BSc in Software Development

Year 3

COMP07030 Software Design Project

*SpellSpell*

*G00313312*

*Andy Blankley*

Contents

[Introduction 3](#_Toc449364243)

[Architecture of the solution 4](#_Toc449364244)

[Class diagram and Data Model 5](#_Toc449364245)

[Technologies used 7](#_Toc449364246)

[Problems Encountered/Solved 8](#_Toc449364247)

[Conclusions / Recommendations 9](#_Toc449364248)

[Installation instructions 10](#_Toc449364249)

Student Number: G00313312

Student Name: Andy Blankley

Supervisor: Kevin O’Brien

GitHub Link: https://github.com/Blankz1035/SpellSpell-FinalYearProject

# Introduction

*Aims/Objectives/Scope*

*I embarked on the task of creating a spelling app/game that I believed would be a great aid for people who occasionally spell words incorrectly. My original plan was to develop this app as a mobile game, but I decided, for purposes of this project, to host it as a windows 10 desktop application.*

*I first thought of this idea when I was in college with a group of friends. We were sitting down chatting and one of the people within the group asked how to spell a fairly simple word, receive. He forgot if it was “ie” or “ei”.*

*Now of course, we did ask why he could not spell such a simple word, but even looking at other people’s work in college, spelling is actually a big issue.*

*Simple words that follow rules like the “I before E except after C” and many others, people were incorrectly spelling, so I decided to undertake a task that could one day improve spelling for many people out there. My vision of this would be later on, with further development and polishing of the app and how it runs, would be introducing into schools, where computers or tablets are used regularly so that people could use the app from a young age.*

*This use of the app from a young age, could potential eliminate the simple spelling mistakes, while educating children with larger, more complex words, giving them a better vocabulary base for when they grow up and progress through life.*

*The scope of this project ranged from coming up with a “User Friendly” design, to getting opinions from people about the app, and then the beginning and ending of the creation process.*

*The design part was easy enough. I tried to target bright colours and a simple layout so that for instance, a 10 year old could easily work the app as well as a 20 year old.*

# Architecture of the solution

*Structure/Storage/Issues - Reasons*

*Give reasons for choosing particular technologies over others.*

The structure of my application is quite simple. It takes advantage of visual studios’ many great features of making different applications and also showing you a design view of what you’re making.

I chose this because I like being able to see what I am making come alive with every change I make. Also it makes it easier to see where you may make mistakes so you can quickly rectify. One other big reason is that visual studio in itself is a very powerful studio for creating GUI applications while implementing services such as SQL and other database API’s.

For my database I chose Azure as a good platform to host from. Doing this meant that I have a secure base from which to send data to and from and a simple GUI to make changes to my tables and data. I was able to setup that when the game finishes or gets reset, it sends the data including username, password and high score to the service which then fills the table accordingly.

The language I used was C# and Xaml as I like these and because hand in hand, they both work very well together. C# is simple to navigate around also, making the creation process that small bit easier. I also wanted to further my knowledge with both languages.

I had originally planned to use MySQL and connect to a remote server, hosting the database. I had multiple issues with this though, as connecting to the database failed on more than one occasion. I trouble-shooted many things, but fixing this issue did not occur.

