



---

# CollegeEssentials - The app for students

---

**Aaron Flanagan**

**Ciaran Brennan**

B.Sc.(Hons) in Software Development

MARCH 24, 2016

**Final Year Project**

Advised by: Damien Costello

Department of Computer Science and Applied Physics  
Galway-Mayo Institute of Technology (GMIT)

# Contents

<b>1</b>	<b>Introduction</b>	<b>6</b>
<b>2</b>	<b>Methodology</b>	<b>9</b>
2.1	Planning . . . . .	9
2.2	Software Methodology . . . . .	9
2.3	Testing . . . . .	10
<b>3</b>	<b>Technology Review</b>	<b>12</b>
3.1	XML . . . . .	12
<b>4</b>	<b>System Design</b>	<b>13</b>
<b>5</b>	<b>System Evaluation</b>	<b>14</b>
5.1	Memory and Server . . . . .	14
5.2	Universality . . . . .	15
5.3	Improvements . . . . .	16
<b>6</b>	<b>Conclusion</b>	<b>18</b>

# List of Figures

2.1	Project Work Flow . . . . .	11
5.1	Heroku Connection Diagram . . . . .	17

# About this project

**Abstract:** "Tech will transform from something we actively use to a more seamless integrated experience that is 'on' all the time." Daniel Baek, Co-founder of Nodes [1]. This paper proposes a new phone application that consists of the most important activities and features in demand for third level students around in Ireland. Feature's that can be accessed 24/7 and effortlessly by any user. The aim of this project was to create a suitable application that would benefit users in all aspects of their studies. Studies show that each household has an average of six devices connected to the internet. These consist of smart phones, laptops, computers and some household appliances. It was decided that this project would be developed for the most portable device held by users, mobile phones. Current deals with network vendors also allow the user to be connected to the internet through 3G or 4G technologies. All colleges maintain their own network for students also, this means that our application can be downloaded and used at all times.

**Authors:** This project was developed by two 4th year development students, Aaron Flanagan and Ciaran Brennan, as part of our Bachelors of Science honours degree in Applied Software Development.

# Acknowledgements:

We would like to acknowledge and thank Damien Costello for supervising this project. His expert advise and support helped guide this project to completion. We would also like to acknowledge the Department of Computer hnology for making this possible and helping us by providing support and any tools we needed during the development.

# Chapter 1

## Introduction

The original idea for this project was the development of a facial recognition music application. The application would begin by capturing an image of the users face and running a mood capturing algorithm to determine the users mood, and it would then generate a play-list based on the users mood. This projects foreground was going to include an animated graphical user interface through the use of the JavaFX library made available by Oracle [2]. The background was running a facial detection program that was described by the OpenCV Team [3]. This algorithm was going to build upon to include facial recognition as it only produced facial detection on screen by drawing shape around the users face, but this needed a feature extraction algorithm to complete it and that's when the idea hit a wall. After many weeks of research and study into the different approaches of facial recognition and feature extraction it was found that there was no Java binding libraries to OpenCV or resources that we could use. After a week of searching nothing was found that could help us, but because it was still early into the year the project idea had to be changed as to avoid further problems and time delay.

It was now October/November and the project was still at the beginning, GMIT hosted a hackathon for other colleges with software development courses to develop a project over a two day period. This time was used to explain the problem and come up with a new idea and build upon it until a clear project goal and list of possible features where decided on. Damien Costello was informed about the problem and advised the project idea could be changed while it was still possible, he was understanding and suggested a few ideas of projects to think about and do some research into. Out of those ideas came our student application idea, the development of an integrated mobile application solely for the purpose of third level students. He suggested doing something in augmented reality. It was thought upon and decided that the projects goal would be to build an application for students and include

the augmented reality section as navigation around the college. A list was constructed of things that tend to be a problem for students and academic staff managing multiple years of students. Further research was done into the market to see what was already done and available. Many developers have developed applications with timetables or notes for your phone screen but we couldn't find any that contained all of these sections integrated into one application, or that were just institution specific. After deciding what kind of application was to be built, the platform technology had to be chosen, this included research into the available systems. Finally a list of features had to be narrowed down to include in the project, what features would be more beneficial and would provide better context to the final application.

