Application Development for Mobile Device and Website Deployment

Joshua Barnett

50:198:493 Senior Design Project

Spring 2020

05/15/2020

**Abstract**

The choice of software language and framework can allow developers to write code that works on multiple platforms and operating systems. C# can be used with CSHTML and XAML to write code that runs in the web browser and as applications in the android and iOS ecosystems. The software development lifecycle consists of multiple stages that together carry a project from an idea to a usable product.

**Introduction**

The general area of this project is website and mobile application development. The specific problem addressed by this project is the software development of reusable code between various deployment spaces. The motivation in choosing this project is to gain a deeper understanding and experience of mobile application and website development. A secondary goal was to gain practice in moving a project through all stages of the software development lifecycle. This topic is important because in practice, code reusability can benefit both software developers and companies. Software developers can potentially write code once to run in Android devices, iPhones, and web browsers, as opposed to writing separate code specifically for each of these platforms. For companies, this approach could save money by allowing them to hire or train employees to code once instead of building separate teams for android applications, iOS applications, and websites. The benefits of a well-executed implementation of the chosen problem should show streamlined planning, development, and testing.

**Background**

Publishing or deployment refers to making an application or website available for the public or target audience to use. Business requirements documents can provide information about the client, summarize the project, and include business requirements, descriptions of different types of users, and technical and integration requirements. Functional requirements document is how a system should behave. Use case explain how an actor or system will use software to accomplish a task. These are related to functional requirements, and include necessary pre-conditions, steps towards success and possible alternative scenarios. Use case diagrams provide a visual representation of an actor’s interaction with a system. Test cases are a form of documentation consisting of actions, inputs, and conditions related to a system that can determine if expected results occur. Test cases can reveal problems in software before the software is released to users. Wireframes are blueprints that show how an application is supposed to look. They should represent the layout of each page, interfaces, and navigation between connected pages.

There are several technologies available to assist in the software development lifecycle. Basecamp is a tool that facilitates project planning amongst software development groups. Axure is a software tool for developing wireframes. Draw.io is a software application for creating diagrams. Xamarin is a tool by Microsoft that allows developers to code in C# and XAML and then run and publish on a variety of mobile devices, including devices that run Android, devices that run iOS like iPhones, and wearable devices. ASP.NET Core is a website framework created by Microsoft that allows developers to code in C# and CSHTML and build web pages and browser-based applications. C# is an object-oriented programming language developed by Microsoft. An Apple device is needed for iOS development. ASP.NET Core can be run on Windows, Mac, or Linux. An Apple Developer Account is necessary for iOS development and testing. SQLite can be used for testing purposes of the website component and mobile app components of the project. Visual Studio or Visual Studio are necessary to run Xamarin.

**Project Description**

**Problem Description**

Can code be reused to develop an application for web and mobile devices? Wireframes, use cases, business requirement documentation, functional requirements, test cases, and a project timeline all contribute to the software development lifecycle of a cross-platform mobile and web application.

**Algorithm Description**

The software development lifecycle consists of planning, analysis, design, implementation, testing and integration and maintenance. The planning stage consists of speaking to interested parties, gathering information about what problem needs to be solved. This information can include a business requirements document. Of note would be to determine what technologies are currently in use by the interested party and what technical knowledge and expertise exists in the interested organization. Analysis includes translating business requirements into functional requirements and then creating use cases based on the functional requirements. Design can include the development of wireframes, use case diagrams, activity diagrams, and misuse case diagrams. Implementation is the process of writing the actual software based on the work done in the previous stages. Testing can include running the code through a variety of valid and invalid input to find bugs that need to be fixed. Integration includes releasing the code for the interested party to use. This can include a product being listed in the app store, a website going live and receiving traffic, and joining larger code bases as an additional service or feature in an existing project. Maintenance includes patching discovered exploits, updating deprecated functions and dependencies, and ensuring users can utilize the service as is. After maintenance, the cycle enters the planning stage again. Throughout this process, as new information is discovered, the developer may have to move back to an earlier cycle. For example, testing may find a bug that requires a redesign of the backend of a system. In the analysis phase, a team may discover that the timeline is too short to fulfill the functional requirements, and the timeline decided upon in the planning stage needs to be modified.

**Implementation Details**

The timeline, business requirements documents, functional requirements, use cases, test cases, and use cases were developed with Emacs and Microsoft Word. The use case diagram was created with Draw.io. The wireframes were created using Axure.

