*3rd Year Project:*

Student Attendance Application

Functional Specification

*Authors:*

Doran, Robert

Nigel Guven

*Supervisor:*

Brian Stone

**Table of Contents**

1. Introduction
   1. Overview……………………………………………………………………….3
   2. Business Context…………………………………………………....………….4
   3. Glossary of Terms………………………………………………………..……5
2. General Description
   1. Product & System Functions………………………………………………….7
   2. Operational Scenarios…………………………………………………………8
      1. Student Scenarios……………………………………………………………….8
      2. Lecturer Scenarios………………………………………………………………9
   3. Constraints…………………………………………………………………….11
3. Functional Requirements
   1. Register user…………………………………………………………………..12
   2. Log In………………………………………………………………………….12
   3. Manipulate Database………………………………………………....………..13
   4. Passive Reader………………………………………………………...……….13
   5. Manual Data Entry…………………………………………………………....14
4. System Architecture
   1. System Design Graph………………………………………………….…..….15
   2. Student Attendance Application…………………………………………...….16
   3. MySQL Student Attendance Database………………………………….……16
5. High-Level Specification
   1. High-Level Diagram Design Student………………………………………...17
   2. High-Level Design Description Student…………………………………..…17
   3. High-Level Diagram Design Lecturer………………………………….……18
   4. High-Level Design Description Lecturer………………………………….....19
6. Project Work Plan………………………………………………………….21
7. Appendices
   1. Research Tools……………………………………………………………..…22
   2. Sites for Inspiration……………………………………………………….…..22
   3. References……………………………………………………………………..22
8. **Introduction**
   1. **Overview**

The system to be published is a student attendance application which will allow DCU staff to track the turnout of students to lectures, labs and tutorials. The application will be an independent entity of the DCU student database. Students will register their details in a section of the application and the registry team will link each student’s critical information to the unique hexadecimal ID of their student card. Each student card has a unique hex identification number which will be their unique identifier or primary key on the student database. The ID cards are NFC/RFID compatible. This allows them to be manipulated by technologies that can utilise NFC. Most modern smartphones such as Android and iPhone can communicate with other devices using NFC. The application will be built on Android as it owns 86.1% of the mobile phone operating system market. They have an open source application builder called Android Studio which is designed to encourage independent developers to create a plethora of software for Android devices such as tablet computers, phones and cameras.

The other section of the application will contain the administration of the application. This will contain a contactless reader which the staff member can use to open a new attendance log for a given period. When a class begins, the staff member may turn on the application and utilise the passive function which reads in student card numbers onto the log. Students must bring their cards to class and swipe their cards on the device. This will read that they are present for the period. Student who do not have their cards with them may be entered in manually. This will be derived from another function.

* 1. **Business Context**

There is a possible business context that would be useful for this attendance application.

* Selling the Application:

The product could be used by Dublin City University. This would be helpful in tracking how many students attend lectures for each day which could be very helpful for statistical purposes. It could also be revised for another university or first/second level school.

* 1. **Glossary of Terms**

***Hexadecimal:***

Hexadecimal is a numeral system with a base of 16. The common numbers are *0-9* with included *A-F* which represent *10-15.* For RFID cards, the unique Hex number is 32 bits e.g. BD EE 5A 71

***NFC:***

NFC or Near Field Communications is a communications protocol that enables two compatible electronic devices to communicate by bringing them within an approximated distance of each other

***RFID:***

RFID or Radio-Frequency Identification uses radio waves to read information stored on an object. This information is relayed to the device that is transmitting the RFID signal.

***Android:***

Android is a mobile OS which was developed by Google and based upon the Linux Kernel. It is specifically designed for handheld devices like smartphones, smart watches and tablets.

***Operating System:***

An operating system is software that can manage a computer’s hardware and software resources. The OS uses multi-tasking to schedule tasks and allocate memory to various applications.

***Android Studio:***

Android Studio is the official IDE for the Android operating system. It comes included with an interface design package and Android SDK.

***Tablet Computers:***

Tablets are portable computers which generally come with touchscreen display and a rechargeable battery. Tablets have some benefits over desktop computers and laptops but do not include as powerful hardware.

***Contactless:***

Usually associated with making payments via credit/debit cards, contactless systems use RFID and NFC to read information stored between two end-to-end objects and the contactless device acts as a ‘middle man’ to communicate between each object.

2.  **General Description**

2.1 **Product / System Functions**

*User functionality System*

Firstly, the students will have to download the Student Attendance App from the Google Play Store. After the students have successfully downloaded and installed the Student Attendance App, they will have open the Application furthermore click into NFC reader to scan their student Identification card. After they scan their Student Id card they will receive their unique NFC hexadecimal number unique to their ID card only. After the students receive their NFC number they will have to input the NFC number as well their Student Identification, Student Name, Student Year and Student Programme.

