

Equitable Course Recommendation Program,
For The Clark County School District

Project Group #4: Lois Cumming, Jennifer Deng, Chandana Jamjam,
Prashanna Prasai, Michael Sisk

Management of Information Systems, UNLV

MIS 740: Project Report

Han-Fen Hu

December 5, 2023

1. Introduction

1.1 Background and Business Problem

Research conducted and published by the Thomas B. Fordham Institute ranked the largest fifty metro areas in the U.S. for school “quality” and unfortunately Clark County ranked 49th out of 50 metro areas.¹ Other national report cards and state education annual rankings place the state of Nevada as 50th out of 50 states and the District of Columbia. The rankings are based on three measures: chance for success, school finance and K-12 achievement. The data points included in the rankings are: fourth and eighth grade math and reading proficiency, high school graduation rates, and Advanced Placement (AP) coursework that is offered to students by the College Board, the organization that administers the SAT and PSAT.

In the context of school districts, equitable access to courses is crucial to ensure that every student receives an education that meets their needs and abilities. However, one prevalent challenge is the potential influence of behavioral patterns and human (teacher/counselor) bias on course recommendations for students. These biases can inadvertently limit a student’s educational opportunities and success and hinder their access to courses they are quantitatively qualified for. In order to address this issue, our aim is to design and develop a course recommendation program that relies on quantitative assessment data rather than subjective qualifications. This program will help standardize and level the playing field for what level of education a student has access to. Furthermore, this program intends to help remove the bias from the recommendation process and overall offer equitable educational opportunities to all students.

1.2 Functional Requirements

This program will have input in the form of data collection and database storage of that information. The program will utilize a database of user-collected student information on their assessment scores, including but not limited to the raw score, scaled score, passing or failing, the type of assessment, when the assessment was taken, and percentile of the score. Depending on the assessments taken at the school district and grade level, the program will allow the end user to input the parameters that they need and define their own dictionary list for their school district. For example, if a school district uses PSAT scores, they can set the program to collect data relating to PSAT scores and then standardize what the scores mean and the baseline scores for

¹ Kenny Guinn for Policy Priorities. (2022, April). *What Nevada Can Learn: Phases One & Two - April to June 2022*. Retrieved from Nevada Government Nevada Education Rankings: https://doe.nv.gov/uploadedFiles/ndedoenvgov/content/Boards_Commissions_Councils/State_Board_of_Education/2022/July/Guinn_Center_Nevada_Education_Rankings_Phase_I_and_II_Report_June_2022.pdf

Project Report: Group 4

what is needed in each course. A PSAT score of X will qualify a student for the Y course, or a PSAT score of Z will qualify a student for W course.

The initial set-up will need the school district to determine the requirements for each course they wish to use assessment data for. After identifying the courses, what assessment data will be used and what is the baseline needed? What grade levels and combinations of scores will meet the course requirements for recommendation? The user can use the program with few parameters, or very complex parameters depending on the needs of the school district. The data collection and storage will allow users to insert new student records, delete student records, and update existing records in the database. This data storage functionality is essential for maintaining an up-to-date repository of student information.

The program will then go through various processing functions including but not limited to data analysis and computation. The program will analyze the quantitative assessment data to identify each student's baseline for where they fall into the parameters set by the school district. It will then compute the suitable course recommendations based on the analysis, considering factors only driven by data. For example, in taking the PSAT, the various subject areas can be broken down and a school district may set that if a 11th grader receives a score of X they will qualify for Class Y, or if that same 11th grader receives a score of Z they will qualify for Class W. The program can generate statistics and reports with the assessment data to provide insights into the recommendation process, ensuring transparency and accountability. It can also show trends, areas of weakness within the school district, and insight into areas for improvement. It could also show school-specific statistics and establish baselines for each school to cater to school-specific strategic goals to better serve our diverse schools within the district by allowing administrators to tailor baselines for the school.

The output is the course recommendations for each student being evaluated with the program. The output will include data visualizations such as charts, figures, and graphs. The visual aids will help school administrators and advisors make informed decisions about course recommendations. This will also give information on how to use school funds, the number of sections needed per course, and how many math, science, English, social studies, or elective teachers to hire. The output will drive how the school district and individual schools within the school district organize their organization and business goals.

1.2.1 Functionalities

Key functional requirements:

1. Data collections and Analysis: The program needs to collect relevant data about the students.
2. Course Recommendation: The program should have an algorithm that can analyze the collected data and recommend appropriate courses for each student

Project Report: Group 4

1.2.2 Dataset

Changes were made to the dataset to reflect simpler string values in the Test Name and to merge the two datasets.

School Number	Student Number	Last Name	First Name	Gender	Grade	Test Code	Test Name	Score
403	682166	Hinojosa	David	M		11 PSAT-1	Reading and Writing	460
403	682166	Hinojosa	David	M		11 PSAT-2	Math	390
403	682732	Zacharie	Manuel	M		11 PSAT-1	Reading and Writing	330
403	682732	Zacharie	Manuel	M		11 PSAT-2	Math	320
403	682780	Renteria Delgado	Demian	M		11 PSAT-1	Reading and Writing	400
403	682780	Renteria Delgado	Demian	M		11 PSAT-2	Math	440
403	683183	Alfaro Cortez	Dante	M		11 PSAT-1	Reading and Writing	420
403	683183	Alfaro Cortez	Dante	M		11 PSAT-2	Math	490

User ID	Course Name	Current Grade	Test Code	Test Name	Requirement
DENGJS	English 12	11	PSAT-1	Reading and Writing	< 400
DENGJS	English 12 H	11	PSAT-1	Reading and Writing	400 - 500
DENGJS	AP English Literature	11	PSAT-1	Reading and Writing	500 - 700
DENGJS	AP Independent Study	11	PSAT-1	Reading and Writing	> 700
DENGJS	Algebra II	11	PSAT-2	Math	< 400
DENGJS	Algebra II H	11	PSAT-2	Math	400 - 500
DENGJS	Precalculus H	11	PSAT-2	Math	500 - 600
DENGJS	Calculus AB	11	PSAT-2	Math	600 - 650
DENGJS	Calculus BC	11	PSAT-2	Math	650 - 720
DENGJS	Calculus III	11	PSAT-2	Math	> 720

1.3 Business Value

This Equitable Course Recommendation Program holds significant business value for school districts and educational institutions. This program and the school districts that utilize it will leverage equitable access to courses for all students, irrespective of behavioral patterns or human biases. It also promotes a fair educational environment where students are recommended for courses based on their academic capabilities and interests. Additionally, by automatic course recommendations, this program will reduce the administrative and scheduling burden on school staff. This streamlined administrative task process will free up time and resources that can be redirected toward other school tasks. In that same line of thought, this program will also allow the school district to use data-driven decision-making. The program's data analysis and output features will empower school administrators and advisors to make data-driven decisions as well. It will provide valuable insights into student performances and course demand trends, enabling proactive course planning. By tracking quantitative measures, the school district can use data to fund and build future initiatives with data-driven measurable outcomes.

Overall, the Equitable Course Recommendation Program seeks to address the critical issue of bias and variability in course recommendations by utilizing the power of quantitative assessment data. By incorporating data storage, analysis, computation, and visual outputs, the

Project Report: Group 4

program aims to provide equitable education for all students while simplifying administrative tasks and supporting informed decision-making. This program significantly improves the quality and equity of education and helps provide a more diverse, equitable and inclusive education experience for all students.

2. Interface

2.1 Description of Changes

The interface of this project will be mostly implemented as a command-line (using jupyter notebook IDE).

Changes were made to the user interface as the program was developed so this section has been revised since the Solution Design document was submitted. The program is now composed of three modules: (1) the Student Information Program, (2) the Course Requirement Program and (3) the Course Recommendation Program.

Some sample screens are shown here to indicate the types of functions each module contains. Please see the **User Manual**, section 6 of this paper, to view screenshots of all the functions contained in the three modules.

2.2 The Student Information Program

The program begins by asking the user to create a user ID. Next, there is a question regarding whether this is the first time using the program and a request to enter a school ID number (4 numbers) to create a database. The program displays the “Main Menu” where the user can choose from four options:

Main Menu:

1. Input new student information
2. Update existing student information
3. Delete existing student information
4. Exit

Enter your choice (1/2/3/4):

2.3 The Course Requirement Program

The purpose of this program is to allow school administrators to add course names, current grades, test codes, test names and requirements to the database.

Project Report: Group 4

When the program runs, the user is first asked to enter their user ID (capital first letter of their first name, last name, and numbers (i.e., Jdeng1221, Lcumming1048). Next, the user is asked if this is the first time he/she is using the program. If the user types “yes”, the user is asked to enter the numerical school ID.

