

# THE BASEMENT THEATRE



## PROJECT PROPOSAL

AARON MORPETH | DAVID GUTLA | JOZELLA ROQUE | KAI SHAO

---

# Contact Information

---

## CLIENT

Name	Email	Phone
Elise	-	-

## SUPERVISOR

Name	Email	Phone
Greig	-	-

## STUDENTS

Name	Student ID	Email	Phone
Aaron Morpeth	-	-	-
David Gutla	-	-	-
Jozella Roque	-	-	-
Kai Shao	-	-	-

# Table of contents

Contact Information.....	1
Client .....	1
Supervisor .....	1
Students .....	1
Table Of Contents.....	2
1 Terms Of Reference .....	4
1.1 Introduction .....	4
1.2 Rationale .....	5
1.2.1 Existing System .....	5
1.2.2 Why The Project Is Needed .....	5
1.3 Scope.....	6
1.4 Objectives .....	7
1.5 Constraints .....	7
1.6 Cost Estimation .....	7
2 Skills - Knowledge - Roles .....	8
2.1 Roles .....	8
2.2 Required Skills .....	9
2.3 Resources .....	9
3 Project Plan .....	11
3.1 Approach .....	11
3.2 Agile .....	11
3.2.1 Agile Manifesto .....	11
3.2.2 Scrum .....	12
3.2.3 Scrum Roles .....	13
3.2.4 Scrum Artefacts.....	13
3.2.5 Scrum Ceremonies .....	14
3.2.6 Planning Poker .....	14
3.2.7 Scrum Overview .....	15

3.2.8 Scrum Phase Overview .....	16
3.3 Project Processes.....	17
3.4 Justification .....	22
3.5 Schedule .....	23
3.6 Up Skilling .....	23
3.7 Sprint Plan .....	24
4 Quality Assurance .....	25
4.1 Quality Control .....	25
4.1.1 Requirements .....	25
4.1.2 Verification & Validation .....	25
4.1.3 Ui Testing .....	25
4.1.4 Inspections And Acceptance Testing .....	26
4.1.5 Coding .....	26
4.1.6 Documentation .....	26
4.1.7 Meetings.....	26
4.1.8 User Story Definition Of Done .....	26
4.1.9 Sprint Definition Of Done .....	27
4.1.9.1 Usability Criteria .....	27
4.1.9.2 Peer Review .....	27
4.1.9.3 Client Reivew.....	27
4.2 Deliverables .....	28
4.4 Communication Plan.....	29
4.5 Risk Management.....	31
4.5.1 Risk Levels.....	31
4.5.2 Risk Impact .....	31
4.5.3 Risk Practices .....	31
5 Glossary.....	35
6 References.....	36
7 Disclaimer.....	37

---

# 1 Terms of Reference

---

## 1.1 INTRODUCTION

---

The Basement Theatre is a theatre which is situated at the heart of Auckland City. The theatre is most famous for upcoming New Zealand artists who are not afraid to step out of their comfort zone and showcase their ability to perform their talent amongst an eager audience. The theatre's artists that showcase are mostly local upcoming performing artists, from comedians, musicians, to creative poets who all contribute to the basement theatre with their. The basement theatre is a safe haven for artists who want to explore new ideas and acts. The marketing team are searching for an approach to gather more audience by means of interaction using technology platforms, something which they have not found. The basement theatre wants to give the audience an opportunity for interaction and feedback, beyond spectating a show.

With the rise of technology, it is becoming much easier obtaining user information to acknowledge customer feedback, and acquire understanding from that information to essentially help improve services for customers effectively.

So far, The Basement Theatre already has an existing online website which consists of their information, but however, the website is being renovated and will relaunch next year in 2016. There is also an existing third party ticketing system, iTicket, which is a ticketing system that was developed in 2014 by a BCIS Research and Development group, but unfortunately the client did not pursue in that release.

The aim of this project is to imperatively create value with customer feedback and statistics through a mobile application, and make the experience enjoyable to the application users.

To make the aim possible, this proposal presents the appropriate processes undertaken to aid in implementing this project. The team has agreed to use an Agile approach; SCRUM, for this project, due to circumstances. The proposal outlines the procedures used to assure quality, manage risk, and plan the project.

## 1.2 RATIONALE

---

### 1.2.1 EXISTING SYSTEM

---

The Basement Theatre is currently in a technological transition. They are developing a new website on word press with more functionality and a cleaner, more professional look. Currently on the website, you are able to view the upcoming events, performer profiles, photos and a currently non-functional booking button. The Basement Theatre has a third party ticketing system, iTicket which provides up to date statistics of their audience members.