After the students have entered their details they will have register for the modules that are participating in the Student Attendance Application. When all the student’s details have been entered it will be submitted to a SQL database to store along with all the other Students details. The database will have a record of all student’s and will organize them by the modules they are in.

*Staff involvement with the system*

Staff will also have to download the Student Attendance App on a NFC compatible phone on the Google Play Store. After the User downloads the Application, they will register as a lecturer and set up their Module with a Module name and Module code that the students will later register for. Staff will have administration rights which will allow them to see/edit the attendance record for each individual student registered for their module. Staff will also have a record of the student’s name, student id number, student year and student programme. The lecturers will at the start of each class they want to take attendance for click into the specific module and enter a name, date and time of the class. After the lecture has entered all the specified details the Application will activate the phones NFC scanner. After the NFC scanner has been activated when each student comes into the class they will have to present their student card to the lecturer who will scan the student card. After a student card has been scanned three different messages can appear “Student Attendance Logged”, “NFC tag has not been registered to a student yet” or “Card not read correctly please try again”.

*Web-interface functionality*

The main appearance of the Student Attendance Android App will be made using Android studio. We plan on using the Android Studio because it is the best software for Android Application development in terms of usability and support. As well as the Android Studio we are going to be using an Android Emulator instead of having to install the Application onto an Android device for any changes that we make. The databases will be developed using SQL.

**2.2 Operational Scenarios**

Our proposed Application has 2 main Users which are student’s and lecturer’s. There are different operational scenarios for both. As both student’s and lecturers will have access to different parts of the application, their scenarios differ greatly.

**Student Scenarios:**

* *Registering Student Id Card:* The User will have to download the application from the Google Play Store. After the User has successfully downloaded and installed the Application on their NFC compatible Android device they will open the application and click into “Register Student ID” and finally “NFC scan”. When the NFC scan is opened the User will put their valid student Id card to the NFC scanner on the Android device which will then automatically enter their unique hexadecimal tag number into NFC information slot. The User will then be prompted to enter their “Student Name”, “Student Number”, “Student Programme” and “Student Year”. Lastly the student will have to register for the modules that are participating in the Student Attendance Application. They can search for the modules by either the module code or module name. The user will then request access to the said module/ modules and wait to be accepted/rejected.
* *Changing NFC Tag:* In the case that User has lost their original student Identification Card and had to get a new one. This would mean that the unique NFC tag would be different for this card. The student would have to contact one of their participating lecturers to change the NFC manually as they are the only Users with authorization to change any details on the student’s information.

**Lecturer Scenarios:**

* *Signing up as a lecturer:* The User would have to download the application on a NFC compatible Android device. Following the successful installation of the Application the lecturer will have to click “Sign Up as Lecturer”. The user will then be prompted to enter the following details “e-mail”, “password” and “Lecturer Name”. After registering the lecture can register the modules they wish to interact with the Student Attendance Application for by clicking “register modules” and entering the number of modules they wish to register. Following the number of modules that the lecturer has entered they will be prompted to enter a Name and Module code for each individual module. This will register each module on the database and allow students to request to join said module.
* *Checking Attendance for a Module:*The User will have to open the Application. Following opening the application they will have to log in with their username/ password. Afterwards the User will click into “Modules” and thereafter click show/change Attendance. Furthermore, a table will come up that will be organized firstly by Attendance best to worst and finally Alphabetically for any of those who have attended the same number of classes.
* *Taking Attendance for a Class:*The User will have to log into the Application with their unique username/ password. The lecturer would then click into “Modules” furthermore the User would click into “Take attendance”. The lecturer would be prompted to enter a date, time and name for the specified Module. Furthermore, the NFC scanner would then be activated on the phone with a message appearing “Please scan a valid Student ID”. The lecturer would then take individual Student ID cards and place the card on the NFC Scanner. Three possible messages can appear after a card has been attempted to be scanned 1.” Student Attendance Logged” 2.” NFC tag has not been registered to a student yet.” 3.” Card not read correctly please try again”. After a student’s attendance has been logged it will automatically update a student’s attendance on the database for the specific module by 1.

**2.3 Constraints**

The main constraint’s that our project faces are 1. Time 2. Storage 3. Number of User’s.

*Time Constraint’s:* As the deadline for the 3rd project is in early March it will be difficult to complete this project to the highest possible standard. As we believe this is a project that can continually improve.