From there, the “Course Recommendations Menu” allows the user to create new courses, update course information and delete course information.

```
Course Recommendations Menu:
1. Create a new course
2. Update course information
3. Delete course information
4. Exit
Enter your choice (1/2/3/4): 1
Enter the new course name: English 12 H
Enter the current grade: 11
Enter the test code: PSAT-1
Enter the test name: Evidence Based Writing
Enter the requirement for the course: 400-500

Do you want to add more courses? (yes/no):
```

2.4 The Course Recommendation Program

This program displays a table showing the Course Name, Requirement and Number of Students that have taken that test. To begin, the user will be asked to input a “Test Code”:

Enter a test code:

Examples of test codes are shown in the following table:

Course Name	Current Grade	Test Code	Test Name	Requirement
English 12	11	PSAT-1	Reading and Writing	< 400
English 12 H	11	PSAT-1	Reading and Writing	400 - 500

Project Report: Group 4

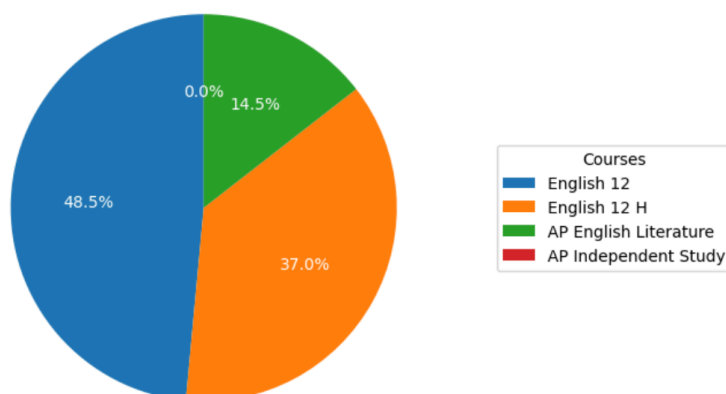
A table showing the number of students who have met the requirement and a pie chart showing the distribution of students is displayed.

Enter a test code: PSAT-1

Results:

Course Name	Requirement	Total Students
English 12	< 400	328
English 12 H	400 – 500	250
AP English Literature	500 – 700	98
AP Independent Study	> 700	0

Distribution of Students in Courses

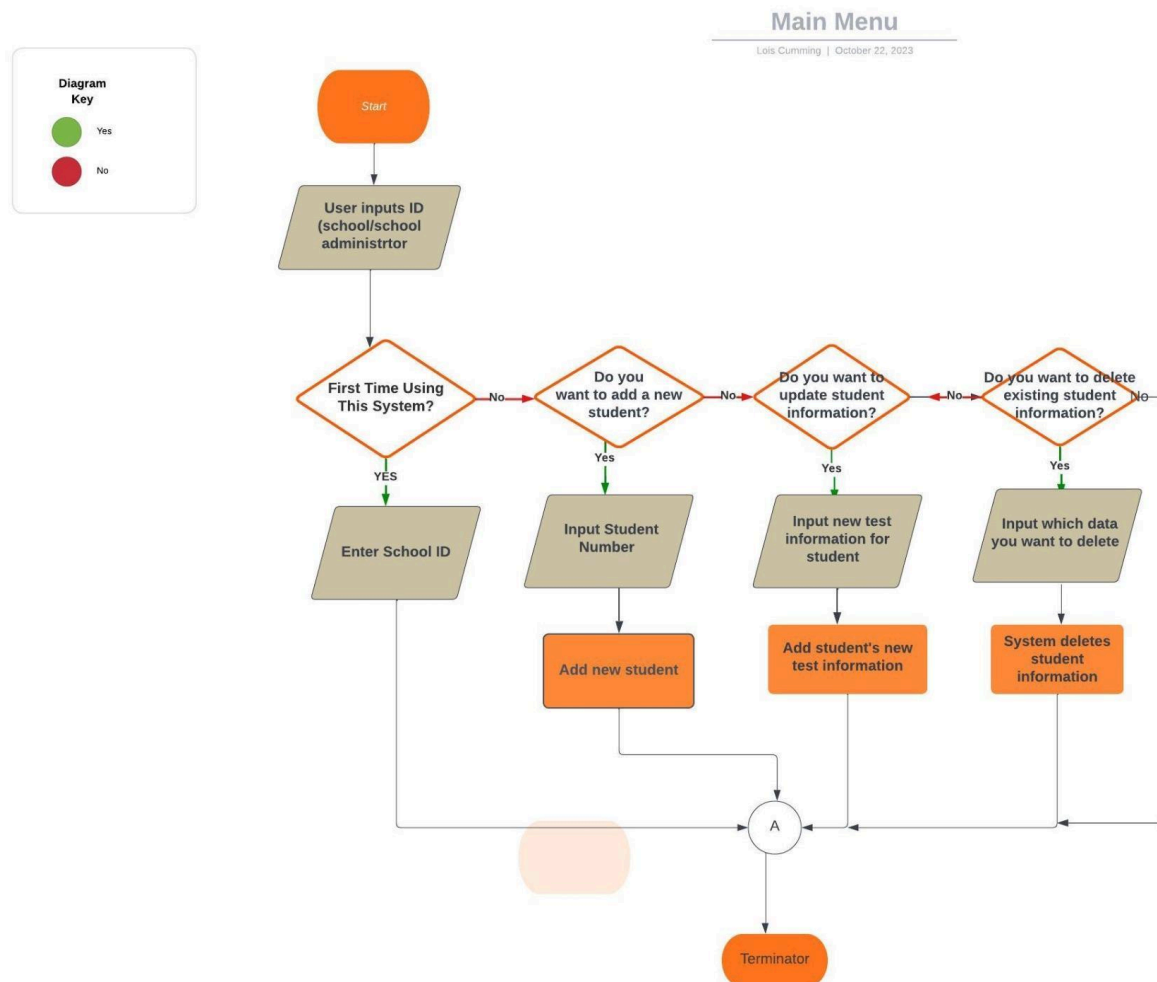


3. Detailed Design

In this section, the functions are described using text and diagrams in the form of flowcharts. The flowchart is to show the logic of each function.

3.1 Main Menu for Student Data Assessment Collection

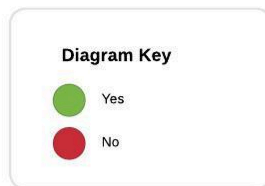
First part of the program is the Main Menu. This function of the program asks the end user to input their end user ID. The end user is presumed to be the school/school administrator. Then the program will prompt the user to indicate if this is the first time they are using the program. If they are, then the program will ask for the school ID number to create a database for them and then terminate this function to move on to the next. If this is not the first time they are using the program, then the program will prompt the user to decide if they wish to input new student information, update existing student information, or delete existing student information. The program then terminates and moves on to the next function.



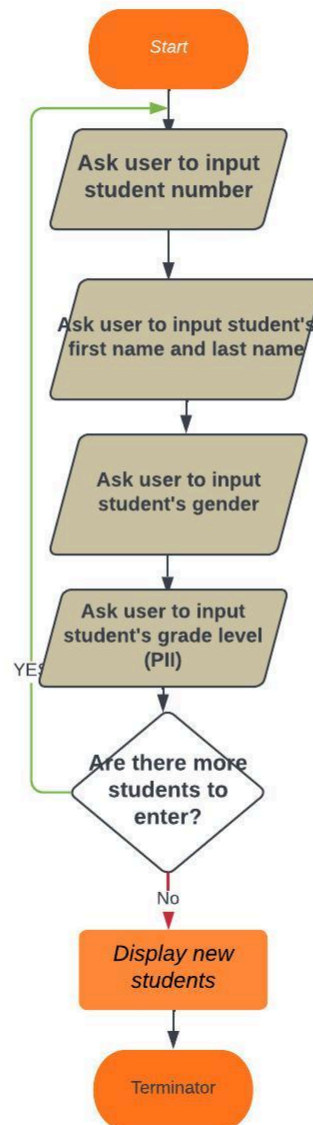
Project Report: Group 4

3.2 Create New Student

The program starts with the option and ability to create and enter new student information. The user will input the student number. Then the user will enter in the last name and first name associated with the student number. The user will then enter the student's gender and student's grade level. The program will ask if there are more students to be entered. If there is, the loop starts over. If there are no more new students to create a profile for, the program displays the new students entered and terminates the program after writing them to the dictionary/database.

