6COSC006W - Final Year Project Report

Contactless Voucher

Shouyi Cui (w1618594)

**Supervisor:** Barbara Villarini

BEng (Hons) in Software Engineering

School of Computing & Engineering

University of Westminster

Date: 13th July 2020

# Declaration

# Abstract

# Acknowledgements

# Table of Contents

# List of Figures

# List of tables

# Abbreviations

# 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.

## 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.

Image sources:  https://www.tradeprint.co.uk/dam/jcr:a3f81a33-30e6-43c7-a798-6a9fdbf89a9b/comp_loyaltycard_170620_0187.jpg

https://www.thesun.co.uk/wp-content/uploads/2017/04/nero1.png



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.

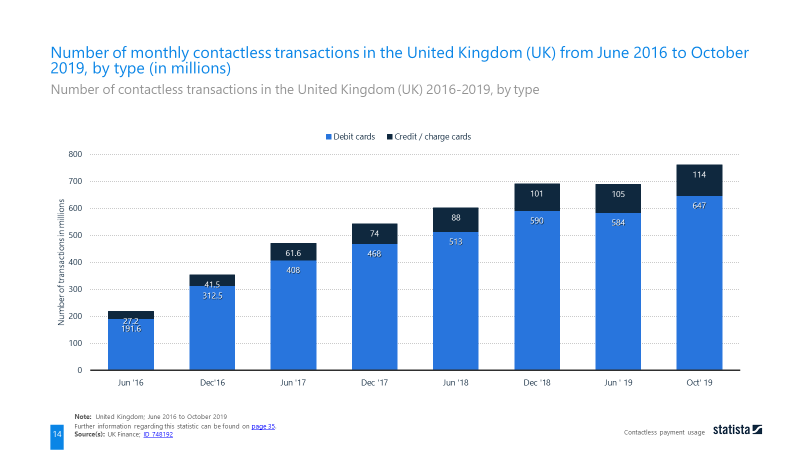


Figure 2. Monthly contactless transaction in the UK from June 2016 to October 2019

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).

## 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

# 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.

## 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.

### 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.

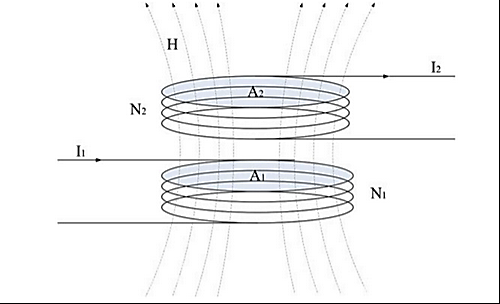


Figure 5. Inductive coupled coils

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

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).

### 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).

### 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.

### 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).

Image source: https://www.ecom-ex.com/fileadmin/_processed_/5/f/csm_17-02-07_RFID-NFC-final_153b8e3d75.jpg


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.

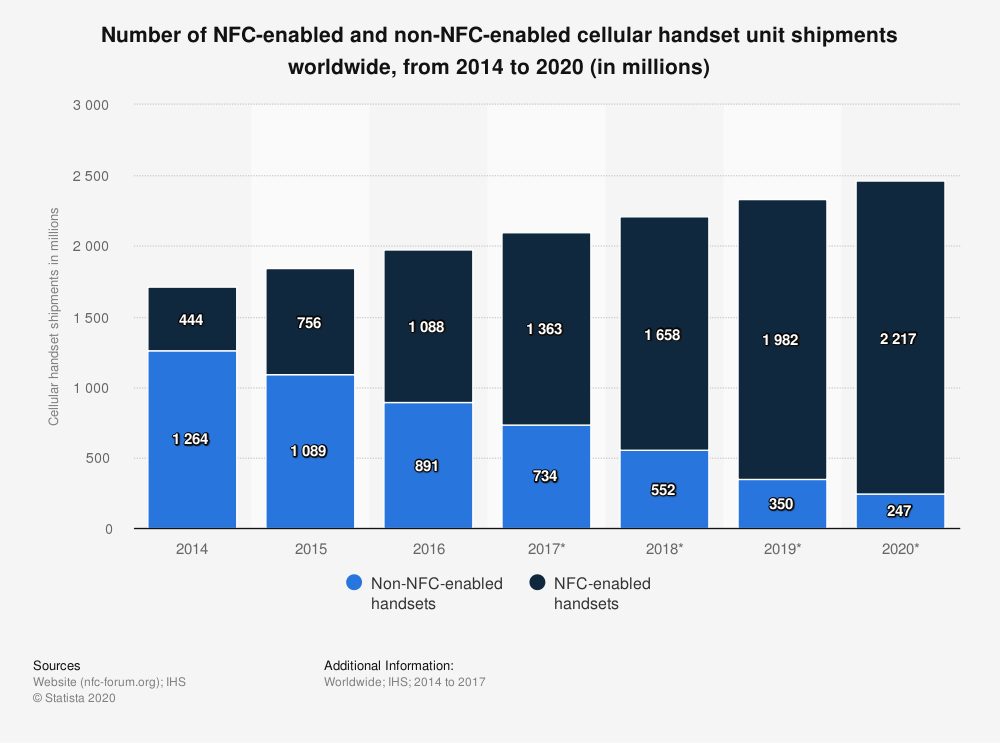


Figure 7. NFC enabled handsets from 2014 to 2020

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).

### 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).

Image source: 
https://w3c.github.io/web-nfc/#the-ndef-record-and-fields

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.

Image source:
https://w3c.github.io/web-nfc/#the-ndef-record-and-fields

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

### 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).

Image source:
https://storage.googleapis.com/support-forums-api/attachment/thread-13605645-4739816311749951377.jpg

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.

## 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.

### 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.

### 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.

### 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.

#### 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).

#### 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).

#### 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.

#### 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).

#### 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).

#### 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.

Immagine screenshot source:
https://lh3.googleusercontent.com/CqPgAtSTKal-DProv1_xMobPhcA_jYlyAM4IZ4005p-bqhew2_v2EFLB6Lq1NOwpwA=w3068-h1444

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.

### 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.

Image source: https://play.google.com/store/apps/details?id=com.embargoapp.app&hl=en_GB


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.

#### 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.

### 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.

### 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.

#### 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.

#### 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:

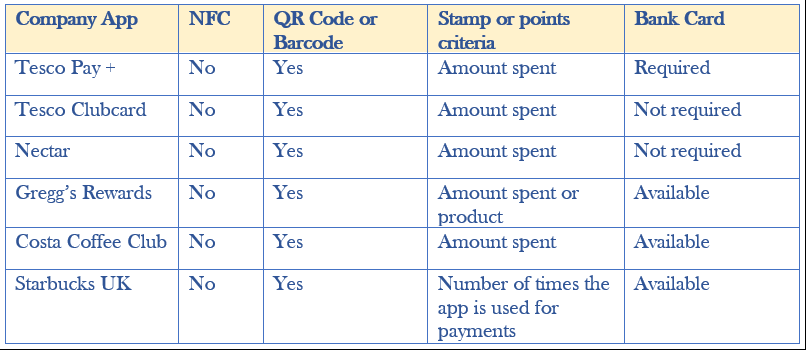


Figure 13.Comparison of company mobile loyalty apps features and loyalty scheme details

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.

## 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.

### 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.

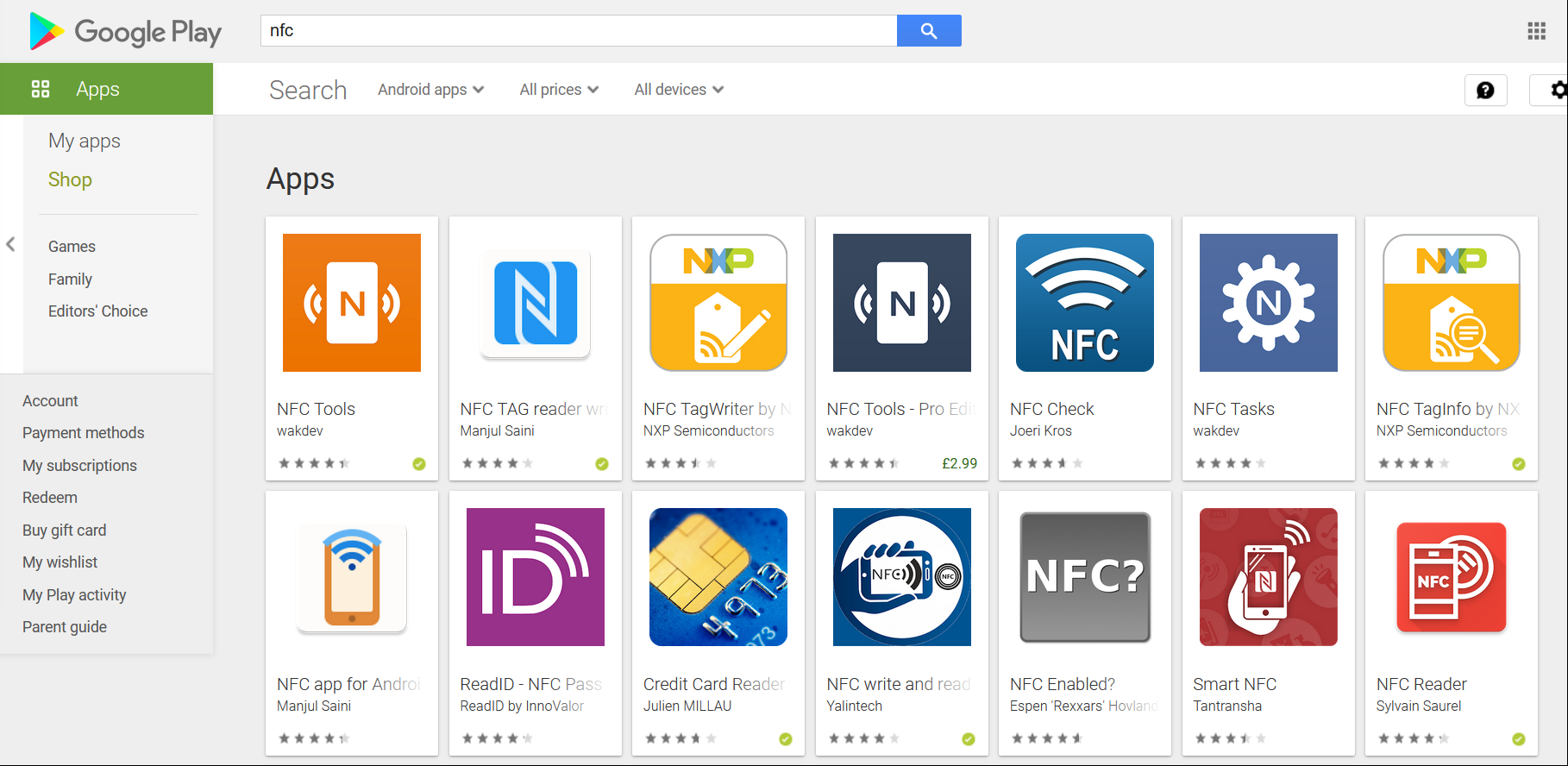


Figure 14. NFC application search result on Google Play (Search made on July 2020)

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.

#### 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).

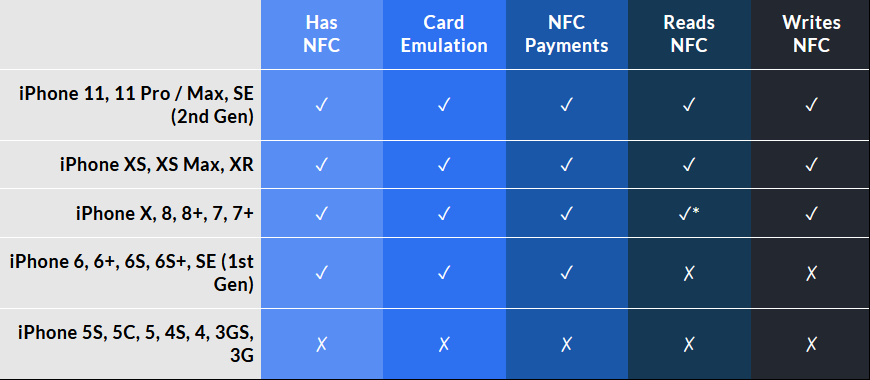


Figure 15. NFC features enabled on iOS smartphones

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

### 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.

### 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.

### .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.

### .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.

### 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.

### 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.

# 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.

## 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 Systemlayer 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.