Further into this report will contain a breakdown and give an in-depth explanation to how the project was approached and developed, the different software and technologies used, an evaluation of the final product and notes taken by the developers about each feature and where they can be improved and why. The following sections will be discussed about the application:

- **Methodology:** This explains the production approach taken to developing the individual features of the application and how they were developed within the project time limit.
- **Technology review:** This section discusses the various technologies used, both hardware and software. It explains how the project uses each component and gives an explanation to why this particular system/software was used and what it was used for.
- **System design:** This section gives a breakdown of the project's architecture. How the system is built and how everything fits together.
- **System Evaluation:** This section evaluates each feature of the application and compares the finished project to the original concept first described.
- **Conclusion:** The final discussion about the project and evaluation of its finished state. Each section will be briefly concluded and its main points will be stated.

This project can be found on Github under the name AaronFlanagan20 or CiaranB1992: <https://github.com/AaronFlanagan20/CollegeEssentialsApp>. The repository contains a small ReadMe file about how to download and run the application. The repository holds all the classes and native android files created and used by Android Studio to compile and run the project. Your computer or laptop must support virtualization, this can be checked

by attempting to install Intel's hardware accelerator software or by checking for in the task manager of your machine. If said machine doesn't support it, you will be unable to run the emulator, instead you must plug in an android device with the developer options enabled.



# Chapter 2

## Methodology

### 2.1 Planning

The first process of the project was planning everything that had to be done. This included when to hold meetings, what software to use for the development, what methodology to approach it with and how to test it appropriately. The first task finished was the creation of a project work flow to create a timely estimate of how long each feature will need and take into account other responsibilities the developers might have with other projects and reports. The project work flow or Gantt chart, was developed to show everything that needed to be done. It leaves out the section that assigns each individual developer to a particular area or feature because the list of features was not particularly large enough to scale out seven months of development and it was more beneficial to work as a pair due to the methodology taken. See Figure 2.1.

### 2.2 Software Methodology

Next was the decision of which methodology to pursue. It was decided this project would be developed based on Agile methodologies. "Agile methodology is an alternative to traditional project management, typically used in software development. It helps teams respond to unpredictability through incremental, iterative work cadences, known as sprints " [4]. The traditional waterfall approach was not taken due to its limitations put on each phase of the project. Each section had to be completed in sequential order and time did not permit backtracking if a problem arose. Instead an incremental approach was taken. Both developers met up to discuss which feature to work on and to begin working on it by researching previously made work

and reading through the Android documentation to find relative libraries available. This meant that each feature could be developed no matter how long it took to finish and different approaches could be taken. One feature may have taken a week at most to develop allowing more time to improve it, while the next might have taken a month to complete. It also allowed more research and the ability to develop up-to date documentation and code during the development of each feature. If the code was re-factored or a different approach was taken that differed from the pre-determined plan specified at the beginning of this project, the developers where free to make the change and it was documented. During the development of that feature however, an iterative approach was taken. Each developer would plan, develop and test the feature and resort back to the original documentation to evaluate it. The same process would be taken again until the feature was eventually finished. At the end of each features cycle a meeting was held to discuss and evaluate it with the supervisor. These consisted of ideas on how to improve it or how to begin on the next.

## 2.3 Testing

At the end of a features cycle, some testing was performed before moving on. This project used three forms of testing. The first form of testing is gradle, that is built into Android Studio. Gradle is a project build automation tool used to compile, test and run a native Java project and slow down build time and code freeze ups [5]. It compiles and runs the project within Android studio. Next used was CircleCI, CircleCI is a cloud platform used for continuous integration. Every time a commit was pushed up to the repository using git, CircleCI would pull down the code from Github and run a selection of tests on the new changes. If it passed the test the commit would integrate with the rest of the project on Github, if failed it would not be integrated until the problem was fixed via another commit to update or a revert to remove the new changes. Finally the developers wrote tests for the application using JUnit. JUnit is "A unit testing framework which is a central element of the Extreme Programming (XP) testing practice." [6] JUnit was used to test the actions of components in the application to see if they performed the way they where suppose to, example if a pop-up box is meant to display on the press of a button.

