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

December 19, 2022

Table of Contents

[1.](#_heading=h.1fob9te) Introduction 4

[2.](#_heading=h.3znysh7) Requirement and Specifications 4

[2.1.](#_heading=h.2et92p0) Product Scope 4

[2.2.](#_heading=h.tyjcwt) Product Features: 5

[2.3.](#_heading=h.3dy6vkm) Users Characteristics: 6

[2.4.](#_heading=h.1t3h5sf) Constraints: 6

[2.5.](#_heading=h.4d34og8) Assumptions and Dependencies: 6

[3.](#_heading=h.2s8eyo1) Use Case Diagram 8

[4.](#_heading=h.17dp8vu) Class Diagram 9

5[.](#_heading=h.17dp8vu) Sequence Diagram 10

6[.](#_heading=h.3rdcrjn) Top-Down Design 10

6[.1.](#_heading=h.26in1rg) Connecting to School Database: 10

6[.1.1.](#_heading=h.1y810tw) Abstract 10

6[.1.2.](#_heading=h.4i7ojhp) Plan for Implementation 11

6[.1.3.](#_heading=h.2xcytpi) Design 11

6[.1.4.](#_heading=h.1ci93xb) Exports 12

6[.1.5.](#_heading=h.3whwml4) Imports 13

6[.1.6.](#_heading=h.3as4poj) Input/Output 13

6[.1.7.](#_heading=h.1pxezwc) Pre and Post Conditions 13

6[.1.8.](#_heading=h.49x2ik5) Error Handling 13

6[.1.9.](#_heading=h.2p2csry) Side Effects 13

6[.2.](#_heading=h.147n2zr) Language Recognition 14

6[.2.1.](#_heading=h.3o7alnk) Abstract 14

6[.2.2.](#_heading=h.23ckvvd) Plan for Implementation 14

6[.2.3.](#_heading=h.ihv636) Design 14

6[.2.4.](#_heading=h.32hioqz) Exports 15

6[.2.5.](#_heading=h.1hmsyys) Imports 15

6[.2.6.](#_heading=h.41mghml) Input/Output 15

6[.2.7.](#_heading=h.2grqrue) Subparts 15

6[.2.8.](#_heading=h.vx1227) Pre and Post Conditions 15

6[.2.9.](#_heading=h.3fwokq0) Error Handling 16

6[.3.](#_heading=h.1v1yuxt) GPA Calculator 16

6[.3.1.](#_heading=h.4f1mdlm) Abstract 16

6[.3.2.](#_heading=h.2u6wntf) Design 16

6[.3.3.](#_heading=h.19c6y18) Exports 16

6[.3.4.](#_heading=h.3tbugp1) Imports 16

6[.3.5.](#_heading=h.28h4qwu) Input/Output 17

6[.3.6.](#_heading=h.nmf14n) Subparts 17

6[.3.7.](#_heading=h.37m2jsg) Pre and Post Conditions 17

6[.3.8.](#_heading=h.1mrcu09) Error Handling 17

6[.3.9.](#_heading=h.46r0co2) Side Effects 17

6[.3.10.](#_heading=h.2lwamvv) Miscellaneous 17

6[.4.](#_heading=h.111kx3o) Course Catalog Viewer 18

6[.4.1.](#_heading=h.3l18frh) Abstract 18

6[.4.2.](#_heading=h.206ipza) Design 18

6[.4.3.](#_heading=h.4k668n3) Exports 18

6[.4.4.](#_heading=h.2zbgiuw) Imports 18

6[.4.5.](#_heading=h.1egqt2p) Input/Output 18

6[.4.6.](#_heading=h.2dlolyb) Pre and Post Conditions 17

6[.5.](#_heading=h.sqyw64) Unofficial Transcript Generator: 17

6[.5.1.](#_heading=h.3cqmetx) Abstract 17

6[.5.2.](#_heading=h.1rvwp1q) Design 18

6[.5.3.](#_heading=h.4bvk7pj) Exports [1](#_heading=h.1rvwp1q)8

6[.5.4.](#_heading=h.2r0uhxc) Imports [1](#_heading=h.1rvwp1q)8

6[.5.5.](#_heading=h.1664s55) Input/Output 18

6[.5.6.](#_heading=h.3q5sasy) Subparts 18

6[.5.7.](#_heading=h.25b2l0r) Pre and Post Conditions 18

6[.5.8.](#_heading=h.kgcv8k) Error Handling 18

6[.5.9.](#_heading=h.34g0dwd) Side Effects 18

6[.5.10.](#_heading=h.1jlao46) Miscellaneous 19

7[.](#_heading=h.43ky6rz) Pseudocode 19

7[.1.](#_heading=h.2iq8gzs) Login 19

7[.2.](#_heading=h.xvir7l) View course Catalog 19

7[.3.](#_heading=h.3hv69ve) Unofficial Transcript generator function 19

7[.4.](#_heading=h.1x0gk37) Advisement on classes and questions 19

7[.5.](#_heading=h.4h042r0) Language processing 20

7[.6.](#_heading=h.2w5ecyt) GPA calculator 21

7[.7.](#_heading=h.1baon6m) What-If GPA 21

7[.8.](#_heading=h.3vac5uf) Letter Grade to Number (LGN) 22

7[.9.](#_heading=h.2afmg28) End Session 22

8[.](#_heading=h.pkwqa1) Detailed Design 23

# Introduction

Automated Voice Activated Advisement System (AVA) is a computer science undergraduate student advisement tool built for Montclair State University as per institutional standards to support the user and institution, providing more effective communication, quality, and accuracy of the academic success of the students majoring/minoring in Computer Science and Information Technology. Newly entering students may not be familiar with the institutional and educational system and the culture and so they are more likely to connect with academic advisors for questions and confusions. This project is built to support this case where students can get advice directly from the Voice Activated Advisement System instead of talking to academic advisors.