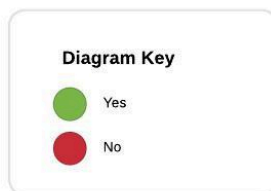


Create New Student

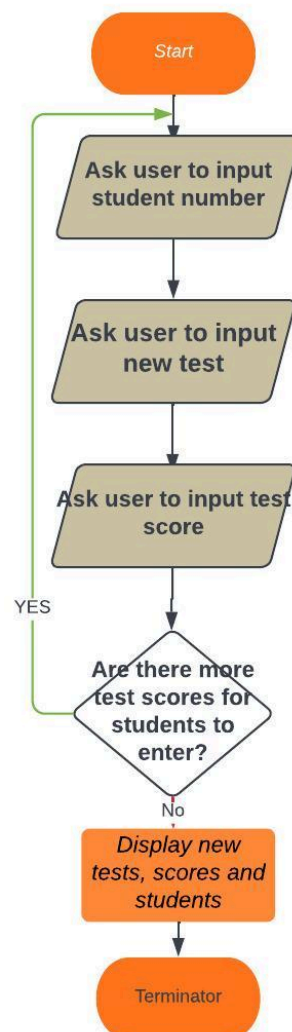


3.3 Add New Tests and Scores For Student

The program starts and asks for the user to input the student number. This will locate the student profile in the database. Then the user inputs new tests, both the name and the test code. Then the user enters in the test score associated with that test. The program asks if there are more test scores to enter. If there are more, the program loops back so the user can enter in new tests and scores. If there are no more tests and scores to be entered, the program displays the students, new tests, and the respective scores.



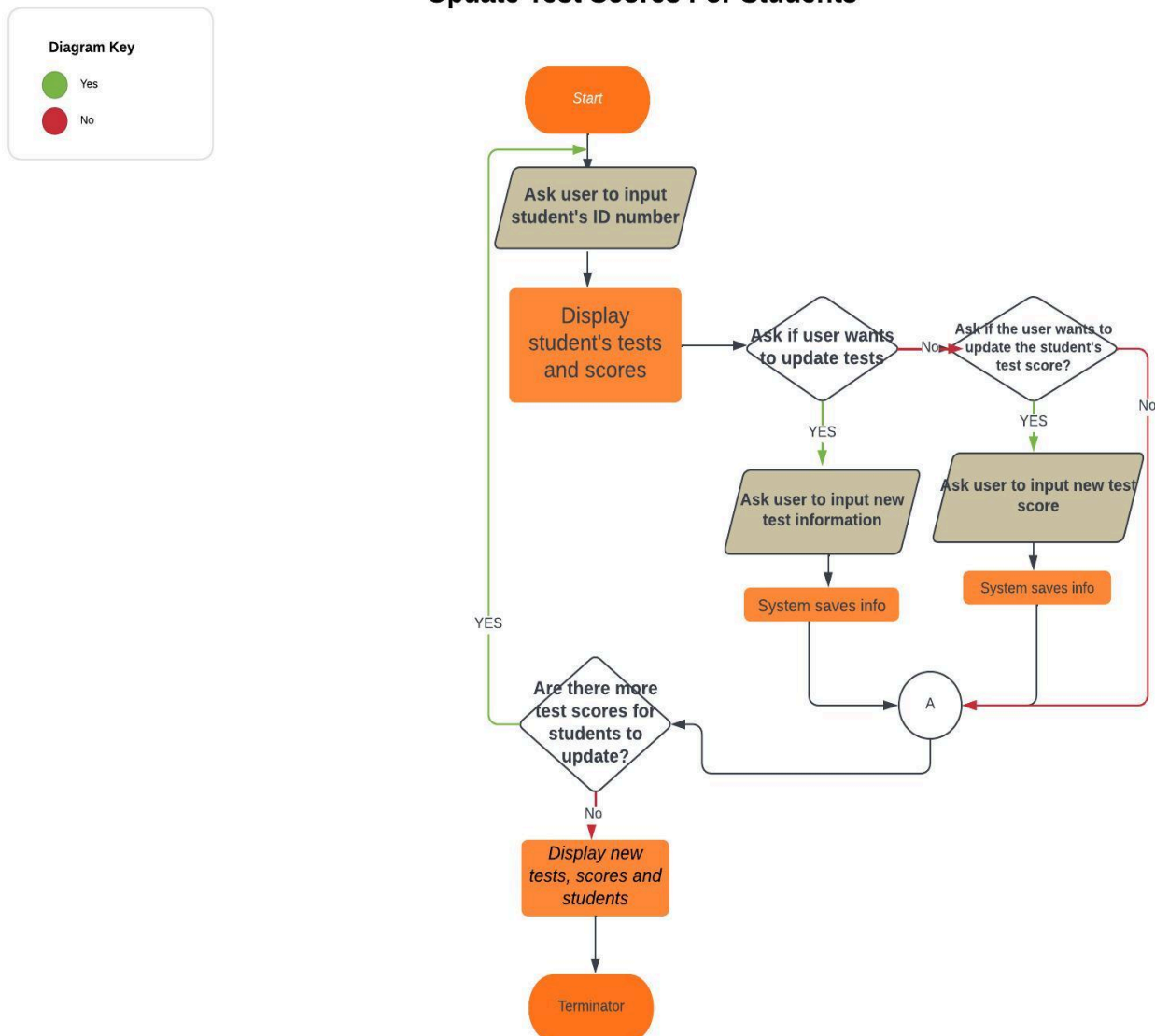
Add New Test Scores For Students



3.4 Update Test Scores For Students

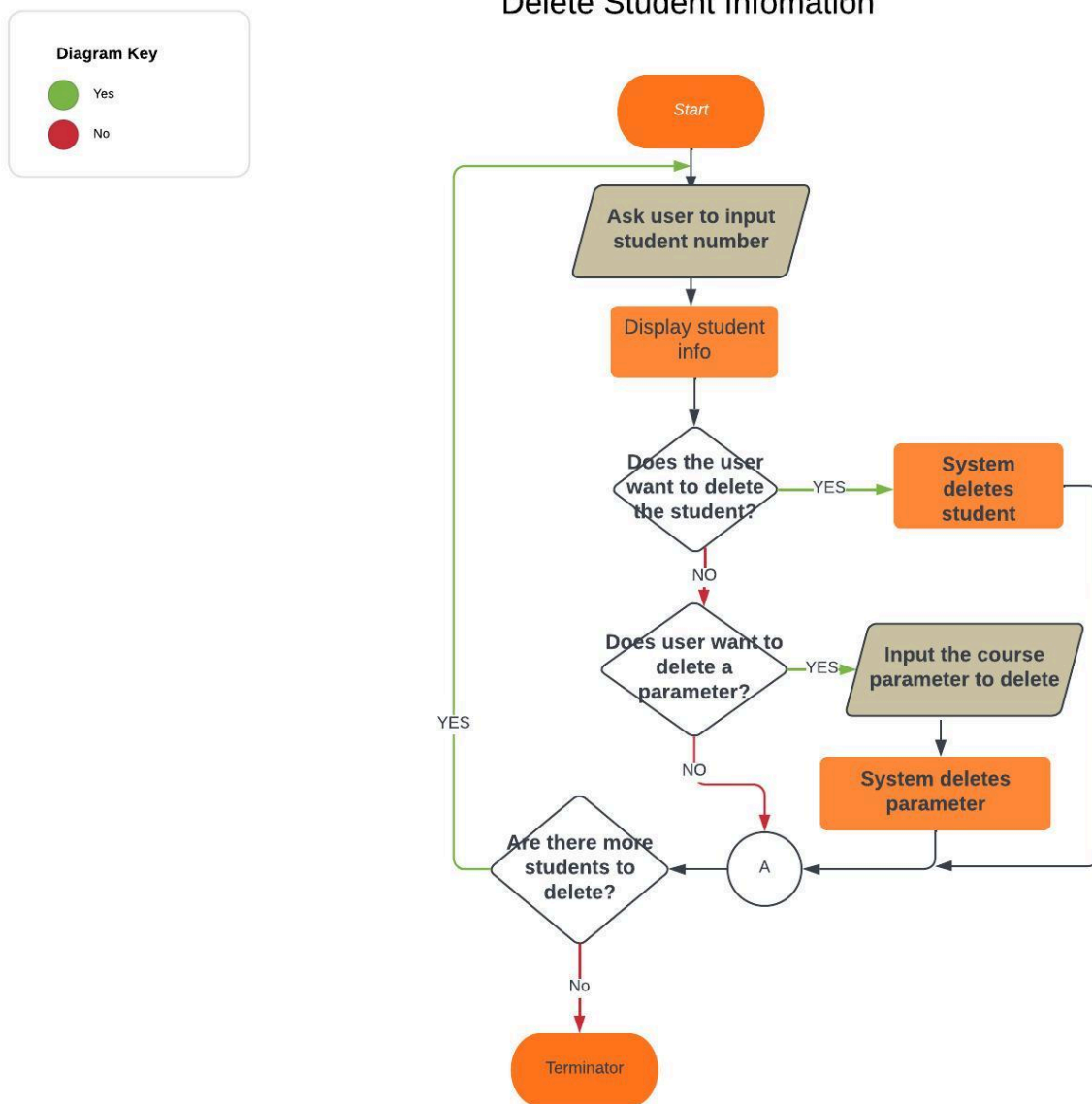
The program starts by asking the user to input a student number to locate the student's profile. The program will then display the student's tests and scores as they are in the database. The program will ask if the user wants to update the tests. If the user does, then the program will ask the user to input the newest test information and then saves over the old information. If the user wants to update the scores for a student, then the user can input the new test score and the system will save the information. The program will then ask if there are more test scores for students. If there are more test scores to be updated, then the program loops back to ask for the student number, if not, the program displays the new tests, scores, and students and then ends the program.