## 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.

### Statistical Analysis

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

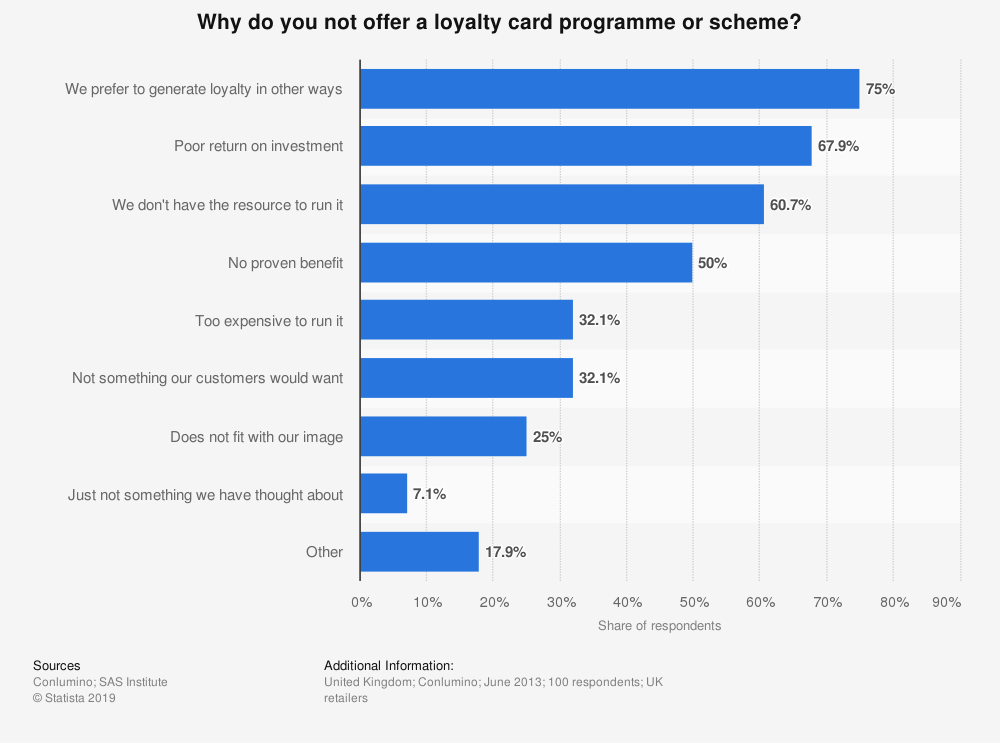


Figure 17. Classification of reasons for not offering a loyalty scheme

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.

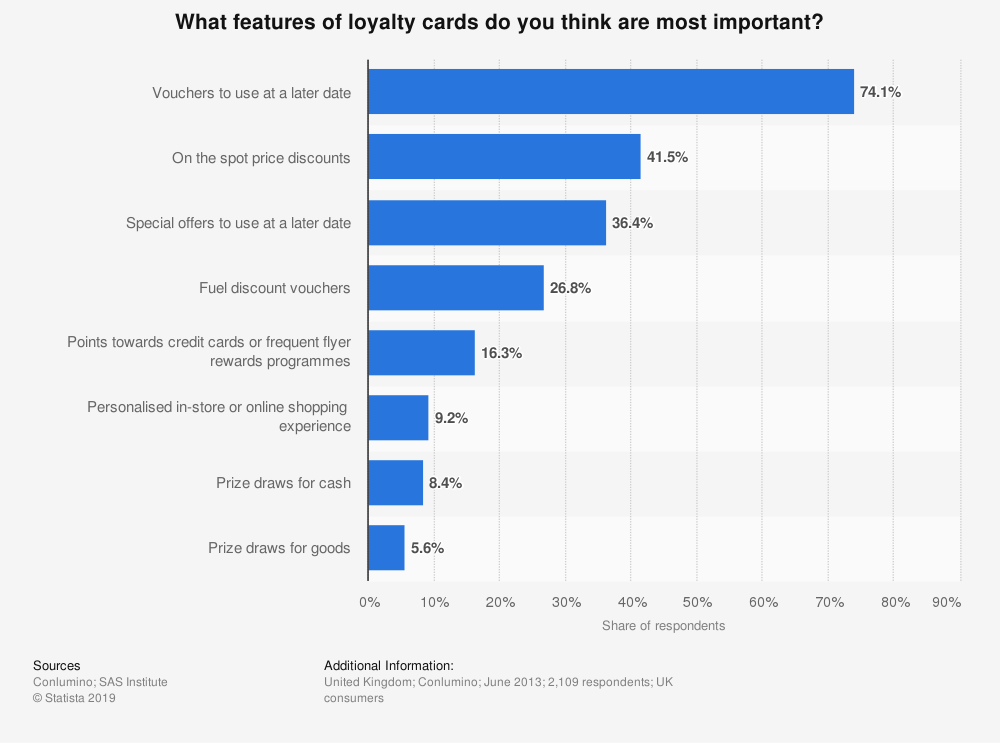


Figure 18. The most important features of a loyalty card for consumers

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.

### Survey Analysis

The survey has been made online through Google Forms and the answers have been analysed (using what?) for further needs.

### Interviews and Feedbacks

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

* Laurence – user registration security and data modelling
* David – engage with the customer to make it appealing to use and become aware of the brand
* Toby for voucher system to be sent through mobile and not email – date validation – scheme validation
* Alessandro – Date validation and scheme
* Takefumi Yoshi for tag id as serial number and identification
* Lincoln for initial help and support

## Modelling requirements and relevant diagrams

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

### Use Case Model

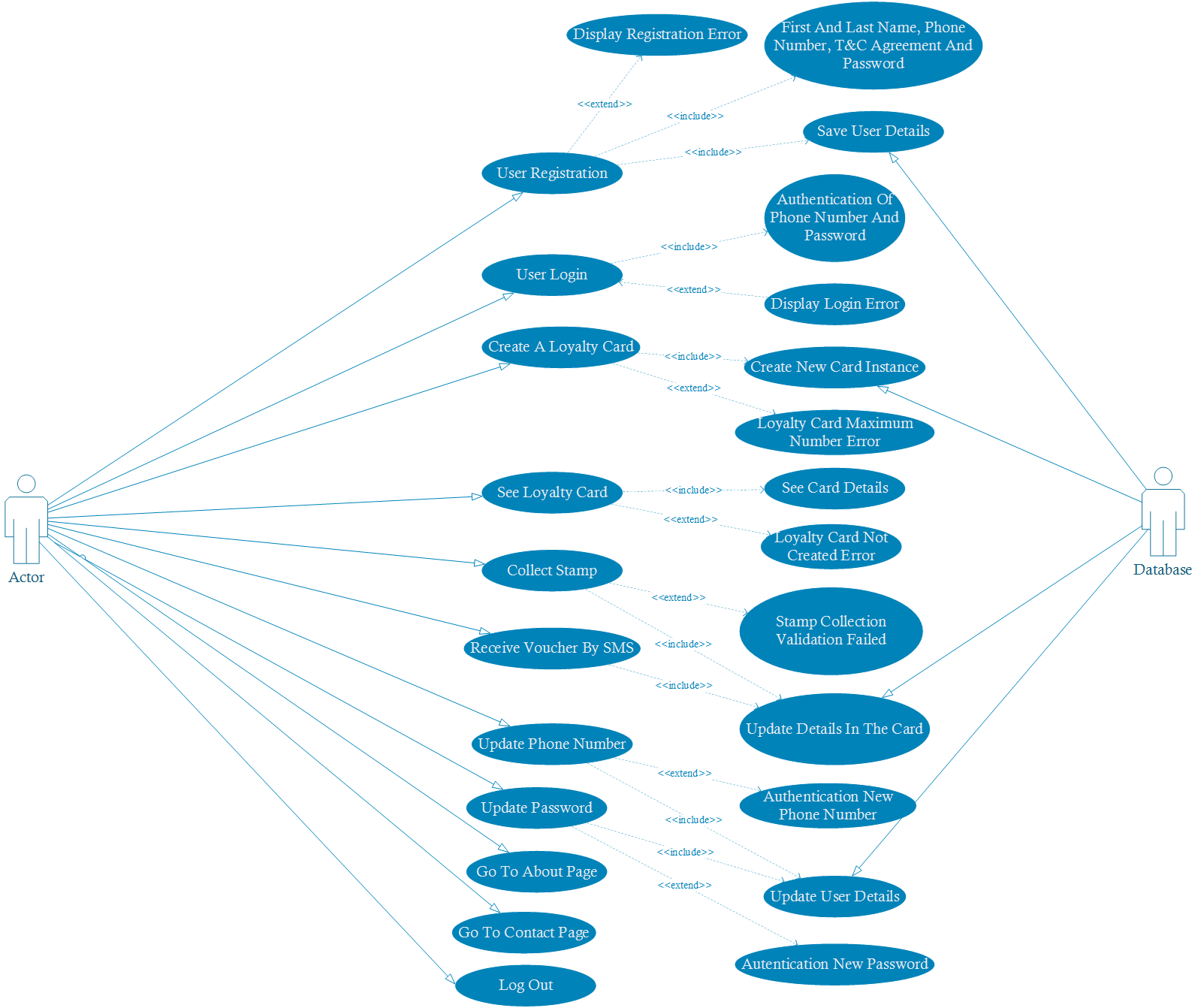


Figure 19. Use Case Diagram

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

### 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)

### UML (Unified Modelling Language)

Figure 20. Main Class Diagram of the Contactless Loyalty Web Application

### ERD (Entity Relationship Diagram)

The following diagram (see Figure 21) 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 21. Database ERD

A future implementation for improved user management in the database could look like the diagram below (Figure 22) where the user can have the possibility to have a two factor authentication using a token and can hold different roles.

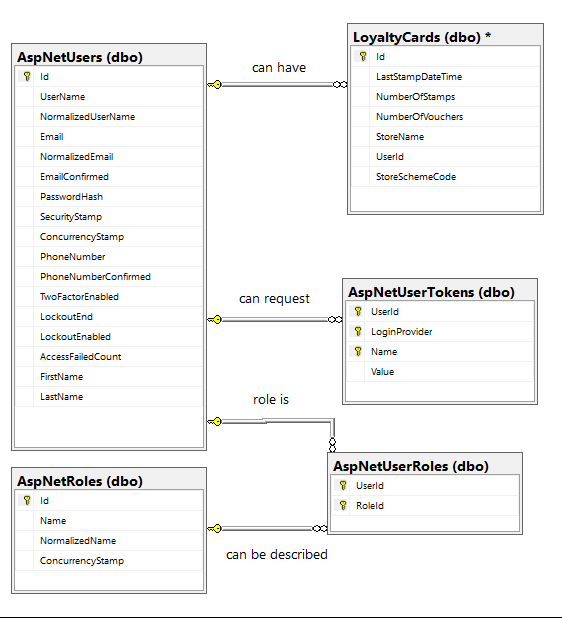


Figure 22. Possible database design with more user management

## 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 and least 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.

### Functional Requirements

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



Figure 23. Table of functional requirements

### Non-Functional Requirements

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

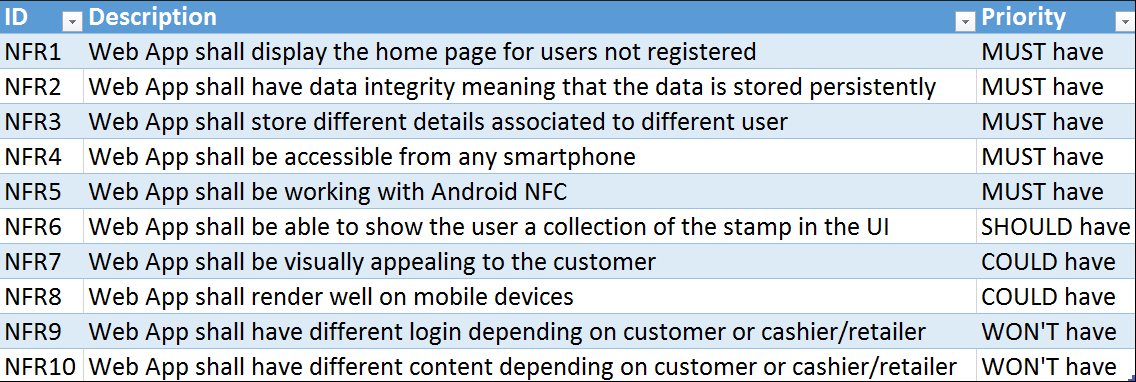


Figure 24. Table of non-functional requirements

## Legal, Social and Ethical Issues

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

### 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).

### 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.

### 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.

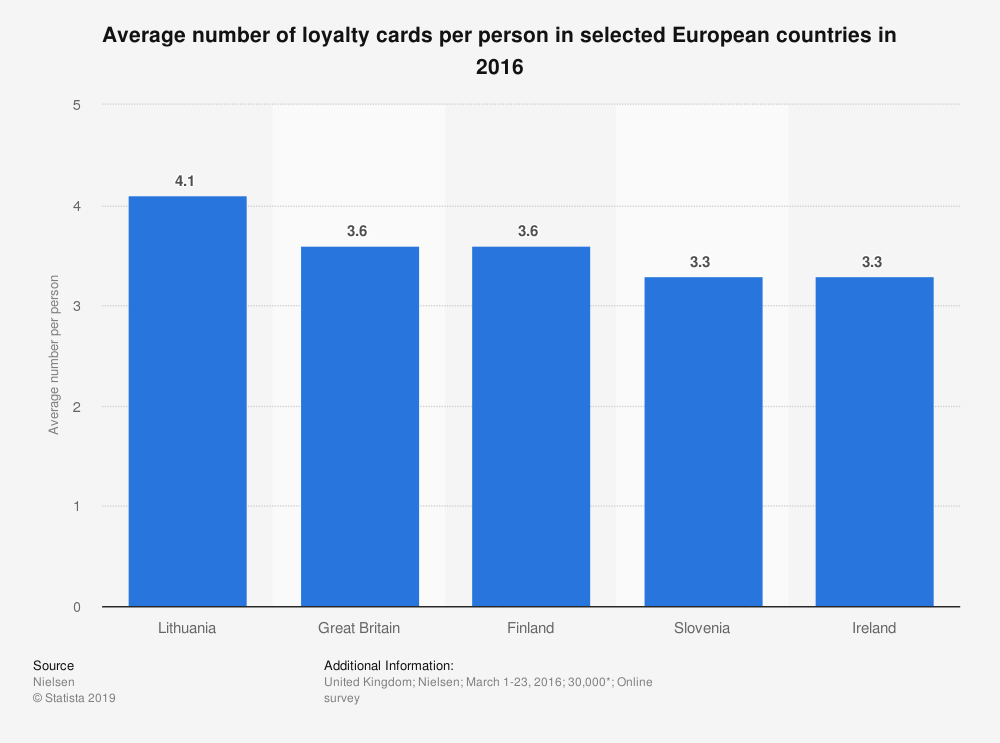


Figure 25. Average number of loyalty cards per person

Combining the previous factor to the statistic of cards per person (see Figure 22), 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 3 digital cards in place of the physical cards.

