University of Edinburgh

School of Informatics

Software Engineering Large Practical

Documentation

Author:
Blake Hawkins

Lecturer:
Stephen Gilmore

December 7, 2013

1 Introduction

This document aims to describe the usage of Timetabler, an Android application developed as part of the Sotware Engineering Large Practical course. Timetabler is desined to display data about a student's lectures in a clear and readable format, using XML data provided by the school of Informatics. This document contains four main sections: Installation, Testing Screenshots, Notes and Diagrams, and Additional Features

2 Installation

2.1 For End-Users

Timetabler is a standard Android application written in Java and XML which runs on your device without necessarily rooting or unlocking it. The only major difference between Timetabler and traditional Android apps is that it is not available on the Play store. Hence the end-user must install it using the APK (Application Package) file.

2.1.1 For Android 2.3 and earlier

Step 1

Copy /bin/timetabler-app.apk to your device's local storage.

Step 2

Enable unknown source installation by navigating to Settings>Applications, and enabling "Unknown Sources".

Step 3

Using a filesystem manager like Total Commander, navigate to your copy of timetabler-app.apk on the mobile device, and open it, accepting any warnings and prompts.

2.1.2 For Android 4.0 and later

Step 1

Copy /bin/timetabler-app.apk to your device's local storage.

Step 2

Enable unknown source installation by navigating to Settings>Security, and enabling "Unknown Sources".

Step 3

Using a filesystem manager like Total Commander, navigate to your copy of timetabler-app.apk on the mobile device, and open it, accepting any warnings and prompts.

2.2 For Developers

Timetabler is designed to target Android KitKat (4.4), but contains ActionBarActivity extensions and other support library features to support a minimum android version of 2.2. Timetabler is built using Eclipse with the android SDK installed (as opposed to Google's Android Studio). Hence, this document will describe how to set up the developer's environment under Eclipse.

Step 1

If not already available, install and configure the Android SDK Tools, Eclipse, and the Android Eclipse plugin per the instructions on the official SDK installation page.

Step 2

If not already available, install and configure the Appcompat support library following the instructions on the official support library setup page.

Step 3

-Import the project - open eclipse and select:

-File>Import>General>Existing Projects into Workspace, next

-Select root directory: <path to timetable project>

-Projects>Select All

-Uncheck both "Copy projects into workspace" and "Add project to working sets"

