6COSC006W - Final Year Project Report

Contactless Loyalty

Shouyi Cui (w1618594)

This report is submitted in partial fulfilment of the requirements for the **BEng (Hons) in Software Engineering Degree** at the University of Westminster.

Supervisor: Barbara Villarini

School of Computing & Engineering
University of Westminster

Date: 13th July 2020

Declaration

This report has been prepared based on my own work. Where other published and unpublished source materials have been used, these have been acknowledged in references.

Word Count: 18848

Student Name: Shouyi Cui

Date of Submission: 13th July 2020

Abstract

This project aims to resolve the issue of the many loyalty cards that one person usually hold. Research about the alternative digital solution was carried out to define what can be developed as a new feature not yet in the market. Based upon a statistical approach and a real case business scenario a solution is provided with the Web-NFC API.

The solution takes into account the budgeting and different limitation of small brands. The target of the service can be considered as the retailers in the small shops. The use of the NFC technology gives a higher level of engagement between the retailer and the customer. At the end of this project, the software developed gives the opportunity for the customer to engage with the retailer. From the retailer point of the view, the web application implemented for this beta version does not meet all the necessary requirements.

Acknowledgements

First and most importantly I would like to thank my supervisor Barbara Villarini for her understanding, recommendations and support in my project.

I am grateful for the support and love from my family throughout my study in London.

I would like to thank all the people involved in the process of my project implementation.

Special thanks go to Laurence T. who provided me with the initial NFC tag used for testing as well as the technical help in resolving bugs.

Recognitions go to i-movo and their team for providing me knowledge and experience in some of the technical aspects of this project.

Other gratitude goes to Nabila U., Rita A., Hela M. and Luke S. who provided me guidance and revision for the process of writing the report.

I would like to acknowledge the University of Westminster for providing me the great opportunity of learning Software Engineering.

Last but not least, Alessandro D. whose perseverance in the belief of statistical data convinced me to make the survey used as resource for gathering the requirements.

"For a better future, contactless

but still very close to each other."

Shouyi Cui

Table of Contents

List of Figures

List of Tables

1. Introduction

This chapter aims to give an overview of the context of this project, the problem the project is trying to resolve and the objectives.

1.1 Problem Statement

Nowadays there are many ways a retailer can reward their most loyal customers. There are loyalty schemes almost for everything, from the coffee shops to flights. The more money you spend with a company, the more likely they are going to offer you special discounts because it is easier and more convenient for the business (Jovancic, 2019).

The current most common type of loyalty schemes available in restaurants such as Starbucks, Caffè Nero and Costa Coffee require the customer to register online on their service and then download a mobile application (DevTeam.Space, 2020). Sometimes it is the cashier that enables the digital stamp or other times it is the record of the purchase in the customer account. Other smaller food restaurants usually have a classic paper card where the cashier can make a stamp on it. The cards are usually made with empty icons that can be filled with the stamp to represent the accumulation of loyalty points.

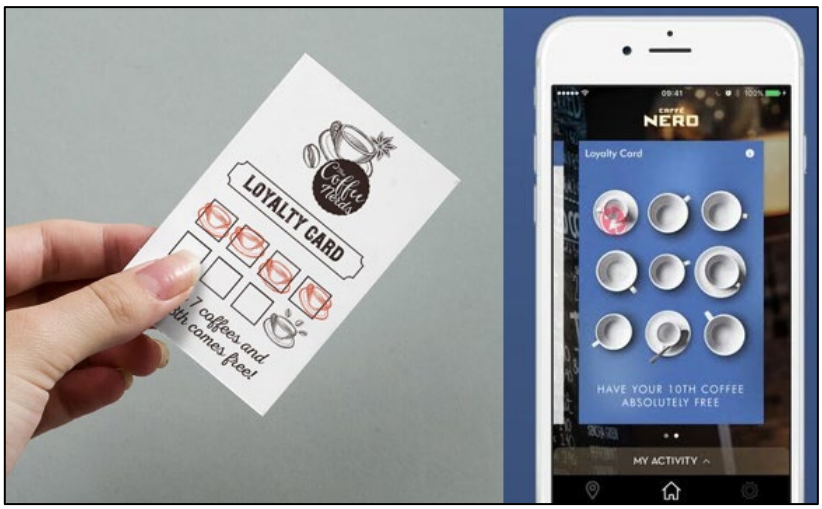


Figure 1. Paper loyalty card and loyalty Mobile App

In the example above (Figure 1.) on the left there is an example of a paper loyalty card and on the right a mobile loyalty app. The concept is similar but the way it works is completely different because the former is physical and the latter one is digital.

This purpose of this is to enable something in between the two existing solutions by using the NFC (Near Field Communication) technology, also known as contactless. This innovation is now available in most of the devices in the world and it is becoming more popular.

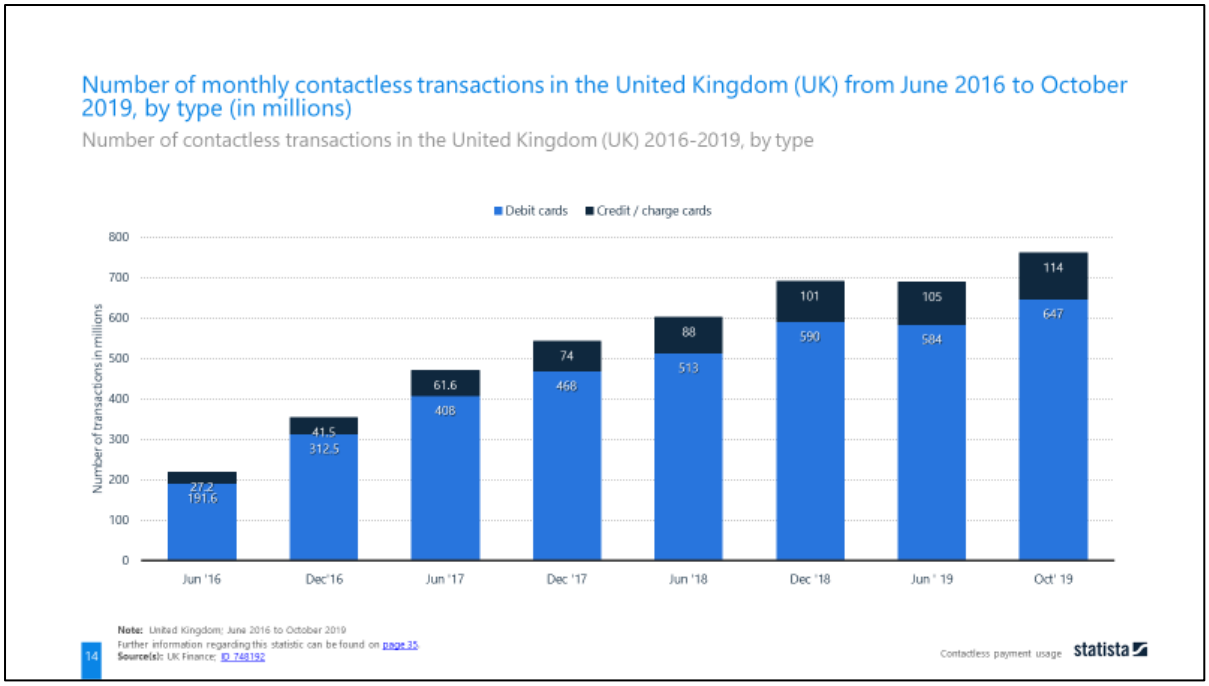


Figure 2. Monthly contactless transaction in the UK from June 2016 to October 2019

This technology is mostly used for payments with a small amount of money involved because it does not require any type of validation. The lack of validation makes the card more vulnerable to fraud (loveMoney, 2019), but that is a reasonable compromise for fast payments. Although it is also possible to set a limit for contactless card payment by the merchant or the card issuer.

Moreover, from an ethical and ecological perspective, this project has the potential to save the waste of plastic and paper by decreasing the demand of printed paper cards and plastic cards that "[...] have actually been the most requested gift in America" (Long, 2015).

1.2 Aim and Objectives

The overall purpose is to create a Web App that uses the Web-NFC experimental feature on the Google Chrome browser on mobile (Bhaumik, et al., 2020) and allows both customer and retailer to manage their loyalty experience the way they want.

The main scope is to allow the customer an easy way to collect stamps or points without the need of a native mobile application. For the retailer, the advantage is a system where the loyalty experience is not restricted to a proof a purchase, but it could be a number of visits throughout a month or maybe an interaction with a new product in the store. With the use of the NFC tag, the retailer is also able to reuse the same piece of technology without investing into more complex machinery.

To achieve the desired goal, I will need to complete this list of objectives:

- Gain in-depth understanding on the NFC capabilities.
- Research NFC security known issues and always be aware on related news.
- Develop a prototype to use as demo for stakeholders.

- Constantly receive feedback from different sources to gain a wider perspective of the project.
- Adopt source code management systems to make sure there is trace of the progress made in case of work lost or not working as expected.
- Use a time tracking application for time management.
- Clear documentation of the project in terms of code and documents.
- Deployment of the project on a stable environment such a cloud service
- Implement an algorithm that compress the small amount of data
- Work with an external API that can send a digital voucher to the customer

Moreover, I would like to achieve these additional features (in descending order of importance):

1. Creation of own images and logos
2. Customisation of the interface per type of user
3. Gamification of the user experience

2. Background

This section introduces the literature review of the project, a comparison of similar or relevant applications for the same customer reward. In addition, a discussion on the possible approaches for the intended solution is included.

2.1 Literature Survey

The following sections of the literature review will cover different aspects of the NFC. Starting from the beginning of this technology and its original creator to the technical distinction of the modern world. The modern enhancement and what are the future capabilities are also discussed within the scope of the project.

2.1.1 The History and Physics Behind

A Russian physicist and inventor called Léon Theremin (also known as Lev Sergeyevich Termen) in 1920 developed a musical instrument (later named after himself) that can produce sounds without being touched. The theremin core principles are heterodyning and capacitance. The former is the result of a combination or mixture of two frequencies (a principle used for FM radios) and the latter is the "ability of a circuit to collect and store energy in the form of an electrical charge" (Fluke Corporation, 2020).

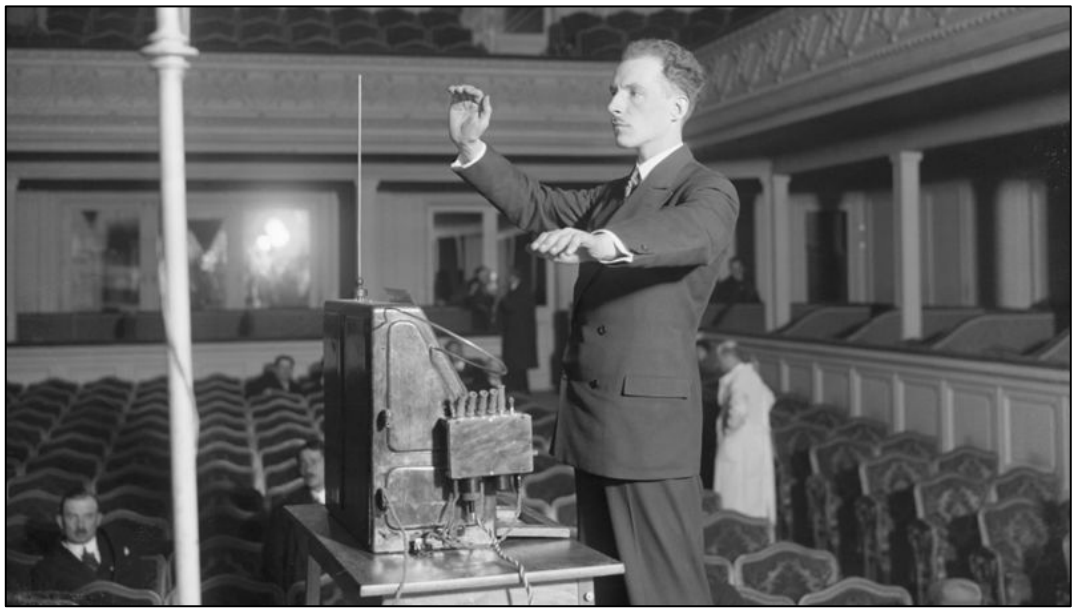


Figure 3. Léon Theremin playing his own invention

The electrical instrument has two metal antennas, one to control the pitch and the other to control the volume. When a hand goes near to an antenna, a natural capacitor is generated, and its capacitance change based upon the distance to the hand. The circuit of the instrument takes the capacitance and set a frequency for the pitch and the volume. Then an inductor inside the

instrument creates the frequency to be combined with the previous one so it can result with an interference that is hearable by the human hear (Huth, 2018).

Later, in 1945 the World War II finally came to an end. On the 4th August in Moscow a group of boys from the Young Pioneer Organization of the Soviet Union went to the American embassy to give a present as a symbol of friendship between the two countries. Averell Harriman, the United States ambassador at that time, took the great wooden ornament (Figure 4) as an important gesture and hung it on the wall of his office. They probably have checked every side of it to make sure it was not going to cause any harm like a Trojan horse, but nobody found anything alarming (Harford, 2019).

Figure 4. The present given to the US ambassador and the hidden device location

Eventually it was found out that the gift was an innovative creation from Theremin commissioned by his government to spy the conversation of the ambassador. It worked secretly for seven years until its discovery and gained the names "The Thing" and "The Great Seal Bug" (Harford, 2019).

The invention of Theremin consisted of a reverse concept of his musical instrument. He created a hidden circuit that had a capacitor that vibrated depending on the voice pattern. The capacitance would set the frequency representing the voice. An interference would be created when beaming a radio frequency signal to the object. This beaming would also power up and activate a response signal to broadcast out, so it could be received and analysed to get the information needed (Crypto Museum, 2015).

This can be conceived as the first example of the modern RFID (radio-frequency identification) technology because of the concept and physics involved.

In fact, the underlying principle of RFID consists of electromagnetic waves and mutual inductance. The latter is a physical principle that describes how the change of current in a coil can produce an electromotive force (EMF) in an inductively coupled coil.

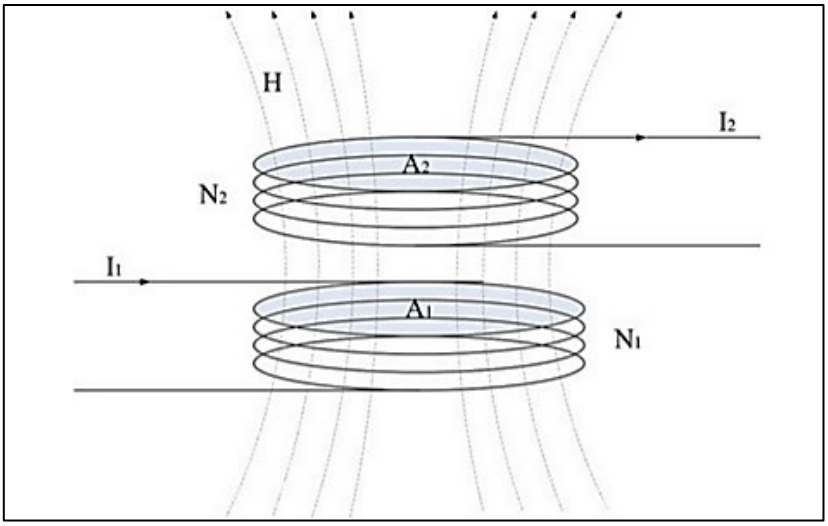


Figure 5. Inductive coupled coils

In the image above (Figure 5) we can consider N_1 to be an initiator that tries to engage to a target N_2. The mutual inductance between the two coils can be calculated by the following formula:

$$M_{12} = \frac{\mu_0 \cdot H(I_1) \cdot N_2 \cdot A_2}{I_1}$$

Being H the magnetic field strength, N the number of loops of the area A, and I the current that flows in the coil (Yang & Hancke, 2017).

2.1.2 RFID (Radio-Frequency Identification)

RFID can be described as a form of wireless communication that uses the aforementioned electromagnetic principle (see 2.1.1) to uniquely identify an object (Rouse, 2007). It is purposely designed for identification because the RFID tags can hold only a small amount of data, usually around a thousand bytes or less (Igoe, et al., 2014).

There are two RFID types of communication mode: active and passive. But first of all, it is essential to define the two actors involved in the exchange: the target and the initiator. The initiator is the device that tries to read or write a tag, it generates the radio field and waits for responses from any target in the field. The target is usually the tag, that will respond with an UID (Unique Identifier Number) to the radio field (Igoe, et al., 2014). Therefore, the communication mode is considered as:

- *Active* when the target is powered independently (e.g. From a battery).
- *Passive* when the target has no power source. It usually gets the power from the radio field. Very similar to "The Great Seal Bug" (see 2.1.1).

At this point, it is worth mentioning there are various type of RFID protocol standards, usually developed by the ISO (International Standards Organisation) along with the major participants in the market. The different standards can change in terms of radio frequencies used (i.e. A lower frequency usually means a shorter read range), data format and data transfer rates (Lowry

Solutions, 2014). These protocols are created for the purpose of having interoperable standards so that the technologies can work together and allow a competitive market from different industries (Igoe, et al., 2014).

2.1.3 NFC (Near Field Communication)

NFC, similarly to the RFID, is also a wireless communication that works on the same physics principles mentioned before (see 2.1.1). It is designed upon the RFID protocols and it is generally possible to interact with the RFID tags (e.g. ISO-14443A tags are compatible with NFC). Its main role is to enable the target and the initiator to communicate by an exchange of meaningful data. This data can be either the capabilities of each other, records or even credentials.

It is important to note that NFC targets are not limited to tags, they can be also programmable devices like smartphones. There are two communication mode exactly like in the RFID: active and passive (see 2.1.2). Moreover, there are three operating modes:

- *Reader/Writer* when a device reads data from a target and/or writes to it.
- *Card emulators* when a device acts like a RFID tag in the electromagnetic field of another NFC or RFID device.
- *Peer-to-peer* when two devices exchange data to each other.

2.1.4 Comparison between RFID and NFC

NFC can be considered as an enhanced version of RFID in the case where the initiator and target are in a short range. NFC is not designed to work the in long range so this limitation cannot be considered a real disadvantage, besides Wi-Fi and Bluetooth technologies are supposed to cover that gap.

Listed below are the common usage of the two technologies (Figure 6).

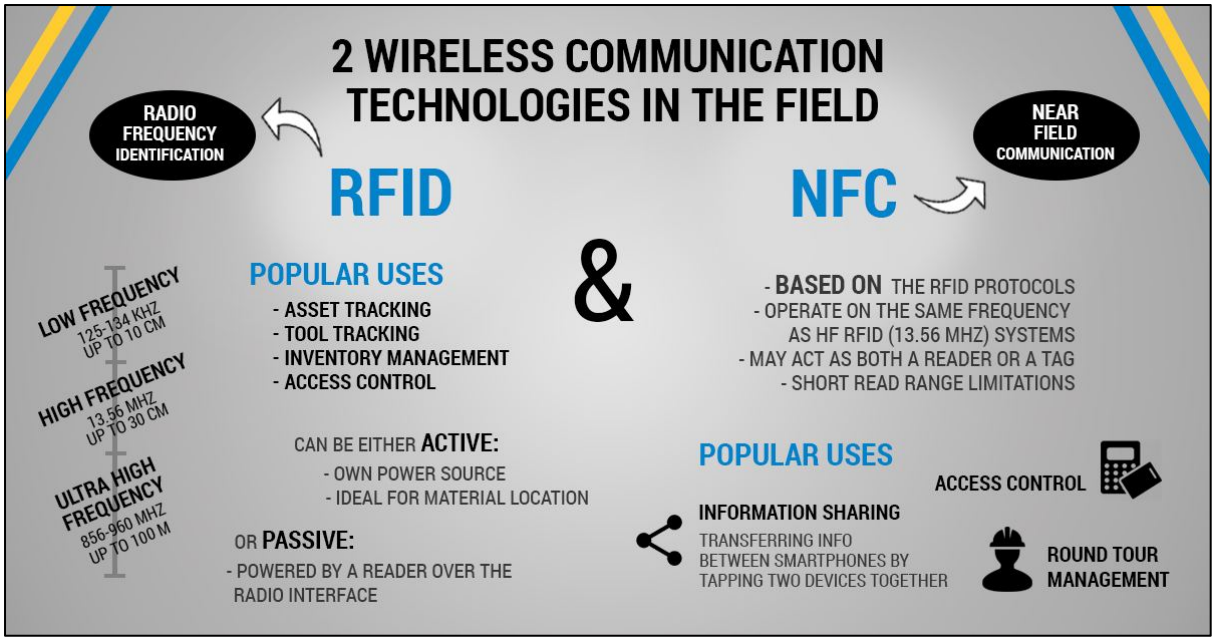


Figure 6. RFID & NFC comparison

A big advantage of NFC is that it has a very low cost in comparison to the RFID, an NFC tag usually cost less than a pound (e.g. NTAG213, NTAG215, NTAG216). The long range advantage of RFID requires the target to be an active tag, and that is where the cost rise. Currently, every single RFID active tag can cost from £25 to £100 depending on the range required. The RFID reader is also very expensive costing from £150 to £1800 also depending on the range required (NextPoints, 2020). On the other hand, NFC readers can cost something around £40 but given that the number of smartphones with the NFC enabled are constantly increasing in numbers, maybe there is not even the need of an additional purchase.

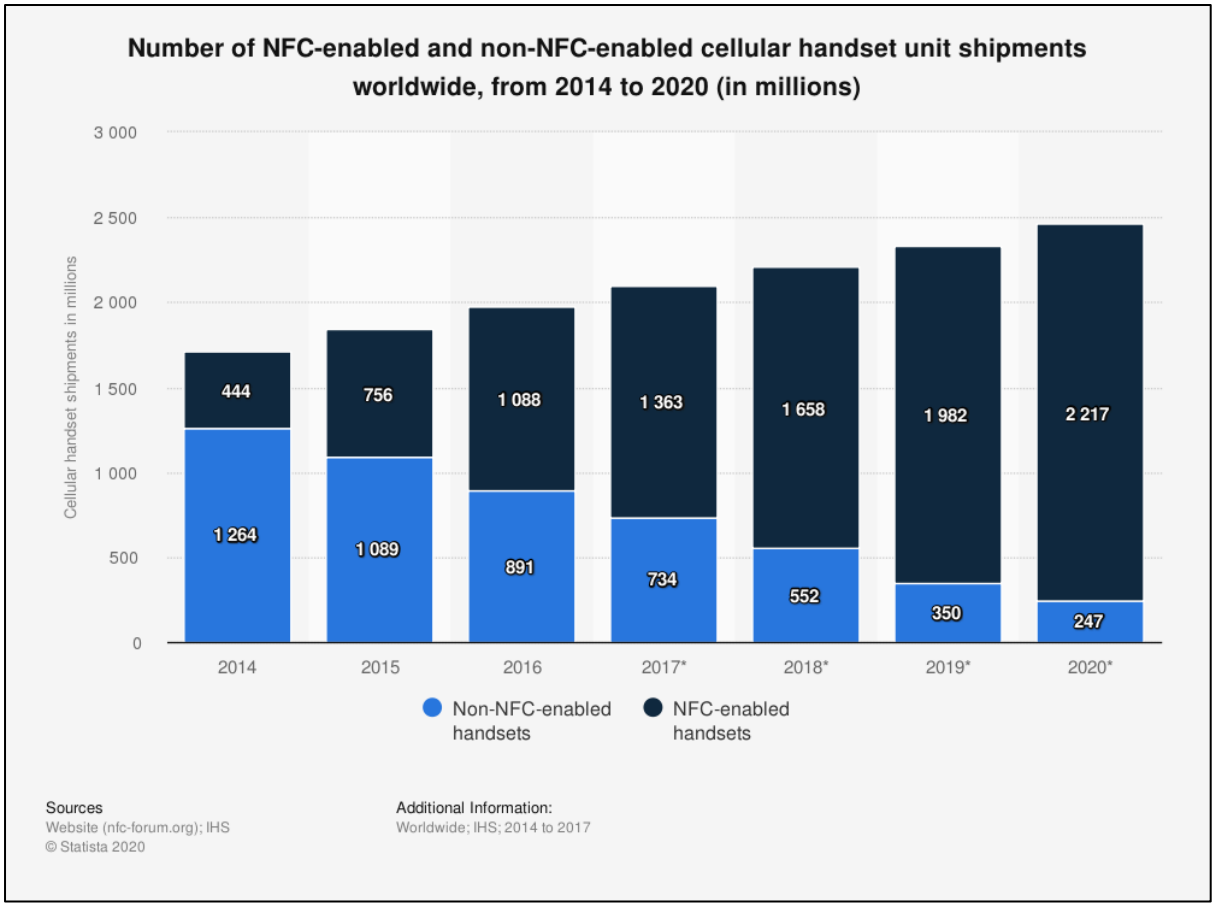


Figure 7. NFC enabled handsets from 2014 to 2020

In the figure shown above (Figure 7) it is possible to see the change over time of NFC enabled smartphones shipped in the world. Although this graph does not represent fully the numbers of all the smartphones in the market, it is fair to assume there is a similar trend because what changes is only the device delivery system.

In 2014 only the 25.99% handsets had NFC. Two years later, this feature increased in availability reaching 54.98%. Now in 2020, that percentage had rose to 89.98% and it is most likely to grow over time (Kenneth Research, 2020).

2.1.5 NDEF (NFC Data Exchange Format)

NDEF is a data format operating across all NFC devices. A common NDEF message contains one or more NDEF records. Each of this record has its own UID, record type, length and payload of data (Igoe, et al., 2014).

A generic record is represented in the figure below (Figure 8).

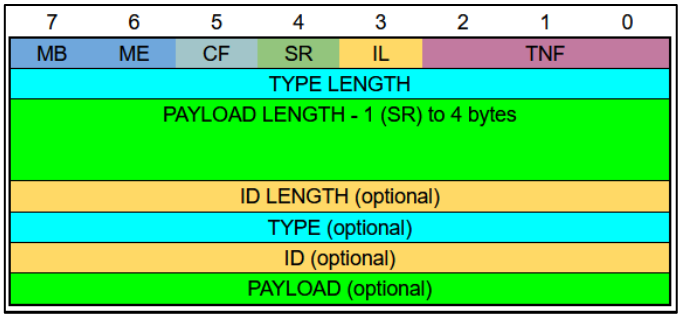


Figure 8. NDEF record structure

- Bit 0-2 indicates the format of the type name.
- Bit 3 indicates the presence of an ID length field.
- Bit 4 indicates a short record.
- Bit 5 indicates whether the payload is chunked across multiple records.
- Bit 6 indicates whether this record is the last in the NDEF message.
- Bit 7 indicates whether this record is the first of the NDEF message.

This is the list of the NDEF record types:

1. **Empty NDEF** (TNF 0) as the name suggest it represent a record with no data and therefore Type Length field, ID Length field and Payload Length field must be 0 and the last three optional fields (Figure 8) must not be present.

2. **Well-known** (TNF 1) which is a set of sub record types standardised by the NFC forum such as text, URL, media and smart posters and handover options.

3. **MIME** (TNF 2) stores binary data with the associated MIME (Multipurpose Internet Mail Extensions) type.

4. **Absolute-URL** (TNF 3) contains the string of the full address that includes protocol and domain name.

5. **External** (TNF 4) is a Uniform Resource Name (URN) with the application data type (e.g. *urn:nfc:ext:domain.org:atype*).

6. **Unknown** (TNF 5) is for storing data that have incomprehensible data and are not associated with a MIME type. The application may assume the latter.

7. **Unchanged** (TNF 6) is a section of a chunked data set, so the payload is spread across multiple NDEF records.

8. **Reserved** (TNF 7) which means reserved by the NFC Forum for future use.

TNF value	Description
0	Empty record
1	NFC Forum well-known type record
2	MIME type record
3	Absolute-URL record
4	NFC Forum external type record
5	Unknown record
6	Unchanged record
7	Reserved for future use

Figure 9. Summary of possible Type Name Format (TNF)

2.1.6 Web NFC

Web NFC is currently an experimental feature which overall goal is to give online sites the ability to read and write NFC tags in a secure and privacy preserving manner (Beaufort & Kenneth, 2020).

At the moment, this technology is limited to NDEF and is available as an original trial in Chrome v.81 until Chrome v.84 (Chrome Origin Trials, 2020) on Android OS smartphones.

The functionality can be enabled in the flags section (chrome://flags/) under the name "Experimental Web Platform features". After the enablement, when surfing on the internet, if there is a website that wants to use NFC features for the first time, it will prompt a request to use it in the page. It is also going to prompt a message asking to turn on the NFC on the device in case it is off, while it is not going to show anything if the feature is not compatible with the device (e.g. iOS smartphones).

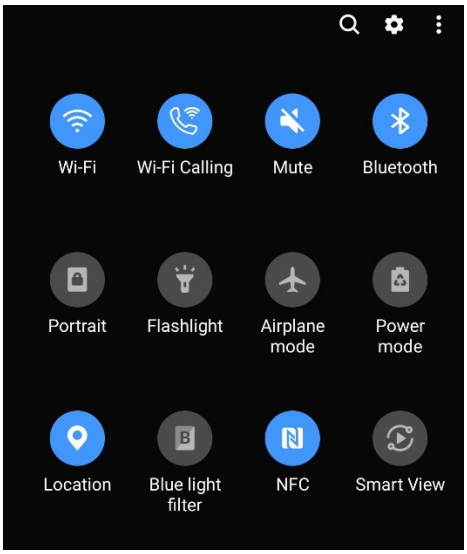


Figure 10. Example of quick dropdown settings of an Android device with NFC

This new enhancement released on January 2020 gives the developers a lot of new potential in various use cases (Kostiainen, 2019).

The benefit of the NFC along with an ad hoc Web Application is that it can improve the UX (User Experience) by making the user interact with the surrounding environment (e.g. treasure hunts).

Hopefully, this Web API will be available in most of the modern mobile browser so that the developers can create more sophisticated solutions overtime. It has the potential of relieving the users from downloading ad hoc native applications on the mobile for simple use case scenarios.

2.2 Review of Projects / Applications

In the following sections, the research on existing projects or applications relevant to the NFC and the intended project implementation will be presented.

2.2.1 Google Pay

Google Pay has a mobile application that links to the user's payment information to create an online payment system and digital wallet. It was developed by Google in 2015, known initially under the name "Google Wallet" and later merged with "Android Pay" (Nieva & Bennett, 2018). There are many features in the application such as image recognition, credit and debit card validation but those are not relevant to the scope of this project.

There are two aspects relevant in the e-wallet. First, the loyalty card system that asks the user to add the details of a physical card of the store by either camera scansion or entering manually. After the card details are successfully added in, the app generates a barcode representing the loyalty card. Secondly, the NFC payment system that uses the HCE (Host Card Emulation) to recreate a previously added credit card to make a payment at the POS (Point Of Sale). Note that the app does not send the exact card details but instead it uses a one-time security code that represents the user account information (Popper, 2015).

Advantages:

- Fast service.
- Secure and reliable.
- Automatic scansion of the card details by mobile camera.
- Unlimited payment amount, although some merchant applies the limit of £30 (Revolut, 2020).
- Available on Android devices running Lollipop 5.0 (released on June 2014) or higher.
- Does not need internet connection from the user point of view.

Disadvantages:

- Limited by the type of card issued by the bank.

- Restricted to Android OS and therefore not available on iOS devices. Considered as disadvantage because from a software development point of view it could mean two different implementations.
- Requires the store or retailer to have a specialised NFC reader that is not ideal in places like street food markets.

2.2.2 Apple Pay (iPhone Wallet)

Apple Pay is also a payment system that has a mobile application as a digital wallet that can be used for store payments. Fundamentally is the same concept of Google Pay. It was initially created for online payments only and later developed the feature of HCE to work at the POS.

Advantages:

- Fast service.
- Secure and reliable.
- Automatic scansion of the card details by mobile camera.
- Available on the Apple devices from iPhone 6 (released on September 2014) onwards and Apple Watches. Some iPads can have the software application but they are unable to process the NFC in-store payment, hence it used for online payments only (Hill, 2020).
- Payment amount is usually unlimited but that depends on the country and the merchant (Apple, 2020)
- Does not need internet connection from the user point of view.

