**Xamarin Till Application**

**A Xamarin Till Application for Professional IT module**

**Matthew Bartley - G00342591**

**Neil Byrne - G00343624**

**Sean O'Gorman - G00314259**

Contents

[Introduction 3](#_Toc7100473)

[System Requirements 4](#_Toc7100474)

[Technology Used and Why 5](#_Toc7100475)

[Architecture of the Solution 6](#_Toc7100476)

[Design Methodology 7](#_Toc7100477)

[Features of the Implementation 8](#_Toc7100478)

[Limitations and Known Bugs 9](#_Toc7100479)

[Testing Plans 10](#_Toc7100480)

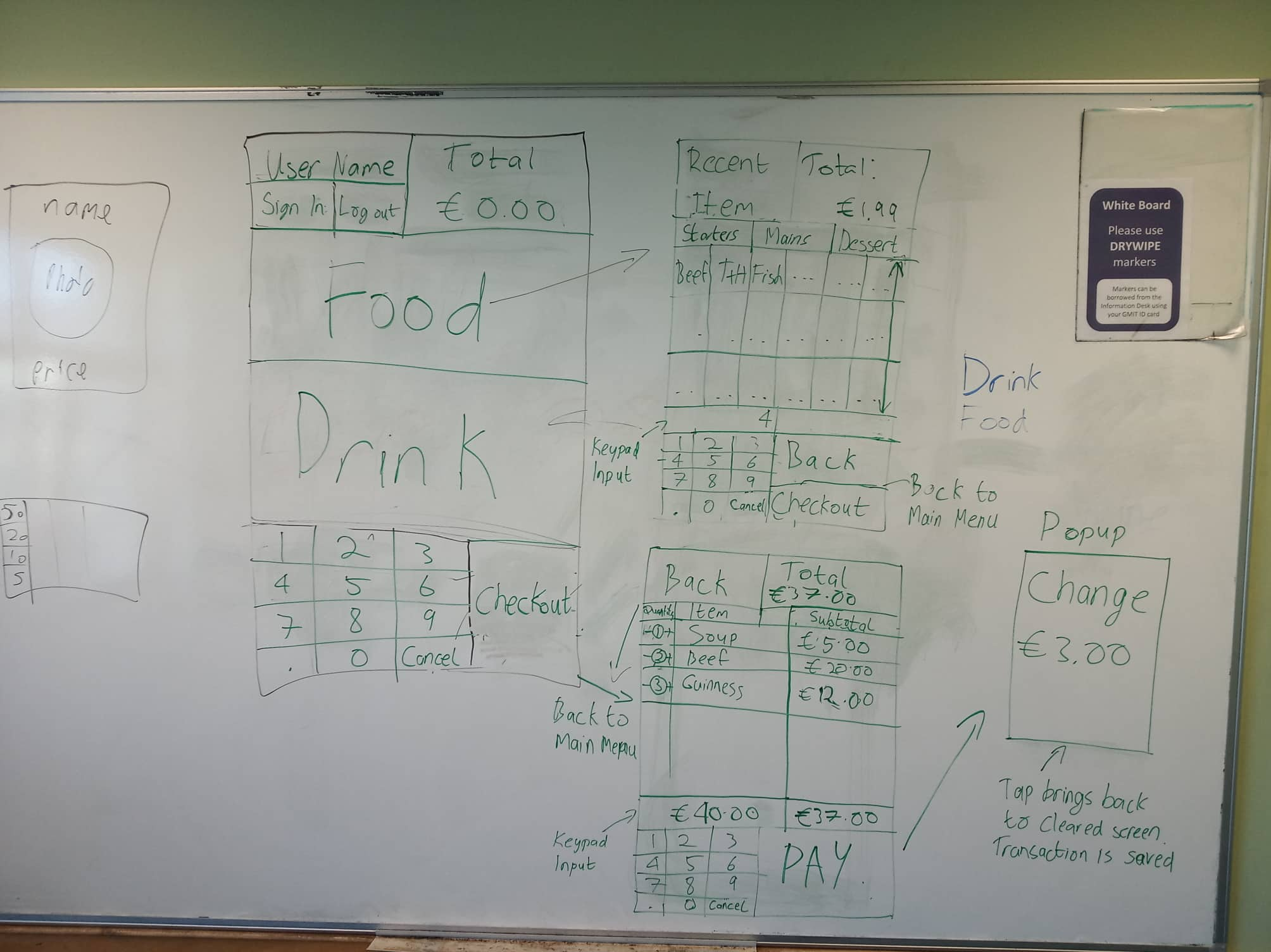
[Recommendations for Future Development 11](#_Toc7100481)