-Finish

3 Testing Screenshots

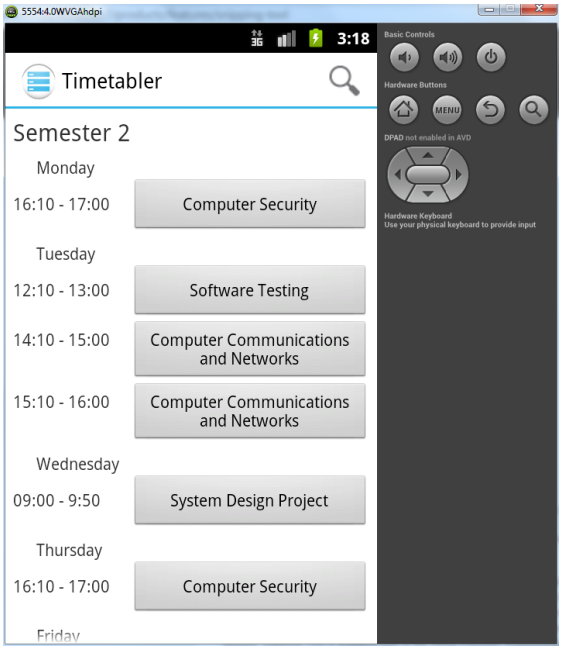

Figure 1: Main Viewer in a Gingerbread Emulator

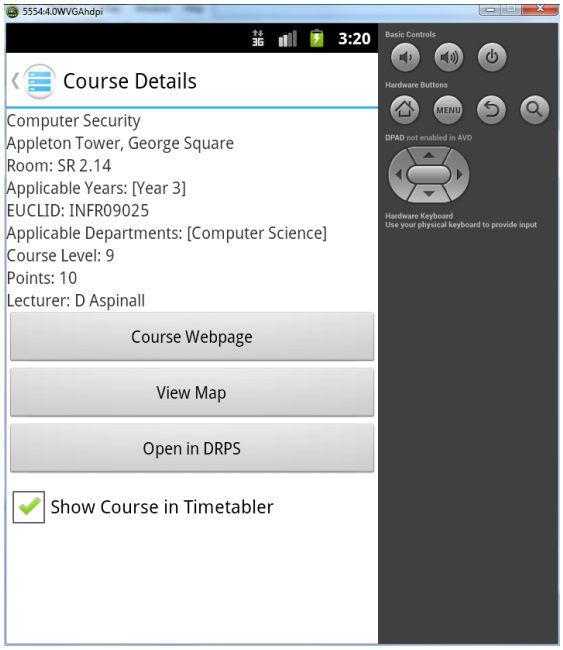

Figure 2: Course Details in a Gingerbread Emulator

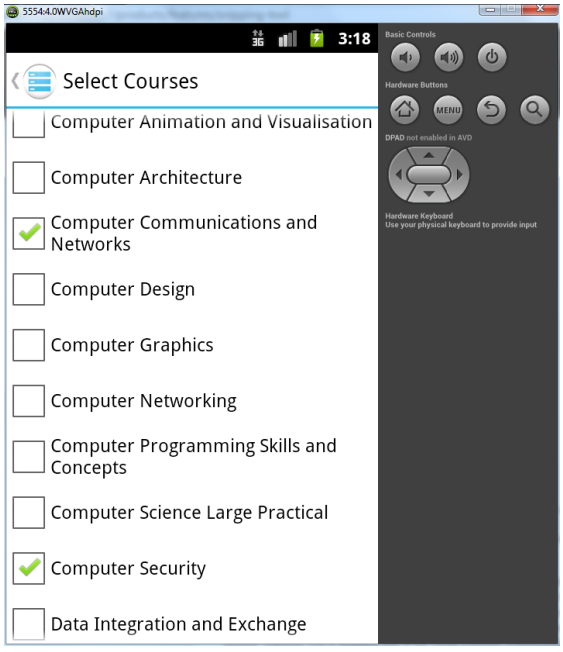

Figure 3: Course Selector in a Gingerbread Emulator

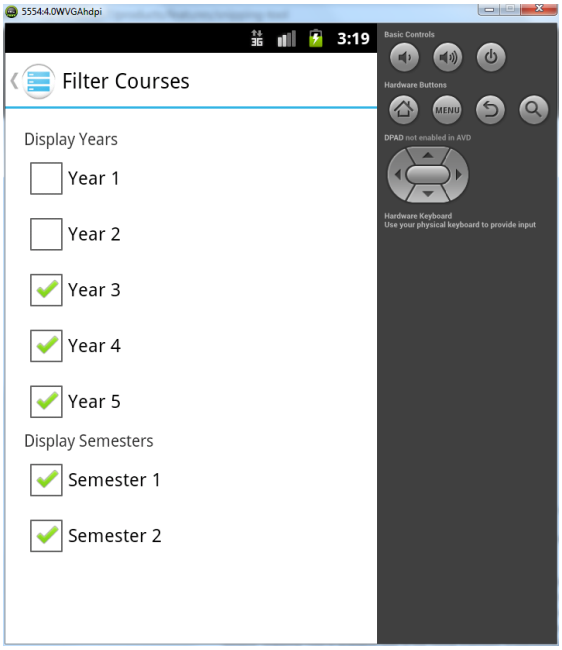

Figure 4: Course Filter in a Gingerbread Emulator

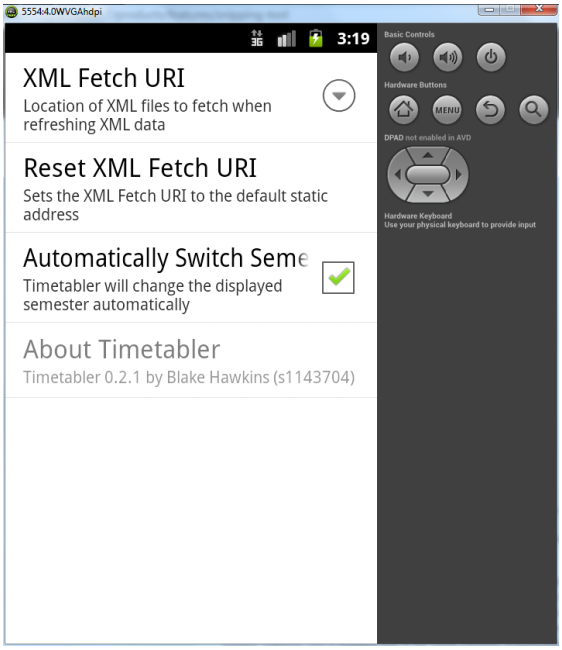

Figure 5: Settings Menu in a Gingerbread Emulator

4 Notes and Diagrams

The contents of Timetabler's code can be categorized into eight types.

