Auckland University of Technology
Shuttle Ticketing System

Project Proposal

V1.0 - Printed 22/03/18

Version

Date	Version	Author	Notes
March 8th	0.1	Hayden Woodhead	Initial document layout and started contents
March 9th	0.2	Izaac Crooke	Project approach and more
March 10th	0.3	Hayden Woodhead	Added requirements
March 13th	0.4	Hayden Woodhead & Izaac Crooke	Continual work on requirements, project approach and more
March 14th	0.5	Dominic Porter	Added risk register
March 20th	0.6	Izaac Crooke & Hayden Woodhead	Final polishing and corrections
March 22nd	0.7	All members	Clarification of how balances/multi pass tickets operate
March 22nd	1.0	All members	Final polish

Client

Auckland University of Technology (AUT) Transport Department:

- Sonia Simpson - Associate Director Facilities Support: sonia.simpson@aut.ac.nz
- Sally Vallely - Parking & Transport Operation Coordinator: sally.vallely@aut.ac.nz

Team Members

Hayden Woodhead (15899353)
Networking & Security and Software Development
qsj6872@aut.ac.nz

Izaac Crooke (15897658)
Networking & Security and Software Development
xgx2741@aut.ac.nz

Dominic Porter (14883486)
Networking & Security and Software Development
qhw9948@autuni.ac.nz

Supervisor

William Liu: william.liu@aut.ac.nz

Executive Summary

The Auckland University of Technology provides shuttle bus services between their campuses. Currently, tickets are made of fragile paper, with no means of verifying their authenticity. We propose replacing this system with a digital system utilizing Quick Response Codes (QR codes). This change would reduce fraud, provide valuable business metrics, and increase driver efficiency. QR codes can be printed on existing paper tickets or displayed by our proposed passenger app.

We have had discussions with our clients and stakeholders and have put together a collective list of requirements which will structure our deliverables and the features of the system.

We hope to deliver a system that provides backend services (including an administrator management panel) and mobile apps. We also hope to deliver full documentation of the system, to ensure a simple and straightforward handover process.

Like all others, this project faces various risks including, but not limited to, loss of data, lack of knowledge, and internal politics preventing deployment.

We will follow the agile development methodology, consisting of three week sprints, with our first sprint being dedicated to planning and being only one week in length.

Our developers are currently highly skilled but will require minimal additional upskilling in our proposed languages and frameworks.

Terms of Reference

Background

The Auckland University of Technology stretches across three geographically distinct campuses in the Auckland region with a campus in Manukau (South Campus), Auckland CBD (City Campus) and Akoranga (North Campus). To link these campuses, AUT provides shuttle bus services in partnership with Pacific Tourways to allow staff and students to easily move between these campuses. Passengers are able to purchase business card sized paper tickets from the University Bookstore or Gym, located at each campus. These tickets currently provide no means of verifying their authenticity and can be easily forged. Passengers present these tickets to the shuttle driver who marks the card with a pen or punch a hole in it. Drivers also record passenger numbers using a paper and pen. Passenger numbers are tallied by Pacific Tourways and presented to AUT on a weekly basis. These numbers are collated by the Transport Operation Coordinator in a monthly report and presented to the Associate Director of Facilities Support. These numbers are then used by the Associate Director of Facilities Support to calculate total revenue for the shuttle system in an annual report.

Project Rationale

The current ticketing system, as outlined in the terms of reference, has three main disadvantages. Tickets have no means of verifying their authenticity and can be easily forged. Tickets are fragile, easy to lose, and create waste. The manual reporting of passenger numbers provides unreliable business metrics to the transport team. By moving to a digital system, we can virtually eliminate ticket fraud, reduce paper waste, simplify drivers' jobs, and provide up-to-date, insightful metrics, and automated reporting features.

Scope and Objectives

Project Goals

Our goal is to improve AUT's shuttle bus ticketing system by mitigating the various disadvantages of the current paper based system. We will do so by designing and producing a system that fulfils the requirements mentioned below, before December 2018.

