UNIVERSITY OF THE WEST *of* SCOTLAND

UWS

BSc (Hons) Computer Science

OrderOut mobile ordering system with cloud backend

Computing Honours Project (COMP10034) Interim Report

Andrew McLaughlin

[B00289311]

[**06/01/2017**]

Supervisor: [**Glenn Affleck**]

Declaration

This dissertation is submitted in partial fulfillment of the requirements for the degree of Web and Mobile Development (Honours) in the University of the West of Scotland.

I declare that this dissertation embodies the results of my own work and that it has been composed by myself. Following normal academic conventions, I have made due acknowledgement to the work of others.

Dissertation Form

First Name: Andrew	**Surname:** Mclaughlin
Banner: B00289311	
Course: BSc (Hons) Computing Science	**Course Code:** COMP10034
Project Supervisor: Glenn Affleck	
Dissertation Title: OrderOut mobile drinks order application with cloud backend	
Session: 2016/17	

Table of Contents

ACKNOWLEDGEMENTS

I would like to take this opportunity in which to acknowledge the following people for their help and guidance throughout this project:

My supervisor Glenn Affleck for advice and guidance throughout the development of this project.

My Parents for their unyielding support, patience and help over the past years at University.

My friends and family who helped in the testing of the application on multiple devices for IOS and Android.

ABSTRACT

In recent years, mobile technology has been on the increase with the proliferation of smartphones and the ever-increasing demand of consumers to use such devices as the means of interacting with businesses or for performing general day to day tasks such as; online banking, online shopping, ordering takeaway food, ordering taxis, to name but a few.

These applications are typically targeted at consumers to provide convenience benefits. However, the benefits of smart technology not only provide consumers with convenience but can also allow businesses to drive sales and build relationships with customers as never before.

Many aspects of the hospitality sector are still set in traditional working practices where it is believed that staff taking orders and payment from customers at the bar is the most effective and established means of serving customers. The result of this is queues building at the bar and customers spending what can be hours in an evening in such queues waiting to purchase a round of drinks. This can result in frustration and even loss of business for the bar as customers decide to move on rather than queue.

The aim of Order-Out is to provide a mobile platform that facilitates the ordering of rounds of drinks in a bar environment for establishments that have subscribed to the service. It uses cloud technologies to provide services that can be consumed by associated Android and IOS applications for both establishment staff and customers.

Admin Bar staff can manage the bars inventory (drinks available on the system), Bar tenders process the orders and notify customers of order processing status via App Notifications, and customers who can build orders via an intuitive Active basket (with repeat order capability) and pay directly from the App.

1 INTRODUCTION

Order-Out is a mobile application with a cloud back end that aims to provide a seamless experience when ordering drinks in a bar environment. A consumer can browse, select and order drinks via a mobile device when sitting at their table, or standing within the premises, or even while travelling to the bar so that the order is ready for collection on arrival.

The decision to select this type of project was due to some important factors;

- The idea of this project has been with me for over a year. While in a bar watching football, I thought it would be much better if I could order drinks on a phone instead of excessive time waiting in a queue at the bar. If I can order a taxi or a takeaway via my mobile device; then why not drinks from a bar?
- In discussions with friends and family about the viability of such an app the feedback was incredibly positive with all expressing great enthusiasm.
- I am keen to learn new technologies and methodologies currently trending within the IT community. The technologies I intend to learn and implement will be cloud based via Microservices built using Microsoft Service Fabric and deployed on Microsoft Azure with a cross-platform Mobile App developed for users to consume the Microservices on Android and IOS devices.

Advances in cloud technology which allows for scaling of systems from not only an application perspective but also from an infrastructure perspective allows applications to evolve rapidly without incurring huge financial costs as many such services have a subscription model of pay-as-use. An example of these are Amazon Web Services and the Microsoft Azure platform. Both provide the mechanisms required to define and grow the infrastructure required of a solution in a programmable manner, with the ability to define rules for seamless automatic scaling of infrastructure components to ensure that the service is reliable and consistently performant under mixed load scenarios.

For Mobile Apps and related backend services whose user base could grow rapidly having an infrastructure platform that is scalable is an essential requirement as degraded performance and reliability would most likely have a major impact on bar and consumer confidence in the service.

This project outlines the prototype for a cloud based application that facilitates the ordering of rounds of drinks in a bar environment. It uses cloud technologies to provide services that can be consumed by a cross-platform mobile web application for both bar staff and consumers.

Initially being developed to cater for the sale of drinks at a bar (this project prototype deliverable) the idea would be to make this extensible so that ordering could include anything a bar sells; food, drinks, memorabilia such as T-Shirts or branded glasses.

The Honours Specification form is in Appendix B.

1.1 Project Scope

The functionality for "Order Out" for this project dissertation will implement all functionality to meet the requirements of a Minimum Viable Product (MVP) with some additional features added to supplement the core MVP features to help streamline the user experience. The MVP will contain core functionality/features and will layout the framework on both Microservice backend and client Apps for additional future feature additions and enhancements.

MVP functionality is classed with a prioritization of "1 - Critical" in the requirements specification.

1.1.1 In Scope Functionality

- Ability for a bar to register and subscribe to the system
- Ability for a consumer to register for the service via a mobile device
- Account/role based secure login to the system
- Mobile client Application being cross platform (including IOS, Android)
- Inventory administration features allowing bar admin users to define the drinks available via the system including pricing and measurements
- Functionality for bar staff to process queued orders by consumers and provide app notification feedback on order processing and completion to consumers.
- Functionality for consumers to build and submit orders via the App.

1.1.2 Out of Scope Functionality

- Provide reporting function that allows pub admin users to see statistics on sales.
- Log user trends such as drinks they like to buy for advertisement
  Provide a Digital Wallet capability with Kitty features
- Any social media integration
- Integration to a payments system (this will be simulated only)

2 BACKGROUND RESEARCH

2.1 MARKET RESEARCH

Market research for the Order-Out application consisted of the following;

- Research into existing drinks and/or food ordering apps that are cloud based
- Technology platforms for deploying the Order-Out application in the cloud

2.1.1 Existing applications

There are some applications that are already available and have similar structure and features that the proposed Order-Out application has in scope:

- Orderella (Orderella, 2016): This is a bar ordering mobile app and service that has many of the features that Order-Out will have. A bar can register to Orderella and a customer can download the application from the app store or google play store. The application requires a customer to register giving their details before entering the application. After registration and login, a customer can use a map interface to find a bar offering the service using the location. Customers can search and navigate a bars menu through the application then a customer can then select drinks and pay for them using online payment methods. According to the guardian Orderella are the market leaders in the UK on this however currently they only have around 150 establishments signed up to use the service. This clearly indicates that this is an up and coming market with huge potential where the technology is not the issue but more the marketing and changing the perception of how this type of service could be of benefit to an establishment.
  (The Guardian, 2016)

- Flowtab (Flowtab, 2013): This is an early manifestation of a mobile drinks ordering from an American based company that allows drinks to be ordered from mobile devices at various American nightclubs. The application features were very much like that of Orderella. Flowtab's model included full integration via software and hardware into a bars Point Of Sale systems and the supply of specialised However, Flowtab is no longer an available application as it was taken off the market in 2013 due to lack of up-take.