# 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.

## 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 26). 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.

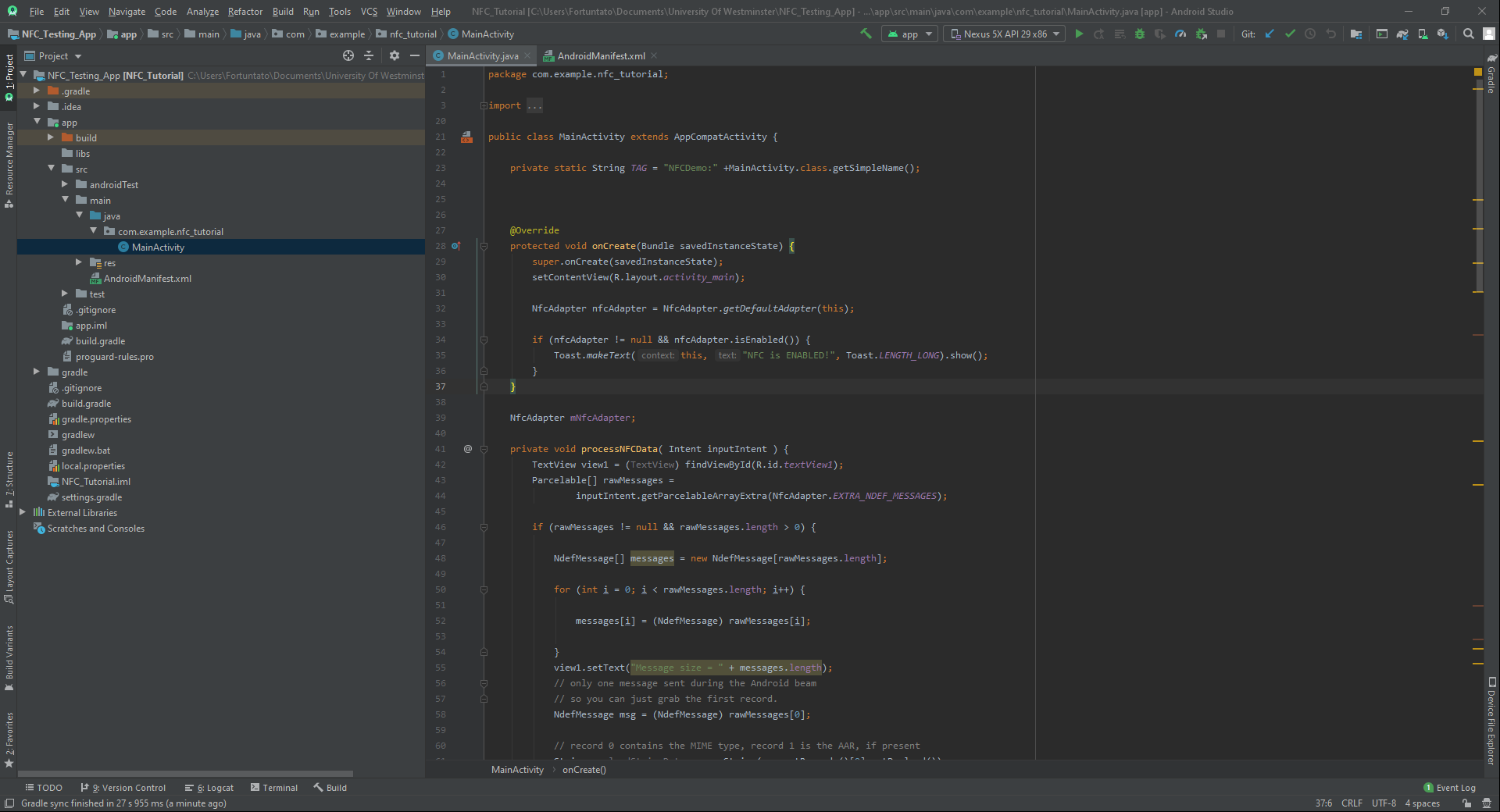


Figure 26. Android Studio project prototype

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 information, a Web Application was created as a playground of the new key features during the primary iterations of the project.

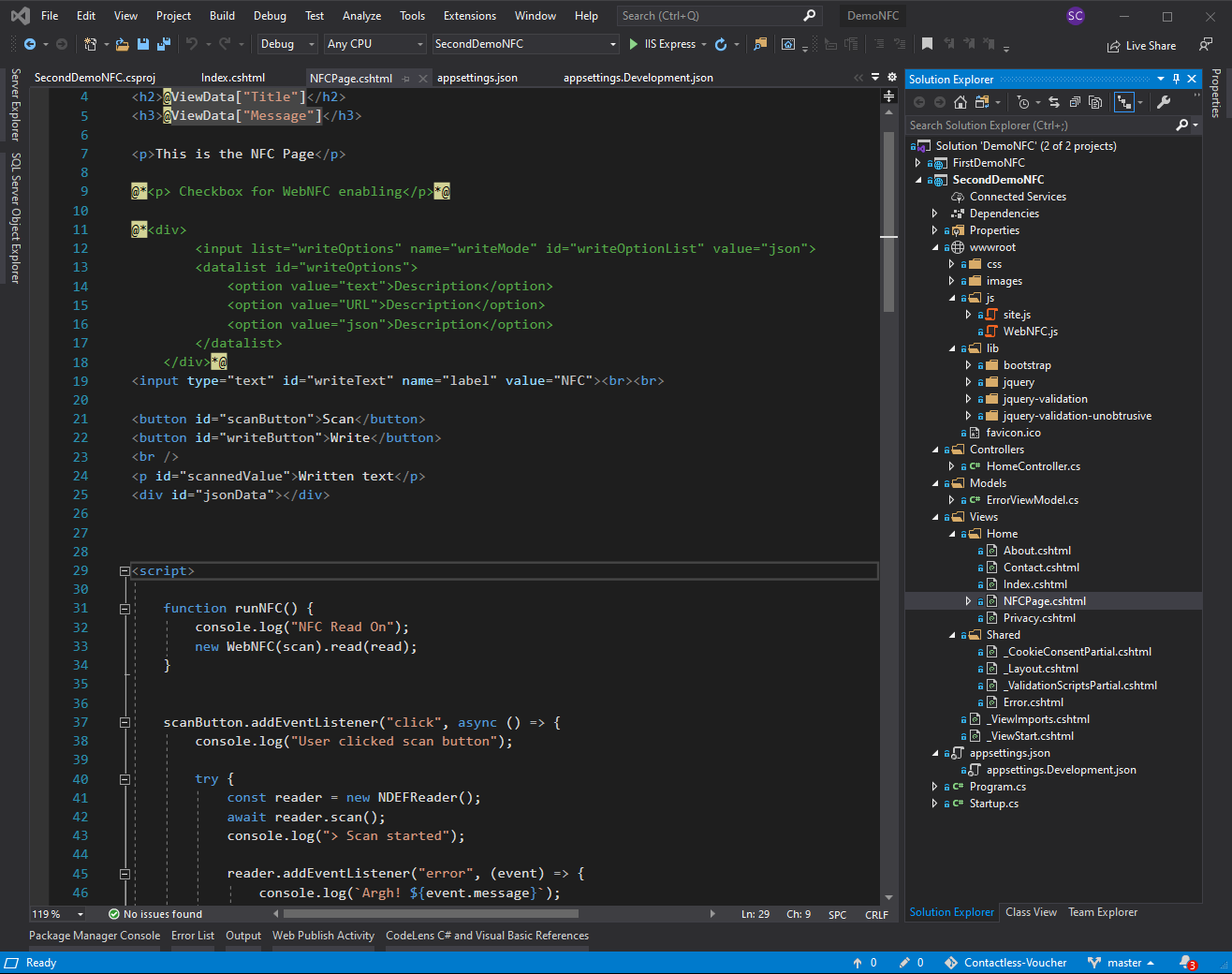


Figure 27. Initial Web App to test Web-NFC (source code)

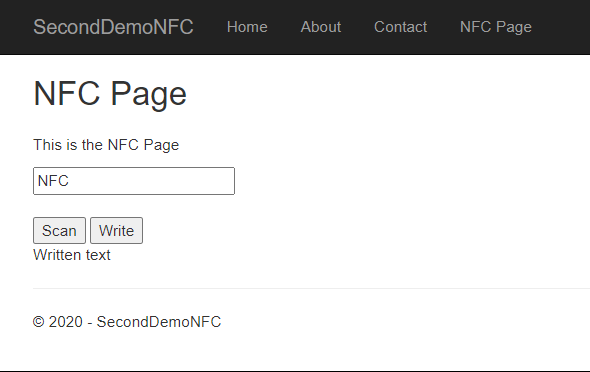


Figure 28. Initial Web App to test Web-NFC (front page)

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 current final solution.

## 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 prove 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.

# 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.

## 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 29).

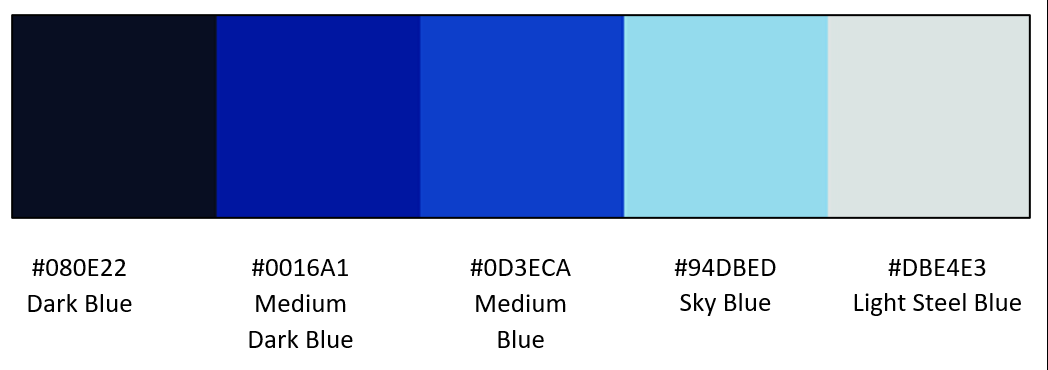


Figure 29. Colour palette for the Web App

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 30) at the top of the login and registration page, all the colours are used.



Figure 30. Banner used at the top of the landing page

## CSS and Bootstrap

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

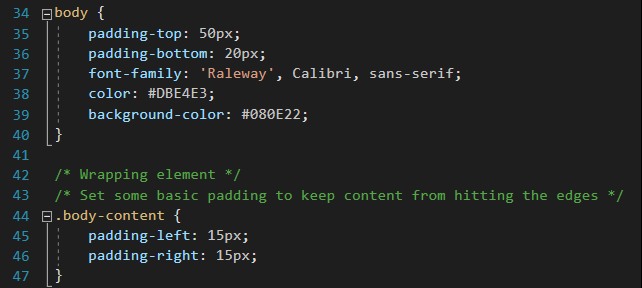


Figure 31. CSS body style and body content padding

For example, in the figure above (see Figure 31) the default font type “Raleway” has been declared. 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).



Figure 32. 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.

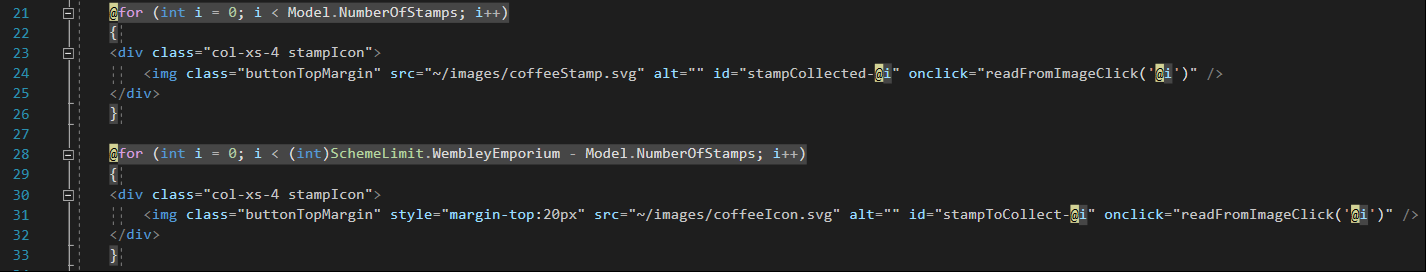


Figure 33. Code showing the Stamps on the Loyalty Card page with Bootstrap class layout (col-xs-4)

## 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 38) is implemented dynamically depending on the data retrieved from the database. As shown in the Figure 32 (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.

## SVG Images

The use of SVG (Scalable Vector Graphics) such as the banner (Figure 30) 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.

