# Enming Zhang

### Auguest 2019

## Introduction

The Integrated Journey Planner is a web application which is designed to bring accurate time schedules and better travel experience for all the customers to plan their journey. It covers a real-time interface which will offer the latest information about timetables for the bus and the light rail. This project, Journey Planner Test Automation, aims to provide firm quality assurance for the static timetables which Journey Planner are currently using. Those timetables, present in General Transit Feed Specification (GTFS) format, need to be validated before it goes into production for public consumption. This project aims to automate the test scenarios for the validation process.

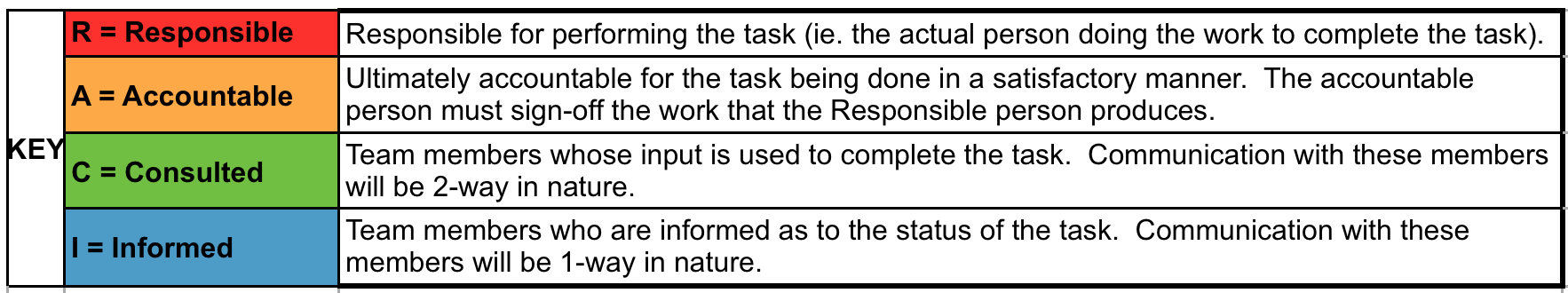
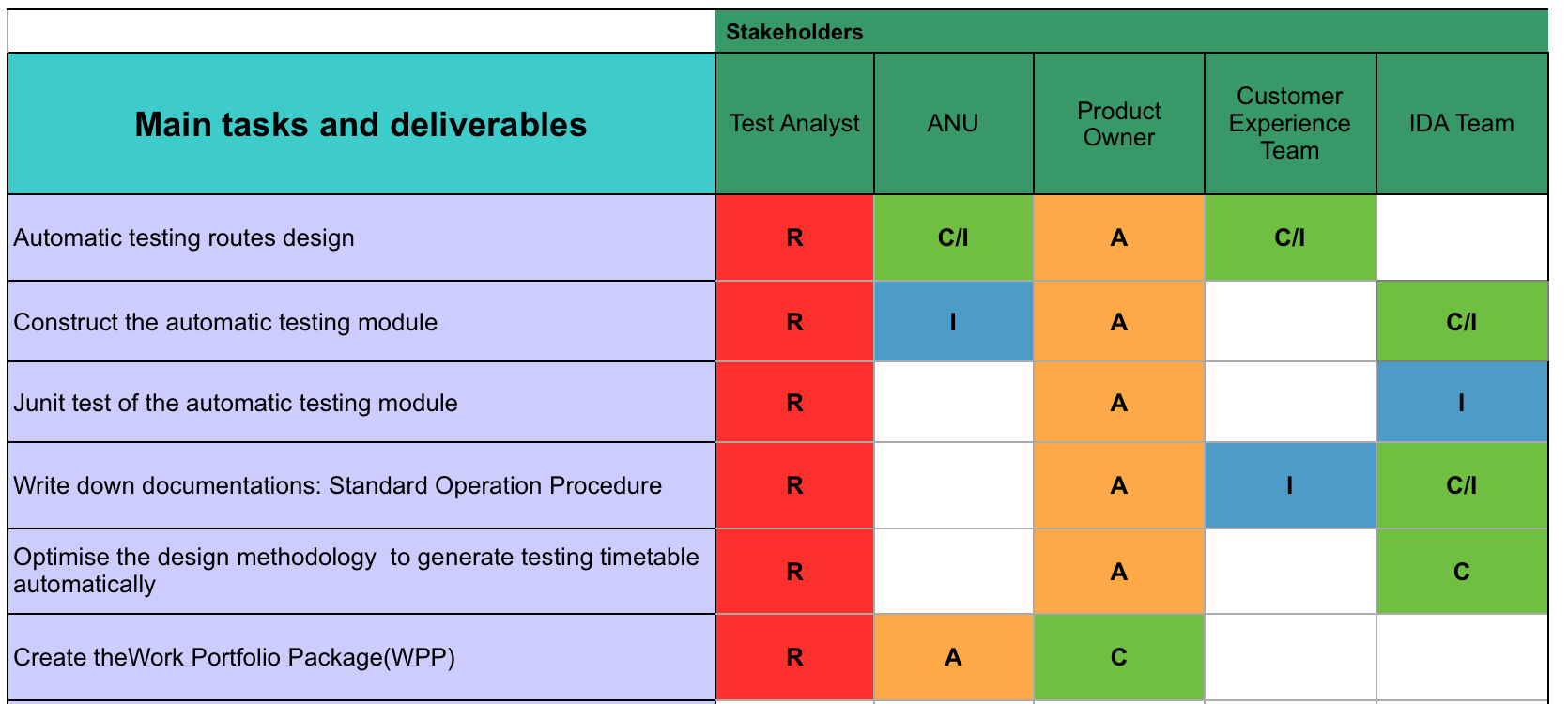
## Objective

