

---

# **Software Requirements Specification**

**for**

**Ticket Tapper**

**CA326**

**Prepared by John Duffy & Michael Solan**

**22/11/2018**

<b>1. Introduction</b>	3
1.1 Purpose	3
1.2 Overview	3
1.3 Business Context	4
1.4 Glossary	4
<b>2. Overall Description</b>	5
2.1 Product Perspective	5
2.2 Product/System Functions	5
2.3 User Classes and Characteristics	6
2.4 Operating Environment	6
2.5 Design and Implementation Constraints	7
2.6 Assumptions and Dependencies	8
<b>3. External Interfaces</b>	8
3.1 User Interfaces	8
3.2 Hardware Interfaces	11
<b>4. System Features</b>	11
4.1 System Architecture	11
4.2 System Requirements	13
4.2.1 Req 1: Pay via NFC	13
4.2.2 Req 2: Retrieve Real-Time Information	13
<b>5. High-Level System Design</b>	13
5.1 High-Level Design Diagram	13
5.2 Description of High-Level Design	14
<b>6. Schedule</b>	16
6.1 Preliminary Schedule overview	16
6.2 Task List	16
6.3 Gantt Chart	17
<b>7. Appendix</b>	17
7.1 References	17

# **1. Introduction**

## **1.1 Purpose**

The following function spec is designed to give a broad overview of the proposed Ticket Tapper Framework. This includes a listing and comprehensive description of the core functions that Ticket Tapper will need to implement to be functional in a real-world environment. To the best of our ability, we will give high-level descriptions of how these functions are expected to be designed. This documents intended audience are system designers, project coordinator, project supervisor and the CA326 demonstration panel.

## **1.2 Overview**

The objective of our project is to design a train and bus system using near-field communication. The main function of our app is for the consumer to select a route on the app then you would use the near-field communication to purchase a ticket and it would return an electronic receipt showing proof of payment. Some of our main challenges with this app idea is connecting our app to a real-time database. We hope by using Firebase it will allow our application data to be synchronized across clients and stored on the Firebase cloud. Another problem is testing but by purchasing NFC cards online we hope to be able to mimic the scanning of your phone on a bus, therefore, being able to see if our app is functional. By using flutter, we expect this app to be designed for both IOS and android greatly increasing our client base and make our app more usable to the public. We plan on testing our app by purchasing near-field communication tags off the internet which can be programmed to serve as an NFC reader that you would find on a Dublin bus and we can, therefore, test the effectiveness of our application.

## 1.3 Business Context

This product will be used by companies such as Dublin Bus as a new way to purchase tickets while riding on the bus. Our target user is someone who regularly uses public transport who wants to be able to use their phone as a method of payment and people who wouldn't be familiar with the transportation system in a different country. For example, if a tourist was to come to Dublin they would be able to purchase a bus ticket using this app instead of the frustrating search for change or the effort of obtaining a leap card. After some extensive research for this project, we found Dublin Bus state that people looking for coins to pay their fare as the second biggest factor in delayed buses, after traffic. We expect this app will help to greatly help reduce these delays.

## 1.4 Glossary

NFC (Near field communication) is a set of communication protocols that enable two electronic devices, one of which is usually a portable device such as a smartphone, to establish communication by bringing them within 4 cm (1.6 in) of each other.

Firebase - Provides a real-time database and backend as a service.

API - Application Programming Interfaces

Flutter - A mobile UI framework

DART - An object-oriented programming language used within the flutter API framework.

## **2. Overall Description**

### **2.1 Product Perspective**

This product's main purpose is to allow the user to use their handheld device as a virtual leap card by tapping their device off the NFC tag for their selected mode of transport. The user will select the route they wish to take before they board their selected mode of transportation and once the data is received by the NFC tag the payment will be deducted automatically from their account.

This should greatly improve the punctuality of services such as Dublin Bus and Bus Eireann as they both state that customers looking for change is one of the main reasons for delays.

The app will also cater for users looking for Real-Time information for their routes

### **2.2 Product/System Functions**

Below is a list of the main functions of our project. This is a preliminary list and is open to additions should we decided there are any extra functions worth adding. Each function has its own unique set of parameters, which will be outlined in more detail in section 3.