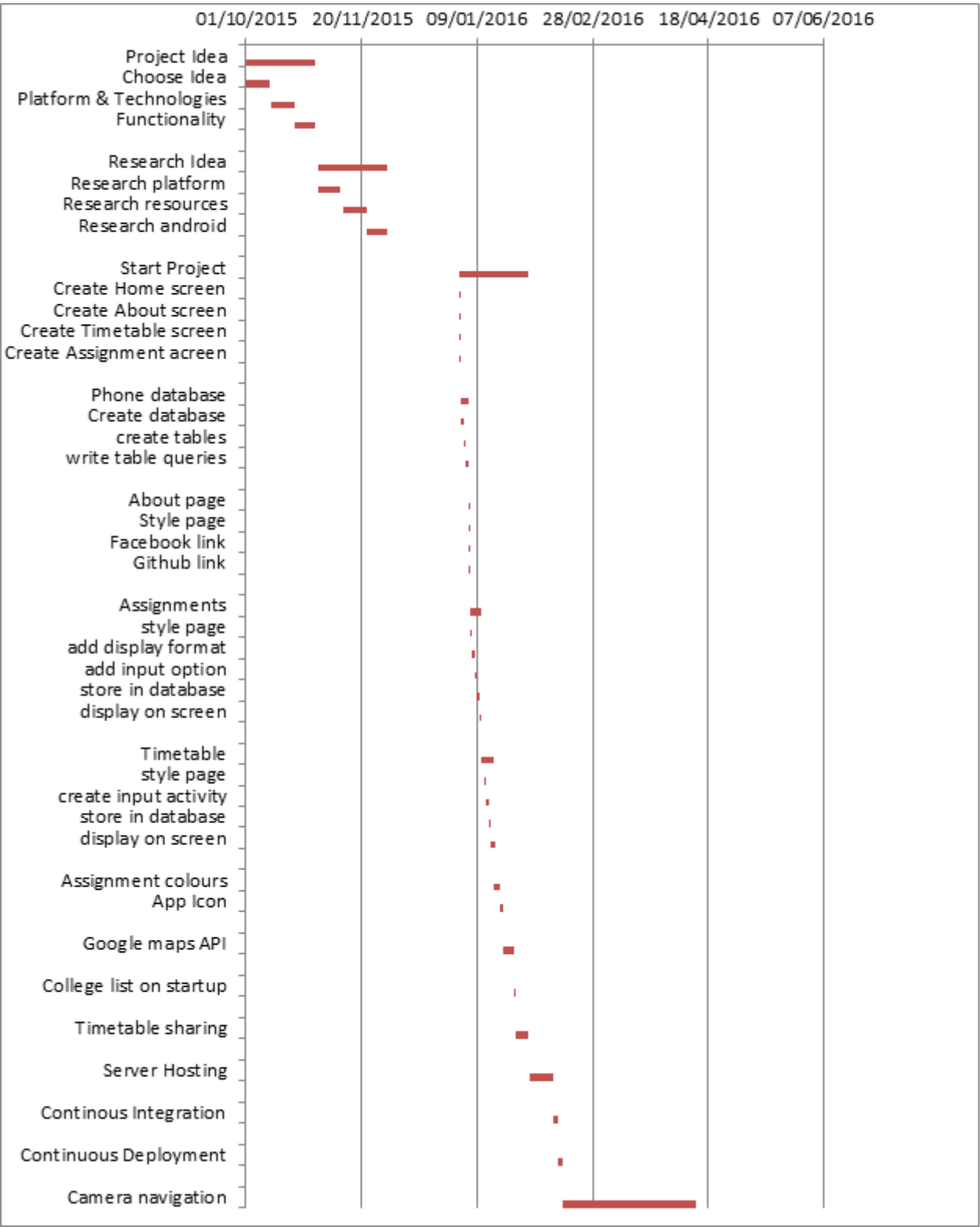


Figure 2.1: Project Work Flow

# Chapter 3

## Technology Review

About seven to ten pages.