### 1.2.2 WHY THE PROJECT IS NEEDED

---

The Basement Theatre would like to attract more audience and various people who normally would not come to the theatre, to step out of their comfort zone. The theatre is a great place to socialise and meet new people. The main reason why the project is needed is so that it can increase ticket sales and at the same time, keep an interaction between theatre and audience via user feedback to help the company better understand their customers. The mobile application will allow the basement theatre to collect information about their audience (name, age, gender), whether they liked the show or not, and their review & feedback. The mobile application will help expose The Basement Theatre to new audiences.

## 1.3 SCOPE

---

This project is a means to identify, explore and develop possible application ideas to improve overall sales and feedback in the future. The users of this applications are in the age range of 20 - 40 who will all be attending shows at the theatre. This application aims to collect and evaluate customer satisfaction and reviews through feedback for shows that they have viewed. The user will have an ability to interact with the application and submit their information by means of feedback form and rating system. The application should let the users have an enjoyable and user friendly experience.

The mobile application will be developed for the basement theatre company to help them interact with the audience through a technology platform. The application will have such features such as feedback form functionality, to gather statistics and help the client understand what their customers desire, so that it will help the company improve the events and services for future. The mobile application will be composed of several tabs. The home page will display the upcoming events and a brief information about that event for users to see. The purchase tab will display upcoming events so that users can search and browse for their desired event that may interest them and purchase it. The purchased tab when selected will allow users to see what they have purchased. The settings will obtain information about name, gender and age via social media. To make this possible, the users will be required to register or sign in to their account via Facebook. Once the users have reached their homepage, they are able to access other pages in the application such as Settings, upcoming shows, purchase shows, and purchased shows. If a show is purchased, it will add the show to the 'purchased' page where users may review and rate the show with emoticons. They also have a choice to complete a survey which will help the theatre better understand customer desires. The client furthermore would also like the project team to look into integrating the third party ticketing system into the application.

## 1.4 OBJECTIVES

---

The Basement Theatre is currently working on a new layout of the website. An app on mobile systems needs to be developed with ticketing function in order to make it more convenient for customer to purchase ticket. To enhance the interaction between audience and actors, a rate & review system needs to be built. By using this system, audience can make comments about the show and give the show a happy/sad face. The app requires the user to register or log in with Facebook account. Each app user will have a unique QR code for their id. Staff in theatre can use QR code scanner to record the audiences attended the show to stamp a mark on their profile page. The final main focus is to collect the information from the user's information. Information such as name, gender, and age will be uploaded to the server.

## 1.5 CONSTRAINTS

---

The new system must be able to run in a cloud environment for iOS/Android.

## 1.6 COST ESTIMATION

---

Resource	Cost
Poster	NZ\$30
Print	NZ\$100
iOS Developer	NZ\$130
Labour	48 Hours per week - 12 per member



---

## 2 Skills - Knowledge - Roles

---

### 2.1 ROLES

Name	Role	Description	Skills
David Gutla	SCRUM Master. Development Team Member, Developer and Tester. Cross functional member available to do anything required, with current or learnable skills	SCRUM Master who facilitates the SCRUM Process for the Development Team and Client. Documentation, Planning, Implementation. Cross functional member available to do anything required, with current or learnable skills	SQL, PHP, HTML, CSS, Java, SCRUM ,Documentation
Jozella Roque	PROJECT MANAGER Development Team Member, Developer and Tester.	Overall Documentation, Planning, Implementation. Cross functional member available to do anything required, with current or learnable skills	SQL, Java, Documentation, HTML, SCRUM
Kai Shao	Development Team Member, Developer and Tester.	Documentation, Planning, Implementation. Cross functional member available to do anything required, with current or learnable skills	Java, SQL, PHP, CSS, HTML, SCRUM
Aaron Morpeth	Development Team Member, Developer and Tester.	Documentation, Planning, Implementation. Cross functional member available to do anything required, with current or learnable skills	Java, SQL, PHP, CSS, HTML, SCRUM

## 2.2 REQUIRED SKILLS

Skill	Present in Team	Justification	Resource to Acquire Skill
Xcode		As Xcode is used to develop iOS, we will need to learn this software	Tutorials lynda.com
Swift/Objective C		iOS runs on swift/Objective C, therefore it must be learned	Tutorials lynda.com
UI Design		UI has to be appealing and simple to use for customers to use, else they may not use the application	Tutorials lynda.com
SCRUM Methodology	All	As this project uses SCRUM; its knowledge must be increased	PMBOK scrumalliance.org
Github	Somewhat	Must be used for continuous integration	Tutorials lynda.com
Agility		Must be able to adapt to change incase to sudden changes made by Product Owner	PMBOK scrumalliance.org
Conflict Resolution		All members must be notified if conflict exists as it will raise problems in the future if not dealt with	PMBOK scrumalliance.org
Time Management	Somewhat	Time must be managed so all allocated tasks are completed on time, as individual members and as a whole	PMBOK

## 2.3 RESOURCES

Resource	Justification
Trello	Trello will be utilized as a place to track the group's progress and discuss up and coming deadlines.
Github	In order to work effectively as a team, we set up a GitHub page to be able to share and update our project remotely.