Update Test Scores For Students



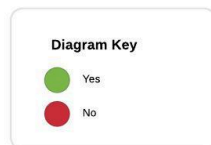
3.5 Delete Existing Student Information

The program starts when the option to delete existing student information is selected from the main menu. If the user wants to delete the student information, the program will ask the user to input the student number to locate the student's profile. This will display the information of the student. The program will ask if the user wants to delete the student profile. If yes, the program deletes the student profile and all associated tests and scores from the database. If the user does not want to delete the student, the program will ask if the user wants to delete a parameter and if so, to input the parameter that the user wants deleted. The system will then delete the parameter and ask if the user wants to delete any other student information. If yes, the program restarts and if no, the program ends.

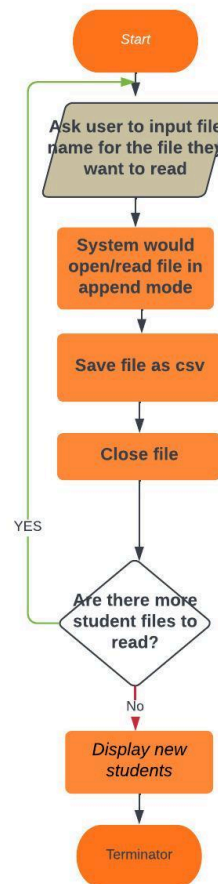


3.6 Read

The program will start by prompting the user to input the file name for the file they want the program to read. Then the program will open the file in append mode so that way the user can add in additional information or display information. Then the program will save the file as csv. The program will close the file so any changes made are saved, and the information will display. The program will then terminate.



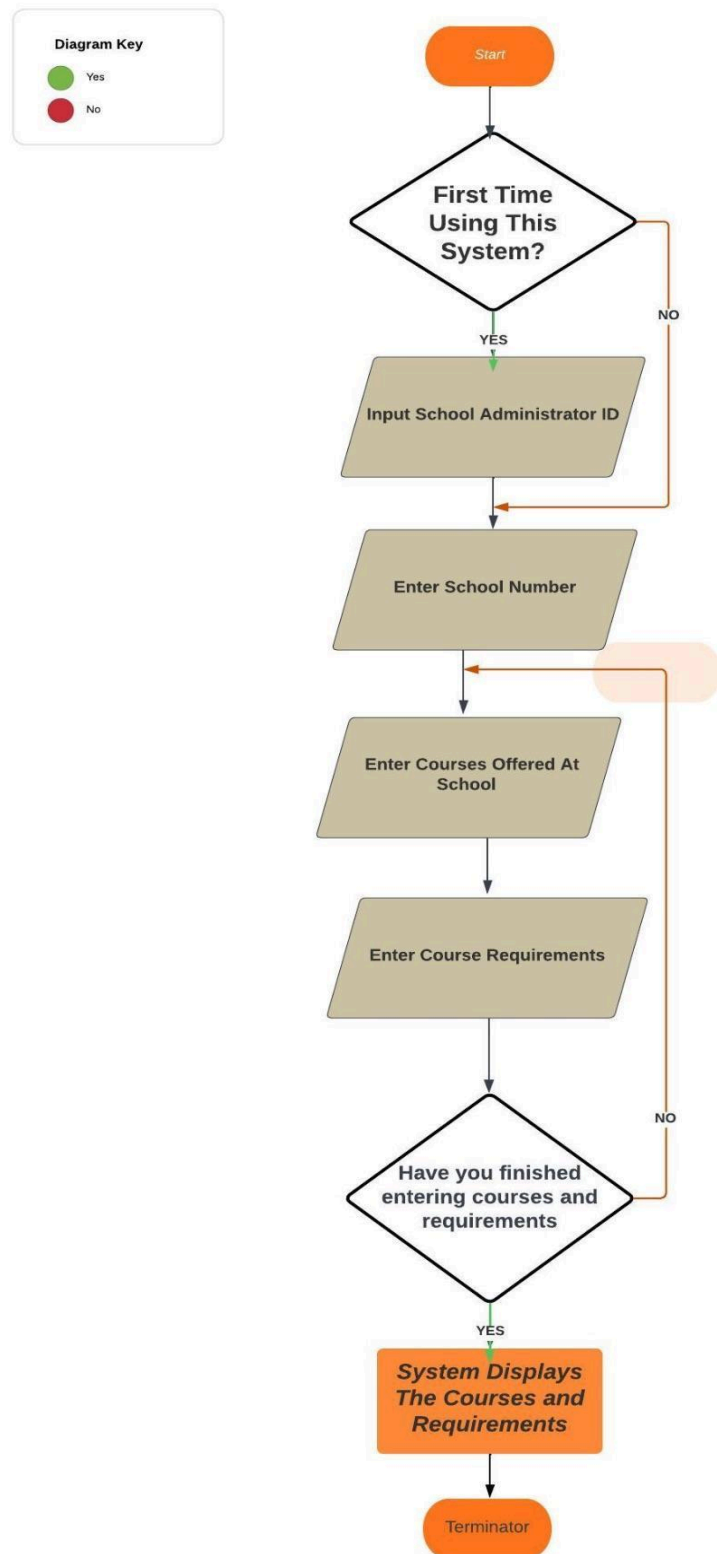
Read Student File



3.7 Main Menu for Course Recommendation

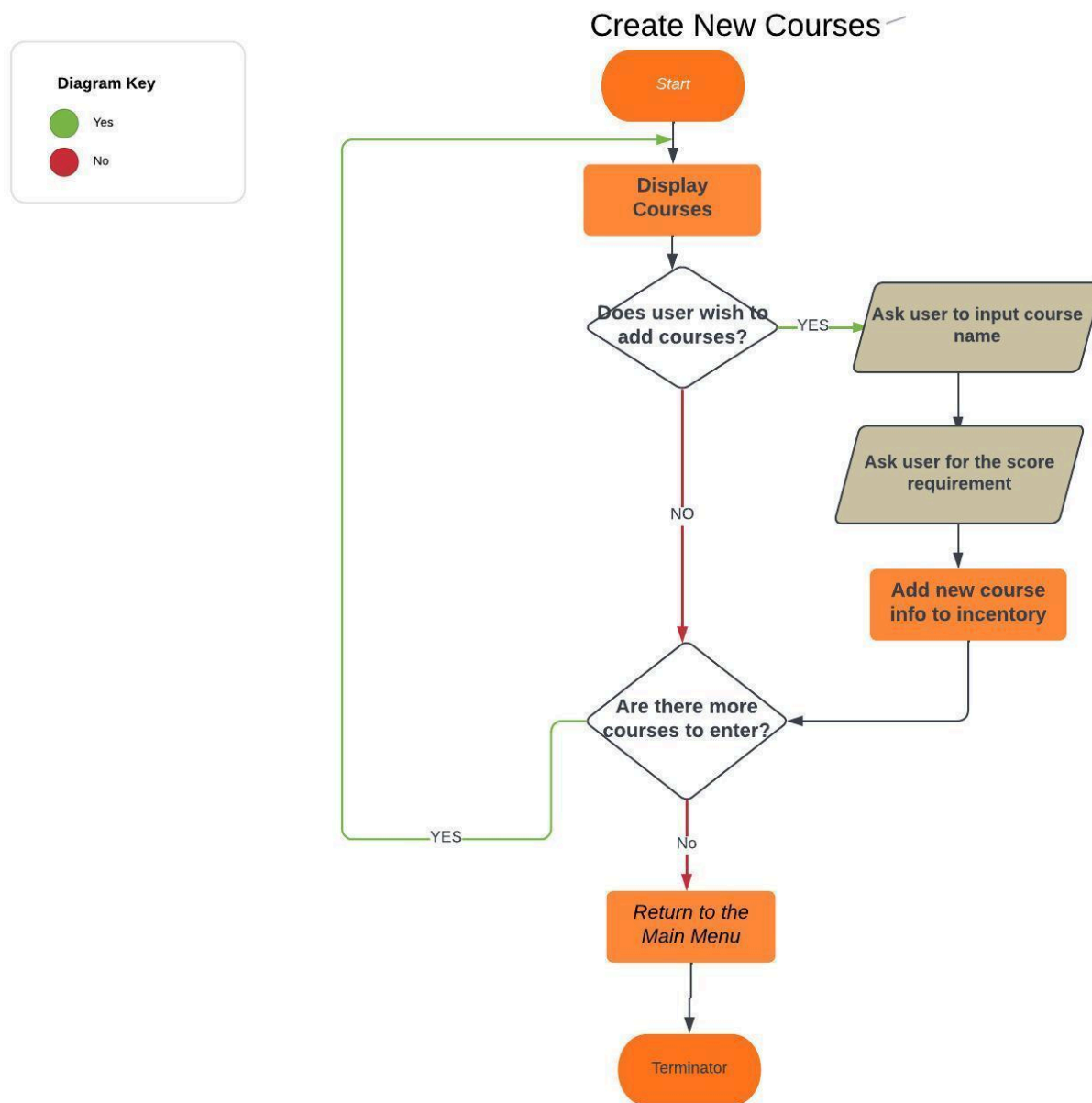
The program will start with the main menu for the course recommendations portion. The program will ask if it is the user's first time using the program. If the user has not used the program before then the program will prompt the end user to enter in their ID. If the user has used the program before then the user will enter the school number to begin entering course names and parameters. The end user will enter in the course names and then the course requirements to see if a student's score will qualify for the course. The program will ask if the user is done entering courses and requirements. If they are not, the program will loop, if they are, the program will display what the user has entered and then terminate the program.