Product Overview

To solve the issues aforementioned, we propose migrating to a digital system. Staff and students will be able to download a native iOS or Android™ app (henceforth referred to as the "passenger app") which will contain a virtual ticket represented by a Quick Response Code (QR code) . This virtual ticket links to a monetary balance in the backend system which may be topped up online or in app with a debit or credit card. Upon boarding the shuttle, the passenger will present their virtual ticket to the driver. The driver will scan this virtual ticket with their companion driver app. The cost (depending on the route) of the ride, will then be deducted from the passengers balance. Passengers will also be able to purchase physical tickets for each specified route with a QR code representing a single ticket or multi ticket pass. These tickets will also be scanned by the driver, and will have their remaining number of rides recorded in the system to prevent their use beyond their stated number of rides. By moving to a digital system, transport team members will be able to view live and historical passenger numbers, as well as produce reports on this data and several other useful functions, including update alerts and notifications.

Products

- Backend
    - Web management panel
    - Mobile app api
- Mobile apps
    - Passenger apps
        - iOS Passenger App
        - Android Passenger App
    - Driver Scanning App (Android)
- Documentation
    - Installation guide
    - System administrators' runbook
    - Instruction manual for transport administrators

Actors

- Passenger: a passenger is a staff member or student of AUT with an install of the passenger app or a paper ticket
- Driver: a driver is a staff member of a third party transport agency who drives the shuttles and performs scanning of tickets
- Administrator: an administrator is a member of the transport team who uses the management panel to gain insight into the ticketing system
- Ticketing system: the collection of backend services that allow the ticketing functionality

Product Requirements

Functional:

- Passengers shall be able to present their smart device displaying a ticket for the shuttle
- Passengers shall be able to view their balance in the passenger app
- Passengers shall be able to view the price of each route
- Passengers shall be able to view the number of rides their balance could purchase for each route
- Passengers shall be able to top up their balance using a debit or credit card, in-app or online
- Passengers shall be able to enable automatic top up of their balance once it falls below a specified balance
- Passengers shall be able to view a list of historical trips in the passenger app
- Passengers shall be able to dispute a historical charge
- Passengers shall be able to share a virtual ticket with another passenger
- Passengers shall be able to present paper tickets in lieu of a smart device
- Passengers shall be able to view current and historical alerts in the passenger app
- Passengers shall be able to view a current timetable in the passenger app
- Passengers shall be able to scan a paper ticket to show how many rides are remaining
- Passengers shall be able to login into the passenger app using AUT single sign on.
- Passengers shall be able to refresh their QR code when required

- Drivers shall be able to select their route from a list of routes in the driver app
- Drivers shall be able to scan tickets presented to them in order to admit passengers onto the service
- Drivers shall be able to login to the app using a unique driver identification code

- Administrators shall be able to view a dashboard showing statistics for the last week, month and year(s)
- Administrators shall be able to create named routes with differing prices

- Administrators shall be able to generate reports for historical passenger data between any two dates
- Administrators shall be able to generate any number of unique tickets in a print ready format
- Administrators shall be able to create alerts to be delivered to passenger devices
- Administrators shall be able to view all active disputes and provide appropriate compensation if the dispute is upheld
- Administrators shall be able to refund a passenger's top up

Non-functional:

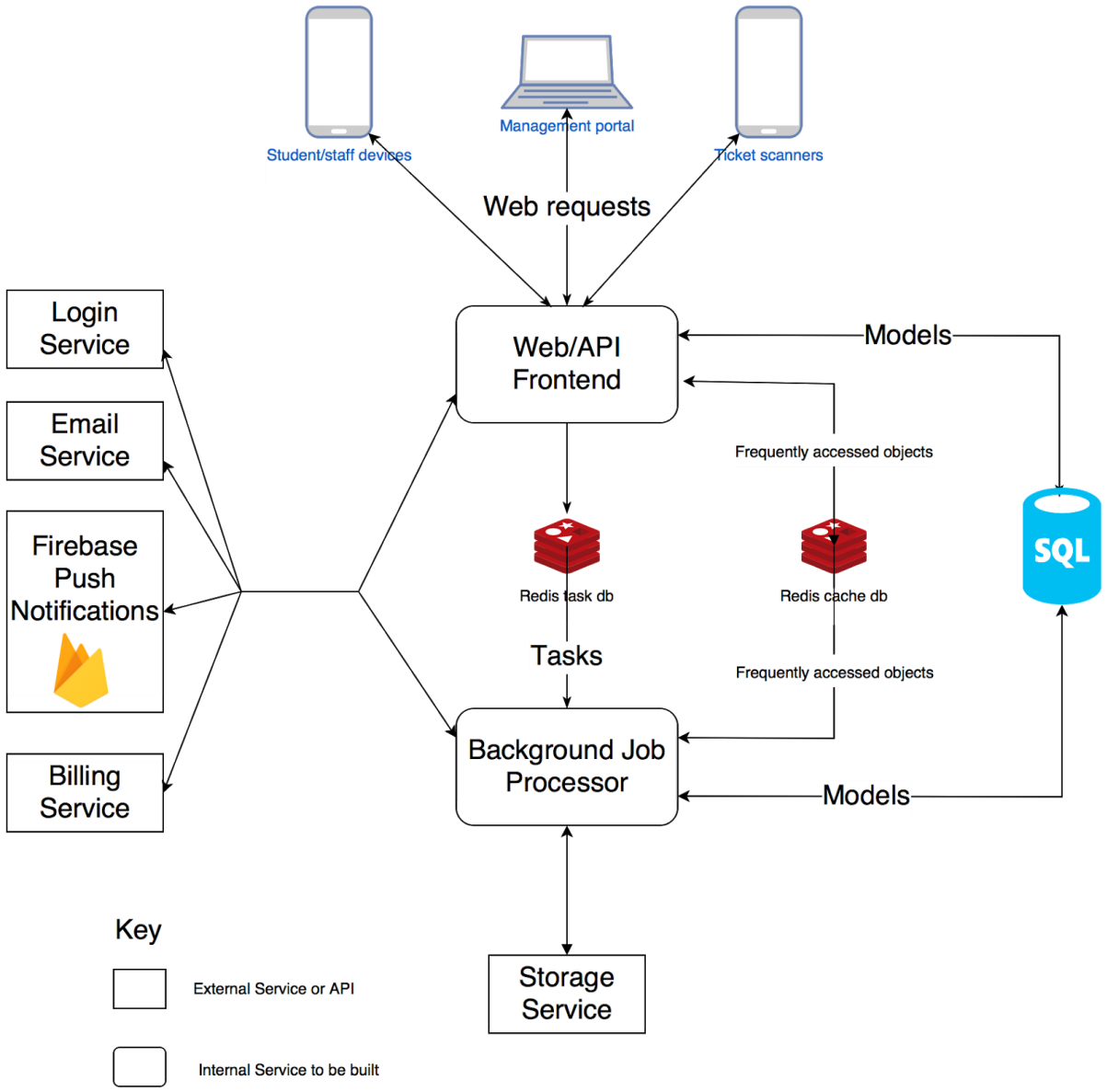
- The driver app must be able to verify passengers tickets without access to the internet
- The driver app must be able to record use of a ticket while offline to be synced once a connection is re-established
- Passenger apps must adhere to the clients brand guidelines
- Paper ticket generation must not use significant computational resources

- The ticketing system must provide idempotency for billing actions
- The ticketing system must provide idempotency for ticketing actions

Risk Register

No.	Risk	Description	Category	Triggers	Root Cause	Potential Responses	Risk Owner	Probability	Impact	Status
1	Lack of knowledge		Human Error	Being underprepared and lack of upskilling	Neglecting to upskill, learn about what is required from us		Everyone	Very Low	Very High	Unencountered
2	Loss of data/research	Loss of data via corruption, or just losing where things are stored	Data	Hardware failure, natural disaster, human error	Lack of hardware maintenance, industrial activity of modern civilisation, intellectual ineptitude	Backup all data	Everyone	Low	High	Unencountered
3	Bereavement leave	Close friends or family member passing away, causing employee to take bereavement leave	Medical	Death of loved one	The human condition, physical injury	Hire a temporary worker to cover the employee whilst they go on leave	Bereaved employee	Very low	High	Unencountered
4	Employee sickness	Employee getting an illness that requires time off work	Medical	Compromised immune system, superbugs, viral infection	The human condition	Hire a temporary worker to cover the employee while they go on leave. Assign work to another employee to cover lost productivity	Sick Employee	High	High	Encountered
5	AUT IT don't implement/ don't deploy the product	AUT IT don't continue and do a full scale deployment off what we have created	Politics	Underdeveloped, AUT IT unable (or unwilling) to takeover the project	Internal politics	Have a meeting with AUT IT to see what would be required to deploy this solution	Everyone	Very High	Very High	Unencountered