Swift/Objective C	To be able to program from iOS platforms
Google Plus	In order to collaborate effectively we will use the communication median google plus to discuss project agendas on a face to face basis.

---

## 3 Project Plan

---

### 3.1 APPROACH

---

Various methodologies are available, however only a limited number of methodologies are relevant to be undertaken for this type of project. All the methodologies that are pertinent to this project fall under the 'Agile' category. The Methodology that is chosen for this project is the Agile 'SCRUM' Methodology.

### 3.2 AGILE

---

Agile is a time-boxed, iterative approach to software delivery that builds software incrementally from the start of the project, instead of trying to deliver everything all at once near the end. Agile works by breaking down the project into smaller pieces of functionality called 'User Stories', prioritising them, and consistently implementing them in short cycles called 'Iterations'. Most methodologies that fall under the 'Agile' category share almost the same ideas and practices. However each methodology does have its own particular formula of practices, terminology, and strategies to make themselves distinct from one another.

#### 3.2.1 AGILE MANIFESTO

---

The agile manifesto will give a formal proclamation of four key values and twelve principles to guide an iterative and people-centric approach to software development which will be followed in this project.

##### Four Key Values

Individuals and Interactions over processes and tools

Working Software over comprehensive documentation

Customer Collaboration over contract negotiation

Responding to Change over a following plan

## Twelve Principles

Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.

Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.

Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.

Business people and developers must work together daily throughout the project.

Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.

The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.

Working software is the primary measure of progress.

Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.

Continuous attention to technical excellence and good design enhances agility.

Simplicity--the art of maximising the amount of work not done--is essential.

The best architectures, requirements, and designs emerge from self-organising teams.

At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behaviour accordingly.

## 3.2.2 SCRUM

SCRUM is a lightweight project management framework that falls under the 'Agile' category. SCRUM consists of three Artefacts (Product Backlog, Sprint Backlog, Burndown Chart ), Sprints (Iteration(s) that are time-boxed to greater than 1 week, but less than 4 weeks), Ceremonies (Sprint Planning Meeting, Backlog Grooming, Daily Stand Up Meeting, Sprint retrospective), and Roles (Product Owner, Scrum Master, Development Team).

### 3.2.3 SCRUM ROLES

Role	Description	Applied
Product Owner	The Product Owner will be responsible for managing the Product Backlog	Elise
SCRUM Master	Mentors and facilitates the team with the SCRUM process.	David
Development Team	Has a goal of producing a product increment during each iteration.	Development Team

### 3.2.4 SCRUM ARTEFACTS

Artefact	Description
Product Backlog	A prioritised list of requirements with estimated times. It includes: <ul style="list-style-type: none"><li>• Documentation updates</li><li>• Bug fixes</li></ul>
Sprint Backlog	A list of tasks, almost the same as the product backlog, but it will have a finite number of Items to complete
Burndown Chart	Will help illustrate progress done by the team. It will be daily re-calculated to indicate total remaining task hours within one sprint. It will show the <u>Work Remaining Across Time</u> in a sprint.

## 3.2.5 SCRUM CEREMONIES

Ceremony	Description
Sprint Planning Meeting	<p>A sprint planning meeting will be conducted in the beginning of each sprint cycle, it will involve the Product Owner and Development Team.</p> <ul style="list-style-type: none"><li>• Plan on what is going to be done</li><li>• Create Sprint Backlog</li></ul>
Daily SCRUM Meeting	<p>Daily meeting time boxed at 15 minutes. During the SCRUM meeting, our team will discuss to each individual member:</p> <ul style="list-style-type: none"><li>• What did I accomplish since the last meeting?</li><li>• What shall I do today to help each other meet our goals</li><li>• Is there any impediment that may prevent me, or the development team from reaching the sprint goal?</li></ul>
Sprint Review	<p>A Sprint Review will be held after each sprint, a demonstration will be shown unto the Product Owner and the stakeholder. The team and the Product Owner will then discuss all the opportunities, constraints and findings. Only completed product functionality can be demonstrated.</p>
Sprint Retrospective	<p>During this meeting, the development team will reflect and review their behaviour from the just-concluded sprint</p>
Backlog Grooming	<p>Conducted before the sprint, this meeting has a purpose of helping the Development Team better understand the User Stories given by the Product Owner</p>

## 3.2.6 PLANNING POKER

Planning Poker in SCRUM is a method that will help the development team estimate User Stories.