Machine generated alternative text:
flaticon 
Authors 
Packs 
More Tools • 
Search more icons in this style 
Style: Lineal x 
Become a contributor 
Pricing Login 
Q 
Register 
o Collections 
Icons • 
P Save 
Search for icons and packs 
(Z Edit icon 
Intemet Technology 
Register for free and download the full pack 
Register & Download 
Smartphone free icon 
PNG 
* Add to collection 
Flaticon License 
SVG 
Free for personal and commercial purpose with attribution. More info 
Attribution is required. How to attribute? 
Vitaly Gorbachev 
View all 39,710 Icons 
More icons from Intemet Technology pack 
Style: Vitaliy Gorbachev Lineal 
NFC 
Follow 
Get 10 free Icons 
Adobe Stock 
st 
Sponsored results by 
NFC 
D)) 
000 
NFC 

Figure 34. Flaticon image research

## 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 35) for the colours and shapes used.

Testo alternativo generato dal computer:
Process 
Subprocess 
Decision 
Start/End 

Figure 35. 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.

### Login and Registration

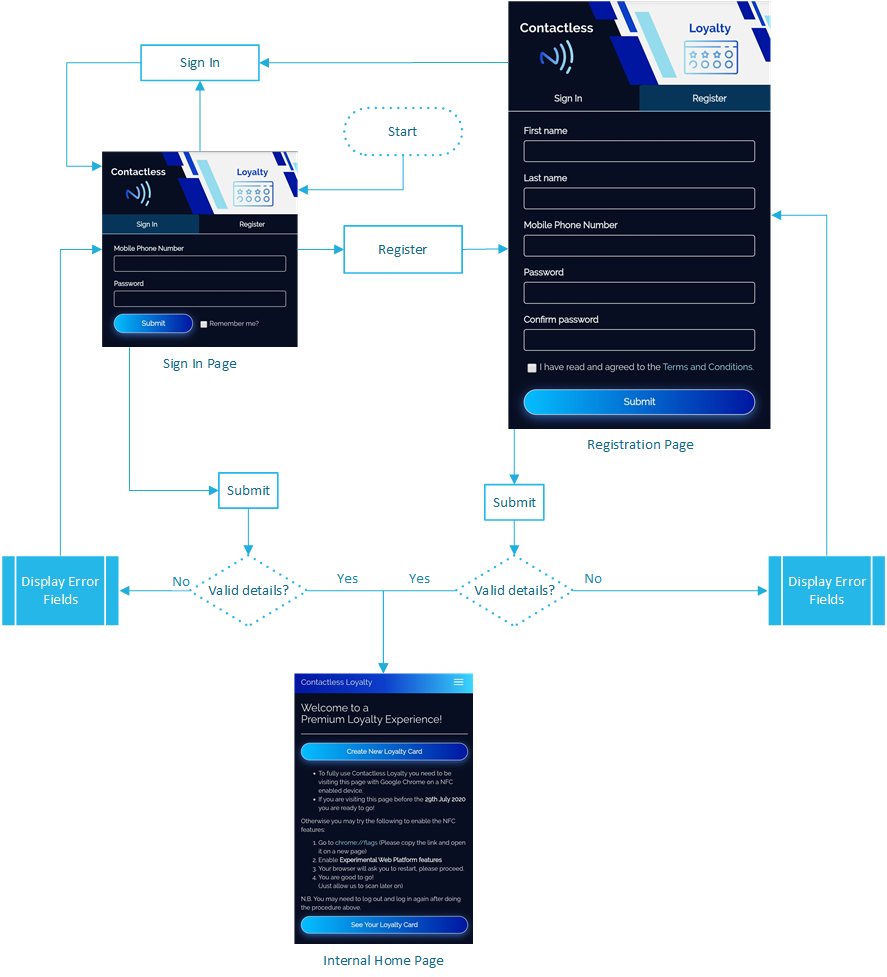


Figure 36. Login and Registration Flowchart with Wireframes

### Internal Home Page after Login

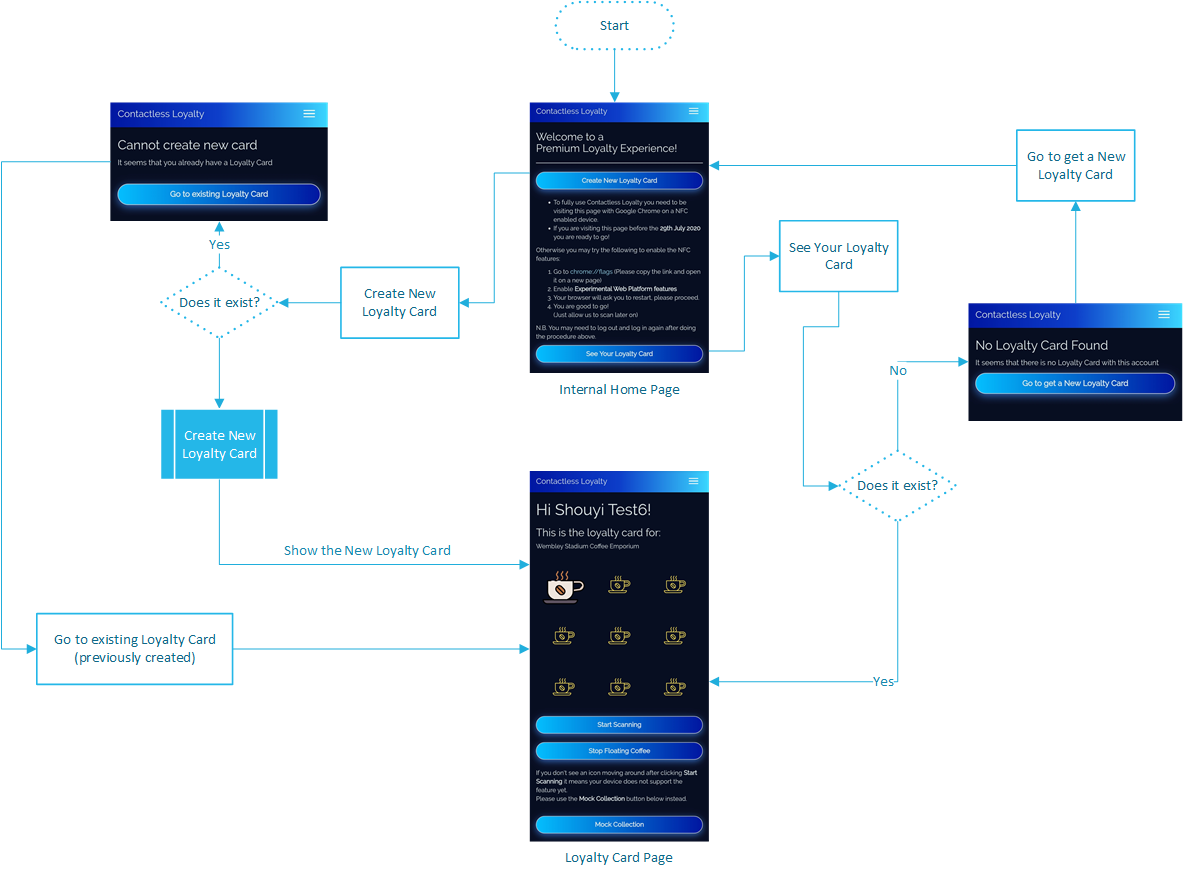


Figure 37. Internal Home Page with Flowchart and Wireframes

### Loyalty Card Page

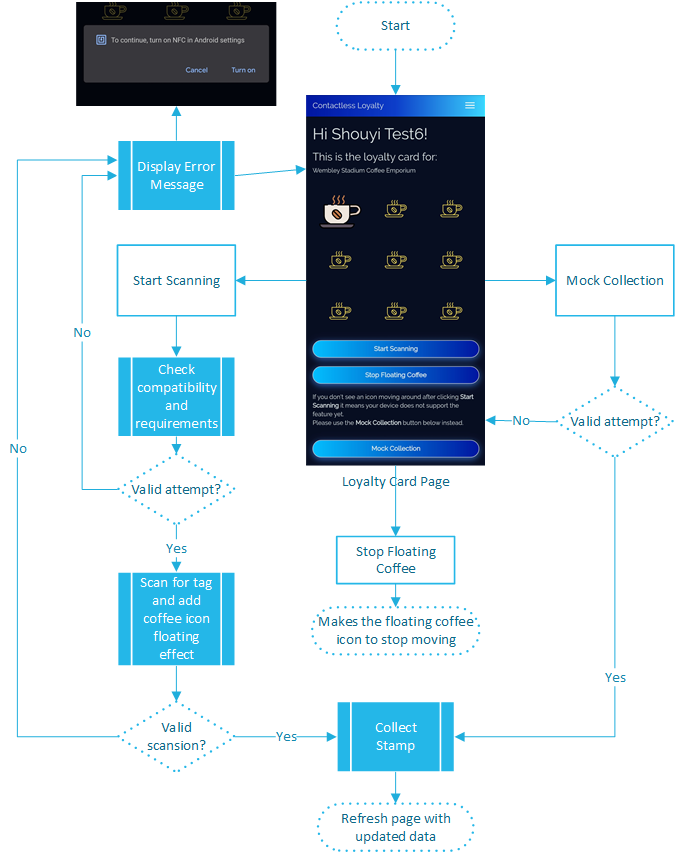


Figure 38. Loyalty Card Page with Flowchart and Wireframes

### Reward Available Flowchart and Voucher Sent Page

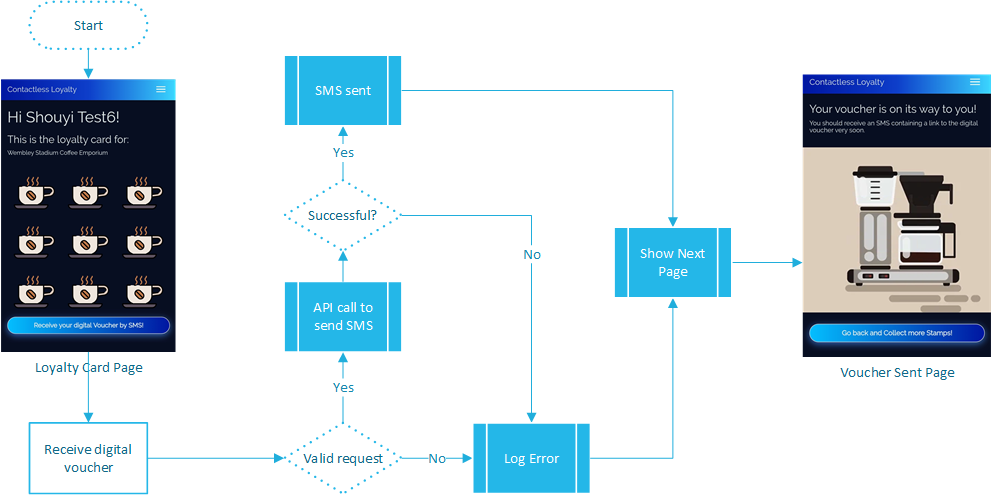


Figure 39. Collection Page Flowchart with Wireframes

### Menu Options and Account Management

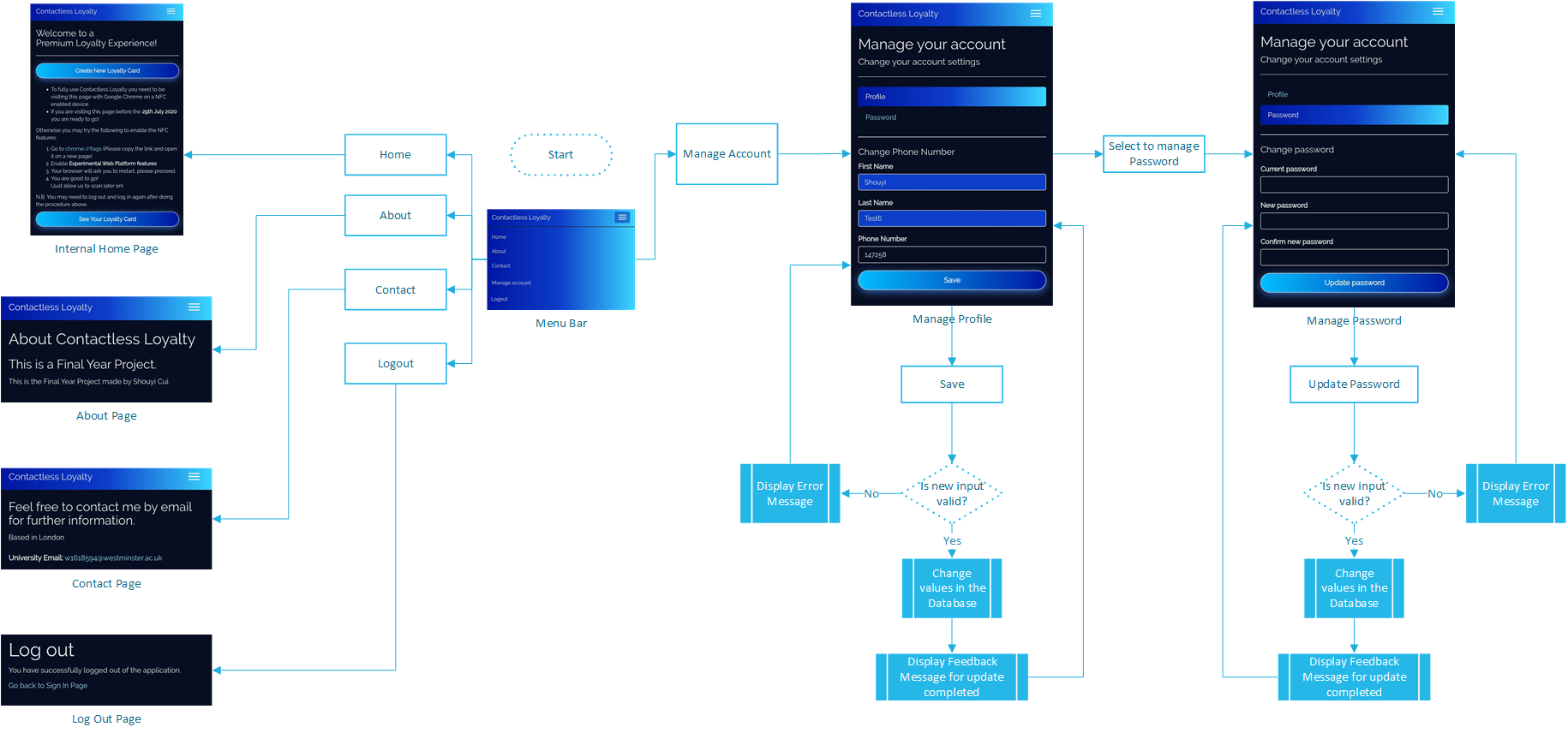


Figure 40. Menu options and Manage Account Page Flowchart with Wireframes

# Tools and implementation

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

## 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.

### 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.

### 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.

### 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.

### 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.

#### 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.

#### Entity Framework Core

As previously mentioned in (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.

