**Inventory System for Minor and Major Equipment Related Work**

**August Davis**

Student

Department of Computer science

[*August.Davis@und.edu*](mailto:August.Davis@und.edu)

**David Erickson**

Student

Department of Computer science

[david.p.erickson@ndus.edu](mailto:david.p.erickson@ndus.edu)

**Christian Hansen**

Student

Department of Computer science

[christian.hansen@ndus.edu](mailto:christian.hansen@ndus.edu)

**Introduction:**  This related work paper will cover several sources that showcase why we choose things such as what framework, database protocols, and even how we plan our application to look. To review, this project is to create an inventory management system for the University of North Dakota to keep an accurate record of their major and minor equipment around the campus. This system will be based initially on an android based software with eventual compatibility to a website and finally a IOS system.

*Microsoft Visual Studios* [1]offers a well-rounded platform for us to develop our application. With their use of .NET, we will be able to create an application that can be accessed across Android, iOS, and windows. Visual Studios also offers the use of Azure App Service to connect the Universities own database.

*Android Studio* [2]offers custom-tailored tools for Android development that helps speed up development and offers a fast and feature-rich emulator. It has prebuilt templates that you can follow, or build one from scratch. This is not the best software for us to use since we also want the use of iOS and Windows in future iterations of our system.

*Xamarin* [3]is an extension that can be added to Visual Studios that allows us to build native apps for multiple platforms on a shared C# codebase. It also allows us to test our app on over 2,000 devices with do-performances such as taps, pinches, swipes and more. This will allow us to create our application using one codebase that will be used across different platforms eliminating the need to write a separate codebase for each one.

*University Of North Dakota (UND)* [4]The University of North Dakota is required to follow “certain identity standards” while still being allowed to be flexible with the web content. UND defines a set of style guides for web content that will be brought across into the design of our application such as the color scheme, consistency of the navigation bar, and formatting of lists. These are just a few of the things that they outline.

*Mobile User Experience Guidelines and Recommendations* [5] Digital Gov published this article which outlines some of the guidelines that have been deemed the most relevant from community events from 2013-2015. They have distilled that 1) make sure your content is structured and chunked appropriately for multiple devices. This is useful to make sure that one key aspect of our design will work across multiple devices without building a dependency on it just to find out it won’t work correctly. 2) Follow industry user interface guidelines and government regulations in the development of your mobile product. For our project we will be using UND’s Identity Guidelines. 3) Leverage the device s features for usability and accessibility. We will be making use of the camera to scan barcodes and the wireless connection to be able to connect to the database anywhere with the necessary login information. 4) Test at multiple points in the design and development process. We intend to continuously test and modify our code to ensure that it is stable and brings an easy experience to the user. 5) Collect and use data to determine what content your users want and where. This will be used in the later stages of our development when we present our prototype to test subjects to see what they do and do not like about it and where it could use improvement. 6) Develop security and privacy guidelines regarding what the app does and how it protects user data. Security will one key aspect since there will be remote access to UND’s database.

*Microsoft Azure* [6] provides a set of cloud services for developers to build, deploy, and manage mobile applications through Microsoft's global network of datacenters. It allows us to connect our application to our databases, and runs on all major platforms, easing the transition from Android to iOS, and to desktop systems like Windows, Linux, and Mac OS. This service integrates cloud-based computing with on-site computing to allow the app to run locally, then upload data seamlessly to the database when a connection is established.

*Microsoft SQL Server* [7] will enable us to store our inventory data on-site at UND. It is a cross-platform database technology which stores and retrieves data as needed by the user. Its use is heavily integrated into Microsoft's other technologies, such as Azure and Visual Studio, allowing for easy usage. It can also pass reports to mobile devices, which will give us a way to display data that can easily be interpreted by the user.

*Oracle MySQL* [8] is another option for a database system which will allow us to store and retrieve data. It is integrated into Visual Studio to allow for seamless use in the Server Explorer tool. This is also a cross-platform technology, enabling use with any system.

*LINQ* [9] is also known as .NET Language-Integrated Query, and is a technology which allows for queries (commonly used for retrieving data from a database) to be written in a language which is more comfortable for a developer. It was created to allow traditionally non-object-oriented operations (such as database queries) to be written in object oriented languages (such as Visual C# and Visual Basic). There is also lots of integration available with Visual Studio, making it an excellent choice for Visual Studio developers.

*SQLite* [10] is an embedded database engine designed for use without a database administrator on systems like cellphones, cars, airplanes, robots, and more. Data is stored on-disk without the need for a server, but can also be setup in a client/server format. It is excellent when used on apps and websites with low to medium traffic. This technology could potentially become hindered as more items are added to the database, as it is limited by the space of the disk it is hosted on.

*THD mobile* is an inventory and person tracking software that is provided and maintained throw the NDSU network of applications. This application’s is designed to work as a standalone application and a web site using the same graphical design. Unfortunately, the design of this applications interface holds significant problems including pore menu layout, pore back tracking and loss of relevant information between screens that we can learn from and avoid with in our own design.

*Mobile Vison* is an image reignition software API developed and maintained by google. This software includes facile recognition, movement recognition and barcode recognition sub API’s. For the purposes of this project the barcode reignition portion contains features that allow for local translation from image to text without connecting to a server, the ability to recognize multiple barcode standards on the fly and return the scanned contents in an easy to use format.

*SQl feature Documentation* allows users to see everything that can be done with an SQL server. It covers in detail what each feature does, which features are enabled by default and how to disable unwanted features. Once it is determined what features will be needed for this project this will help facilitate the securing of all unneeded features.

*Database security guide lines* in general provide information on what can be done to keep data base information safe. This project requires the handling of university data that should not be public with failure to secure this data possibly having legal conciseness. Any guide lines that facilitate this goal are useful.

*The Family Educational Rights and Privacy Act (FERPA)* details security regulation and student information handling laws. This project will be handling staff and student account, therefore will need to compile with these standards. More reading required on specific standards

# Works Cited

|  |  |  |
| --- | --- | --- |
| [1] |  | "Microsoft," [Online]. Available: https://www.visualstudio.com/vs/mobile-app-development/. [Accessed 25 10 2017]. |
| [2] |  | "Android Studio Features," [Online]. Available: https://developer.android.com/studio/features.html. [Accessed 25 10 2017]. |
| [3] |  | "Xamarin," [Online]. Available: https://www.xamarin.com/. [Accessed 25 10 2017]. |
| [4] |  | University of North Dakota, "Digital Content Standards," [Online]. Available: https://und.edu/identity/web-content-standards.cfm. [Accessed 25 10 2017]. |
| [5] |  | Digital Gov, "Mobile User Experience Guidelines and Recommendations," [Online]. Available: https://www.digitalgov.gov/resources/mobile-user-experience-guidelines-and-recommendations/. [Accessed 25 10 2017]. |
| [6] |  | "Microsoft Azure," [Online]. Available: https://azure.microsoft.com/en-us/services/app-service/mobile/ . [Accessed 25 10 2017]. |
| [7] |  | "SQL Server 2017," [Online]. Available: https://www.microsoft.com/en-us/sql-server/sql-server-2017. [Accessed 25 10 2017]. |
| [8] |  | "Oracle MySQL," [Online]. Available: https://www.mysql.com/why-mysql/windows/. [Accessed 25 10 2017]. |
| [9] |  | "Linq," [Online]. Available: https://msdn.microsoft.com/en-us/library/bb308959.aspx. [Accessed 25 10 2017]. |
| [10] |  | "SQLite," [Online]. Available: https://www.sqlite.org/. [Accessed 25 10 2017]. |