At the start of Planning Poker estimation exercise, each member will be given Planning Poker cards. Each card represents an estimate.

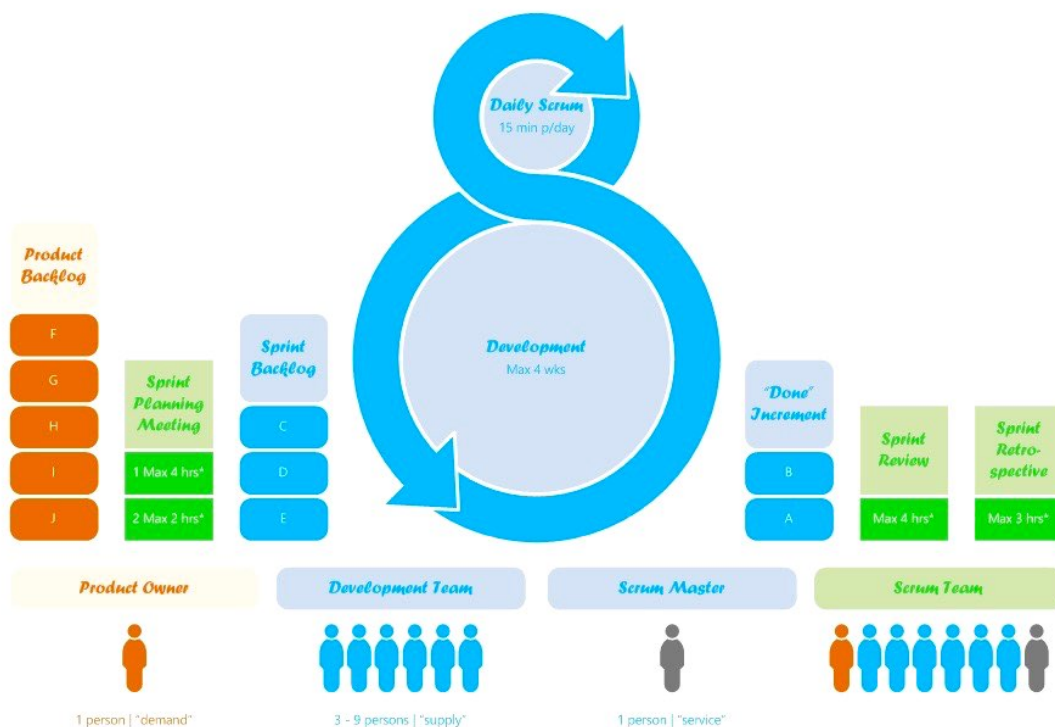
For each User Story to be estimated, a member will read a User Story out loud from the Sprint Backlog.

After listening and discussing, each member in the Development Team selects a card in secret, and depending on the thought of difficulty for a User Story, that will be their choice.

All member will then reveal their cards and discuss the differences/similarities. All member will move to next User Story if all the estimates are the same

Else each member will re-estimate. Often, the estimates will converge by the second round. If not, the process is repeated until the team agrees on a single estimate to use for the User Story.

## 3.2.7 SCRUM OVERVIEW





## 3.2.8 SCRUM PHASE OVERVIEW

Phase	Process
Initiation	<ul style="list-style-type: none"><li>• Create project scope</li><li>• Identify people involved in the project</li><li>• Gather SCRUM Team</li><li>• Create Product Backlog</li><li>• Prioritise Backlog</li><li>• Establish Functional and Non-Functional Requirements</li><li>• Create User Stories from Epics</li><li>• Create Acceptance Tests</li></ul>
Planning	<ul style="list-style-type: none"><li>• Develop Project Plan</li><li>• Create Sprint Backlog</li></ul>
Estimation	<ul style="list-style-type: none"><li>• Play 'Planning Poker' to estimate the User Stories from the Sprint backlog.</li><li>• Allocate tasks to members</li></ul>
Implementation	<ul style="list-style-type: none"><li>• Daily SCRUM Meetings</li><li>• Update the Burndown Chart on a daily basis</li><li>• Update and maintain the backlog</li><li>• Develop (writing test code, writing product code, reviewing, testing)</li></ul>
Sprint Review	<ul style="list-style-type: none"><li>• Demonstrate finished product increment</li></ul>
Sprint Retrospective	<ul style="list-style-type: none"><li>• Retrospect sprint</li></ul>
Release	<ul style="list-style-type: none"><li>• Ship the deliverables</li></ul>

