

All in 1 Windsor

Project Report



University of Windsor

School of Computer Science

Group 10
12-4-2015

Abstract

This report is an “Elaboration phase II” document. Here, the whole requirements of the software are programmed and tested, the majority of requirements are discovered and stabilized and the major risks are mitigated or retired.

This software is an android application which can be helpful in locating places like restaurants, malls, hospitals and worship places etc. This application is precisely engineered with respect of business case of “International students” arriving at Windsor city.

This multicultural App will be helpful in locating the maps and selecting a location which will show the distance from current location to destination and the different ways to choose the transport. It will also show bus routes wherever applicable.

Table of Contents

1. Introduction

- 1.1 Scope
- 1.2 Technologies to be used
 - 1.2.1 Hardware Interfaces
 - 1.2.2 Software Interfaces
 - 1.2.3 Memory Constraints
- 1.3 Dependencies

2. The Overall Description

- 2.1 Use Case Diagrams

3. Project Planning

- 3.1 Software Process Model
- 3.2 Project Plan
- 3.3 Project Scheduling & Code Repository
- 3.4 Risk Identification

4. Implementation

5. Testing

- 5.1 Black-box testing
- 5.2 White-box testing

6. Future Enhancements

7. Conclusion

1. Introduction

This software is an android application which can be helpful in locating places like restaurants, malls, hospitals and worship places etc. This application is precisely engineered with respect of business case of “International students” arriving at Windsor city.

This multicultural App will be helpful in locating the maps and selecting a location which will show the distance from current location to destination and the different ways to choose the transport. It will also show bus routes wherever applicable.

1.1 Scope

This android application allows user to get information about nearby restaurants, police stations, parks, shopping malls, etc. The business scope of the project is to target new students coming to Windsor, Canada for the first time. The main aim of the project is to make most services we can, available on the finger tips of the user.

1.2 Technologies to be used

1.2.1 Softwares Interfaces

Application Development Tools: Eclipse Mars (with ADT Plugin)

SDK: Android SDK

Languages:

Coding: Java

Deigning: XML

AVD: Genymotion (Free for personal use)

External Interface: Google Maps

Android version: API 18 (Jelly Bean) to API 22 (Lollipop)

1.2.2 Hardware Interfaces

Processor: ARM v7 Processor or above, 1GHz Minimum

RAM: Minimum 256 MB

HDD: 60 MB free space

GPS: GPS sensor

1.2.3 Memory Constraints

User should have at least 100 mb free space for the application to run smoothly and to install successfully minimum 50 mb free space is required.

1.3 Dependencies

The software is dependent on Google Maps application and GPS sensor in order to show full map functionalities.

2. Plot of Plan

2.1 Use Case Diagrams

Here the Use case diagram can be seen where the Actor selects an option and it leads either to the sub menu or maps (extending the menu).