*Storage Constraint’s:* The main storage constraint that we would face if this project was implemented for many Modules/Lectures is the storage that we possess on the database. As for some Modules/Lectures there can be upward of 100 students. Of course, this all depends on how many Users we manage to utilize our Student Attendance Application.

*Number of User’s Constraint’s:* The number of Users will be the main constraint we may encounter unless we convince a lecturer to implement our Application in one or more of their Modules. Otherwise we will have to set up the User’s with student’s we know inside of the college to be our students to check the functionality of the Application. We would prefer if we could launch the application on a larger scale as we believe it is very practical and efficient way to Automate student attendance.

**3 Functional Requirements**

**3.1 Register User**

* Description

This is the first function which must register a student to the database to track their attendance. This will be a button on the interface which links to a registration form. The student must submit the relevant information i.e. student ID, name, DCU email, course code. Once this information is submitted, the registration team will link the student card HEX id to the student on the database.

* Criticality

This function is essential as there must be some way to create a new object to be placed into the database. Once this function has created a new student object, it can be tracked when using the NFC reader set to a standard schema by the administrator.

* Technical Issues

The main issue to arise is if a student may be accidentally created twice or more. This would produce a performance problem on the database.

* Dependencies

None.

**3.2 Log In**

* Description

There will be a login button on the primary interface which will allow the administrators to sign in to control the application. The admin or lecturer is given these special rights which will allow them to create a new session to track attendance.

* Criticality

It is crucial that there is some way that administrators are only given access to the database and attendance logs. This allows only privileged members to alter data on the application.

* Technical Issues

There may be dealings where an administrator is locked out of their account which will have to allow for the changing of passwords.

* Dependencies

The log-in is dependent upon the registration of users. This log-in is only accessible if a user is granted special privileges.

**3.3 Manipulate Database**

* Description

Database manipulation is very important for the application. This function must be able to search, edit and remove students from the database given the circumstance. This will be all done using the SQL database query language.

* Criticality

Having a database that is organised and logical is extremely important. A well-planned database can provide the administrator with up-to-date, accurate information. The primary keys for our database are entirely unique. Normalisation should be applied to examine if the tables have a strong structure.

* Technical Issues

If the database is corrupted or has missing information, the SQL query that a user might send will produce an error. This could be fatal if the database becomes very large and cumbersome to maintain.

* Dependencies

The manipulation of the database is dependent upon inputs and outputs of objects. Therefore, any functions that handle data transmission i.e. user registration, passive NFC reader, must be reliable for manipulation to work effectively.

**3.4 Passive Reader**

* Description

The passive reader is a function which will be ready to read RFID chips on student cards. There is a button within the administrator tab which when clicked will turn on the reader. Students swipe their cards onto the phone which reads the hex identification and first checks if a student is on the database, then either acknowledging their attendance or prompting a negative response

* Criticality

This function is important as it reads in the information which allows the application to function as an attendance module. The RFID reader allows for a non-laborious way to check if a student is attending a class.

* Technical Issues

Such issues may arise. For example, a student who may have registered incorrectly will prompt a negative card-read response. Also, if the NFC reader is faulty on the device, then it may not read the RFID correctly.

* Dependencies

This function is dependent upon the database as it must read in a card and then check the hex id and match it to some object contained within the database.

**3.5 Manual Data Entry**

* Description

If a student does not have their card, they must be noted into the attendance log manually. While this is more laborious, it is important as it may skew the analytical side of the application against the attendance of students.

* Criticality

Manual data entry is placed into the application as a stand-by function just in case there is a technical issue arising from the NFC reader or missing student card from the user end.