#### 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.

Machine generated alternative text:
Error List 
Entire Solution 
Code 
A 
596 Warnings Oof 5 Messages Build + IntelliSense 
Description 
SA1633. 
• CSharp.Documentation : The file has no header, the header Xml is invalid, or the header is not located at the top of the file. 
SA1516. 
• CSharp.Layout : Adjacent elements must be separated by a blank line. 
SA16DO. 
• CSharp.Documentation : The class must have a documentation header. 
SA16DO. 
• CSharp.Documentation : The method must have a documentation header. 
SA1513. 
• CSharp.Layout : Statements or elements wrapped in curly brackets must be followed by a blank line. 
SAIDD9. 
• CSharpSpacing : Invalid spacing around the closing parenthesis. 
: CSharp.ReadabiIit,' : The closing parenthesis or bracket must be placed on the same line as the last parameter in the parameter list. 
SA12DO. 
• CSharp.Ordering : All using directives must be placed inside of the namespace. 
SA1633. 
• CSharp.Documentation : The file has no header, the header Xml is invalid, or the header is not located at the top of the file. 
SA12DO. 
• CSharp.Ordering : All using directives must be placed inside of the namespace. 
SA12DO. 
• CSharp.Ordering : All using directives must be placed inside of the namespace. 
SA12DO. 
• CSharp.Ordering : All using directives must be placed inside of the namespace. 
SA12DO. 
• CSharp.Ordering : All using directives must be placed inside of the namespace. 
SA12DO. 
• CSharp.Ordering : All using directives must be placed inside of the namespace. 
SA1210. 
• CSharp.Ordering : Using directives must be sorted alphabetically by the namespaces. 
SA12DO. 
• CSharp.Ordering : All using directives must be placed inside of the namespace. 
SA12DO. 
• CSharp.Ordering : All using directives must be placed inside of the namespace. 
SA12DO. 
• CSharp.Ordering : All using directives must be placed inside of the namespace. 
SA12DO. 
• CSharp.Ordering : All using directives must be placed inside of the namespace. 
SA12DO. 
• CSharp.Ordering : All using directives must be placed inside of the namespace. 
SA12DO. 
• CSharp.Ordering : All using directives must be placed inside of the namespace. 
SA16DO. 
• CSharp.Documentation : The class must have a documentation header. 
SA16DO. 
• CSharp.Documentation : The field must have a documentation header. 
SA13D9. 
• CSharp.Naming : Field names must not start with an underscore. 
SA16DO. 
• CSharp.Documentation : The field must have a documentation header. 
SA13D9. 
• CSharp.Naming : Field names must not start with an underscore. 
SA16DO. 
• CSharp.Documentation : The constructor must have a documentation header. 
SA1126. 
• CSharp.ReadabiIit,' : The call to _signInManager must begin with the 'this.', 'base.', 'object.' or 'LoginModeI.' or 'PageModeI.' prefix to indicate the intended method call. 
SA1126. 
• CSharp.ReadabiIit,' : The call to _ logger must begin with the 'this.', 'base.', 'object.' or 'LoginModeI.' or 'PageModeI.' prefix to indicate the intended method call. 
SA16DO. 
• CSharp.Documentation : The propety must have a documentation header. 
SA16DO. 
• CSharp.Documentation : The propety must have a documentation header. 
SA16DO. 
• CSharp.Documentation : The propety must have a documentation header. 
SA16DO. 
• CSharp.Documentation : The propety must have a documentation header. 
SA16DO. 
• CSharp.Documentation : The class must have a documentation header. 
SA16DO. 
• CSharp.Documentation : The propety must have a documentation header. 
SA16DO. 
• CSharp.Documentation : The propety must have a documentation header. 
SA16DO. 
• CSharp.Documentation : The propety must have a documentation header. 
SA1201 
: CSharo.Orderina : All classes must be olaced after all methods. 
P roject 
Contact less Loyalty 
Contact less Loyalty 
Contact less Loyalty 
Contact less Loyalty 
Contact less Loyalty 
Contact less Loyalty 
Contact less Loyalty 
Contact less Loyalty 
Contact less Loyalty 
Contact less Loyalty 
Contact less Loyalty 
Contact less Loyalty 
Contact less Loyalty 
Contact less Loyalty 
Contact less Loyalty 
Contact less Loyalty 
Contact less Loyalty 
Contact less Loyalty 
Contact less Loyalty 
Contact less Loyalty 
Contact less Loyalty 
Contact less Loyalty 
Contact less Loyalty 
Contact less Loyalty 
Contact less Loyalty 
Contact less Loyalty 
Contact less Loyalty 
Contact less Loyalty 
Contact less Loyalty 
Contact less Loyalty 
Contact less Loyalty 
Contact less Loyalty 
Contact less Loyalty 
Contact less Loyalty 
Contact less Loyalty 
Contact less Loyalty 
Contact less Loyalty 
Contact less Lovalt.' 
Search Error List 
Identity HostingStartup.cs 
Identity HostingStartup.cs 
Identity HostingStartup.cs 
Identity HostingStartup.cs 
Identity HostingStartup.cs 
Identity HostingStartup.cs 
Identity HostingStartup.cs 
Login.cshtml.cs 
Login.cshtml.cs 
Login.cshtml.cs 
Login.cshtml.cs 
Login.cshtml.cs 
Login.cshtml.cs 
Login.cshtml.cs 
Login.cshtml.cs 
Login.cshtml.cs 
Login.cshtml.cs 
Login.cshtml.cs 
Login.cshtml.cs 
Login.cshtml.cs 
Login.cshtml.cs 
Login.cshtml.cs 
Login.cshtml.cs 
Login.cshtml.cs 
Login.cshtml.cs 
Login.cshtml.cs 
Login.cshtml.cs 
Login.cshtml.cs 
Login.cshtml.cs 
Login.cshtml.cs 
Login.cshtml.cs 
Login.cshtml.cs 
Login.cshtml.cs 
Login.cshtml.cs 
Login.cshtml.cs 
Login.cshtml.cs 
Login.cshtml.cs 
Loain.cshtml.cs 
Line 
11 
13 
15 
31 
32 
32 
Supp... 

Figure 41. Example of StyleCop warnings

Machine generated alternative text:
File Edit 
View 
P roject 
Build 
Debug 
Debug 
Test Analyze 
Any CPU 
Tools 
Extensions Window 
Contact less Loyalty 
Help 
Search (Ctrl+Q) 
IIS Express 
p 
Contact less Loyalty 
Login.cshtml.cs 
DashboardControIIer.cs 
Contact less Loyalty' 
Identity HostingStartup.cs 
puOI_LC s L r • ang 
[Required] 
Register .cshtml.cs 
r _Lr•s gec 
AccountContactIessLoyaItyUser 
Data baseCo ntex t 
ndex .cshtml.cs 
PhoneNumber 
Das h board.cs 
Startup.cs 
ContactIessLoyaIty.Areas.Identity.Pages.Account.RegisterModeI.I nputModeI 
41 
42 
43 
45 
47 
49 
50 
51 
52 
53 
54 
55 
56 
57 
58 
120% 
[Data Type (Data Type. Text) ] 
[Di splay (Name 
"Last name") ] 
4 references 
public string 
LastName { get; 
[Required] 
[ Phone] 
sec ; 
set; 
[Di splay (Name 
"Mobile Phone Number") ] 
5 references 
public string PhoneNumber { get; set; } 
ErrorMessage = 
"P1ease enter a valid UK phone number starting with 44 or 07") | // Conmented out fc 
[Required] 
[StringLength (100, ErrorMessage 
"The {0} must be at least {2} and at max {1} characters long. " , 
Mi nimumLength 
[Data Type (DataType. Password ) ] 
[Di splay (Name 
"password") ] 
4 references 
public string Password { get; 
[Data Type(DataType . password) I 
set; 
No issues found 
49 
Ch: 17 
SPC 
CRLF 
Show output from: StyleCop 
StyleCop 6.1 (build 6.1.7838.1) started - 
Pass I: 
Pass I: 
Contact lessLoyaIty 
Contact lessLoyaIty 
StyleCop completed - 
Violation Count: 
\Areas \ Identity\ Pages \Account \ Regi ster. c shtml . c s 
\ControI lers \DashboardControI ler. c s 
Package Manager Console 
Error List 
Output 
Web Publish Activity 

Figure 42. Example of StyleCop without warnings

#### 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.

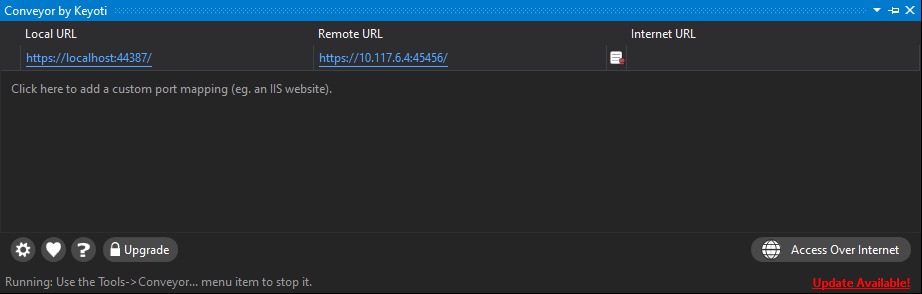


Figure 43. Conveyor running on Visual Studio example

### 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.

### Git

Git is an open source version control system. It has the capability to be used as task management but that aspect of the project development is covered by Paymo (see 6.1.10). Git has been used for source version control since the first iteration of the project including documentation.

Image source: 
https://github.com/Fortuntato/Contactless-Voucher/tree/master/Browser_NFC/Development/ContactlessLoyalty/ContactlessLoyalty 

Figure 44. 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 and (see 10.5).

Justification:

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

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.

