AVA An Automated Voice Activated Advisement System

(Test Plan)

By: The Software Gurus

Team Members:

Omar Obidat

Jesse Parron

Allen Asencio

Dan Ferdetta

Sumit Mistry

Software Engineering 515

Fall 2022

Professor Johnson

November 14, 2022

**Table of Contents 2**

**Introduction 3**

Overall objective and success criteria: 3

Summary of the integration plan: 4

Dependency graph 10

Summary of the module-to-test-techniques: 11

Summary of the monitoring, reporting, and correcting procedures: 13

Proposed dates for submission of individual test reports: 14

**Discussion 15**

Defense for the integration plan: 15

Details of tests with objective and success criteria: 15

Details of the monitoring, reporting, and testing procedures: 25

Details of individual team member assignments: 26

**INTRODUCTION**

1. **Overall objective and success criteria:**

The objective of this document is to plan how the testing process of the software is expected to proceed. Testing is one of the key components for a successful launch. This document aims to describe how we will test our software once it is created. Additionally, it will explain the testing methodologies or tools we are planning on using. Each module will be both tested independently, as well as, combined with other related modules. Furthermore, this document will include a schedule of: the plan with which the modules will be tested, the order they will be tested, and the tester who will be performing the respective testing.

Our success criteria will be as follows:

* All modules must be tested using multiple different scenarios. In the event a bug has been found, the module will be fully retested to confirm no additional bugs were introduced in the process.
* Each module must perform as specified in the design and specifications documents. If a change is needed, it should be addressed by the formal Change Request(CR) procedure and revised within the updated specification.
* The testing schedule must account for possible bugs in the software, and include sufficient time for the debugging process.

1. **Summary of the integration plan - a list of tests, dates, and people responsible**

| **Module Number** | **Module Name** | **Role of user** | **Objective** | **Dates** | **Tester Name** |
| --- | --- | --- | --- | --- | --- |
| 1. | Login | Student / MSU administration | Will allow the user to input their login information, and verifies with the database | 11/16/22 | Omar Obidat |
| 2. | Connecting to School Database | Student / MSU administration | After a successful login, the user will be redirected to different pages based on their respective role | 11/16/22 | Omar Obidat |
| 3. | Maintaining database connection | System | Will ensure the user is logged in in order switch between pages, and will redirect to the login page upon logging out or being inactive for too long | 11/16/22 | Omar Obidat |
| 4. | View own GPA | Student | Will calculate the GPA of all of a student’s completed courses with a valid letter grade listed. | 11/17/22 | Dan Ferdetta |
| 5. | Calculate possible GPA | Student / MSU administration | 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 manually inputted letter grade | 11/17/22 | Dan Ferdetta |
| 6. | View Own Transcript | Student | Will be a place for students to access all the courses they have completed in list form | 11/17/22 | Dan Ferdetta |
| 7. | Register for courses | Student | Will give students the ability to register for courses | 11/17/22 | Sumit Mistry |
| 8. | View All available courses | Student/MSU administration | Will allow students to view all available to take courses in a specific semester | 11/17/22 | Sumit Mistry |
| 9. | Filter courses | Student / MSU administration | Will provide parameters for the students to filter courses by time, name, and suitability | 11/17/22 | Sumit Mistry |
| 10. | View course catalog | Student / MSU administration | Will allow students to view all available courses in the Computer Science/ Information Technology department | 11/17/22 | Allen Asencio & Jesse Parron |
| 11. | View course information | Student / MSU administration | Will allow students to view specific information regarding a course | 11/17/22 | Allen Asencio & Jesse Parron |
| 12. | Bypass course limits | MSU administration | Will allow the bypass of course prerequisites and course capacity limits | 11/17/22 | Allen Asencio & Jesse Parron |
| 13. | Add course | MSU administration | Will add a course to the course catalog | 11/17/22 | Allen Asencio & Jesse Parron |
| 14. | Update course | MSU administration | Will update a course in the course catalog | 11/17/22 | Allen Asencio & Jesse Parron |
| 15. | View student’s GPA | MSU administration | Will view any desired student’s GPA | 11/17/22 | Dan Ferdetta |
| 16. | View student’s transcript | MSU administration | Will view any desired student’s transcript | 11/17/22 | Dan Ferdetta |
| 17. | Language recognition | System | Will allow the system to recognize the input from the user that will allow the user to interact with all the other functions within the system. | 11/18/22 | Omar Obidat |
| 18. | Language recognition with GPA viewer. | Student | Allow the student to verbally request to view their GPA. | 11/18/22 | Jesse & Omar |
| 19. | Language recognition with GPA calculator . | Student/MSU administration | The student or administration will be able to verbally ask for the student's GPA to be calculated. | 11/18/22 | Jesse & Omar |
| 20. | Language recognition with transcript viewer. | Student | The student may verbally request to see their unofficial transcript. | 11/18/22 | Jesse & Omar |
| 21. | Language recognition with registering for courses. | Student | The student may verbally register for courses using the academic advisement capabilities. | 11/18/22 | Jesse & Omar |
| 22. | Language recognition with viewing all available courses. | Student/MSU administration | The student or administration may verbally request to view all available courses. | 11/18/22 | Jesse & Omar |
| 23. | Language recognition with filtering courses. | Student/MSU administration | The student or administration may verbally filter the courses. | 11/18/22 | Jesse & Omar |
| 24. | Language recognition with viewing course catalog | Student/MSU administration | The student or administration may ask the system to view the course catalog. | 11/18/22 | Jesse & Omar |
| 25. | Language recognition with viewing course information | Student/MSU administration | The student or administration may request to view the course information. | 11/18/22 | Jesse & Omar |