Sign Up

Create Profile

- Log in
- Add payment info
- Edit profile
- Set preferences
- Delete Profile
- Edit/Delete Payment info
- Select Route
- Journey cost
- View Real-time info

## **2.3 User Classes and Characteristics**

As this application is intended for use on public transport systems the product is accessible to anyone who owns a smartphone which is NFC compatible and uses any of the following services: Dublin Bus, DART, LUAS, Bus Eireann or Commuter Rail services. However. Our target audience would be anyone who uses these services regularly aged between 15-40, we would also target tourists who intend on using these services during the time in Ireland.

The application will be easily accessible and user-friendly to accommodate users with little knowledge of smartphone applications or the Irish transport system.

## **2.4 Operating Environment**

The Framework will require the Android 6.0 for Android users and IOS version 8.1 for iPhone users, as the base level operating systems for as these are the first operating

systems on their respected devices to be fully compatible with NFC.

## **2.5 Design and Implementation Constraints**

Below is a list of possible constraints that our team will be placed under while developing this project.

### **Time Constraints**

The project has a completion due date imposed by DCU which is April 2019.

### **Testing Constraints**

As we do not have full access to the NFC scanner which the leap card will be used on we will have to create our own testing environment

### **Software Constraints**

Our team will be developing this project using the Flutter SDK which is recently just out of beta mode.

### **Language requirements**

This application must be developed in DART as we are using the Flutter SDK.

### **Security constraints**

As this application will be used as a payment method we have to ensure this is accurate and secure.

## **2.6 Assumptions and Dependencies**

The assumptions we have made when writing our SRS are:

The NFC API that is used with Flutter is fully functionally with IOS

There are no hidden security features on the NFC tags used in association with the leap card

## **3. External Interfaces**

### **3.1 User Interfaces**

#### **Download Application**

The user should be able to download the application through their devices application store given they have the required space on their device. The application should be free to download.

#### **Download and update new releases**

Once an update has become available it should notify the user and give them an option to update.

#### **Sign Up/ Register.**

This function is the first step that a user will take in using our app. Once the customer chooses to click the register button they will be redirected to the sign-up page where they will enter their details including some mandatory pieces of



information for instance name and date of birth. This function is essential to the whole pay for the ticket idea of our project.

### **Create Profile.**

After completing the initial registration process, the user will be brought to the create profile interface. Here they will enter more information about themselves in order to create a user profile. The profiles themselves will contain information like, Name, Age and Area, Username and password.

### **Log In.**

When the user completes in creating the profile they will be then directed to the login interface. This is where they will enter their username and password so they can use the app.

### **Add Payment Info.**

Once the user has logged in the next step for the registration is to add their payment information. This is very important because it will allow the person to pay for the ticket on their phone. The user will have the option of entering their credit/debit card information or linking their apple or android pay to their profile.

### **Edit Profile.**

After registration, the user will have the option of changing their details on the app. For example, they can change their username by going to the edit profile page.

### **Set Preferences.**

Once the user has set up their profile they can set up their preferences. This means that they can add their favourite routes, for example, their way to work. Another example of this is a trip to the city centre. This allows the user to quickly pay for their favourite trips.

**Delete Profile.**

If the user decides that this app is not for them they can delete their profile off of the database. They can do this by going to the settings and clicking the option delete profile.

**Edit / Delete Payment Information.**

This allows the user to change or delete their payment information so they can continue using the app. For example, if the customer was to lose their bank card and requested another one they would be able to continue to use the app once they enter the new details.

**Select Route.**

Once the user has completed the registration and has updated their payment information they can then go on to select the route. For example, if they were to choose to travel from DCU to O'Connell Street they would chose their starting point and their destination and then choose their bus number.

**View Journey Cost.**

After they have chosen their route they will see what their journey will cost them. This will be decided by the length of the journey and the Dublin bus prices. After seeing this price the user can decide whether they want to pay for the journey or not.

**View Real-time info.**

Our users will be able to see the real-time information of Dublin Bus. They will be able to see the time until the next bus and when what bus route is coming. This will allow our users to be able to plan their root efficiently without big waits at the bus stop.