### 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.

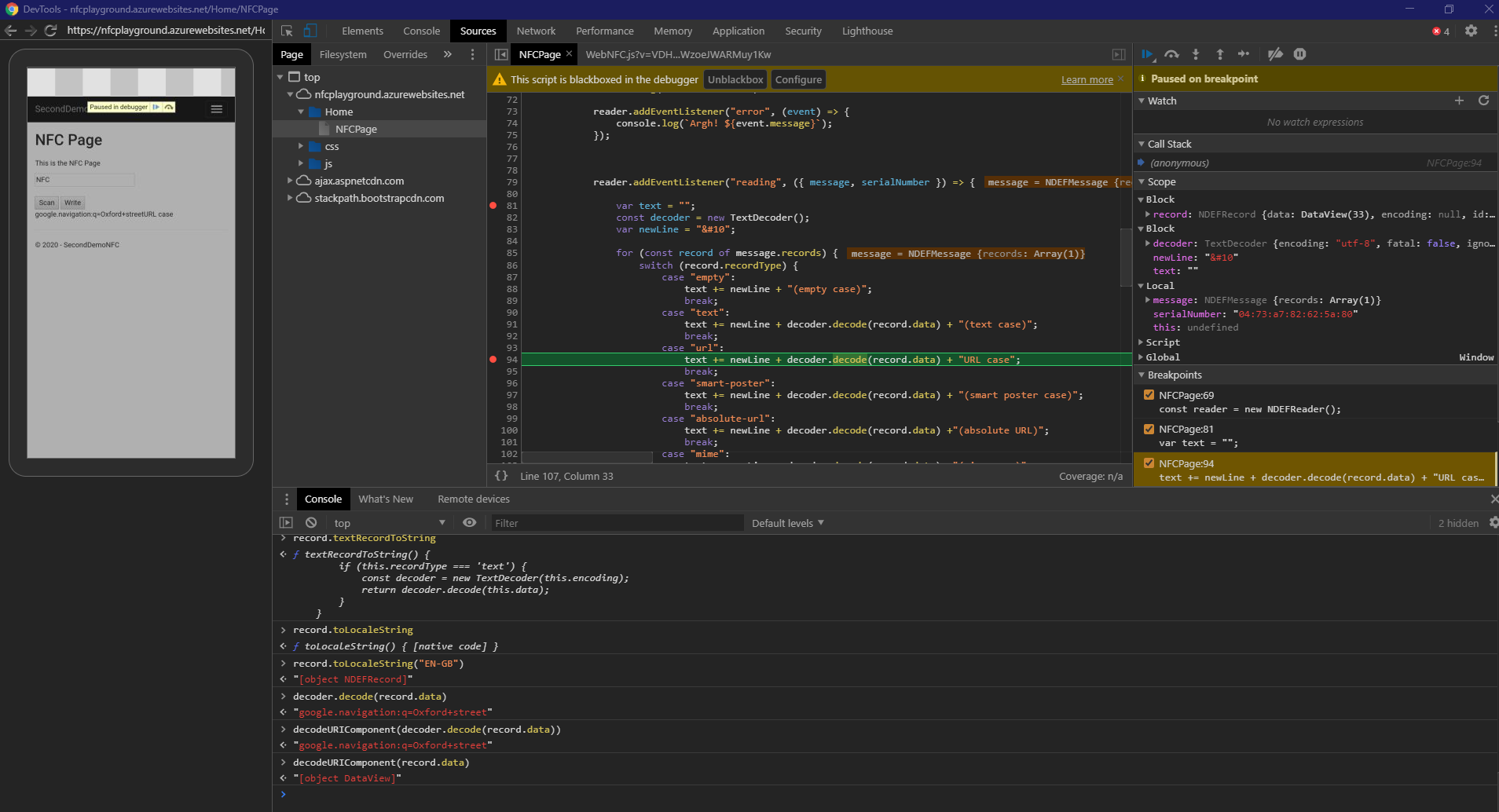


Figure 45. Example of early stage DevTools utilisation

### 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 large diagrams.

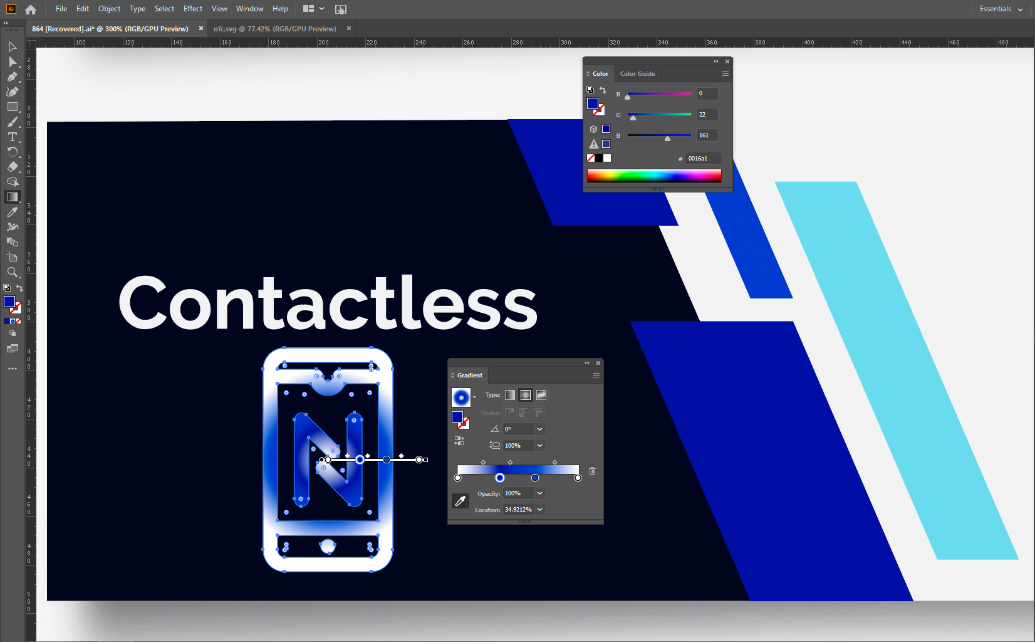


Figure 46. Adobe Illustrator banner template in development

### 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 19) 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.

### 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

### 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.

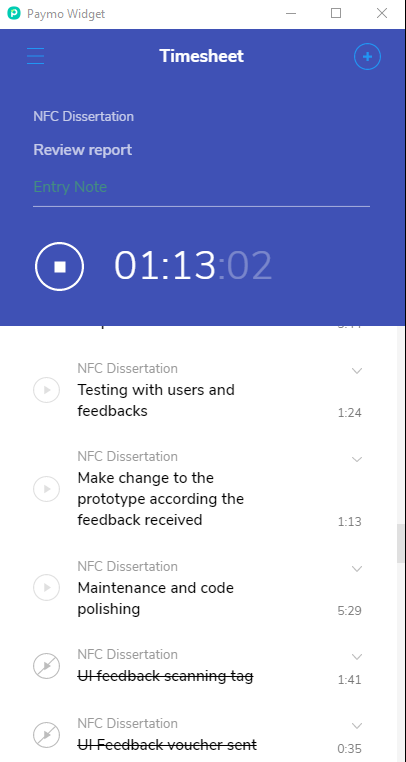


Figure 47. Paymo Desktop Widget example of active time tracking

## Implementation - Loyalty Scheme system

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

### 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 current Visual Studio 2019 version used for development is Version 16.6.3 which is the most up-to-date version.

A list of the required frameworks and packages is shown below (see Figure 48).

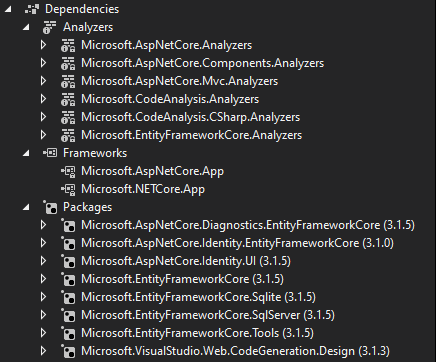


Figure 48. Project dependencies

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

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

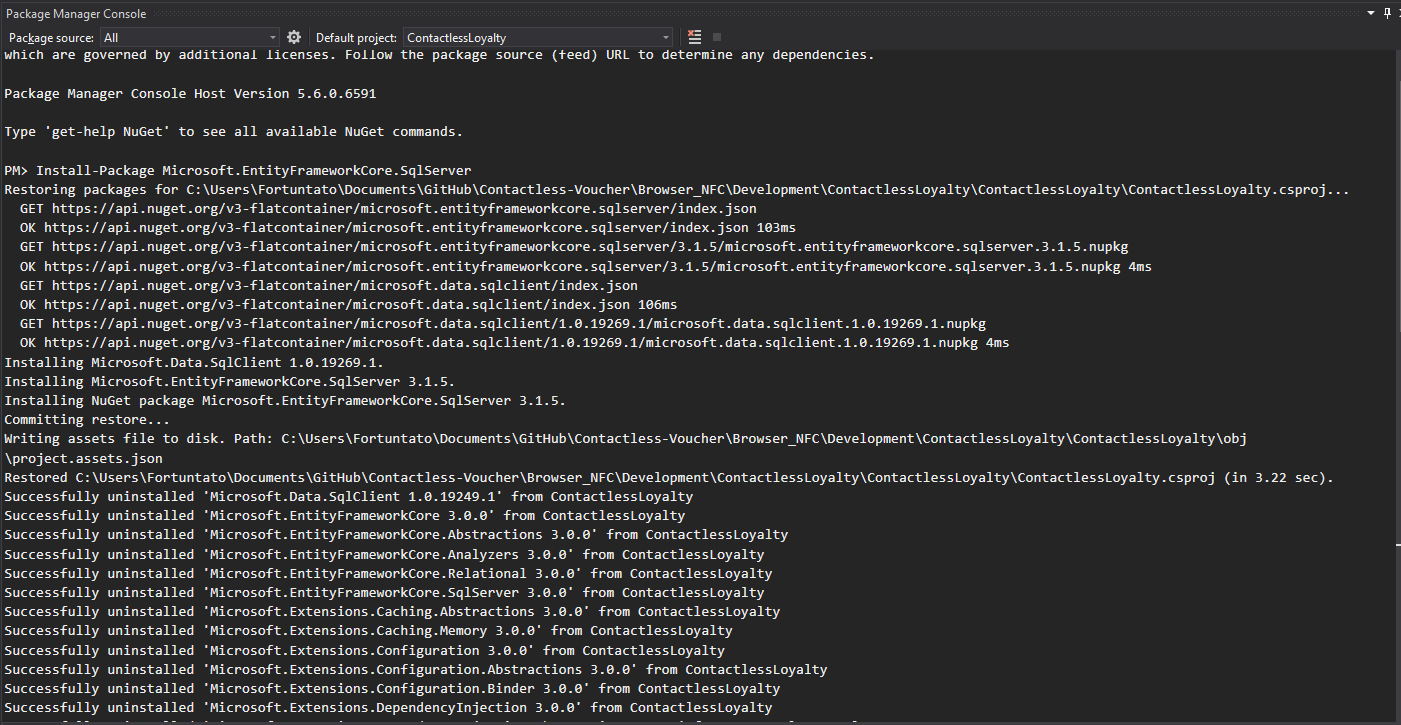


Figure 49. Installing Microsoft Entity Framework Core on Package Manager Console

### Database Setup

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

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

1. Edit the connection string the appsettings.json file. By default Visual Studio tries to reach uses the main settings and then it overrides some values depending on the environment (see Figure 50).
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 51).

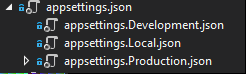
.

Figure 50. Settings for the project with the different environments

Machine generated alternative text:
Package Manager Console 
Package All 
Default prcject: CcntactIessLcyaIty 
Microsoft. EntityF rameworkCore. Int rastructure L 18483 J 
Entity Framework Core 2.1.8-servicing-32e85 initialized 'Authenticationcontext• using provider 
To undo this action, use Remove-migration. 
PM) Update-Database 
The EF Core tools version '2.1.1-rtm-38846' is older than that cf runtime '2.1.8-servicing-32e3S' 
Microsoft. EntityFrameworkCore. Inf rastructure [184B] 
Entity Framework Core 2.1.8-servicing-32e85 initialized 'Authenticationcontext• using provider 
Microsoft. EntityF rameworkCore. Database. Command [28181 
Executed DbCommand (1, 563ms) [Parameters=[], CommandType= 'Text • , CommandTimeout= ] 
•microsoft . EntityFrameworkCore.Sq1Server• with options: None 
Update the tools for the latest features and dug fixes. 
•microsoft . EntityFrameworkCore.Sq1Server• with options: None 
CREATE DATABASE [Contact1essLoya1ty]; 
Microsoft. EntityF rameworkCore. Database. Command [28181] 
Executed DbCommand (17øms) CommandType= • Text' , 
IF SERVERPROPERTY( 'EngineEdition') 5 
BEGIN 
CommandTimeout= 68 ] 
ALTER DATABASE [Contact1essLoya1ty] SET READ COMMITTED_SNAPSHOT ON; 
END; 
Microsoft. EntityF rameworkCore. Database. Comma nd [28181 
Executed DbCommand (23øms) CommandType= • Text' , 
CREATE TABLE [ EFMigrationsHistory] ( 
[Migrationld] nvarchar(15e) NOT NULL, 
CommandTimeout= • 38 ] 
[ProductVersion] 
CONSTRAINT [PK 
120% 
Package Manager Console 
Error List 
nvarchar(32) NOT NULL, 
EFmigrationsHistory] PRIMARY KEY ([migrationld]) 
Output Web Publish Activity 

Figure 51. 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.



Figure 52. Database context declaration in Startup.cs

The line 40 in Figure 52 tells the application to look for a connection string with the key name equal to the string in quotes.

### 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 in to Contactless Loyalty.

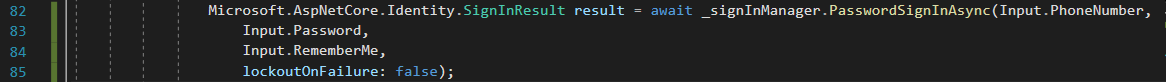


Figure 53. Sign with Phone Number

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

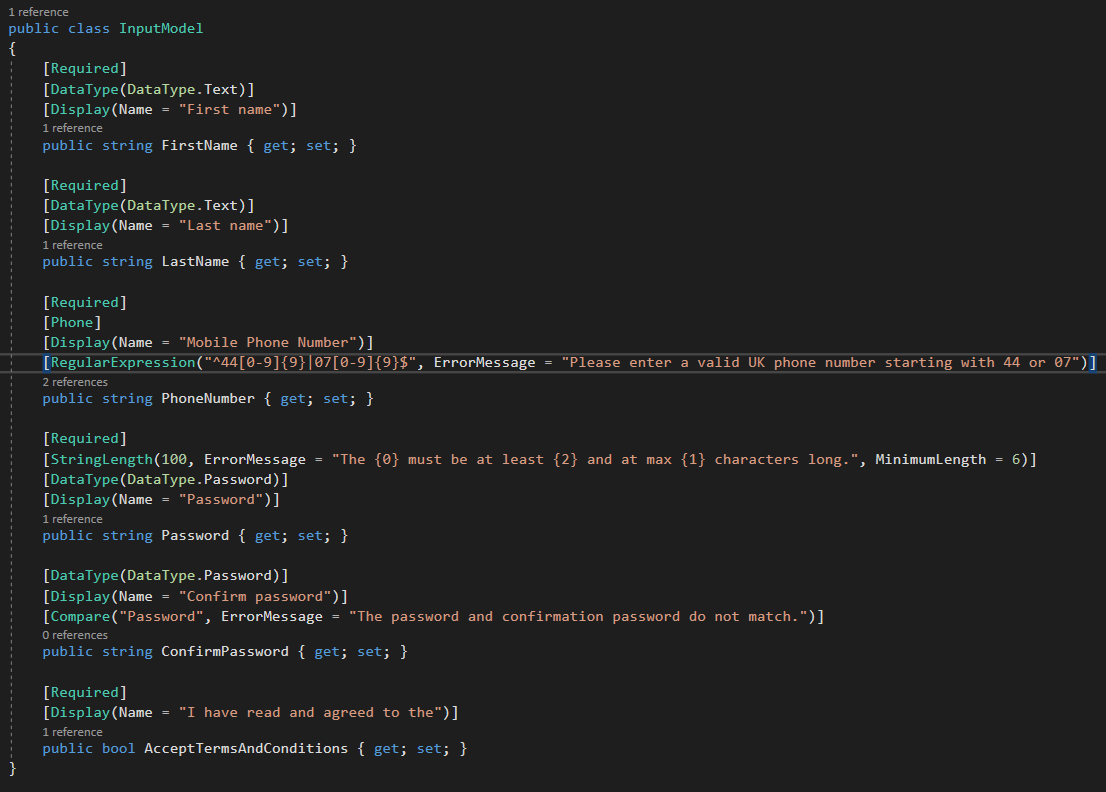


Figure 54. InputModel for the Registration Page

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 55). 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).