Disadvantages:

- Limited by the type of card issued by the bank.
- Restricted to iOS and therefore not available on Android OS devices. Considered as disadvantage because from a software development point of view it could mean two different implementations.
- Requires the store or retailer to have a specialised NFC reader that can be is not ideal in places like street food markets.

2.2.3 Other Digital Wallets and Payment Solutions

There are many other digital wallets like Google Pay and Apple Pay currently having the NFC payment method or QR loyalty system. Therefore, those company solutions are just going to be briefly mentioned as a proof of the increased demand and popularity of this technology. It is also to prove the demand of loyalty schemes in the market.

2.2.3.1 Samsung Pay

Like Google Pay concepts, but with the difference that is enabled on Samsung devices only. It adds the ability of the MST (Magnetic Secure Transmission) that other e-wallets do not have (Whitwam, 2020). Released on August 2015 and available on most devices from Galaxy Note 5 onwards (Savvides & Orellana, 2019).

2.2.3.2 Microsoft Pay

Identical models as the previously mentioned e-wallets. The only difference is that is designed for Windows OS and the feature on mobiles has been withdrawn on 29th February 2019 (Thorp-Lancaster, 2019).

2.2.3.3 WeChat Pay (service inside WeChat)

It is similar to the previous payments systems but with the difference that the NFC capability is not popular in the stores because of the Chinese market. It is, in fact, used occasionally in the Chinese underground turnstiles (Borak, 2019). It is far more common the use of the QR (Quick Response) code as quick payment identification between users. The store does not need a POS anymore but instead it uses a smartphone or sometimes just a printed QR code. The scenario in a usual store transaction between customer and retailer are as follows:

1. Customer scans the retailer QR code.
2. Customer enters the money amount.
3. Customer validate transaction by either code or fingerprint
4. The app sends instantly the credit to the retailer.

Otherwise another option is:

1. Retailer scans customer QR code.
2. Retailer requests a payment to be sent via chat to the customer
3. Customer confirm transaction by code or fingerprint.

This solution is incorporated within the WeChat app that is available in all devices but Windows Phone (discontinued in 2016) and it also allows in-app store and web payments for loyalty schemes (Henriot, 2019). This app is popular among Chinese customers only (Borak, 2019), therefore out of the intended market of this project.

2.2.3.4 PayPal Wallet

Close to the aforementioned ideas, PayPal have developed this application to allow PayPal customers to pay using the QR code in the stores. This implementation released on May 2020 allows similar payment method as WeChat Pay, regarding the NFC aspect they rely on a Google Pay partnership (Smith, 2020).

2.2.3.5 Alipay Wallet

This company service needed to be commented due to the popularity and the high usage around the world. Alipay is considered to be the leading mobile payment platform but it does not have any NFC capability because solely based upon the QR code system (Heggestuen, 2014). It has a loyalty programme, but it works only with payments within the app (Basu, 2018).

2.2.3.6 Loopy Loyalty - PassKit

Loopy Loyalty is a web-based application part of the PassKit company solutions. It allows retailers to create a customised loyalty scheme by adding own brand images and colour palette.

The service offered works as loyalty card to be stored in Apple Wallet or Google Pay. An example of user journey case is:

1. The customer is registered in the loyalty scheme by email prior to the payment process.
2. The customer buys a product and therefore satisfies the stamp collection criteria (to be decided by the retailer).
3. The customer shows the QR code on its digital wallet that represents his details.
4. The retailer scans within the Loopy Loyalty Stamper mobile application the customer QR code.
5. The retailer has the option of choosing the number of stamps to give and redeem rewards if available (see Figure 11).
6. After the retailer has confirmed the options selected, the customer will have the loyalty card details updated.

Figure 11. Options for stamp record on Loopy Loyalty Stamper app

Although this system is very efficient, in practical terms, it is hard to use. The retailer would need to have his smartphone available all the time after a transaction is successfully completed. He would also need some time to interact on the smartphone to decide all the options for the customer. It increases the responsibility for the cashier and the time process for each customer.

2.2.4 Embargo

Embargo is a mobile application where there are rewards scheme for people who often go to social places such as pubs and restaurants. The app is also available in various food market sites and it just requires the user to register into their service with no bank details.

The app gives rewards based upon the number of visits the customer accrue over time in the same place, like a classic loyalty scheme. A certain number of visits allow the customer to redeem rewards at the venue. Some special discounts are also included depending on the specific place.

As shown below (see Figure 12), the app has a very simple interface with few buttons and icons that result in a simple user experience which is very important for this type of service.

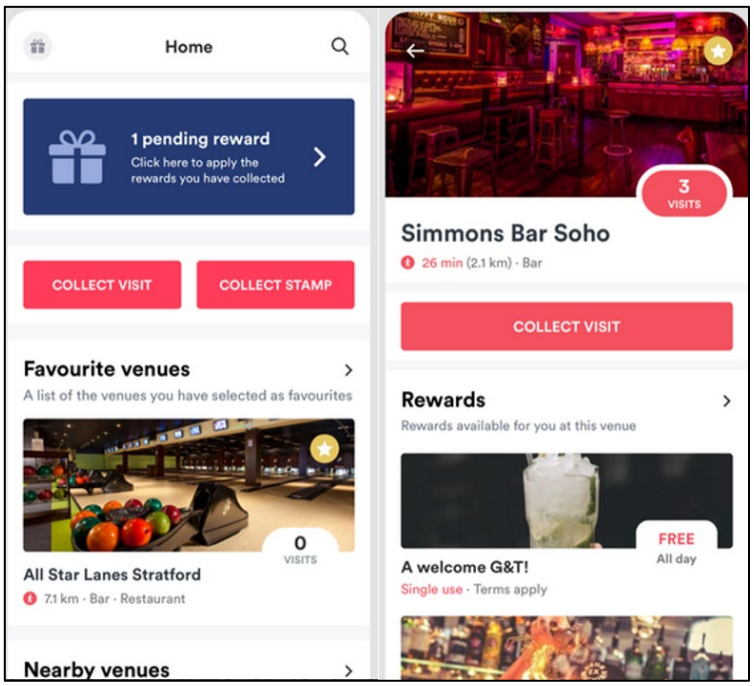


Figure 12. Embargo App screenshot example from the Google Play Store

This case is different to the previous because there is not either NFC or QR code involved. Instead, there is beacon technology that requires the use of Bluetooth connection. To locate better the customer position in relation to the venue the app also requires GPS (Global Positioning System) to be enabled (Embargo Lifestyle Limited, 2020).

Advantages:

- No need of user personal details such as bank cards.
- It works similar to the NFC, but it uses Bluetooth, GPS and Wi-Fi.
- Complete user journey to bring more customer to the venues.
- There is an ability to automatically connect to the venue Wi-Fi network.

Disadvantages:

- The user must have Bluetooth, GPS and Wi-Fi enabled in order to work.
- Relies on a good internet connection environment which can be troublesome in a very crowded place.

2.2.4.1 Beacon Technology

The beacon technology is the closest alternative to the RFID and NFC use case scenarios. It uses the consolidated Bluetooth technology to send signals in the nearby in the form of UHF (Ultra High Frequency) radio waves around 2.45 GHz (Haines, 2020). Like the RFID active mode (see 2.1.2) it requires both ends to have their own power source but it can send more data, like in the NFC(see 2.1.3). It is designed to work in a short or medium range around 10 to 30 meters although it has the capability to reach 300 meters. The drawback is in terms of cost and availability because it is very hard to get one, and the cheapest transmitter, a keychain beacon, costs around £20. The price rises to around £40 for longer ranges (Ratna, 2020) which makes the solution more limited depending on the company's budget. This was the major reason to not choose this technology for this project implementation.

2.2.5 Jisp

Jisp is a company that provides a mobile application where you can buy products from selected retailers by using only the NFC feature in the smartphone. The user must register with a payment system like a bank card in the app to start purchasing products. A possible use case can be described as:

1. The customer is registered to the app and has the payment method set up.
2. The customer goes to the shop that have special tags for the products on the shelves.
3. Whenever the customer decides to pick a product, he scans the tag of the product to add it in the basket.
4. Once the customer has finished the choose the products, he goes to a checkout.
5. The checkout is another NFC specialised tag that completes the purchase by making the payment with the card details previously stored.

Advantages:

- Built around the NFC but it also can scan QR codes.
- User can use the app also to find information related to the product scanned.
- Can monitor user behaviour to customise what they may want.
- Gamification service to engage with the user such as quizzes.
- Does not require the user to register for using some app features such as scanning product details.

Disadvantages:

- Requires an account with debit card set up.
- It is necessary to have a customised shop with their service and technology, therefore the availability in store is limited.

2.2.6 Branded loyalty apps

This section covers an overview of the existing loyalty mobile application available from different brands such as Costa Coffee and Tesco. It is focused on Caffè Nero as main example because is the most similar to the intended development.

The difference between the following services is that they are developed by the retailer itself. They do not work as medium between retailer and customer; they are the direct seller of the product and the app user can only be the customer.

2.2.6.1 Caffè Nero App

The coffee house company based in London have their own loyalty scheme reward system on a mobile application as well as the paper card version (see Figure 1).

The paper stamp card is similar to the ones provided by other popular brands. The customer buys a coffee and earns a stamp as a form of loyalty point. After collecting ten stamps, the customer is rewarded with a free coffee. Usually it is the cashier that marks the stamp on the card.

The other digital version provides the same service but on the smartphone. Instead of the cashier stamping on the card, he scans the QR code that the customer shows to him. It is possible to ask the cashier to transfer the existing stamp from the paper card to the app (Caffè Nero, 2019).

This product has the same results of the goal of this project. The difference is in the technology used. This project has the intent to use NFC and avoid the need for the customer to download an app.

Advantages:

- Flexibility of choice for the customer between paper or digital card
- Does not require and bank card associated with it although it is possible to add it, so the customer can earn the loyalty stamp and pay at the same time (Caffè Nero, 2019).
- The voucher awarded is kept separately, so the user can continue collect new stamps.
- It can be added to the Apple Wallet or Google Pay to collect stamps.
- It only requires an email address and a password to register.
- Does not require the user to be connected over the internet to collect the stamp.

Disadvantages:

- Must be scanned prior to the purchase.
- Can be used only within Caffè Nero shops.

2.2.6.2 Others

To avoid redundancy of the content, a table of other popular loyalty mobile apps that are available in the UK territory is listed below:

Company App	NFC	QR Code or Barcode	Stamp or points criteria	Bank Card
Tesco Pay +	No	Yes	Amount spent	Required
Tesco Clubcard	No	Yes	Amount spent	Not required
Nectar	No	Yes	Amount spent	Not required
Gregg's Rewards	No	Yes	Amount spent or product	Available
Costa Coffee Club	No	Yes	Amount spent	Available
Starbucks UK	No	Yes	Number of times the app is used for payments	Available

Figure 13.Comparison of company mobile loyalty apps features and loyalty scheme details

From this overview (see Figure 13), it is clear that the NFC technology is still not used for loyalty schemes. In the Tesco Pay + case, the loyalty point collection is incorporated within the payment solution so the customer does not need to do additional scanning or verification because it is automatically done within the company services (Tesco, 2020).

Gregg's Rewards and Costa Coffee Club are relatively similar to the Caffè Nero scenario (see 2.2.6.1) while Starbucks is different. The latter rewards the customer if paying using a pre-loaded Starbucks Card that can be added into the app.

2.3 Review of Tools and Techniques

There is a variety of tools to work with NFC. In this section an analysis is carried on to give a list of possible approach for the intended solution.

2.3.1 Mobile Native Android Application

A first popular solution for a software with NFC is to make a mobile application. It is the most flexible because it allows people to use the full potential of the technology.

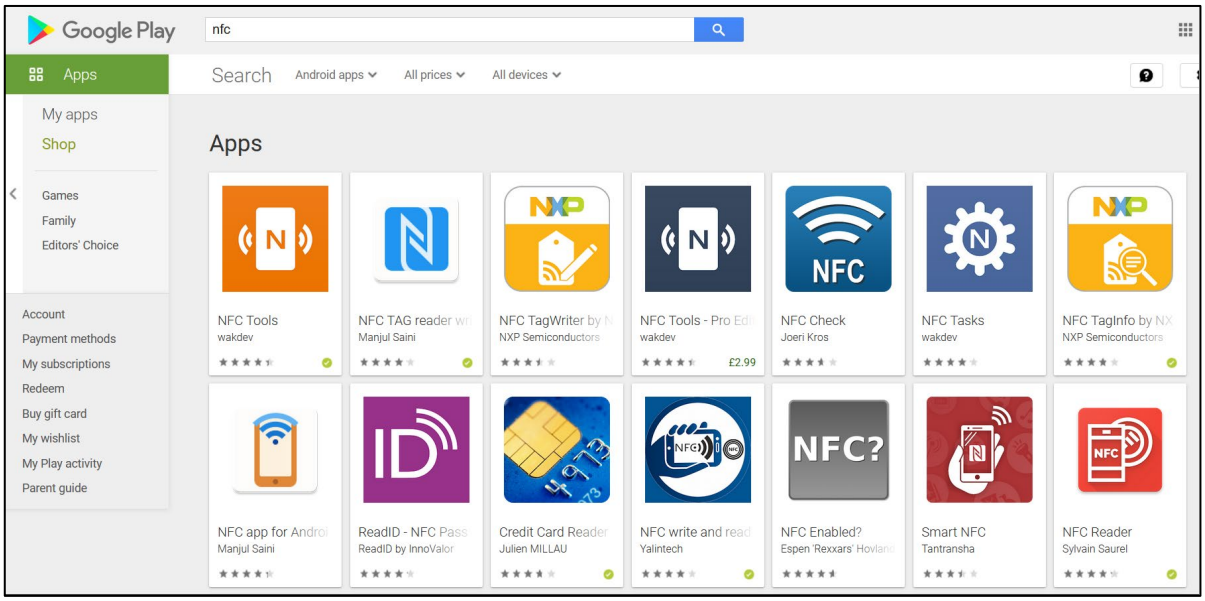


Figure 14. NFC application search result on Google Play (Search made on July 2020)

There are plenty of applications that can enable the user to interact with NFC (see Figure 14). Also, even if the user did not have any application installed, it can automatically respond to the data inside the tags because of the Android OS. For example, if a user has saved a URL link into a tag. Whoever scans it will open the default browser and go to the website saved (Frew, 2020).

Advantages:

- Full capability of the NFC technology. Includes HCE used for payments and peer-to-peer communication.
- Limited only by the specification of the device.
- The operating system must approve the app, so it assures quality, security, and device compatibility.
- The app can receive support from the app stores to help distribute the product.
- Can work offline without internet connection.

Disadvantages:

- Slow to develop and very hard to update. It would require constant updates from the customer even for small changes.
- Difficult to maintain once there are many different devices using it on different versions.
- Cannot work on other OS devices.

2.3.1.1 Mobile Native iOS Application

Similar to the previous technique (see 2.3.1), iOS applications are subject to the same features and problems. Unlike Android, the possibility to read and write from NFC tags on iOS has been introduced only in the Autumn of 2019 even though the capability was already in the devices long ago (BlueBite, 2020).

	Has NFC	Card Emulation	NFC Payments	Reads NFC	Writes NFC
iPhone 11, 11 Pro / Max, SE (2nd Gen)	✓	✓	✓	✓	✓
iPhone XS, XS Max, XR	✓	✓	✓	✓	✓
iPhone X, 8, 8+, 7, 7+	✓	✓	✓	✓*	✓
iPhone 6, 6+, 6S, 6S+, SE (1st Gen)	✓	✓	✓	✗	✗
iPhone 5S, 5C, 5, 4S, 4, 3GS, 3G	✗	✗	✗	✗	✗

Figure 15. NFC features enabled on iOS smartphones

To avoid repetitiveness regarding the advantages and disadvantage, please refer to the list in *2.3.1 Mobile Native Android Application*.

2.3.2 Web Application

Thanks to the W3C community Group there is a capability to read and write NDEF messages on tags through Google Chrome browser on mobile (Kostiainen, 2019). This gives the opportunity to design a Web App with NFC features to enhance the UX like never before.

A Web App is what is commonly known as a Website that uses improved back-end capabilities to perform specific task. It is a software that is made to be accessible by any web browser.

Advantages:

- Easy maintenance.
- Cross platform among all the smartphones.

- Updates do not require any new installation or action from the user.
- Low cost for the implementation and deployment on the internet.
- The data stored are not on the user device and therefore no data limitation.

Disadvantages:

- Harder to discover because not listed in a marketplace.
- Slower and less responsive than the native apps.
- Relying on the internet connection.

2.3.3 Hybrid Application

These types of mobile applications are considered to be something between a Web App and a Native Mobile Application. They are installed on a mobile device like an app, but in order to work they need to surf the internet like a Web App.

Advantages:

- Require only one codebase across all mobile platforms.
- Easier to update depending when the changes are not related to the native code.
- Updates do not require any new installation or action from the user.
- Combines the best from the user experience and the agile development cycle.

Disadvantages:

- Compared to the native application they are less performant.
- Higher is the customisation needed and higher than the native application is the difficulty of implementation.
- Relying on the internet connection.

2.3.4 .Net Core 3.1

.NET Core 3.1 is currently the latest version of the ASP.NET CORE Microsoft frameworks. It is a web development framework for building Web Applications, services and console application. It is open-source and based on .NET platform (Microsoft, 2020).

Advantages:

- Works on multiple platforms such as Windows, Linux and Mac.

- It is fast and scalable to work the modern libraries and programming languages.
- The architecture of the system is modular. It gives the possibility to modify a component without affecting the rest of the system.
- Uses C# as main programming language.

Disadvantages:

- It has many features not yet fully developed, although it will be updated over and over.
- It has memory leaks that increase the quantity of data not used.

2.3.5 .Net Framework

.NET Framework is the framework solution created by Microsoft before .NET Core for the same purpose. It is used to develop Web Applications, service and console applications on Windows.

Advantages:

- Has more functionalities than the other new framework although there should be less over time.
- The longer presence in the market means a higher number of solutions and documentation among the users.
- Uses C# as main programming language.

Disadvantages:

- It is not open source.
- Development restricted on Windows platform.

2.3.6 Entity Framework

Entity Framework is an open-source Object-relational Mapping framework for .NET applications supported by Microsoft. It gives the developers the option to work with the database through .NET objects in the software with more flexibility.

Advantages:

- It is developed as an Open Source product.
- It has stored procedures support.
- It provides auto generated code that simplifies the development time and cost.

Disadvantages:

- It handles data in a different way from SQL.
- Strictly linked to the database schema. It requires constant updates to match the tables in the source solution.

2.3.7 QR Code Generation

This type of approach for the intended project development is possible. In fact, there is a possibility of using QR code generation for the representing the loyalty card with a unique ID in the same way as the previously mentioned applications (see Review of Projects / Applications2.2) but this project also aims to demonstrate the capabilities of the NFC and what alternatives it can present with the latest technologies.

Advantages:

- Versatility of being used for different purposes.
- Very-low cost to use and easy to implement.

Disadvantages:

- Must be scanned through a device that has a camera and understand what data is the QR code representing.
- The camera quality must be enough to clearly capture the image.
- The data represented by the code can only hold up to 3Kb.

3. Requirements

This section presents a review of the different stages of the software development process. The initial step is the identification of the project goal. Based on the goal, it is possible to provide a list of function and non-functional requirements that the system must have. This process is known as Requirements Engineering or Gathering.

The following sections provide a presentation of the stakeholders, the project promises and the issues related to the legal and ethical aspects.

3.1 Stakeholders

The stakeholder is defined as a person who is involved with an organization and has responsibilities and interest its success (Cambridge Dictionary, 2020).

Creating an onion model is considered a good approach to understand the stakeholders structure of the project (Alexander & Beus-Dukic, 2009).

Figure 16. Basic Onion Model stakeholders

In the figure above (see Figure 16), the inner circle there is The Product which is the Web Application meant to be created (i.e. Contactless Loyalty).

The System layer contains the essential part of the product and service, the people who interact directly with it. Therefore, there is the cashier, who will allow the customer to scan the tag through the Web App, so that the customer will receive the loyalty point (i.e. Stamp on the card). The technical team will be the people maintaining the service up and running smoothly. They will deal with the customer or cashier in case of product issues.

In the Containing System there are the brand or retailer owners that are the people who decide to use this service to be applied in their shops. The Item provider is whoever provides the final rewarded product to the customer (e.g. A free coffee, a free meal or a discount). For example, if the brand or retailer is a supermarket (e.g. Tesco) they can reward customers with different items in the shop (e.g. Chocolate bars) but they are not the original item provider, meaning they do not create the item. Although depending on the scenario, they could be the same stakeholder. If the retailer owner is a small shop like a coffee stand in a street food market, they will automatically be also the item provider. Ultimately, this layer contains stakeholders that may not interact with the product directly, but they get advantage from it like brand or product awareness.

In the last layer, Wider Environment, there are financial beneficiaries such as stockholders and the public in case of a public impact and similar entities. Moreover, the providers of the Web NFC API, Microsoft Azure for cloud services and NFC tag suppliers are considered to be the technology suppliers who are not directly involved with the product but are still related to it as an intermediate. There is also a consideration for possible hackers or malicious customers that intend to steal and illegally take advantage through this service.

3.2 Gathering Requirements

To gain a more knowledge on the demand of this type of project, a survey has been carried out along with interviews and statistical analysis.

3.2.1 Statistical Analysis

The project basic requirements have been decided using online resources and documentation about loyalty card schemes.

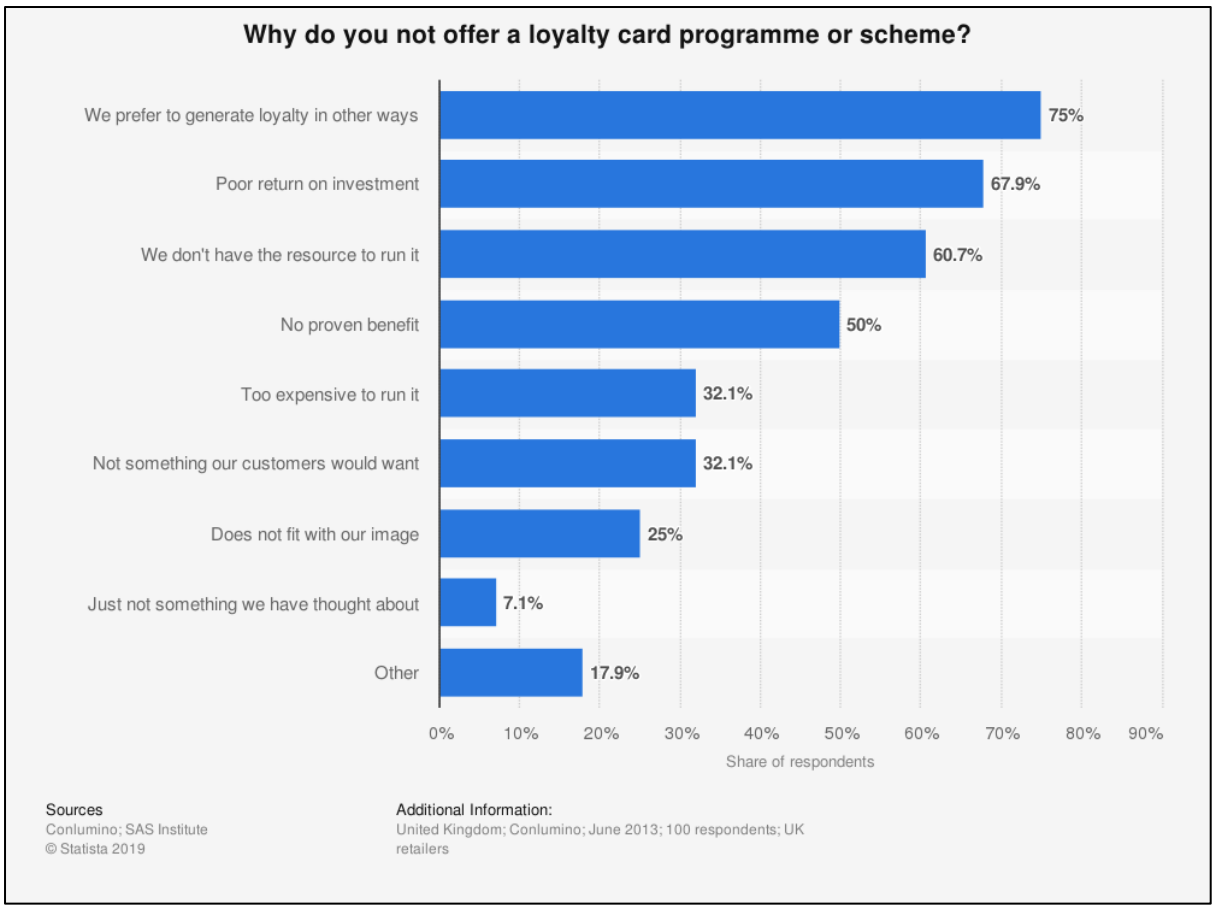


Figure 17. Classification of reasons for not offering a loyalty scheme

The data shown above (Figure 17) shows that the top three drawbacks for offering a loyalty scheme to the customers from the point of view of UK retailers are:

1. Not interested in the card programme.
2. Poor return on investment.
3. Not appropriate resource to run the scheme.

This project can tackle the second and third problem. It provides a very low-cost technology, as well as a service that can be included with a membership cost to charge while the scheme is active.

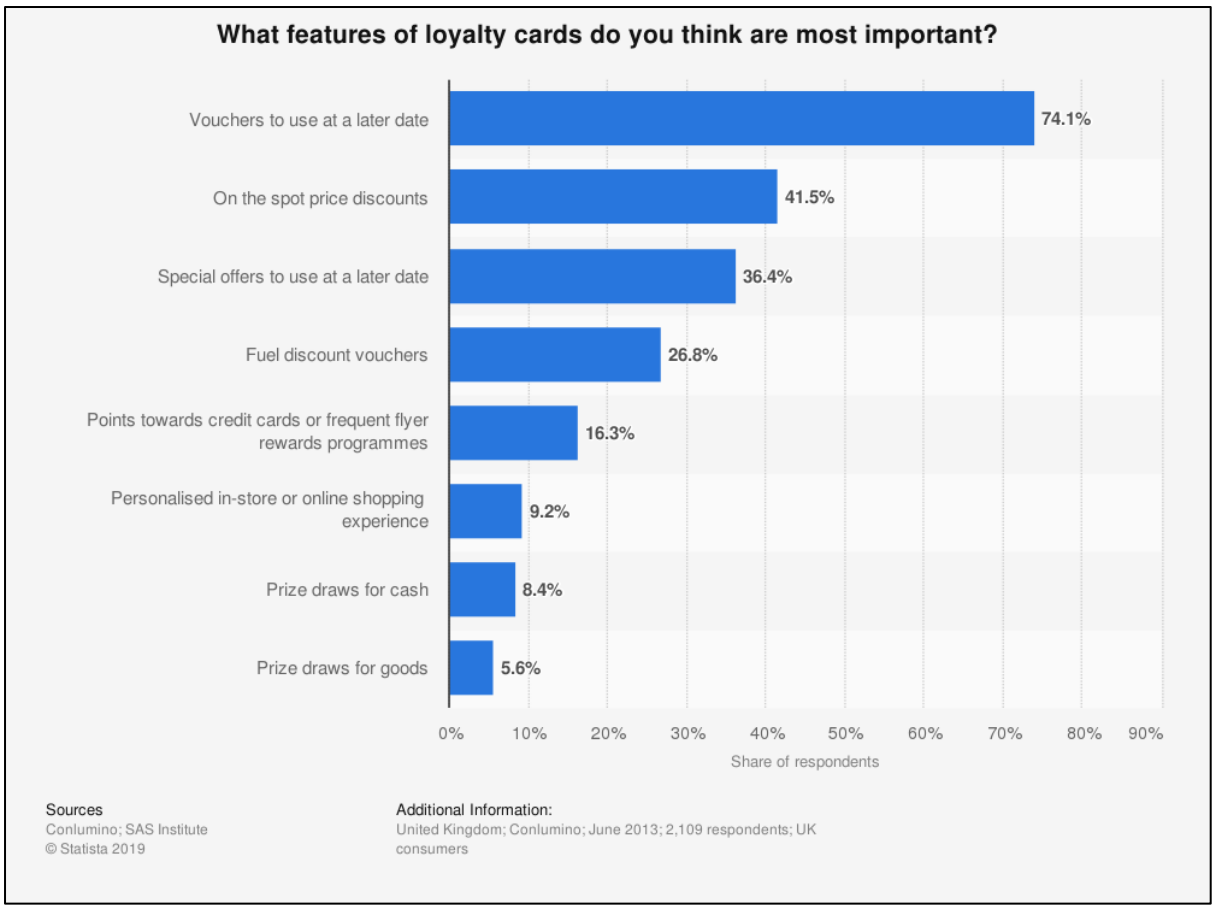


Figure 18. The most important features of a loyalty card for consumers

The data shown above (see Figure 18) provides a classification of the most valued features for the UK consumers. It is evident that the most important factor is the ability to use vouchers at a later date.

Although the software for the voucher rewards has not been implemented in this project, it is included with the involvement of a digital voucher company (i.e. i-movo). The back-end of the Web App can validate the reward when collecting points and tracking when to reward the customer. At the right moment, the Web App will allow the user to decide to receive the digital voucher by SMS on their phone from i-movo through an API. The external company provides a secure voucher system that sends out different type of vouchers where different parameters, such as voucher expiry date, can be customised depending on the retailer or client needs.

3.2.2 Survey Analysis

A survey has been made online through Google Forms with 88 anonymous participants. The answers have been analysed with market research criteria.

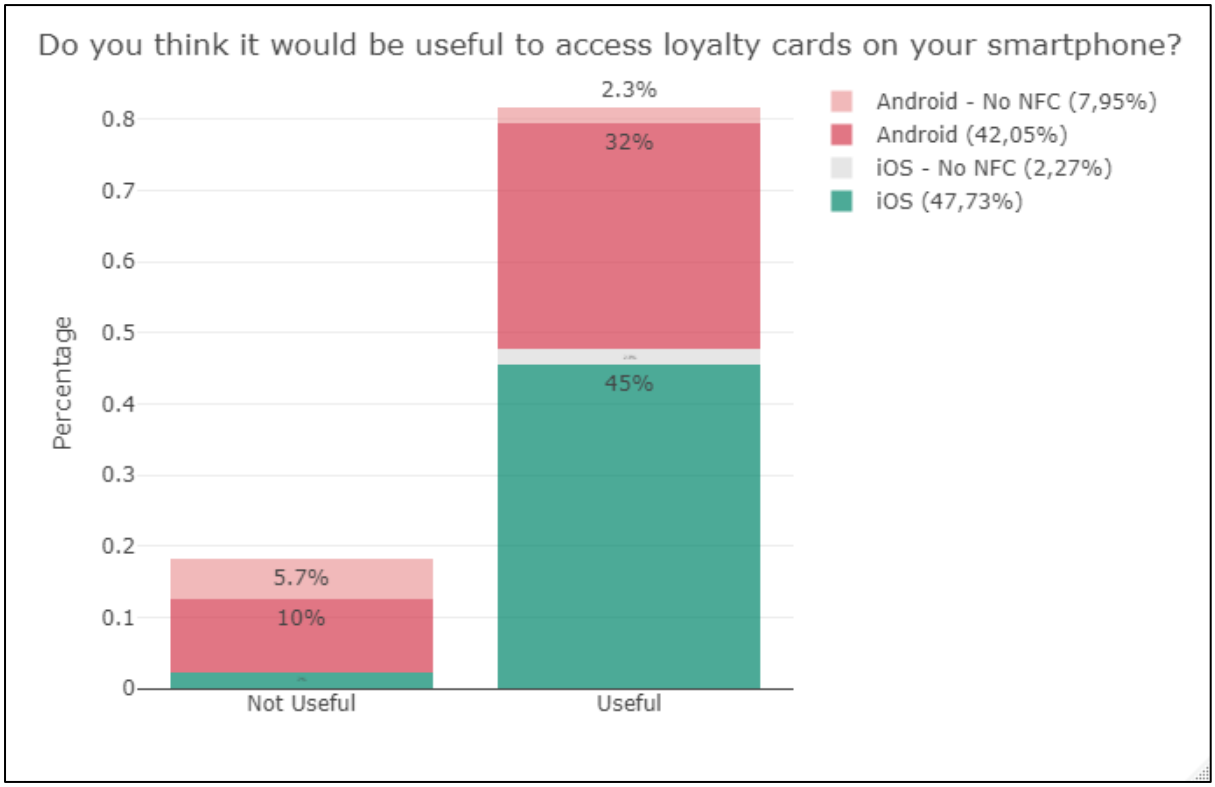


Figure 19. Usefulness to access loyalty cards on the smartphone

This chart (see Figure 19) shows the partial percentages for both Android and iOS (the only two OSs available from the participants) so that the sum of each modality is the total percentage for that modality. The variables define whether the user thinks it is useful to have access to the loyalty cards on the smartphone or not.