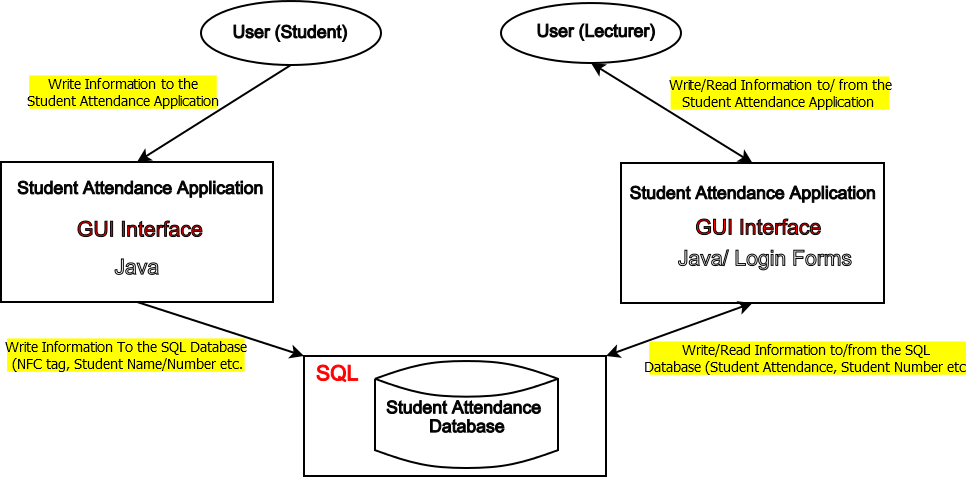
* Technical Issues

Where the NFC reader malfunctions, a manual entry function must be put into place. In this way the application can function albeit with more time consumption.

* Dependencies

The manual reader is dependent upon the database as the hex id of the student card must match the primary key on the database. Users must enter the information correctly which must contain reliable and necessary information

**4. System Architecture Design**

****

**Figure 4.1** above displays the Architecture of our proposed project. As the above diagram displays there are five distinct aspects of our projects architecture. 1. User(student) 2. User(lecturer) 3. Student Attendance Application (The side of the project the student uses, can only write data to.) 4. Student Attendance Application (The side that the lecturer interacts with, can read and write data from/to) 5. The SQL Student Attendance Application Database.

**4.2 Student Attendance Application**

The Android Application is the front end of our Project. It is where both Users will be interacting, writing and receiving information from. Students can only write information to the Application, as in giving information such as NFC tag, Student number, Student ID, Student name and Student Programme. Apart from this the user (student) will not have much interaction with the application.

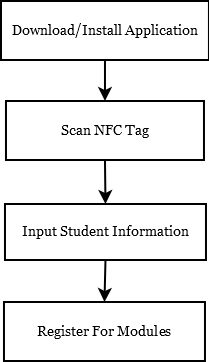
Lecturers can both write and read information to/from the application. The users will be able to access student’s information (i.e. Student Attendance, number, NFC tag etc.) they will be able to modify this information if needed.

**4.3 MYSQL Student Attendance Database**

The database relates both directly and indirectly to every aspect of the of the Project Architecture. The database will store information sent to it by the Android application, it will also send information for the lecturers to read and edit if needed. So, the information that the Lecturers are seeing is stored on the SQL database sent to the Application and formatted by java on Android Studio for the users to view.

**5. High-Level Specification**

**5.1 High-Level Diagram Design Student**

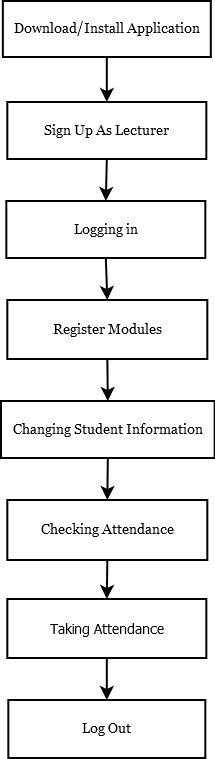


**5.2 High-Level Design Description Student**

Diagram 5.1 is explained below.

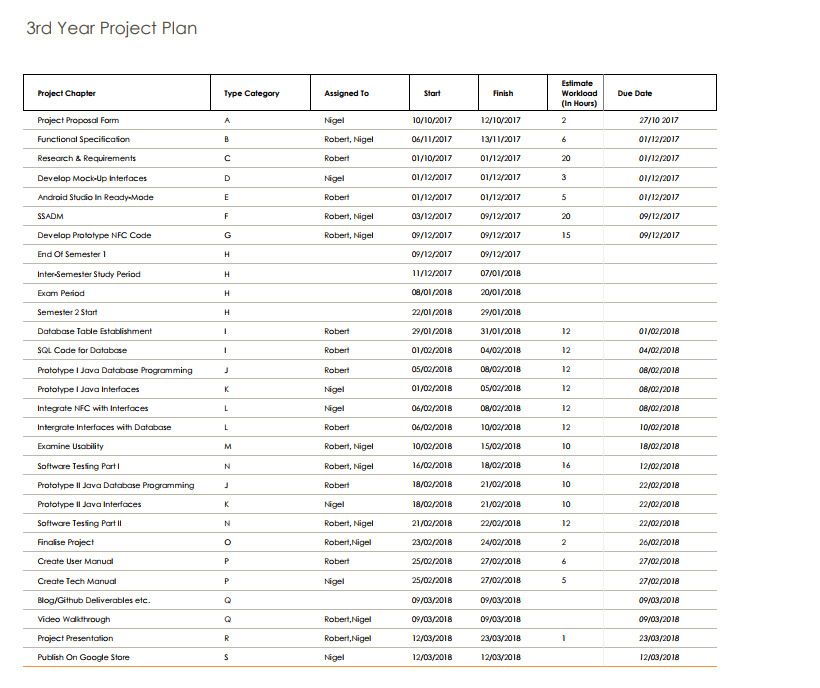
1. *Download/Install Application:* Download the Application “Student Attendance” from the Google Play Store.
2. *Scan NFC Tag:* Open the Application, the user will click into “Register Student ID” and finally “NFC Scan”. This will prompt the user to put their valid Student ID card up to their NFC scanner on their device. This will get their unique NFC hexadecimal tag.
3. *Input Student Information:* After the NFC scan is complete the User will be prompted to enter their student card details. The NFC tag will be inputted automatically. The user will have to input their “Student Name”, “Student ID number”, “Student Year” and “Student Programme”.
4. *Register for Modules:* When the User has inputted all of their Student details they will be asked to register for the Modules that are using the Student Attendance Application. The user will be able to search for these modules by “Module Name” or “Module Code”. When the user finds the desired module, they will click “Register”.