- Just eat (Just eat, 2016): An application that allows customers to order food from a registered takeaway restaurant. The application connects customers and businesses through technology that makes ordering and paying for takeaway food easier. Just eat has substantial benefits for both businesses and customers. The technology improves the ordering process and enhances customer experience by offering an online ordering platform that provides convince through smartphone interaction. Just Eat has two roles - 'Restaurants' who offer the food service and 'Consumers' who place orders. (Quora, 2016)

2.1.2 Technologies/ platforms

Mobile cloud applications can be developed in a multitude of platforms and technologies. To create Order-Out research was carried out to determine what was the most viable and appropriate technologies to use for developing. The research considered aspects including: experience, integration between development and deployment tooling, ease of use, familiarity and scalability.

2.1.2.1 Microsoft Visio 2010

Microsoft Visio is a diagraming application. It allows for create of mock up user interfaces and UML diagrams that are good for illustrating the design of both the interfaces and the working of the application through use case diagrams. Microsoft Visio was chosen because of its ease of use and availability.

2.1.2.2 Client App Development

The development platform of choice for the mobile client side application chosen was Visual Studio 2015 (VS).

VS provides a single integrated environment for cross-platform app development using Cordova (Apache Cordova, 2016) or Xamarin (Xamarin, 2016). Cordova apps can be built and tested directly within VS either via Emulators or via plugged in devices with changes in code during debug updated on the app immediately. It also provides the capability to use XAML (Extensible Application Markup Language) (Microsoft, 2016) and C# which allows code to be developed once and native applications for IOS, Android (Studio, 2016) and Windows generated directly from the IDE.

Eclipse (Eclipse, 2016) on the other hand does support Cordova development but not Xamarin. Xamarin is a very powerful cross-platform language that allows all aspects of native app development to be done on a platform by platform basis with the majority of the code done once and shared. Xamarin also offer a testing environment where multiple devices can be tested asynchronously, ensuring a host of different devices on each platform are tested.

Other technologies considered were Phonegap with Cordova however after comparison, Xamarin had more features and seems to require less of a learning curve when it comes to implementing and testing the Apps with the server side of the application.

2.1.2.3 Server Side Development

For the server side development and deployment targets AWS and Azure/Service Fabric where assessed along with an additional option of a Hosted environment with rented virtual servers.

Microsoft Azure and Service Fabric is a Platform as a Service (PaaS)

This type of cloud computing provides virtualized servers where users can run existing applications or develop new ones without worrying about the maintenance of the operating systems, server hardware, computing capacity or load balancing. Azure and Service Fabric are seamlessly integrated allowing for rapid development and deployment of application services via a single integrated environment; Visual Studio 2015+.

AWS is Infrastructure as a Service (IaaS)

This form of cloud computing provides virtualized servers, networks, storage and systems software designed to add to or replace the functions of an entire data centre. AWS does not have a fully integrated platform, application services are built using different tools and deployed to AWS as a separate process although via configuration integration can be achieved to an extent.

Criteria	AWS & Eclipse with Cordova and Java Web Services	MS Azure & VS2015	Web Hosted Stateless Services (Java or WCF), PhoneGap, Cordova
Single Integrated dev environment	No	Yes	No
Tools for cross-platform mobile development	Yes but requires different tools for Server and Client testing	Yes	Yes Requires separate tools for client and server development
Familiarity	Yes I have limited java experience but not confident that can be learned in timescales	Yes I have used over the last 2 years on various assignments and projects	Yes I have limited java experience but not confident that can be learned in timescales
Documentation & support	Yes – good online documentation and support but scattered across internet	Yes One stop shop at Microsoft	Yes – good online documentation but scattered – web hosting setup can be complicated and error prone
Integrated testing/debugging	Yes – but requires additional integration of third party tools	Yes, can test debug as you go via Emulators or via plugged-in devices	Yes – but requires additional integration of third party tools
Easy to deploy target platform	Not so simple as AWS would be configured out-with dev environment tools	Yes – can deploy Services direct from VS to Azure	Not straight forward – lots of different parts – likely configuration issues
Monitoring	AWS has built in monitoring for the infrastructure however services monitoring would need to be built into the back-end services	Yes – fully integrated monitoring capabilities built into Service Fabric services and the Azure platform	Hosting environment has monitoring however services monitoring would need to be built into the back-end services
Scalability	AWS provides auto scaling of Infrastructure components, however services would need to be designed with scalability in mind	Azure and Service Fabric provides auto scaling of Infrastructure components and services integrated as part of the platform	N/A – hosting usually a set number of servers, although can be shared is typically physical or virtual servers aligned to a specific business

After comparing the technologies on the criteria set above, the conclusion for the technologies that will be used is **Microsoft Azure and Visual Studio 2015**. The programming language use for server side development will be C# with service fabric. For client side Xamarin C# and XAML.

3 DESIGN - PROCESS OVERVIEW

3.1 PROPOSED USER PROCESSES

This Sections outlines the high-level interaction with the application outlining the main actors and the tasks they would be able to carry out. These processes are initial designs and were deduced from the research of similar applications such as just eat and more specifically, Orderella.

To illustrate the interactions, use case diagrams and BPMN charts are created. Each of these have their advantages:

1. A use case diagram is a high-level diagram that shows the tasks a user is able or should complete within the application.
2. A BPMN Uses the Use Case diagrams principles and expands on them showing a flow of the main interactions of the users with the system and each other.

3.1.1 Customer

3.1.1.1 Customer Use case Description

Use Case	Description
Register	The customer registers to the app with email and password
Login	The customer can login and use the applications Customer interfaces
Select pub	Allows the customer to select the pub they wish to browse and order drinks at.
Create order	Allow the customer to create an order: • Browse pubs drink catalogue • Select drinks to order • Apply any discounts that may be on offer • Setup repeat order
Payment	Allows the customer to complete the order via online payment method such as PayPal, or by bank card (Simulated in MVP version).
Cancel order	The customer can cancel the order and clear the order basket, this can only be done before payment.