Main Menu For Course Recommendations



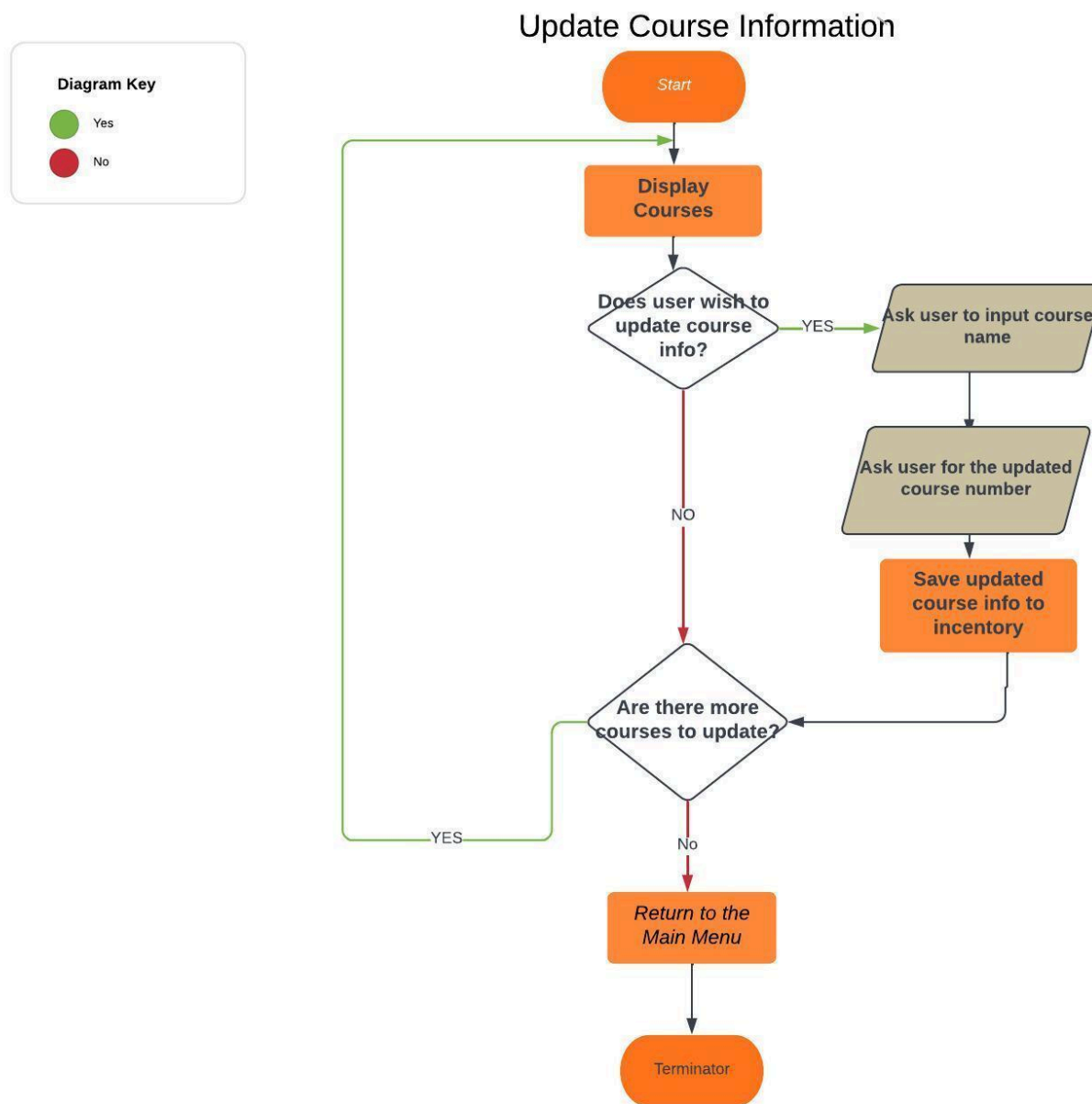
3.8 Create New Courses

The program will start by displaying the courses that are in the system. Then the program will ask if the user wishes to add more courses. If they do, then the user will input the course name and the score requirement for the course. The course will be added into the inventory. Then the program will ask the user if they wish to enter more courses, and if they do it loops back. If they don't then the program will display the courses in the system and return to the main menu and terminate from there.



