AVA An Automated Voice Activated Advisement System

(Requirements/Specifications Draft)

By: The Software Gurus

Team Members

Omar Obidat

Jesse Parron

Allen Asencio

Dan Ferdetta

Sumit Mistry

Software Engineering 515

Fall 2022

Professor Johnson

October 10, 2022

Product Scope:

College campuses are known for having a surplus of majors/minors and hundreds to thousands of potential options for classes a student may need to take. With that being said, it may be hard to figure out which courses one should take when pursuing a degree. Students may accidentally take classes they don't need, or take too many classes and overload themselves. In order to advise students in an efficient manner that doesn't take too many labor hours, a voice activated advisement system has been developed in order to provide assistance in all areas of student advising. The system will allow a user to verbally communicate with an automated adviser, who will guide them through the process of picking the correct course(s).

[handwritten: This means it already exists]

[handwritten: This is rather ambitious!! There is a reason it was required that the system be developed for the CS dept.]

Product Features:

- Recognize user voices.
- System should allow administrative views for editing and updating software.
- Be up to date and flexible with changing university graduation requirements and courses no longer being taught. *[handwritten: Will this be someones responsibility or will this be done automatically by the system?]*
- Allow students to log in and interact with the advising system. Such functionalities are: *[handwritten: The functionalities which students will be allowed to perform, are:]*
    - View their course catalog.
    - View their transcript.
    - View available courses that can be taken.
    - Talk with an automated system that will provide feedback on any questions they have. *[handwritten: ?? vague!! State the specific kinds of questions!!]*
    - System should be user friendly and easy to use. *[handwritten: ??]*
    - Student can filter the advisor system to specific: *[handwritten: will be able to / determine the ??.]*
        - Credit amounts in a given semester. *[handwritten: number of which can be taken]*
        - Full time/part time course loads, if an undergraduate student is partaking in 12 or more credits they are considered full-time, anything less is part-time. Anything less is part-time.
        - Time period of classes in the day.
        - Preferable concentrations of what courses they want.
- Ask the system to calculate the current GPA of students, and the GPA they would have after the completion of following semesters courses. If GPA is low, the advising system will notify students about possible academic probation.
- System should be able to generate multiple courses the user can pick from and the amount of courses they would need in order for the student to graduate on time. *[handwritten: The / number]*
- System can give different instructor choices for the classes that are offered during the semester of the students choice. It will display the names of each professor that will be teaching the course, so if a student has taken/likes that professor then they can choose that course.

[handwritten: Whether the intent was to refer to "that professor" or "a student", this cannot be "they"]

Users Characteristics:
- Students
    - The typical undergraduate student is between 18-23 years of age, although there are cases where some will be outside of said range.
    - Education level is undergraduate.
    - All undergraduates using the system are majored and/or minored in the Computer Science and Information Technology (CSIT) Department.
- School Administration
    - Presidents, Deans, Chairmen, Professors.
    - Has the privileges to add/update any courses or students that need to be modified in the system.
    - Can view any information that the system advises to a student.
    - Bypass course size limits for students who require an already full class for graduation.

Constraints:
- Development tools; system will be built using Python, HTML, CSS, Bootstrap, JavaScript, Sqlite, and Flask.
- The developers are limited in terms of the level of security of the system; The developers can implement hashing for the passwords and disconnect user sessions when trying to go back after logging out.
- The developers will follow school regulatory procedures for the system.

Assumptions and Dependencies:
- The developers have access to the university's database.
- The development team all have access to the system's code.
- The system has a reliable server or system to be installed on.
- Changes in the original requirements may occur.