## **3.2 Hardware Interfaces**

### **Device**

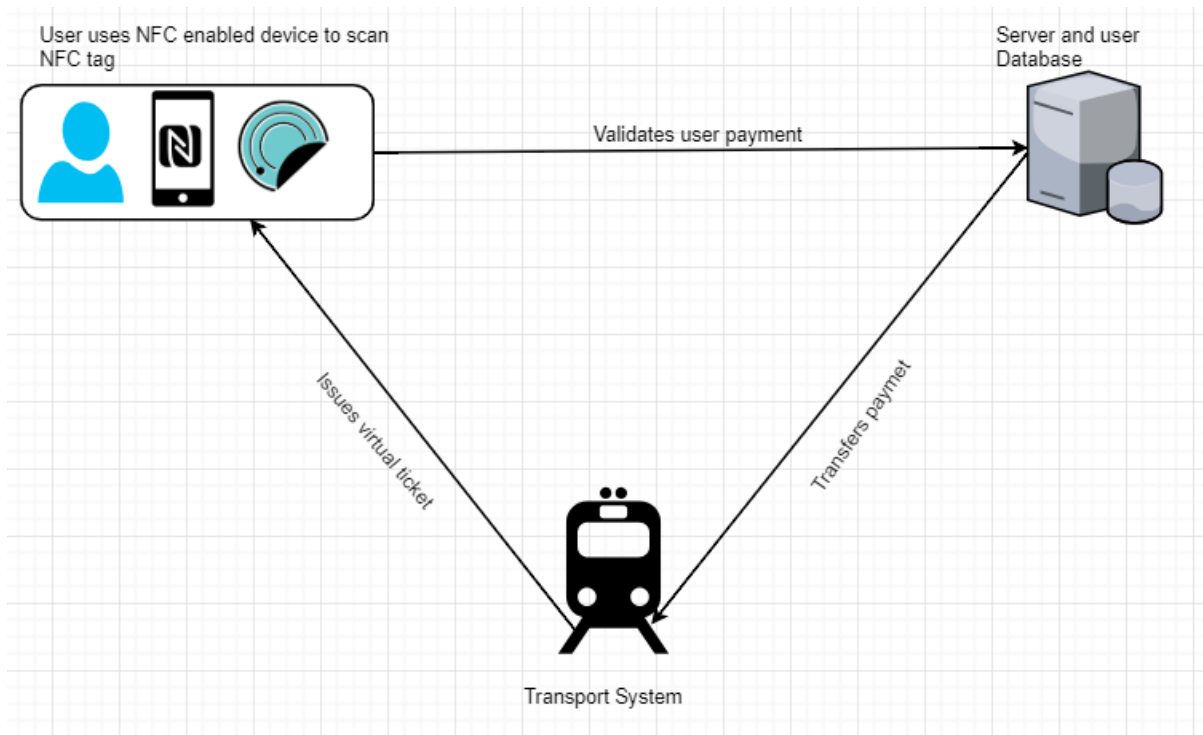
This application can run on handheld devices which are NFC compatible and have a working internet connection. Android devices need to be running the Android 6.0 or later. iPhone devices need to be running IOS 8.1 or later

### **NFC tag**

There will also need to be an NFC tag reader on the receiving end of the connection. Once the connection is made the Firebase database will be queried and once confirmed payment will be credited from the users' account. One of the main challenges with our project will be programming the NFC tag reader to simulate the tag reader you would find in a Dublin Bus. This is key for our app as it would be near impossible to test the app if we fail to successfully mimic the purchasing of a ticket on a Dublin Bus.

## **4. System Features**

### **4.1 System Architecture**



**Fig 4.1** illustrates the architecture of the product. As the diagram shows there are 3 distinct aspects to the product. The first is the app (front end) and the NFC tag. Once it is tapped off the NFC tag it will query the database to validate the payment and finally, the user will receive a confirmation of payment