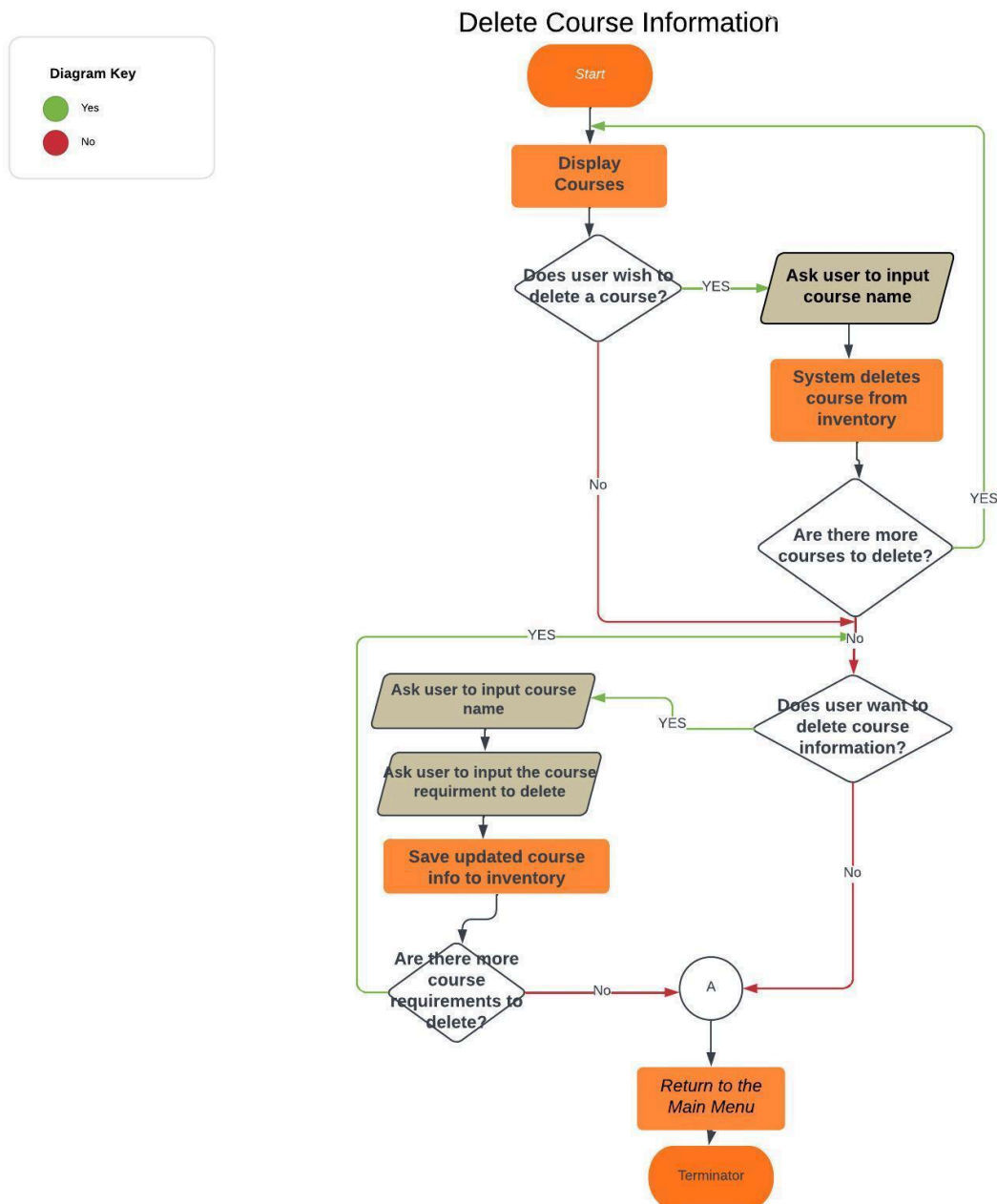
3.9 Update Course Information

The program will start by displaying the courses in the system. Then the program will ask if the user wants to update course info. If they do not, then the program will go back to the main menu and terminate. If they do, then the program will prompt the user to enter the course name to locate the data to update. The user will enter in the updated course information and requirements. Then the program will save the updates into the inventory and ask if the user has any more courses they want to update. If they do, the program loops, if not, the user returns to the main menu and the program terminates.



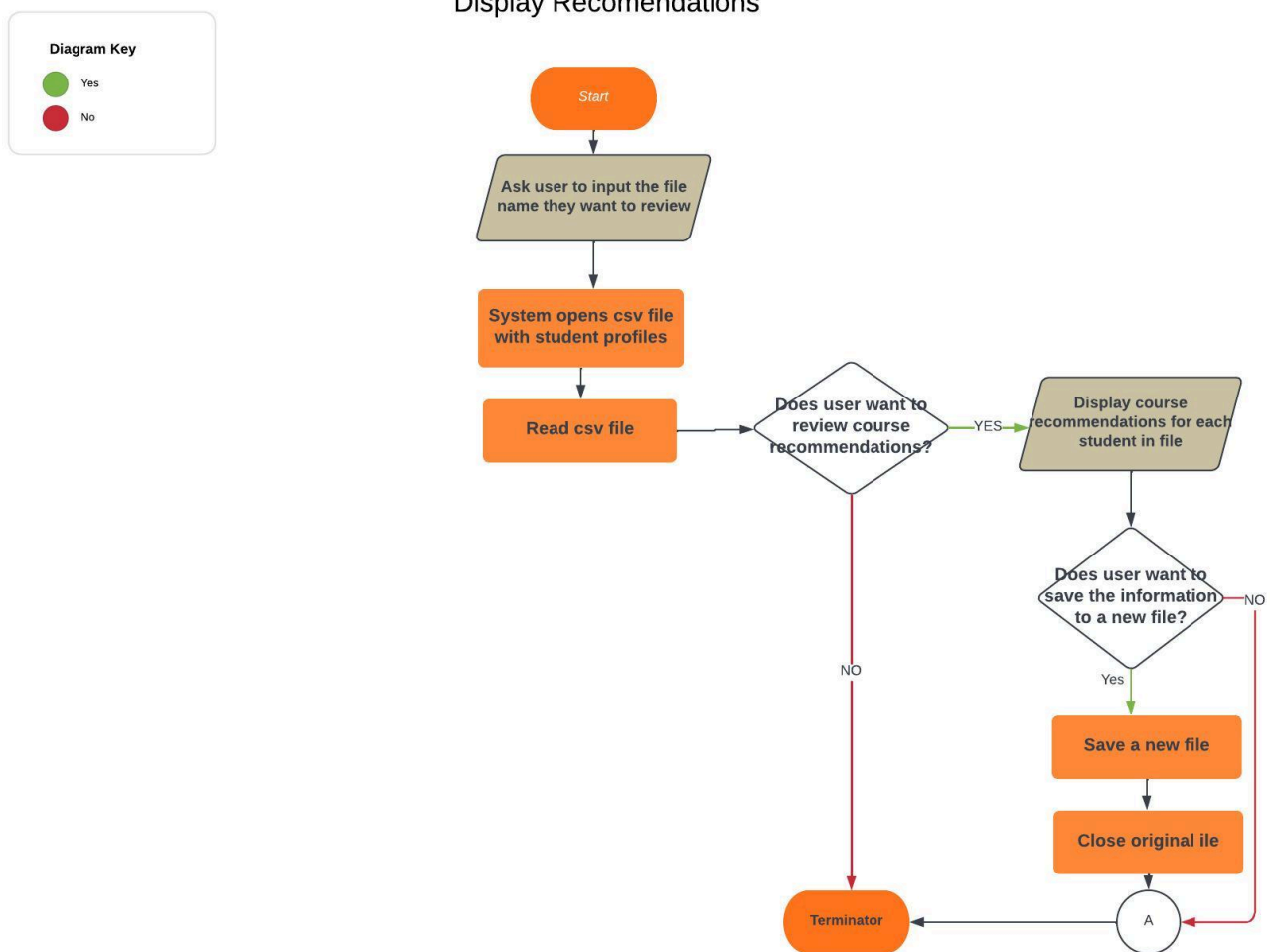
3.10 Delete Course Information

The program starts by displaying the courses that are in the system. The program will then ask if the user wishes to delete a course entirely. If the user does, then the program will ask for the course name, and the program will delete the course name and information from the inventory. Then the program will loop to ask if the user wants to delete any more course or course information. If the user does not want to delete a course, then the program will ask if the user wants to delete information on the course such as the requirement. If not, then the program asks if there are any courses to delete and if not, brings the user back to the main menu and terminates the program. If the user does want to delete course information then the program will prompt the user to enter in the information that needs to be changed by locating the course name and the requirement to update. Then the updates are saved to the inventory and loops to ask if the user has more courses or course information they wish to update. If the user does not, then the program brings the user to the main menu and terminates.



3.11 Display Recommendations

Once the program starts, ask the user to input the file name of the student profiles that contain the student number, name, test codes, and scores. Then the program will read and parse through the file. The program will ask the user if they wish to display course recommendations based on the parameters and the information on the file. If yes, the program will run and display the course recommendations for each student on the file. The user will then have the option to save the information to a new file. The program will close the original file of the student profiles. The program will then terminate.



Project Report: Group 4

4. File or Database Design

The program will create, read, update, and delete its data in the .csv file format. We are using sanitized student data sourced from the Clark County School District as well as course requirements from CCSD to determine our recommendations, but our users will be able to fully manipulate (CRUD) student data and course requirements using our program.

Our first data dictionary covers the courseRequirements.csv

Variable Name	Data Type	Description
course_ID	String	Name of Course
grade	Integer	Grade level of the Course
test_ID	String	ID of relevant test
test_Score_Range	Integer	Acceptable Score Range

course_ID	grade	test_ID	test_Score_Range
English 12	11	PSAT-1	< 400
English 12 H	11	PSAT-1	400 - 500
AP English Literature	11	PSAT-1	500 - 700
AP Independent Study	11	PSAT-1	> 700
Algebra II	11	PSAT-2	< 400
Algebra II H	11	PSAT-2	400 - 500
Precalculus H	11	PSAT-2	500 - 600
Calculus AB	11	PSAT-2	600 - 650
Calculus BC	11	PSAT-2	650 - 720
Calculus III	11	PSAT-2	> 720

Project Report: Group 4

Our second data dictionary is a little more complex, hosting student data, in this case, our sanitized data from CCSD. We collect and use student identifying information and their applicable test scores.

Variable Name	Data Type	Description
school_Number	Integer	School ID Number used by Districts
student_Number	Integer	Student ID Number to Uniquely identify students
last_Name	String	Last Name of Student
first_Name	String	First Name of Student
gender	String	Gender of Student
grade	Integer	Grade Level of Student
test_PSAT1	String	Test ID for the PSAT1
test_Score_PSAT1	Integer	Student Test Score for PSAT1
test_PSAT2	String	Test ID for the PSAT2
test_Score_PSAT2	Integer	Student Test Score for PSAT2

School Number	Student Number	Last Name	First Name	Gender	Grade	Test Code	Test Name	Score
403	682166	Hinojosa	David	M		11 PSAT-1	Reading and Writing	460
403	682166	Hinojosa	David	M		11 PSAT-2	Math	390
403	682732	Zacharie	Manuel	M		11 PSAT-1	Reading and Writing	330
403	682732	Zacharie	Manuel	M		11 PSAT-2	Math	320
403	682780	Renteria Delgado	Demian	M		11 PSAT-1	Reading and Writing	400
403	682780	Renteria Delgado	Demian	M		11 PSAT-2	Math	440
403	683183	Alfaro Cortez	Dante	M		11 PSAT-1	Reading and Writing	420
403	683183	Alfaro Cortez	Dante	M		11 PSAT-2	Math	490

5. Project Fulfillment Estimation

The following breakdown will go over the requirements necessary to fulfill the project. The subsections that will be necessary to achieve success for this project will reference areas within this project design.

