Contents

[Introduction 2](#_Toc512357425)

[System Requirements 2](#_Toc512357426)

[Technology Used and Why 2](#_Toc512357427)

[Architecture of the Solution 2](#_Toc512357428)

[Design Methodology 2](#_Toc512357429)

[Features of the Implementation 2](#_Toc512357430)

[Limitations 2](#_Toc512357431)

[Known Bugs 2](#_Toc512357432)

[Recommendations for Future Development 2](#_Toc512357433)

[Conclusions 2](#_Toc512357434)

# Introduction

This design document should give an overall introduction to how our application works and the various different software applications we used to complete our timetable app. This document should help understand the purpose of our application and give information such as the architecture, structure and various different implementations used in our app.

Upon registering and logging into the app, the main page displays an option to search for any course in GMIT. We designed this application to provide easier access for students and staff to view their timetables while providing them push notifications to alert users before their next class and to alert them of what class they have and where it is being held. The login, registration and timetable view pages are designed with the idea of being easy to follow. Once the user is logged in, their credentials are remembered and won’t have to login twice.

# System Requirements

Any Android phone, apple iphone, windows phone or any computer that can run the latest browsers.

# Technology Used and Why

The main purpose of our app is to give the students and staff member of GMIT a central application for accessing and informing students of various classes going on for each specific user of GMIT’s timetable.gmit.ie website. As a team, we realized accessing the timetable website is time consuming and navigating through various different links to get to the users timetable is difficult, so we wanted an application that would allow you to create an account, select your course, and allow the user to open the app at any time without having to enter in any more info.

# Architecture of the Solution

Ionic

We choose the frontend framework Ionic for the simple reason of it’s a new technology & developing for multiple platforms is made very simply with this framework.

Angular

Angular 5 is widely used in front end development, ionic’s framework depends on this, allowing us to compile typescript files with html, CSS and various JSON files.

MongoDB

The MongoDB is a NoSQL database program that allows JSON-like documents to be easily stored with schemas.

Node.JS

Node.js is an open source, cross-platform runtime environment for developing server-side and networking applications. Node.js applications are written in JavaScript, and can be run within the Node.js runtime on OS X, Microsoft Windows, and Linux. The actual server uses Node.js to allow various different GET and POST requests.

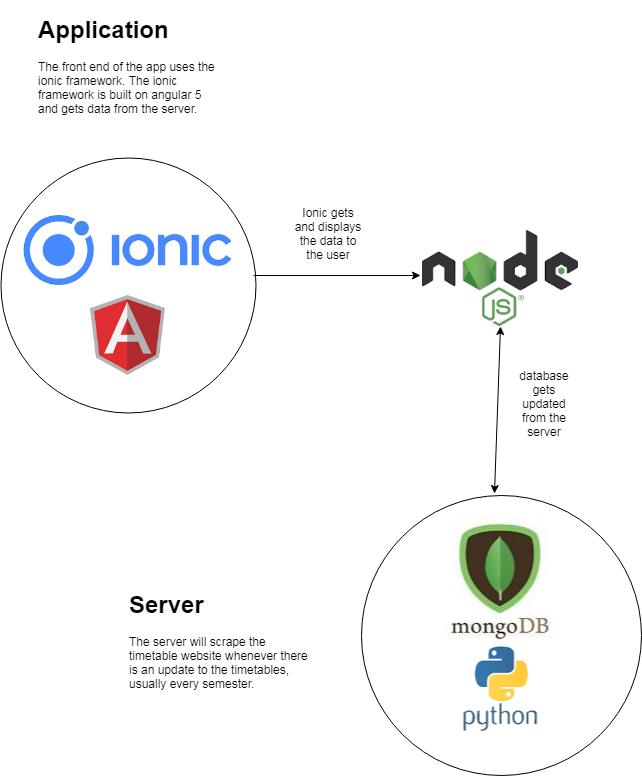
Python

We used python because of the various different libraries for http and for scraping the gmit timetable website; we also used various plugins for python to allow for web scraping.

BeautifulSoup

Mechanize

# Design Methodology



# Features of the Implementation

# Limitations

# Known Bugs

# Recommendations for Future Development

# Conclusions