3.1.1.2 Customer Use case

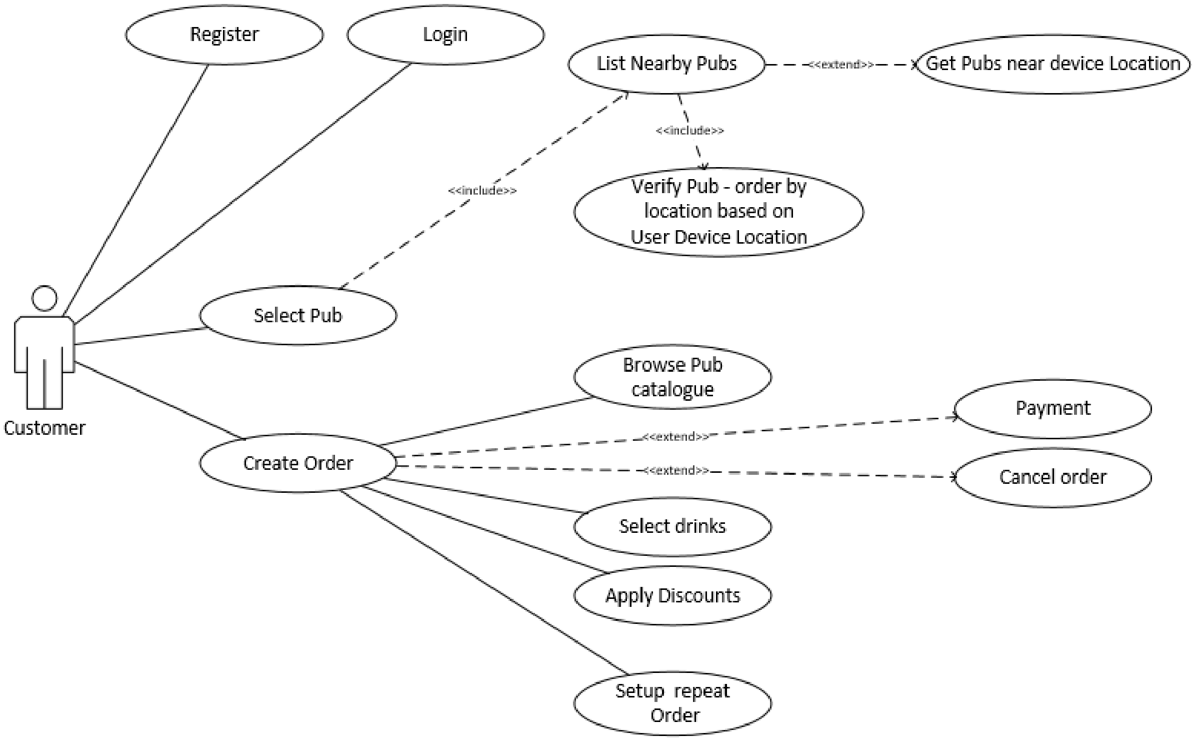


Figure 1 -Customer Use case

3.1.2 Bar User

3.1.2.1 Bar user Use case Description

Use Case	Description
Register App	The bar user/ bar staff will be given an account and told to register with work email and password.
login	The bar user will be able to login to the system and access the application but is limited to the permissions given by bar admin.
Process Next Order Queue item	This is broken into other tasks which include; • Notifying the Customer via app that order is being processed • Pouring drinks • Completing order and determining the type of delivery to the customer (waiter or Bar collection)