### 3.3 PROJECT PROCESSES

Process	Action	Justification	Agile Principle
1 <sup>st</sup> Phase - Initiation			
Create Project Scope	Understand and create the project scope with the client.	No scope means wasting necessary time and effort on something that may not need to be done	Our highest priority is to satisfy the customer through early and continuous delivery of valuable software. - helps understand the outcome
Sprints	Sprint length of two weeks	Two week time box pushes the team to develop an increment more frequently	Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
Establish Functional and Non-Functional Requirements	Establish the requirements from the Product Owner	The requirements are essential for the team to understand the goals/purpose	Business people and developers must work together daily throughout the project.
Obtain Epics from the Product Owner	Obtain the 'big picture' stories from product owner that is to be broken into smaller pieces	Many problems will arise in the project if the product owner does not give us overviews	Our highest priority is to satisfy the customer through early and continuous delivery of valuable software. - A clearer understanding will help the overall process
Create User Stories from Epic	Based on the Epics, create User Stories and Acceptance Tests with client	To give a clearer, more accurate, and simpler story.	Our highest priority is to satisfy the customer through early and continuous delivery of valuable software. - Unambiguous User Stories will help process the project through
Develop Project Plan	Establish Project Plan	Develop Project Plan to monitor and manage the project	Business people and developers must work together daily throughout the project.

Process	Action	Justification	Agile Principle
<b>2<sup>nd</sup> Phase - Planning and Estimation</b>			
Approve User Stories	Do not proceed if User Stories cannot be completed within Sprint	Problems arise when too many user stories taken that cannot be completed during the Sprint. Too many User Stories taken that cannot be completed will take a longer time	Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale. - Reduce time by increasing understanding
Ambiguous User Stories	Discuss unclear stories with Team and Product Owner until they are clear.	Working on unclear User Stories will result in mistakes and an dissatisfying product increment	Business people and developers must work together daily throughout the project. - Communicate and discuss concerns about ambiguous User Stories
Prioritise User Stories	Prioritise tasks by means of the Product Owner	Without discipline priorities, there is no order and all user stories will be a jumble.	Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale. - Prioritise help order the work and manage time before hand
Establish Product Backlog	Create a Product Backlog with all User Stories	A project can not be guided without a Product Backlog	Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale. - Product Backlog holds the project structure
Create Acceptance Tests	Create acceptance tests to confirm User Story completion	Without acceptance tests, there is no way to confirm User Stories	Working software is the primary measure of progress. Acceptance tests will help measure progress of User Story completion

Estimate User Stories (Planning Poker)	Estimate the User Stories by playing Planning Poker	Without estimates, delivery and workload times are unknown; therefore too much/less workload may be chosen	Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale. - Estimation will provide approximate times, and Sprint workload can be managed well
Incorrect Estimation	Re-estimate and review wrong estimates as to why they are wrong, and learn from them	To have accurate measures (workload, delivery times)	Our highest priority is to satisfy the customer through early and continuous delivery of valuable software. - Faster delivery through better understanding
Establish Tasks	Establish tasks, and allocate the tasks to team members	So tasks are assigned and members are conscious of what they are doing	Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done. - Allocate tasks
Create Sprint Backlog	Establish a Sprint Backlog consisting of all tasks that are to be completed in the Sprint	Work to be done during Sprint is unknown if a Sprint Backlog is not created	Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale. - Sprint Backlog outlines work to be done during Sprint, ready to implement

Process	Action	Justification	Agile Principle
3 <sup>rd</sup> Phase - Implementation			
Develop (create, test, refactor)	Develop given tasks, usability test, and Acceptance Test	It is mandatory to test and refactor at all times to improve code overall. All acceptance and usability tests must pass	Working software is the primary measure of progress.
Daily SCRUM Meeting	Conduct Daily SCRUM Meeting	Frequent meetings must be made to inform each other what is currently happening. However Due to ongoing work, a minimum of one day a week is sufficient.	Business people and developers must work together daily throughout the project.  At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behaviour accordingly.
Update Burndown Chart	Update Burndown Chart	The Burndown Chart is used to measure progress, in order to see what has been and needs to be done.	Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale. - The Burndown Chart helps measure progress, and the estimated time for completion
Maintain Sprint Backlog	Update/maintain	Keep the User Stories in understanding with the group	Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale. - Maintenance helps keep track of the total Sprint Backlog Process

Process	Action	Justification	Agile Principle
<b>4<sup>th</sup> Phase - Review and Retropect</b>			
Sprint Review Meeting	Demonstrate Sprint	Showcase the what has been developed so far in the sprint to the Product Owner. Feedback is vital.	Working software is the primary measure of progress.  Business people and developers must work together daily throughout the project.
Ship Product Increment	Deliver validated product increment	Deliver Product Increment to the Product Owner	Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
Sprint Retrospective	SCRUM Master and Development team meet to discuss sprint.	Review the just concluded sprint, discuss what went well/wrong so that the learned outcome can be used for next time	At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behaviour accordingly.