Architecture Diagram



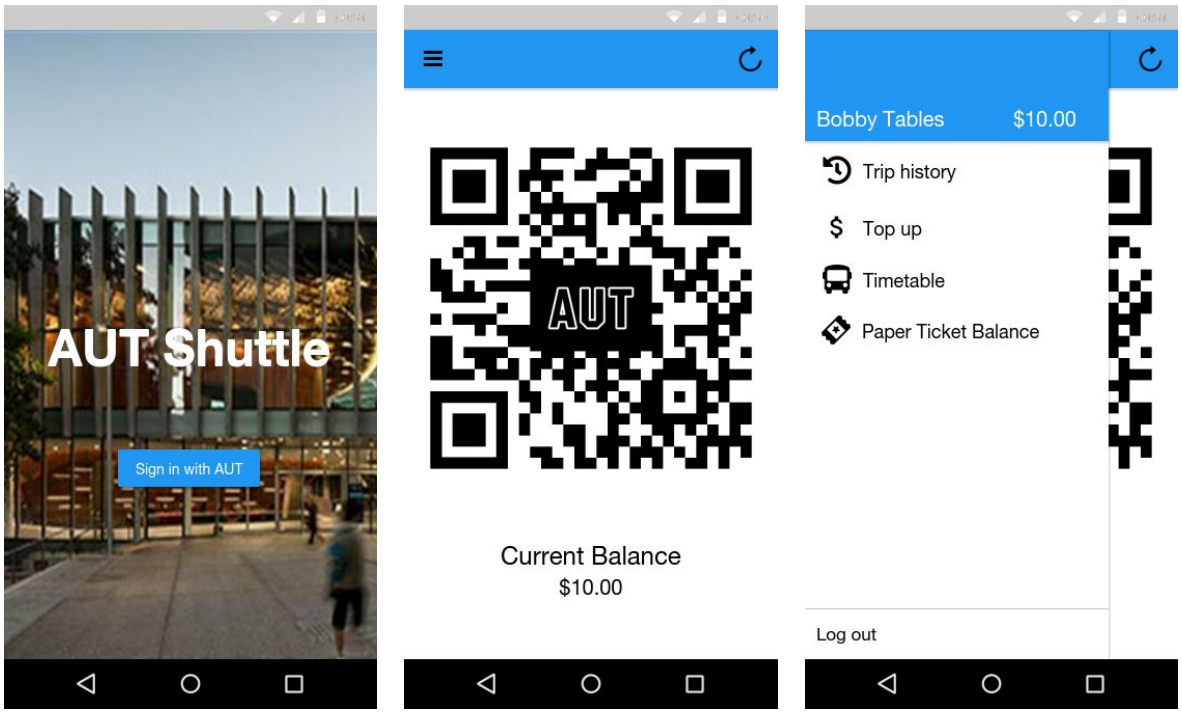
We propose using a relatively simple system design that lends itself to a language agnostic implementation. The web/api front-end receives all requests from devices and is responsible for returning valid Javascript Object Notation (JSON) and Hypertext Markup Language (HTML) back to the client. It directly accesses the databases (SQL and Redis) to manipulate models and save results. It also contacts several other services including login, email, and billing. The web front-end saves task directives to the redis task database, from which the background job processor pulls jobs. These jobs run longer than a standard HTTP request, allows and saves the results to the database layer, and may use other services to notify of successful completion of the job. The billing service is responsible for charging users debit or credit cards (this system may be provided by AUT or an external provider). The email system allows the system to send emails to users (this system will either be maintained by AUT or an external provider). The storage service will either be static files served by the server running the system or Amazon S3. Finally, the push notification service allows us to provide

instant alerts to passenger app users (this system is reliably maintained and provided by Google/Firebase).