It is possible to see how the number of users that deemed such technology not useful is sensibly higher for Android than iOS users, and vice-versa. In addition, every iOS user that does not have access to NFC functionality on their iPhone thinks that such service would be useful, unlike Non-NFC Android users. This is to show that making available the Web-NFC feature on iOS would be more beneficial than on Android in the market.

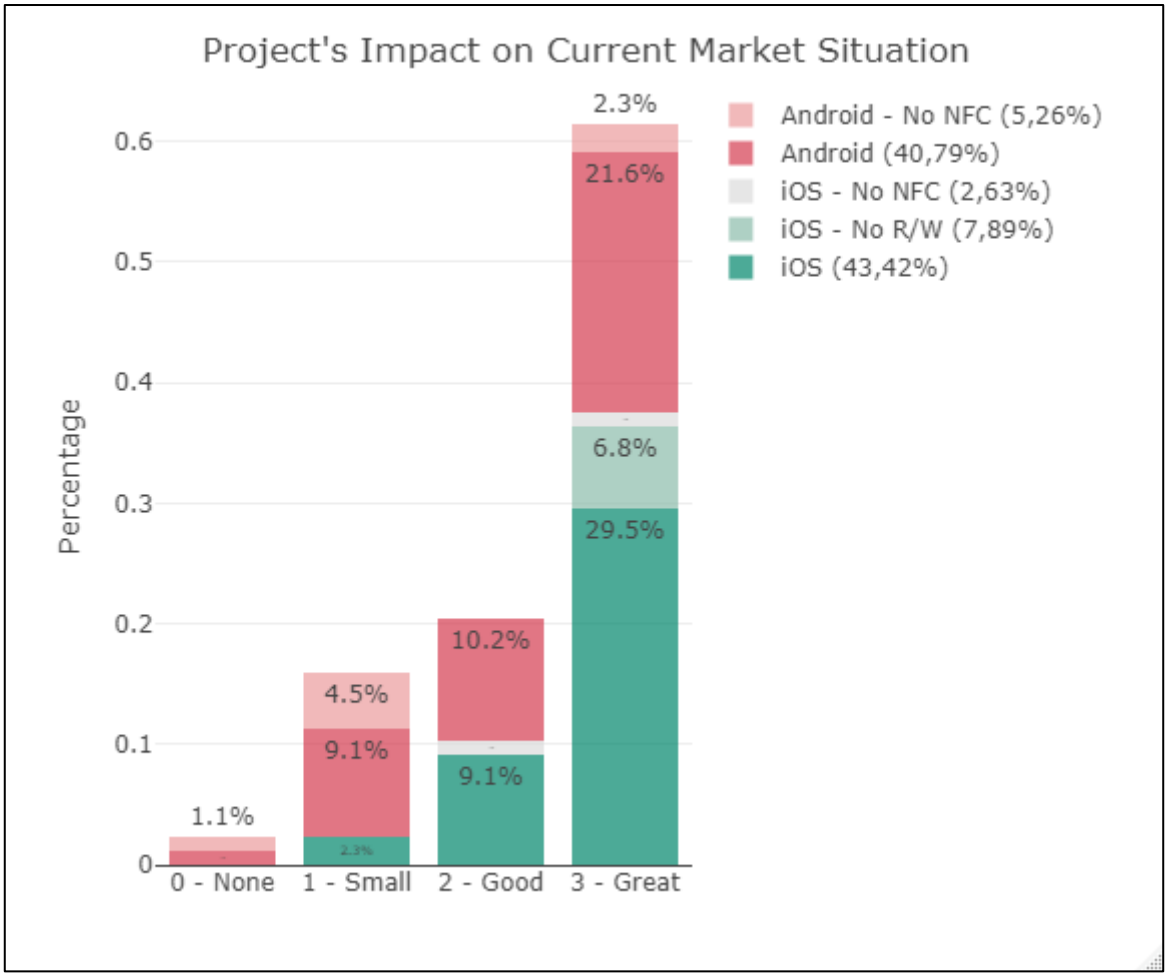


Figure 20. Project Possible Impact on the Current Market Situation

This chart shows partial percentages for both OSs, so that the sum of each modality is the total percentage for that modality. For Android, the modalities are with or without NFC. The same modalities applies for iOS but with the addition of smartphones that have "No read and write functionality for NFC" (see the models in Figure 15) which are the features used in this project.

It is possible to see how the 61.3% of the participants had a high interest in the project. Furthermore, iOS respondents without NFC or who cannot benefit from it selected the highest answers. This also to support the previous affirmation stating that enabling the feature on iOS smartphone would benefit the NFC popularity in the market.

3.2.3 Interviews and Feedbacks

Further feedbacks and comments have been taken from interviews with the stakeholders such as potential customers and loyalty scheme providers.

An important aspect to be taken into consideration was to engage with the customer. The aim for the retailer is usually to increase the brand awareness in the market. If the Web App is a platform that involves many people, it will automatically become a place to be take into consideration for marketing strategies.

During a meeting with the COO of the company i-movo, a method for the voucher delivery was chosen. It was decided that the voucher would have been sent by SMS because of the modest amount of effort required. Since the development of this project was in the early stage, the easiest approach to complete a full user journey was by using that method.

3.3 Modelling requirements and relevant diagrams

In the following sections the description and diagrams relevant to this scenario are provided to support the software decisions.

3.3.1 Use Case Model

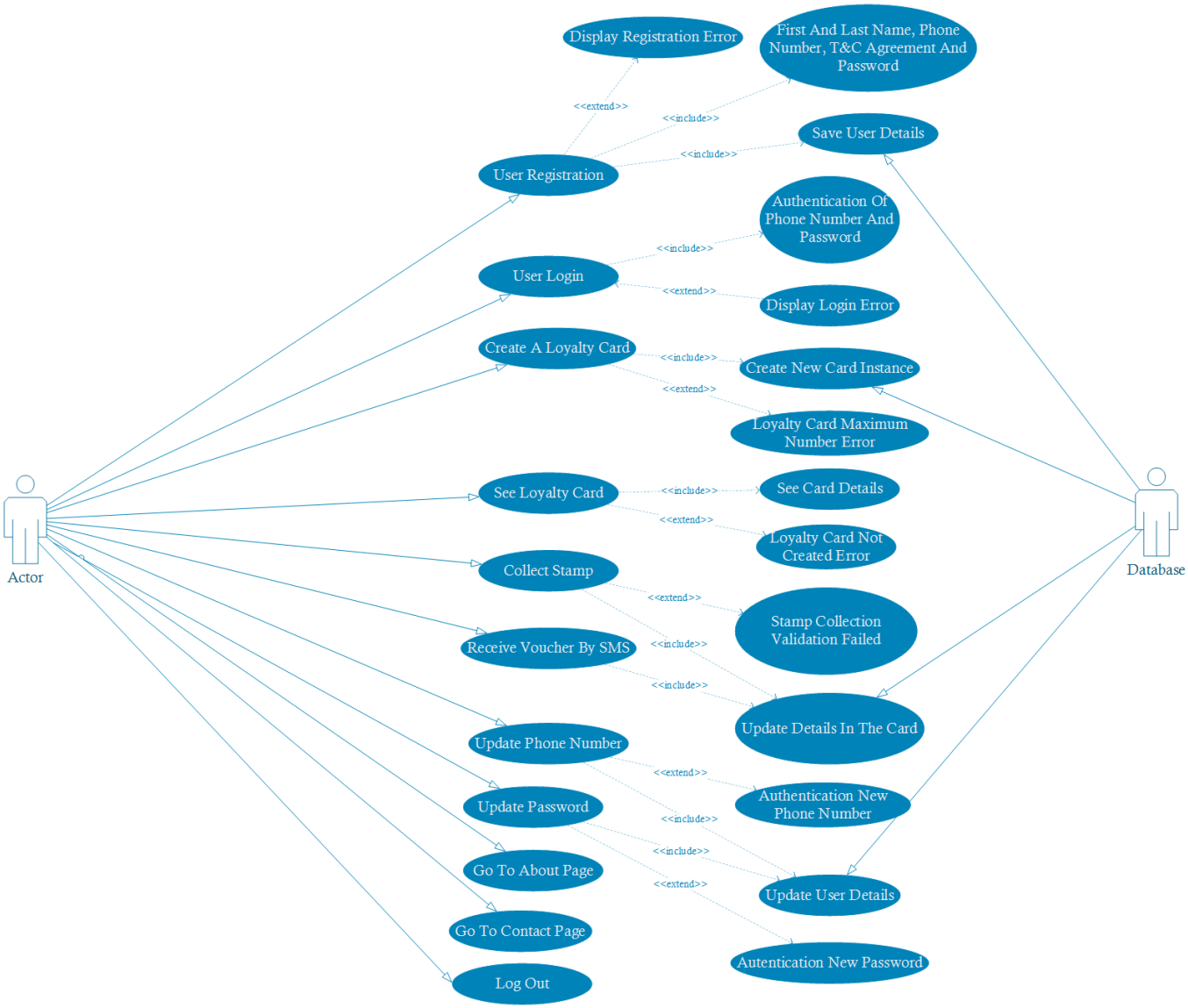


Figure 21. Use Case Diagram

The above diagram shows all the possible actions that a user can do in the Web App.

3.3.2 Use Case Description

Use Case ID	UC0
Use Case Name	Registration
Description	This use case is for when a user attempts to register on the Web App by providing some details such as First and Last Name, Password Mobile Phone Number and accepting the Terms and Conditions.
Primary Actor	Customer User
Pre-Conditions	Web App shall be up and running and the user must have clicked the Register button.
Post Conditions	The system validates the input data and the details are stored in the database. The user is automatically redirected to the internal homepage after successful registration.
Main Success Scenario	1. The user clicks on the Register Button. 2. The user enters all the details (First Name, Last Name, Mobile Phone Number, Password, Confirm Password and checks the box for agreeing the Terms and Conditions). 3. The system performs a data validation. 4. The system updates the database with the new information. 5. The system automatically logs in the user with the information just saved. 6. The system shows the internal home page.
Exceptions	• The user enters a mobile phone already saved in the database. • The password is shorter than 6 characters. • The password does not contain uppercase. • The password does not contain lowercase. • The password does not contain a digit. • The password does not contain a special character. • The password and confirm password do not match. • The phone number entered is not UK format. • The user has not accepted the Terms and Conditions.

Table 1.Use Case Description (UC0)

Use Case ID	UC1
Use Case Name	Login

Description	This use case is for when a user attempts to sign into the Web App with a Mobile Phone Number and Password.
Primary Actor	Customer User
Pre-Conditions	• Web App shall be up and running. • User must have previously registered in the Web App. • User must be in the login page.
Post Conditions	• The system validates the input data with the details stored in the database. • The user is logged in and can choose all the internal features.
Main Success Scenario	1. The user clicks on the Sign In Button. 2. The user enters Mobile Phone Number and Password. 3. The system performs a data validation. 4. The system shows the internal home page.
Exceptions	• The user enters a Phone Number not saved in the database. • The Password and Password in the database do not match. • The user has not entered any data.

Table 2. Use Case Description (UC1)

Use Case ID	UC2
Use Case Name	Create New Loyalty Card
Description	This use case is for when a user clicks on Create New Loyalty Card button.
Primary Actor	Customer User
Pre-Conditions	• Web App shall be up and running. • User must be logged in. • User has pressed the Create New Loyalty Card button.
Post Conditions	• The system generates default details for a new card. • The user can see the new card on the Card page.
Main Success Scenario	1. The user clicks on the Create New Loyalty Card button. 2. The system generates a default loyalty card. 3. The system updates the database with the new card details. 4. The system shows the card details page.

Exceptions	• The user has already an existing card. Currently the limit is set to be one card per customer.

Table 3. Use Case Description (UC2)

Use Case ID	UC3
Use Case Name	See Loyalty Card
Description	This use case is for when a user clicks on See Your Loyalty Card button.
Primary Actor	Customer User
Pre-Conditions	• Web App shall be up and running. • User must be logged in. • User has pressed the See Your Loyalty Card button.
Post Conditions	• The system shows the customer card details such as loyalty points.
Main Success Scenario	1. The user clicks on the See Your Loyalty Card button. 2. The system retrieves the card data linked to the user. 3. The system shows the details of card in a new page.
Exceptions	• The user does not have a card created yet.

Table 4. Use Case Description (UC3)

Use Case ID	UC4
Use Case Name	Collect a Stamp by Scanning
Description	This use case is for when a user clicks on the Start Scanning button.
Primary Actor	Customer User

Pre-Conditions	• Web App shall be up and running. • User must be logged in. • User must have pressed the See Your Loyalty Card button. • User clicks on Start Scanning button. • User must have accomplished the condition to receive the NFC tag or Smart Card to scan from the cashier. • User must enable NFC on the smartphone. • User must allow the Web App to use NFC. • User must be on Android. • User must be on Chrome or Chromium mobile browser. • User must have a Loyalty Card registered in the database.
Post Conditions	• The system refreshes the page. • The system shows the updated data of the card.
Main Success Scenario	1. The user clicks on the Start Scanning button. 2. The user scans an appropriate NFC tag or Smart Card. 3. The system validates the scansion. 4. The system updates the record of the card in the database. 5. The system refreshes the page with the new data to display.
Exceptions	• The NFC tag or Smart Card is not correct. • The NFC tag does not have an expected scheme code. • The NFC tag Serial Number is not expected. • The user has exceeded the number of collections by either hour, day or week.

Table 5. Use Case Description (UC4)

Use Case ID	UC5
Use Case Name	Collect a Stamp by Mock Collection
Description	This use case is for when a user clicks on the Mock Collection button.
Primary Actor	Customer User
Pre-Conditions	• Web App shall be up and running. • User must be logged in. • User must have pressed the See Your Loyalty Card button. • User clicks on Mock Collection button. • User must have a Loyalty Card registered in the database.

Post Conditions	• The system refreshes the page. • The system shows the updated data of the card.
Main Success Scenario	1. The user clicks on the Mock Collection button. 2. The system validates the attempted action. 3. The system updates the record of the card in the database. 4. The system refreshes the page with the new data to display.
Exceptions	• The user has exceeded the number of collections by either hour, day or week. • The user has maliciously changed the values in the Mock Collection button.

Table 6. Use Case Description (UC5)

Use Case ID	UC6
Use Case Name	Receive voucher by SMS
Description	This use case is for when a user clicks on the Receive Voucher button.
Primary Actor	Customer User
Pre-Conditions	• Web App shall be up and running. • User must be logged in. • User must have pressed the See Your Loyalty Card button. • User must have collected enough points or stamps (by default 9) to receive the reward.
Post Conditions	• The system shows the voucher sent page. • The user receives an SMS with the digital voucher.
Main Success Scenario	1. The user clicks on the Receive Voucher button. 2. The system successfully makes the API request to send the voucher. 3. The system updates the record of the card in the database. 4. The system redirects the user to the voucher sent page.
Exceptions	• The user has not collected enough points or stamps. Hence, the button will not display. • The user has an invalid phone number saved.

Table 7. Use Case description (UC6)

Use Case ID	UC7
Use Case Name	Change Phone Number
Description	This use case is for when a user clicks on “Profile” under the Manage account menu button.
Primary Actor	Customer User
Pre-Conditions	• Web App shall be up and running. • User must be logged in. • User must be in the Profile section of the Manage Account
Post Conditions	• The system shows the updated phone number on the page. • The system shows a statement as a response.
Main Success Scenario	1. The user clicks on the Manage Account button from the menu options (Hamburger menu button). 2. The user edits the Phone Number field. 3. The system validates the new Phone Number. 4. The system updates the user record in the database.
Exceptions	• The user has not changed the phone number. • The user is changing phone number to one already existing. • The user is using an invalid phone number format.

Table 8. Use Case Description (UC7)

Use Case ID	UC8
Use Case Name	Change Password
Description	This use case is for when a user clicks on “Password” under the Manage account menu button.
Primary Actor	Customer User
Pre-Conditions	• Web App shall be up and running. • User must be logged in. • User must be in the Profile section of the Manage Account.
Post Conditions	• The system shows the updated phone number on the page. • The system shows a statement as a response.

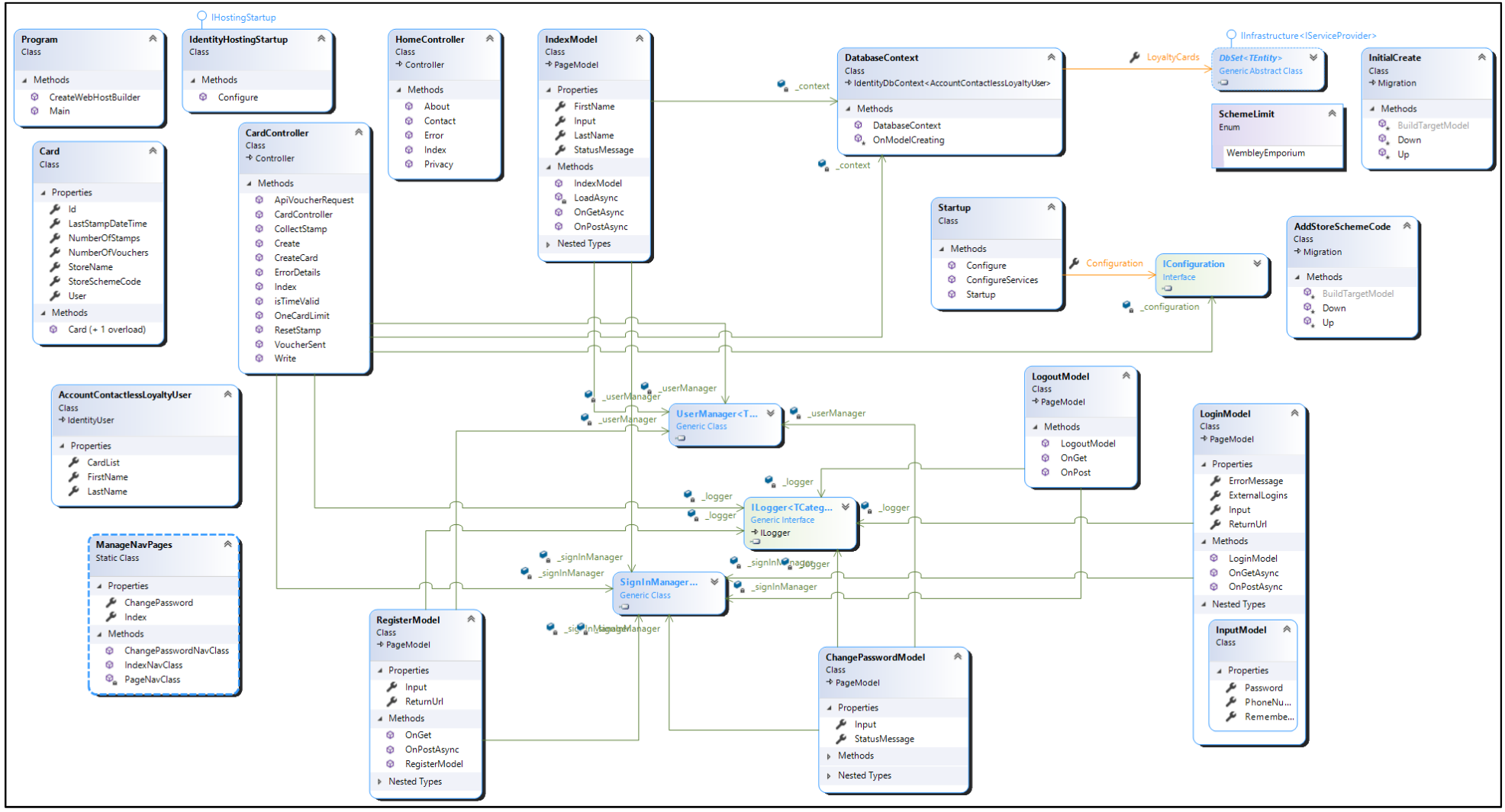
Main Success Scenario	1. The user clicks on the Manage Account button from the menu options (Hamburger menu button). 2. The user selects the Profile section. 3. The user enters Current Password and the New Password twice. 4. The system validates the New Password. 5. The system updates the user record in the database.
Exceptions	• The user has not changed the password. • The user is changing phone number to one already existing. • The user is using an invalid phone number format. • The password is shorter than 6 characters. • The password does not contain uppercase. • The password does not contain lowercase. • The password does not contain a digit. • The password does not contain a special character. • The password and confirm password do not match.

Table 9. Use Case Description (UC8)

Use Case ID	UC9
Use Case Name	Visit About or Contact page
Description	This use case is for when a user clicks on About or Contact from the menu button.
Primary Actor	Customer User
Pre-Conditions	• Web App shall be up and running. • User must be logged in.
Post Conditions	• The system shows the About or Contact page
Main Success Scenario	1. The user clicks on the About or Contact button from the menu options (Hamburger menu button). 2. The system shows the page content.
Exceptions	• The page is not found.

Table 10. Use Case Description (UC9)

3.3.3 UML (Unified Modelling Language)



b

Figure 22. Main Class Diagram of the Contactless Loyalty Web Application

3.3.4 ERD (Entity Relationship Diagram)

The following diagram (see Figure 23) shows the structure model of the Web Application. Because of the solution initially provided the database is relatively simple. The main two tables are the AspNetUsers and LoyaltyCards. The former has automatically created attributes that will be user in future stage of the software.

The AspNetUsers is identified with an ID used a Primary Key and it will also have a UserName matching the PhoneNumber, FirstName, LastName along with a PasswordHash.

The LoyaltyCards have an ID as Primary Key and the attributes: LastStampDateTime, NumberOfVouchers, NumberOfStamps, Storename, StoreSchemeCode and Foreign Key UserId associated with the AspNetUsers table.

Since the user is one and he will be able to store zero or more cards, the multiplicity of the relationship is one to many. The AspNetUsers table will have participation and cardinality 1 while LoyaltyCards will have 0 participation and many as cardinality.

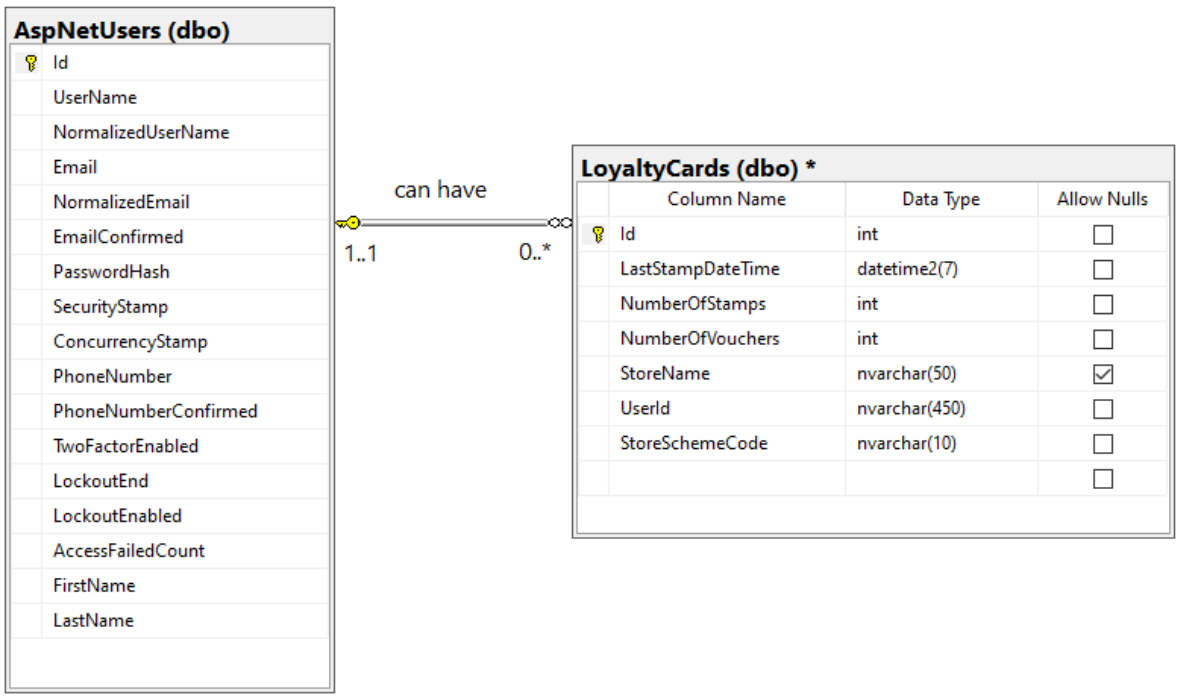


Figure 23. Database ERD

A future implementation for improved user management in the database could look like the diagram below (Figure 24) where the user can have the possibility to have a two factor authentication using a token and can hold different roles.

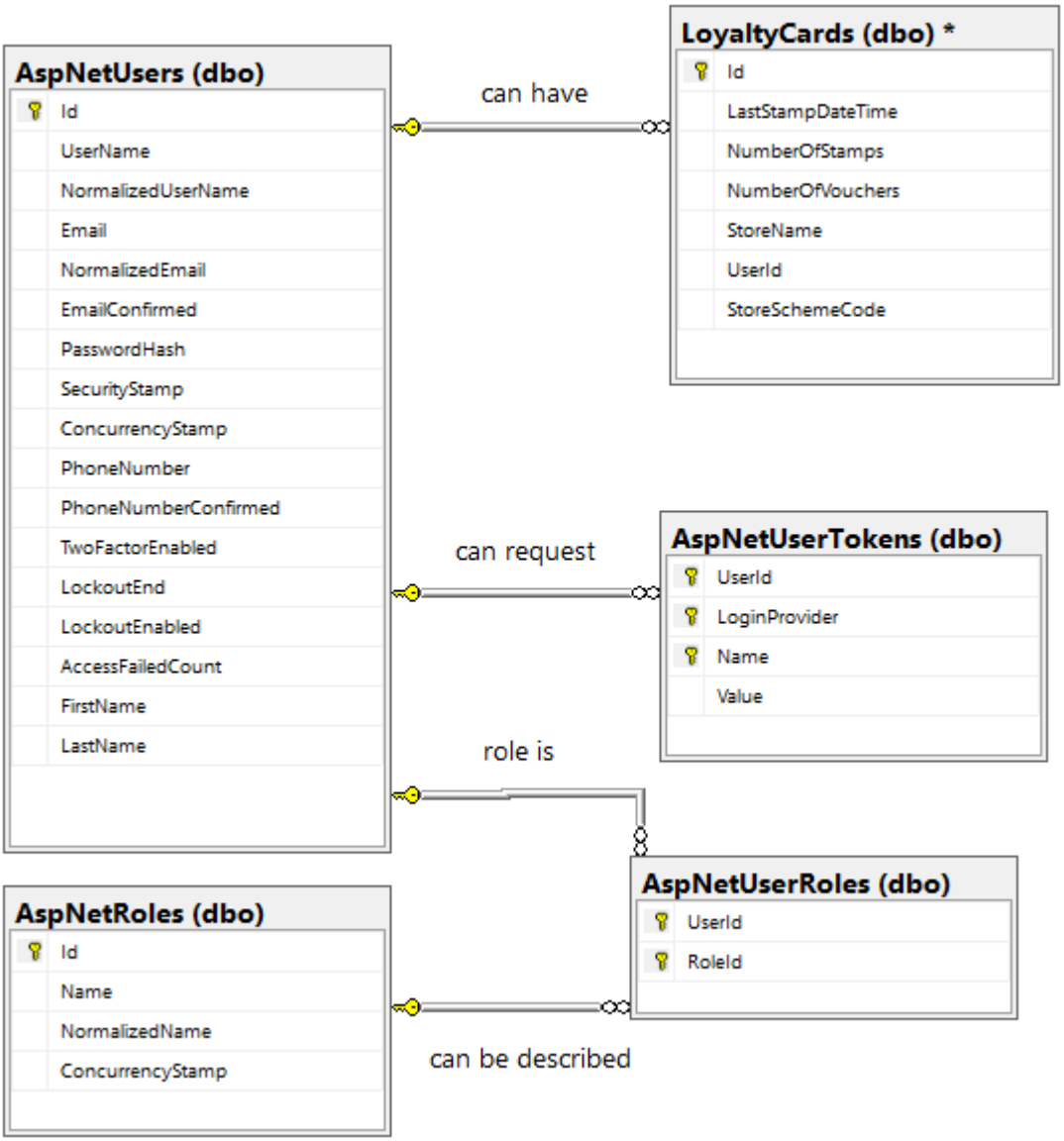


Figure 24. Possible database design with more user management

3.4 List of Project Requirements

In this section a table with the list of functional and non-functional requirements is shown.

The list is made following two systems called Simple Ranking and MoSCoW.

The former is a method that ranks the requirements importance by a number from 1 to n, where higher the number less important it is (Hatton, 2008).

The latter is a hierarchical priority method that uses four different groups (Hatton, 2008).

1. MUST have: the failure to deliver these requirements would cost the success of project.
2. SHOULD have: characteristics that would be good to have if possible.
3. COULD have: features that would be nice to have but not valuable as the second group.
4. WON'T have: also called "wish list", are the requirements that are still important but that will be implemented in a future stage.

3.4.1 Functional Requirements

The following table shows the list of requirements based on the initial feedbacks and desired functionality from the stakeholders.

ID	Description	Priority
R1	User shall be able to register in the Web App	MUST have
R2	User shall be able to login in the Web App	MUST have
R3	User should be redirected to homepage if not logged in	MUST have
R4	User shall be able to create a loyalty card	MUST have
R5	User shall be able to view the loyalty card	MUST have
R6	User shall be able to scan a NFC tag	MUST have
R7	User shall be able to see About page	SHOULD have
R8	User shall be able to see Contact page	SHOULD have
R9	User shall be able to Manage password details	SHOULD have
R10	User shall be able to Manage mobile phone number	SHOULD have
R11	User shall be able to collect a stamp or point for the loyalty card	SHOULD have
R12	User shall be able to logout from the Web App	SHOULD have
R13	User shall be able to collect all stamps and receive a voucher by SMS	COULD have
R14	User shall be able to register by email address	WON'T have
R15	User shall be able to have multiple cards	WON'T have
R16	User shall be able to collect stamps by QR code	WON'T have

Figure 25. Table of functional requirements

3.4.2 Non-Functional Requirements

The following table shows the list of qualitative requirements that do not involve the final service of the software.

ID	Description	Priority
NFR1	Web App shall display the home page for users not registered	MUST have
NFR2	Web App shall have data integrity meaning that the data is stored persistently	MUST have
NFR3	Web App shall store different details associated to different user	MUST have
NFR4	Web App shall be accessible from any smartphone	MUST have
NFR5	Web App shall be working with Android NFC	MUST have
NFR6	Web App shall be able to show the user a collection of the stamp in the UI	SHOULD have
NFR7	Web App shall be visually appealing to the customer	COULD have
NFR8	Web App shall render well on mobile devices	COULD have
NFR9	Web App shall have different login depending on customer or cashier/retailer	WON'T have
NFR10	Web App shall have different content depending on customer or cashier/retailer	WON'T have

Figure 26. Table of non-functional requirements

3.5 Legal, Social and Ethical Issues

This section takes into consideration the different issues exposed to this project.