* Find out a high effective way to test and improve the quality of GTFS data.
* Save time and cost for the testing team in Transport Canberra.
* Improve the customer experience for the customer service team.

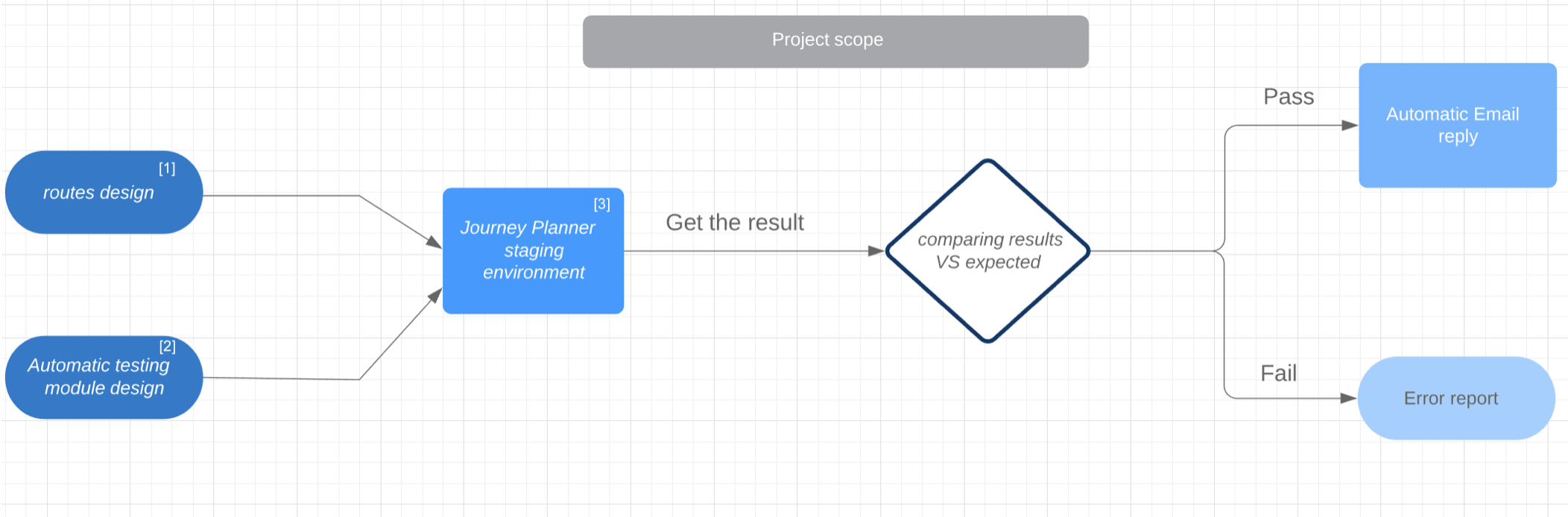
## Project Benefits

The benefits of this project are many. First, as mentioned above, the automatic testing module will provide a more effective way to find the wrong timetable from the mess GTFS data. It could reduce both time and economic costs for the Transport Canberra. Second, this project could improve the quality of the GTFS data that Journey Planner will use, which will help Journey Planner present more accurate timetables, and will lead to a better customer experience. In a word, this project could provide a guarantee for the product quality, and make contribution to a better user experience.

