



---

# CollegeEssentials - The app for students

---

**Aaron Flanagan**

**Ciaran Brennan**

B.Sc.(Hons) in Software Development

MARCH 20, 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>5</b>
<b>2</b>	<b>Methodology</b>	<b>8</b>
<b>3</b>	<b>Technology Review</b>	<b>11</b>
3.1	XML . . . . .	11
<b>4</b>	<b>System Design</b>	<b>12</b>
<b>5</b>	<b>System Evaluation</b>	<b>13</b>
<b>6</b>	<b>Conclusion</b>	<b>14</b>

# 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 Science and Applied Physics in Galway-Mayo Institute of Technology 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 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 the features were developed within the project time limit.
- **Technology review:** This section discusses the various technology platforms 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

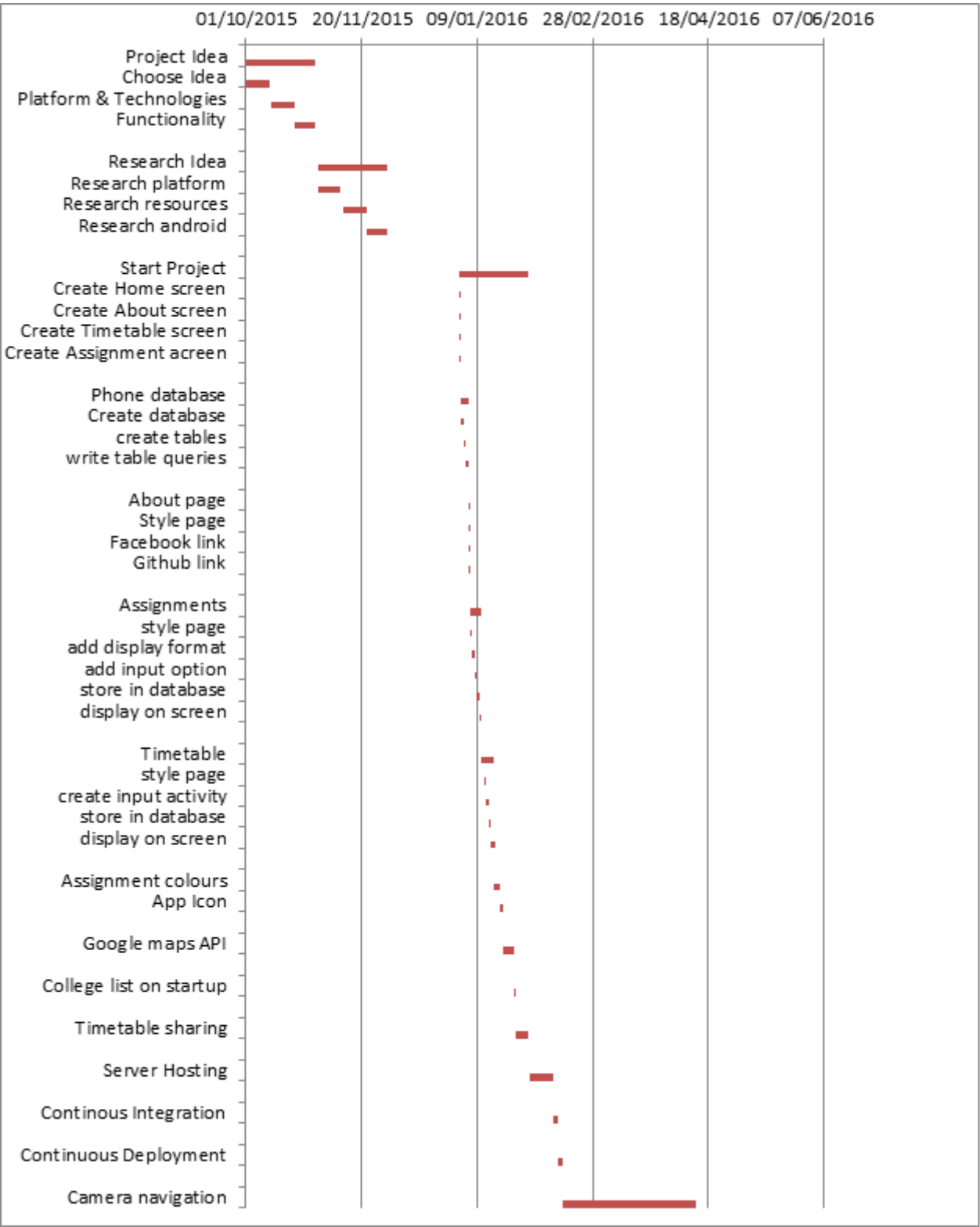
by attempting to install Intel's hardware accelerator software or by checking for in the task manager of your machine. If said machine doesn't support it, you will be unable to run the emulator, instead you must plug in an android device with the developer options enabled.

## Chapter 2

# Methodology

This section of the report will discuss the approach taken to develop the project and all the components used. It will include details about the planning performed, the meetings attended, the version control system used to maintain it and the testing technologies used. The project was developed based on Agile methodology. "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 first step taken was the planning section. A project work flow was developed to time out each feature and take into account other responsibilities for the other college modules. The Gantt chart was developed leaving out the section that assigns every individual developer to a particular area because the list of features was not particularly big and it was more beneficial to work as a pair. Work flow chart:





The traditional waterfall approach was not taken due to its limitations put on each phase of the project. Instead an incremental approach was taken. Both developers met up to discuss which feature to work on and to begin working on it. This meant that each feature could be developed no matter how long it took to finish. One feature may have took a week at most to close off while the next might have taken a month. It also allowed us to research and 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. An incremental approach was taken when adding in each feature, meaning one feature at a time was developed. 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 and a meeting with the supervisor was held to update him on the progress.

This project used three forms of testing. First used 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]. 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 run a selection of tests on the new changes. If it passed the test then the commit would integrate with the rest of the project, if failed it would not be integrated until the problem was fixed. 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.

# 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

As many pages as needed.

- Prove that your software is robust. How? Testing etc.
- Use performance benchmarks (space and time) if algorithmic.
- Measure the outcomes / outputs of your system / software against the objectives from the Introduction.
- Highlight any limitations or opportunities in your approach or technologies used.

# 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, “Agile methodology.” <http://agilemethodology.org/>. Accessed: 2016-03-20.
- [5]
- [6]