3.5.1 GDPR (General Data Protection Regulation)

On the 14th April 2016, the European parliament ratified the GDPR to give users more data and privacy protection.

Articles 13 and 14 state that the user must have access to information where the personal data is collected, how it is processed, and the duration which the data is kept. To comply with these regulations, the registration process includes a consent field to allow Contactless Loyalty to process the data provided. This field is an agreement checkbox where the user can see how the data is used and collected.

The Article 15 (intersoft consulting, 2016) of the aforementioned law, explains the right to access the data collected. This Web App project is built to prevent unauthorised access to its data from a third-party organisation. It will give appropriate access only to authorised individuals.

The application is designed in a way that the user data information is stored only where necessary and consented (i.e. *Remember me* and *Cookie policy* options). Data entered by the user is also sanitised to prevent code injection wherever possible. The data stored in the registration process such as password is protected with HMAC-SHA256 (i.e. Hash-Based Message Authentication Code using SHA256 Algorithm), 128-bit salt, 256-bit subkey, 10000 iterations (Haggerty, 2020).

3.5.2 Intellectual Property

The intellectual property of this project belongs to the student developer and the University of Westminster. This is to protect against invention theft, design, look and brand of this product. Although not officially registered, this is a recognised piece of work from the University of Westminster.
This Web Application also includes some contents adapted from online open source designs, hence a reference to those are appropriately added in the source code.

3.5.3 Environmental

As previously mentioned (see 1.1), this project also aims to be eco-friendly.
Avoiding the production of plastic and paper cards can be very beneficial for the environment because it prevents the waste of materials (Robertson, 2018). Even though this solution still requires the use of NFC tags, which is made out of coils, plastic and glue (Seritag, 2020), it is still significantly less than those used for creating cards. Moreover, the NFC tag can be programmed again by the retailer avoiding the need of producing or buying new tags.

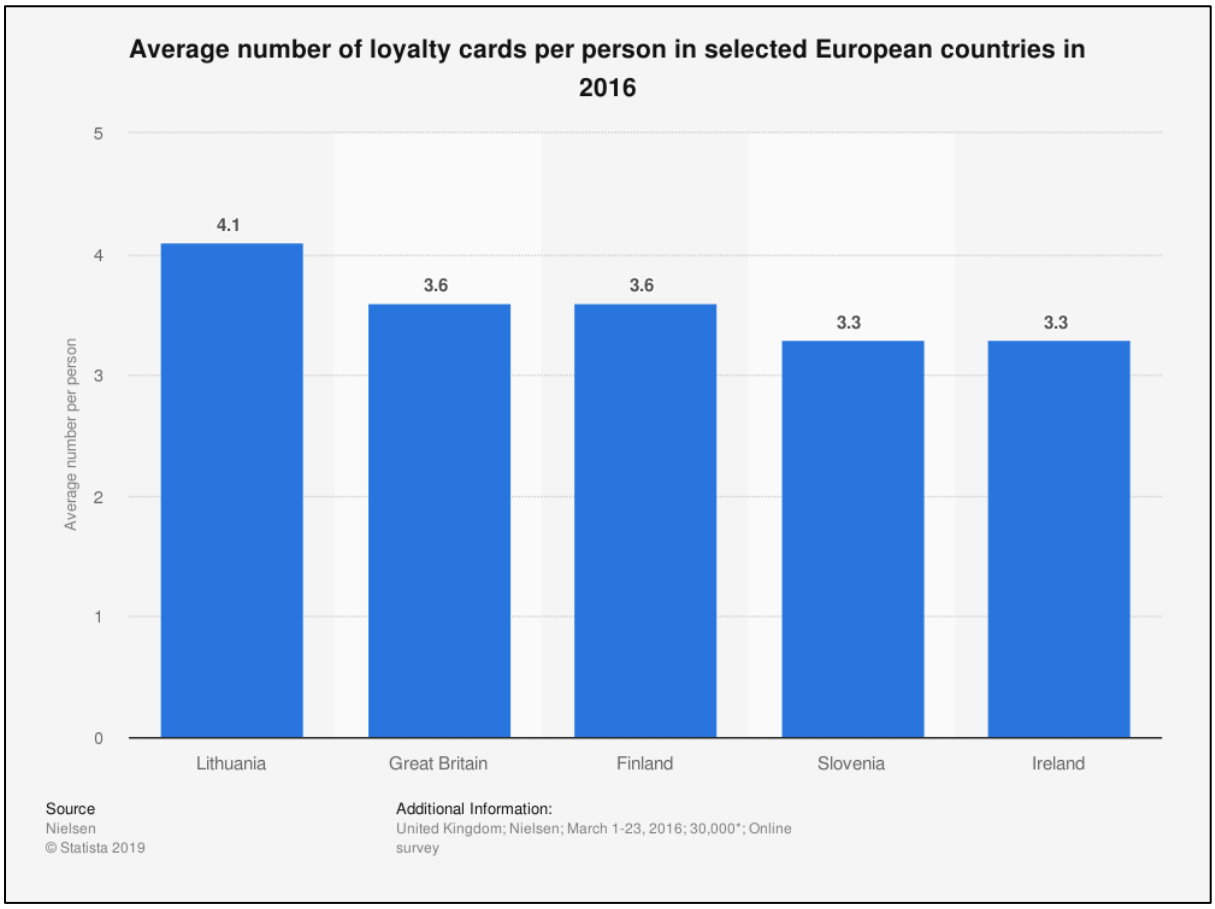


Figure 27. Average number of loyalty cards per person

Combining the previous factor to the statistic of cards per person (see Figure 27), it is possible to point out that by using this solution a lot of waste can be reduced. This is because each person could have three digital cards in place of the physical cards.

4. Methodology

Scrum Agile development methodology was chosen over the traditional software development such as the Waterfall method because of its scalability and flexibility (M. Mahalakshmi, 2013).

Although agile may produce less documentation of the software (Vijayasarathy & Turk, 2008), it is fair to say that the current development is carried out by just one person. Therefore, there is no need for much documentation regarding the application in the early stage of the project. In any case, GitHub will be used as version control tool that can also provide enough documentation about the software.

Moreover, a big advantage of this technique is the early involvement of the stakeholders (Yu Beng Leau, 2012). This aspect means they can provide constant feedback on the prototype delivered, which is crucial for testing and demand purposes.

4.1 Early Iterations of The Project

It is sensible to mention the fact that, initially, the project was intended to be developed through a Mobile Native Application (see Figure 28). Eventually, the new approach was taken because later in the stage, the new capability of the same feature (i.e. Web-NFC) was released and discovered. This was possible thanks to the methodology chosen.

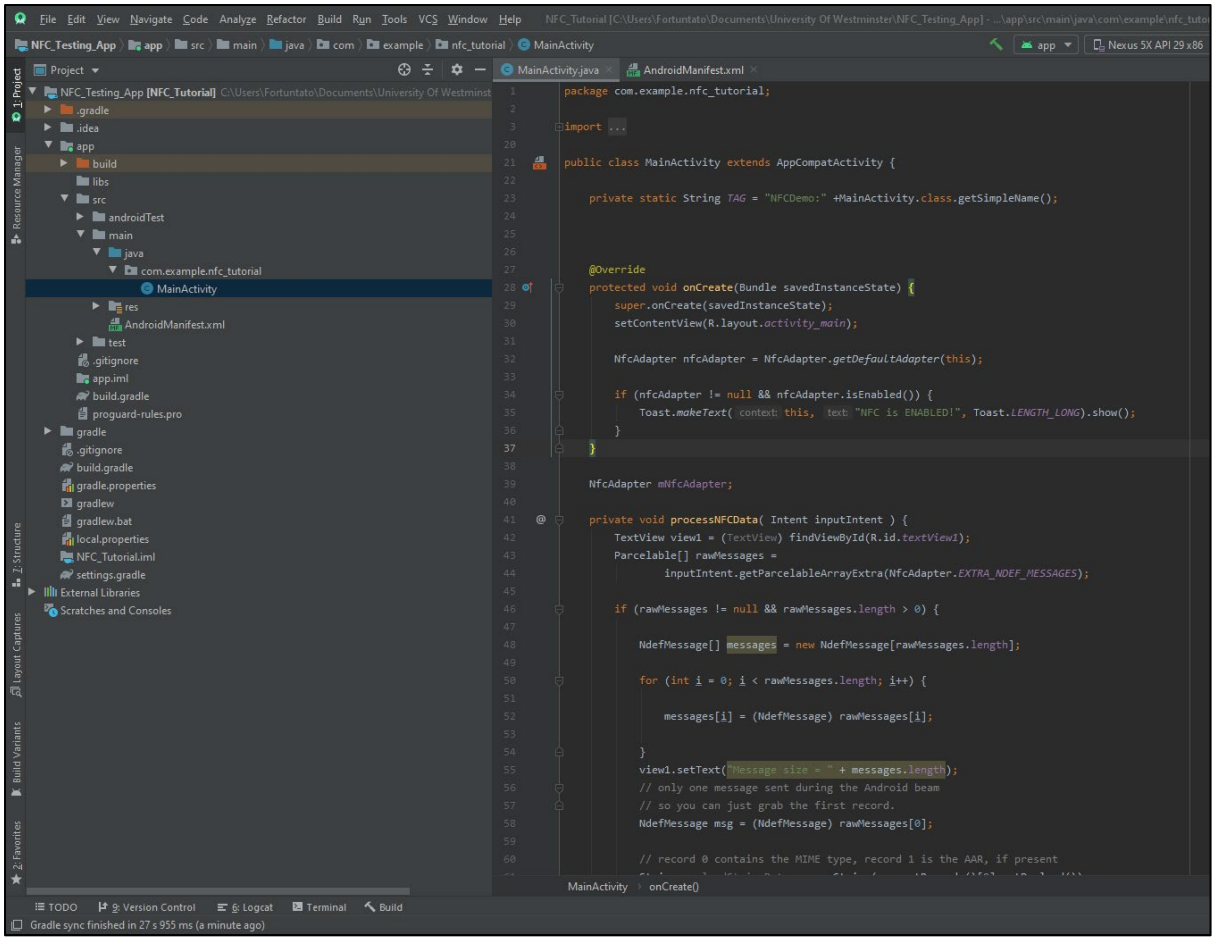


Figure 28. Android Studio project prototype

The initial part of the project was solely concentrated into the research aspect. Being new to the technology implementation, many resources were used to gain knowledge and experience. The first major source was the draft documentation written by W3C (https://w3c.github.io/web-nfc/). Additional guidance was found in the issues board on GitHub (https://github.com/w3c/web-nfc/issues). Using these resources, a Web Application was created as a playground of the new key features during the primary iterations of the project.

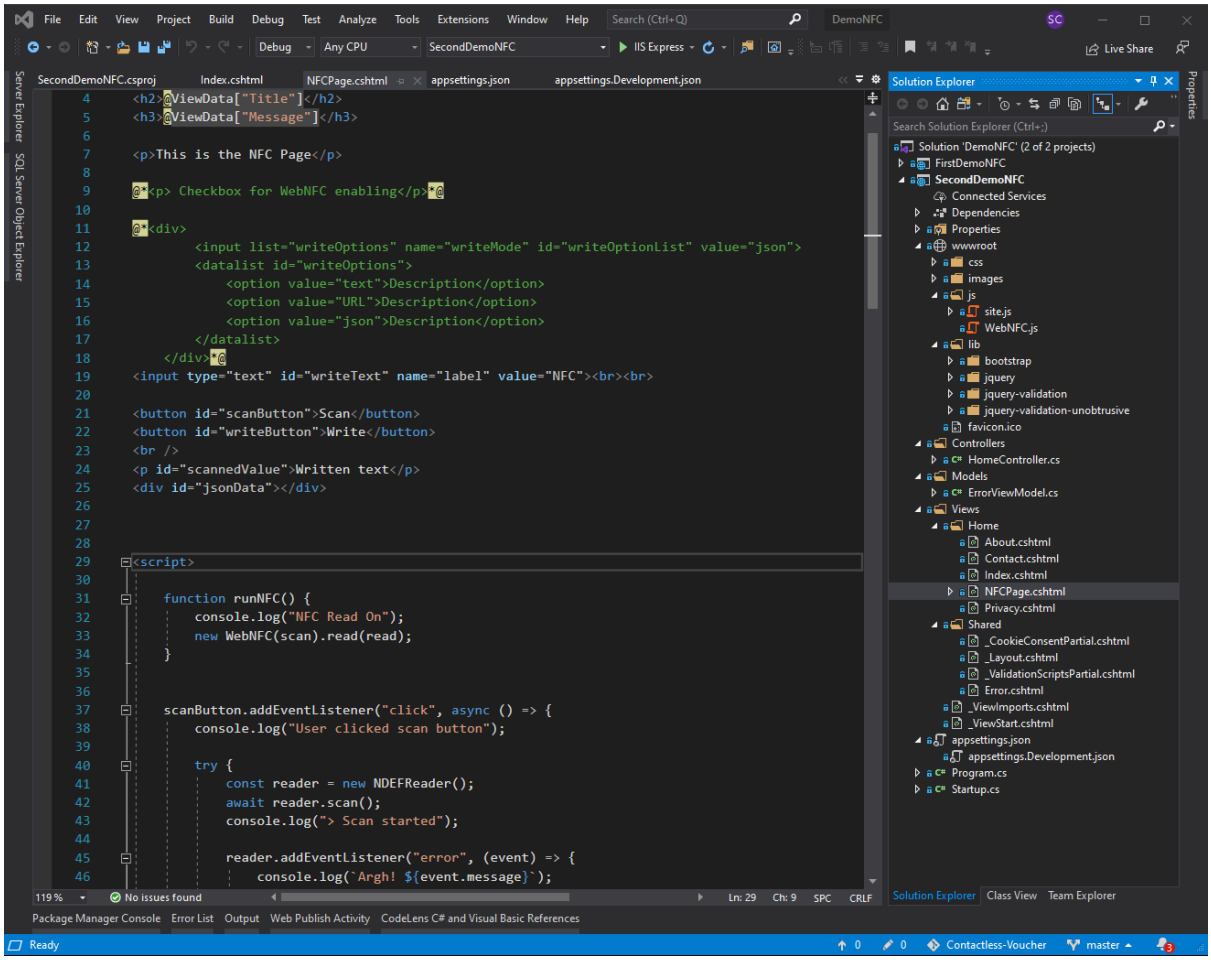


Figure 29. Initial Web App to test Web-NFC (source code)

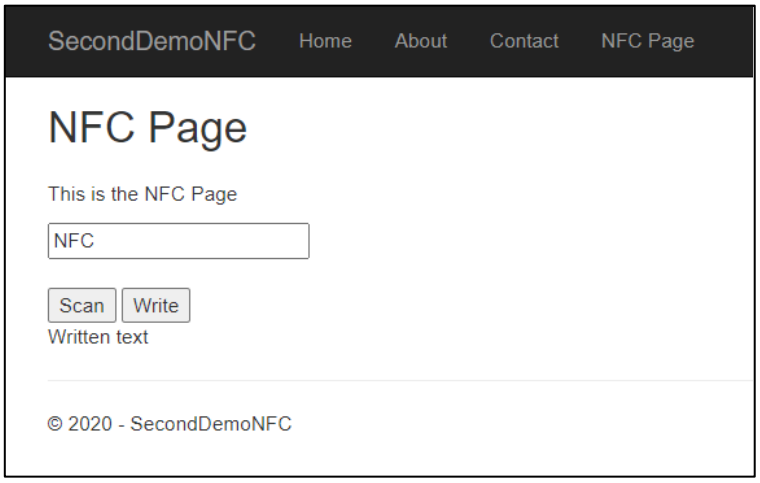


Figure 30. Initial Web App to test Web-NFC (front page)

Thanks to these initial milestones, a list of possible features was created and the interviews with the stakeholders were planned. These made the project take a direction towards the final beta version.

4.2 Iterations with Stakeholders

Throughout the project development, the potential users were always in contact and providing opinions and ideas. Adapting the software to the new designs was possible and practical thanks to the weekly or daily sprints. This key advantage of being able to adjust the software in terms of needs was the proof of the right methodology used.

The core idea of the digital loyalty card was ready to be implemented, but the way the Web App was initially built was different.

The development was following the real business scenarios considered. In fact, this project can be counted as a real case study because of the various commercial aspects (e.g. Loyalty scheme completely digital, innovative and interactive). Additional milestone were created to make sure that the essential legal and ethical issues were met. Therefore, the user registration and login functionality were implemented first. Later, the actual digital card was introduced. Finally, the Web-NFC capability was embedded into the project.

Further changes and design issues were resolved in order to improve the final result.

5. Design

This chapter describes the design solutions for the previously indicated requirements on the front-end and back-end (see 3.4).

Before starting, it is necessary to point out that the Web App is designed to be visited only on smartphone devices. Although, the Web App is accessible by all the platforms, the design issues are not concerned by the alternative layouts because the main capability of the Web-NFC is only available through the smartphone. Therefore, it would not make any sense to implement an optimised design for a desktop version.

5.1 Colour Scheme and Palette

In order to choose an appropriate colour scheme, two major companies where taken into account: Pantone and Adobe (i.e. Adobe Color Blog more specifically).

Adobe is a popular software company that provides powerful tools for designers to create their work and has a big community of artists.

Pantone provides a colour matching system that is the main influence in colour design and trends over the years (Velarde, 2019). The colour chosen for the year 2020 is Classic Blue.

The colour tries to convey a sense of calm, confidence and connection (Milis, 2019). Based upon these values and the colour palette suggested in the Adobe's blog the following colour scheme was chosen (see Figure 31).

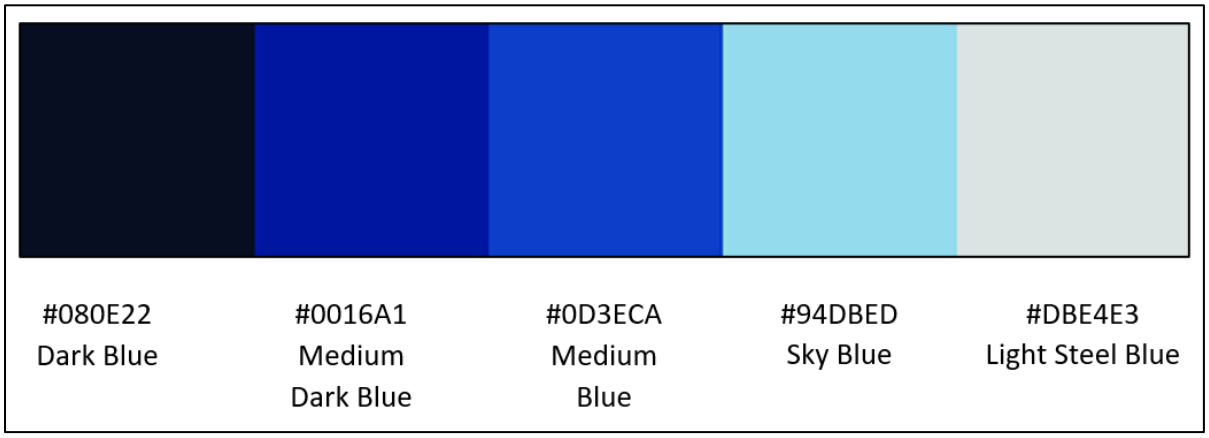


Figure 31. Colour palette for the Web App

These colours are dominant in the whole website, the Dark Blue has been used for the background to give a sense of a dark mode. In contrast, the Light Steel Blue has been used for the text colour as well as some menu options. A gradient of the colour spanning from Medium Dark Blue to Sky Blue has been used for the buttons and the navigation bar. In the banner (see Figure 32) at the top of the login and registration page, all the colours are used.

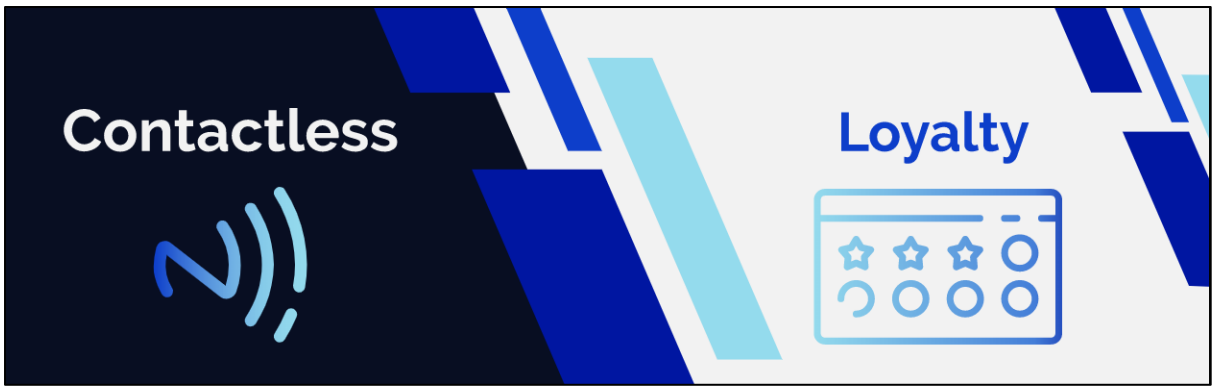


Figure 32. Banner used at the top of the landing page

5.2 CSS and Bootstrap

Regarding the font type, size and general layout of the page CSS (Cascading Style Sheets) and Bootstrap were used.

```
34  body {
35      padding-top: 50px;
36      padding-bottom: 20px;
37      font-family: 'Raleway', Calibri, sans-serif;
38      color: #DBE4E3;
39      background-color: #080E22;
40  }
41
42  /* Wrapping element */
43  /* Set some basic padding to keep content from hitting the edges */
44  .body-content {
45      padding-left: 15px;
46      padding-right: 15px;
47  }
```

Figure 33. CSS body style and body content padding

For example, in the figure above (see Figure 33) the default font type "Raleway" has been declared and used. To make sure the website could find the font-family, there is a reference to the Google Forms library in the head tag of the layout page (i.e. _Layout.cshtml).

```
23      <link href="https://fonts.googleapis.com/css2?family=Raleway&display=swap" rel="stylesheet">
```

Figure 34. Font-family library source declaration

The open source Bootstrap framework has been used for the layout formats as well as making sure the elements on the page were appropriately placed.

```
21  @for (int i = 0; i < Model.NumberOfStamps; i++)
22  {
23  <div class="col-xs-4 stampIcon">
24      <img class="buttonTopMargin" src="~/images/coffeeStamp.svg" alt="" id="stampCollected-@i" onclick="readFromImageClick('@i')" />
25  </div>
26  }
27
28  @for (int i = 0; i < (int)SchemeLimit.WembleyEmporium - Model.NumberOfStamps; i++)
29  {
30  <div class="col-xs-4 stampIcon">
31      <img class="buttonTopMargin" style="margin-top:20px" src="~/images/coffeeIcon.svg" alt="" id="stampToCollect-@i" onclick="readFromImageClick('@i')" />
32  </div>
33  }
```

Figure 35. Code showing the Stamps on the Loyalty Card page with Bootstrap class layout (col-xs-4)

5.3 Razor Page

Razor pages are a type of page which are created with ASP.NET Core framework which helps build dynamic and data-driven web sites (Anderson & Nowak, 2020).

With this capability in mind, the most important page of Contactless Voucher (i.e. Loyalty Card Page. See Figure 40) is implemented dynamically depending on the data retrieved from the database. As shown in the Figure 35 (lines 21 and 28), it is possible to integrate the page with C# code. This makes it convenient to manage the content of the page and integrate advanced layers of security such as Forgery Token validation.

Moreover, it has been discovered and tested, during the sprints, that C# could also be used in the JavaScript section of the page. This has shortened the code implementation and allowed the essential feature of posting the retrieved data back to the controller from the tags.

5.4 SVG Images

The use of SVG (Scalable Vector Graphics) such as the banner (Figure 32) were used for the Loyalty Card page. They make it possible to maintain the same image quality no matter the size of the page. This was a good reason to use these type of illustrations. The main two coffee vectors were taken from Flaticon (https://www.flaticon.com/) and then edited to match the colour scheme and size of the pages.

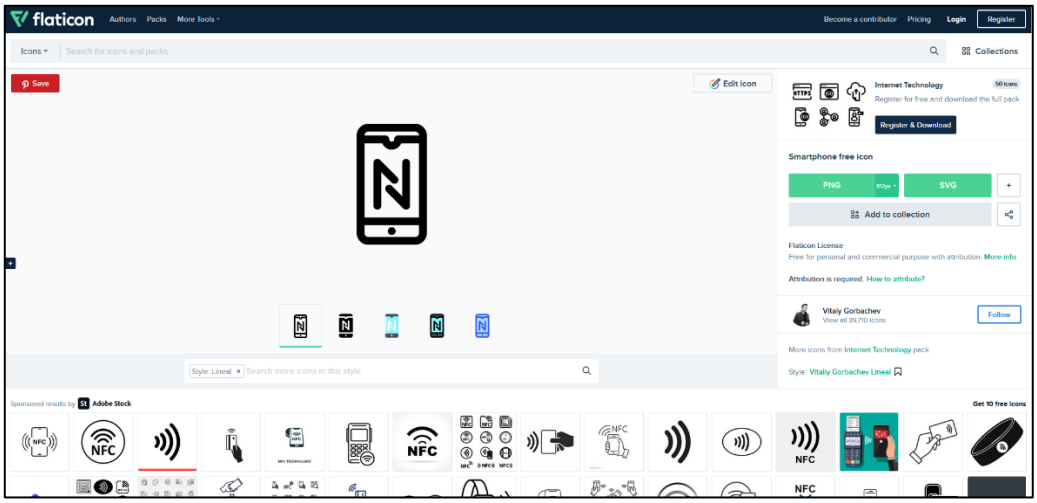


Figure 36. Flaticon image research

5.5 Web App Wireframes with Flowchart

The wireframes of the Web App with corresponding flowchart are presented. For the illustrations please refer to this legend (see Figure 37) for the colours and shapes used.

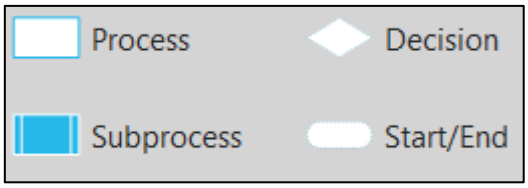


Figure 37. Flowchart legend

In the scenario in which the flowchart does not show the shape corresponding to "End", it is because it has been replaced with the last application wireframe linked.