Main Viewer

The main viewer activity is the primary activity that appears when Timetabler is launched - it is the only launcher activity.

Child Activities

From the main viewer, many child activities can be reached, including the course selector, course details, search results, and course filter.

Settings Activity

The settings activity is a special activity that extends PreferenceActivity - this automatically stores its views' contents in the app's local storage.

FetchThread

FetchThread is a special class that asynchronously requests data from a URI and feeds it to the application.

Managers

Two facade classes have been written for managing access to the application preferences and XML data, respectively. These classes provide static 'helper' methods for executing procedures that would otherwise be repeated across activities.

Parsers

To read and access the data in the XML files provided, parsers are written to convert text data to abstract datatypes.

Abstract Datatypes

To store and manage the data parsed from XML files, Venue, Lecture, and Course objects are written.

XML Descriptors

Besides the XML files that are parsed, there are also some XML files that aid in development as part of the android platform. These include strings, activity content descriptors, preference descriptors, and more.

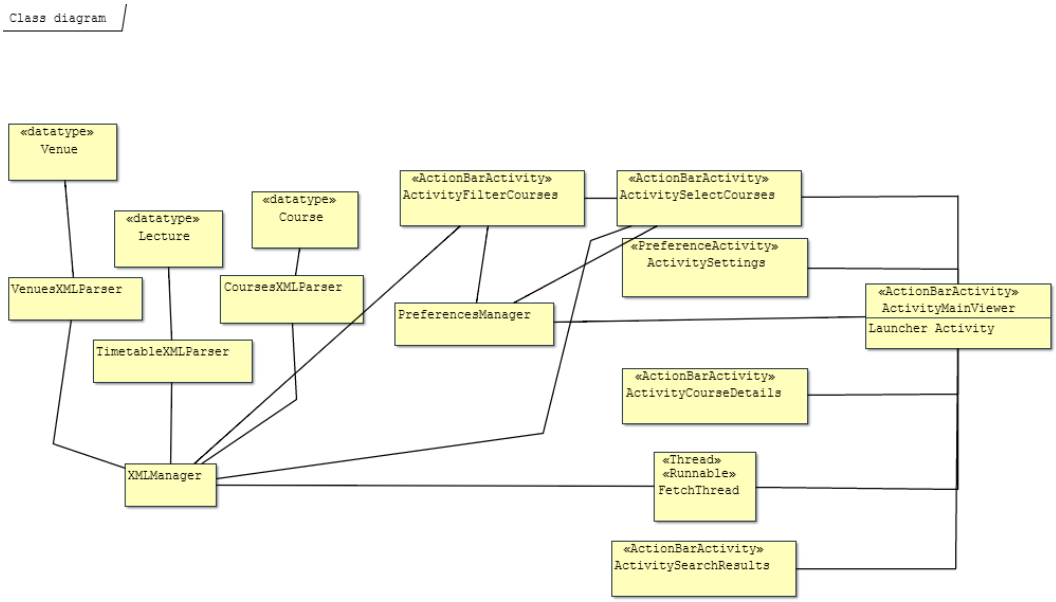

Figure 6: UML Class Diagram

5 Additional Features

Beyond the standard requirements for the application as part of the SELP course descriptor, some basic additional features have been added.

Compliant 4.x Support

By targeting Android KitKat (4.4) and using the appcompat library, Timetabler not only works in newer versions, but also matches their thematic standards and UI style.

Asset Management

Although Timetabler is designed to fetch XML data from a web URI, it also contains static assets of the XML files packaged, so it works "out-of-the-box" in case of limited mobile internet access

Configurable XML URI

While reading the course descriptor, I was disappointed to see that the required URI showed no hint of being permanant, nor being updated on a regular basis. To guard against similar issues, Timetabler contains a setting

for adjusting the XML URI. As part of the testing process, I suggest testing with the URI `http://blakehawkins.info/temp/xml-ext/`, which will remain online until at least February of 2014.