Process	Action	Justification	Agile Principle
<b>5<sup>th</sup> Phase - Release</b>			
Release Deliverables	Commit all changes to the product	Release, so that the product can be utilised	Working software is the primary measure of progress.

## 3.4 JUSTIFICATION

---

For this given project, we have decided to use the SCRUM Methodology as it best suits this project. This approach is well suited towards this project because it fulfils the criteria:

- Teams that are newly formed
- Client wants something creative, if any new ideas appear, we may have to be adaptable to change
- Customer collaboration is required for upcoming changes, it allows us to work together
- Shorter sprints allow us to fulfil the work designated in the given time-box
- We want a method of constantly getting feedback from the client
- Progress in the methodology can easily be measured via Burndown Chart
- Less documentation which results in more productive work

## 3.5 SCHEDULE

Task	Start	Finish
Personal Log Book	20/07/2015	24/06/2016
Sprint Zero	23/07/2015	12/08/15
Initiation Phase	21/07/2015	12/08/2015
Upskilling	17/08/2015	14/09/2015
Sprint One	15/09/2015	29/09/2015
Sprint One Review	30/09/2015	06/10/2015
Sprint Retrospective		
Sprint Two	09/10/2015	31/10/2015
Mid Project Review	12/10/2015	16/10/2015
Sprint Two Review	25/10/2015	31/10/2015
Sprint Retrospective		
Sprint Three	03/11/2015	17/11/2015
Sprint Three Review	19/11/2015	25/11/2015
Sprint Retrospective		
Upskilling	26/11/2015	01/02/2016
Sprint Four	03/02/2016	25/02/2016

## 3.6 UP SKILLING

Task	Duration	Start Date	Members
Github	3 days	11/08/15	Development Team
Xcode	14 days	31/08/15	Development Team
Swift	14 days	17/08/15	Development Team
iOS Web Integration	14 days	17/08/15	Development Team
UI Design	14 days	26/11/15	Development Team



## 3.7 SPRINT PLAN

Task	Duration	Day
Sprint planning meeting	1.5 hours	First Tuesday (first day in the sprint)
Scrum meeting	0.25 hour	Second Tuesday
Scrum meeting	0.25 hour	Third Tuesday
Implement	16 hour	Starts at the first day ,ends 1 day in advance
Sprint review	1 hour	Last day of the sprint
Commit changes	1 hour	Last day of the sprint
Sprint retrospective	1 hour	Last day of the sprint
Backlog grooming	1 hour	Last day of the sprint
Total	22 hours	

---

## 4 Quality Assurance

---

### 4.1 QUALITY CONTROL

---

In order to control the quality of the project the development team will review what has been done at the end of each sprint, the quality of the product will be compared with the requirements. If discrepancies are found they will be discussed amongst the team in order to find a resolution.

#### 4.1.1 REQUIREMENTS

---

- The development team will generate a list of requirements to guide each development team member. Unclear requirements should be discussed during the meeting
- The development team must be adaptable to any changes made by the client
- Requirements will be obtained by getting Epics by the Product Owner, which then will be broken down into User Stories

#### 4.1.2 VERIFICATION & VALIDATION

---

Assess the requirements to identify key performance areas of software. All members must check the function completion and verify the outcome.

Cycle through Acceptance Tests for performance boundaries under stress conditions. Document Acceptance Test Results and plan the documentation of test tasks.

#### 4.1.3 UI TESTING

---

The entire Development Team must always test the code for each allocated section. Usability tests must also be done by the client to check if it is user friendly and simple.

## 4.1.4 INSPECTIONS AND ACCEPTANCE TESTING

---

Nearly complete activities that require inspections and/or acceptance testing will be specified in work documentation. Acceptance tests are from the user's perspective and the external perspective of the system. Acceptance tests give acknowledgment to confirm how the condition of something changes.

Acceptance Testing should be conducted when necessary to verify that all required processes perform as planned.

## 4.1.5 CODING

---

All code should be legible for easy error finding and understanding. All code must have its documentation and in code commenting for quality improvement.

## 4.1.6 DOCUMENTATION

---

All members in the Development Team must proof read and check to see if the documentation is readable.

## 4.1.7 MEETINGS

---

- Meetings must occur at least once a week so that it give the Development Team Members the opportunity to discuss their current progress. Meetings are compulsory.
- Client must be met at least once every month to make sure everything is going according to what they initially wanted.

## 4.1.8 USER STORY DEFINITION OF DONE

---

- All acceptance criteria for the User Story are met
- Code when required is annotated, legible, and is fully functional in accordance to the User Story.
- Functional acceptance testing is performed and tried by a team members (other than the sole creator of the code).
- A complete backup is made to the entire system.

