**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.

*Microsoft Azure* 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 seamlessley to the database when a connection is established.

*Microsoft SQL Server* 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* 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* 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 a great choice for Visual Studio developers.

*SQLite* 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.

# Works Cited

|  |  |
| --- | --- |
| [1] | "Microsoft," [Online]. Available: https://www.visualstudio.com/vs/mobile-app-development/. [Accessed 25 10 2017]. |

Azure - https://azure.microsoft.com/en-us/services/app-service/mobile/

Microsoft SQL Server - https://www.microsoft.com/en-us/sql-server/sql-server-2017

Oracle MySQL - https://www.mysql.com/why-mysql/windows/

LINQ - https://msdn.microsoft.com/en-us/library/bb308959.aspx

SQLite - https://www.sqlite.org/