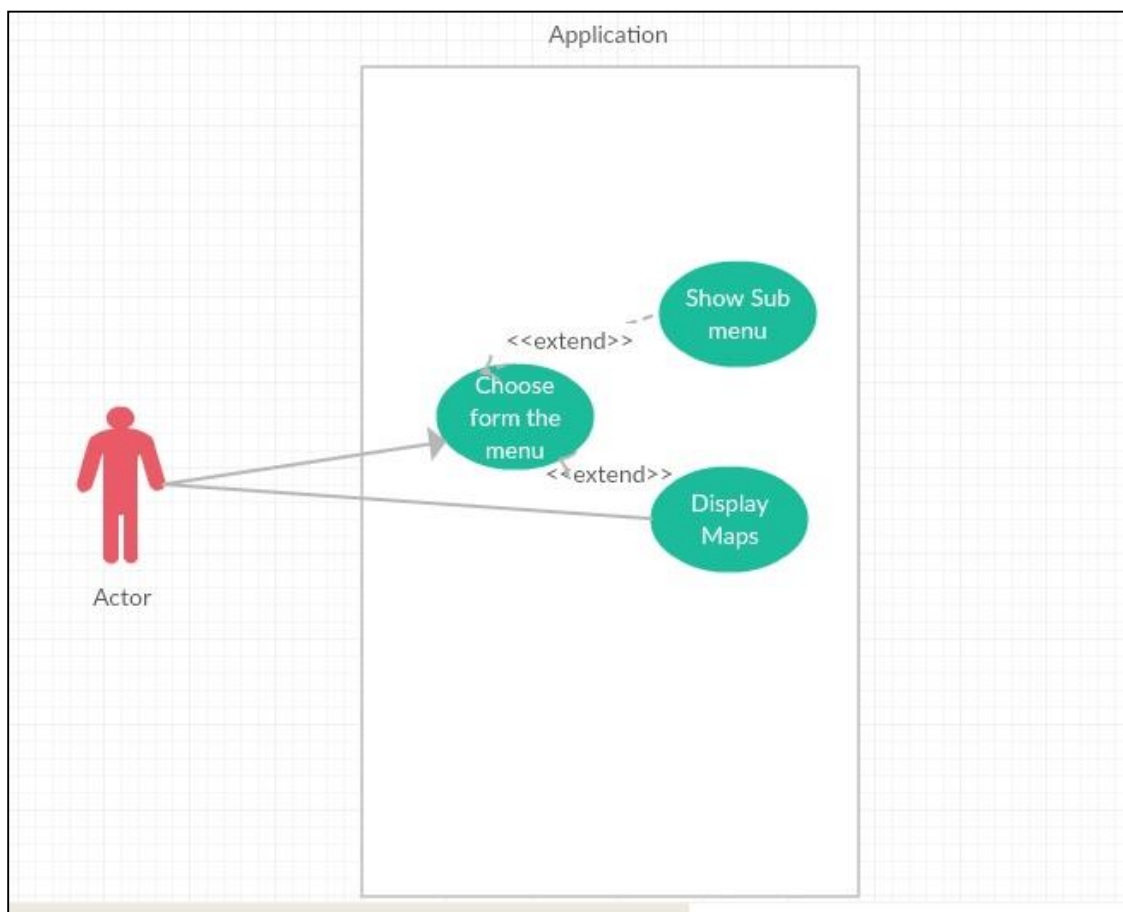


Fig 1. Actor interacts with Application which has different use cases.

3. Project Planning

3.1 Software Process Model

Agile Software Development is a set of software development methods in which requirements and solutions evolve through collaboration between self-organizing, cross-functional teams. It promotes adaptive planning, evolutionary development, early delivery, continuous improvement, and encourages rapid and flexible response to change. [Source: Wikipedia]

To solve actual problems in an industry setting, software engineer or a team of engineers must incorporate a development strategy that encompasses the processes, methods, tool layers and generic phases. This strategy is often referred to as process model or a software engineering paradigm. A process model for software engineering is chosen based on the nature of the project and application, the methods and tools to be used.

