AVA An Automated Voice Activated Advisement System

(Design Document)

By: The Software Gurus

Team Members:

Omar Obidat

Jesse Parron

Allen Asencio

Dan Ferdetta

Sumit Mistry

Software Engineering 515

Fall 2022

Professor Johnson

October 31, 2022

Introduction is required

Use the "Dot Leader" feature available in a Word Processor to format the table of contents properly!!

Requirement and Specifications

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).

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.
- Allow students to log in and interact with the advising system. Such functionalities 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.
- System should be user friendly and easy to use.
- Student can filter the advisor system to specific:
  - Credit amounts in a given semester.
  - 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.
  - 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.
- 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.

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.

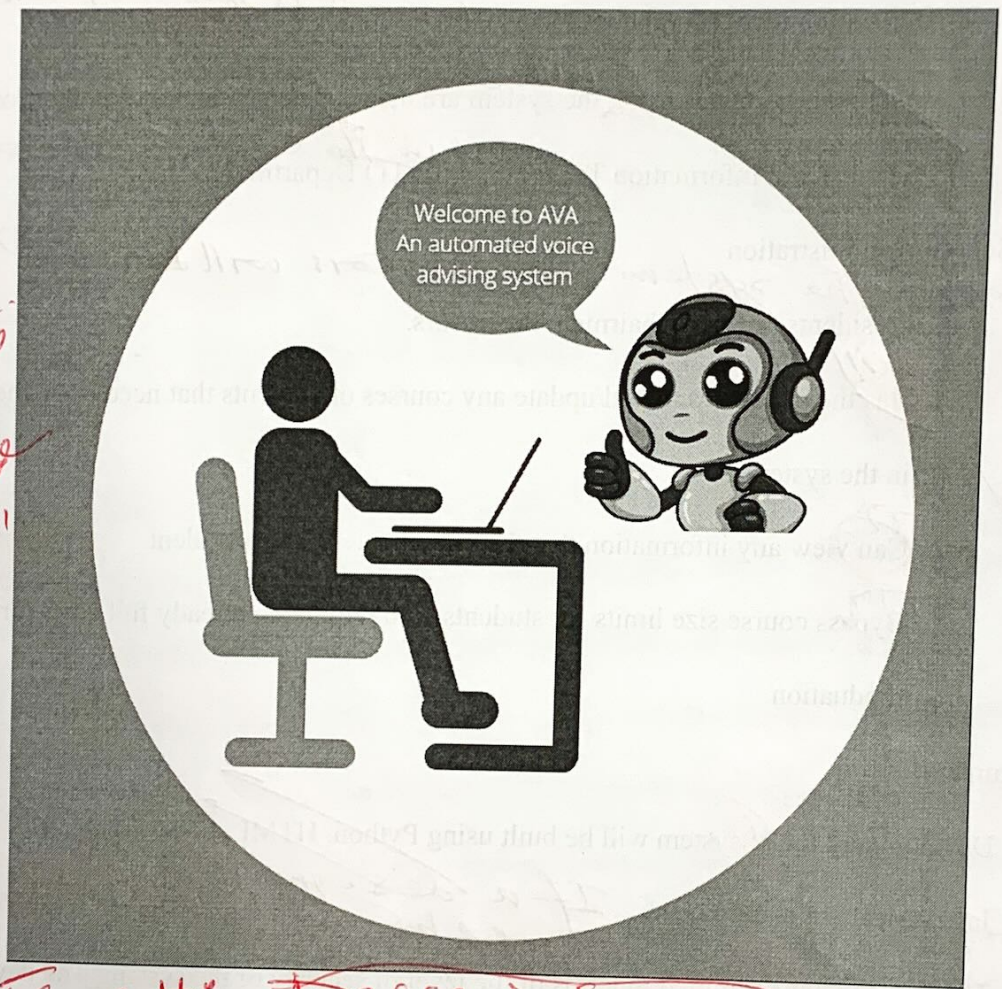
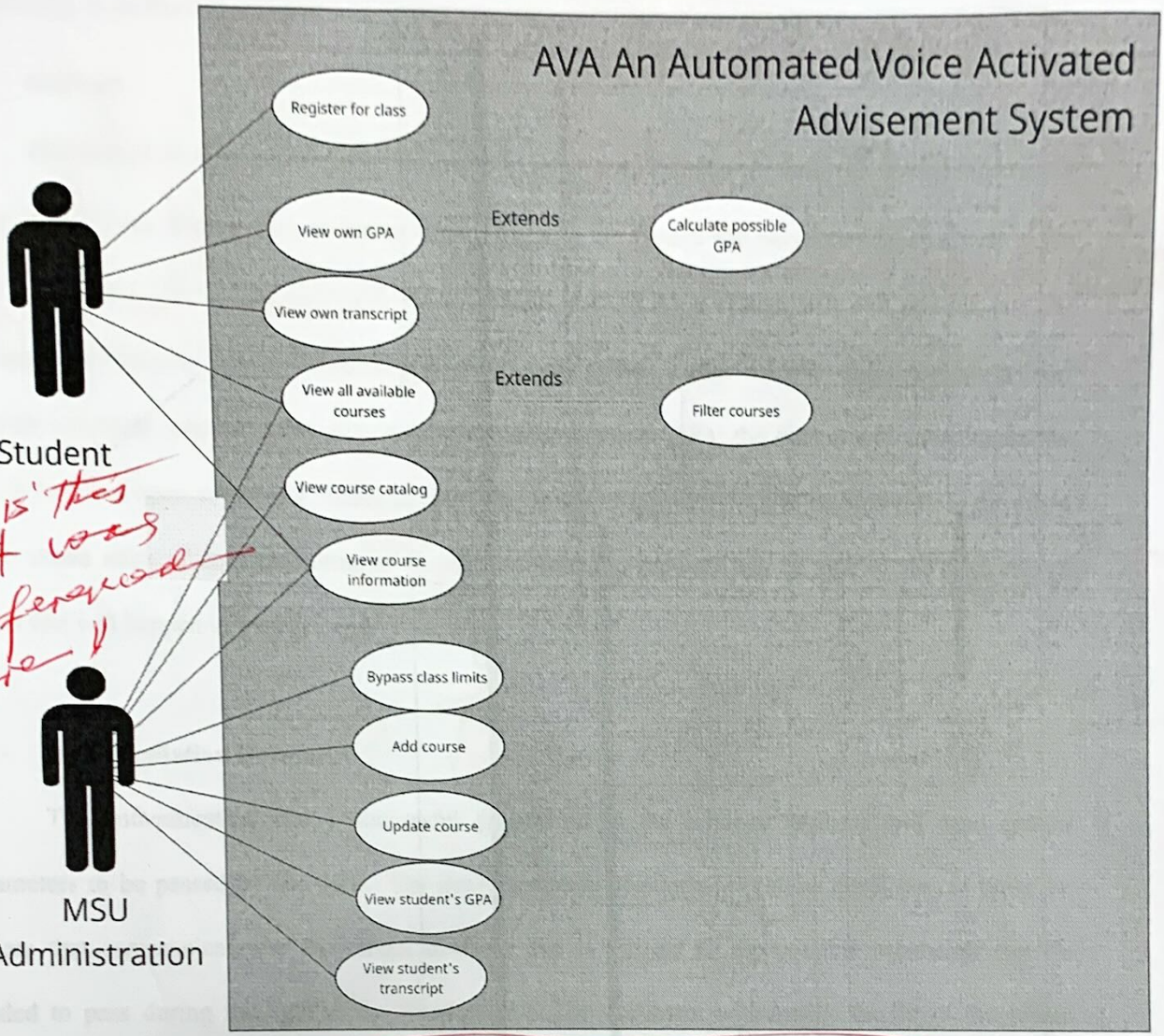


Figure 1

Use Case Diagram for ??



What is this for? It was not referenced anywhere !!

Description Required.

