
* Keep a record of our team meetings and have it available for all members.
* Be prepared for our meetings, with an agenda and plan of action.
* Provide 24 hours’ notice if unavailable or a situation arises without warning.
* Keep a document of any major or minor issues that may arise and have it available with for all members.
* Provide 48 hours' notice to schedule an additional meeting with the team to allow members to prepare.
* Reach meetings no later than 15 minutes.
* Notify the team on Slack if we might be late to the meeting.
* Conduct weekly meetings on Tuesday at AUT. Exact times will vary.

**Roles and Responsibilities:**

* Vinicius Alves
  + Project Manager
  + Recording meeting minutes; Guide meetings; Allocate tasks
* Karanjit Gahunia
  + Development Manager
  + Arrange and book meeting rooms ; Post meeting summaries in Slack ; Communicate with Supervisor
* Hayley-Belle Cleverdon
  + Chief Architect
  + Technical Writer
  + Proof-read and confirm details of meeting summaries
* Alex Lu
  + Inform client of milestones, issues and arrange meetings with client;
  + Write agenda for meeting with supervisor(moderator) and client
* Seung-Kyu Jin
  + Build Engineer
  + Post weekly summaries of tasks conducted by the team.

All team members will be developers and testers.

By signing this agreement, you agree to the terms listed in the team contract and commit to follow those terms accordingly.

Vinicius Alves  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date: / /2017

Karanjit Gahunia  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date: / /2017

Hayley-Belle Cleverdon  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date: / /2017

Alex Lu  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date: / /2017

Seung-Kyu Jin  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date: / /2017