5.5.1 Login and Registration

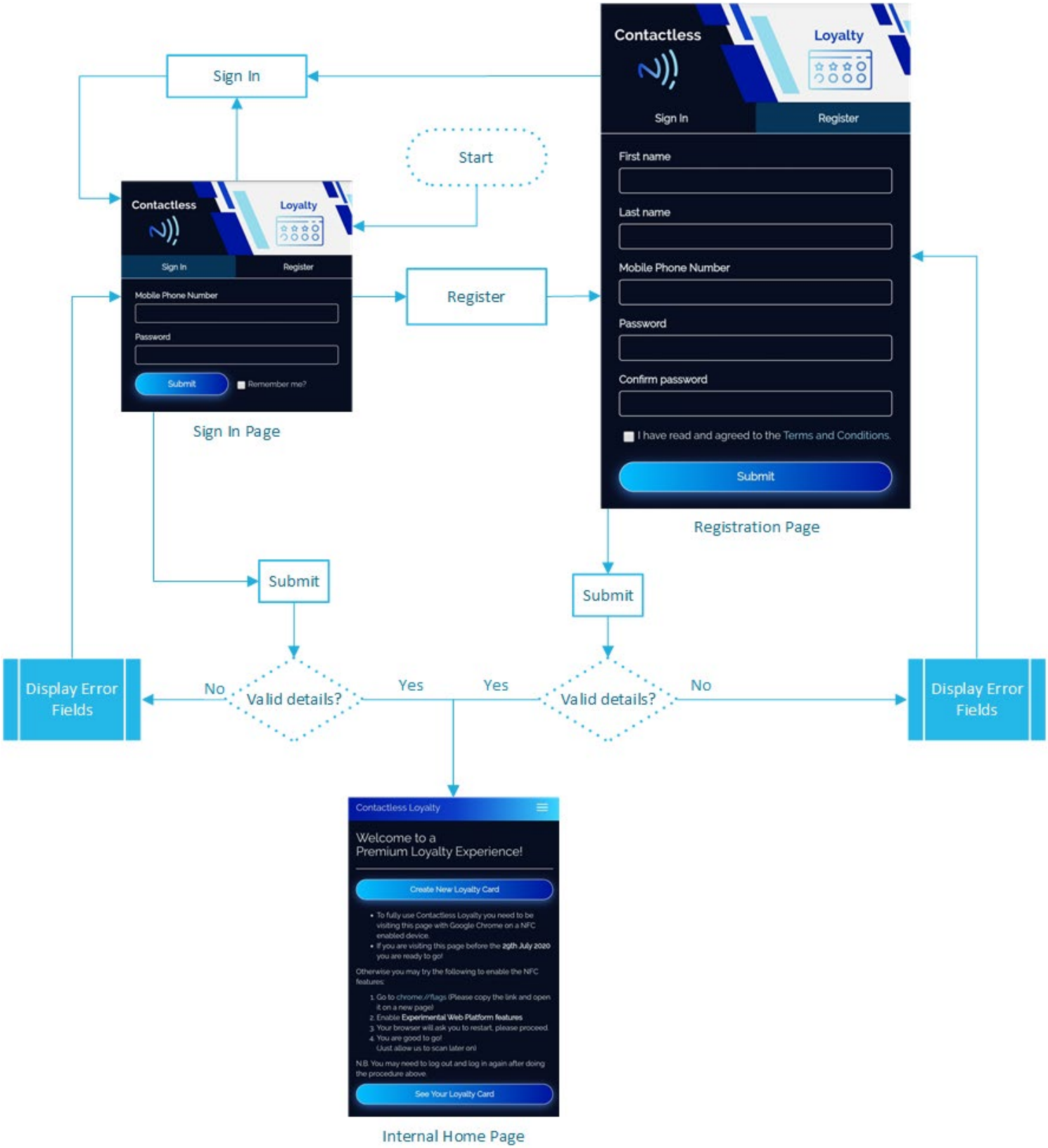


Figure 38. Login and Registration Flowchart with Wireframes

5.5.2 Internal Home Page after Login

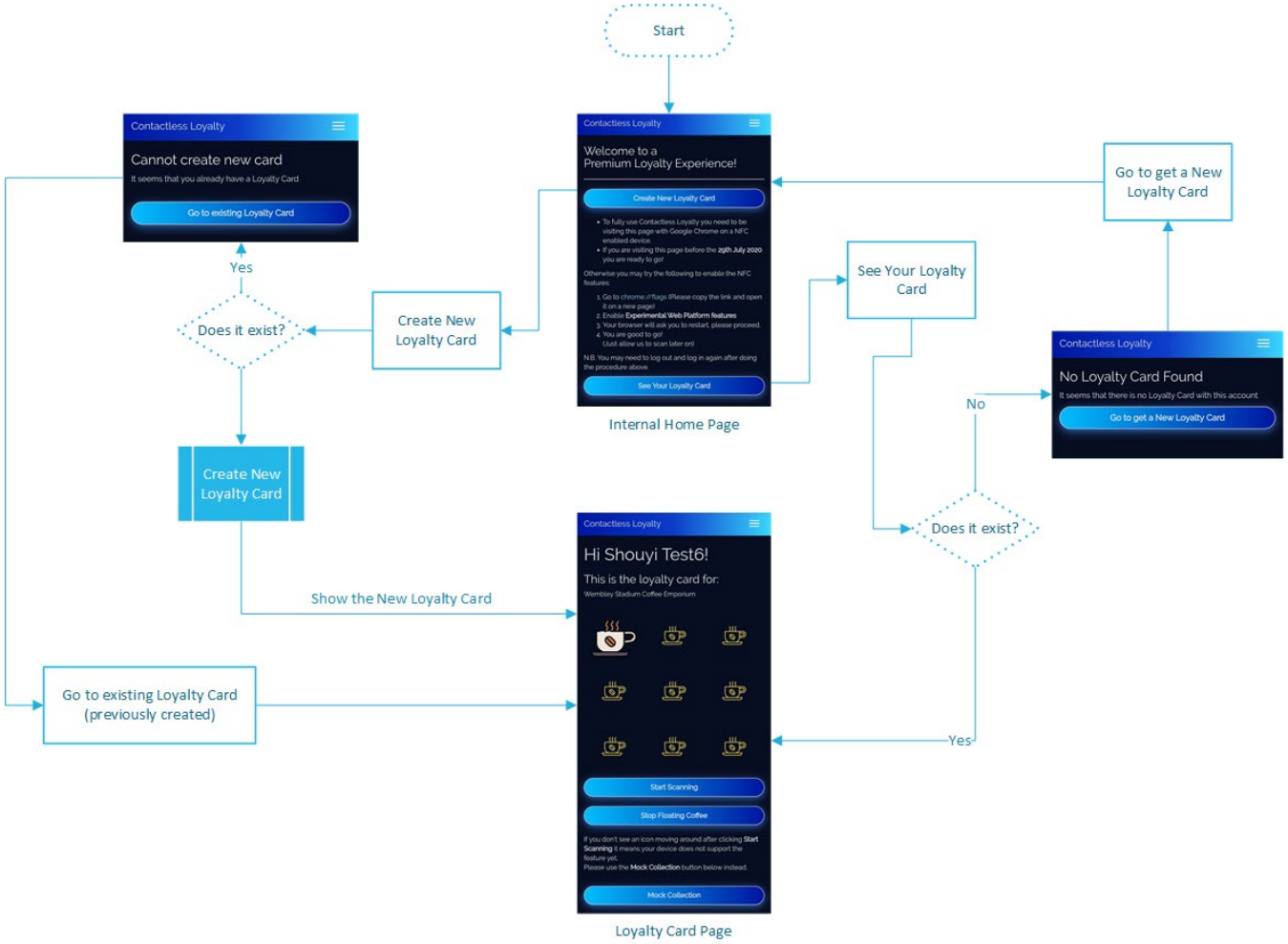


Figure 39. Internal Home Page with Flowchart and Wireframes

5.5.3 Loyalty Card Page

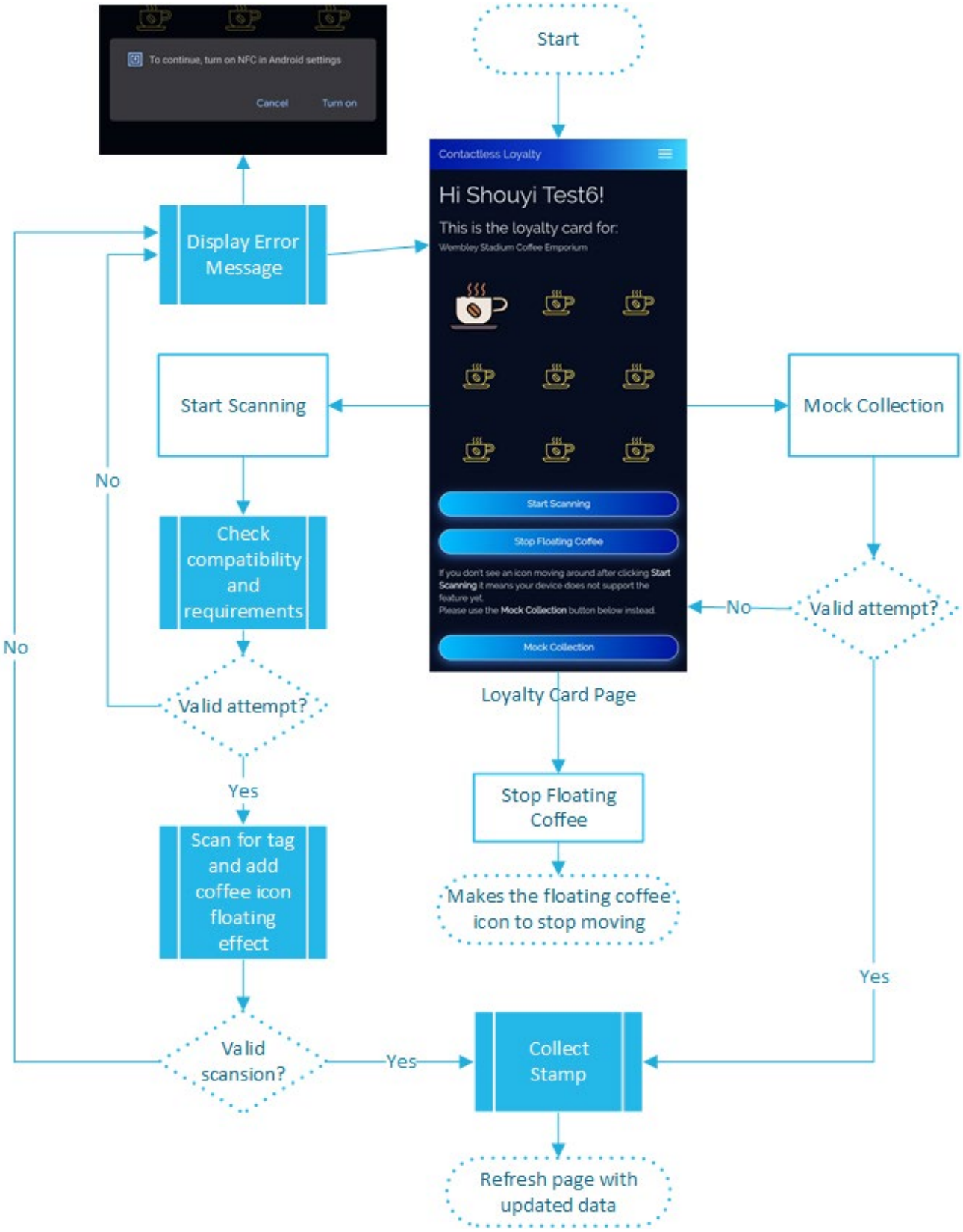


Figure 40. Loyalty Card Page with Flowchart and Wireframes

5.5.4 Reward Available Flowchart and Voucher Sent Page

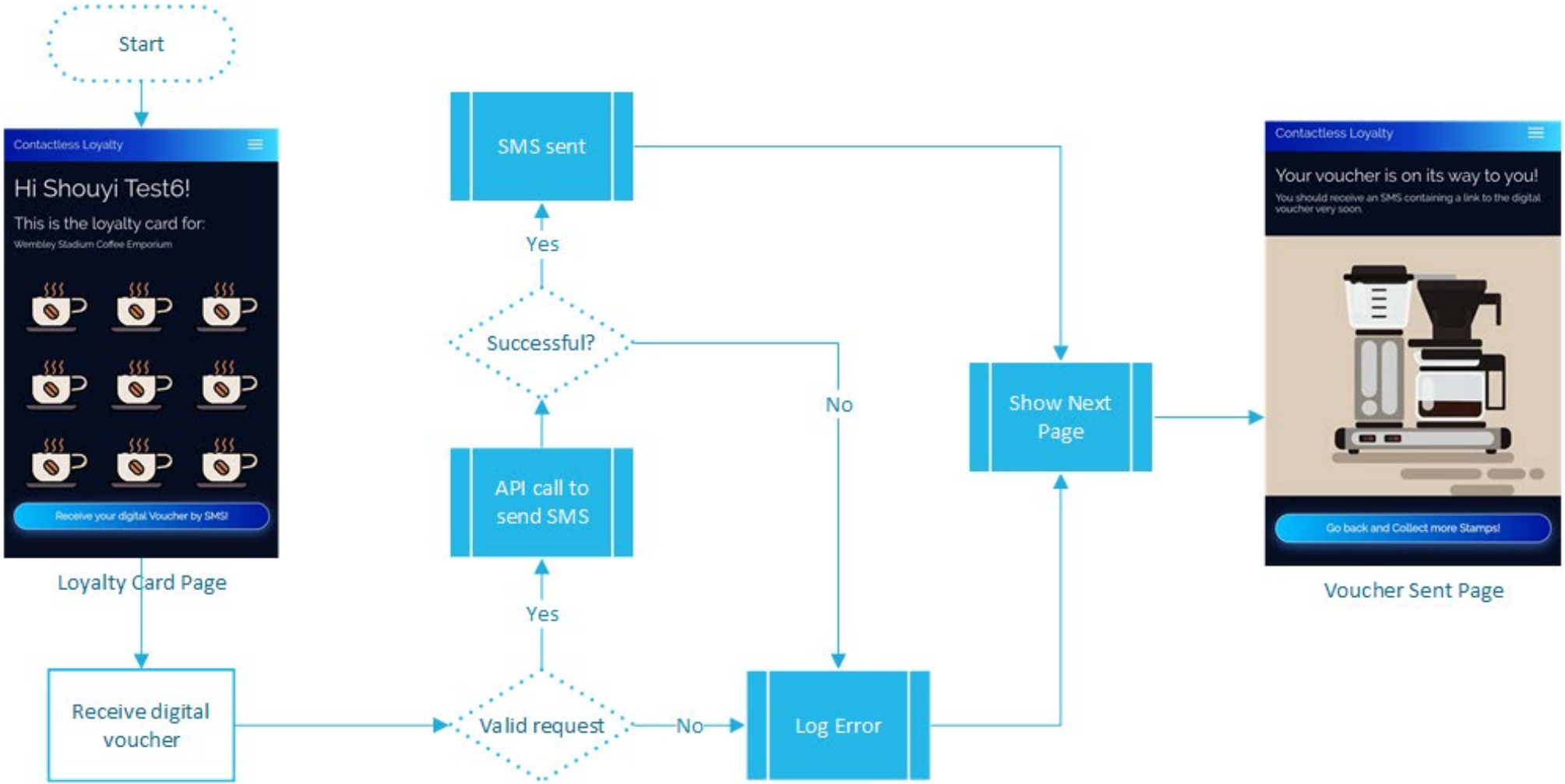


Figure 41. Collection Page Flowchart with Wireframes

5.5.5 Menu Options and Account Management

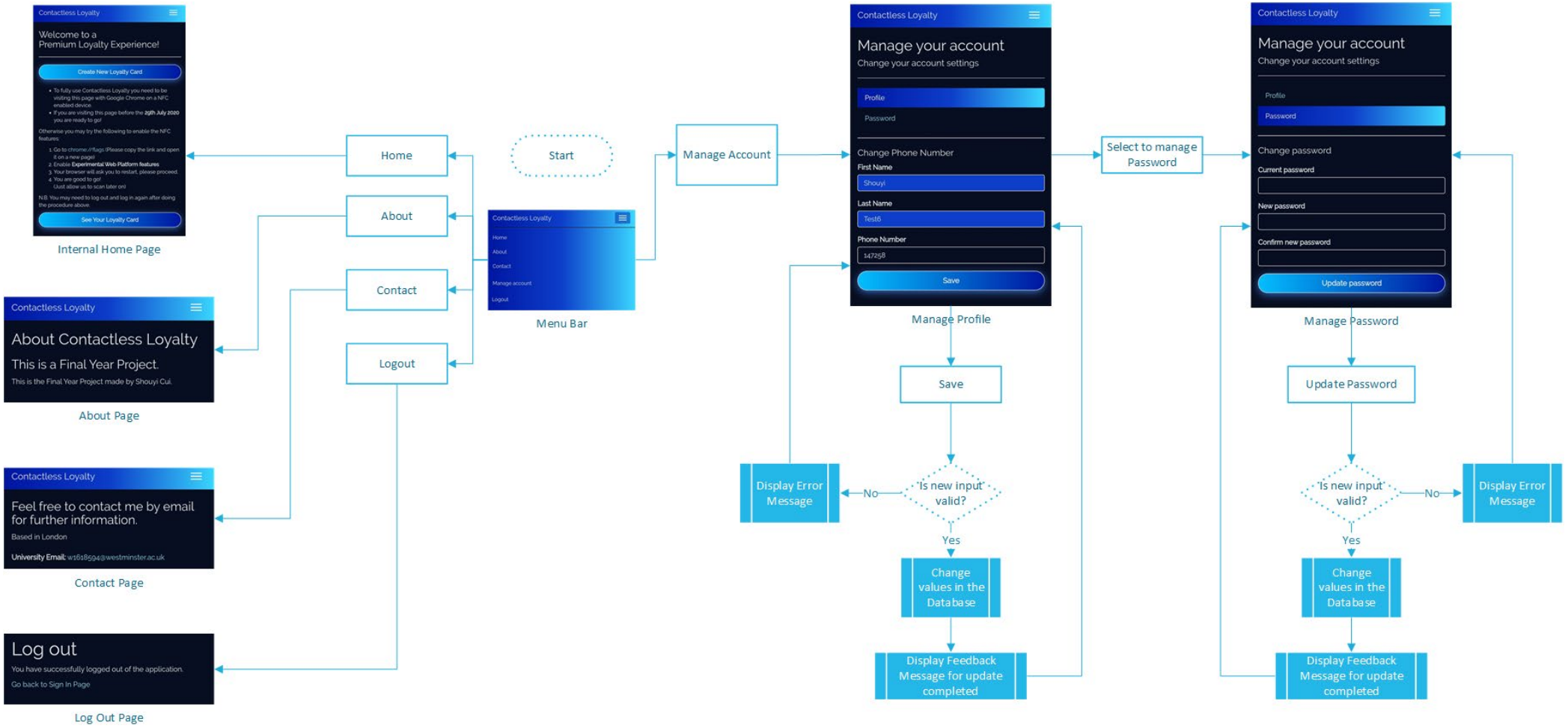


Figure 42. Menu options and Manage Account Page Flowchart with Wireframes

6. Tools and Implementation

This chapter aims to introduce the tools used for the software development and an explanation of the implementation provided.

6.1 Tools

This section provides a list of programs and libraries used in the process of delivering the final Web Application. A justification is provided to support the choice of the tools.

6.1.1 C#

C# is an object-oriented programming language that was created by Microsoft to develop software on systems such as Linux, iOS and Microsoft itself. It is used as the primary language in the Web App source code.

Justification:

- Microsoft provides a rich selection of frameworks to work with this language to deliver Web Apps such as Entity Framework Core an ASP.NET Core.
- Microsoft provides clear documentation for each framework.
- There is a great community support about this language on the Internet.

6.1.2 SQL (Structured Query Language)

SQL is a programming language invented with the scope of managing the data stored in the databases. It is possible to retrieve specific information from the database by running queries. It has been used for checking the data stored with the values expected on the front-end and to update the architecture of the database system when needed.

Justification:

- It is the most popular and efficient way to retrieve, update, insert, delete records (and much more) in a database.
- It has a big community of users that provides solutions online.

6.1.3 HTML, CSS and JavaScript

These set of languages are the basic structure for any web page. HTML is the markup language that structures the different elements on the page. CSS is the language for styling the elements on the page. JavaScript is a programming language created to implement complex features on the web pages such as animation and updating content (Mozilla Web Docs, 2020).

Justification:

- They are the standard language used to deliver web pages on the Internet.

6.1.4 Visual Studio 2019

Visual Studio is an IDE (Integrated Development Environment) developed by Microsoft that helps to create applications.

Justification:

- It is the most popular IDE for Microsoft application development.
- Visual Studio 2019 (v.16.6.3) is currently the latest version available.
- It is possible to install the latest features available from open source plugins.

6.1.4.1 ASP.NET Core 3.1

As previously mentioned in (see 2.3.4) it is an open-source software framework.

Justification:

- See advantages mentioned in section 2.3.4.

6.1.4.2 Entity Framework Core

As previously mentioned (see 2.3.6) is an open-source Object-relational Mapping framework. The chosen version is Core as a result of the choice of ASP.NET Core.

Justification:

- See advantages mentioned in section 2.3.6.

6.1.4.3 StyleCop

StyleCop is an open-source static code analysis. It is part of the available extensions in Visual Studio Marketplace.

Justification:

- It helps in maintaining a clear code layout structure for reading.
- It helps in maintaining a documentation in the source code.

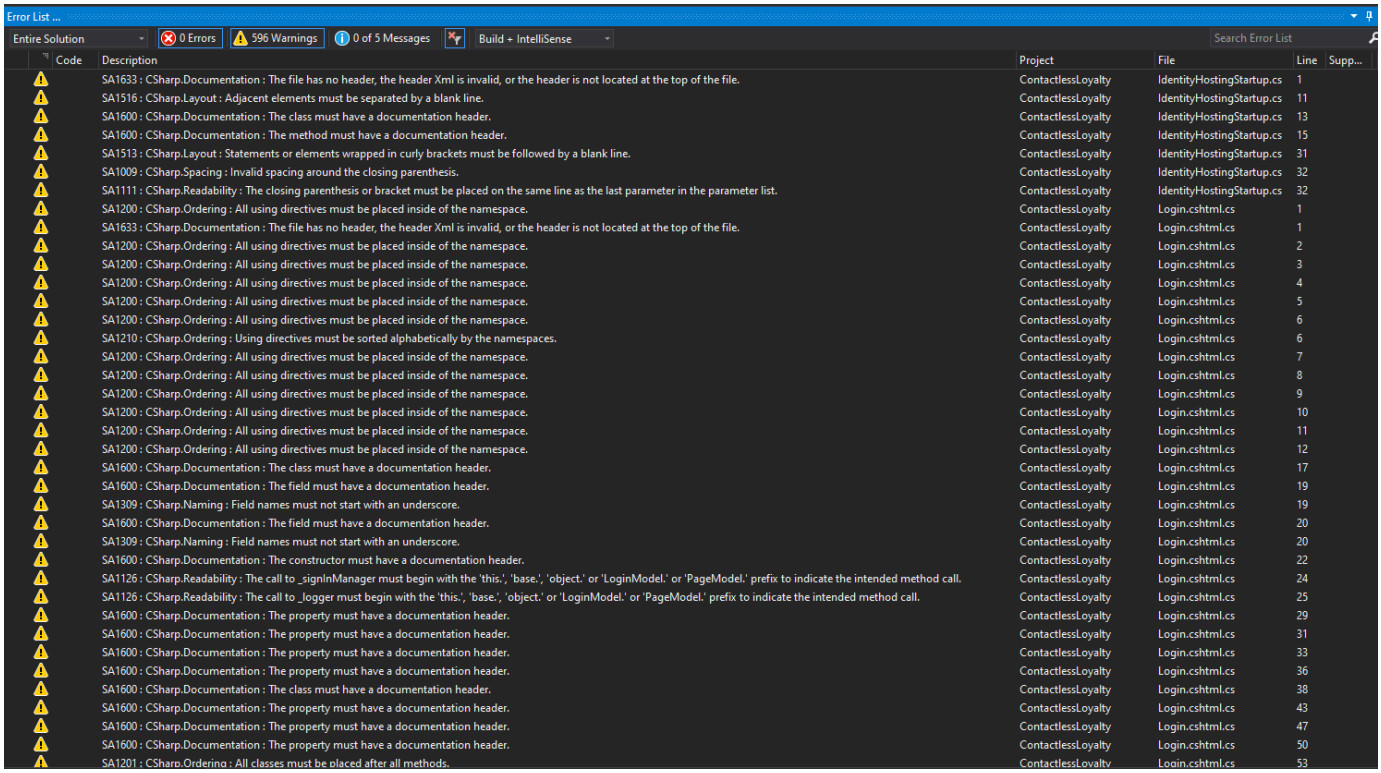


Figure 43. Example of StyleCop warnings

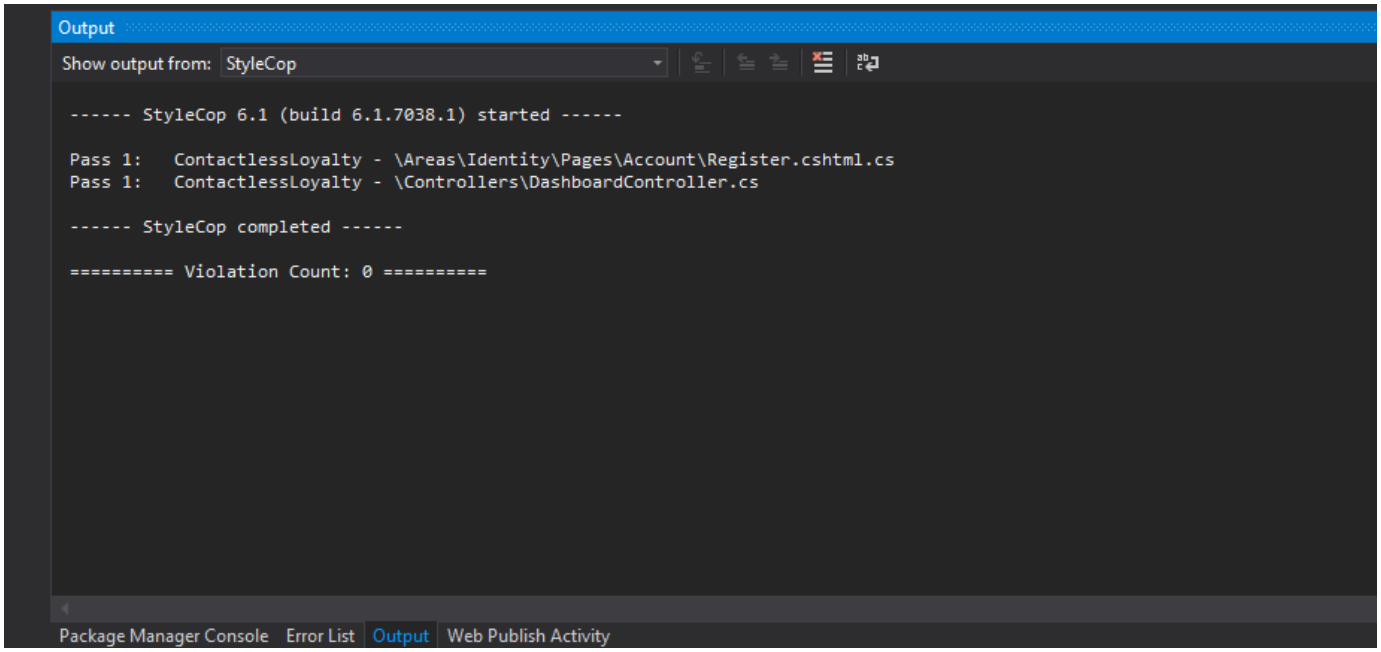


Figure 44. Example of StyleCop without warnings

6.1.4.4 Conveyor

Conveyor is an extension available in the Visual Studio Marketplace. It is designed to make easier the testing of the application across devices. It was used to enable the access of the Visual Studio development server on the smartphone.

- Easy setup and integration with Visual Studio.
- Quick testing environment on the smartphone without using cables.

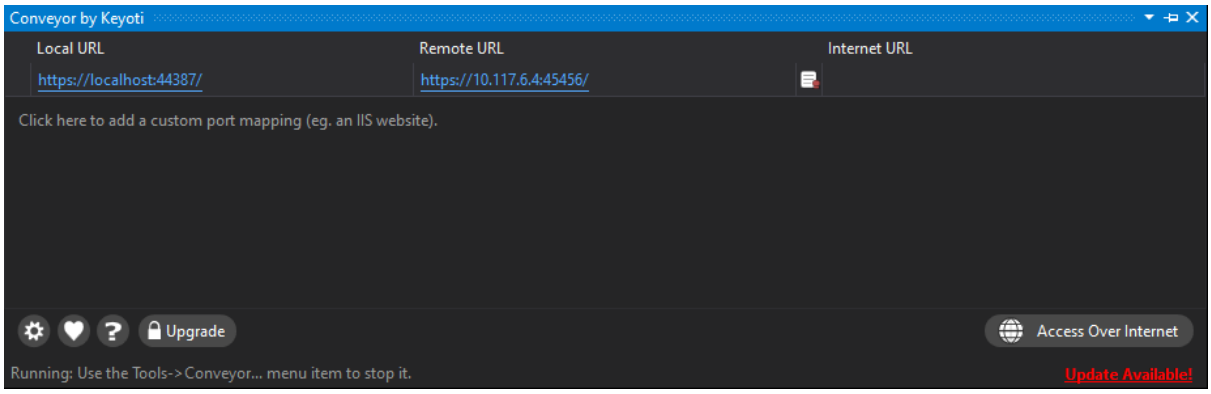


Figure 45. Conveyor running on Visual Studio example

6.1.5 Microsoft SQL Server Management Studio

Microsoft SQL Server is a software application used to access database server and configure, manage and retrieve data.

Justification:

- It is part of the same organisation provider of Visual Studio.
- It is simple to use and well documented.
- Provides a set of additional tools for database management such as stored procedures.

6.1.6 Git

Git is an open source version control system. It has the capability to be used as a task management tool but that aspect of the project development is covered by Paymo (see 6.1.11). Git has been used for source version control since the first iteration of the project including the documentation files.

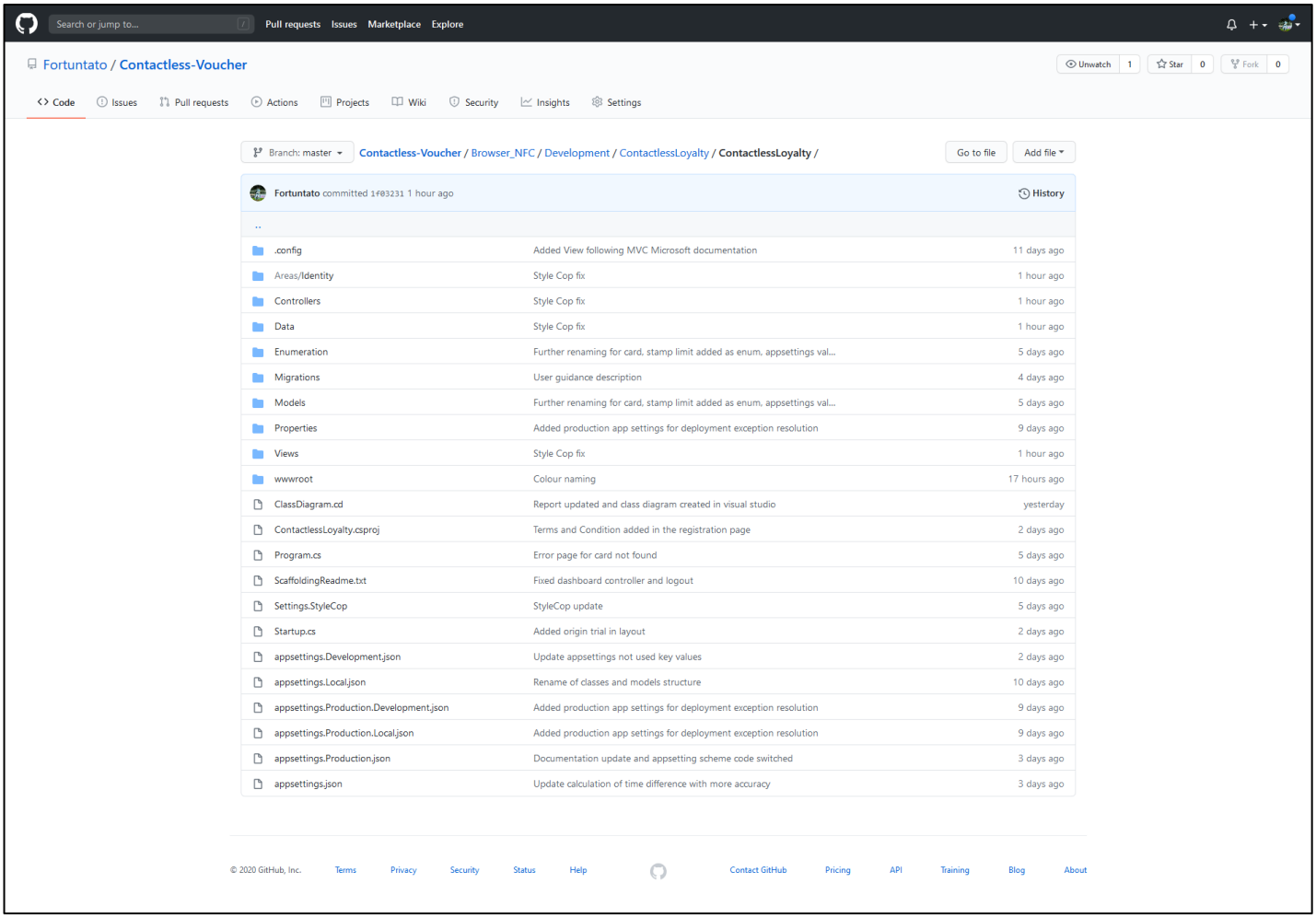


Figure 46. Git Repository of the Contactless Loyalty source code

Furthermore, GitHub (https://github.com) has been used as a source of discussion with the developers of the Web NFC API (see 10.7).

Justification:

It is easy to install and use given that the developer knows the principles of version control.

It is the most popular and open source version control with a large community support.

It allows the project to be visible by others that may wish to collaborate.

6.1.7 Chrome Developer Tools

Chrome Developer Tools is a set of tools built in Google Chrome that allows developers to edit pages on-the-fly and debug the code to find out problems.

Justification:

- Already built in the browser used for testing.

- Belongs to the company (i.e. Google) that allows the use of the Web-NFC.

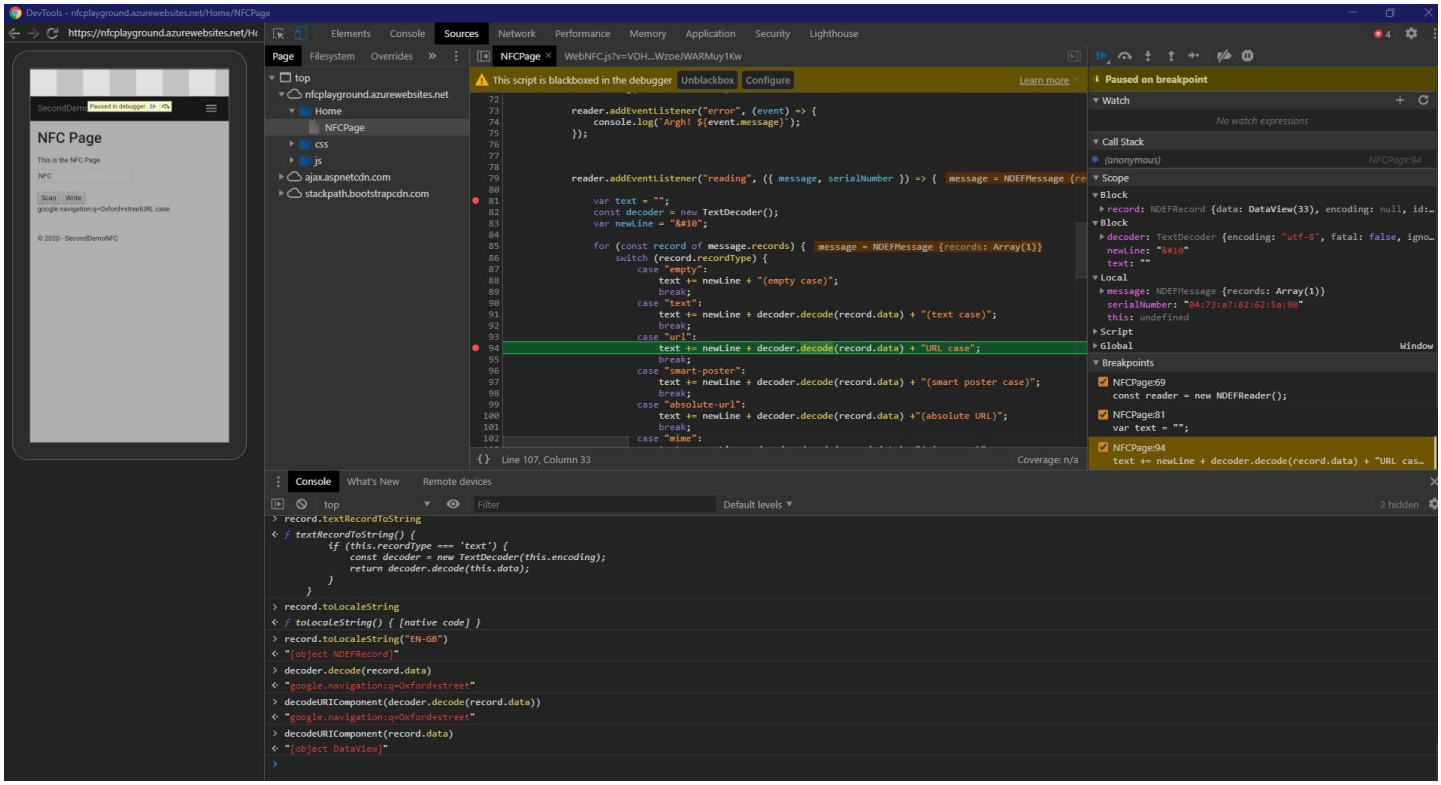


Figure 47. Example of early stage DevTools utilisation

6.1.8 Adobe Illustrator

Adobe Illustrator is a vector graphics editor. It is a software developed by Adobe and it has a lot of capabilities such as creating vectors out of a PNG image.