This will reduce the work labor needed from academic advisors on campus as well as allowing students to get the support quickly rather than waiting for an appointment to talk to the advisor in person. The system can advise and provide end-to-end solutions as well. It can assist users in various ways using fully voice activated methods, especially when users don’t need to press or select any navigation menu option like the traditional methods.

The system is exclusively accessible for Montclair State University students majoring/minoring in Computer Science and Information Technology and it has few integrated functionalities that a user can vocally request for assistance. After successfully passing the authentication, the user can view all the available courses for the term, request advisement for classes they need to take, view course catalog for the program, view specific course information, view user’s own GPA and Transcript and access the GPA calculator. All these primary functions will deliver users the basic and essential information for their needs. Once done, they can logout of the system.

# Requirement and Specifications

## Product Scope

College campuses are known for having a surplus of majors/minors and hundreds of potential options of courses a student may need to take. With that being said, it may be difficult 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 will be developed in order to provide assistance in picking courses that count toward the student’s major/minor that they will be able to take. The system will allow a user to verbally communicate with an automated adviser, which will guide them through the process of picking the correct course(s) to ensure the student knows which courses they must enroll in.

## Product Features:

* Recognize user voices.
* 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 will be:
  + View the course catalog.
  + View their transcript.
  + View available courses that can be taken.
  + Talk with an automated system that will provide all information on the course specified.
  + The system will be user-friendly and easy to use.
  + A student can request for academic advisement on course(s) they should take, the system will check previous taken courses, then display a list of courses they are eligible to enroll in.
  + Ask the system to calculate the current GPA of students, and the GPA they would have after the completion of following semesters courses.
  + The system will be able to generate multiple courses the user can pick from.
  + In the course catalog the system will give different instructor choices for the courses that are offered during the semester of the students choice. It will display the names of each professor who will be teaching the course, so if a student has taken/likes that professor then the student can choose that course.

## Users Characteristics:

* Students
  + Education level will be undergraduate, as this is for undergraduate students.
  + Undergraduates who will use the system are majors and/or minors in the Computer Science and Information Technology in the Computer Science Department.

## Constraints:

* The developers are limited in terms of the level of security of the system; The developers can implement session times, so when a user exits the system or tries to go back from certain pages, the system will stop the user and require them to log in again.
* The developers will follow school regulatory procedures for the system.

## Assumptions and Dependencies:

* The developers will have access to the university's database.
* The development team will all have access to the system’s code.
* The system will have a reliable server or system to be installed on.
* Changes in the original requirements may occur.

# Use Case Diagram

Use Case Diagram for the Automated Voice Activated Advisement System

**Diagram

Description automatically generated**

**Figure 1**: This is the Use-Case diagram for the Automated Voice Activated Advisement System. It shows the functions each user will be able to perform.

# Class Diagram

# **Figure 2**: This is the class diagram which illustrates how each function connects to each other. It also shows that each function will generate after it has run.

# Sequence Diagram

# **Figure 3**: This is the sequence diagram which illustrates how users can traverse through the program, and what is expected to happen.

