**Glyndwr Timetable SatNav Test Plan**

**Prepared By:**  
Connor Laurence  
(Date)

**TABLE OF CONTENTS**

1. INTRODUCTION

The product – Glyndwr Timetable SatNav – is an android application that has two major functions. The first is searching, selecting, and browsing a timetable for the university to find information and the second is an indoor map allowing user localisation, routing and wayfinding through the university. Both components are linked directly though one element – navigating by a event selected from a timetable.

1. OBJECTIVES AND TASKS

2.1. Objectives

- Integration test of directly linked components between timetables and indoor map using a use case test

- Unit Testing of; Timetable Web Scrape, Timetable Contents Web Scrape, Search Timetables, Timetable Selection, Week Selection, Event Selection

- Decision Tables with two routes to ensure only allocated rooms can be navigate too (One route for an Online lecture and one for a room-allocated module)

- Equivalence Testing on timetable search

- Data-Flow Testing of local SQL database in its varying functions using Logcat

2.2. Tasks

Pre-testing phase:

* setup decision tables for two specific routes
* Insert Logcat messages into each aspect of data flow within application regarding local SQLite database
* Insert Logcat messages in units specified to demonstrate successful unit function

Testing phase:

* Screenshot successful and unsuccessful functioning units
* Document and screenshot results of decision table, equivalence testing and data-flow testing

Post-testing phase:

* If there are bug issues, document and rectify immediately
* Attach test plan to project document appendix and document appropriately in Chapter 8: Testing

3.0. SCOPE

In this project, five types of testing (four of which are black-box whilst the last is white-box) are included. These have been chosen to ensure that enough aspects of the product are thoroughly tested whilst not being too burdensome with regards to time.

All aspects of this testing will be undertaken by myself, and all tests should be completed within the week following end of sprint two. Any smaller tests incomplete by that stage must be discarded and documented within the testing stage to stay on target with planning.

1. Testing Strategy
   1. Alpha Testing (Unit Testing)

Unit testing will test separate units in turn such as; web scraping, timetable search, timetable selection, week selection and event selection. These will be done primarily through logcat messages demonstrating a successful test as well as UI changes which demonstrate a successful transaction.

4.2. System and Integration Testing

An integration test fully tests aspects from initially clicking a timetable through to navigating to that destination as a user may do and will be successful if the user can get to that end destination through that means. The two decision tables similarly act like integration but follow two set paths and end at the maps activity or in the event dialog depending on whether the room is allocated or not respectively.

4.3. Data Flow Testing

Data-flow testing uses an in-depth knowledge of the system to log data flow through Logcat messages at each point a flow of data may occur with the SQLite database. A successful test will show what data is moving and is backed up by being viewed in the UI i.e., Searching timetables interacts with data and the logcat results should match the search results in the UI.

4.4. User Acceptance Testing

Equivalence testing will be undertaken on the timetable search feature and will deliberately make erroneous searches like ‘modul’ instead of module or typing in numbers, searching for nonsensical strings, or trying to search for multiple different things like module and a room name. Successful results should ensure no crashes occur or functionality does not break in this unit. A blank result simply shows no results matching that string could be found and is not indicative of a failure.

4.7. Beta Testing

Following direct software testing by myself, the app will be given out to a select number of users to test and submit a user survey. The design of which comes from the design elements in Chapter 6 of the project document under ‘Survey Design’.

1. Hardware Requirements

Android Smartphone using Android 7.0+ (Nougat) or higher

Computer

Internet Connection

1. Environment Requirements

For 4.1 through 4.4, the environment is based at home with the workstation. The workstation requires Android Studio 3.5, an active internet connection and a functioning android emulator and log console, furthermore a tool like Windows snipping tool or Gyazo is necessary to screenshot elements throughout the test that ought to be documented. No special test tools are needed as all tools used within the development process can be reused to undertake this testing.

For 4.7 though, the environment changes to the university building and as such, a secure online storage tool for storing the app’s apk to later retrieve is necessary, a standalone device supporting android 7.0 or higher and a connection to Glyndwr University Wi-Fi network ‘eduroam’ is necessary at all times.

1. Test Schedule

The testing phase is initially earmarked for Tuesday 20/7/21 until Monday 2/8/21, however, this is subject to change depending on whether sprints are altered, or aspects of design or development over run.

1. Control Procedures

If problems are encountered, an incident will be documented in a table also attached to the appendix of the main project document. Not many incidents are anticipated due to its smaller scale development scope, so a table indicating incident number, description, proposed fix and confirmation of fix should be all that is necessary to satisfy this component.

|  |  |  |  |
| --- | --- | --- | --- |
| Incident No. | Description | Proposed Fix | Was it fixed? |

1. Features to Be Tested

Timetable list, timetable search, info dialog, week spinner and load button, week contents list, contents dialog, SQLite database, Route to room POI, user location

1. Features Not to Be Tested

General wayfinding to selected POIs, AR, accessibility routes as not enough time is allotted to also cover these aspects and they generally do not directly tie in with the research goals.

1. Resources/Roles & Responsibilities

All aspects, roles and responsibilities of software testing is for me with the exception of partial roles for users in the beta testing phase.

1. Schedules

* Test Plan – in main project document’s appendix
* Test Cases – in main project document – Chapter 8
* Test Incident Reports – in main project document’s appendix
* Test Summary Reports – in main project document – Chapter 8

1. Dependencies

The major dependency in this test plan is time constraint given the short timeframe of the proposed project. Also, with limited resource and it is just me undertaking the testing. Getting enough user testers is also a constraint as limited users may signify inconclusive results.

1. Risks/Assumptions

|  |  |
| --- | --- |
| Risk | Mitigation |
| Not enough time | Alter test plan to test for less use cases, less in-depth data flow testing to suit time available |
| Not enough users for survey | More qualitative questions in survey and extract what results can be had from available data |
| Survey and user testing affected by pandemic |  |

1. Tools

Android Studio

Windows Snipping Tool

Microsoft Office – Word

Google Sheets