Justification:

- Large community of artists who can provide guides and tutorials.
- There was experience and qualification for this program prior to the start of this project.
- Uses vectors graphics that guarantees high quality image that is essential for a good UI.

Figure 48. Adobe Illustrator banner template in development

6.1.9 Microsoft Visio

Microsoft Visio is an application developed for making diagrams and flowcharts. It was used a major tool for creating all the diagrams like the Use Case Diagram (see Figure 21) and the illustrations in Web App Wireframes with Flowchart (chapter 5.5).

Justification:

- It provides many useful templates.
- It provides various type exporting format.
- Uses vectors graphics that guarantees high quality image that is essential for large diagrams.

6.1.10 Postman

Postman is a software development tool created for testing API calls and display returned values in various format. This tool has been used to test the i-movo API calls with different parameters.

Justification:

- Easy to install and use.
- Popular choice among developers.
- Store data of the tests with the corresponding response.

6.1.11 Paymo Web App/ Paymo Widget

Paymo is a company that provides a cloud-based work and project management solution. They provide a Web App, a Desktop Widget App and a Mobile Application for time tracking. It has mostly been used for task list and time tracking.

Justification:

- Easy to use with simple UI.
- Great customer service and support.
- Cross platform and available on all devices.

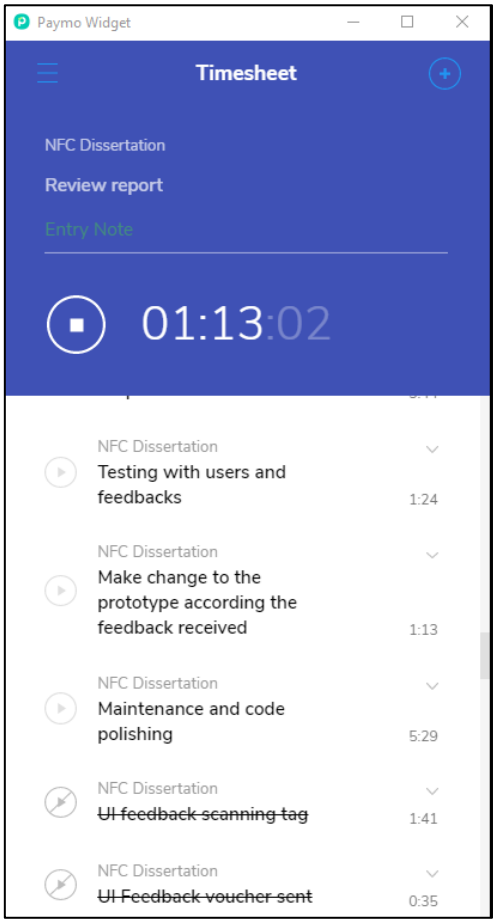


Figure 49. Paymo Desktop Widget example of active time tracking

6.2 Implementation - Loyalty Scheme system

This section aims to cover all the key functions of the Web Application along with the explanation of code produced.

6.2.1 Requirements and Dependencies

Since the project solution uses many frameworks as previously mentioned, in order to build the project correctly and run it, an initial setup may be needed.

First of all, as previously mentioned (see 6.1.4), the Visual Studio 2019 version used for development is 16.6.3 which is the most up-to-date version.

A list of the required frameworks and packages is shown below (see Figure 50).

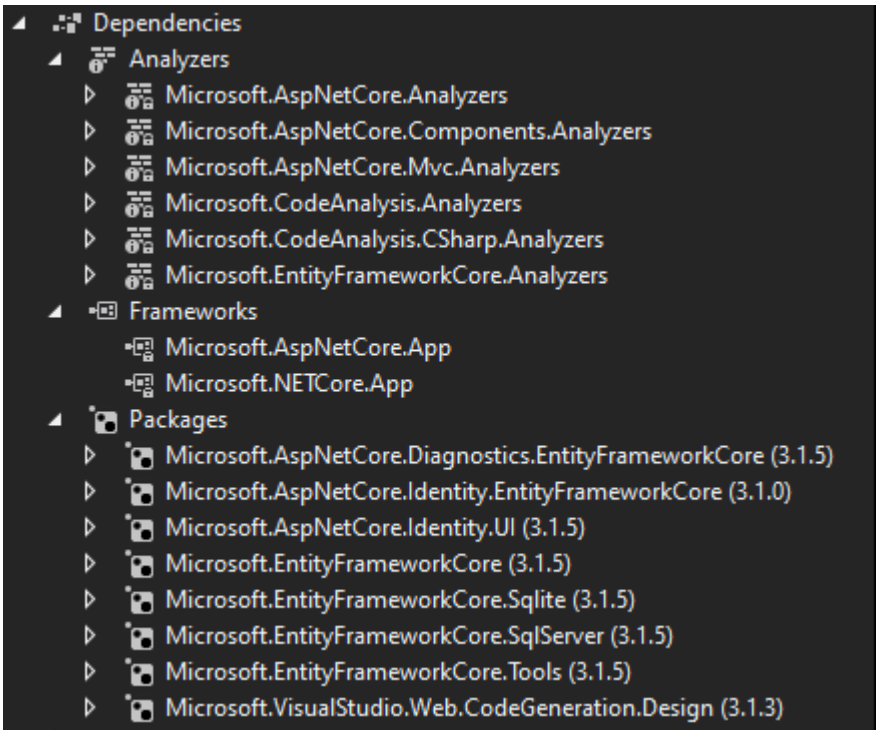


Figure 50. Project dependencies

Follows an example for the package installation through Package Manager Console available in Visual Studio.

The command entered is “Install-Package Microsoft.EntityFrameworkCoreSqlServer”.

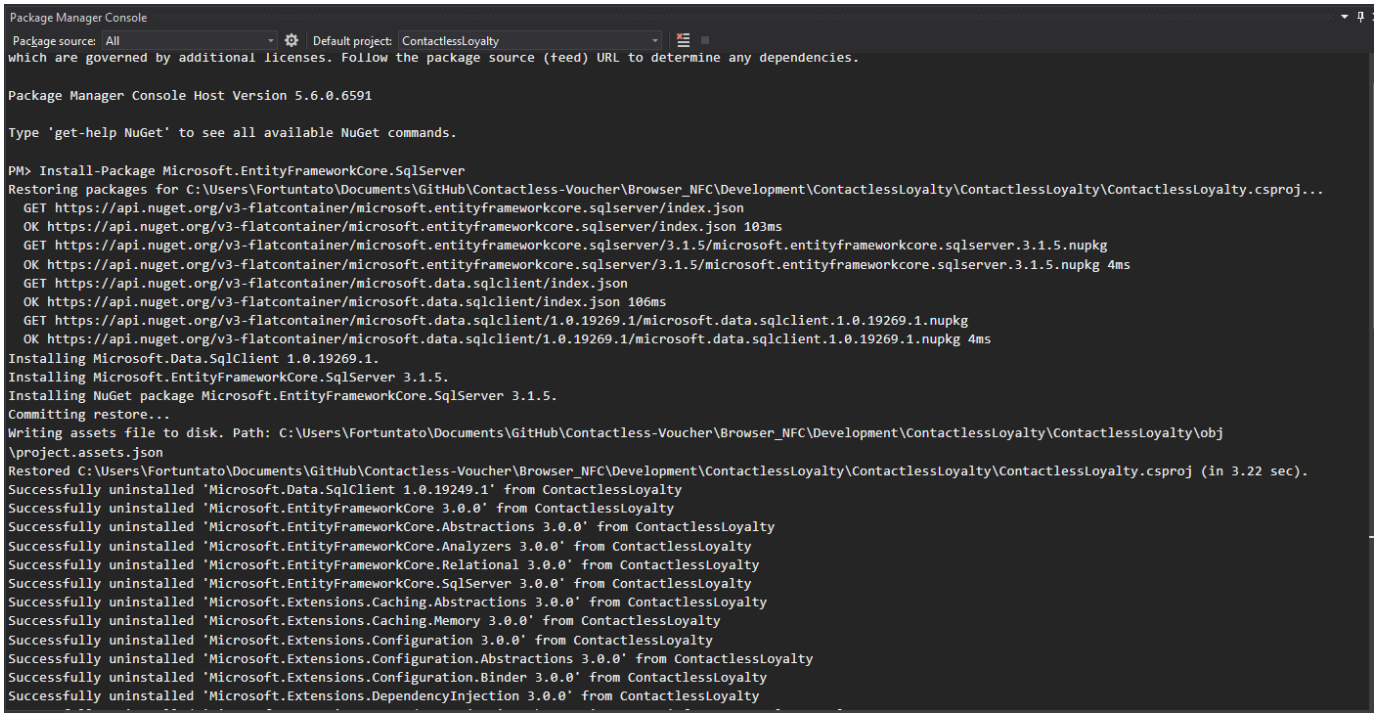


Figure 51. Installing Microsoft Entity Framework Core on Package Manager Console (line 4 onwards)

6.2.2 Database Setup

The resources from Entity Framework Code were used for the setup of the database.

If the source code of the project is taken from the GitHub repository, in order to create a database ready to use only two steps are required:

1. Edit the connection string in the appsettings.json file. By default Visual Studio uses the original application settings and then it overrides some values depending on the environment (see Figure 52).

2. Go to Package Manager Console and run the command "Update-Database". This will take the InitalCreate.cs code to run the script in the targeted database (see Figure 53).

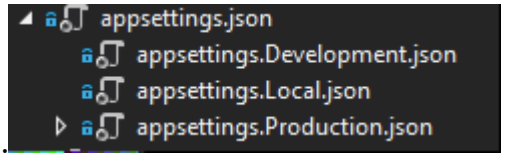


Figure 52. Settings for the project with the different environments

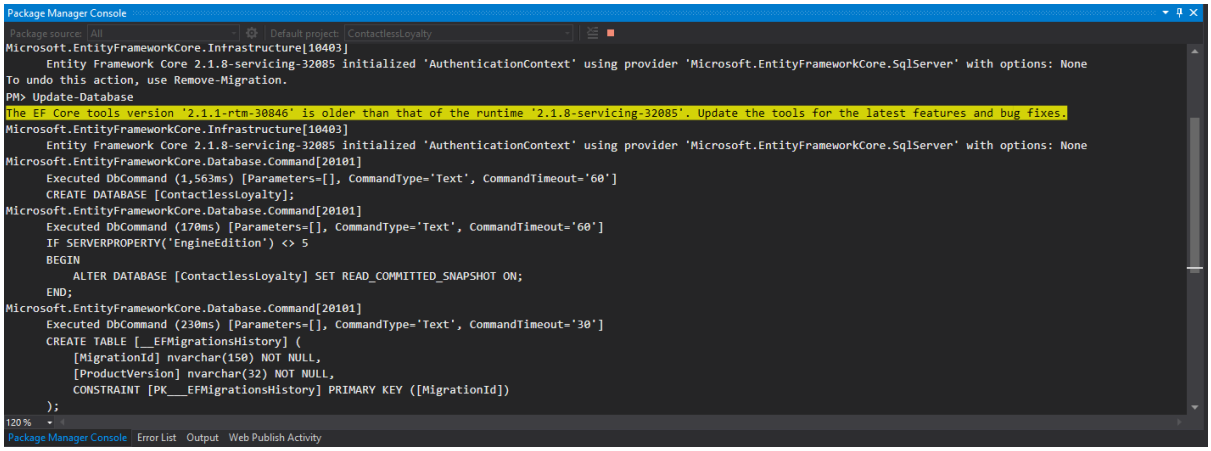


Figure 53. Example of Update-Database command successfully running

The key value to be changed is "AuthenticationContextConnection" because it refers into the database context declared in the Startup.cs.

```
39          services.AddDbContext<DatabaseContext>(options =>
40          options.UseSqlServer(Configuration.GetConnectionString("AuthenticationContextConnection")));
41          services.AddRazorPages();
```

Figure 54. Database context declaration in Startup.cs

The line 40 in Figure 54 tells the application to look for a connection string with the key name equal to the string in quotes.

6.2.3 Login and Registration

After evaluating the different approaches for implementing the user registration and login, it was chosen to use the "scaffolding" technique to generate default classes rather than implementing everything from zero.

The class was then customised to use the mobile phone number as way to sign into Contactless Loyalty.

```
82          Microsoft.AspNetCore.Identity.SignInResult result = await _signInManager.PasswordSignInAsync(Input.PhoneNumber,
83              Input.Password,
84              Input.RememberMe,
85              lockoutOnFailure: false);
```

Figure 55. Sign with Phone Number

On the other hand, for the registration process some additional check were added to conform with the GDPR. The validation for the user input could be added in the model properties because the system is designed to use MVC.

```
1 reference
public class InputModel
{
    [Required]
    [DataType(DataType.Text)]
    [Display(Name = "First name")]
    1 reference
    public string FirstName { get; set; }

    [Required]
    [DataType(DataType.Text)]
    [Display(Name = "Last name")]
    1 reference
    public string LastName { get; set; }

    [Required]
    [Phone]
    [Display(Name = "Mobile Phone Number")]
    [RegularExpression("^44[0-9]{9}|07[0-9]{9}$", ErrorMessage = "Please enter a valid UK phone number starting with 44 or 07")]
    2 references
    public string PhoneNumber { get; set; }

    [Required]
    [StringLength(100, ErrorMessage = "The {0} must be at least {2} and at max {1} characters long.", MinimumLength = 6)]
    [DataType(DataType.Password)]
    [Display(Name = "Password")]
    1 reference
    public string Password { get; set; }

    [DataType(DataType.Password)]
    [Display(Name = "Confirm password")]
    [Compare("Password", ErrorMessage = "The password and confirmation password do not match.")]
    0 references
    public string ConfirmPassword { get; set; }

    [Required]
    [Display(Name = "I have read and agreed to the")]
    1 reference
    public bool AcceptTermsAndConditions { get; set; }
}
```

Figure 56. InputModel for the Registration Page

Before storing the user details in the registration process, an additional check in the back-end was added to see if the user had checked the box for the "Terms of Use" (see line 88 in Figure 57). Moreover, the default registration process already implements the uniqueness of the new user to not match an already existing one (i.e. There cannot be two users with the same phone number).

```
0 references
public async Task<IActionResult> OnPostAsync(string returnUrl = null)
{
    returnUrl = returnUrl ?? Url.Content("~/");

    if (!Input.AcceptTermsAndConditions)
    {
        ModelState.AddModelError("AcceptTermsAndConditions", "Please accept the Terms and Conditions agreement.");
    }

    if (ModelState.IsValid)
    {
        AccountContactlessLoyaltyUser user = new AccountContactlessLoyaltyUser
        {
            FirstName = Input.FirstName,
            LastName = Input.LastName,
            UserName = Input.PhoneNumber,
            PhoneNumber = Input.PhoneNumber
        };

        IdentityResult result = await _userManager.CreateAsync(user, Input.Password);
        if (result.Succeeded)
        {
            _logger.LogInformation("User created a new account with password.");

            await _signInManager.SignInAsync(user, isPersistent: false);
            return LocalRedirect(returnUrl);
        }

        foreach (IdentityError error in result.Errors)
        {
            ModelState.AddModelError(string.Empty, error.Description);
        }
    }
```

Figure 57. Registration Post Request for new user

6.2.4 Creating a Loyalty Card

```
public async Task<IActionResult> CreateCard()
{
    // Get the user id to store with the new card
    AccountContactlessLoyaltyUser user = await _userManager.GetUserAsync(User);
    if (user == null)
    {
        return NotFound($"Unable to load user with ID '{_userManager.GetUserId(User)}'.");
    }

    // Check if the user has already a card
    List<Card> userCards = await _context.LoyaltyCards.Where(x => x.User.Id == user.Id).ToListAsync();

    if (userCards.Count > 0)
    {
        return RedirectToAction("OneCardLimit", "Card");
    }

    // Parameters for creating a new card
    Card newCard = new Card(
        0,                                  // Number of initial vouchers
        new DateTime(2013, 05, 26),         // Initial date
        1,                                  // Number of initial stamps
        "Wembley Stadium Coffee Emporium", // Store name
        _configuration.GetValue<string>("CustomSettings:StoreSchemeCode"), // Number of stamps required for reward
        user);

    _context.Add(newCard);
    try
    {
        await _context.SaveChangesAsync();
    }
    catch (Exception error)
    {
        _logger.LogError(error.Message);
    }

    return RedirectToAction("Index", "Card");
}
```

Figure 58.CreateCard method in CardController

The method shown in Figure 58 is what creates a new card in the database. It does not need any input by the user because it uses some default values (lines 94 to 97) and retrieves the information of the user by the identifying who is logged in.

Lines 85 and 87 make a search in the database to see if the user have already created a card by creating a list of cards that matches the userID (see the "where" condition on line 85). If the user has already created a card, he is redirected to an error page (line 89). The error page is referenced as declared in the line 66 in Figure 59, action name and controller.

```
63          app.UseEndpoints(endpoints =>
64          {
65              endpoints.MapRazorPages();
66              endpoints.MapControllerRoute("default", "{controller=Home}/{action=Index}/{id?}");
67          });
```

Figure 59. Definition of controller route

6.2.5 Loyalty Card Information

To display the appropriate Loyalty Card details, a Card Model with the appropriate data is sent to the View that renders the information given.

```
49          List<Card> userCards = await _context.LoyaltyCards.ToListAsync();
50
51          // At the moment is getting only the first card found in the DB. TODO: Check the correspondent card
52          Card dashboard = userCards.Where(x => x.User == user).FirstOrDefault();
53
54          if (dashboard == null)
55          {
56              return RedirectToAction("ErrorDetails", "Card");
57          }
58
59          return View(dashboard);
```

Figure 60. CardController providing data for the View through a Card Model class

On line 52 in the Figure 60, the Card details associated to the user logged in are found. To note that is taking only the first element in the list declared on line 49. Only card is taken to avoid the problem of the user accidentally having multiple cards. For this beta version the user is allowed to only have one card. In the future, the user would be allowed to hold multiple cards and therefore additional validation would be needed.

The View is a Razor Page that allows the use of C# (as mentioned in Razor Page5.3) to display the appropriate number of "Stamp" icons (i.e. Coffee vector images).

Line 21 and 28 in Figure 35 are the two for loops used to show the right number of coffee images.

To note that in each image created there is a different number added to the ID name. The number values is taken from the value of the variable in the for loop (see line 24 in Figure 61).

```
@for (int i = 0; i < Model.NumberOfStamps; i++)
{
<div class="col-xs-4 stampIcon">
    <img class="buttonTopMargin" src="~/images/coffeeStamp.svg" alt="" id="stampCollected-@i" onclick="readFromImageClick('@i')" />
</div>
}
```

Figure 61. C# variable added in HTML image id value

6.2.6 Stamp Collection and Validation

The collection of the stamp or loyalty point through the scansion of an NFC tag is the main aspect of Contactless Loyalty and the key innovating function of the Web App.

There are two ways to enable the Web-NFC API on the browser.

The first option needs a couple of steps for the user:

1. Go to chrome://flags
2. Search for the flag called "Web Experimental Platform Features" (see Figure 62).
3. Enable the flag.
4. Restart the browser.

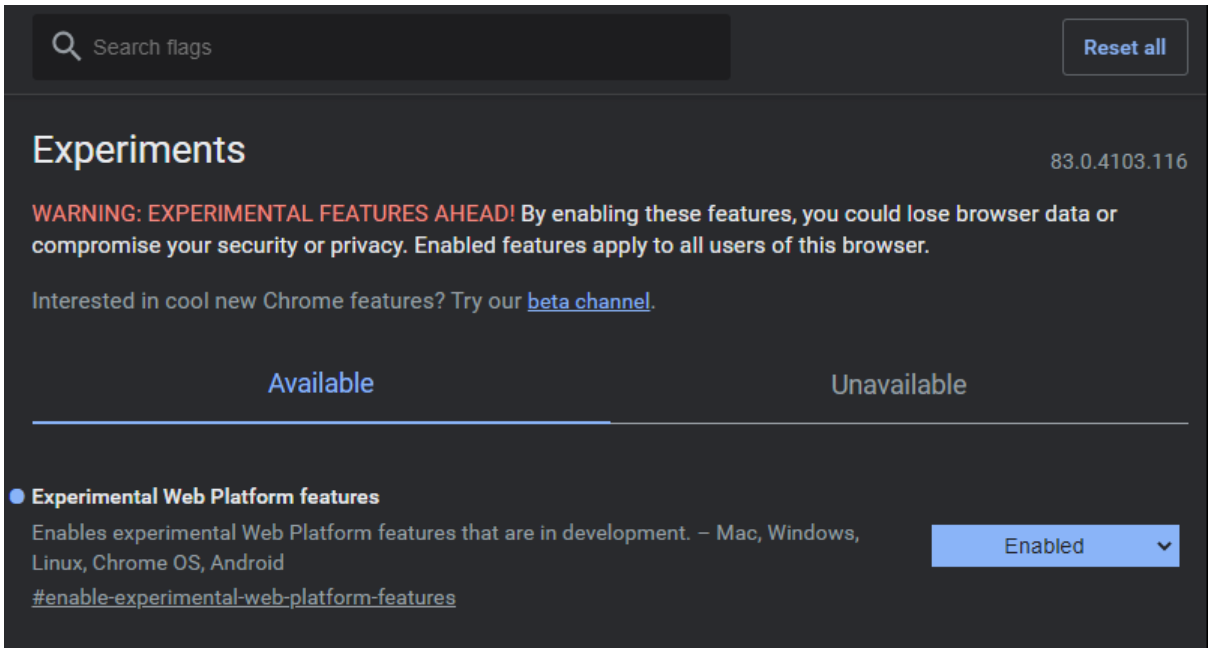


Figure 62. Chrome flag: Experimental Web Platform features

The second option is to enable the feature from the Web App by requesting an Origin Trial token. See Figure 63 for the suggested steps.

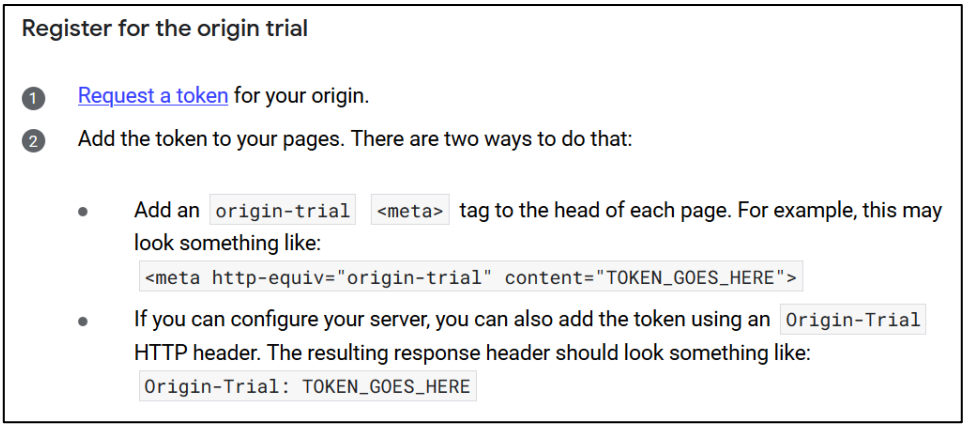


Figure 63. Registration process for Chrome Origin Trial token

After following the steps and getting the token. The meta tag was added in the _Layout.cshtml because it is the first part loaded in each page inside the Web App, including the Loyalty Card Page showing all the coffee images.

Thanks to the playground project developed in the initial sprints (see 4.1), the scanning function was easily added to the page.

```
45      <div>
46          <input class="btn btn-default btn-shadow buttonTopMargin" type="button" value="Start Scanning" onclick="startNFC()" />
47      </div>
```

Figure 64. Scan button HTML code

The Start Scanning button visible in the UI below the coffee images is a simple HTML code that calls a JavaScript function when clicked (see Figure 64).

```
75  async function startNFC() {
76      try {
77          const reader = new NDEFReader();
78
79          await reader.scan();
80          activateIcon();
81          console.log("> Scan started");
82
83          reader.addEventListener("error", (event) => {
84              alert("No valid NFC feature. " + event.message);
85          });
86
87          reader.addEventListener("reading", ({ message, serialNumber }) => {
88              console.log("serial number = " + serialNumber);
89              //document.getElementById("tagSR").value = serialNumber;
90
91              const decoder = new TextDecoder();
92
93              for (const record of message.records) {
94                  console.log(record.recordType);
95                  switch (record.recordType) {
96                      case "empty":
97                          document.getElementById("uniqueStoreCode").value = "NFC_TagEmpty";
98                          document.getElementById("tagSR").value = "NFC_EmptySR"; //Override for empty case
99                          document.getElementById("hiddenForm").submit(); // Test to see if demo can be made with other cards
100                         break;
101                     case "mime":
102                         if (record.mediaType === 'application/json') {
103                             const json = JSON.parse(decoder.decode(record.data));
104                             console.log(json);
105                             document.getElementById("tagSR").value = serialNumber;
106                             // Value to be taken from the NFC tag and passed to the back-end to be checked with the database
107                             document.getElementById("uniqueStoreCode").value = json.schemeCode;
108                             document.getElementById("hiddenForm").submit();
109                         }
110                         break;
111                     default:
112                         console.log("Something unexpected: " + record)
113                         break;
114                 }
115             }
116         });
117     } catch (error) {
118         console.log(error);
119     }
120 }
```

Figure 65. JavaScript main function to start scanning for NFC tags

The function called by the Start Scansion button is what makes the phone start scanning for tags.

Line 80 is calling a method to activate the floating effect (see Figure 66) to the first coffee image representing an empty Stamp (i.e. Yellow Coffee Image). This feature was added to give a visible feedback to the user that the phone was ready to scansion a tag.

```
122  function activateIcon() {
123      if (iconToFloat != null) {
124          iconToFloat.classList.add("iconFloatingEffect");
125      }
126  }
```

Figure 66. Function activateIcon for coffee image floating effect

All the line numbers in the following sections refers to the Figure 65.

When the user scans an NFC readable item such as smart cards and NFC tags, the Web App performs the function from line 87.

Line 96 case “empty” is implemented only for demonstration purposes. This case happens when the user scans most of the smart cards in circulation such as a student card, bank card or Oyster

cards. Because the API is not able to retrieve an NDEF message stored in the smart card, it behaves as there was none. This gave the opportunity to develop a broader interactive demonstration with all the stakeholders.

Line 101 case "mime" implementation is for the real case scenario. The code tries to parse the data stored in json format in the tag.

On line 107, the scheme code value saved in the NFC tag as a json key value, is retrieved and saved into a HTML tag element value with ID "uniqueStoreCode". Moreover, the serial number is taken from the tag and saved to another HTML element on line 105.

Both cases in the switch statement of the startNFC function are submitting values through a form which is supposed to be hidden.

```
56    <form asp-action="CollectStamp" asp-controller="Card" method="post" id="hiddenForm">
57        @Html.AntiForgeryToken()
58        <input name="StoreSchemeCode" class="form-control" type="hidden" value="DefaultValueStore" id="uniqueStoreCode" />
59        <input name="TagSR" class="form-control" type="hidden" value="DefaultValueTag" id="tagSR" />
60        <input type="submit" value="Mock Collection" class="btn btn-default btn-shadow" />
61    </form>
```

Figure 67. HTML code for the Hidden Form

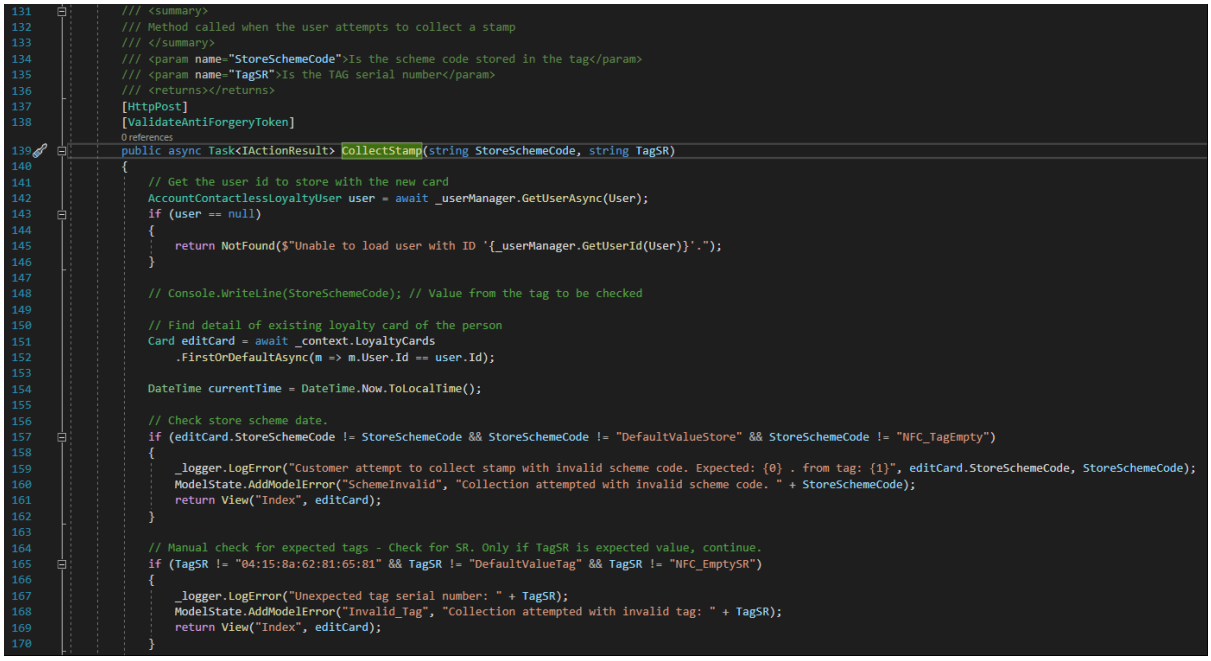
For demonstration purposes, this form has been left visible with a "Mock Collection" submit button which emulates the scansion of a valid NFC tag.

To note that on line 57 there is a request to create a "Forgery Token" input for the form in C#.

The hidden form is submitted when one of the following conditions happens:

- The user scans a valid NFC tag (line 101).
- The user scans a smart card (line 96).
- The user clicks on the "Mock Collection" button.