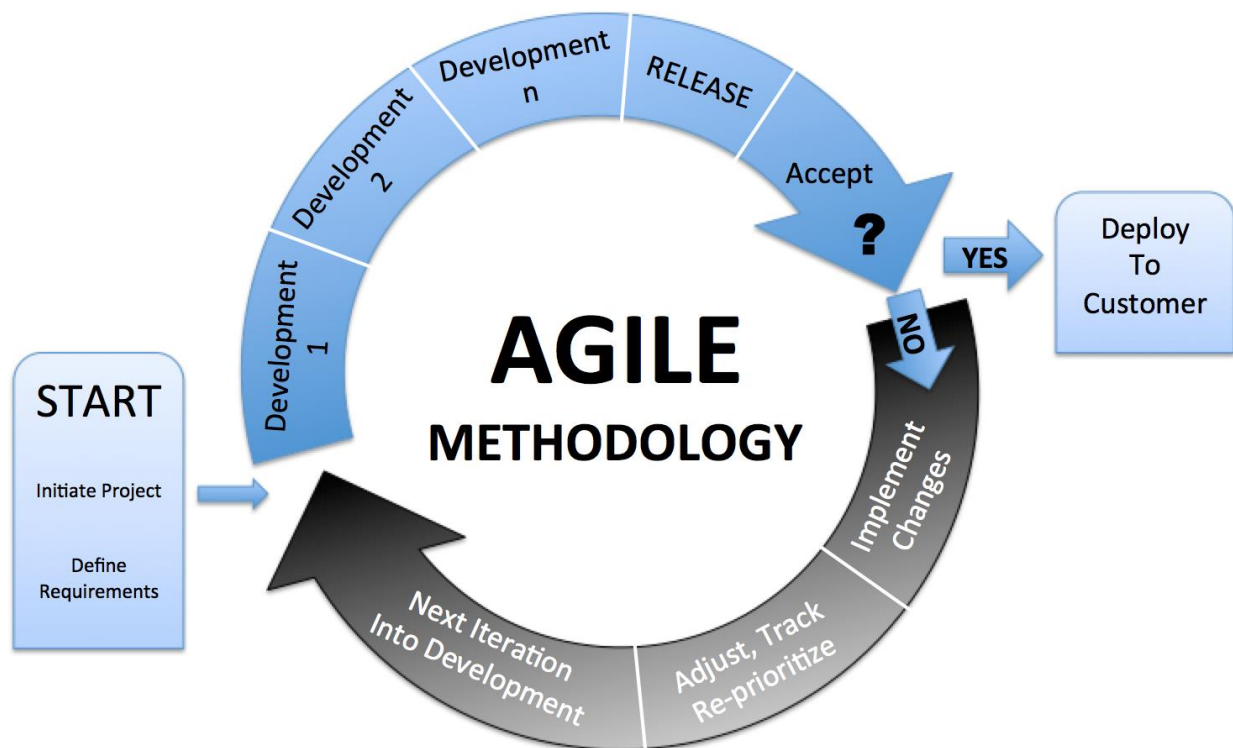


Fig 2. Agile methodology

[Source: <http://www.code4nord.com/wp-content/uploads/2015/09/Agile.png>]

3.2 Project Plan

In the development of this project, it was first checked if the project is feasible functionally, technically and economically. Then the requirements from the end users were collected and then analyzed.

3.3 Project Scheduling & Code Repository

We have made use of red mine for keeping track of the issues and distribution of tasks. It can be accessed via link: <https://redmine.cs.uwindsor.ca>

Code Repository: <https://github.com/DhawalRank/MultiCulturalApp>

The screenshot displays the Redmine web application interface. At the top, there is a navigation bar with links for Home, My page, Projects, and Help. The main header shows the project name 'Group-10 Multi Cultural App' and a search bar. Below the header, there are tabs for Overview, Activity, Issues (selected), New issue, Calendar, News, Documents, Wiki, and Settings. The 'Issues' section is active, showing a list of issues with columns for Status, Priority, Subject, Assignee, Updated, and Due date. The issues are filtered by Status (any) and are sorted by priority. The list includes 11 issues, all of which are 'Resolved'. The interface also includes a sidebar on the right with links for View all issues, Summary, and Calendar. At the bottom, there is a footer indicating the page number (1-11/11) and a note that the content is also available in Atom, CSV, and PDF formats.