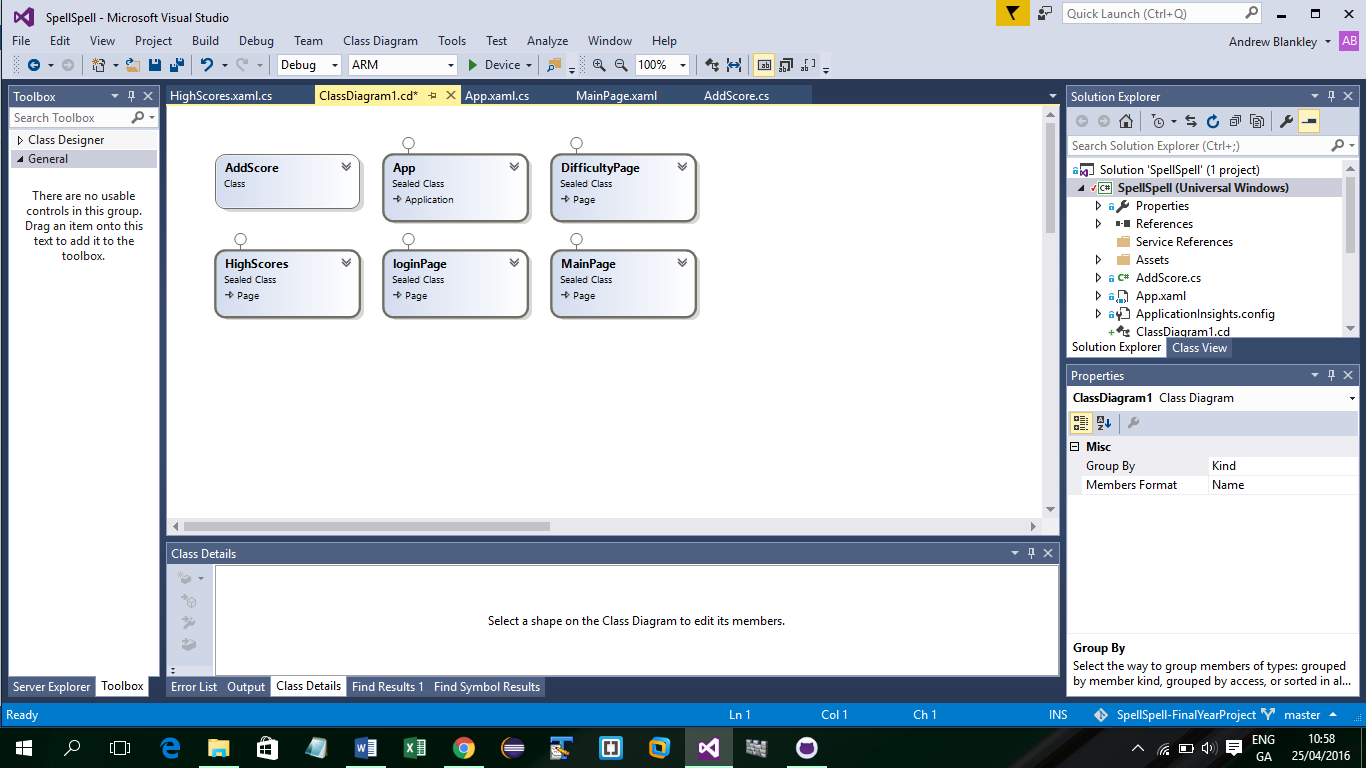
# Class diagram and Data Model

*This section should include a class diagram and data model as well as the reasons why they were designed as they were.*

The classes I used are defined in the screenshot below. Each class consisted of its on methods that all in turn make the functionality of the application run together.

The addScore() class is a self-made class that had to be created for the connection to the database to work properly. The class is the same name as the database as in the code which sends to the database, the name of the class has to be the name of the table. This all worked perfectly and was a great experience trying new endeavours.

I also set the loginPage as the first loaded page in the App class, as I wanted this to be the first page the user sees.