When the form is submitted, the Collect Stamp action in the CardController is called.



```
131         /// <summary>
132         /// Method called when the user attempts to collect a stamp
133         /// </summary>
134         /// <param name="StoreSchemeCode">Is the scheme code stored in the tag</param>
135         /// <param name="TagSR">Is the TAG serial number</param>
136         /// <returns></returns>
137         [HttpPost]
138         [ValidateAntiForgeryToken]
            0 references
139         public async Task<IActionResult> CollectStamp(string StoreSchemeCode, string TagSR)
140         {
141             // Get the user id to store with the new card
142             AccountContactlessLoyaltyUser user = await _userManager.GetUserAsync(User);
143             if (user == null)
144             {
145                 return NotFound($"Unable to load user with ID '{_userManager.GetUserId(User)}'.");
146             }
147
148             // Console.WriteLine(StoreSchemeCode); // Value from the tag to be checked
149
150             // Find detail of existing loyalty card of the person
151             Card editCard = await _context.LoyaltyCards
152                 .FirstOrDefaultAsync(m => m.User.Id == user.Id);
153
154             DateTime currentTime = DateTime.Now.ToLocalTime();
155
156             // Check store scheme date.
157             if (editCard.StoreSchemeCode != StoreSchemeCode && StoreSchemeCode != "DefaultValueStore" && StoreSchemeCode != "NFC_TagEmpty")
158             {
159                 _logger.LogError("Customer attempt to collect stamp with invalid scheme code. Expected: {0} . from tag: {1}", editCard.StoreSchemeCode, StoreSchemeCode);
160                 ModelState.AddModelError("SchemeInvalid", "Collection attempted with invalid scheme code. " + StoreSchemeCode);
161                 return View("Index", editCard);
162             }
163
164             // Manual check for expected tags - Check for SR. Only if TagSR is expected value, continue.
165             if (TagSR != "04:15:8a:62:81:65:81" && TagSR != "DefaultValueTag" && TagSR != "NFC_EmptySR")
166             {
167                 _logger.LogError("Unexpected tag serial number: " + TagSR);
168                 ModelState.AddModelError("Invalid_Tag", "Collection attempted with invalid tag: " + TagSR);
169                 return View("Index", editCard);
170             }
```

Figure 68. CollectStamp action in the CardController (pt.1)

Follows the analysis of the CollectStamp method in Figure 68. Therefore, all the line numbers mentioned in this section are referring to the figure above.

On line 137 there is property of the method that states that the request must be of type "HttpPost".

On line 138 there is another property that validates the "Forgery Token" submitted along with the form. This code is added to make sure the request is legitimate.

On line 157 there is a validation check for the scheme code received from the CardController. If it is not any of the expected cases it will throw an error.

On line 165 there is a validation check for the serial numbers expected. At the moment there is only one serial tag number that has been used for testing during the development process. In the future it should be taken as a value from a database storing all the tags issued by Contactless Loyalty.

```
172         // Check for valid collection rate. Customer can collect depending on the key value in the app settings
173         if (IsTimeValid(currentTime, editCard.LastStampDateTime, _configuration.GetValue<string>("CustomSettings:CollectionRate")))
174         {
175             editCard.LastStampDateTime = currentTime;
176             editCard.NumberOfStamps++;
177         }
178         else
179         {
180             _logger.LogError("Customer attempt to collect stamp on invalid date. Last stamp: {0}. Collection Rate:{1}", editCard.LastStampDateTime,
181             ModelState.AddModelError("RateInvalid", "Collection attempted on invalid day.");
182             return View("Index", editCard);
183         }
184
185         // The following case is to prevent user from collecting more stamps before collecting the voucher.
186         if (editCard.NumberOfStamps > (int)SchemeLimit.WembleyEmporium)
187         {
188             _logger.LogWarning("User {0} attempt to collect new stamp but reached limit {1}", user.Id, (int)SchemeLimit.WembleyEmporium);
189             editCard.NumberOfStamps--; // Remove stamp collected over limit
190         }
191
192         _context.Update(editCard);
193         try
194         {
195             await _context.SaveChangesAsync();
196         }
197         catch (Exception error)
198         {
199             Console.WriteLine(error);
200         }
201
202         return RedirectToAction("Index", "Card");
```

Figure 69. CollectStamp action in the CardController (pt.2)

The last validation for the collection is a date and time check. On line 173 in Figure 69, the time when the collection is attempted and the time of the last stamp are checked to see if the user is allowed by the scheme collection rate (e.g. daily or weekly).

After all successful validation, the page is refreshed and the updated data is displayed.

6.2.7 i-movo API request

```
248         using (HttpClient client = new HttpClient())
249         {
250             client.BaseAddress = new Uri(apiRequestUrl);
251             client.DefaultRequestHeaders.Accept.Clear();
252             client.DefaultRequestHeaders.Accept.Add(new System.Net.Http.Headers.MediaTypeWithQualityHeaderValue("application/json"));
253
254             HttpResponseMessage response = client.GetAsync(apiRequestUrl).Result; // Adding Result make the call to synchronous
255             string content = response.Content.ReadAsStringAsync().Result;
256
257             if (response.IsSuccessStatusCode)
258             {
259                 _logger.LogInformation("API call made for receiveSMS.");
260                 return true;
261             }
262         }
```

Figure 70. i-movo API voucher request call

The above figure (see Figure 70) shows a section of the method used for making the API request to send a voucher to the customer. For privacy and security reasons the variables are not shown because containing sensitive information.

This is a basic Http API request that can be found in the Microsoft documentation (https://docs.microsoft.com/en-us/aspnet/web-api/overview/advanced/calling-a-web-api-from-a-net-client).

6.2.8 Hidden Page

There is a hidden page that allows users to write to an NFC tag a value entered in a text field. This was initially implemented in the playground process for the purpose of enabling stakeholders to write their own tag values. The page is only available for user logged in the Web App that edit the URL to end with "/Card/Write".

6.3 Deployment

To make the Web App accessible to the public a cloud server was chosen as deployment target. The phases for the publication of the Web App on the Internet are explained in the next sections.

6.3.1 Microsoft Azure – Web Application Setup

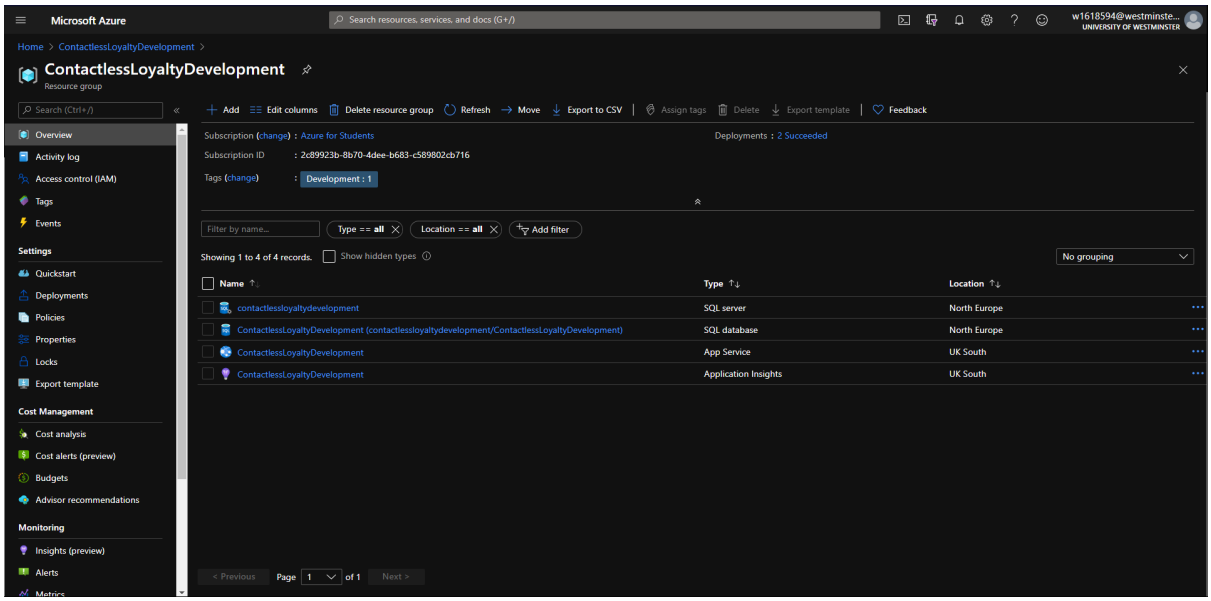


Figure 71. ContactlessLoyaltyDevelopment Resource Group on Azure Portal

Using the credentials provided by the University of Westminster, a Microsoft Azure account was created. On the Azure portal (https://portal.azure.com/) the appropriate elements (see Figure 71) were created after following the Microsoft documentation.

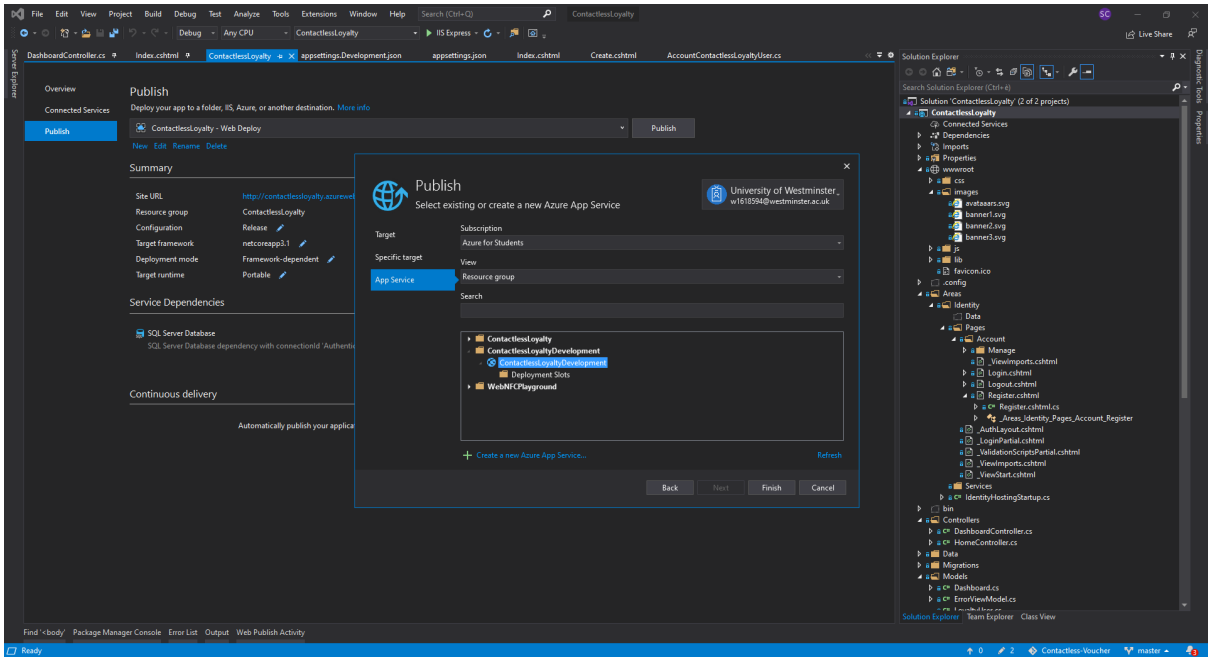


Figure 72. Publish Profile creation

On Visual Studio a corresponding "Publish Profile" was created to deploy the Web App to the cloud server.

6.3.2 Database Setup

To provide the database of the Web App a server and database elements were also created on the Azure Portal (see Figure 71).

MSSQL was used to generate a script from the existing local database.

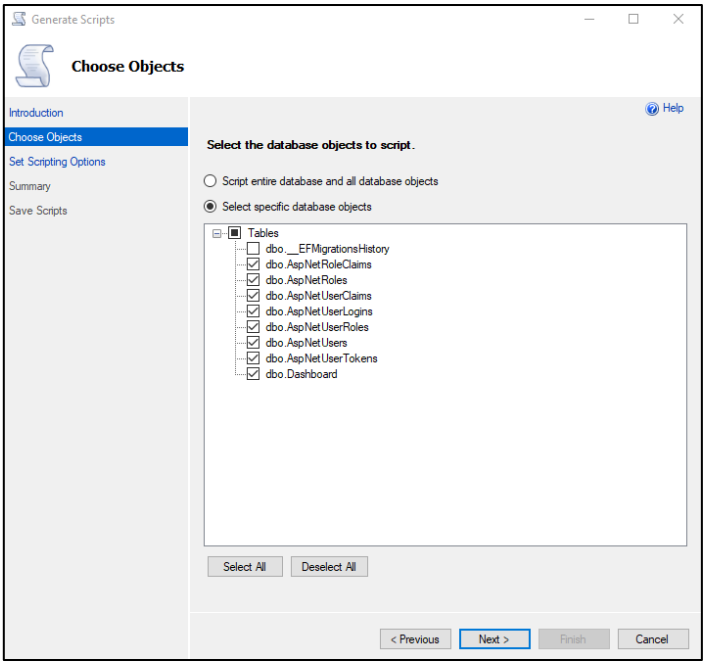


Figure 73. Script Generation Wizard on MSSQL

The generated script created all the tables and relationship automatically like an exact copy of the local database.

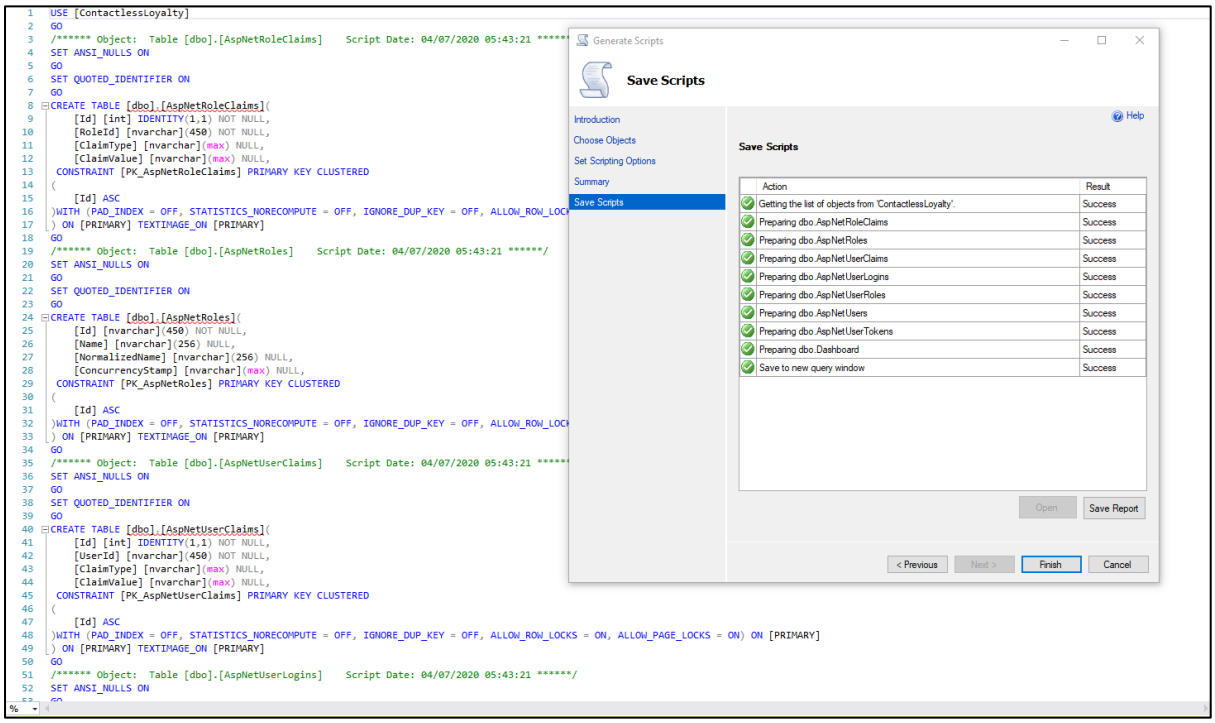


Figure 74. Generated Query from creating the Database table

After this process, the live environment reflected the same structure of the local Web App.

7. Testing

An important part of the development of a software is the testing. As Toby B. (COO at i-movo) would say to all the new developers in the company "The rule number 1 is: test everything!".

Therefore, this section provides the explanation on how testing was carried on during the process of this project development.

7.1 Functional Testing

Test Case ID		User_001	Test Case Description	Test the Registration Functionality	
S #	Prerequisites:			S #	Test Data
1	Access to Chrome Browser			1	First Name = Shouyi
2				2	Last Name = Test
3				3	Phone Number = 07708053895
4				4	Password = Test123!
Test Scenario	Validation on all values entered, password to meet requirements the customer can register and login automatically				
Step #	Step Details		Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Navigate to https://contactlessloyaltydevelopment.azurewebsites.net/		Site should open	As Expected	Pass
2	First and Last name		Credential can be entered	As Expected	Pass
3	Mobile Phone number		Number Regex is checked	As Expected	Pass
4	Password and Password Confirmation		Must make validation according to UC0 (Use case 0)	As Expected	Pass
5	User consent to Terms and Condition		User can check the box	As Expected	Pass
6	Click Submit		Cutomer is registered and logged in	As Expected	Pass

Figure 75. Registration Test

Test Case ID		User_002	Test Case Description	Test the Login Functionality	
S #	Prerequisites:			S #	Test Data
1	Access to Chrome Browser			1	Phone Number = 07708053895
2				2	Password = Test123!
3				3	
4				4	
Test Scena	Verify on entering valid phone number and password, the customer can login				
Step #	Step Details		Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Navigate to https://contactlessloyaltydevelopment.azurewebsites.net/		Site should open	As Expected	Pass
2	Enter Phone number & Password		Credential can be entered	As Expected	Pass
3	Click Submit		Cutomer is logged in	As Expected	Pass

Figure 76. Login Test

Test Case ID		User_003	Test Case Description	Test the Change Phone Number Functionality	
S #	Prerequisites:			S #	Test Data
1	Access to Chrome Browser			1	Phone Number = 07708053896
2	Navigate to https://contactlessloyaltydevelopment.azurewebsites.net/			2	
3	Successfully Sign In			3	
4				4	
Test Scenario	Verify that phone number can be modified				
Step #	Step Details		Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Manage Account		Manage Account open with Change phone number	As Expected	Pass
2	Enter Phone New Phone Number		Credential can be entered	As Expected	Pass
3	Click Save		Cutomer update phone number	As Expected	Pass

Figure 77. Change Mobile Phone Number Test

Test Case ID	User_004	Test Case Description	Test Change Password Functionality	
S #	Prerequisites:		S #	Test Data
1	Access to Chrome Browser		1	Current Password = Test123!
2	Navigate to https://contactlessloyaltydevelopment.azurewebsites.net/		2	Confirm new Password = Test124!
3	Successfully Sign In		3	Confirm new Password = Test124!
Test Scena	Verify that password can be modified			
Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Manage Account	Manage Account open with Change phone number	As Expected	Pass
2	Click on Change Password Section	Credential can be entered	As Expected	Pass
3	Enter New Password	Credential can be entered	As Expected	Pass
4	Enter Current Password	Credential can be entered	As Expected	Pass
5	Click Save	Password is updated	As Expected	Pass

Figure 78. Change Password Test

Test Case ID	Stamp_001	Test Case Description	Test theStamp Collection method 1	
S #	Prerequisites:		S #	Test Data
1	Access to Chrome Browser		1	Scansion test NFC tag
2	Navigate to https://contactlessloyaltydevelopment.azurewebsites.net/		2	
3	Successfully Sign In		3	
4	See existing Loyalty Card		4	
Test Scenario	Verify that on valid NFC tag scan after clicking scan button and validation satisfied, the page is updated with new loyalty stamp			
Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	See Loyalty Card	Loyalty Card is loaded	As Expected	Pass
2	Click Start Scanning	Button can be clicked	As Expected	Pass
3	NFC tag has been scanned	Page is refreshed with new details	As Expected	Pass

Figure 79. Stamp Collection by NFC Test

Test Case ID	Stamp_002	Test Case Description	Test theStamp Collection method 2	
S #	Prerequisites:		S #	Test Data
1	Access to Chrome Browser		1	Scansion test smart card
2	Navigate to https://contactlessloyaltydevelopment.azurewebsites.net/		2	
3	Successfully Sign In		3	
4	See existing Loyalty Card		4	
Test Scena	Verify that on valid smart card scan after clicking scan button and validation satisfied, the page is updated with new loyalty stamp			
Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	See Loyalty Card	Loyalty Card is loaded	As Expected	Pass
2	Click Start Scanning	Button can be clicked	As Expected	Pass
3	Smart card has been scanned	Page is refreshed with new details	As Expected	Pass

Figure 80. Stamp Collection by Smart Card Test

Test Case ID	Stamp_003	Test Case Description	Test theStamp Collection method 3	
S #	Prerequisites:		S #	Test Data
1	Access to Chrome Browser		1	Click Mock Collection button
2	Navigate to https://contactlessloyaltydevelopment.azurewebsites.net/		2	
3	Successfully Sign In		3	
4	See existing Loyalty Card		4	
Test Scenario	Verify that on mock collection button clicked and validation satisfied, the page is updated with new loyalty stamp			
Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	See Loyalty Card	Loyalty Card is loaded	As Expected	Pass
2	Click Mock Collection	Button can be clicked	As Expected	Pass
3	Smart card has been scanned	Page is refreshed with new details	As Expected	Pass

Figure 81. Stamp Collection by Mock Collection Button Test

7.2 User Testing

To gain more understanding of exploits of the system, the stakeholders like potential customers were contacted to test the Web Application.

During an interview and demo with a stakeholder, it has been discovered that also Brave browser on mobile works with this Web-NFC feature. This is most likely because Brave is based upon the free open source Chromium browser which is also the base for Google Chrome.

After taking notes of the comments and ideas of the stakeholders involved in the development, there are a number of features that have been added in.

The improvements that have been included thanks to the user testing are:

1. Update the phone number can be done only if the number the user is changing to, is not already being used or saved in the database. This is to avoid duplication of user identities.
2. Changing the password is possible only if the user old and new password are different. Otherwise do not change the data because there is no actual change. This saves a request of data editing in the database.
3. Floating speed of the coffee image in the Loyalty Card Page has been slowed down.

Moreover, further details have been added in the first page after login to give the user a guidance to use the NFC feature on the Web App.

8. Conclusions and Reflections

This chapter aims to point out the outcomes of this project. The current state is going to be compared to the aims and requirements initially declared. Furthermore, a brief consideration of future work is included.

8.1 Current State of the Project

Contactless Loyalty is a deployed and live Web Application publicly accessible by anyone who knows the URL: https://contactlessloyaltydevelopment.azurewebsites.net/.

One of the aims of the project was to deliver an easy solution to collect stamps without the need of installing native application on the phone, this was achieved through the deployment of the Contactless Loyalty Web App on the Azure Cloud Server.

The features related to the NFC are enough to engage with customers and retailers to start making a real user experience and analyse the outcome. It is possible, through some change of the hardcoded values in the code to perform what is considered to be the retailer freedom of choice for the loyalty scheme. A proposed solution would be to place the NFC tags in different areas of a big clothes store. The NFC tags should be placed in areas where the shop owner or retailer wants the customer to visit because of the new merchandise arrived in the shop.

The customer would need to engage, by tapping the phone on the NFC tag, to receive a stamp or point on the Web App. This would mean that the customer has seen the new merchandise in the shop. This cover the aspect of brand awareness mentioned in the requirements (see 3.2.3).

8.2 Limitations

Since the Web-NFC API is trial feature on Chrome and still on development, it was hard to implement a valid application for all devices. A big limitation was the impossibility to test the features on iOS devices as they represent half of the market of this project. Although, it was already a big opportunity to work on this capability because it was released only this year.

A drawback for this technology, when compared to the mobile native application option, was the lack of HCE capability. If there was a way to emulate the existing loyalty cards and other smart cards through the Web App, there would have been a bigger impact.

8.3 Improvements and Future Work

All the work carried out during the project development have made possible to see that there is a potential demand for this type of product. A first improvement to do regarding this project would be an improved design of the database architecture to allow to record more details about the card. For example, an historic record of stamp collection could be used to target a customer with special promotion during specific time of the month. Another table for storing the serial

numbers with correspondent loyalty scheme would be needed to facilitate the automation of the NFC tag setup.

This software can also be improved if the future capabilities of the Web-NFC API are going to be similar to those available for mobile native applications.

8.4 Closing Remarks

The various stage of this software development lifecycle has increased my knowledge on all the different aspects that delivers a fully working application.

It has definitely improved my capabilities of documentation and research skills. The discovery made about the physics and the history of the technology was a fascinating subject to cover. The curiosity about this technology has been fed; enough to encourage me to use this technology more often in the future. Ultimately, I consider this piece of work the best representation of the experience and knowledge I have gained throughout the three years of study at the University of Westminster and the two years of experience gained working with i-movo.

9. References and Bibliography

Alexander, I. F. & Beus-Dukic, L., 2009. Discovering Stakeholders. In: J. Wiley, ed. *Discovering Requirements: How to Specify Products and Services.* Chichester: John Wiley & Sons, p. 478.

Anderson, R. & Nowak, R., 2020. *What Is Razor Pages?.* [Online]
Available at: https://docs.microsoft.com/en-us/aspnet/core/razor-pages/?view=aspnetcore-3.1&tabs=visual-studio
[Accessed 10 June 2020].

Apple, 2020. *About limits when using Apple Pay in shops.* [Online]
Available at: https://support.apple.com/en-gb/HT207435
[Accessed 27 June 2020].

Basu, S., 2018. *Alipay to revise loyalty point rules starting April 1.* [Online]
Available at: https://medium.com/@techgenyz/using-alipay-makes-you-eligible-for-alipay-loyalty-points-which-may-be-used-to-rank-up-or-cashed-6aaae323dbe9
[Accessed 28 June 2020].

Beaufort, F. & Kenneth, R. C., 2020. *Web NFC explained.* [Online]
Available at: https://github.com/w3c/web-nfc/blob/gh-pages/EXPLAINER.md
[Accessed 26 May 2020].

Bhaumik, R., Han, L. & Wu, D., 2020. *W3C Web NFC API implementation in Chromium.* [Online]
Available at: https://www.chromium.org/developers/design-documents/web-nfc
[Accessed 25 March 2020].

BlueBite, 2020. *iPhone NFC Compatibility.* [Online]
Available at: https://www.bluebite.com/nfc/iphone-nfc-compatibility
[Accessed 6 July 2020].

Borak, M., 2019. *QR code payments make long commutes even longer in China.* [Online]
Available at: https://www.scmp.com/abacus/culture/article/3021409/qr-code-payments-make-long-commutes-even-longer-china
[Accessed 28 June 2020].

Caffè Nero, 2019. *FAQ - Stamps & Loyalty Cards.* [Online]
Available at: https://caffenero-yoyowallet.zendesk.com/hc/en-us/articles/115002860965-Stamps-Loyalty-Cards-
[Accessed 1 July 2020].

Cambridge Dictionary, 2020. *Definition of stakeholder.* [Online]
Available at: https://dictionary.cambridge.org/dictionary/english/stakeholder
[Accessed 7 July 2020].

Chrome Origin Trials, 2020. *Trial for Web NFC.* [Online]
Available at: https://developers.chrome.com/origintrials/#/view_trial/236438980436951041
[Accessed 25 June 2020].

Crypto Museum, 2015. *The Thing - Great Seal Bug.* [Online]
Available at: https://www.cryptomuseum.com/covert/bugs/thing/index.htm#ref_8
[Accessed 11 June 2020].

DevTeam.Space, 2020. *Top 10 Loyalty Apps of 2020.* [Online]
Available at: https://www.devteam.space/blog/top-10-loyalty-apps/
[Accessed 06 June 2020].

Embargo Lifestyle Limited, 2020. *Terms and Conditions of Use.* London: s.n.

Fluke Corporation, 2020. *What is a capacitance?.* [Online]
Available at: https://www.fluke.com/en-gb/learn/best-practices/measurement-basics/electricity/what-is-capacitance
[Accessed 10 June 2020].

Frew, J., 2020. *How to Use NFC: 7 NFC Uses That'll Impress Your Friends.* [Online]
Available at: https://www.makeuseof.com/tag/9-awesome-ways-use-nfc-thatll-impress-friends/
[Accessed 5 July 2020].

Haggerty, K., 2020. *ASP.NET Core 3.1 - Password Hasher.* [Online]
Available at: https://kenhaggerty.com/articles/article/aspnet-core-31-password-hasher
[Accessed 8 July 2020].

Haines, E., 2020. *5 Things You Need to Know About Beacon Technology.* [Online]
Available at: https://www.wordstream.com/blog/ws/2018/10/04/beacon-technology
[Accessed 2 July 2020].

Harford, T., 2019. *The Cold War spy technology which we all use.* [Online]
Available at: https://www.bbc.co.uk/news/business-48859331
[Accessed 10 June 2020].

Hatton, S., 2008. Choosing the "Right" Prioritisation Method. In: I. C. Society, ed. *ASWEC '08: Proceedings of the 19th Australian Conference on Software Engineering.* Washington DC: IEEE Computer Society, pp. 517-526.

Heggestuen, J., 2014. *Alipay Overtakes PayPal As The Largest Mobile Payments Platform In The World.* [Online]
Available at: https://www.businessinsider.com/alipay-overtakes-paypal-as-the-largest-mobile-payments-platform-in-the-world-2014-2?r=US&IR=T
[Accessed 30 June 2020].

Henriot, R., 2019. *New Loyalty: 5 types of rewardable behaviors on WeChat..* [Online]
Available at: https://blog.splio.com/zh/new-loyalty-5-types-of-rewardable-behaviors-on-wechat
[Accessed 27 June 2020].

Hill, S., 2020. *What is NFC? Here's everything you need to know.* [Online]
Available at: https://www.digitaltrends.com/mobile/what-is-nfc/
[Accessed 27 June 2020].

Huth, J., 2018. *Science Sounds Strange* [Interview] (8 November 2018).

Igoe, T., Jepson, B. & Coleman, D., 2014. Chapter 2. NFC and RFID. In: R. Roumeliotis, A. MacDonald, N. Shelby & J. Kwityn, eds. *Beginning NFC.* 2014 ed. Sebastopol(California): O'Reilly Media Inc., p. 256.

intersoft consulting, 2016. *Art.15 GDPR - Right of access by the data subject.* [Online]
Available at: https://gdpr-info.eu/art-15-gdpr/
[Accessed 10 July 2020].

Jovancic, N., 2019. *LeadQuizzes.* [Online]
Available at: https://www.leadquizzes.com/blog/7-examples-of-customer-loyalty-programs/
[Accessed 5 June 2020].

Kenneth Research, 2020. *Near Field Communication Market Analysis, Size, Share, Growth, Trends and Forecast to 2025.* [Online]
Available at: https://www.marketwatch.com/press-release/near-field-communication-market-analysis-size-share-growth-trends-and-forecast-to-2025-2020-05-01?mod=mw_quote_news
[Accessed 25 June 2020].

Kostiainen, A., 2019. *Web NFC reaches a key milestone.* [Online]
Available at: https://www.w3.org/community/web-nfc/2019/12/17/web-nfc-reaches-a-key-milestone/
[Accessed 25 June 2020].

Long, H., 2015. *https://edition.cnn.com/business.* [Online]
Available at: https://money.cnn.com/2015/08/10/investing/gift-cards-soar-in-popularity/
[Accessed November 2019].

loveMoney, 2019. *Contactless payment security, concerns and considerations.* [Online]
Available at: https://www.lovemoney.com/guides/75138/contactless-card-payment-security-concerns-considerations-safety-fraud
[Accessed 11 June 2020].

Lowry Solutions, 2014. *What Are the Different Types of RFID Technology?.* [Online]
Available at: https://lowrysolutions.com/blog/what-are-the-different-types-of-rfid-technology/
[Accessed 21 June 2020].

M. Mahalakshmi, D. S., 2013. Traditional SDLC Vs Scrum Methodology –A Comparative Study. *International Journal of Emerging Technology and Advanced Engineering,* 3(6), p. 5.

Microsoft, 2020. *What is ASP.NET CORE?.* [Online]
Available at: https://dotnet.microsoft.com/learn/aspnet/what-is-aspnet-core
[Accessed 5 July 2020].

Milis, B., 2019. *The Pantone 2020 Color of the Year is a Calm, Enduring Refuge.* [Online]
Available at: https://theblog.adobe.com/pantone-color-of-the-year-2020/
[Accessed 26 May 2020].

Mozilla Web Docs, 2020. *What is JavaScript?.* [Online]
Available at: https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First_steps/What_is_JavaScript
[Accessed 11 July 2020].

NextPoints, 2020. *How much does an RFID tag cost.* [Online]
Available at: https://nextpoints.com/en/rfid-blog/how-much-does-rfid-tag-cost/
[Accessed 5 July 2020].

Nieva, R. & Bennett, B., 2018. *Google merges payment platforms under Google Pay brand.* [Online]
Available at: https://www.cnet.com/news/google-launches-google-pay-mobile-payment-service/
[Accessed 26 June 2020].