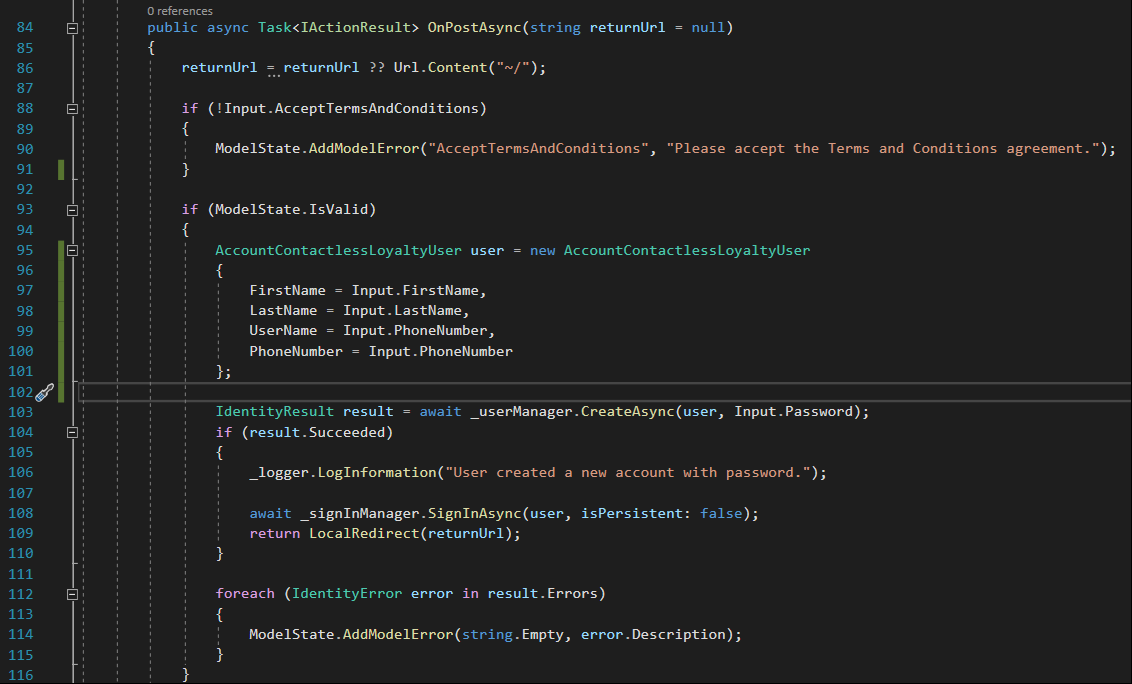


Figure 55. Registration Post Request for new user

### Creating a Loyalty Card

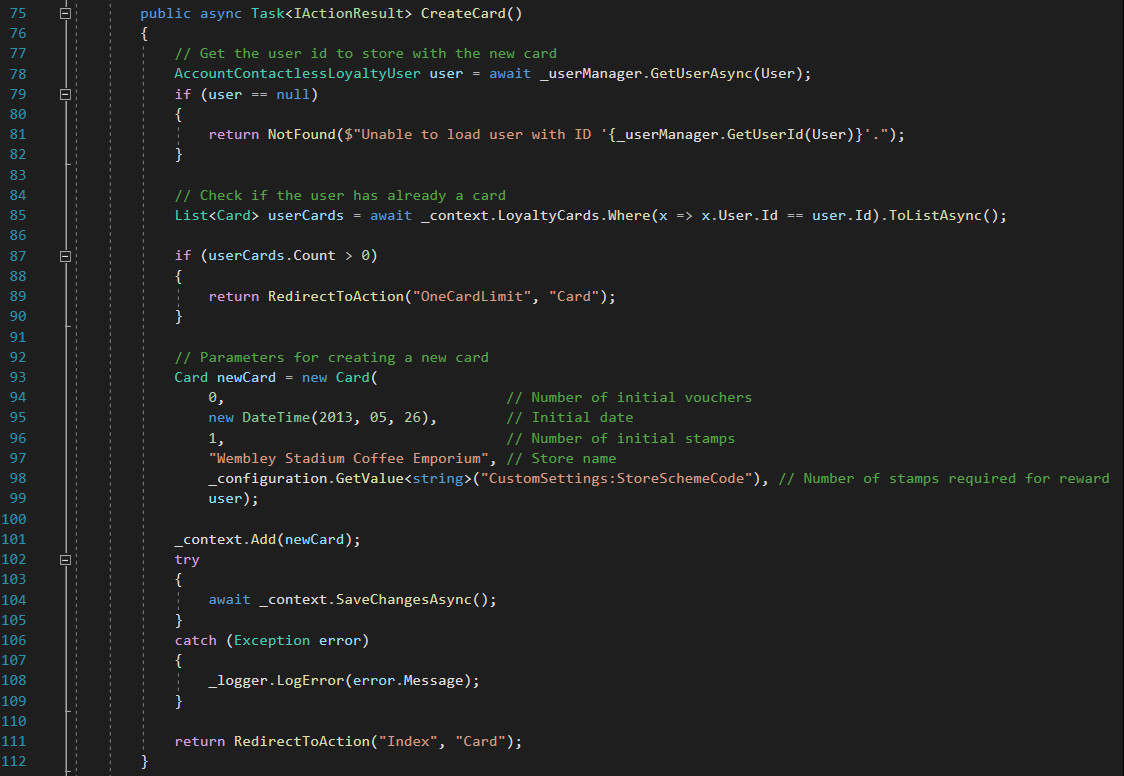


Figure 56.CreateCard method in CardController

The method shown in Figure 56 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 57, action name and controller.

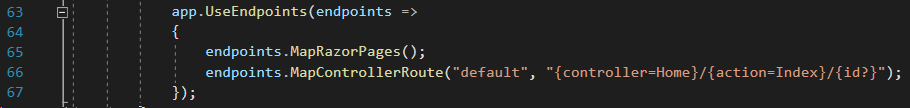


Figure 57. Definition of controller route

### 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.

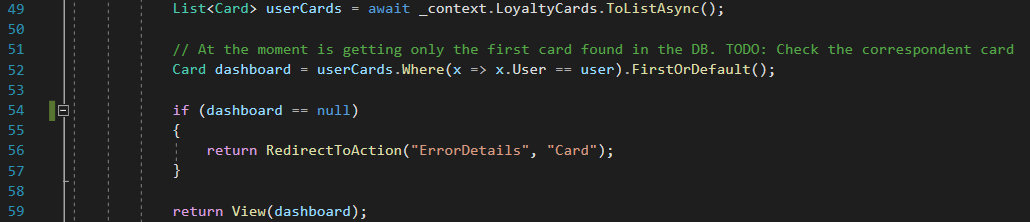


Figure 58. CardController providing data for the View through a Card Model class

On line 52 in the Figure 58, 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. This may be a problem if the user has multiple cards and wants to see the second card listed, but for now, we are not concerned by this flaw because the requirement does not involve the handling this scenario.

The View is a Razor Page that allows the use of C# (as mentioned in 5.3) to display the appropriate number of “Stamp” icons (i.e. Coffee vector images).

Line 21 and 28 in Figure 33 are the two for loops used to show the right number 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 58).

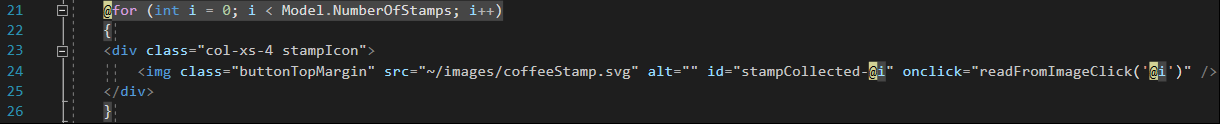


Figure 59. C# variable added in HTML image id value

### Stamp Collection and Validation

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

### i-movo API request

### Hidden Page

## Deployment

### Microsoft Azure – Web Application Setup

### Database setup

# Testing

# Conclusions and Reflections

## Current state of the project

## Limitations

## Improvements and Future Work

## Closing remarks

# Reference and Bibliography

# Appendix

## Video Demo

## Web NFC Community Group

Machine generated alternative text:
COMMUNITY & BUSINESS GROUPS 
CURRENT GROUPS 
JOINING THE WEB NFC COMMUNITY GROUP' 
REPORTS 
i 
ABOUT 
The Web NFC Community Group will create a Near Field Communication API that is browser-friendly and adheres to the Web's security model. We 
believe that means the API will not expose full, low level NFC functionality, but rather a higher level subset that is safe for Web pages, protects user 
privacy, and does not annoy users with unnecessary or complex permission requests See the Web NFC Community_GLQup Charter and the Web NFC 
specification for more information. 
SHOUYI CUI 
Employee or Representative of University 
of Westm inster 
*l confirm that I have a significant employment relationship with University ofWestminster_ (If incorrect update affiliation 
first) 
...on behalf of your organization 
W3C seeks to create Web technology that may be 
implemented by anyone Royalty-Free To ensure the 
maximum protection for implementers, we ask 
organizations to sign the Contributor Agreement; see 
the 
Please note that no Membership or fee is required to 
contribute to this group. 
I agree to participate in the Web NFC Community 
Group under the terms of the Community and 
Business Group Process / agree to abide by the 
terms and spirit of the Code of Ethics and 
Professional Conduct I certify that I am authorized 
to execute the W3C 
Agreement and hereby do execute that Agreement 
on behalf of the organization named above, and that 
all promises made therein relating to the 
Specification(s) produced by this group are 
commitments of that organization. 
*On behalf of my organization, I agree to the 
...as an individual 
In some cases, W3C accepts individual participation 
under the Contributor Agreement; see the summary 
of the Contributor Agreement Ifyou own all the 
rights to the material you may contribute, that will 
increase the likelihood we can accept your request to 
participate as an individual; read the full guidelines 
for individual requests. 
If my request is approved, I agree to participate as an 
individual under the W3C Community Contributor 
License Agreement I agree to abide by the terms 
and spirit of the W3C Code of Ethics and Professional 
Conduct To assist our evaluation, please provide the 
following information: 
*Do you own all the rights (copyright, patents) to any 
material you may contribute to this group? 
don't know 
*Does the organization named above have a 
business interest in the work of this group? 
don't know 
Optionally, use the space below to provide any other 
information you think would be helpful in evaluating 
your request For example, whether you are a 
university student or whether your employer 
supports your participation. 
*On my own behalf, I agree to the above terms. 
DIVIDUAL CONTRIBUTOR AGREEMENT 
Note All actions 
carried out 
through this 
system are 
reported by 
email to the 
relevant parties; 
copies are sent 
to w3c- 
archive@w3.org 
(Member- 
readable 
archive). 
above terms. 
SIGN CONTRIBUTOR AGREEMENT 
NAVIGATION 
HOME 
STANDARDS 
PARTICIPATE 
MEMBERSHIP 
ABOUT wac 
COPYRIGHT C 2019 W3C 
CONTACT W3C 
CONTACT 
HELP AND FAQ 
SPONSOR / DONATE 
SITE MAP 
FEEDBACK (ARCHIVE) 
UPDATES 
ERCIM KEIO BE-HANG) USAGE POLICIES APPLY. 