- Describe each of the technologies you used at a conceptual level. Standards, Database Model (e.g. MongoDB, CouchDB), XML, WSDL, JSON, JAXP.
- Use references (IEEE format, e.g. [1]), Books, Papers, URLs (timestamp) – sources should be authoritative.

### 3.1 XML

Here's some nicely formatted XML:

```
<this>
  <looks lookswat="good">
    Good
  </looks>
</this>
```

# Chapter 4

## System Design

As many pages as needed.

- Architecture, UML etc. An overview of the different components of the system. Diagrams etc... Screen shots etc.

Column 1	Column 2
Rows 2.1	Row 2.2

Table 4.1: A table.

# Chapter 5

## System Evaluation

### 5.1 Memory and Server

The database management system used is Sqlite. This was chosen because all mobile platforms support it. This enables the phone to have a certain degree of control on what information to get when needed. Once the user selects a college on start-up of the application a database with the college name is created on the phone allowing each selection to have it's own database to store a timetable, assignments and custom google map markers the user may have placed. The database model is quite simple. It stores two tables Timetable and Assignment. Timetable contains three columns room, teacher and subject, while Assignment contains two, subject and due-date. With information stored in multiple databases the applications overall size averages around 27.5 mb, which compared to a huge application likes Facebook's social network it is miniscule because Facebook's is currently at 365mb. A connection is also made to Heroku's Postgresql database to pull down an image of the college stored and all the co-ordinates needed for the navigation section. It was done this way to save memory on the phone. Instead of each database per college storing the image and wasting memory it retrieves the binary of said image and re-converts it to a bitmap from the main screen every time the application is used and places it on screen. To do this a separate project was made to act as the Heroku maintenance code of the database, also using java. This code is used to test the pushing and pulling of information to the server. Once everything is pushed up it never has to be done again, but the code is then integrated into the project to deal with retrieving the image.

## 5.2 Universality

A decision to make the application universal was made during the development. All students can use the timetable and assignment features and the navigation section will be constantly updated to allow other various colleges to use it. However at the moment the application is only accessible to Android users. Android was chosen because it's native language is java and this is well-known by the developers. However the initial intentions of the project was for it to be available on all platforms. Different technologies exist that allow the application to be compiled cross-platform to the IOS operating system for iPhone and Microsoft Windows operating system for Microsoft phones, but this was not accomplished because time ran out. The overall structure of the project will be concrete regardless of the operating system because of the platform neutral technologies chosen. The chosen database explained previously in the technology review of this report is Sqlite. Sqlite is supported by all mobile platforms as a database choice. The syntax of all the internal queries made in the application will remain the same. The same concept applies to the college selection, as previously explained Heroku's Postgresql database is entirely independent from the applications architecture. The only difference being the driver that enables the connection on each platform. Android uses java's JDBC driver to connect. This driver will be of no use to the other platforms because IOS is written in either Objective-C or Swift using the ODBC driver, and Windows is written using C-Sharp and uses the ADO.NET driver. This means each operating system specific deployment of the project will need to have it's own driver included and the syntax within the CollegeSelection class of the project will have to change slightly to replace the JDBC driver with the new one. Each language uses it's predefined driver in order to make a connection to Heroku's database and retrieve information based on an SQL query the application makes. This was done so the phone doesn't waste memory storing information on all colleges. See Figure 5.1. In conclusion the application is universal due to the number of colleges and universities that can use it. It can also be easily transformed to be cross-platform with some minor changes needed to one section of code that makes the initial connection.

## 5.3 Improvements

Each section of the application has areas where improvements could be made if more time was available. Below is a list of possible improvements that can be made in the future for the application in general and each individual section.