3.1.2.2 Bar User Use case

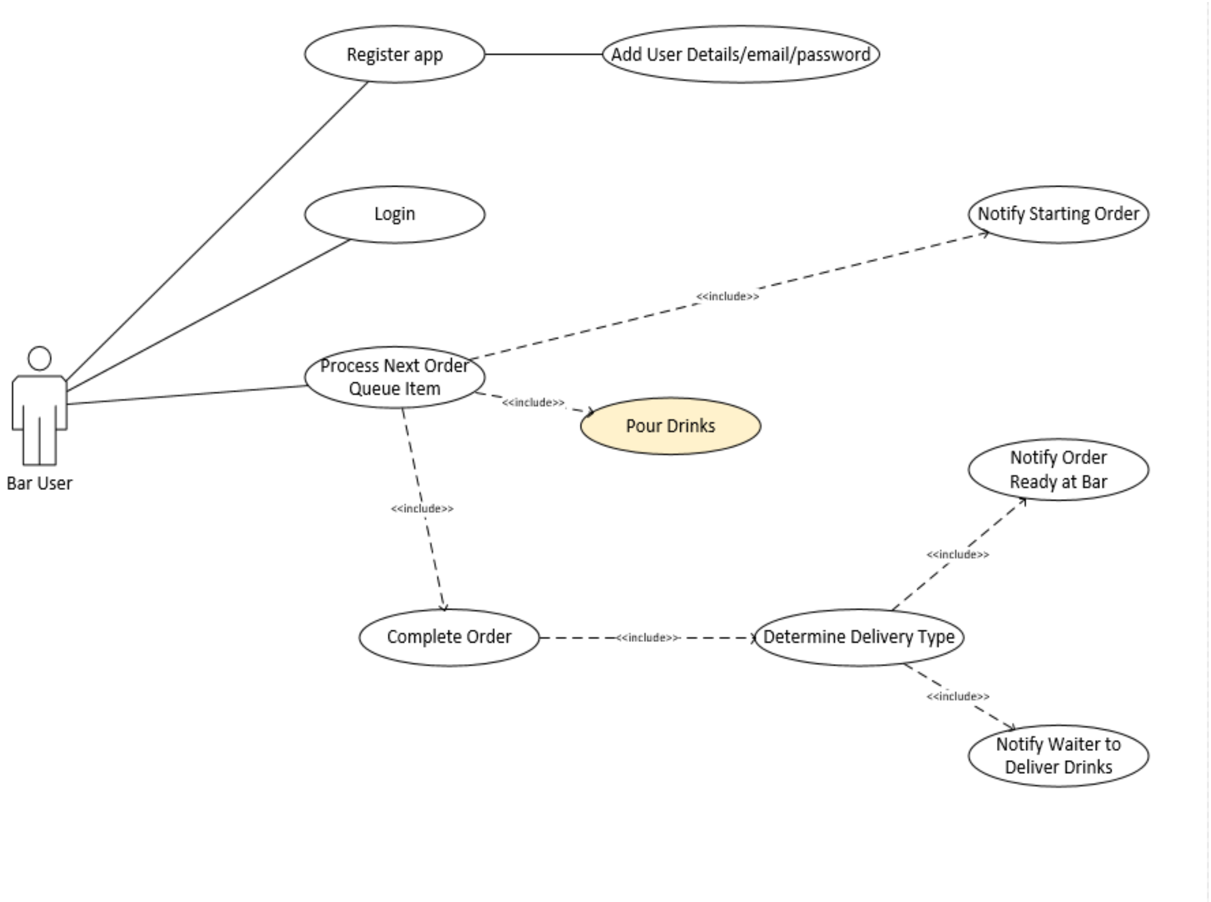


Figure 2 - Bar User Use Case

3.1.3 Bar Admin

3.1.3.1 Bar Admin Use Case Description

Use Case	Description
Service Registration	This is the task of registering for the service, it includes; Adding the details of the bar (Address, location). Setting up the service subscription and the setup of Administrative User credentials.
Login	After registration, the user can login to their account and access the applications administrative features
Setup Inventory	This task illustrates how the admin can setup the bars inventory. This task would include; selecting drinks from a global catalogue and the addition of Items to bar catalogue (drinks and pricing)
Manage Bar Users	This task allows the Admin to create delete and edit Employees accounts to the application. From here the admin can set permissions for employees allowing or denying them access to areas of the application.
Reports	This task will be for the admin to see reports on areas such as sales, pricing and drink favorites.

3.1.3.2 Bar Admin Use case

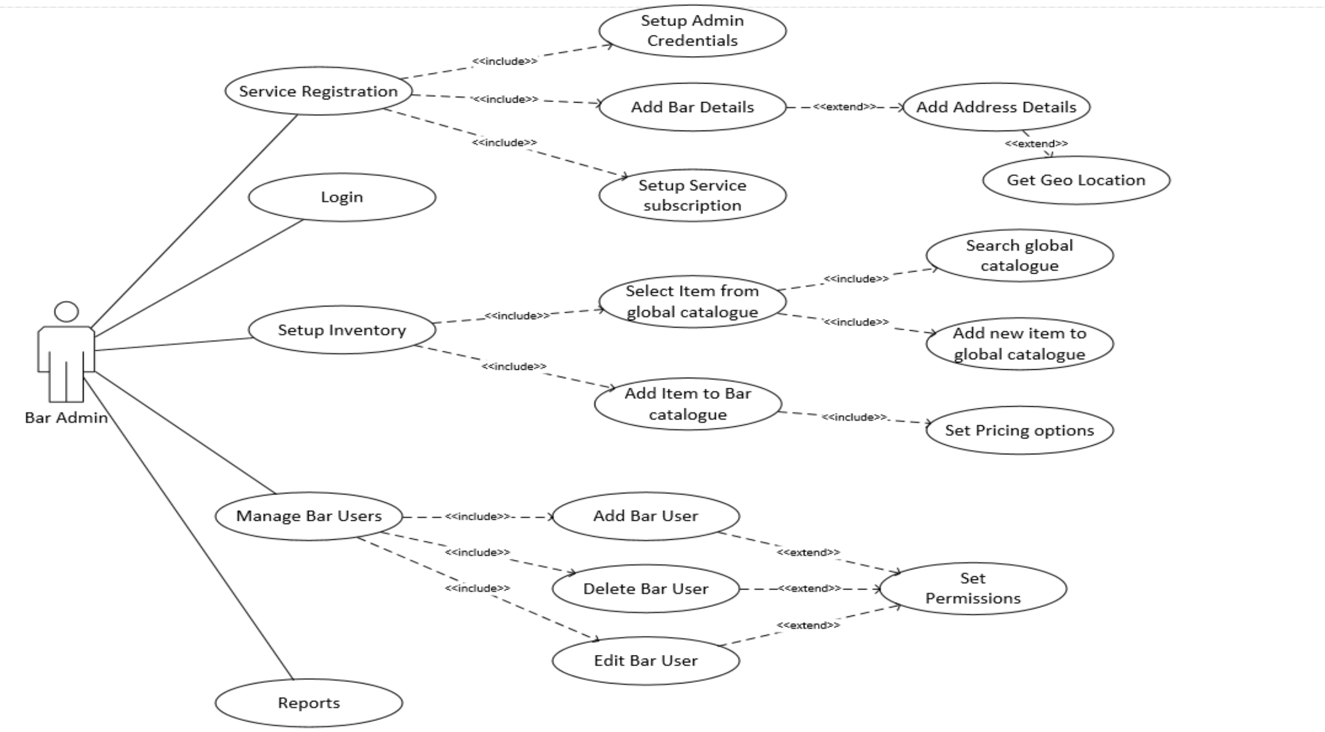


Figure 3 - Bar Admin Use case

4 Design - Requirements

To ensure that a project has a clear direction and method of working it must have a defined set of requirements. The importance of this is proven in many software development projects as without properly defining requirements, projects tend to go overtime and over budget due to requirements creep.

The requirements shown were produces on the basis of the market research described in the research section.

Value	Rating	Description
1	Critical (MVP)	This requirement is critical to the success of the project. The project will not be possible without this requirement.
2	High	This requirement is high priority, but the project can be implemented at a bare minimum without this requirement.
3	Medium	This requirement is somewhat important, as it provides some value but the project can proceed without it.
4	Low	This is a low priority requirement, or a “nice to have” feature, if time and cost allow it.
5	Future	This requirement is out of scope for this project, and has been included here for a possible future release.

4.1 Functional Requirements

Req#	Priority	Description	Use Case Reference
FR-001	1	A bar administrator must be able to register and subscribe to the system, this includes giving details of their name, location etc. and creating an administrative account.	Bar Admin (register)
FR-002	1	A user must be able to register for an account to the system. This includes the user giving details including, Name, age, payment details etc. A widget shall be defined in the repository via a unique identifier and name combination.	Customer(register)
FR-003	1	Bar administrators must have access to the catalogue of drinks for selection to comprise the online menu of the bar.	Bar Admin (Add item to bar catalogue)
FR-004	1	Users and bar administrators must be easily able to login to their accounts.	All users (login)
FR-005	2	Users must be able to easily navigate the application and select the drinks they wish to order; these selections go into a basket interface.	Customer (Create Order)

Req#	Priority	Description	Use Case Reference
FR-006a	1	The basket interface must have the option to remove drinks, add drinks or proceed to checkout	Customer (Create Order)
FR-006b	2	Intelibasket – refers to an active basket that can be set to produce repeat orders at given intervals.	Customer (Create Order)
FR-007	3	Checkout interface must give the user a clear pricing of the drink selection and the option to cancel or proceed with payment.	Customer (Payment)
FR-008	3	Notification must be sent to Bar Admin device to notify that an order has come through. From this the order, can be comprised and when complete notification alerts user device.	Bar user(Process next order queue item)
FR-009	4	Users have a running total of how much they have spent in the session on the application interfaces.	
FR-010	3	All checkout sales should be logged against the user	Bar Admin (reports)
FR-011	2	Bar admin should be able to add or remove drinks from the bar menu	Bar admin (setup inventory)
FR-012	1	Users must be able to select multiple of the same drink	Customer (Create Order)
FR-013	4	Users must be able to add a basket order to a "Rounds tab" which remembers the order for the next time they wish to purchase.	
FR-G-014	1	The application should allow the user and bar administrator to log out of the session.	
FR-G-015	1	All accounts must be secured behind password protection.	