5.a Flow Control

The basis of our flow control is outlined in section 2 and section 3. Section 2 goes over the interface design that shows the main menu and how the program will proceed based on the input from the user. In section 3, we display the flowcharts of each process in the CRUD requirements. Each function the user accesses includes flow control of the overall program.

5.b Functions

Functions are gone over in section 1, in our introduction of what the project solution design ultimately hopes to accomplish using each function in our program. Each part of the program is a separate function that will produce information into a data set that the end user can manipulate, use, and achieve the goal of a more equitable education recommendation.

5.c Lists, Dictionaries and/or Dataframe

For this project solution design we will be utilizing lists in the form of dictionaries, that will save and be readily available in csv formats to the end users. Section 4 of this solution design goes over some of the data dictionaries.

5.d File or DBMS Operations

The project will be able to create, read, update, and delete files within the dictionary set for the end user depending on the end users needs at the time of use. Section 2 and section 3 outline the areas in which the input from the end user controls the flow of how the file and data is manipulated.

5.e Data Computation and Visualization

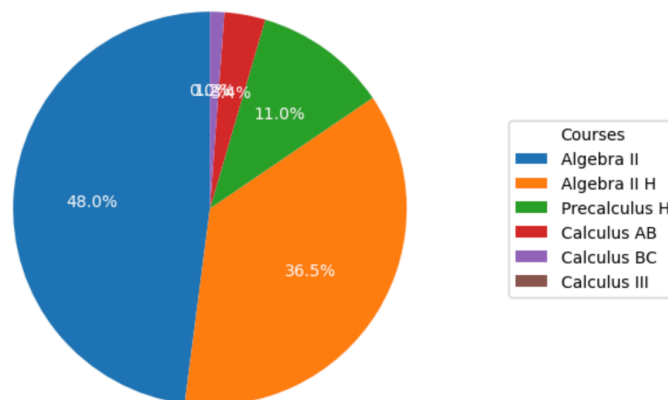
Once all student information and assessment scores have been inputted into the dictionary, and the end user has put in all the course names and requirements for the courses based on the test scores, graphs and charts can be used to visualize the data. This can help the end user and the school have information on subsets of the population and can sort them by grade and gender. The school district as a whole can also utilize this and group the data into charts and figures by school number and region.

This can help determine the level of rigor at each school, and the number of teachers that will be needed for the upcoming school year.

Enter a test code: PSAT-2
Results:

	Course Name	Requirement	Total Students
0	Algebra II	< 400	328
1	Algebra II H	400 - 500	250
2	Precalculus H	500 - 600	75
3	Calculus AB	600 - 650	23
4	Calculus BC	650 - 720	8
5	Calculus III	> 720	0

Distribution of Students in Courses



6. User Manual

Welcome to the Clark County **Equitable Course Recommendation Program**

Equitable access to courses is crucial to ensure that every student in the Clark County School District receives an education that meets their needs and abilities. The initial set-up relies on the school district determining the requirements for each course they wish to use assessment data for. After identifying the courses, decisions regarding what assessment data will be used and the baseline needed. Once the school district has decided which grade levels and combinations of scores meet the course requirements for recommendations, school administrators will be able to use this program to make course recommendations based on quantitative data and analysis. The program's data collection and storage will allow users to insert new student records, delete student records, and update existing records in the database. This data storage functionality is essential for maintaining an up-to-date repository of student information.

This program has been implemented as a command-line interface using Jupyter Notebook IDE, so please follow the instructions in this link to download the program for your Windows or Apple computer ([Installing Jupyter Notebook — Jupyter Documentation 4.1.1 alpha documentation \(test-jupyter.readthedocs.io\)](https://jupyter.readthedocs.io/en/latest/install.html)). Once you have Jupyter installed, there are five files that need to be uploaded to Jupyter Notebook in order to run the program:

1. Student Information Program Final Version.ipynb
2. Course Requirement Program Final Version.ipynb
3. Course Recommendation Program Final Version.ipynb
4. coursesfinal.csv (created after running program 2)
5. students.finalcsv (created after running program 1)

This program consists of three modules:

1. **The student information program**
2. **The course information program**
3. **The course recommendation program**

6.1 Student Information Program

After uploading the five files in Jupyter Notebook, begin by opening the Student Information Program. Click on Cell and choose Run All.

1. The program will ask you to create your user ID. Please input your school district ID (first letter of your first name capitalized, your last name, numbers). For example, my name is Lois Cumming and my ID is Lcumming1034.

Create a user ID:

2. The program will then ask if this is the first time using the program:

Create a user ID: Lcumming1034

Is this your first time using the program? (yes/no):

Project Report: Group 4

3. Next, answer the question regarding whether this is the first time using the program and enter your school ID number (4 numbers) to create a database:

Create a user ID: Lcumming1034

Is this your first time using the program? (yes/no)

Enter the school ID number to create a database:

4. The program will then show you a menu of options and ask what you would like to do:

Main Menu:

1. Input new student information
2. Update existing student information
3. Delete existing student information
4. Exit

Enter your choice (1/2/3/4):

5. After choosing which function you want to use, when prompted, enter your school ID #:

Main Menu:

1. Input new student information
2. Update existing student information
3. Delete existing student information
4. Exit

Enter your choice (1/2/3/4): 1

Enter the school ID number:

6. If you select 1, to **input new student information**, the program will ask you to enter the student number (6 numbers), the student's last name, first name, gender (M,F,X) and grade (use numbers for the grade, do not spell it out):

Main Menu:

1. Input new student information
2. Update existing student information
3. Delete existing student information
4. Exit

Enter your choice (1/2/3/4): 1

Enter the school ID number: 7628

Enter the student number: 123456

Enter the student's last name: Wang

Enter the student's first name: Caroline

Enter the student's gender: F

Enter the student's grade:

Project Report: Group 4

7. Then the program will ask if there are any more new students to create:

Are there any other students to create? (yes/no):

8. If you choose “no”, the program will show the Main Menu again and ask you to choose an option:

Main Menu:

1. Input new student information
2. Update existing student information
3. Delete existing student information
4. Exit

Enter your choice (1/2/3/4):

9. Choosing #2 (**Update existing student information**) will ask for the student number:

Enter the student number to locate the student's profile:

10. If the student number exists in the database, the program will display the student's information:

Main Menu:

1. Input new student information
2. Update existing student information
3. Delete existing student information
4. Exit

Enter your choice (1/2/3/4): 2

Enter the student number to locate the student's profile: 123456

Student Information:

Student Number: 123456

Last Name: Wang

First Name: Caroline

Gender: F

Grade: 11

11. The program will display the **update** menu and then ask you to make a choice between the following:

Update Menu:

1. Add new tests and scores
2. Update existing test scores
3. View student and test scores
4. Return to main menu

Enter your choice (1/2/3/4):

Project Report: Group 4

12. Choosing “**Add new tests and scores**” means the program will ask for the new test name, the test code, and the student’s score. After this information is added, the program will ask if you have more test scores to enter. (Examples of test names are: Evidence Based Writing (code: PSAT-1) and Evidence Based Math (code: PSAT-2).

Update Menu:

```
1. Add new tests and scores
2. Update existing test scores
3. View student and test scores
4. Return to main menu
Enter your choice (1/2/3/4): 1
Enter the new test name: Evidence Based Writing
Enter the new test code: PSAT-1
Enter the test score for the student: 700
```

Are there more test scores to enter? (yes/no):

13. Choosing “no” will bring you back to the “**Update Menu**” where you can choose to go back to the “**Main Menu**”.

Update Menu:

```
1. Add new tests and scores
2. Update existing test scores
3. View student and test scores
4. Return to main menu
```

Enter your choice (1/2/3/4):

14. Choose #3 to “**Delete existing student information**”. The system will ask for the user to enter the student number to locate the student’s data:

Main Menu:

```
1. Input new student information
2. Update existing student information
3. Delete existing student information
4. Exit
```

Enter your choice (1/2/3/4): 3

Enter the student number to locate the student's data:

Project Report: Group 4

15. After entering the student number, the program will display the student's information and ask if you want to delete the student profile and all associated tests and scores? If the answer is yes, the program confirms the student's profile is deleted.

```
Student Information:
Student Number: 123456
Last Name: Wang
First Name: Caroline
Gender: F
Grade: 11
Do you want to delete the student profile and all associated tests and scores?
(yes/no): yes
Student profile deleted.
```

16. The user will see the “**Main Menu**” again and if #4 is chosen, to “**Exit**” in the program, the final function asks the user if they want to display the contents of the csv. file.