#	Status	Priority	Subject	Assignee	Updated	Due date
1832	Resolved	High	Design and code Leisure Full Module	Dhawal Rank	11/25/2015 03:32 PM	11/25/2015
1831	Resolved	High	Design and code Shopping Full Module	Pooja Jain	11/25/2015 03:32 PM	11/25/2015
1830	Resolved	High	Design and develop public services	Shehraj Gupta	11/24/2015 04:52 PM	11/24/2015
1829	Resolved	High	Design and healthcare module	Mustafa Misri	11/24/2015 03:41 PM	11/24/2015
1828	Resolved	High	Design and code Food Cravings Full Module	Siddharth Khobare	11/24/2015 12:01 PM	11/24/2015
1827	Resolved	High	Design and code Worship Places Full Module	Dhawal Rank	11/24/2015 04:02 AM	11/24/2015
1826	Resolved	High	Designing main theme of the application	Dhawal Rank	11/24/2015 04:03 AM	11/23/2015
1788	Resolved	Normal	Redesigning of Logo	Mustafa Misri	11/12/2015 10:11 AM	11/11/2015
1715	Resolved	Normal	Phase 1 Report	Mustafa Misri	10/28/2015 04:50 PM	10/29/2015
1703	Resolved	Normal	Design logo for the app	Dhawal Rank	10/13/2015 01:49 AM	10/13/2015
1702	Resolved	Immediate	Interfacing Eclipse with Genymotion AVD	Dhawal Rank	10/12/2015 03:23 PM	10/12/2015

Fig 3. Red mine Screenshot.

3.4 Risk Identification

Risk Identification is a systematic and procedural attempt to specify threats to the project plan. These threats can be of many types as listed below.

There are two types of risks: Generic and Product Specific.

One method for identifying risks is to create a risk item checklist. The checklist can be used for identification and focused on some subset of known and predictable risks in the following generic subcategories:

Product Size

Risks associated with the overall size of software to be built or modified. In this case it is the risk associated with the defined web application to be built.

Business Impact

Risks associated with constraints imposed by management or the marketplace. But the application being developed as the final year project, the marketplace constraints are not too strict.

Process Definition

The risk of getting out of the line of definition. In other words, risks associated with the degree to which the software process has been defined and is followed by the development organization.

Development Environment

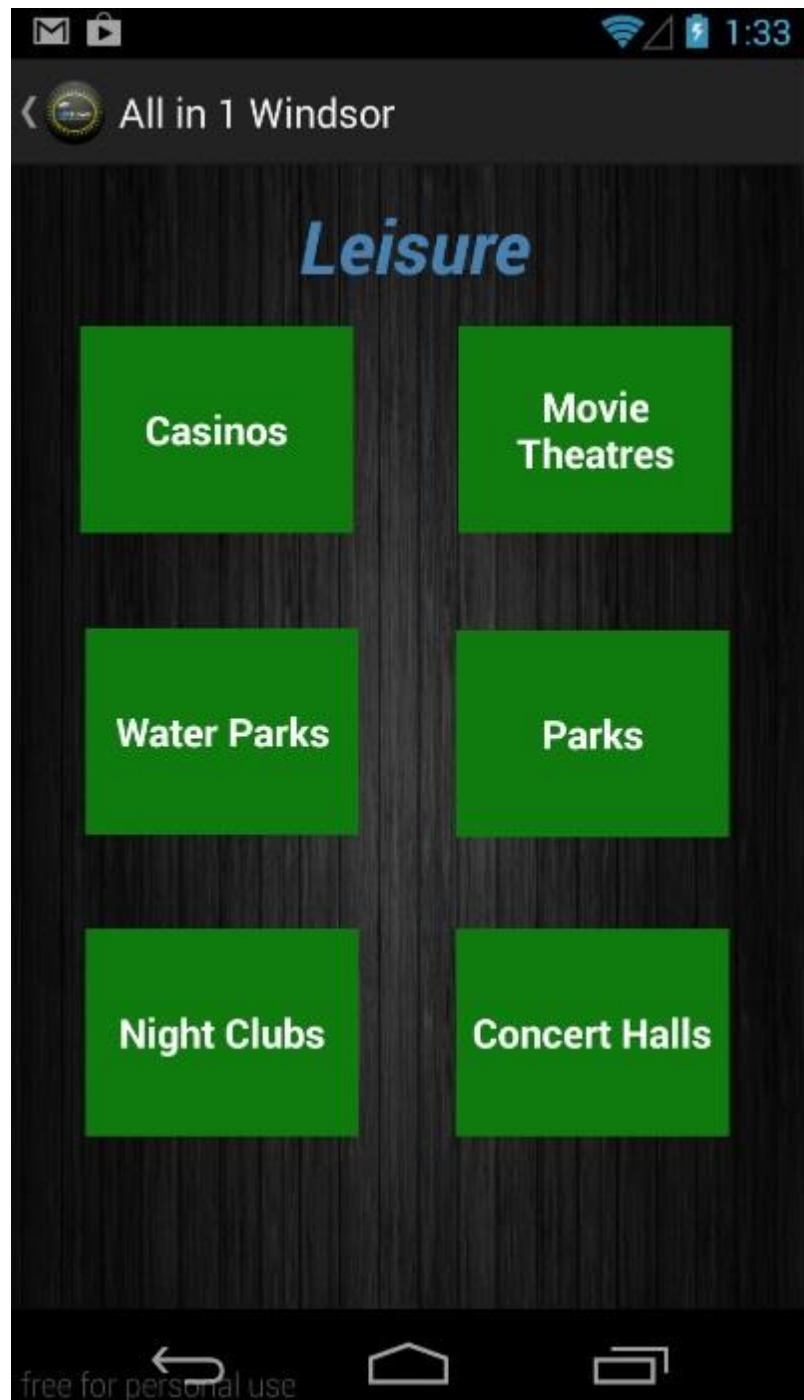
Risks associated with the availability and quality of tools to be used to build the product. But all the quality tools and genuine software applications are provided so this risk is indeed eliminated.