# Top-Down Design

## Connecting to School Database:

### Abstract

The School Database stores a student’s account specifics like student number, name, a password, course history, and grades; as well as information regarding Montclair State University’s courses, such as the number of credit hours, course number, and instructor. Students are expected to know their student number and password in order to sign in. For students to gain access to the course information, the system will need to establish access to the school’s database. The database connection component will be an integral part of the Automated Voice Activated Advisement System (AVA) that will provide just the database authentication and a connection to the system. When a user requests information from AVA such as; currently available courses in the semester or simply the semester Grade Point Average (GPA), the system will authenticate the student first and upon successful authentication, the requested information, in this case GPA, will be provided. In the event 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.

### Plan for Implementation

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 Figure-4, this database connection component will need the authentication for the user to interact with the system. Upon user’s passing with their student ID and password, the authentication step will give the user database access.

Upon successful authentication, a call session with AVA will be started, and the user will be allowed to request the desired information only as per the defined AVA system protocols. The user will be allowed to access designated permissions and pieces of information based on the role that was authenticated, which can be shown in Figure-2. The AVA system will submit the query based on the user’s specifications, then the system will retrieve the user requested information from the database and inform the user of the result.

Once the user has all their queries from AVA answered, they will end the call session, and the database connection will be dropped automatically.

Diagram

Description automatically generated

**Figure 4**: Database connection strategy. This diagram shows what happens in a successful and unsuccessful user authentication for the database.

### Exports

Presence of the database and its connection makes the entire system live, 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 will import 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 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.

### Side Effects

The Implementation document containing the school’s server address and database information will carry a risk for the school against unforeseen hacking and cyber-attacks.

## Language Recognition

### Abstract

The automated language recognition component of the system is the main component and the most important. This function will allow the user to interact with all the other functions within the system. Once the user is logged on, they will then be able to vocally ask the system to perform certain functions. These functions will include: course advising, view a catalog of courses available, view an unofficial transcript, calculate the user's GPA, or simply ask questions on any other areas of concern that the system can handle. The system will handle these requests and respond accordingly. When a user asks for course advising, the system will check for completed past enrollments, and will then output a list of courses the user is able to take based on the prior courses. If the user wants to utilize any of the other functions, they can just specify the name of it, and the system will instantiate the module, switch pages, and display the information included in the module.

### Plan for Implementation

Users of the system will have to speak in a clear voice and ensure that questions are not vague. Users will also have to read the user guide prior to using the system to ensure complete understanding, and utilize functions listed as named, otherwise the system may not recognize what is being asked.

### Design

The language recognition advising component of the system will basically be the overhead. This component will interact and connect with all the other functions across the platform. The user will be able to log in and directly interact with the advising system. The user will be able to ask questions and utilize built in modules by stating the functions title. The first is the unofficial transcript, the user will be able to ask the system to use the “unofficial transcript” function. The overhead language recognition will pass the request on and the system will instantiate the function to produce an unofficial transcript for the user to view. If the user says to go home, the system will act accordingly. The second function connected to the language recognition overhead is the “course catalog”. The student will be able to state that they want to “view the full course catalog”, this will be recognized, sent, and received by the course catalog function and will produce a full list of all available courses for the upcoming semester. The user can then either state the course name for more information on a specific course, or ask to go home. The third function is the GPA calculator, a user will be able to state that they want to “calculate their GPA” this will produce the user's current GPA. The fourth function is the “what-if” GPA which will ask the user how many “What-if” grades they want, and then the user can list each grade one after another. It will then produce the what-if gpa for the student. Lastly, the user will be able to be advised on what classes to take based on previous completed courses.

### Exports

The language recognition component of the system will make language recognition available to all other components of the system.

### Imports

The course catalog is used when advising students on what to take next semester. Since it is a list of all classes that are available, it will choose the classes from there. The students unofficial transcript is also utilized in order to track progression and see what courses were already taken and completed.

### Input/Output

Inputs would include: login credentials, vocal commands

Outputs: outputs of functions being utilized, class schedule options.

### Subparts

GPA calculator, course catalog viewer, unofficial transcript generator

### Pre and Post Conditions

A user must have a valid log in, a user must be able to speak clearly and concise for the system to understand.

### Error Handling

If the system does not recognize what was said, it will produce an error message stating that it did not understand and to ask again. Ways to test this would be by speaking at different volume levels, different languages, and different types of phrases.

## GPA Calculator

### Abstract

The GPA calculator will consist of simply two modules: calculateGPA and whatIfGPA. The calculateGPA module will calculate the GPA of all of a student’s completed courses with a valid letter grade listed. The whatIfGPA will calculate a student’s GPA with the addition of courses not yet completed that can be added in at the student’s discretion with a vocally inputted letter grade.