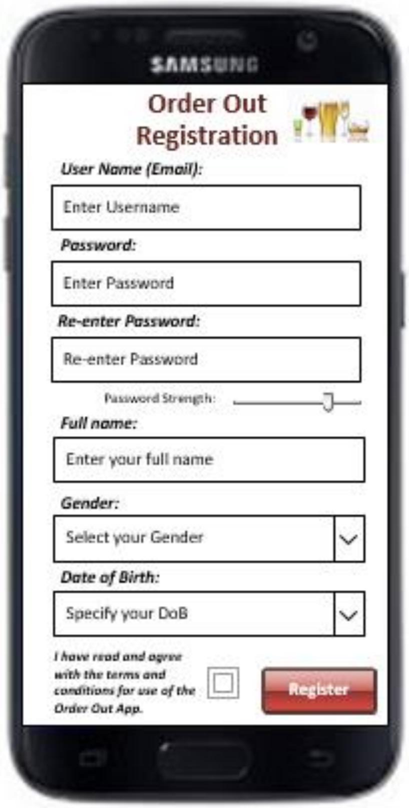
4.2 Non-Functional Requirements

ID	Requirement
NFR-001	The application should be intuitive and performant when a user is navigating and selecting options.
NFR-002	The application should be created for android and easily downloadable onto a mobile device.
NFR-003	The application is a prototype however must be designed with a multitier architecture that allows extendibility and scalability features to be implemented in the future.
NFR-004	The prototype application must be able to work efficiently with up to 10 users at a time
NFR-005	Application being cross platform (including IOS, Android)

5 Design – Initial User Interface

Interface designs are based on the requirements and research shown previously. These are initial User interface designs and are subject to change.

5.1 Register screen consumer



Order Out Registration

User Name (Email):

Enter Username

Password:

Enter Password

Re-enter Password:

Re-enter Password

Password Strength:

Full name:

Enter your full name

Gender:

Select your Gender

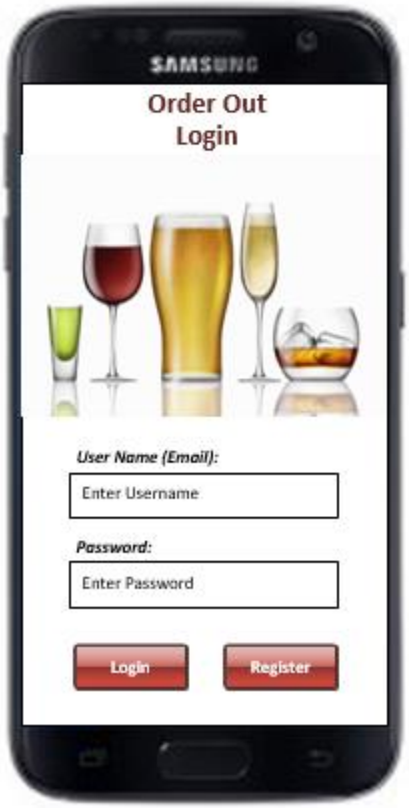
Date of Birth:

Specify your DoB

I have read and agree with the terms and conditions for use of the Order Out App. □

This screen allows a customer to register to use the Application.

5.2 LOGIN SCREEN CONSUMER



Order Out
Login

User Name (Email):

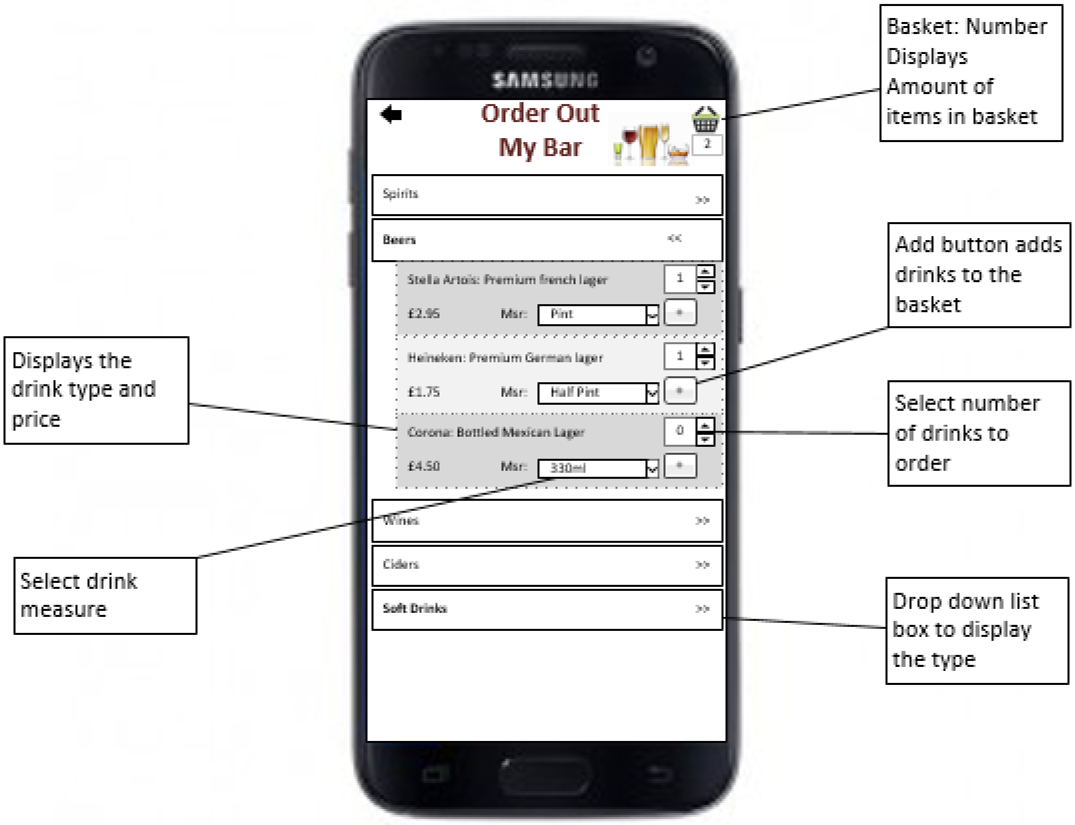
Enter Username

Password:

Enter Password

Once registered a customer can login to use the application

5.3 Menu screen



Stella Artois: Premium french lager 1
£2.95 Msr: Pint

Heineken: Premium German lager 1
£1.75 Msr: Half Pint

Corona: Bottled Mexican Lager 0
£4.50 Msr: 330ml

Basket: Number Displays Amount of items in basket

Add button adds drinks to the basket

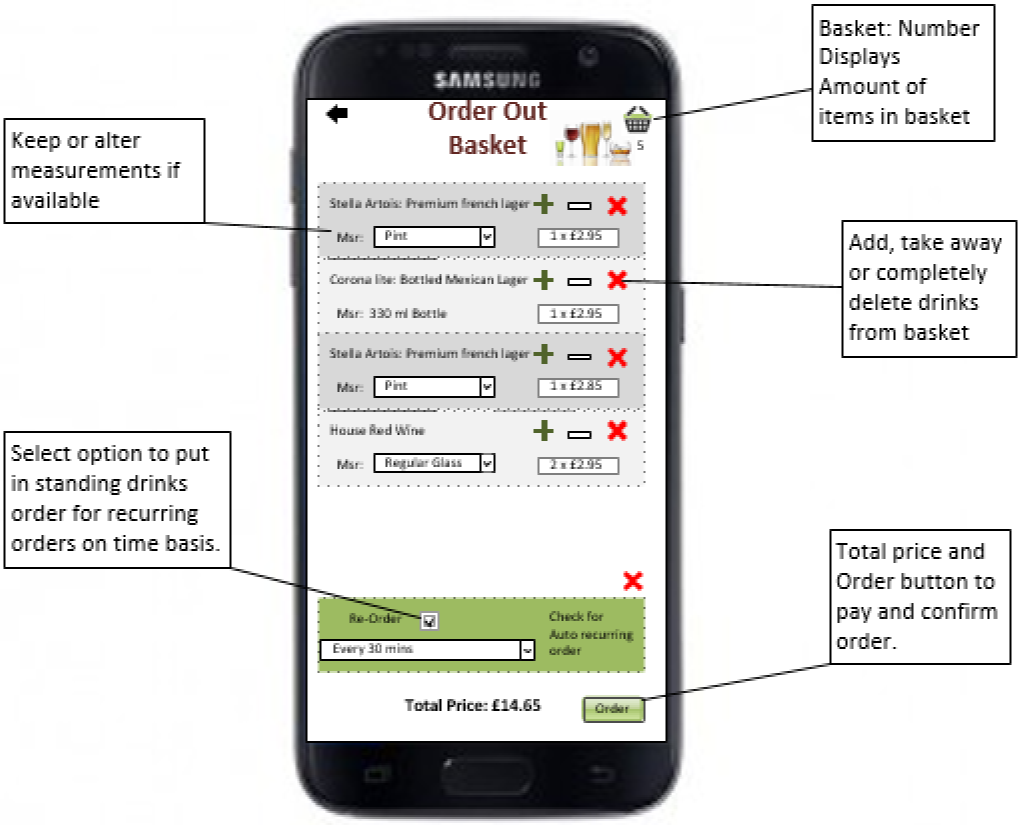
Displays the drink type and price

Select number of drinks to order

Select drink measure

Drop down list box to display the type

5.4 Order screen



Order Out Basket

5

Stella Artois: Premium french lager

Msr: Pint

1 x £2.95

Corona lite: Bottled Mexican Lager

Msr: 330 ml Bottle

1 x £2.95

Stella Artois: Premium french lager

Msr: Pint

1 x £2.85

House Red Wine

Msr: Regular Glass

2 x £2.95

Re-Order ☑

Every 30 mins

Check for Auto recurring order

Total Price: £14.65

Basket: Number Displays Amount of items in basket

Keep or alter measurements if available

Add, take away or completely delete drinks from basket

Select option to put in standing drinks order for recurring orders on time basis.

Total price and Order button to pay and confirm order.

6 PROJECT MANAGEMENT

6.1 PLANNING

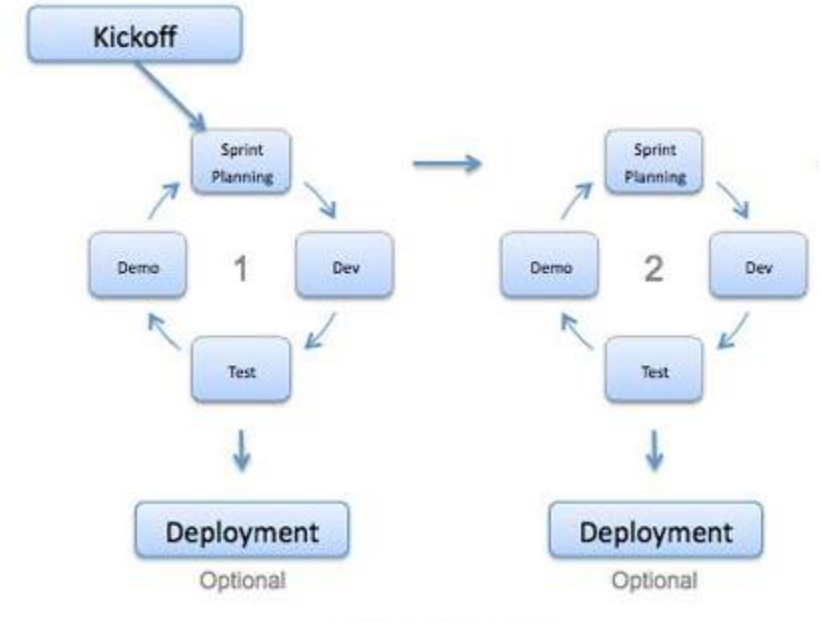


Figure 4- Agile Development Model

With all large development tasks, planning is a vital part in the development of the project to ensure that deadlines are being met. The development model chosen for this project was the agile model shown above.

The first task of the design stage was to take the initial requirements and objectives of the project and break them down into reasonable and manageable sized tasks. To achieve this many aspects of the project and time management had to be considered including, difficulty of task, estimated time, other commitments out with university and other module commitments such as exams and coursework's. Several milestones were also identified by the project deliverables. This task list can be found in appendix.