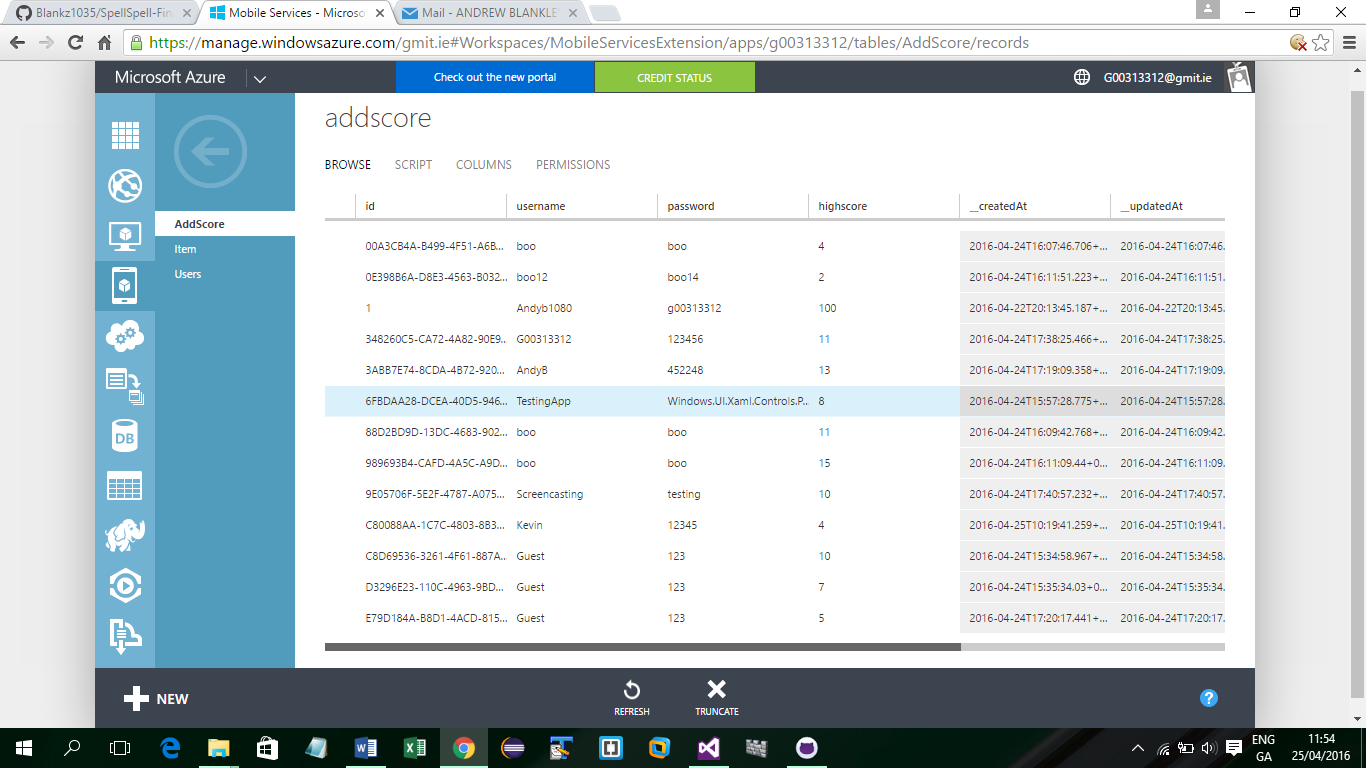
The data model I used was Azure. I created a SQL database on the site, which also had its own connection string and host address. I in turn used these in my connectivity to the database to post data to the table.

The table made was called AddScore. I added 4 fields consisting of Id, Username, Password, high score. It also came with some predefined columns which basically shows a timestamp of when the data was uploaded/updated.

The username holds a string, which will either be a predefined “Guest” name or a user’s Name.

The password holds a string which will either be a predefined “123” or a user’s password.

The high score holds a INT which will be the score in which the user achieved whilst playing the game!



# Technologies used

*This section should describe technologies you used, for example the programming language(s), database environment, development environment, and other technologies.*

*It should also state why these technologies were used – this could be because you wanted to gain experience using a particular technology, or because it was a client requirement.*

During the making of this project, I chose many technologies which I knew would aid the development of my application.

Giving that I decided to develop a windows 10 desktop application, I thought that using visual studio was the right tool for the job.

The coding language that is used for this is C# and xaml. Both of these languages are powerful and great for making a dynamic GUI and backend for the app. I chose this as I am proficient in both and enjoy the coding process and seeing my creation come to life.

Visual studio also shows a design view which in turn with the changes you make in the xaml code, show on screen so you can make adjustments accordingly.

For the database, I tried many options, including using a remote server to host the database, using Azure. Unfortunately, issues arose with connectivity, so I decided to go with a service azure offers, being Azure mobile services.