Popper, B., 2015. *Google introduces Android Pay, a replacement for its wallet app on mobile.* [Online]
Available at: https://www.theverge.com/2015/5/28/8661867/google-introduces-android-pay-replace-wallet-app
[Accessed 27 June 2020].

Ratna, S., 2020. *Buy beacons: 7 things you need to know before buying beacons.* [Online]
Available at: https://blog.beaconstac.com/2018/08/things-you-need-to-know-before-you-buy-beacons/
[Accessed 2 July 2020].

Revolut, 2020. *Is There a Google Pay Limit?.* [Online]
Available at: https://blog.revolut.com/google-pay-limit/
[Accessed 27 June 2020].

Robertson, T., 2018. *Digital vs. Card Based Customer Loyalty Programs: Which is Right for You.* [Online]
Available at: https://getsparkage.com/blog/2018/09/18/https-getsparkage-com-blog-2018-09-18-digital-vs-card-based-customer-loyalty-programs-which-is-right-for-you/
[Accessed 10 July 2020].

Rouse, M., 2007. *RFID (radio frequency identification).* [Online]
Available at: https://internetofthingsagenda.techtarget.com/definition/RFID-radio-frequency-identification
[Accessed 15 June 2020].

Savvides, L. & Orellana, V. H., 2019. *Apple Pay vs. Google Pay vs. Samsung Pay: Mobile payments compared.* [Online]
Available at: https://www.cnet.com/news/apple-pay-google-pay-samsung-pay-best-mobile-

payment-system-compared-nfc/
[Accessed 26 June 2020].

Seritag, 2020. *NFC Tags Explained.* [Online]
Available at: https://seritag.com/learn/using-nfc/nfc-tags-explained
[Accessed 10 July 2020].

Smith, C., 2020. *PayPal is making it even easier to never handle cash again.* [Online]
Available at: https://www.trustedreviews.com/news/paypal-touch-free-contactless-qr-code-4032924
[Accessed 29 June 2020].

Tesco, 2020. *FAQs.* [Online]
Available at: https://pay-plus.tesco.com/FAQs#g-q1
[Accessed 2 July 2020].

Thorp-Lancaster, D., 2019. *Microsoft Wallet for Windows Phone to be retired in February.* [Online]
Available at: https://www.windowscentral.com/microsoft-wallet-windows-phone-be-retired-february
[Accessed 28 June 2020].

Velarde, O., 2019. *What Is the Pantone Color of the Year and Why Is It Important?.* [Online]
Available at: https://visme.co/blog/pantone-color-of-the-year/
[Accessed 10 May 2020].

Vijayasarathy, L. R. & Turk, D., 2008. Agile software development: a survey of early adopters. *Journal of Information Technology Management,* XIX(2), p. 8.

Whitwam, R., 2020. *Google Pay vs. Samsung Pay: Which tap to pay system is best?.* [Online]
Available at: https://www.androidpolice.com/2020/05/09/google-pay-vs-samsung-pay-which-mobile-payment-provider-should-you-use/
[Accessed 28 June 2020].

Yang, A. & Hancke, G. P., 2017. RFID and Contactless Technlogy. In: K. Mayes & K. Markantonakis, eds. *Smart Cards, Tokens, Security and Applications.* London: Springer Nature, p. 530.

Yu Beng Leau, W. K. L. W. Y. T. S. F. T., 2012. Software Development Life Cycle AGILE vs Traditional Approaches. *2012 International Conference on Information and Network Technology (ICINT 2012),* 37(37), p. 6.

10. Appendix

10.1 Video Demo

This the link for the video demonstration of the Web App: https://youtu.be/IBRZFg2auvA

10.2 Web NFC Community Group

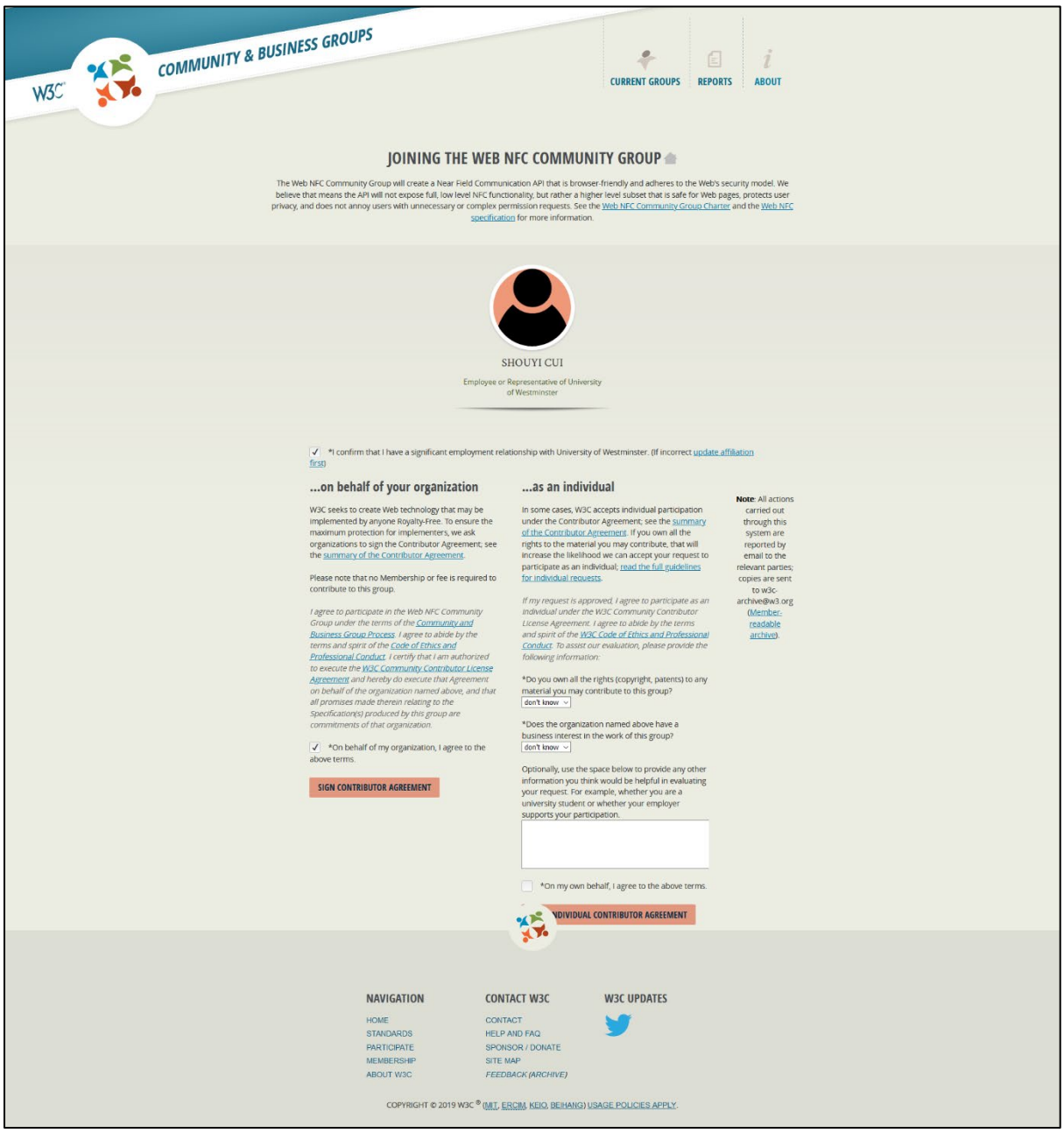


Figure 82. Registration into the Web NFC Community Group

10.3 Paymo

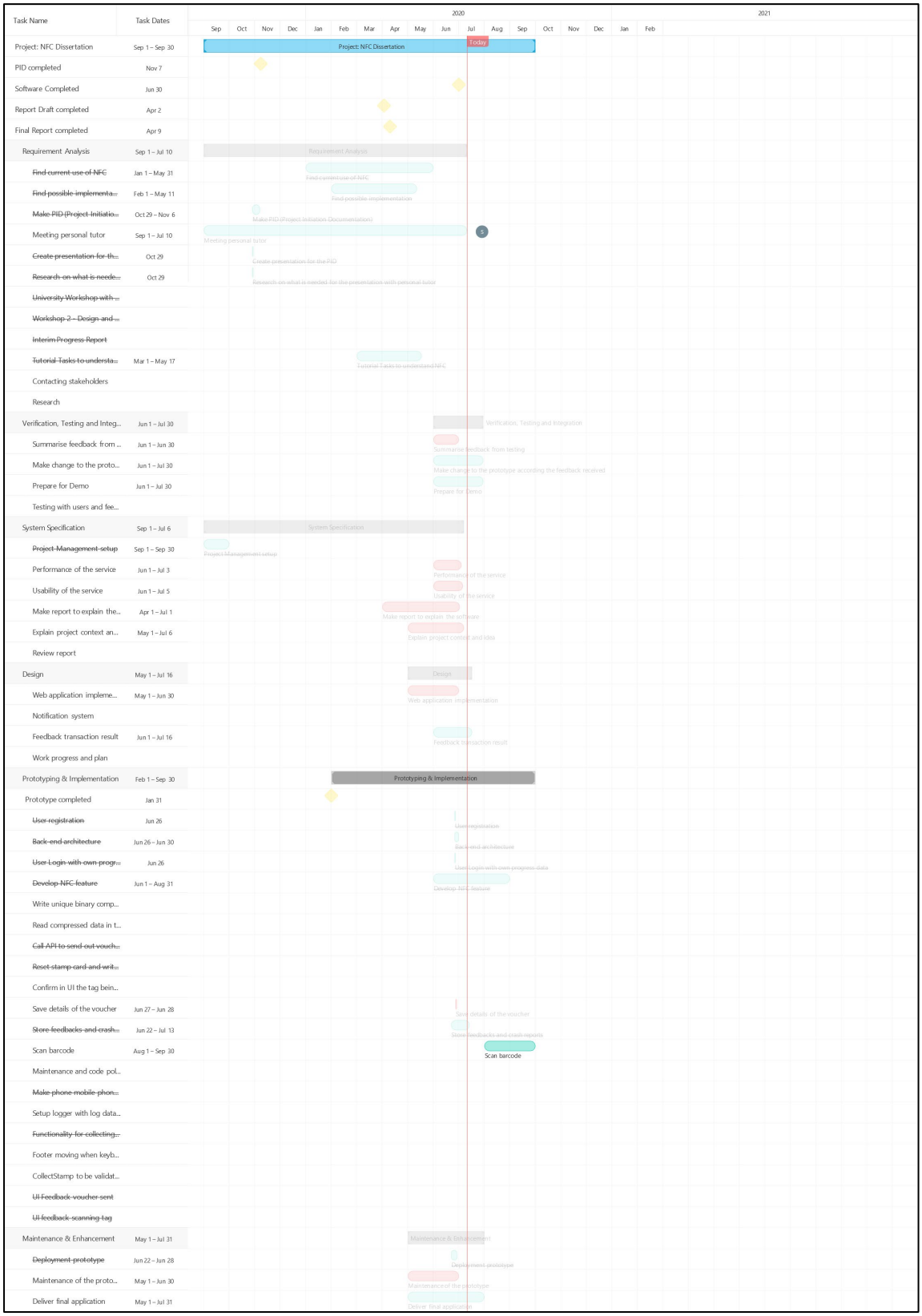


Figure 83. Gantt Chart produced based upon the task saved on Paymo

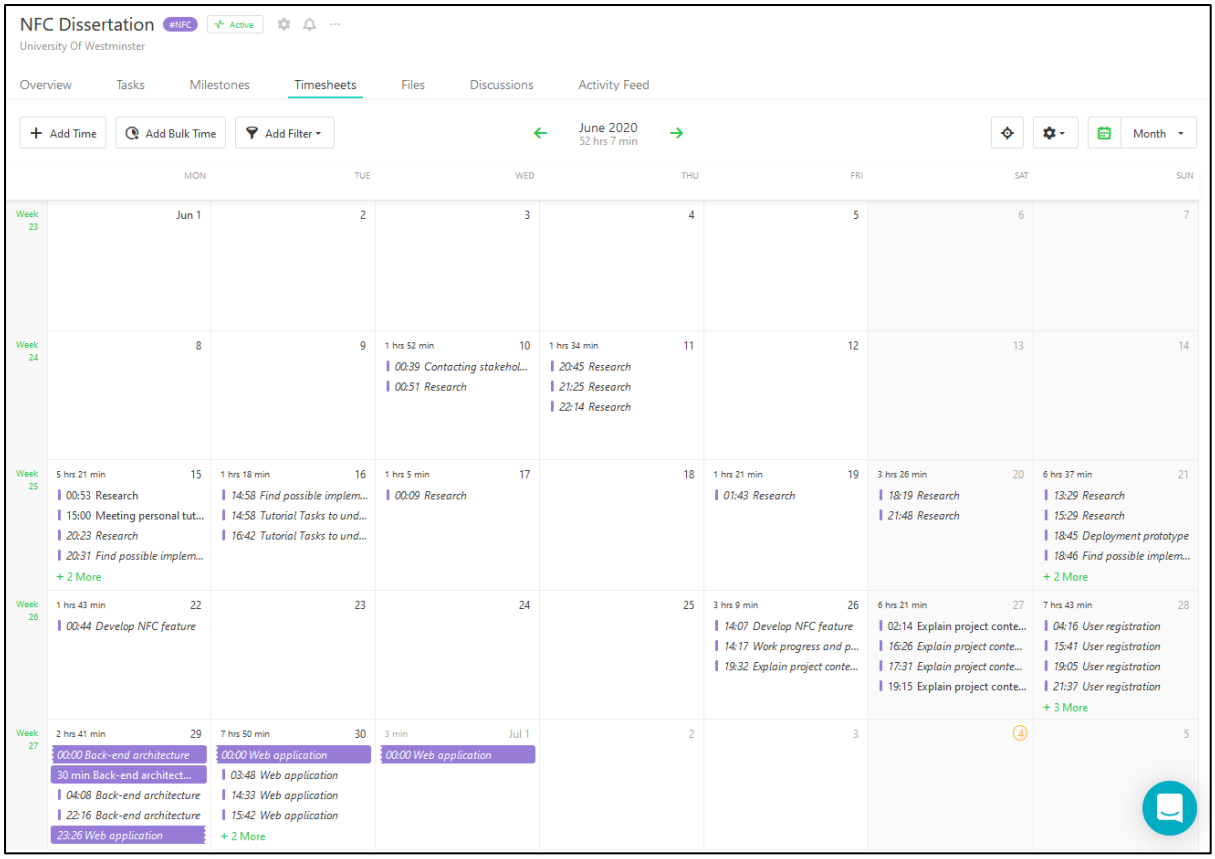
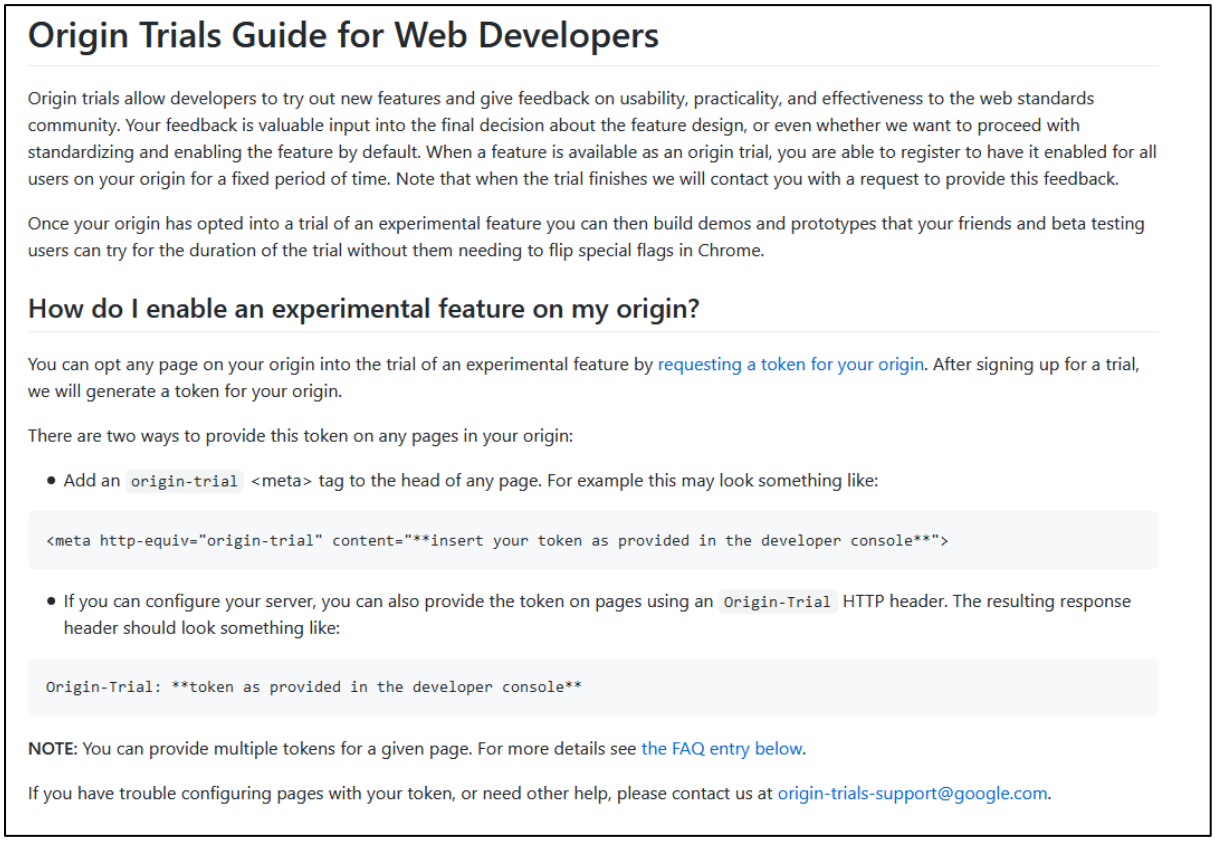


Figure 84. Example of Timesheet provided by Paymo for June

10.4 Chrome Origin Trials



Origin Trials Guide for Web Developers

Origin trials allow developers to try out new features and give feedback on usability, practicality, and effectiveness to the web standards community. Your feedback is valuable input into the final decision about the feature design, or even whether we want to proceed with standardizing and enabling the feature by default. When a feature is available as an origin trial, you are able to register to have it enabled for all users on your origin for a fixed period of time. Note that when the trial finishes we will contact you with a request to provide this feedback.

Once your origin has opted into a trial of an experimental feature you can then build demos and prototypes that your friends and beta testing users can try for the duration of the trial without them needing to flip special flags in Chrome.

How do I enable an experimental feature on my origin?

You can opt any page on your origin into the trial of an experimental feature by requesting a token for your origin. After signing up for a trial, we will generate a token for your origin.

There are two ways to provide this token on any pages in your origin:

- Add an `origin-trial` <meta> tag to the head of any page. For example this may look something like:

```
<meta http-equiv="origin-trial" content="**insert your token as provided in the developer console**">
```

- If you can configure your server, you can also provide the token on pages using an `Origin-Trial` HTTP header. The resulting response header should look something like:

```
Origin-Trial: **token as provided in the developer console**
```

NOTE: You can provide multiple tokens for a given page. For more details see the FAQ entry below.

If you have trouble configuring pages with your token, or need other help, please contact us at origin-trials-support@google.com.

Figure 85. Explanation of Origin Trials

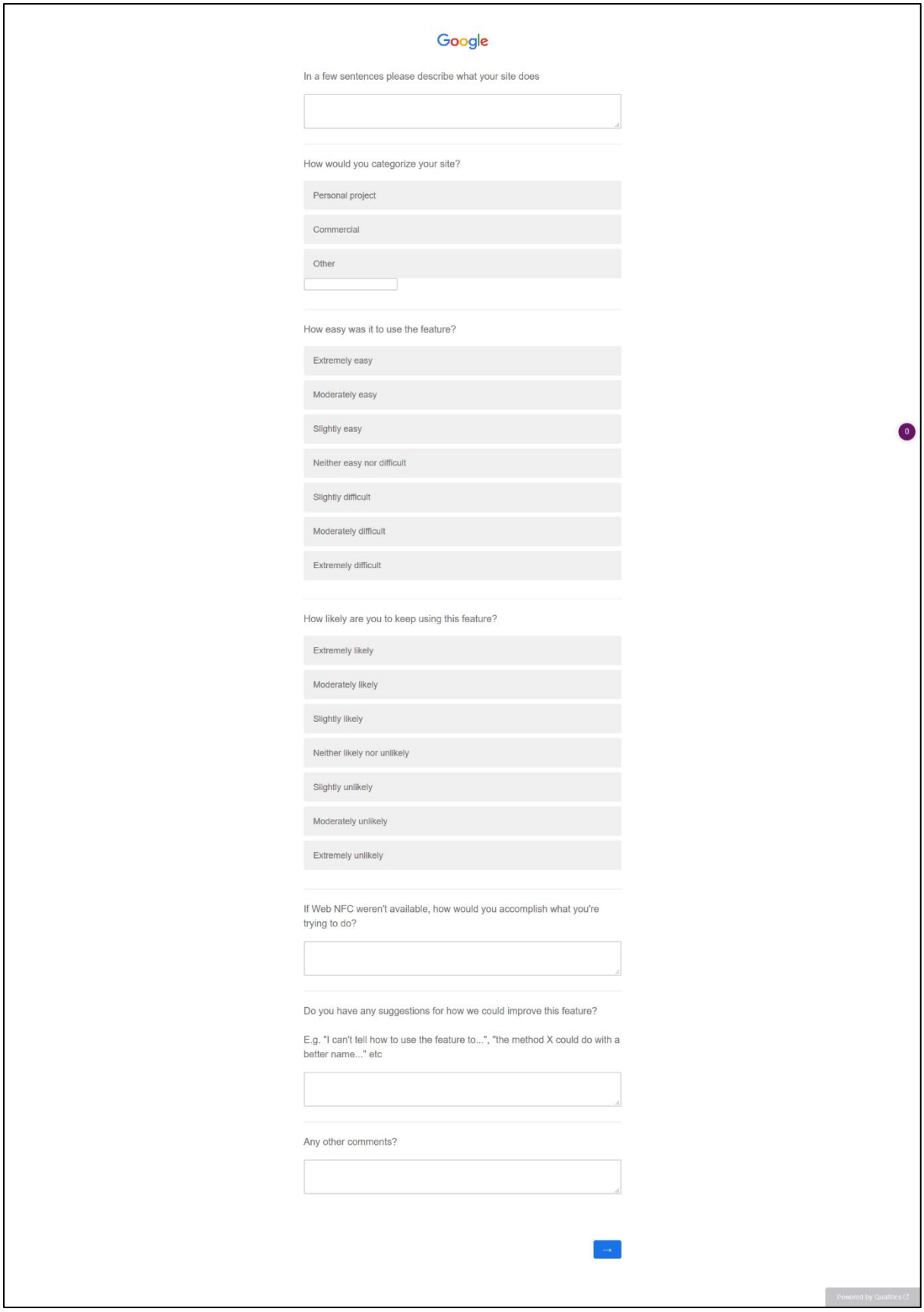


Figure 86. Feedback form to be filled at the end of the project

10.5 Feedback on Project Report

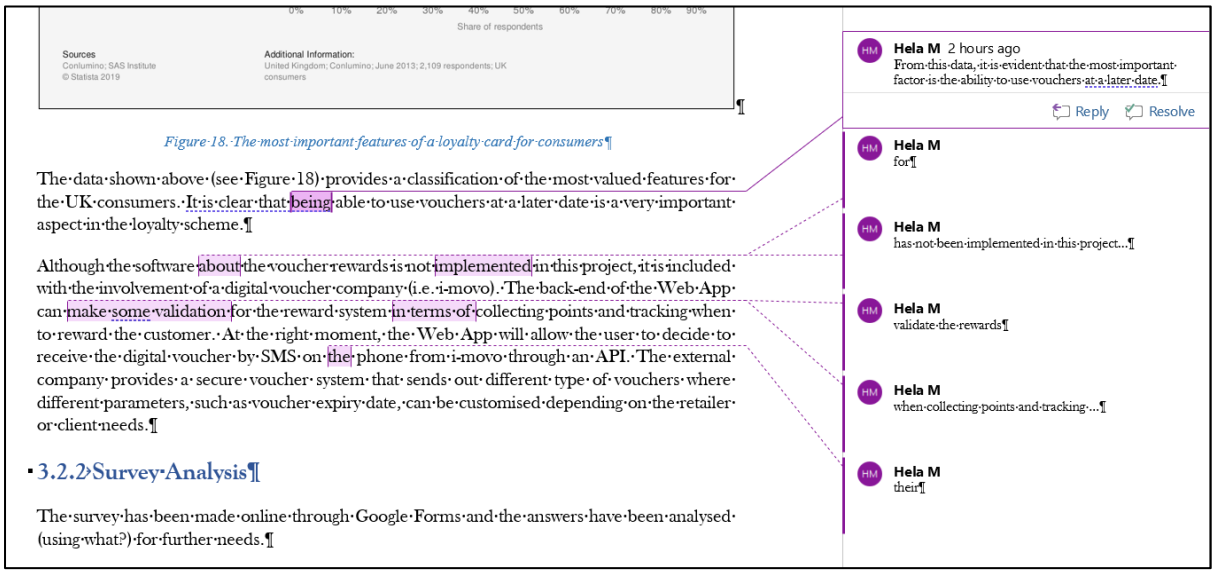


Figure 87. Comments on Project Report by reviewer

10.6 Survey Questions and Possible Answers

1) What type of smartphone do you have?

- A: Android, iOS, Other.

2.1) If the respondent chose "Android" in 1)

2.1.1) Which brand is your Android smartphone?

- A: Asus, BlackBerry, Elementary, Fairphone, Google, Honor, HTC, Huawei, LG, Motorola, Nokia, OnePlus, Oppo, Samsung, Sony, Xiaomi.

2.1.2) What is the model?

- A: Open answer.

2.2) If the respondent chose "iOS" in 1)

2.2.1) What is your iPhone model?

- A: "3G, 3GS, 4, 4S, 5, 5C, 5S"; "6, 6+, 6S, 6S+, SE (1st Gen)"; "7, 7+, 8, 8+, X"; "XS, XS Max, XR"; "11, 11 Pro, 11 Max, SE (2nd Gen)".

2.3) If the respondent chose "Other" in 1)

2.3.1) Which is the model of your smartphone?

- A: Open answer.

3) If the respondent chose "Android" or "Other" in 1)

3.1) Does your smartphone have NFC functionality?

3.1) Description: "NFC is a close-range wireless communication that enables your smart device to use features like contactless payments."

- A: Yes, No.

4.1) If the respondent chose an iOS model >= "6, 6+, 6S, 6S+, SE (1st Gen)" in 2.2.1) or if they chose "Yes" in 3.1)

4.1.1) How often do you use your smartphone to pay in shops?

4.1.1) Description: "Replacing your Credit/Debit Card with Apple Pay, Google Pay, Samsung Pay, or other..."

- A: "ALWAYS - You only pay with your smartphone", "OFTEN - You mostly use your smartphone to pay", "SOMETIMES - You use either the smartphone or the bank card to pay",

"RARELY - You use your smartphone to pay only if you have to", "NEVER - You only pay with your bank card or cash".

4.2) If the respondent chose an iOS model = "3G, 3GS, 4, 4S, 5, 5C, 5S" or if they chose "No" in 3.1)

4.2.1) Would you use your smartphone to pay in shops if it would have NFC functionality?

4.2.1) Description: "You would be able to replace your Credit/Debit Card with Apple Pay, Google Pay, Samsung Pay, or other..."

- A: Yes, No.

5) The "Bonus questions!" section is available to all respondents, regardless of their previous answers

5.1) How many times did you lose the chance to use a loyalty card because you did not have it with you?

- A: "0", "1-5", "6-10", "More than 10".

5.2) Do you think it would be useful to access loyalty cards on your smartphone?

5.2) Description: "The project prototype on development is a loyalty card web app that uses NFC."

- A: "1 to 5" slider, where 1 = "No benefit" and 5 = "Extremely useful".

* All questions were obligatory.

10.7 GitHub Issue Discussion

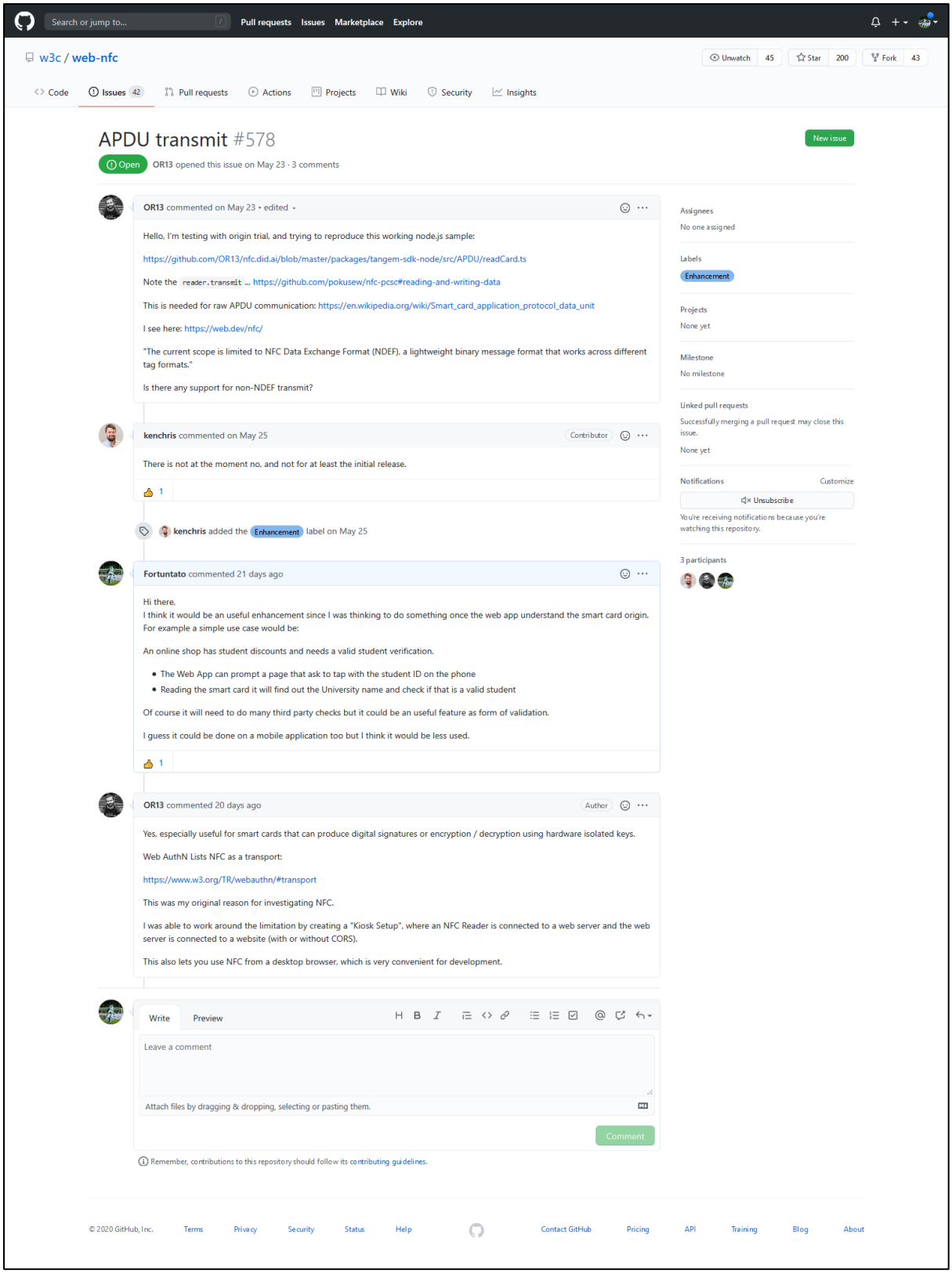


Figure 88. Discussion about future enhancement of the Web API to include APDU transmit