Also, taken into consideration were any foreseeable problems that could arise throughout the development process. A risk analysis was produced, listing many possible problems and detailing proposed solutions. The combination of each risk frequency and severity were considered and the result of analysing these risks as well as the tasks complexity provided a good method to estimate timeframes for tasks.

The task list was then converted into a Gantt chart. This allows for a clearer visual representation of the tasks against time. It also gives an indication of critical tasks lines and what tasks have leeway for timescale. The Gantt chart was key to keeping on top of project deliverables, updated when situations changed and refereeing back to ensure deliverables were on track and in line with deadlines. Thus, far the Gantt chart has been updated during the project to best reflect any substantial changes from the original task timelines.

6.2 MANAGEMENT

To ensure the safety of the documentation, notes and code of the project an external hard drive was used to make backups. This includes making backups of any recent application changes and any major version change on documentation. Previous versions of the application and older iterations of the documentation. This was done to minimise the likelihood of work being lost due to incident such as drive failure, bugs or theft but also backups of previous versions should be kept in case any major fault in coding needed to be reversed to a previous working version.

Meetings were setup between myself and my supervisor to ensure that at key stages the project was on the right track. These meetings were conducted verbally and a note was kept of some suggestions of areas and methods of research that were useful for the progress of the project.

Obeying to coding standards during the development of the project is vital in keeping code easy to understand and read. These standards are keeping uniform layouts and styles throughout, ensuring that all variable and functions adhered to follow the camelCase naming convention.

6.3 DIFFICULTIES

To date the project has been slightly behind schedule. As revealed in the Gantt chart, the hand in of the interim report should have concluded most design tasks, permitting the remainder of the time for development and final documentation. However, as the main database design has not been complete there will be a slight setback giving tasks in the implementation stage a narrowed timeframe for completion.

7 FUTURE PLANNED WORK

The planned work from now until the 10th February 2016 would be to get a minimal viable product (MVP) working end to end, this is achieved in Sprint 1. The second sprint from 10th February 2017 to 24th March 2017 will drive through the priority 2 requirements (priority 3+ requirements are on a time available basis).

The sprint planning sessions will outline the work to be carried out during the sprint.

The initial designs for interfaces and user processes are underway as sprint 1 had started at the time of this report submission. Work to be completed in each sprint includes:

- Microservice API's and data models updates (design, build test)
- App Feature Development updates (Xamarin)
- Final Report Documentation Updates
- Presentation preparation Updates

The bulk of work is taken up by the design and application development as it will entail further research.

8 REFERENCES

Apache Cordova. (2016, December). *Cordova*. Retrieved from http://cordova.apache.org/

eat, J. (2015, 04 01). *annual report business model*. Retrieved from Just eat: http://www.just-eat.com/wp-content/uploads/2015/04/Annual_Report_our-business-model.pdf

Eclipse. (2016, December). *eclipse*. Retrieved from http://www.eclipse.org/

Flowtab. (2013, December 01). *Flowtab home*. Retrieved from Flowtab: http://flowtab.com/

Just eat. (2016, December 21). *just eat home*. Retrieved from Just eat: https://www.just-eat.co.uk/

Microsoft. (2016, january 01). *What is XAML?* Retrieved from Microsoft: https://msdn.microsoft.com/en-us/library/cc295302.aspx

Orderella. (2016, December 12). *Orderella Home*. Retrieved from Orderella: http://www.orderella.co.uk/

parallel project training. (2016). Retrieved from http://blog.parallelprojecttraining.com/news/importance-good-business-requirements-analysis-projects/

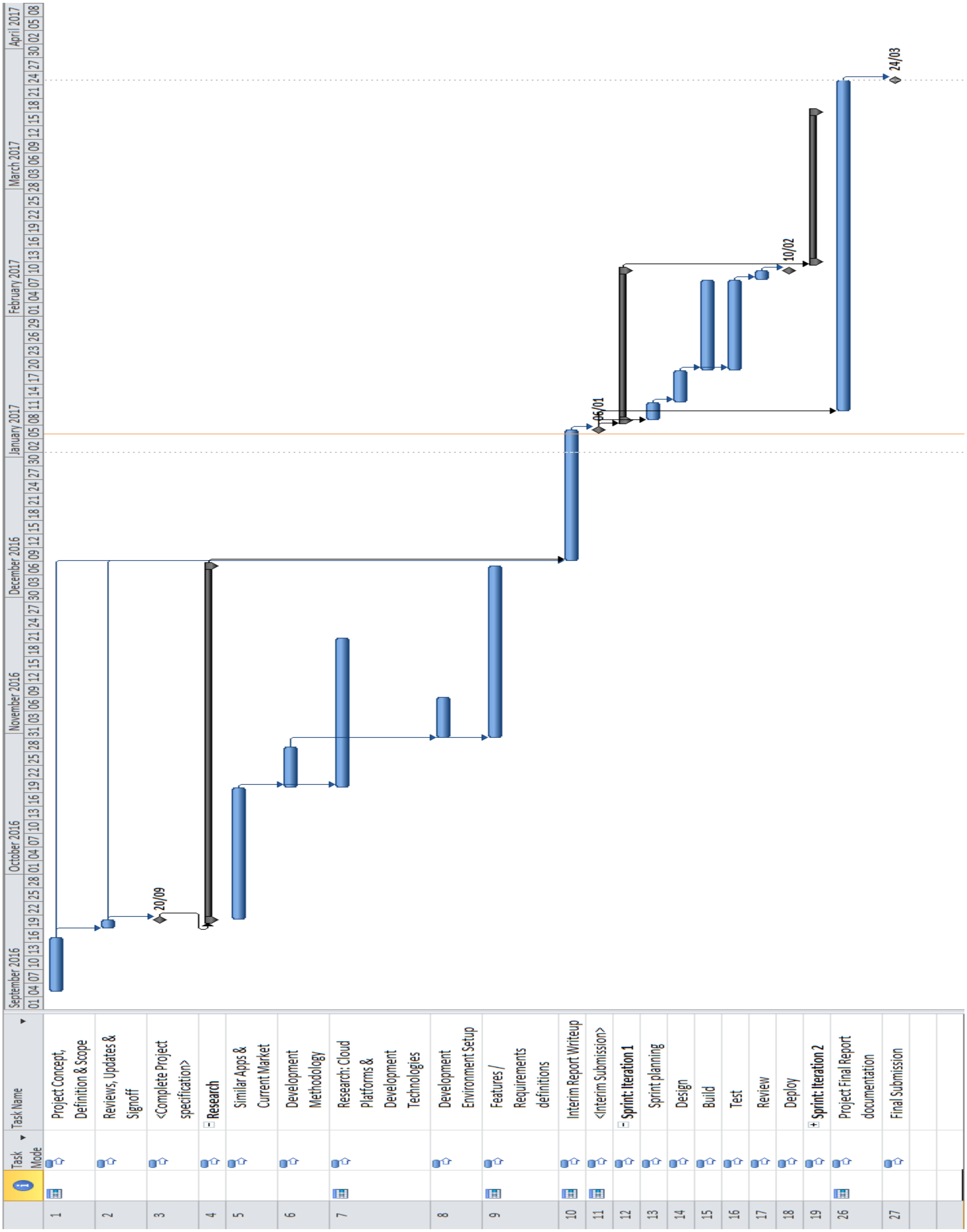
Quora. (2016, 12 9). *Quora what is the just eat business model*. Retrieved from Quora: https://www.quora.com/What-is-the-Just-Eat-business-model