### Design

The design of the Grade-Point Average calculator will grab all courses a student has taken and completed, then calculate a GPA based on those letter grades. As job applications often ask for a student’s GPA, this is an essential tool to include

### Exports

The GPA calculator will export a student’s calculated GPA based on all of their completed courses.

### Imports

The GPA calculator will import a list of a student’s completed courses with letter grades, to be converted into a standard 4-point GPA scale. There will additionally be optional user input if the student would like to test out not yet finished classes to determine potential GPA shifts.

### Input/Output

Inputs will be taken from the system’s database while also providing functionality for the user to input potential “What-If” grades to see shifts in GPA. Outputs will be returned to the user in the form of a decimal number between the values of 0.00 and 4.00

### Subparts

The two subparts or modules of the GPA calculator include calculateGPA which will determine the student’s GPA based on completed courses with a letter grade between A and F.

The other module, whatIfGPA must be manually called on and will allow a student to calculate a GPA with courses that are not yet completed that the student is enrolled in.

### Pre and Post Conditions

As a precondition, students must have a student ID and have completed at least one course. There are no postconditions in order to complete this task.

### Error Handling

As a precaution, anything outside of the A to F range will not be included in GPA calculation.

### Side Effects

The basic GPA number will be available at all times.

### Miscellaneous

The calculateGPA module will always be immediately run upon the student opening the page. “What-If” GPAs must manually be selected and calculated.

## Course Catalog Viewer

### Abstract

For students to make an educated decision they require a clear understanding of the available and required courses they will need to take to graduate. The course catalog will include one module that will help students choose the most optimal route to graduate. The module will allow students to view all available courses in the Computer Science/ Information Technology department.

### Design

The course catalog will use sql queries to filter the course list and display them appropriately. Using css the course catalog will have an easy to use interface.

### Exports

The module will export a list of courses that are available.

### Imports

The module will import the list of courses available from the database.

### Input/Output

The main input for the module will be the student asking the system to view the course catalog. The output will be all available courses in the upcoming semester.

### Pre and Post Conditions

As a precondition students must be logged in. This module has no postcondition.

## Unofficial Transcript Generator:

### Abstract

The Unofficial Transcript Generator will be located on the students dashboard/side of the site and will be a place for students to access all the courses they have completed in list form. Along with all the courses that the student has taken in this list it will show when they took the course, and what grade they got on the course. Each student should have access to only their own unofficial transcript student.

### Design

The design for the unofficial transcript will be on the backend with each course that the student has taken attached to the student. On the front-end it will be built with css, and html, and flask framework implementation.

### Exports

The generator will display a list of courses organized by semester taken along with grades associated with each course.

### Imports

The generator will take all the courses the student previously took along with when they took the course and the grade they achieved in the respective course.

### Input/Output

Input: Students login credentials.

Output: The student’s unofficial transcript, they had requested.

### Pre and Post Conditions

Precondition: The student must be logged in for them to access the unofficial transcript.

Postcondition: none

### Error Handling

If the student has not finished the course that they are currently enrolled in it will not appear on their transcript, since the course has not been finished and no grades have been given.

### Side Effects

The user will be able to access this whenever they want their unofficial transcript.

### Miscellaneous

Once the student hits the button to request for the unofficial transcript it will immediately display their transcript.

# Pseudocode

## Login

User accesses the domain of the voice activated advising system

If username == username stored in database

&& password == password store in database:

Login and displays homepage

Else:

System displays error message:

“user credentials are incorrect please try again”

## View course Catalog

This function pulls up the course catalog through the usage of a voice command.

If user requests course catalog:

Query database information on all courses

Output all courses one after the other

If user requests course information:

request which course:

output specific course information

## Unofficial Transcript generator function

This function creates an unofficial transcript for a student

* + The user requests an unofficial transcript through the usage of a voice command

for (every course applicable to the student)

print course information including letter grade,

credits, course name, and course number

return the result to the advisement tool

## Advisement on classes and questions

This function will take the recognizer, microphone, and transcript as its arguments. This will allow the advisement component to understand what the student is asking.

With microphone as source:

request advisement

advisement module checks the database for past enrollments:

advisement outputs courses for the students grade

level, and excludes courses that were previously

taken

## Language processing

System utilizes python's “speech\_recognition” library for speech recognition.