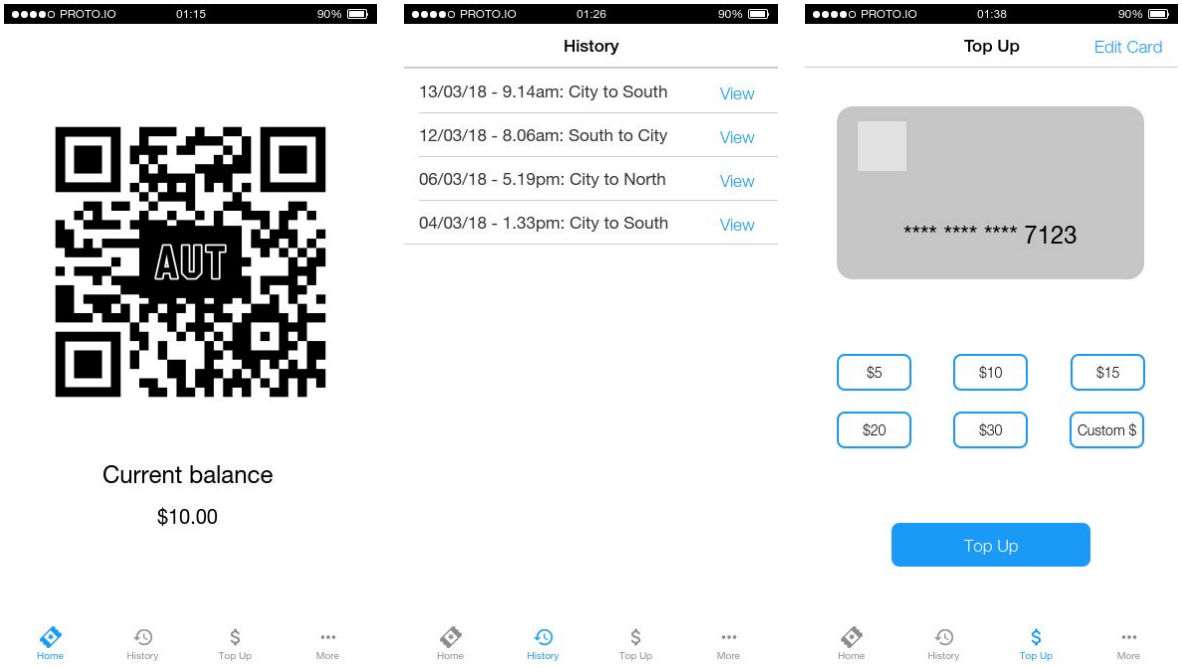
App Mockups

We have chosen to stick to a simple design language adhering to the individual platforms design guidelines. These apps are only mockups to give an idea of what's possible. Full client review and adherence to AUT's brand guidelines is still required.

Android App



iOS App


PROTO.IO 01:15 90%
AUT
Current balance
$10.00
Home History Top Up More
PROTO.IO 01:26 90%
History
13/03/18 - 9.14am: City to South View
12/03/18 - 8.06am: South to City View
06/03/18 - 5.19pm: City to North View
04/03/18 - 1.33pm: City to South View
Home History Top Up More
PROTO.IO 01:38 90%
Top Up Edit Card
**** **** **** 7123
$5 $10 $15
$20 $30 Custom $
Top Up
Home History Top Up More

Project Approach

Our chosen methodology for this project is Agile Project management because we have used this methodology in a prior development project and are proficient in following it. We have decided to undertake 3 week sprint intervals (with a 1 week sprint for planning) for the duration of this project. We have chosen this approach as we believe that as a collective group, 3 week sprints will optimize our group's development productivity, minimize trivial supplementary documentation, and more efficiently distribute our workload to achieve greater results.

Our main phase for the project will produce a shuttle bus ticketing system which allows quicker and easier access to ticketing and mitigate fraud. However, we also have a second phase which will integrate data tracking into the system. We hope to address this after we have successfully completed phase one and have satisfied our clients main requirements. Upon successful completion of phase one, we will submit a second proposal outlining our second phase.

Following the agile methodology, our deliverables will be influenced by progress in our sprints, as well as client and user feedback collected in a meeting after each sprint. These factors will influence what is to be worked on in the subsequent sprint.

Quality Assurance

We aim to produce high quality code. To achieve this we have created a set of rules and practices dictating how individual code is admitted to our codebase. A developer will select a feature or user story to work on, creating a new github branch. They will produce code which, as a rule, must contain comments where necessary, as well as python and java docstring information. Python code must adhere to the PEP8 style guide, java the Google style guide, and swift the linkedin style guide. Once the feature is complete the developer will write tests to verify the functionality created. Upon completion of the tests, the developer will submit a pull request. This pull request will trigger an automatic script which runs automated tests on the codebase as well as running linters and style guide checkers to ensure the code has no regressions and meets the style guides. Upon passing these automated tests, another team member will perform a manual review of the pull request. Only after this other team member has checked the pull request and accepted it will this new code be merged into the codebase.