Figure 2

Activity Diagram

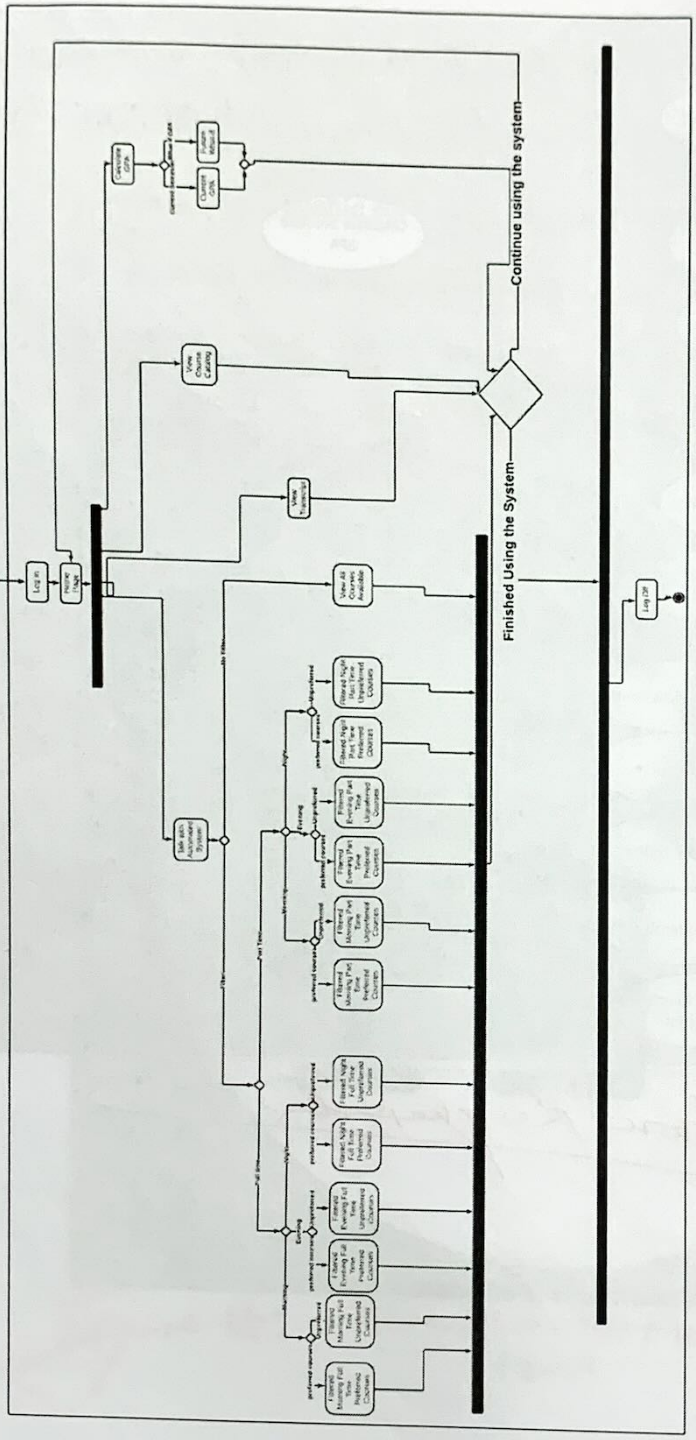


Figure 3

See comments on pages 6 and 7 pertaining to figures.
Neither figures nor Tables can be just placed in a document without reference to them. The purpose of a figure or a Table, in a document is to support/illustrate a point being made !!

Top Down Design

Connecting to School Database:

- **Abstract**

The School Database stores the information about the student and Montclair State University's various departments. For students to access the information, the system will need to establish access to the school's database. This component will be an integral part of AVA system that will provide just the database authentication and a connection to the system. When a user asks for the information like; currently available course in semester or simply the last semester GPA, the system will authenticate the student first and upon successful authentication, the requested information will be provided. In the event of the above steps of communicating with the School database, the user will be able to see the series of events that will happen in the back end.

- **Implementation Document**

The authentication and a successful connection to the school's database will need certain parameters to be passed by the users. The implementation document will allow developers to build the system that communicates to the school database and it will list all the possible parameters that are needed to pass during the application development. The document will contain the list of the school database server details, the database type, database name, and the authentication types to use.

- **Design**

The design section will cover the user-initiated database connection strategy. As illustrated in the figure-4, this component will need the authentication decision to begin for the user to interact with the system. Upon user's passing in the default parameter, the authentication step will take care of the database access. Three consecutive unsuccessful tries will automatically disconnect the user for the further tries and the user will be asked to retry after some time. Upon a successful authentication, the user will be

allowed to request the desired information of interest only as per the defined AVA system protocols. The user will be allowed to access a certain part of the database based on the role that was authenticated. Once the AVA system designs the query, it will retrieve the user requested information so the user can be informed of the resulting outcome retrieved from the database. Once the user ends the call session, the database connection will be dropped automatically.