Studio, V. (2016). Retrieved from Visual Studio: https://www.visualstudio.com/vs/

The Guardian. (2016, September 5). *Can't face the queue at the bar? Don't worry, the next round's on Orderella*. Retrieved from The Guardian: https://www.theguardian.com/small-business-network/2016/sep/05/drinks-bar-pub-app-orderella

Xamarin. (2016, December). *Xamarin*. Retrieved from https://www.xamarin.com/

APPENDIX A – PROJECT PLAN GANTT CHART



	Task Name
1	Project Concept, Definition & Scope
2	Reviews, Updates & Signoff
3	<Complete Project specification>
4	Research
5	Similar Apps & Current Market
6	Development Methodology
7	Research: Cloud Platforms & Development Technologies
8	Development Environment Setup
9	Features / Requirements definitions
10	Interim Report Writeup
11	<Interim Submission>
12	Sprint: Iteration 1
13	Sprint planning
14	Design
15	Build
16	Test
17	Review
18	Deploy
19	Sprint: Iteration 2
26	Project Final Report documentation
27	Final Submission

APPENDIX B - COMPUTING HONOURS PROJECT SPECIFICATION FORM

Project Title: “Order out” Cloud based ordering service for drinks at a pub

Student: Andrew McLaughlin
Banner ID: B00289311

Supervisor: Glenn Affleck

Moderator: Mark Davison

Outline of Project:
The idea is to create a cloud based prototype application that allows registered customers to interact with signed up Pubs to purchase drinks without having to queue at the bar. This will give a relationship between the customer and pub.
This will allow an easy and fair ordering system by making busy bars less congested and stop people waiting or getting “skipped” in the queue.
A Bar would register to the app and supply details of their drinks menu.
Customers would be signed up to the app giving details about themselves.
Customers would interact with the Pubs menu by selecting drinks and going through payment.
When an order is made the bar staff can see the order via a tablet optimised interface and respective table number or name of customer and begin to prepare drinks.
Once drinks are made staff will update the order and the customer will be notified to collect from the bar or be delivered by staff to the table.

The application will be a prototype so payment methods will not be fully implemented.

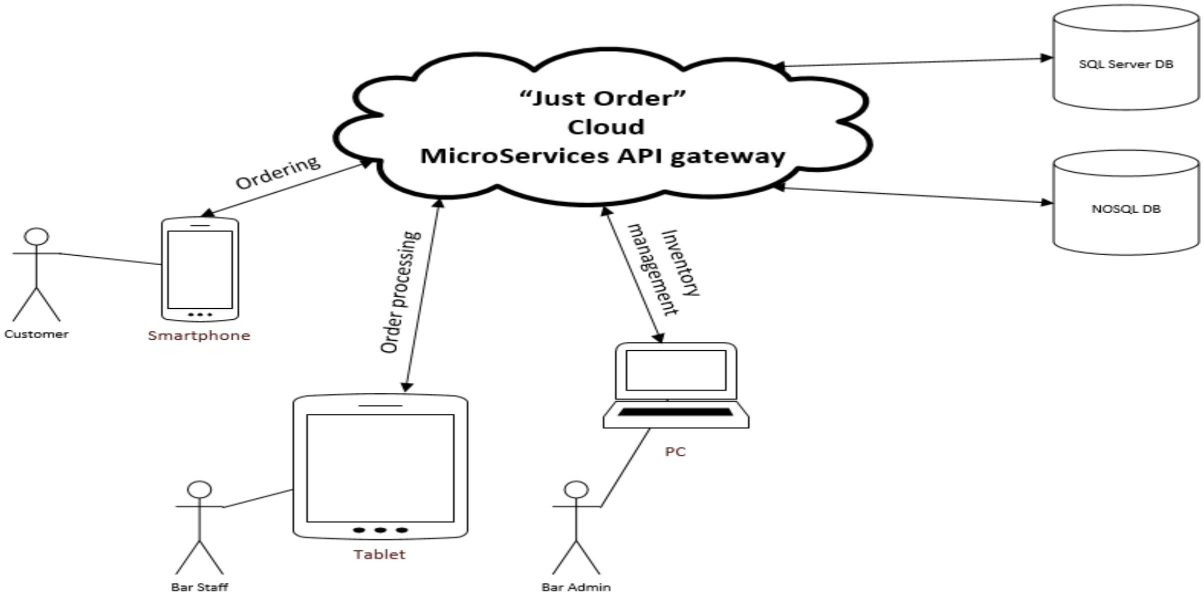


Figure 5- Conceptual Architecture

The diagram indicates the various types of interaction between actors and the application interfaces. Some of these are listed below:

- Customer can order via their smartphone
- Bar staff can update and process orders through a tablet device
- Bar admin can update drinks menu through PC

A Passable Project will:

- *Full end to end functionality of the app*
- *Most priority functionality working to reasonable standard*
- *Good standard of documentation which is understandable.*

A First Class Project will:

- *Full end to end functionality of the app*
- *All priority and some additional functionality incorporated and working to a good standard.*
- *High standard of documentation that is easy to understand and follow the processes of the project.*
- *Good use of UML and Microsoft Project to illustrate the Project Processes and Design*
- *Well thought out and concise application code*

Reading List:

- Azure Cloud based architecture and MicroServices
- Service fabric platform implementation
- NoSQL database design and implementation
- Android Device development

Resources Required: *(hardware/software/other)*

- PC
- Azure
- Service Fabric
- C#
- WCF
- NoSQL(Document store)
- SQL Server
- Android SDK
- Microsoft Office Suite

Proposed Marking Scheme:

Introduction	**10**
Project planning	**25**
Implementation	**25**
Implementation of extra features	**5**
Documentation	**20**
Conclusion/Evaluation	**10**
Critical Self- Appraisal	**5**