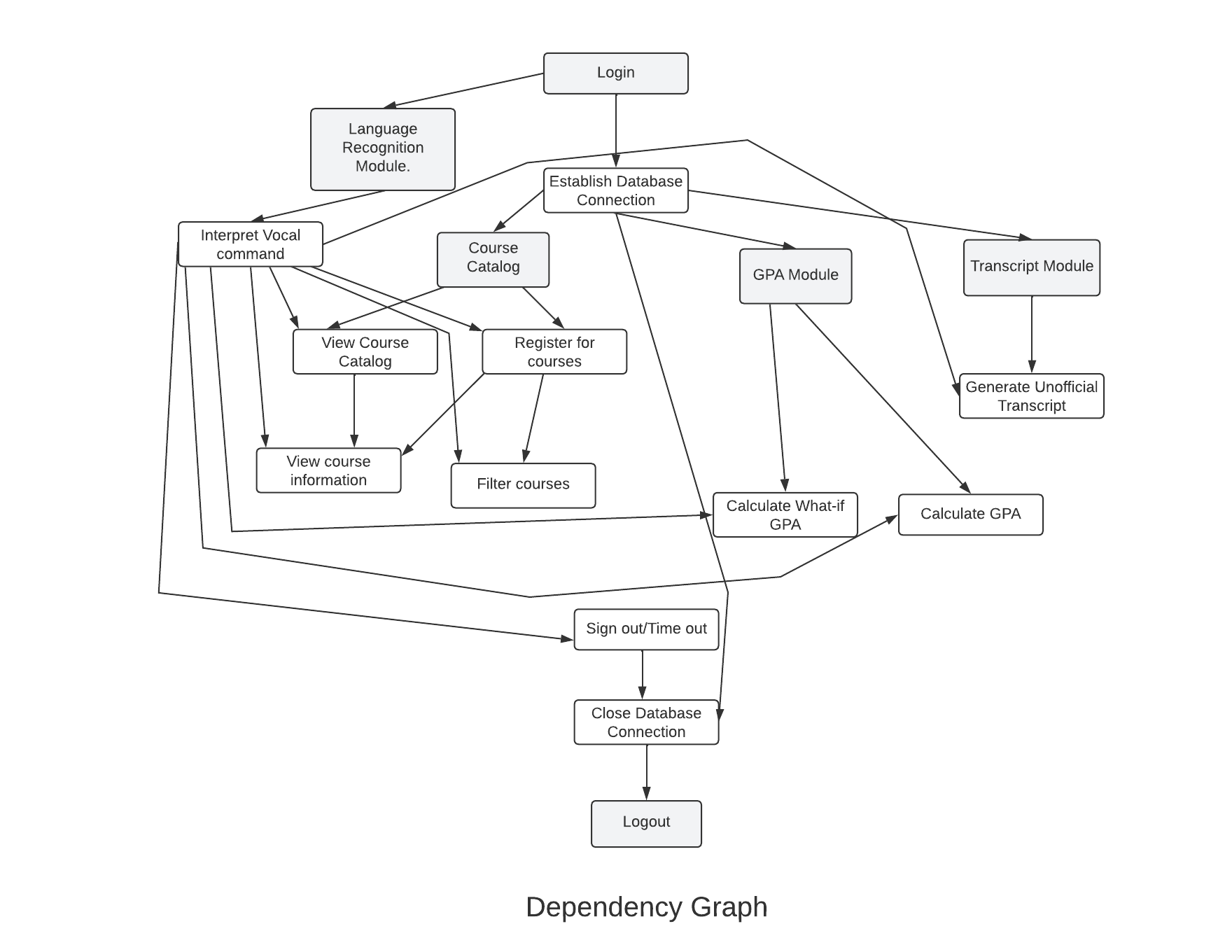
****

Figure 1. **Dependency graph**:

Some modules in the system are dependent on others to be complete and functioning properly. As illustrated in the dependency graph above. By following the dependency graph the testing process of the system will be easier to integrate.

1. **Summary of the module-to-test-technique mapping for the four required testing techniques**

The module that will be mapped with the four required testing techniques will be the **language recognition module.**

| **Testing** | **Walkthroughs** | **Extensive Logic Testing** | **Input/Output Testing** | **Verification** |
| --- | --- | --- | --- | --- |
| Unit Testing | Walk the team through the code for the language recognition module. And allow feedback on the development, if any errors or problems come up. | Build a decision table to list the conditions and the conditions values, as well as the actions and their outputs. This will allow the user to test the language recognition and ensure that it performs as intended in the decision table. | Test the input of the language recognition code, and see what the output is. Using different words or patterns of speech, ensure accuracy is high with interpreting speech. | Ensure the language recognition was developed as per the requirements and works as per the requirements |
| Integration testing | Have the team review and ensure that the code is compatible and performs correctly with other modules. | Build a decision table for the language recognition module interacting with other modules in the system. List the components and actions, and ensure that it is logically accurate and doesn’t encounter fallacies. | With the language recognition module combined with others, test inputs and outputs by verbally stating commands and seeing what the system does. | Ensure that the language recognition component works as intended when integrated with the other models in the system as specified in the specification and requirements documents. |
| Performance evaluation | Each member of the team tests the language module, rating the performance off of: accuracy, speed, bugs, readability, and requirements. | Based off of the decision tables from before, ensure it meets all actions based off of each condition. | Provide a range of inputs for the language recognition module, and review the outputs. Ensure the outputs are correct. | Fully evaluate the module and ensure that it works as specified in the requirements and specification documents. |
| Functional Testing | Check to see if all the features of the language recognition module are debugged and no issues with the interpretation of voice. | Based off the decision table run all the actions, to make sure all work to full functionality | Test the input of the language recognition code, and see what the output is. Using long phrases, patterns of speech, or quick conversations to ensure accuracy is high with interpreting speech. | Fully evaluate the whole module. |

1. **Summary of the monitoring, reporting, and correcting procedures**

After testing, each team member is expected to submit a full report explaining the error found and possible solutions. Once each member of the team completes the testing phase, the team can work together on resolving the bugs found and reinitiating the testing phase to provide a high quality software that is bug free using the below tools and techniques.