[Conclusions 12](#_Toc7100482)

Introduction

The purpose of the Professional IT module is to bring what we have learned from all the modules in the previous semesters and put them to best practice. There were no requirements, it is up to the students to design, develop and deploy a project.

We discussed in depth of ideas we had for a project and what we would like to do, how we would go about it and what learning difficulties might be on the road ahead. However, with our Supervisor, Kevin O’Brien we decided to design and develop our own Till App. We wanted it to be accessible and intuitive for any user to operate and be available to anyone. For this reason, we decided to develop in Xamarin using Visual Studio 2017. Xamarin provides a way to develop quick and easy on Windows, Android and iOS simultaneously. This means that anyone familiar with a smartphone will be able to use the app, and virtually everyone can access it.

With our architecture decided we had our first meeting in the library to flesh out the details. We drew our first sketch of the app and discussed how we would accomplish everything. Assigning ourselves and each other work to research and develop on.

# System Requirements

* Visual Studio 2017
* Xamarin plugins

# Technology Used and Why

**Xamarin** - “*Xamarin is a Microsoft-owned cross-platform ecosystem offering a single language – C#, class library, and runtime that works across all three mobile platforms of iOS, Android and Windows Phone, will still compiling native applications that are performant enough even for demanding games*”

There are a few reasons why we picked Xamarin as our programming language, primarily because of its cross-platform capabilities and accessibility to all users. We had some knowledge of the language, having completed one module and currently learning more from the current module “Mobile Applications Development” (MAD) and “Advanced Mobile Applications Development” (AMAD) respectively. Giving the steep learning curve of Xamarin, our previous knowledge and help from our lecturer Damien Costello was a great resource. The development of the app would also be extra learning for our AMAD module and assist us in our project for it.

**Visual Studio 2017 -** *“Microsoft Visual Studio is an integrated development environment from Microsoft. Supporting 36 different programming languages, and with full-featured integrated development for Android, iOS, Windows, web and cloud.”*

Having developed Xamarin in Visual Studio 2017 and currently doing so in some modules it made the most sense to use this IDE.

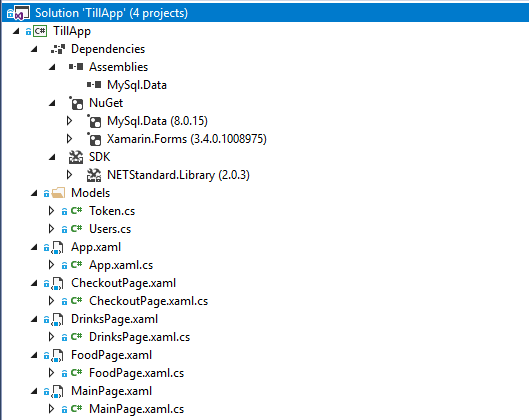
**Amazon Web Services (AWS) - “***Amazon Web Services (AWS) is a subsidiary of Amazon that provides on-demand cloud computing platforms to individuals, companies and governments, on a metered pay-as-you-go basis”*

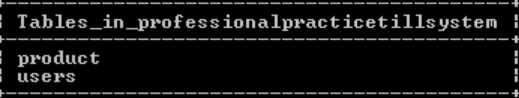
Early in the design process we knew we would want our data to be stored on a database. Having the food items and user information stored elsewhere on a database will reduce the amount of time spent managing the data ourselves. AWS allows the data to be analyzed in a variety of ways and promotes a disciplined approach to data management which we will have to face in a working environment. As with Visual Studio 2017, we had good experience using AWS and felt that we could use this understanding and improve on it. AWS is especially good for providing free credit to students, so we were never charged for a time on the server. Like a reason why we picked Xamarin, the use of AWS for this project would further improve our skill, benefiting us in other modules.