**5.3 High-Level Design Diagram Lecturer**

****

**5.4 High-Level Design Description Lecturer**

Diagram 5.3 is described below.

1. *Download/Install Application:* Download the Application “Student Attendance” from the Google Play Store.
2. *Sign Up as a Lecturer:* Open the Application, click in “Sign up as Lecturer”, the user will then be prompted to input “E-mail”, “Password” and “Lecturer Name”. After inputting all the users details they will click “Create Account” and will be logged in as a lecturer.
3. *Logging in:* Open the Application, click in “Log in” the user will then have to input their e-mail and password. Following a successful input of e-mail and password the user will click “Log in” and be successfully logged into their account.
4. *Register Modules:* After successfully logging in the user will click “Register Modules”, where they will be prompted to input the number of modules they wish to register. After inputting the number of modules, the user will have to input a “Module Name” and “Module Code” for each unique module. The user will click save to successfully register the modules.
5. *Changing Student Attendance:* After the user has successfully logged in they will click into “Modules” and find the module in which they want to change a student’s information (Attendance, Student Number, NFC tag etc.). After clicking into the desired module, they have two choices “Take Attendance” or “Show/Change Attendance”, the user would choose the latter. A table will show up with all the students information, the user will then click into the search bar at the top and input either the “Student name”, “Student ID number” or “NFC tag”. The user will then be brought to the Student Information in question and can change any of it as they have admin access to this table.
6. *Taking Attendance:* After the user has successfully logged in they will click into “Modules” and find the module in which they wish to take attendance. After clicking into the desired module, they have two choices “Take Attendance” or “Show/Change Attendance”, the user will click the former. The user will be Prompted to enter “Class Name” (i.e. lab week 2), “Date” and “Time”. Afterwards the NFC scanner would be activated on the device with a message appearing “Please scan a valid student ID”. The lecturer would then scan the student’s individual student ID cards with three possible outcomes. 1.” Student Attendance Logged” 2.” NFC tag has not been registered to a student yet.” 3.” Card not read correctly please try again”. After a student’s attendance has been logged it will automatically update a student’s attendance on the database for the specific module by 1.
7. *Logging Out:* The user will go back to the home screen where they will see their name in the top right corner. After clicking on their name an option to “Log out” appear, which they then click and will have successfully logged out.
8. **Project Work Plan**

**7. Appendices**

**7.1 Research Tools:**

* <https://www.androidauthority.com/what-is-nfc-270730/>
* <https://www.youtube.com/watch?v=-TwwqdDTzfU>
* <https://www.youtube.com/watch?v=cZwLIxM7PdE>
* <https://www.androidauthority.com/android-studio-tutorial-beginners-637572/>
* <https://developer.android.com/studio/index.html>
* <https://www.w3schools.com/sql/>
* <https://code.tutsplus.com/tutorials/learn-java-for-android-development-introduction-to-java--mobile-2604>

**7.2 Sites for Inspiration:**

* <https://www.leapcard.ie>
* <https://sumup.ie/air-nfc-card-reader/>
* <https://www.visa.ie/pay-with-visa/mobile-contactless-payments>

**7.3 References:**

* <https://github.com/nadam/nfc-reader>
* <http://www.homeandlearn.co.uk/java/java_and_databases.html>
* <http://www.umlet.com/>
* <https://www.office.com/>
* <https://stackoverflow.com/questions/12159215/rfid-based-reading-and-writing-in-java>