The project used the ASP.NET Core web framework, the Xamarin mobile application framework, and the SQLite database management system. The code for the web application was organized to follow the Model View Controller (MVC) model. The views were written in CSHTML. The models and controllers were written in C#. The mobile apps were written in C# and XAML (Extensible Application Markup Language). All coding was done on a machine running Windows 10. A remote connection was created from the Windows machine to a MacBook Pro running MacOS. This was necessary for emulating an iPhone and building and testing the iOS app. Visual Studio was used for editing, navigating between files, and running the programs. The project was defined as part of a single solution, with four projects. These projects were the web application, the general mobile application, and one project each for the Android and iOS specific code.

A screenshot of a cell phone

Description automatically generatedA screenshot of a cell phone

Description automatically generatedA screenshot of a social media post

Description automatically generated

**Conclusion**

A Basecamp page was created to assist in project organization. The timeline was successfully created in the first week of the semester. The business requirements document, functional requirements document, use case diagrams, and use cases were all completed a few days behind their respective schedules. The wireframes were completed behind schedule. The context diagrams and test cases were not completed. Development did not end on schedule; the functional testing and bug fixing tasks were not begun as a distinct process apart from development. The project presentation did not occur, the project write-up was not completed on schedule.

There are many parts of the project that should be built and expanded upon in the future. The tasks of development, testing, bug fixing, deployment, and maintenance have not been completed and need to be finished. The web application should initially be deployed to Azure. There may be benefits to using Microsoft cloud services to host a web app built using Microsoft’s web framework. The mobile apps should be published to Android’s Google Play store and Apple’s App Store. The list of issues on the mobile apps should be implemented. Drop down menus in all the apps should be implemented where appropriate. The ability to upload files should be made possible for the apps. The user passwords should be hashed, salted, and stored in a separate database. The data for the webapp should be stored in an SQL server database as opposed to SQLite. The mobile app should continue to use SQLite for local storage. A layer should be created between the mobile database and the SQL server to share data between the mobile apps and web app. The applications should also be made more aesthetically pleasing, with less empty spaces.

**Self- Review**

There were problems encountered in working on this project. Tutorials in the Frameworks’ documentation were completed in order to overcome unfamiliarity with ASP.NET Core, Xamarin, and C#. StackOverflow was consulted when official documentation failed to provide solutions to IDE errors. A strategy was not found for completely reusing code between the mobile application and the web-based application. For example, the C# files for issue and account information were very similar between the mobile and web files. However, there were validation functions in packages available for the web application that were not available for the mobile application. Also, the project veered from the planned timeline within a few weeks of the semester beginning.

Positive lessons included learning how to create a business requirement, functional requirements, wireframes, use cases and use case diagrams. Apple technologies learned remote logins to a MacBook Pro, emulation of an iPhone for development, and setting up an Apple Developer account. Microsoft technologies learned included coding in C# and CSHTML and utilizing the ASP.NET Core and Xamarin frameworks.

References

ASP.NET documentation. C2020. Microsoft; [accessed April 3, 2020]. https://docs.microsoft.com/en-us/aspnet/core/?view=aspnetcore-3.1

Hassan, U. September 12, 2018. Create A Simple Sign Up Page in Xamarin.Forms. C# Corner.

[accessed April 15, 2020]. https://www.c-sharpcorner.com/article/create-a-simple-sign-up-page-in-xamarin-forms/

Marisc I. 2012. Software Engineering. New Brunswick, New Jersey: Rutgers University;

[September 12,2012, February 1, 2020]. https://www.ece.rutgers.edu/~marsic/books/SE/book-SE\_marsic.pdf

Mustafa, Safi. February 27, 2018. Validation in Xamarin using DataAnnotation. Stack Overflow.

[Accessed April 30, 2020]. https://stackoverflow.com/questions/49006331/validation-in-xamarin-using-dataannotationxamarin-using-dataannotation

Xamarin.Forms documentation. c2020. Microsoft; [accessed April 10, 2020].

https://docs.microsoft.com/en-us/xamarin/xamarin-forms/

**Appendix**

Various Screenshots of Application pages running in desktop web browser, Android mobile emulator, and iOS mobile emulator.A screenshot of a cell phone

Description automatically generatedA screenshot of a cell phone

Description automatically generatedA screenshot of a cell phone

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a social media post

Description automatically generatedA screenshot of a social media post

Description automatically generatedA screenshot of a cell phone

Description automatically generatedA picture containing monitor, screenshot, computer, video

Description automatically generatedA screenshot of a cell phone

Description automatically generatedA screenshot of a computer

Description automatically generated

Issue class, mobile application, and web application. The web application supports validation rules with the DataAnnotations namespace.

A screenshot of a computer screen

Description automatically generatedA screen shot of a computer

Description automatically generated