The team will be utilizing a web-based application called Trello. It allows for lists to be made with progression indicators. This will allow the team members to notify each other of the progress they have made. Not only that, but lists of bugs could be made and added to anytime. All work that is completed is updated with a green check mark, and indicates who has completed it with their initials embedded. Trello will allow us to work efficiently and be organized by allowing us to have clear expectations and understandings between all team members. Lists such as: what is still being coded, what has a bug, what can be tested, the type of tests being performed, and what modules are fully implemented and correct as per the specifications requirements documentation. These are all examples of the type of lists that will be implemented in our Trello workspace. A separate list will be utilized to specify which modules need to be corrected and can be picked up by other members who are finished.

1. **Proposed dates for submission of individual test reports**

| **Individual** | **Proposed Individual Test Report Date** |
| --- | --- |
| Omar Obidat | November 19th, 2022 |
| Jesse Parron | November 19th, 2022 |
| Allen Asencio | November 20th, 2022 |
| Dan Ferdetta | November 20th, 2022 |
| Sumit Mistry | November 20th, 2022 |

**DISCUSSION**

1. **Defense for the integration plan (about half a page)**

For our integration plan we decided to start with the database and login module because other modules are dependent on it working properly. Each module is going to be tested separately and created as functions to allow us to be able to test them separately with text input and as a complete system with the language recognition module. Each team member was assigned testing based on their experience in that field. Omar and Jesse have experience with language recognition and database integration that's why he will handle the testing of related modules. Allen has experience in security testing that is why he will be handling the administration view testing. Danial and Sumit have experience in front end testing that is why Danial will be testing all the modules that are related to the GPA. and Sumit will be testing modules related to the course catalog. The dates were chosen to allow for any bugs to be handled before the deadline while also allowing each member to work on their respective modules without waiting for another member to finish theirs.

1. **Details of the tests with objective and success criteria for each test or group of tests**