**Structured Query Language (SQL) -** *"SQL is a domain-specific language used in programming and designed for managing data held in a relational database management system, or for stream processing in a relational data stream management system”*

SQL is a light but powerful relational database, being the standard language for relational database systems (according to ANSI), it was the obvious choice for the project. SQL is the first database language we learned and is the strongest. For these reasons and for our wide use of it in current modules it was the obvious choice.

# Architecture of the Solution

Below is the architecture design of the project.



# Design Methodology

We held weekly meetings in order to go over any issues we had come across and divide up the work necessary for the next phase of development. At each meeting, we would talk about what we had developed over the last week, any problems we had encountered, and what we planned to accomplish next. These meetings were extremely helpful in keeping us on track with the development of the app.

By our second meeting we realised just how important it is to fully talk through how we would face a problem, and to write everything down. Sketching out the UI was a great first step and we encouraged each other to write any thought or idea on the board. This helped us to think about the project in many different scopes. On the micro scale; what variable types to use, how to parse information around the project etc. On the macro scale; what will the UI look like, how will the user navigate through the app.

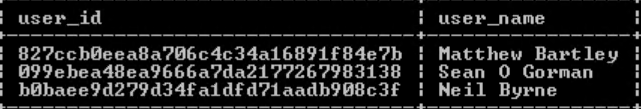
We also wrote out the main notes from our meetings on the GitHub Wiki and created tasks for us to keep track of in the GitHub issues page. This was very useful for maintaining a record of both what had been completed, and what was yet to be done.

As we had decided to use technology which we had only begun to learn in this semester, it took quite a bit of time before we were comfortable enough with Xamarin to create a working application at the level necessary for our intended project. Some other drawbacks included some technical difficulties with a laptop's configuration. Which was rectified with the help of lecturers. And other project deadlines resulted in the bulk of our work on the project being completed in the later stages of the semester. This created some difficulties as it meant that our time was spread thinly among several other projects, as well as studying for frequent assessments.

# Features of the Implementation

A major feature of the project is a database we use to store information. Currently the database stores the Food and Drink information (Name, Price) and User information (Name, Passcode). Storing this data in a relational database makes it very easy to manipulate. In future this information can be used to graph the sales of certain products, buying trends etc. and use this information to make educated stock management.

The User Passcode is hashed given a strong level of encryption, securing access to the application to only Users programmed into the system. Since the database is stored on a server via AWS multiple users can be on the system simultaneously. This feature has not been extensively tested to find any threshold although. Some potential issues would be the database not updating across all users.