I created a database on the site and using the mobile services API added to visual studio, I was able to post data to the cloud across the internet and update the data. This was great as a part of the app, and what I wanted to achieve was online leader board type application. I wanted this as in my experience and research about apps that have gained a lot of attention in the technology world, you need to create something that generates some sort of interest and appeals to a person’s competitive side. This does this as users can play to be the top. By doing this, the users are gaining knowledge about spelling, whilst also competing to be the top scorer!

I chose all of these technologies as I want to improve my knowledge in all areas of windows development. This was a task that I feel I achieved, gaining great intel on azure and the workings of their mobile services, learning how to connect each page with complicated code and designing an app, that I feel would appear well to all age groups who could potential gain use.

# Problems Encountered/Solved

*This section should document any problems you encountered during the development of your project and how you overcame them.*

*Perhaps you spent some time looking at a particular technology/algorithm, but decided it was not appropriate for your project.*

*I encountered multiple problems in the making of my application, as would be expected with a project like this.*

*The first problem I encountered was connection to the virtual server where I planned to be hosting my app’s database from. I was planning to be using PHP to connect to the service to connect and also send/receive data. In theory, this was perfect and the method and code I was using should have worked perfectly. Unfortunately, this wasn’t the case.*

*I could not make a connection to the database at all. I was able on the virtual machine to query the database once the host name was set to “localhost”, but when I set as the address, it wouldn’t connect. I tried opening ports on both azure and the server, and adding the ports to the firewall, but this was without success.*

*My solution to this was to use azure mobile services. Using this, I was able to connect with ease to the database service and also send data to the table specified. This solved the problem of having no connectivity and gave me a base to be able to carry on with development.*

# Conclusions / Recommendations

*This section should discuss what you learned from the process and from the work you did.*

*This is a key reflective portion of the write up.*

*To conclude on my project.*

*I have learned a great deal in the making of this windows 10 desktop app.*

*From the research of the ideas for the project structure and to using services which I have never used before like azure mobile service and the process of online databases, I feel I have grown my developer skills that little bit more.*

*My aim for this project was to construct an application that I feel could benefit not only myself but that could develop over time to something quite spectacular. Given further development and a few tweaks, I believe this could be something that could not only put on the app store, but also put into schools to hopefully become part of a pupils study. This would benefit them a great deal, improving vocabulary whilst being fun to play with friends.*

*I hope to also convert the application so it works on all platforms, being android, IOS and windows phone. For this I will be doing more research and further development, to code in java for android and also Swift for IOS.*

*I hope to also add another database connection, changing the way the words are displayed. As of now for demo purposes, I have arrays set up for each category with a small amount of words. I would change this, making each one of these methods, call to a database with many more words to display. This would finish the application, with the ability to add more words with each update to the program.*

*I would also like to add in a proper login and registration system where by validation and authentication would be involved also.*

*Finally to full conclude, I have learned a great deal from this project, and I believe this is only the start of my development career.*

# Installation instructions

*This section describes how a user could obtain this application on their own machine.*

*Visual studio comes built in with the option to package your application once it is fully complete.*

*I did some research on submitting to the app store also. One’s app needs to conform to specifications from Microsoft like auto snapping, copyright and basic functionality. You also need to have a developer account which, at present, I don’t hold. This is also something I will be looking into.*

*For now, on GITHUB, I have packages ready for installation. There are many ways this can be done.*

*The user, if they have visual studio 2015 installed, can run the project and once the project runs, it installs on the machine. This is very simple and would only take 10 minutes.*

*The other way if the user didn’t have visual studio, would be to download the App package folder from GitHub, and then in the folder there is a add app packages. This is a PowerShell file. The user should right click on the file, and then run with PowerShell. This then installs onto the machine as an app.*

*There is no need to install anything else. The database is predefined in the app, so anyone who uses the app actually connects to the same database.*

*Once a user installs, the login page is shown. The user can just play as guest (shown as a button), or enter a username and password. This will then be your game name and password, to be sent with high score to the cloud. You can change user by returning to the login page and entering different credentials.*