|  | **Module** | **Test Case** | **Expected Result** | **Actual Result** | **Test Status** | **Test Input Data** | **Comment** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | Login | Successful login | The user is redirected to their designated role’s home page |  |  |  |  |
|  | Login | Unsuccessful login due to having incorrect credentials | The user is unable to login and an error message is displayed |  |  |  |  |
| 2 | Connecting to School Database | Login is accepted | Connection proceeds |  |  |  |  |
|  | Connecting to School Database | Login is not accepted | Connection does not proceed |  |  |  |  |
| 3 | Maintaining database connection | User continues interacting | User stays logged in |  |  |  |  |
|  | Maintaining database connection | User chooses to log out | User is logged out |  |  |  |  |
|  | Maintaining database connection | User does not interact for 15 minutes | User is logged out |  |  |  |  |
| 4 | View own GPA | The student views their GPA | The GPA is correctly calculated and applied for that respective student |  |  |  |  |
| 5 | Calculate possible GPA | Student inputs complete what-if course information | The what-if GPA is applied accurately |  |  |  |  |
|  | Calculate possible GPA | Student inputs incomplete or incorrect what-if course information | The what-if GPA does not apply and an error is displayed |  |  |  |  |
| 6 | View Own Transcript | Student chooses to view their transcript | The transcript is shown and applied correctly |  |  |  |  |
| 7 | Register for courses | Student is able to register for a course | The student can register for the course and it is applied to their ID |  |  |  |  |
|  | Register for course | Student is unable to register for a course due to invalid prerequisite | The student cannot register for that course |  |  |  |  |
|  | Register for course | Student is unable to register for course due to class capacity reached | The student cannot register for that course |  |  |  |  |
|  | Register for course | Student is unable to register for course due to it overlapping with an existing course | The student cannot register for that course |  |  |  |  |
| 8 | View All available courses | Student requests to view all available courses for their degree | All courses that are not full and count toward that student’s degree are visible |  |  |  |  |
|  | View All available course | Student requests to view all available courses for their degree | All courses not applicable to the student are not visible |  |  |  |  |
| 9 | Filter courses | Users filter by selecting from available courses. | Filter is applied correctly |  |  |  |  |
|  | Filter courses | User chooses to filter by typing the course name | Filter is applied correctly |  |  |  |  |
| 10 | View course catalog | User requests to view the course catalog | The course catalog is visible and shows all courses |  |  |  |  |
| 11 | View course information | User will enter the required course information and click the submit button. | The course information will be visible |  |  |  |  |
| 12 | Bypass course limits | Bypass course size limit | MSU administration is able to allow a student to register for course regardless of size limit |  |  |  |  |
|  | Bypass course limits | Bypass prerequisite | MSU administration is able to allow a student to register for course regardless of prerequisites |  |  |  |  |
| 13 | Add course | MSU Admin chooses to create a course with accurate information | The course will be added |  |  |  |  |
|  | Add course | MSU Admin chooses to create a course with incomplete or inaccurate information | The course will not be added and an error will be shown |  |  |  |  |
|  | Add course | MSU Admin chooses to create a course that already exists | The course will not be added and an error will be shown explaining that a course with the same information already exists |  |  |  |  |
| 14 | Update course | MSU Admin chooses to update a course with proper updates | The course will be updated |  |  |  |  |
|  | Update course | MSU Admin chooses to update a course but does not properly fill out fields | The course will not be updated and an error will be shown |  |  |  |  |
| 15 | View student’s GPA | Administrator chooses to view GPA of a student with an existing student ID | The GPA of that student is listed |  |  |  |  |
|  | View student’s GPA | Administrator chooses to view GPA, but types a student ID that does not exist | An error will show on the screen showing that student could not be found |  |  |  |  |
| 16 | View student’s transcript | Administrator views student’s transcript using a proper student ID | The transcript is shown as it should |  |  |  |  |
|  | View student’s transcript | Administrator attempts to view a transcript of a student with an incorrect ID | The transcript is not shown as the student does not exist |  |  |  |  |
| 17 | Language recognition | Wrong input is given | The system asks the user to repeat the input. |  |  |  |  |
|  | Language recognition | Proper input is given | The system replies with the correct action requested by the user |  |  |  |  |
|  | Language recognition | Input that has already been processed is given again | The system replies that the action has already been fulfilled |  |  |  |  |

1. **Details of monitoring, reporting, and testing procedures**

In order to monitor the progression of all the modules completion, the team is incorporating the usage of a web application named Trello. With the help of this web app, members will be able to monitor other members progress of their assigned modules. Trello allows for task tracking by creating lists and tasks. Members will have tasks assigned, the task will show an indicator that the objective has not been completed. Once the task has been completed, members can then update the task to complete and store the task within the completed list. This will notify members that the task has been completed. If bugs, or problems occur members can insert comments, and store the tasks within the specified lists. Once tasks are inserted into the completed list, they can then be moved into the testing list for the specified member to test. If bugs occur during testing, the bug should be reported, and then store the task in the bug list until it is updated and fixed. We can also specify the tests that need to be accomplished by inserting the task in multiple lists. If we decide to do three different tests, then we can store the same task in three different lists, and update upon completion. This will allow for the team to stay organized and up-to-date monitoring of everyone's progression as the project moves forward. If a member is not completing their assigned duties in a timely manner, they will be notified to complete their tasks before the due date.

1. **Details of individual team member assignments**

| **Team Member** | **Assignments** |
| --- | --- |
| **Omar Obidat** | Creating a database and establishing connection between frontend and back end. Establishing login, logout, and session data/security. |
| **Jesse Parron** | Implementing language recognition and connecting with other modules. |
| **Allen Asencio** | Unofficial Transcript Generation |
| **Dan Ferdetta** | GPA calculation and transcript viewing |
| **Sumit Mistry** | Create user friendly front end interface for all modules within the system as per the specifications. |
| **Additional for everyone** | Ensure functionality and testing of the components assigned by the due date specified. Follow up with assigned tasks on Trello. |