Figure 60. Registration into the Web NFC Community Group

## Paymo

Testo alternativo generato dal computer:
Task Name 
Task Dates 
Sep 
Project: NFC Dissertation 
sep 1 -sep 30 
PID completed 
Nov 7 
Software Completed 
Jun30 
Report Draft completed 
Apr 2 
Final Report completed 
Apr 9 
Requirement Analysis 
Sepi-Jul 10 
Jan 1 —May 31 
Feb 1 -May 11 
Oct29 - Nov 6 
Meeting personal tutor 
Sepi-Jul 10 
Oct 29 
Oct 29 
Mar 1 
-May 17 
Contacting stakeholders 
Research 
Verification, Testing and Integ... 
Junl 
-Jul 30 
feedback from . 
Jun 1 
-Jun30 
Make change to the proto... 
Junl 
-Jul 30 
Prepare for Demo 
Junl 
- Jul 30 
Testing with users and fee.. 
System Specification 
Sepi-Ju16 
sep 1 -sep 30 
Performance of the service 
Junl 
-Ju13 
Usability of the service 
Junl 
-Ju15 
Make report toexplain the... 
Apr 1 
-Jul 1 
Explain project context an... 
May 1 
-Ju16 
Review report 
Design 
May 1 
-Jul 16 
Web application impleme.. 
May 1 
-Jun30 
Notification system 
Feedback transaction result 
Junl 
-Jul 16 
Work progress and plan 
Prototyping & Implementation 
Feb 1 -Sep 30 
Prototype completed 
Jan 31 
Jun26 
Jun26-Jun30 
Jun26 
Junl— Aug 31 
Write unique binary comp... 
Read compressed data in t.. 
Confrm in Ul the tag bein.. 
details of the voucher 
Jun27- 28 
Jun22- 13 
barcode 
Maintenance and code POI.. 
logger with log data... 
Footer moving when keyb... 
CollectStamp tobe validat... 
Maintenance & Enhancement 
Jun22-Jun28 
Maintenance of the proto... 
Deliver final application 
OC t 
Nov 
Dec 
Jan 
Feb 
Mar 
Apr 
2020 
Jun 
Aug 
Sep 
OC t 
Nov 
Dec 
Jan 
2021 
Feb 
Today 
Project: N FC Dissertation 
-Requ ent An aly$s 
System Spedfkatior. 
Veïfta-.tn. -esthc and n:ecra-.tr 
Su Tl Tiar se eec:3Eck froTl 
Ni' Eke Chang -.o accordhg the feedback rece"ed 
2re:3Ere for 
2erforTlEnc 
of seri,' ce 
anc icea 
Aug 1 
May 1 
May 1 
May 1 
- sep 30 
-Jul 31 
-Jun30 
-Jul 31 
-.r nsacton resu- 
deta]s of the voucher 
Scan barcode 
Varitenance & Enha cemen: 

Figure 61. Gantt Chart produced based upon the task saved on Paymo

Machine generated alternative text:
NFC Dissertation 
University Of Westminster 
Overview 
Add Time 
Tasks 
Milestones 
Timesheets 
Add Filter • 
16 
Files 
Discussions 
@ Add 
MON 
Jun I 
15 
Activity Feed 
-lune 2020 
52hrs7 min 
1 2Ce45 Research 
1 27:25 Research 
1 22: 74 Research 
18 
DD:53 Research 
1 15:00 Meeting personal tut... 
2023 Research 
2037 Find possibIE implem.. 
+ 2 More 
0&44 Develop NFC fEûturE 
29 
3303 Back-end orchitEcturE 
min Back-end architect... 
1 0408 Back-end orchitEcturE 
1 22:76 Back-end orchitEcturE 
23•25 Web oppficûtion 
7458 Find possibIE implem.. 
7458 Tutorial Tosks to und... 
7642 Tutorial Tosks to und... 
92 m 
Cûû3 V/Eb oppficûtion 
7433 Web application 
7542 Web application 
+ 2 More 
Di3g Contocting stokEhoL.. 
m57 Research 
mog Research 
Web appficûtion 
18 
25 
07:43 Research 
19 
26 
1 7&7g Research 
1 27:48 Research 
6 21 
20 
27 
7407 Develop NFC 
7477 Work progress p... 
1 7932 Exploin project conte... 
1 02:14 Explain project conte... 
1 7626 Exploin project conte... 
1 77:37 Exploin project conte... 
1 19:15 Explain project conte... 
Month 
1 73:29 Research 
1 7529 Research 
1 7&45 DEPIoyrnE,nt prototypE 
7&46 Find possibIE implem.. 
+ 2 More 
1 04 76 User registro tion 
1 7547 User registrotion 
1 7905 User registro tion 
1 27:37 User registro tion 
+ More 

Figure 62. Example of Timesheet provided by Paymo for June

## Chrome Origin Trials

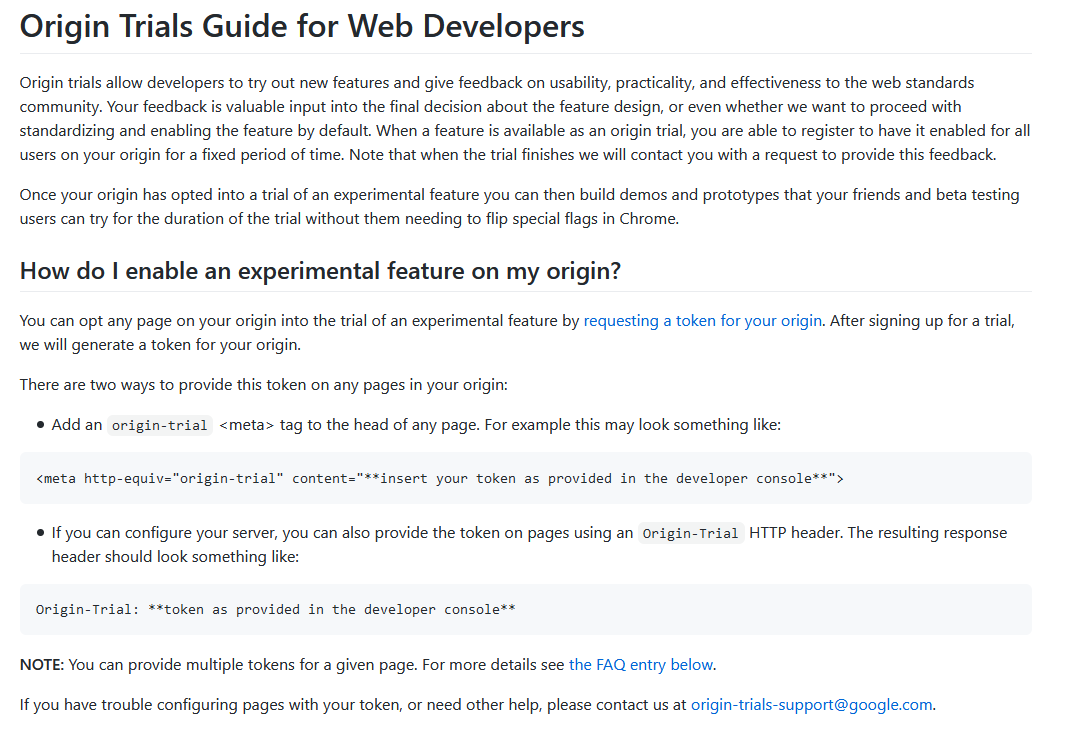


Figure 63. Explanation of Origin Trials

Testo alternativo generato dal computer:
Google 
In a few sentences please describe what your site does 
How would you categorize your site? 
Personal project 
Commercial 
Other 
How easy was it to use the feature? 
Extremely easy 
Moderately easy 
Slightly easy 
Neither easy nor difficult 
Slightly difficult 
Moderately difficult 
Extremely difficult 
How likely are you to keep using this feature? 
Extremely likely 
Moderately likely 
Slightly likely 
Neither likely nor unlikely 
Slightly unlikely 
Moderately unlikely 
Extremely unlikely 
If Web NFC werent available, how would you accomplish what you're 
trying to do? 
Do you have any suggestions for how we could improve this feature? 
E.g. "l can't tell how to use the feature to...", "the method X could do with a 
better name..." etc 
Any other comments? 
O 
Powered by Qualtrics C} 

Figure 64. Feedback form to be filled at the end of the project

Feedback on Project Report

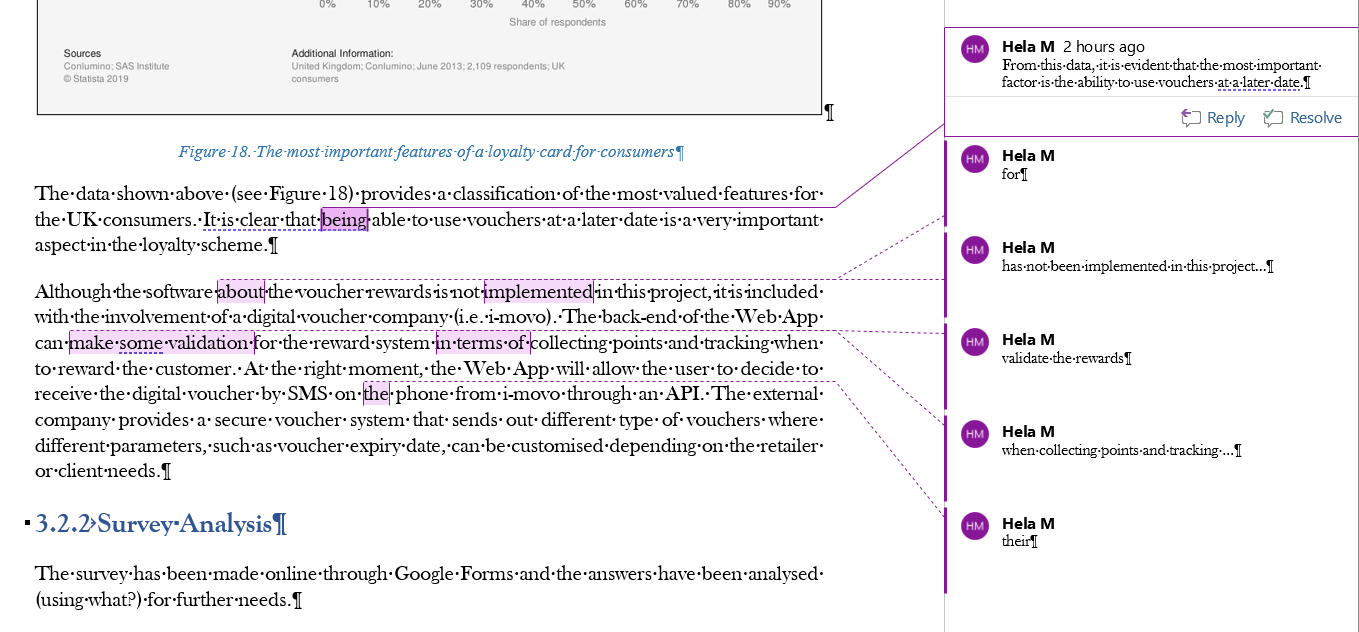


Figure 65. Comments on Project Report by reviewer

## GitHub Issue Discussion

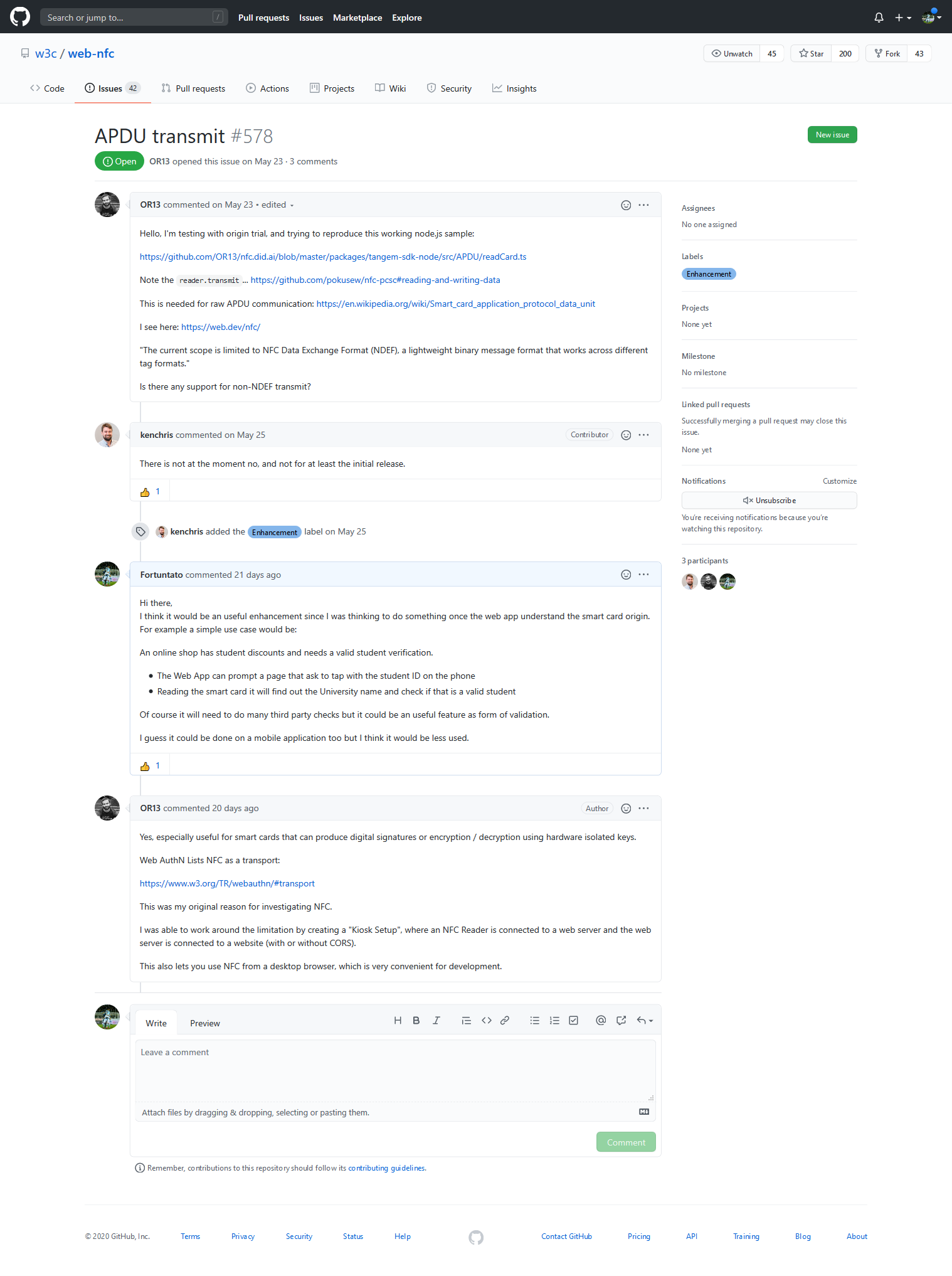


Figure 66. Discussion about future enhancement of the Web API to include APDU transmit