📖 AnonGuy / **Structure**

| Branch: devel ▾ | **Structure** / write_up / **README.md** | | Find file | Copy path |

🌀 **AnonGuy** Update README.md   046f156 21 seconds ago

1 contributor

---

73 lines (49 sloc)   4.19 KB

Analysis

Problem Definition and Stakeholders

The stakeholders of my application will be **fellow students of Loreto Sixth Form College**. Being students, they need to organize revision sessions, in order to maintain the grades required for their chosen university. Many Loreto college students have issues with organising a revision timetable. For example, creating study sessions that do not clash with classes, or setting up reminders for each of these study sessions, can be difficult. The needs of a College student also include regularly checking up on their performance (grades, attendance) to ensure that they are making the best of their study periods.

The current system for acheiving these tasks is the following:

- Checking the current timetable
- Checking grades and subject performance
- Use this information to create a manual revision timetable
- Regularly update this timetable manually based on performance

The issue with this system is that it is entirely manual, which means that students may neglect to create or update a revision timetable. Also, the current system does not allow srudents to add their own custom lessons, or be notified for a lesson. My idea is to have a user-friendly system that integrates all these features, in one easy to use web application.

My application will solve this problem, by **integrating revision session management**, **homework reminders** and **performance analysis** in one simple dashboard. Students will be able to access this dashboard via a **Django web application**, and possibly a **React Native mobile application**.

In order to solve this problem with the suggested solution, I will need to learn the following:

- Usage of the Django web framework
- Usage of the PostgreSQL Database Management System with Python
- Usage of JavaScript graphing libraries such as D3.js

Justification for Computational Method

This problem lends itself to utilization of computational methods for several reasons:

- Student performance data will be abstracted, instead displayed as as a simple line graph.
  - In order to display data as a graph, a computational device is needed.
- The application will also automatically generate a study timetable that adapts to this data.
  - This solution utilizes algorithms that can only be executed on a computational device.
- The application stack is a system that includes my webserver, which will run on a computer.

Therefore, there is no alternative solution that would not require a computer to execute.

Applications of Computational Methods

There are several ways that computational methods can be applied to this project.

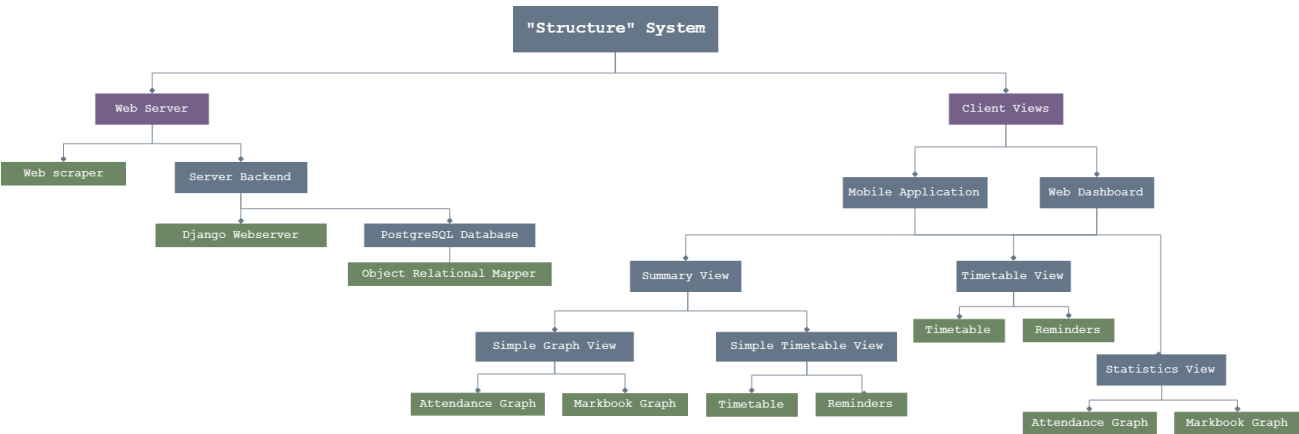| Method of computational thinking	Explanation of relevant features
Abstract	My web application will be as user-friendly as possible. this will require me to hide certain details from the user - for example, the raw data that is scraped by my webscraper. Instead, the user will only see data that is relevant to their current page: If they are on the summary page, they will only see simple summarized data and graphs for their performance.
Heuristic	The summary page of my dashboard will include rounded, approximate values to represent the correlation between various properties of the student's study time, free time, grades, and punctuality.
Procedural	My application stack will be comprised of several Python modules, each separated into classes, methods and functions. I will be able to remove modules, test modules and edit modules in a way that does not affect any irrelevant components of the application.
Concurrent	The web application must be able to handle multiple HTTP requests simultaneously. I will be applying concurrent thinking to plan and design the processing and response for each request.

Research

I found a service that is a similar solution to my own, **showmyhomework.co.uk**, by Satchel:

Show my Homework

Design

Decomposing the Problem

Database Design

User

PK	UserID	Integer
FK	StudentID	Integer
	Username	Character Varying(8)
	Password	Character Varying(20)

Form

PK	FormID	Integer
FK	ClassID	Integer
	Form_Group	Character Varying(10)
	Hall_Head	Character Varying(50)

Class

PK	ClassID	Integer
	Teacher	Character Varying(50)
	Subject	Character Varying(50)
	Room	Character Varying(10)

Student(django.contrib.auth.User)

PK	StudentID	Integer
FK	FormID	Character Varying(50)
	Markbook	JSON Binary
	Avatar	Character Varying(20000)
	Reference_Number	Character Varying(12)
	Name	Character Varying(20)
	Timetable	JSON Binary
	Email	Character Varying(29)