AdviseMe

Help Advise Systems: Philip Osborne, Trevor White, Howard Aleshire, & Timothy Bell

**Project Definition**

**Why:**

Many students have difficulty with classes due to lack of guidance from current advising systems.

**What:**

An automated advising system that recommends classes and provides students with a schedule based on their desired parameters. The system will ensure that it only recommends classes that will be beneficial to the students major, allowing them to graduate in a timely manner.

**How**:

It is achieved using Python, and it is approached by focusing on one major at a time.

**Project Requirements**

**Functional**

Users will be able to view the courses they should take for their next semester.

The system will determine the recommended courses based off various parameters, e.g. major requirements, prerequisites, number of credit hours remaining to graduate, etc.

Users will be able to view which classes will transfer to and/or from their current institution to another, and vice versa.

Users will be able to select courses based off of various constraints, such as course category, subject, time, etc.

If a course is full or a student needs permission to register for a course, the system tells it in the course and offer an option to email the instructor of the course through the email system of their respective university.

If necessary, a user will be able to message a human advisor.

**Usability**

**Performance**

The system will prioritize efficient sorting and searching algorithms.

Efficient database queries are a must, since most of the systems’ calculations will be done on remote servers via relational databases.

**User interface**

The main GUI will contain buttons that allow the user to navigate to the other subsystems’ respective GUIs.

The subsystems’ GUIs will contain links back to the GUI of the root system, in addition to their own unique GUI features.

**System**

**Hardware**

Cross-platform desktop application that works on the major operating systems.

**Software Downloaded application**

Users will download the application from the internet. The users will need internet connectivity in order to communicate with the application backend.

**Database**

At this stage, the exact DBMS is uncertain, but currently being considered are Pandas, MySQL, and PostgreSQL.

**Security**

The majority of security concerns will revolve around user login, i.e. verifying the user is who they say they are.

**Project Specification**

**Focus / Domain / Area**

Academic advising does not have much in terms of software. Colleague and Workday are two advising aids that allow both student and advisor to track things like academic standing, required courses, and grades. Such applications are helpful, but do not share the same focus as AdviseMe. Our focus is a system that will generate a suggested schedule, rather than allow students to track taken and required courses.

**Libraries / Frameworks / Development Environment**

At this stage, not much is set in stone as far as libraries, frameworks, and environments are concerned. We will probably utilize the pandas library in python, and work in the IDLE development environment, also for python.

Platform (Mobile, Desktop, Gaming, Etc)

AdviseMe will be a web-based application, much like Colleague.

**Genre** (Game, Application, etc)

AdviseMe is an application designed to be used by students.

**Feasibility**

It is on approach to let this project be feasible because feasibility depends on approach when making an advising system. Time is better spent helping feasibility than knowing it.

**Approach**

To help the Feasibility of this project, the focus is on getting this project to work on one major at a time. This one major is Computer Science. The quick prototyping nature of Python allows changes to this project to be quick, and implementing a subsystem to this project changes this product.

**Draft model**