## 4.1.9 SPRINT DEFINITION OF DONE

---

- All User Stories included in the sprint are done
- All tests (that are applicable) including Acceptance Tests successfully pass
- No critical bugs exist in the final version
- A complete backup is made to the entire system

### 4.1.9.1 USABILITY CRITERIA

---

#### Definition

User Friendly	Easy to use from the Development Team's and client's perspective
Fast	Should not make the User irritate at loading times

### 4.1.9.2 PEER REVIEW

---

- Current progress needs to be reviewed by all other Development Team members. Peer review must occur at least once a week. The time before or after SCRUM meetings are recommended.
- Members being reviewed should make changes according to the review result.
- Face-to-face meetings are ideal for peer reviews, but it is acceptable to do peer review online if there are no exceptions.

### 4.1.9.3 CLIENT REVIEW

---

- Current progress of all members needs to be reviewed by the client at least once a month. Ensure the Development Team's progress meets the client's desired outcomes.
- The development team must make changes according to the results from the review.
- Recommended before or after client meetings.

## 4.2 DELIVERABLES

Phase	Deliverable	Description
Initiation	<ul style="list-style-type: none"><li>• Project Proposal</li><li>• Requirements</li></ul>	The entire plan for the proposed project
Planning and Estimation	<ul style="list-style-type: none"><li>• Product Backlog</li><li>• Acceptance Tests</li><li>• Sprint Backlog</li></ul>	<p>The Product Backlog contains the list of deliverables for the product</p> <p>The Sprint Backlog contains the list of User Stories that are to be completed during the sprint</p>
Implementation	<ul style="list-style-type: none"><li>• Product Increment</li></ul>	The increment completion for each Sprint
Review and Retrospect	<ul style="list-style-type: none"><li>• Functional Product Demonstration</li></ul>	Demonstration gives the Product Owner and opportunity to see and give feedback on the Product. Retrospect meeting will let the Development Team discuss the success/failures of the just concluded Sprint
Release	<ul style="list-style-type: none"><li>• Fully Functional Product</li></ul>	The full completed product

## 4.4 COMMUNICATION PLAN

All four team members have agreed to meet at least once a week face to face. Twice if there is urgency. All team members must attend, consequences are outlined in the team agreement. If team member cannot make a meeting, they must be updated by another team member via face to face, email or video conference. If a team member has a query about the project, they can contact their members via Facebook messenger or SMS for it is the most efficient medium for everyone. Trello is used to allocate work and track team members progress.

Meetings with the supervisor will be held every other week to discuss and review the project status. Team members can ask questions they may have about the project. All team members will attend this meeting.

Client meetings will be held once a month, twice a month if urgent, to discuss and update the project status. The team can ask the client about potential changes or issues. All team members will attend this meeting. One team member will be communicating with the supervisor and client via email or phone.

Type	Audience	Medium	Frequency	Purpose
Weekly meeting	All team member	Face to face	Twice a week	To review the status of the project and also to clarify work allocation
Supervisor meeting	All team member	Face to face	Once every 2 week	To review the status of the project. To see if project is on track and opportunity to ask questions about the project
Meeting time	All team members	Google drive	After each meeting	To update team members of what happened in the meeting and how many minutes
Sprint planning meeting	All team member	Face to face	Once every 2 weeks before the sprint	Decide how work will be done

Sprint Review	All team members & client	Face to face	Once every 2 weeks after each sprint	Decide what the next step will be
Emergency meeting	All team member	Phone	When necessary	To review and discuss any urgent issues regarding to the project. That cannot wait for the weekly meeting

## 4.5 RISK MANAGEMENT

---

### 4.5.1 RISK LEVELS

---

Risk levels (High, Medium, Low) will be observed and reported through the project to the project manager.

### 4.5.2 RISK IMPACT

---

- High – Risk that has the potential to greatly impact project cost, project schedule or performance.
- Medium – Risk that has the potential to slightly impact project cost, project schedule or performance.
- Low – Risk that has relatively little impact on cost, schedule or performance.

### 4.5.3 RISK PRACTICES

---

- At all times a risk log created on google drive will be kept by the project manager and will be reviewed at the project weekly meetings.
- Every big risk will be connected to a project team member for observing, to check that the risk will not exceed the limits.



