**Graph Theory Project 2017:**

**GMIT Timetable.**

**Name:** Conor Tighe.

**Id:** G00314417.

Introduction

For a project in the module Graph Theory I was told to take the timetable found on the GMIT website and redesign it into my own schema that will allow users to interact with the elements within the timetabling system. The goal of this project is to hopefully improve the timetabling system found on GMIT using the design patterns we studied in the module and my own critical thinking skills. I have decided to cover the Software development and Digital Media semester 2 timetables as this will allow me to create a large schema rich in data to interact with without making the scope of the project unrealistic. In this document, I will explain the process taken in creating the schema, the challenges encountered and the design and architecture of the model. I will also discuss the technologies/methods used to complete the task and how I used them.

Technologies

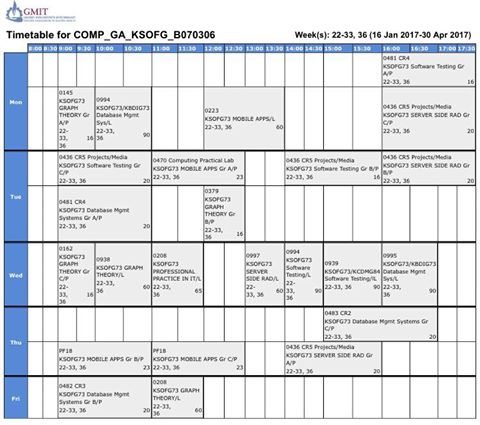
**Neo4j:** This is a NoSQL Graph database management system which can be used to display data as nodes and their edges as relationships. It is currently the most popular graph database. Neo4j displays its data-sets in eye-pleasing bubble layout that lets large amounts of information easy to read and interact with.

**Python:** This is a programming language first released in 1991 but still very popular. I decided to use this language to complete any programming parts of the project as it is easy to use yet powerful making it perfect for developing a web crawler.

**Microsoft Excel:** This is a program for developing spreadsheets and manipulating the data found in the rows and columns of the spreadsheet.

Planning and architecture

Before creating anything in Neo4j I wanted to have a design in mind of how my database would represent the timetable and the relationships between elements. To start designing I had to study the GMIT timetable and the page that displayed it first. After looking at the information displayed I decided the time, module and room where what I would base my database around as they were what a student would most likely to search for. The data like class capacity could but made into sub-information under room as it would be unusual for a student to need that data displayed first. After reviewing the timetable below I decided the best way to go about extracting the information is dividing it up by the days along the left and the time along the top. The reason for this would enable by database to have a more in detailed search by not just looking for a matching time but also the day once I feed the information into my database.



Extracting data

After now figuring out what I needed to make up my timetable I started researching ways to take information off web sites. I wasn’t as easy as simply copying and pasting the data as all the timetables where made up of html table cells found across multiple pages. This meant that manually retrieving the data would be too time consuming. After using google chromes inspect source feature to review the code that made up the timetable and examined where the data I wanted was located. I noticed all the data was hardcoded the exact same way on each page making a perfect pattern for a web crawler.

Python seemed like a good choice for this job as it offers a clean and simple syntax with a wide variety of library’s. This would mean I could spend minimal time on writing an effective web scraping program as that is not the goal of the project. The clean syntax would also make dealing with large amounts of data easier. After looking online I read an article that discussed web scraping with a library called beautiful soup and then began reading the documentation on the library. I wrote the following program which has comments through-out explaining how it retrieves the data from the GMIT timetables html.