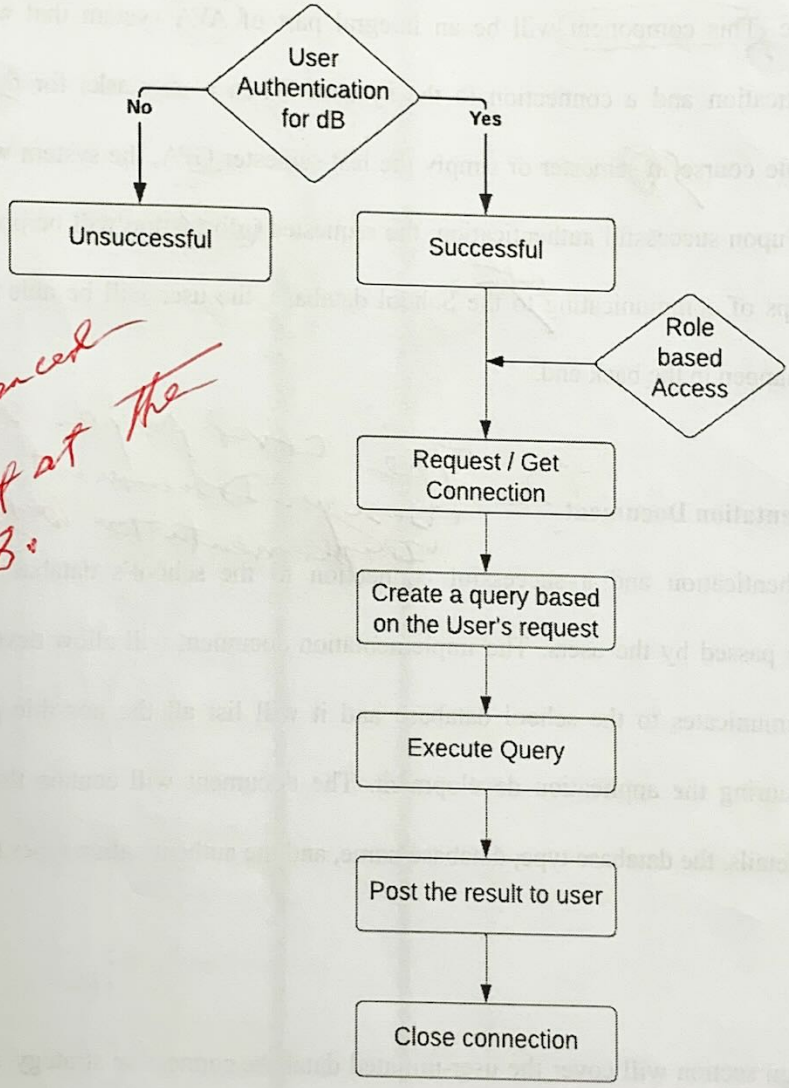


Figure 4: Database connection strategy

- **Exports**

Presence of Database and its connection make the entire system Live and so it can retrieve data from the database. This component exports information into Course Catalog Viewer that displays available courses from the catalog, Unofficial Transcript Generator which generates an unofficial transcript and GPA calculator that calculates the GPA. This hierarchy system is interdependent.

- **Imports**

When a user calls, this component imports the data from the Language recognition component to generate a successful database connection. The database imports data from the user's choice of course selection and class schedule for the semester.

- **Input/Output**

The Input for the component will be the user request transformed query. This query will help users to get the information from the school database.

The Output will be an action and or the information that the user requested. The result obtained by running the query will be the output of the component.

- **Pre and Post Conditions**

Preconditions for the successful database connection will be listed in the implementation document. The precondition for this component will be that the user is required to provide certain parameters to fully authenticate into the database and perform certain activities.

- **Error Handling**

After a user's successful authentication, if the data is found missing or has a glitch, the system will report the information using the logger, and such activities will be monitored to maintain the application integrity.