Risk	Category	Description	Potential Responses	Impact Score	Probability	Impact	Risk owner
Failed proposal	Assessment	Proposal didn't pass the presentation	Read proposal guide carefully. Proof read the entire proposal before handing in. Send it to supervisor before presentation	4	Low	Cost time for resubmission and presentation. May cause project delayed or failed.	Project Team
Team member dropped the paper	Team management	Team member dropped the paper	Split the work for the absent team member, contact university and supervisor.	3	Low	New member's join is possible. The team roles have to be reconsidered.	Project Team
Client cancelled the project	External	Client cancelled the project	Contact school for new project. If there is none, project group dismiss and wait for new project.	5	Low	Project cancelled.	Project Team
University changed supervisor for the project/ Supervisor not available	External	Project team have a new supervisor due to university or supervisor's problem.	Need to contact new supervisor for further meetings. Need to let new supervisor read all project documents. .	3	Low	Cost time for reading documents and contacting supervisor.	Project Team

Team Member Absent	Performance	Member of group being absent due to emergency or being sick	Keep him updated, post progress information on Trello.	2	Medium	Important idea/ information can be missed during meeting because of the absent.	Project Team
Data accidentally lost	Assessment	Losing data due to accident. E.g. Losing USB, software bug, blue screen	Redo the lost part. Keep an extra copy on google drive weekly. (Daily if necessary)	2	Medium	Lose progress, repeated work need to be done.	Project Team
Client changed project details. E.g. Requirements, scope.	Assessment	Client changed project details due to some reasons. E.g. Cost, changes of the market	Changes need to be made on project documents about the it. Need approval on tasks from client before continue development.	2	Low	Cost extra time. The size of the project may change. Project time need to be rescheduled.	Client
Conflict between team members	Cooperation	Disagreement between team members	One of the people having disagreement need to step back otherwise the project will be negatively effected.	2	Low-Medium	Cost extra time. Vital to project final product quality, may even cause project failure.	Project Team

Conflict between team member and client	Cooperation	Disagreement between team members and client	Explain to client the reason for disagreement, if they insist their opinion. Follow their decision.	2	Low	Cost extra time. Loss of cooperation may negatively effect project quality. Can cause the failure of the project.	Project Team
Conflict between team member and supervisor	Cooperation	Disagreement between team members and supervisor	Explain to supervisor the reason for disagreement, if he insist his opinion. Follow his decision.	2	Low	Cost extra time. Loss of cooperation may negatively effect project quality.	Project Team
Google Drive offline	External	Can not access to google drive.	Contact other team member and find a temporary substitute website	1	Low	Cost extra time. The project team may have argument while deciding the substitute website	Project Team

---

## 5 Glossary

---

Term	Definition
CSS	Cascading Style Sheets – a style sheet language used to describe the format of a document
GitHub	A web-based revision control hosting service for software development and code sharing (Rouse,2012)
HTML	Hyper Text Mark-up Language- a coding language used to create websites
iOS	An operating system manufactured by Apple
Java	A multi purpose programming language
Methodology	Defines how an organisation produces and delivers systems
Product Backlog	A document which outlines every requirement for a system, project or product (Scrum Basics, n.d)
PHP	A server-side scripting language used to develop web development
Rationale	A set of reasons for a course of action
SCRUM	An agile methodology
Sprint	A set period of time during which specific work has to be completed and made ready to review (Rouse, 2015)
Sprint Backlog	A list of tasks identified by the SCRUM team to be completed during the sprint (Cohn,n.d)
SQL	Structure Query Language – programming language used to create and manage databases

---

## 6 References

---

<http://www.versionone.com/agile-101/agile-methodologies/>

<https://www.mountaingoatsoftware.com/tools/planning-poker>

<http://www.agilemanifesto.org/principles.html>

Rouse, M. (2012, May 1). What is GitHub? - Definition from WhatIs.com. Retrieved August 6, 2015, from <http://searchdatacenter.techtarget.com/definition/GitHub>

Rouse, M. (2015, June 1). What is sprint (software development)? - Definition from WhatIs.com. Retrieved August 6, 2015, from <http://searchsoftwarequality.techtarget.com/definition/Scrum-sprint>

Scrum Methodology. (n.d.). Retrieved August 6, 2015, from <http://scrummethodology.com/the-scrum-backlog/>

Cohn, M. (n.d.). Topics in Scrum. Retrieved August 5, 2015, from <https://www.mountaingoatsoftware.com/agile/scrum/sprint-backlog>

Rouse, M. (2012, May 1). What is GitHub? - Definition from WhatIs.com. Retrieved August 6, 2015, from <http://searchdatacenter.techtarget.com/definition/GitHub>

Rouse, M. (2015, June 1). What is sprint (software development)? - Definition from WhatIs.com. Retrieved August 6, 2015, from <http://searchsoftwarequality.techtarget.com/definition/Scrum-sprint>

Scrum Methodology. (n.d.). Retrieved August 6, 2015, from <http://scrummethodology.com/the-scrum-backlog/>

Cohn, M. (n.d.). Topics in Scrum. Retrieved August 5, 2015, from <https://www.mountaingoatsoftware.com/agile/scrum/sprint-backlog>

---

# 7 Disclaimer

---

## **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.*