4. Implementation

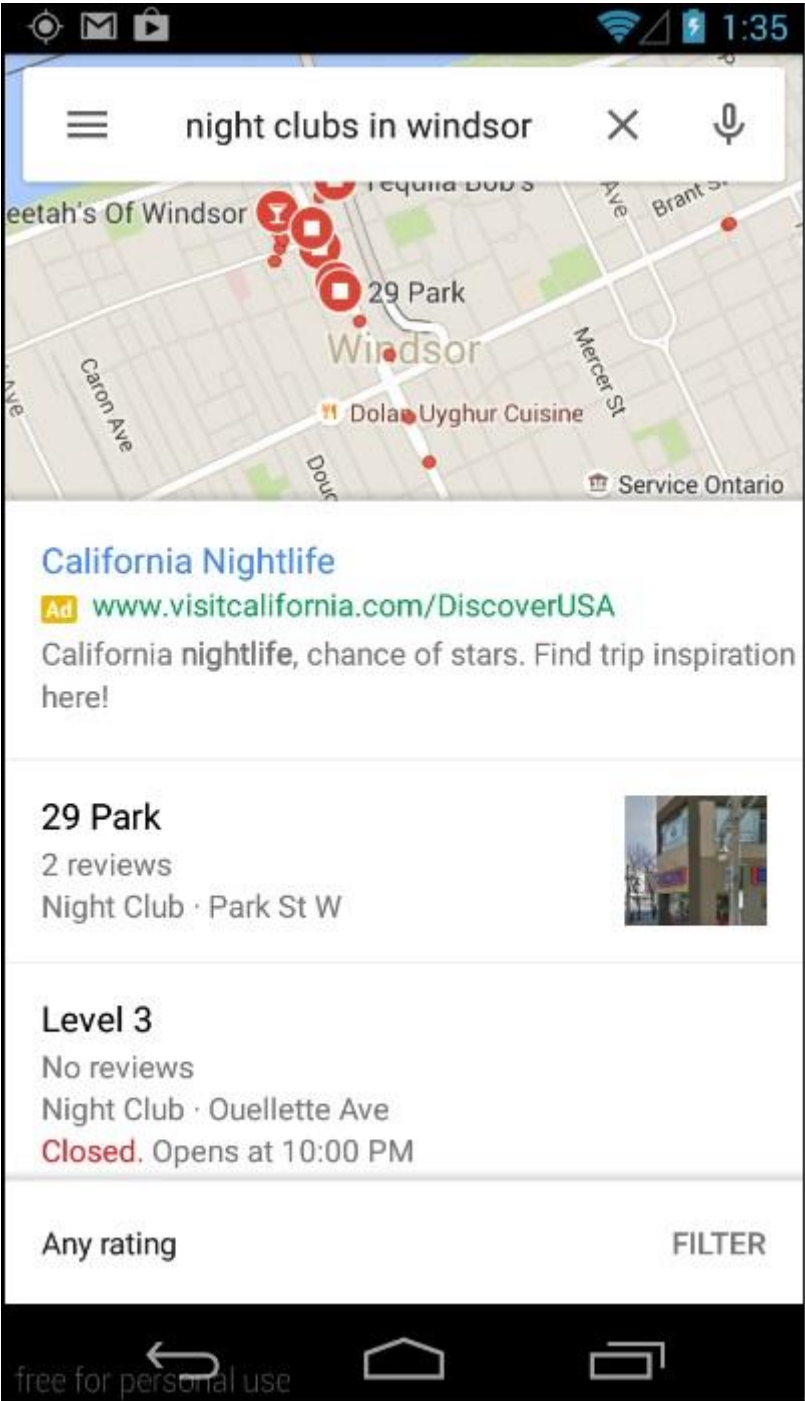
Home Page:



Leisure:



Directed to Google Maps on clicking Night Clubs in Leisure Page:



5. Testing

5.1 Black-box Testing

In Black-Box Testing or Functional Testing, developers are concerned about the output of the module and the application, i.e. whether the application gives proper output as per the requirements or not. In other words, this testing aims to test a program's behavior against its specifications without making any reference to the internal structure of the program or the algorithms used. Therefore, the source code is not needed, and so even purchased modules can be tested. The program just gets a certain input and its functionality is examined by observing the output. This can be done in the following way:

- Input Interface
- Processing
- Output Interface

The tested program gets certain inputs. Then the program does its job and generates a certain output, which is collected by a second interface. This result is then compared to the expected output, which has been determined before the test.

5.2 White-Box Testing

White Box testing is used as an important primary testing approach. Here code is inspected to see what it performs; tests are designed to exercise the code. Code is tested using code scripts, driver, etc. which are employed to directly interface with and drive the code. The tester can analyze the code and use the knowledge about the structure of a component to derive the test data.

We have implemented following two test methods for our application.

```
import android.test.ActivityUnitTestCase;
import android.widget.Button;
public class HomePageTest extends ActivityUnitTestCase<HomePage> {
    public HomePageTest(Class<HomePage> activityClass) {
        super(activityClass);
    }
    public void testNextActivityWasLaunchedWithIntent() {
        Intent mLaunchIntent = new Intent(getInstrumentation().getTargetContext(), HomePage.class);
        startActivity(mLaunchIntent, null, null);
        final Button launchNextButton = (Button) getActivity().findViewById(R.id.worshipPlaces);
        launchNextButton.performClick();
        final Intent launchIntent = getStartedActivityIntent();
        assertNotNull("Intent was null", launchIntent);
        assertTrue(isFinishCalled());
        final String payload = launchIntent.getStringExtra("activity_worship_places");
        assertEquals("Payload is empty", HomePage.class, payload);
    }
}
```

Here in the code “assertTrue ()” will make sure that the intent opened on button click was not null and the other method “assertEquals ()” will make sure that the open intent was the desired intent or not.

6. Future Enhancements

In future iterations we can include features like live news feeds from local news paper agencies so that we can keep our users updated with the local news. As we know that Canada experiences harsh weather, we are also planning to provide weather updates in our application. Moreover, we plan to provide the CIC updates in our application. In short, we want to provide as much information as we can in a single platform so that our user has no need to go on google searching everything.

7. Conclusion

We value Canadian diversity and we have tried our best to meet all the non-functional requirements. The application already meets most of the requirements stated but a software is never complete. There are always chances of new additions to the software. And as we are using agile methodology, changes are always welcomed.