## **4.2 System Requirements**

### **4.2.1 Req 1: Pay via NFC**

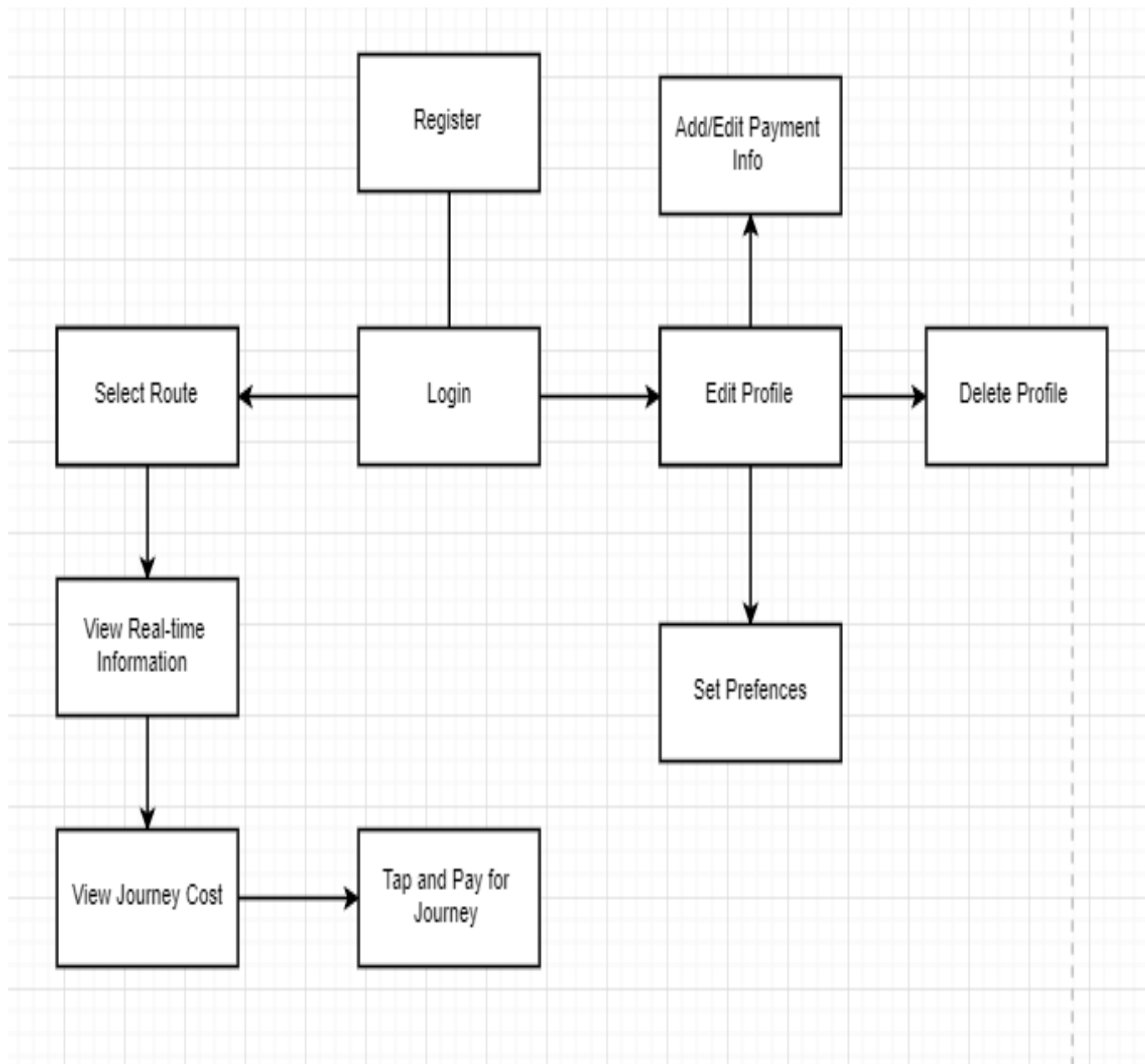
Description: This feature will allow the user to pay their train or bus fare using NFC technology. They will select their route in the application and tap the NFC tag as they board their selected transport system. This is a high priority as we aim to make it the main feature of this application

### **4.2.2 Req 2: Retrieve Real-Time Information**

Description: This feature will retrieve real-time information for transport systems using the RTPI Irish Transport System API. This feature would be a low priority as there are other applications on the market that can already achieve this.