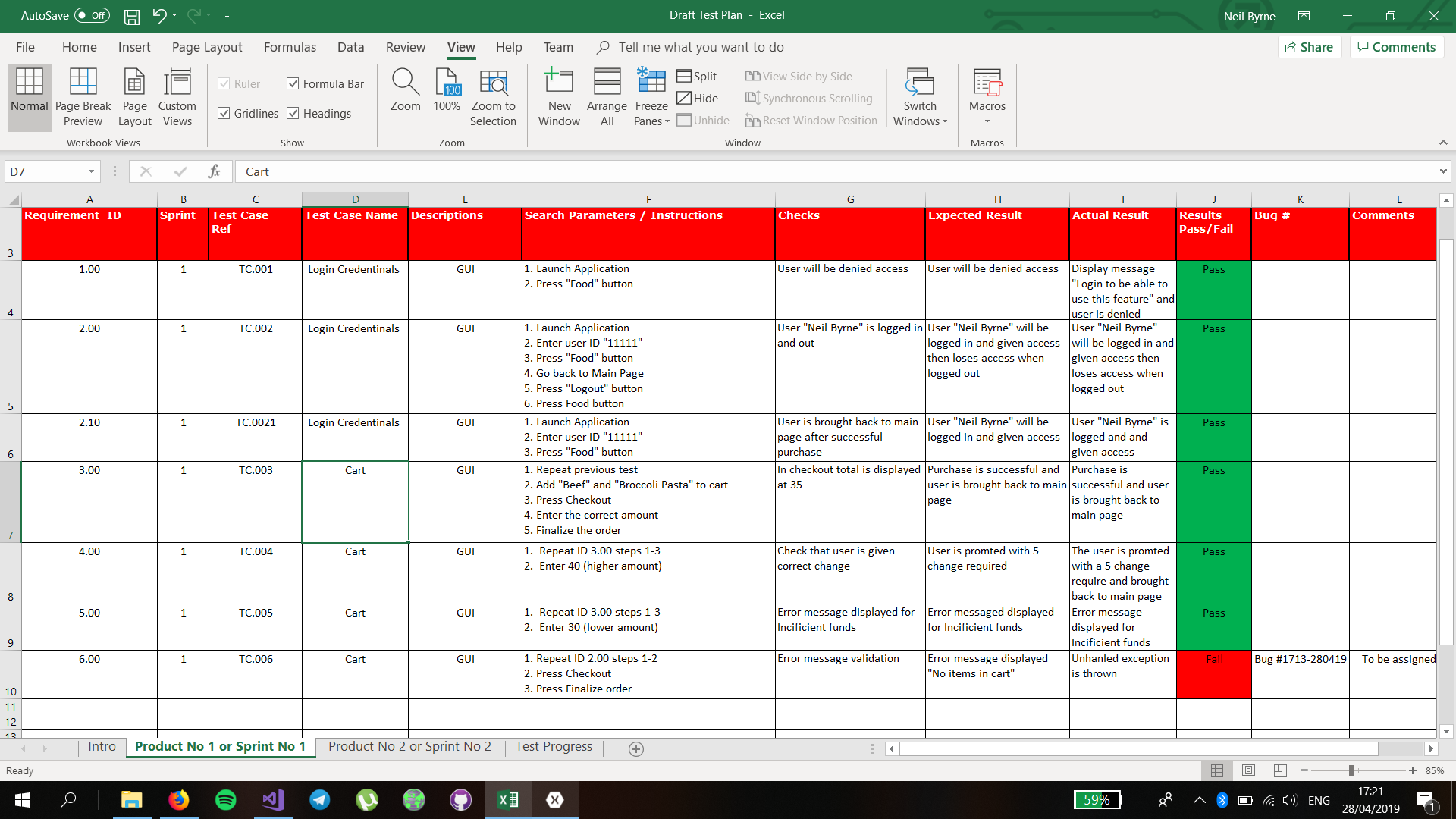


# Limitations and Known Bugs

* When a user finalizes a sale without anything in the cart an unhandled exception is thrown and the application terminates.
* There is no user-friendly way to remove an item from the cart at checkout.
* Extra users cannot be added on the application, they must be hardcoded.

# Testing Plans

A testing plan was conducted in an excel spreadsheet.



# Recommendations for Future Development

For future development we discussed and planned various things from the start of the project. Nothing is too ambitious but some we knew we would not accomplish given the deadline. Some possible developments include the following.

* Store every purchase locally and backed up to the database. With this feature, receipts could be included and allow tracking of past sales.
* Include “Admin” privileges to perform tasks that would be inaccessible for normal users.
* Be able to add new users in the application rather than hardcoding.
* Allow products to be added in the application and edit prices.

Conclusions

**Conclusions:** Throughout the course of this development we understood this would be our most ambitious project to date. Having to tie in all the things we learned in other modules proved to be a tough but rewarding task. We found that the most important thing to do for a project of this scale is to plan it out thoroughly. Our weekly meetings were the most important aspect of the development, allowing us to discuss our ideas and create a comprehensive strategy to execute. Some minor hardware problems slowed development slightly but allowed more refection on how we were going to program the application.

**Recommendations:** Our recommendations are that during the research and design process to have a free thinking environment. Record any ideas for the development and plan out the design so that everyone understands what must be done.

**Reflective:** On reflection we have learned quite a bit about Xamarin and SQL. Going into this project we had a good grasp on these languages but learned even more in our own learning. Researching more into these languages helped us with other projects we were assigned and certainly aided to our knowledge.