Project Plan

Sprint 0: This sprint will consist of activities required to setup the project for agile development. We will use tools such as planning poker to give user stories appropriate story points to aid in timeframe estimation. Following planning poker, we will categorize user stories by priority to further structure our subsequent sprints. We then select a number of user stories to work on in our first sprint. Stories will then be assigned to developers for coding. This sprint will also conclude developer upskilling activities.

Sprint 1: This sprint will see the start of development of the project. We plan to begin working on user stories related to backend systems, with the creation of domain specific models and associated actions being our priority.

Sprint 2: This sprint is likely to be a continuation of backend development. Models are expected to be finalized by this point, with requirements such as ticket printing and metrics now being the priority. Due to our agile approach, it is difficult to predict the exact contents of this sprint.

Sprint 3 onwards: The details of all other sprints will be heavily dependant on feedback from clients, progress on particular deliverables, and other contributing factors. After each sprint, we will able to more accurately estimate and better plan for subsequent sprints.

Milestones

Week 4: 19 - 23 March: Project Proposal Due
Week 5: 26 - 30 March: Sprint 0
Week 6: 2 April: Start of Sprint 1 & Development Start
Week 11: 24 May: Status Report Due
Week 12: 28 May - 1 June: Mid-Project Review Interviews

Further R&D deliverable dates have not been published yet

23 - 26 October: Delivery of final product to client

Skills and Knowledge

While our client's IT department currently utilizes a Microsoft stack with development skills focused on C#, our system will instead utilize Python, Java and Swift. Our native iOS and Android apps will use their respective native languages (Java and Swift). There are alternatives such as Xamarin, which utilizes C#. However, Xamarin apps have greater complexity, lower performance, and are not compatible with the native libraries we require for QR code reading. Our backend will be built with Python, using the web framework Django. Django is a "batteries included" framework which significantly reduces developer load and contains many best practice solutions of common problems (such as session storage). By using Django, we believe we can produce a better quality product than if we were to build the backend in another language.

Skills Required

Software Development:

- Python
- Django
- Redis
- PostgreSQL
- SQL
- HTML & CSS (Bootstrap)
- JSON
- Java
- Android Development
- Swift
- iOS Development
- Restful APIs

Networking and Security:

- Amazon Web Services
- Linux Sysadmin
- HTTP

Current Skills & Knowledge

Hayden Woodhead:

- Python & Django
- Redis
- PostgreSQL and SQL
- HTML & CSS
- Linux Sysadmin
- Java and Android Development
- Amazon Web Services
- Restful APIs

Izaac Crooke:

- Python
- HTML & CSS
- SQL
- Java and Android Development
- JSON
- Restful APIs

Dominic Porter:

- PostgreSQL and SQL
- Linux Sysadmin
- Java
- Python
- JSON
- Restful APIs

As shown above, we possess a wide range of relevant skills required to undertake this project. Hayden has extensive experience with Python and Django, while Izaac and Dominic are proficient in Python. Both Izaac and Hayden are proficient in Android development, while Dominic is proficient in Java and will require some upskilling in order to be able to contribute to the Android app. All team members are proficient in JSON and RESTFUL apis. The only skills lacking from the team is Swift and iOS development skills. We do not expect any great difficulty in upskilling ourselves to the required level.

To upskill our Python and Django development we plan on using https://docs.python.org/3/tutorial/ and https://docs.djangoproject.com/en/2.0/intro/tutorial01/.

To upskill our Swift and iOS development when plan on using https://developer.apple.com/library/content/documentation/Swift/Conceptual/Swift_Programming_Language/TheBasics.html#//apple_ref/doc/uid/TP40014097-CH5-ID309 and https://developer.apple.com/library/content/referencelibrary/GettingStarted/DevelopiOSAppsSwift/

To upskill our android development we plan on using https://developer.android.com/guide/index.html

Upskilling is currently underway with estimated completion being before start of sprint 1.

Budget

Resource	Cost
Team labour	5 Hours Weekly (5*3 developers = 15) 15 * $50 per hour = $750 Per Week = $3,000
Development server	$10 per month
Supervisor labour	1 hour per week $65 per hour = $65 per week = $195
Travel	$80 per week = $320
Catering	$100 per week = $400

Total Monthly Cost: $3925

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, 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 it’s associated documentation), both in relation to their use and results from their use.