SDV602 – Portfolio Project – I’ve Been There

Mark Christison

Nelson Marlborough Institute of Technology, New Zealand

Contents

[Contents 2](#_Toc49336808)

[Introduction 3](#_Toc49336809)

[Xamarin 3](#_Toc49336810)

[Background 4](#_Toc49336811)

[Problem description 4](#_Toc49336812)

[Storyboards 5](#_Toc49336813)

[Screen 1 – Map 5](#_Toc49336814)

[Screen 2 – Search 5](#_Toc49336815)

[Screen 3 – 5](#_Toc49336816)

[Requirements 5](#_Toc49336817)

[System Requirements 5](#_Toc49336818)

[User Requirements 6](#_Toc49336819)

[*User Stories* 6](#_Toc49336820)

[UML 6](#_Toc49336821)

[Use case diagram 6](#_Toc49336822)

[Use case descriptions 6](#_Toc49336823)

[Initial Functional Test Cases 6](#_Toc49336824)

[CRC Cards 6](#_Toc49336825)

[Domain Class Diagram 6](#_Toc49336826)

[Design Class Diagram 6](#_Toc49336827)

[Screen Designs 7](#_Toc49336828)

[Screen 1 7](#_Toc49336829)

[Screen 2 7](#_Toc49336830)

[Screen 3 7](#_Toc49336831)

[Conclusion 7](#_Toc49336832)

[Bibliography 7](#_Toc49336833)

# Introduction

# Xamarin

Xamarin (pronounced zam-rin from the root tamarin) is a free, cross-platform open source project from Microsoft for developing applications (apps) for iOS, Android, UWP and many other mobile platforms or devices.

Apps are written with .NET and C# in the shared cross platform development environment Visual Studio. With integration with other tools in this development environment such as hot reloading, Xamarin is a powerful tool for making modern apps for mobile devices.

Ref: (What is Xamarin?)[ https://dotnet.microsoft.com/learn/xamarin/what-is-xamarin]

Xamarin extends the base components of C# by introducing several key features including:

* Extensible Markup Language (XAML)
* Editor extensions
  + Visual Studio
  + Visual Studio Code
  + Atom
  + Vim
* API’s to additional libraries such as:
  + Google
  + Apple
  + Facebook

The primary benefit of developing with Xamarin, is that no matter what platform you are developing for, the business logic and platform specific API’s are handled under the same shared C# code.

Using other tools from the Xamarin ecosystem such as Xamarin.Forrms extends the already powerful tool set. Xamarin.Forms is a cross-platform (Android, iOS, Universal Windows Platform UWP) UI framework. Using Xamarin.Forms allows 1 UI to be made for the 3 platforms.

Due to Xamarin being open source, for any project that requires additional functionality out of the already built libraries, project specific libraries can be created.

## Background

The history of the development of Xamarin goes back to the late 90’s. Miguel de lcaza and Federico Mena started working on the GNOME Desktop project. Their idea was to “develop a free and complete set of user-friendly applications and desktop tools, similar to CCDE and KDE but based entirely on free software” (source)[https://mail.gnome.org/archives/gtk-list/1997-August/msg00123.html]. Miguel founded Helix code which later became Ximian, and focused on developing the project with free and open source software, instead of the propriety licence it had been up until that point.

When it was announced that Microsoft was going to release their first .NET framework, Miguel and his company became instantly excited about the project. They set to work in teams to work on the Gnome binding infrastructure. The goals where to:

* Build tools to improve their productivity
* Create applications with the tools that they built
* Reduce development time and cost

The project was launched on 19th July 2001 at the O’Reilly conference under the name “Mono”. Miguel saw the project as “a means to an end: a technology to help Linux succeed on the desktop.” (Internet archive, originally posted 2003 October Miguel de lcaza)[https://web.archive.org/web/20110606210557/http://lists.ximian.com/archives/public/mono-list/2003-October/016345.html]

# Problem description

The software to be designed will be like that of the popular website Geocaching (<https://www.geocaching.com>). A user would take a photo of a location, that location would be geo tagged with its location. Other users would be able to find that location later.

Complexity could be added later by allowing users to make a set of waypoints. For users to follow the path they would need to fine the location and scan it with their camera on their phone. The photo would confirm that users have been at a location.

Users could see “start points” on a map and start a “path” of “waypoints” that other users create. The user would be required to take a photo of each way point. The idea here is that this could be expanded into an app that would allow people to see points of interest along a walkway for example, such as a famous landmark. Some additional information could be provided to the user at this point such as a blub about the landmark/location. Additionally, users could leave comments at each way point or create their own branch of landmarks off of the main “path” that was originally developed.

Another feature could be a “capture the flag” idea, whereby a user would take a picture of a location like a piece of public art. From there, they would take a picture of another location but that would not have a geo location. The idea being that a second user would have to find the second location, take a photo of it to confirm that they have been there, and then move to another location and take a photo of it. The system would log the geo location of the new location and show it to all the other users to attempt to find.

# Storyboards

Map, Search, title screen, login, location

Top level diagram as an Overview

Detail of each screen

## Screen 1 – Map

## Screen 2 – Search

## Screen 3 –

# Requirements

## System Requirements

Functional Requirements

Non-Functional Requirements

## User Requirements

### *User Stories*

# UML

## Use case diagram

## Use case descriptions

|  |  |  |
| --- | --- | --- |
| Use-Case | | Search for Item |
| Version / Date | | Version 2 / 3 May 2020 |
| Author | | Mark Christison |
| Summary | | User searches for item and receives a list of matching items |
| Priority | | High |
| Dependencies | Includes |  |
| Included By | Search for item by search term  Search for item by category |
| Actors | | Customer |
| Trigger | | User makes a search request |
| Pre-Conditions | | Items are known to the system user is on search page |
| Description | | 1. User inputs search term of item they are looking for 2. User submits search 3. System generates results 4. System displays results to user |
| Extensions / Exceptions | | 1a. User limits search to a specific category  5. User requests detailed results of a product  5a. System displays one product details page |
| Post-Conditions | | User is displayed a list of items or one items complete details |

## Initial Functional Test Cases

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Use Case** | **Function Being Tested** | **Initial System State** | **Input** | **Expected Output** |
| Application Start-up | System is started when the switch is turned “on” | System is turned off | Activate the “on” switch | System requests initial cash amount |
| Change Screen |  |  |  |  |
| Get Location Information |  |  |  |  |
| Get Route Information |  |  |  |  |
| Navigate to Location |  |  |  |  |
| Start route |  |  |  |  |
| Show route |  |  |  |  |
| Show location |  |  |  |  |
| Follow route |  |  |  |  |
| Add location |  |  |  |  |
| Add route |  |  |  |  |
| Add photo |  |  |  |  |
| Update route |  |  |  |  |
| Update location |  |  |  |  |
| Delete route |  |  |  |  |
| Delete Location |  |  |  |  |
| Create User Account |  |  |  |  |
| Update Account |  |  |  |  |
| Delete Account |  |  |  |  |

## CRC Cards

## Domain Class Diagram

## Design Class Diagram

# Screen Designs

## Screen 1

## Screen 2

## Screen 3

# Conclusion

# Bibliography