Import speech\_recognition as sprec

(sprec stands for speech recognition)

Create a recognizer instance: recognize = sprec.Recognizer

Create an instance for the microphone:

microphone = sprec.Microphone

(microphone on the hardware)

With microphone as source:

recognize.adjust\_for\_ambient\_noise(source)

Speech = recognize.listen(source)

rec\_Sentence = recognize.recognize\_google(Speech)

Print (rec\_Sentence)

This will output a string for what was interpreted from the microphone and recognized by the google recognizer API.

If rec\_Sentence == “array of words that a user may say to <view course catalog> ”

view\_Course\_Catalog(recognize, microphone)

Else if rec\_Sentence == “array of words that a user may say to <view unofficial transcript>”

unofficial\_Transcript(recognize, microphone)

Else if rec\_Sentence == “array of words that a user may say to <generate gpa>”

GPA(recognize, microphone)

Else if rec\_Sentence == “array of words that a user may say for <academic advisement>”

academic\_advisement(recognize, microphone)

## GPA calculator

This function calculates the user’s GPA through the usage of a voice command

for every course in student’s transcript

if letterGrade = (grade A - F)

increment coursecount

add (call LGN with letterGrade) to gpa

divide gpa by coursecount

return new gpa

## What-If GPA

This function calculates the user’s what-if GPA through the usage of a voice command

for every course in student’s transcript

if letterGrade = (grade A - F)

increment coursecount

add (call LGN with letterGrade) to gpa

for every course the student is enrolled in currently

name a letter grade

convert letter grade to 4.0 scale

increment coursecount

add (call LGN with letterGrade) to gpa

divide gpa by coursecount

return new what-if gpa

## Letter Grade to Number (LGN)

This is a helper function to convert letter grades to values.

double LGN (char num)

switch :

A case for each distinction from A to F including pluses and minuses

return num

## End Session

This allows the user to end the session with the system

user vocally states they want to end session

system ends session

# Detailed Design

Within this document is an encapsulation of the entire design of the automated voice advisement system also known as AVA. With a thorough understanding of how each module works in order to achieve the goal specified by the stakeholders. The system will have the ability to listen to a user's vocal commands, and produce the expected result the user is requesting. After a user signs in with official credentials, the system will have the capabilities to produce a course catalog, display an unofficial transcript for the user, calculate one's GPA for the current and following semester, and allow the user to ask questions and be advised on what courses to take for the semester. All undergraduate students apart of the CS/IT department will have access to use this system to their advantage in achieving academic success.

Users that have access to the system would be all undergraduate students apart of the computer science and information technology department in the school. These students will need a valid ID and password that the school gave them in order to sign in. With valid credentials the user can then log in and will be brought to a home page where they can choose from different functions that the system supports. These functions are: academic advisement, view course catalog, course information, view unofficial transcript, view current GPA, and calculate what-if GPA. The user can then vocally select which function they would like to utilize. Once a student vocally announces which function they would like to use, the system interprets the language, and activates the function. If any problems were to occur with the language interpretation, the system will output an error message such as “Sorry speech not recognized, please try again”.

If the user vocalizes that they would like to partake in academic advisement with the system, the system will activate the academic advisement function which will pull all previously passed courses, and switch the interface to a new page that displays all courses the user can take depending on their grade level. It will output courses for their current year and the year above them, but will exclude all courses that the user has already taken from those two years.

From the homepage, the second function is the ability to view a full course catalog. The user will be able to vocally state “view course catalog” the system will interpret the command and activate the course catalog function. This function will pull all data from the schools database for all classes that are available for computer science and information technology undergraduate students. It will then create a list of all classes that will be available and display all the courses for the user to look through. If the user wants more information on a class, the user may state the course name and the system will display extra information regarding the specified class. Afterwards the module will then return to the homepage where the user can then choose another function or log out.

If the user vocalizes “transcript” the system will instantiate the unofficial transcript function. This function will then pull all the information from the schools database regarding the logged in user. This includes all the classes taken in the past, and all grades that were received from those classes that were taken. The function will then produce an unofficial transcript for the user to view. Afterwards the user can then state that they want to go back to the homepage.

If the user vocalizes that they want to calculate their GPA, they can state “calculate GPA”. The system will then switch interfaces to a new page where the user's current GPA is calculated and displayed. The user can then state that they want to go back home.

If the user vocalizes that they want their “what-if GPA” the system will switch to a new screen that requests for the amount of grades they want to enter, and then states to enter each grade point. The system will then calculate the new GPA with the already established grades, and the newly entered grade points.