## **5. High-Level System Design**

### **5.1 High-Level Design Diagram**



## 5.2 Description of High-Level Design

### Explanation of Fig 5.1

- **Register**
  - Add a username and password to the database to give you the ability to login
- **Login**

- Login in with your username and password combination to gain access to the application
- **Edit Profile**
  - Give user the ability to edit their personal details
- **Add/Edit Payment Information**
  - Allows the user to add or edit the payment details
- **Delete Payment Information**
  - Allows the user to edit their payment details
- **Set Preferences**
  - Lets the user add their favourite routes this allows them to quickly access these routes when inputting their journey details
- **Select Route**
  - Here the user will input their journey details
- **View Real-Time Information**
  - The user will be able to see how far their desired mode of transport is away from their location
- **View Journey Cost**
  - The user will be able to see the amount of money deducted from their account once they pay for their selected journey
- **Tap and Pay for the Journey**
  - The user will tap their device of the NFC tag which will detect the journey cost from their account and generate a virtual ticket.

## 6. Schedule

### 6.1 Preliminary Schedule overview

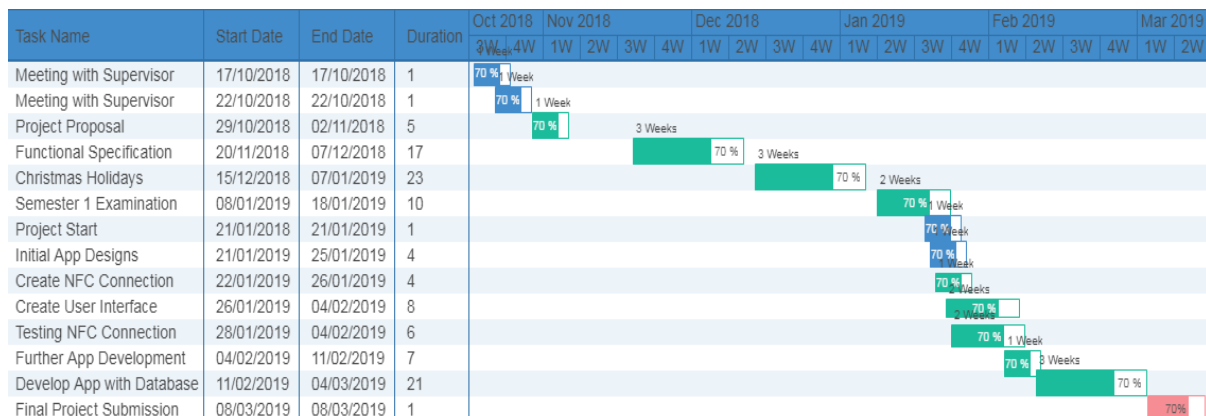
This section provides an initial version of our project timeline visualized through a Gantt chart. The task list table also identifies the tasks that need to be completed, estimated duration, start and completion date of these tasks. These dates are expected to change throughout the duration of our project due to unforeseen issues.

### 6.2 Task List

Task Name	Start Date	End Date	Duration
Meeting with Supervisor	17/10/2018	17/10/2018	1
Meeting with Supervisor	22/10/2018	22/10/2018	1
Project Proposal	29/10/2018	02/11/2018	5
Functional Specification	20/11/2018	07/12/2018	17
Christmas Holidays	15/12/2018	07/01/2019	23
Semester 1 Examination	08/01/2019	18/01/2019	10
Project Start	21/01/2019	21/01/2019	1
Initial App Designs	21/01/2019	25/01/2019	4
Create NFC Connection	22/01/2019	26/01/2019	4
Create User Interface	26/01/2019	04/02/2019	8
Testing NFC Connection	28/01/2019	04/02/2019	6
Further App Development	04/02/2019	11/02/2019	7
Develop App with Database	11/02/2019	04/03/2019	21
Final Project Submission	08/03/2019	08/03/2019	1



## 6.3 Gantt Chart



## 7. Appendix

### 7.1 References

Firebase - <https://firebase.google.com/>

Flutter API - <https://docs.flutter.io/>

Flutter tutorials - <https://classroom.udacity.com/courses/ud905>  
<https://flutter.io/docs/reference/tutorials>

Dart language - <https://www.dartlang.org/tutorials>

NFC tags - <http://www.airnfc.com/buy/nfc-tags-ntag213>

SRS templates and examples - <https://www.computing.dcu.ie/~davids/CA326/lorcan.pdf>  
<https://www.computing.dcu.ie/~davids/CA326/potool.pdf>