## Key Stakeholders



## Scope

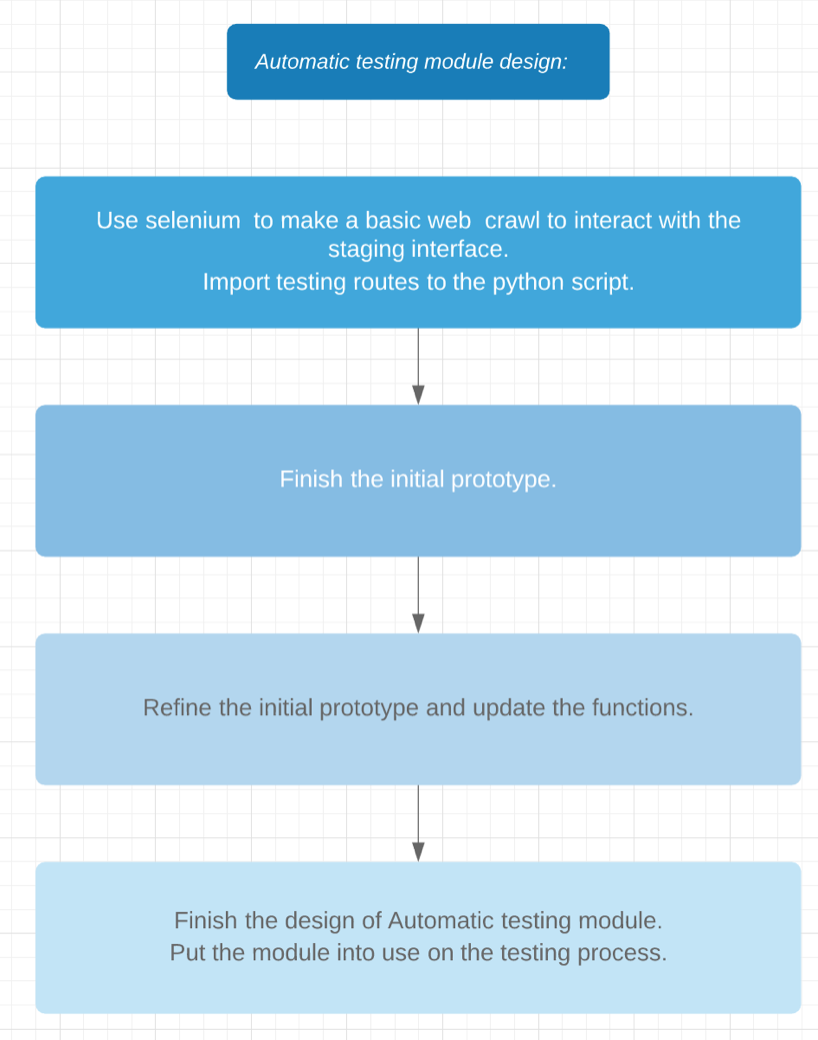


*1.0* The *overall scope of this project*

annotations:

[1]. the first stage of this project is about design both scenarios and the testing framework. We plan to design 20 routes covered different bus and light rail stops in workdays, weekends, and public holidays, to stimulate the expected customers’ usage. The expected result will also be settled down with the designed routes.

[2]. at the same time, the automatic testing module need to be built, tested and refined following the process listed below:



1.1 The design process of the automatic testing module

[3]. see appendix

After finishing the development of Automatic testing module, it needs to be put into use in the staging environment, and retrieve the result from the front end. Then the module will compare the retrieved result with the expected result. If all the comparisons could match, then a notification Email will be automatically sent to the IDA team, pointing out that the GTFS files in staging environment could be released into production. Conversely, a report of the error will be sent to the team to explain the problem.

## Setup Tools

| **Tool** | **Decision Making Process** |
| --- | --- |
| Python | It has the most powerful standard libraries for data analysis compared with other programming languages. |
| Selenium | It’s a suite of tools which owns a rich set of testing functions specifically geared to the needs of testing of web applications like the Journey Planner. |
| Bitbucket | Powerful version control tool used by the IDA team to upload codes. |
| Outlook email | Booking meetings and set up the working calendar. |

## Main Tasks:

The main tasks of this project are listed below:

1. Routes design  
   **Description:** The automatic testing module need to read a file which contains 20-25 random routes with equal weight. These routes will be spread out and cover different districts and bus stops in Canberra. Besides, since the testing mode need to stimulate real user behavior while using the map, the travel date and time also need to be designed and distributed on public holidays, workdays and weekends, to ensure the test is complete.
2. Construct the automatic testing module  
   **Description:** Design the automatic testing module using python and the web testing tool called selenium. This module will be able to read the file, record all the routes and their expected results, then stimulate users’ behavior in the staging environment, get the actual results and compare them with the expected results. After the comparison, this module should also send notification email automatically to the project owner, to send an error report or just inform that all the test have been passed.
3. Junit test of the automatic testing module  
   **Description:** Write Junit test codes to test different functions of the automatic testing module. Refine the python script based on the test result. This task is driven by the test analyst and approved by the project owner.
4. Documentation  
   **Description:** Since this automatic testing module need to be put into production, A Standard Operating Procedures (SOP) will be needed. In the SOP, the scope and usage of the code and the designed routes will be well documented.

## Milestone Schedule

**Phase 1** **Project Initiation**

**Purpose** To explore the main tasks in this project and be familiar with the working environment, understand the project scope and set up working plan.

**Activities**

Project start meeting

Discover the basic use of the working tools

Set up working plan

**Deliverables**

Initial working schedule

**Phase 2** **Basic Function Implementation**

**Purpose** To build an infrastructure of the testing module and finish the statement of work

**Activities**

Finish the statement of work

Build a draft script using python and selenium

Implement some basic functions in the script

**Deliverables**

Statement of work

Draft script of the automatic testing module

**Phase 3** **Initial Prototype**

**Purpose** Present the initial porotype of the testing module

**Activities**

Finish the initial automatic testing prototype

Design five sample testing routes

**Deliverables**

Initial prototype of the testing module

**Phase 4** **Optimization and Documentation**

**Purpose** Refine the testing module and improve the efficiency, write SOP for the module.

**Activities**

Refine the prototype

Design and validate other routes, then add them into the module

Check all the comments and write SOP

**Deliverables**

Prototype after refining

Standard Operating Procedures (first edition)

Progress and acceptance

**Phase 5** **Project Closure**

**Purpose** Finish the project and present the final achievement

**Activities**

Wrap all the content and deliver it to the host

Finish the Work Portfolio Package (WPP)

**Deliverables**

The automatic testing module

Standard Operating Procedures (final edition)

WPP

## Deliverables (for the project)

* Automatic Testing module
* 20-25 standard testing routes
* Standard Operation Procedure

## Cost and Resources

*List any resourcing requirements including; people, software, hardware or datasets and note any associated costs for each.*

**1. Resources:**

* Python 3.7.4 documentation (2019, August 04). Retrieved from: <https://docs.python.org/3/>
* Selenium documentation (2019, July 01). Retrieved from: <https://www.seleniumhq.org/docs/index.jsp>
* Learn Git with Bitbucket Cloud. Retrieced from: <https://www.atlassian.com/git/tutorials/learn-git-with-bitbucket-cloud>

**2. Potential costs:**

Since all the libraries,data and tools involved in this project are free open source tools, there’s no potential cost that need to be considered so far.

## Risk, Issues or Constraints

**1.Risks:**

* Using git in an inappropriate way may cause data lost or break the module after an update.
* Faulty testing routes and module may lead to an opposite effect on the quality of GTFS data.

**2. Constraints:**

* Conflictions between different Python versions:

Some libraries and tools are incompatible with the Python version 3.7.

Solution: Install Python 2.7 and run the tools first, at the same time try to find the alternative tools.

* Operation system issue:

At the working place, we need to work on the laptop provided by the host. As a Mac user it will take me some time to get familar with the Windows system.

Solution: Added self-teaching process about learning basic operations on Windows system on schedule, to make sure this won’t impede the working efficiency.

……………………………………….

Signature of Supervisor

……………………………………….

Print name

## Appendix

**Journey Planner Staging Environment:**

A staging environment (stage) is a space created for software testing. It contains a copy of all the information like codes, database and catche, which makes it a nealy exact relica of the real application. The Journey Planner Staging Environment here refers to the staging environment prepared for the Journey Planner. All the tests will be runned in this environment.

**General Transit Feed Specification (GTFS) format:**

According to the Google Transit API, The General Transit Feed Specification (GTFS) defines a common format for public transportation schedules and associated geographic information. It’s wildly used in many journey planner tools to store the data of routes, stops and timetables.

Reference:

Staging environment, Rouse M, Retrieved from: https://searchsoftwarequality.techtarget.com/definition/staging-environment

Google Transit APIs. Retrieved from: <https://developers.google.com/transit/>