- Application: A priority queue added to handle threads because the current implementation creates a lag in the connection.
- Application: Add in an instant messenger section for students to talk to one another about their timetables and current assignments.
- Application: Add a section that users can personalize it to their preference.
- Camera navigation: More colleges and universities in the country can be added to the list.
- Camera navigation: Better graphics can be used for augmented reality.
- Camera navigation: More responsive software to deal with multiple locations within the geo-fencing area.
- Camera API: A swap from the deprecated Camera API to the new camera API made available for newer Android SDK's.
- Assignment: The ability for users to share assignments across platforms with each other.
- Assignment: Section for lecturers to upload assignments and the users can just enrol to that assignment.
- Timetable: The ability for users to share timetables with each other.
- Timetable: Section for colleges to upload timetables automatically and users can download it.
- Timetable: Colour code each section of the timetable.
- Google Maps: Option to view each individual floor of a building being seen.
- Google Maps: Provide directions from current location to a marker chosen by the user.



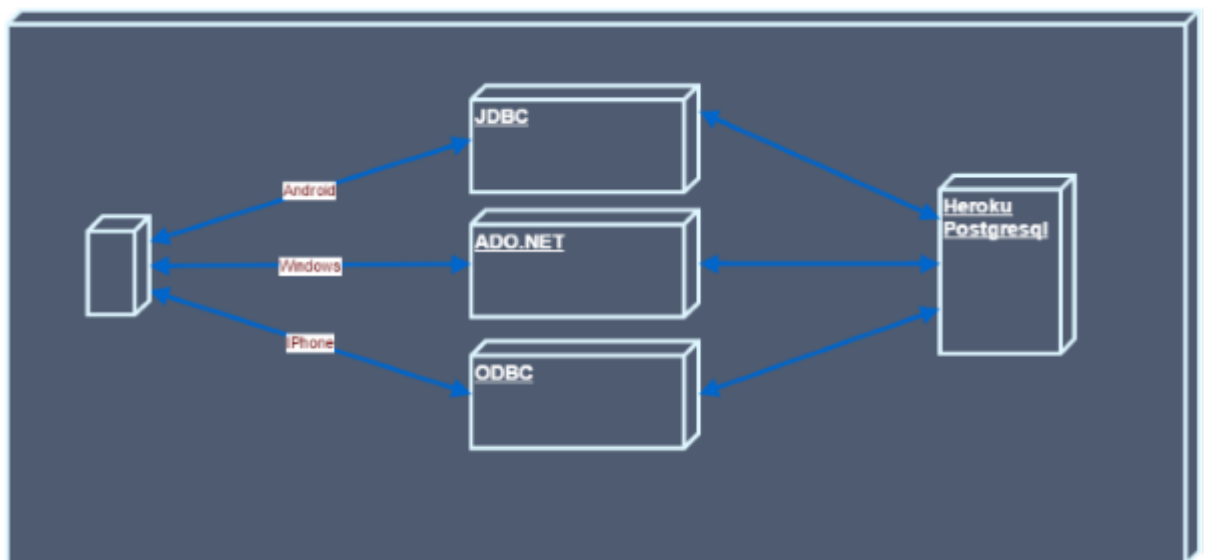


Figure 5.1: Heroku Connection Diagram

# Chapter 6

## Conclusion

About three pages.

- Briefly summarise your context and ob-jectives (a few lines).
- Highlight your findings from the evalua-tion section / chapter and any opportuni-ties identified.

# Bibliography

- [1] D. Baek, “We take pride in awesome mobile app development.” <http://www.nodesagency.com/>. Accessed: 2016-03-09.
- [2] M. Pawlan, “What is javafx?” <http://docs.oracle.com/javafx/2/overview/jfxpub-overview.htm>. Accessed: 2016-03-10.
- [3] opencv dev team, “Face recognition with opencv.” [http://docs.opencv.org/2.4/modules/contrib/doc/facerec/facerec\\_tutorial.html](http://docs.opencv.org/2.4/modules/contrib/doc/facerec/facerec_tutorial.html). Accessed: 2016-03-10.
- [4] admin, “Understanding agile methodology.” <http://agilemethodology.org/>. Accessed: 2016-03-20.
- [5] Gradle.org, “Emerge from build hell.” <http://gradle.org/>. Accessed: 2016-03-20.
- [6] JUnit, “JUnit 5.” <http://junit.org/>. Accessed: 2016-03-20.