Note #1: The Student Information Program will look to see if the user has a file named: studentsrawdata.csv. If the user is running the program for the first time that file won't exist yet, so the program will create, open in append mode, write the information and close the file. The next time the user is running the program to enter more data or update information, the program will locate the file, read the lines of the file, store it and open the file in append mode so the user can continue to work on the studentsrawdata.csv dataset.

Note #2: When the user is ready to run their dataset with the Course Recommendation Program, they will need to make a copy of the studentsrawdata.csv file and rename it to studentsfinal.csv so the two files can be maintained independently between the programs.

6.2 Course Requirement Program

The purpose of this program is to allow school administrators to add course names, current grades, test codes, test names and requirements.

This is a sample of the data needed to input into this program for courses; current grade, test codes, test names and the requirement is as follows:

Course Name	Current Grade	Test Code	Test Name	Requirement
English 12	11	PSAT-1	Reading and Writing	< 400
English 12 H	11	PSAT-1	Reading and Writing	400 - 500

Project Report: Group 4

AP English Literature	11	PSAT-1	Reading and Writing	500 - 700
AP Independent Study	11	PSAT-1	Reading and Writing	> 700
Algebra II	11	PSAT-2	Math	< 400
Algebra II H	11	PSAT-2	Math	400 - 500
Precalculus H	11	PSAT-2	Math	500 - 600
Calculus AB	11	PSAT-2	Math	600 - 650
Calculus BC	11	PSAT-2	Math	650 - 720
Calculus III	11	PSAT-2	Math	> 720

1. When the program runs, the user is asked to enter their user ID (capital first letter of their first name, last name, and numbers (Jdeng1221, Lcumming1048)).

Enter your user ID:

2. Next, the user is asked if this is the first time using the program:

Is this your first time using the program? (yes/no):

3. Then the user is asked to enter the numerical school ID:

Enter the school ID number to create a database:

Project Report: Group 4

4. The user is presented with a menu and asked to make a choice:

```
Enter your user ID: Lcumming1048
Is this your first time using the program? (yes/no): yes
Enter the school ID number to create a database: 7869
Database created for school ID 7869.
```

Course Recommendations Menu:

1. Create a new course
2. Update course information
3. Delete course information
4. Exit

Enter your choice (1/2/3/4):

5. Choose #1 to **create a new course** and fill in a course name:

Course Recommendations Menu:

1. Create a new course
2. Update course information
3. Delete course information
4. Exit

Enter your choice (1/2/3/4): 1

Enter the new course name:

6. Then enter the current grade:

Enter the current grade:

7. Enter the test code:

Enter the test code:

8. Enter the test name:

Enter the test name:

9. Enter the requirement for the course:

Enter the requirement for the course:

400-500

10. Finally, the program will display the information that has been added and then ask if the user wants to add more courses.

Project Report: Group 4

Course Recommendations Menu:

1. Create a new course
2. Update course information
3. Delete course information
4. Exit

Enter your choice (1/2/3/4): 1

Enter the new course name: English 12 H

Enter the current grade: 11

Enter the test code: PSAT-1

Enter the test name: Evidence Based Writing

Enter the requirement for the course: 400-500

Do you want to add more courses? (yes/no):

11. **Update course information:** if the user selected “no” they do not want to add more courses, the program will confirm the previous entry and then return to the Course Recommendations menu. If the user wants to update course information, select #2 from the menu:

Courses:

English 12 H - Current Grade: 11, Test Code: PSAT-1, Test Name: Evidence Based Writing, Requirement: 400-500

Course Recommendations Menu:

1. Create a new course
2. Update course information
3. Delete course information
4. Exit

Enter your choice (1/2/3/4):

Course Recommendations Menu:

1. Create a new course
2. Update course information
3. Delete course information
4. Exit

Enter your choice (1/2/3/4): 2

Do you want to update course information? (yes/no):

12. After selecting “yes”, the program asks the user to enter the course name to locate the data (NOTE: the user can only update course information for courses they entered using the “Create a new course” function) :

Enter the course name to locate the data:

Project Report: Group 4

13. Next, the program will display the information about that course and ask the user to enter the parameter to update:

Current Data for English 12 H:

Current Grade: 11, Test Code: PSAT-1, Test Name: Evidence Based Writing, Requirement: 400-500

Enter the parameter to update (current grade/test code/test name/requirement):

14. After entering the parameter, the program will ask the user to enter the updated requirement:

Course Recommendations Menu:

1. Create a new course
2. Update course information
3. Delete course information
4. Exit

Enter your choice (1/2/3/4): 2

Do you want to update course information? (yes/no): yes

Enter the course name to locate the data: English 12 H

Current Data for English 12 H:

Current Grade: 11, Test Code: PSAT-1, Test Name: Evidence Based Writing, Requirement: 400-500

Enter the parameter to update (current grade/test code/test name/requirement): requirement

Enter the updated requirement:

If the user has no more updates, the program will return the user to the main menu.

15. Delete course information: choose option #3 from the menu.

Course Recommendations Menu:

1. Create a new course
2. Update course information
3. Delete course information
4. Exit

Enter your choice (1/2/3/4):

16. The program will ask if the user wants to delete course information.

Do you want to delete course information? (yes/no):

Project Report: Group 4

17. If the answer is yes, the program will ask if the user wants to delete the entire course.

Do you want to delete an entire course? (yes/no):

18. If the user does not wish to delete the entire course, the program will ask the user to input the course name to locate the data. Then the user will be asked which course parameter the user wants to delete:

Courses:

English 12 H - Current Grade: 11, Test Code: PSAT-1, Test Name: Evidence Based Writing, Requirement: 500

Do you want to delete course information? (yes/no): yes

Do you want to delete an entire course? (yes/no): no

Enter the course name to locate the data:

Courses:

English 12 H - Current Grade: 11, Test Code: PSAT-1, Test Name: Evidence Based Writing, Requirement: <400

Do you want to delete course information? (yes/no): yes

Do you want to delete an entire course? (yes/no): no

Enter the course name to locate the data: English 12 H

Current Data for English 12 H:

Current Grade: 11, Test Code: PSAT-1, Test Name: Evidence Based Writing, Requirement: <400

Enter the parameter to delete (current grade/test code/test name/requirement):

The program will ask the user twice if they have any more courses they want to delete.

19. Once the user has finished creating new courses, updating course information, deleting course information, the user can choose “Exit” from the menu and the program will ask if the user wants to display the contents of the CSV file.

Course Recommendations Menu:

1. Create a new course

2. Update course information

3. Delete course information

4. Exit

Enter your choice (1/2/3/4): 4

Do you want to display the contents of the CSV file? (yes/no):

Note #1: The Course Requirement Program will look to see if the user has a file named: coursesrawdata.csv. If the user is running the program for the first time that file won't exist yet, so the program will create, open in append mode, write the information and close the file. The next time the user is running the program to enter more data or update information, the program will locate the file, read the lines of the file, store it and open the file in append mode so the user can continue to work on the studentsrawdata.csv dataset.

Note #2: When the user is ready to run their dataset with the Course Recommendation Program, they will need to make a copy of the coursesrawdata.csv file and rename it to coursesfinal.csv so the two files can be maintained independently between the programs.

6.3 Course Recommendation Output and Visuals

Running this program requires installing pandas and matplotlib.

import pandas as pd

import matplotlib.pyplot as plt

1. The program begins by asking the user to enter a test code:

Enter a test code:

- a) The program will display a table showing the Course Name, Requirement and Number of Students that have taken that test:

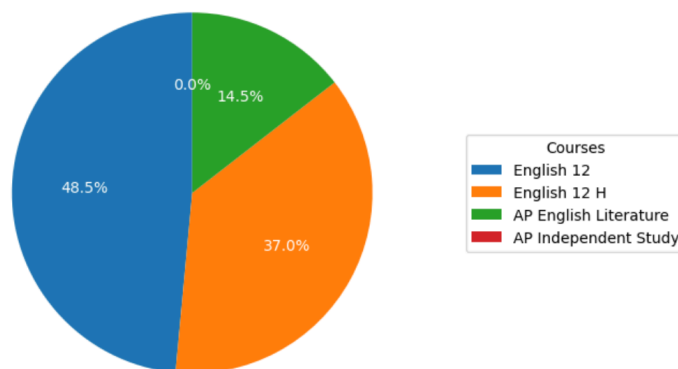
Enter a test code: PSAT-1

Results:

Course Name	Requirement	Total Students
English 12	< 400	328
English 12 H	400 – 500	250
AP English Literature	500 – 700	98
AP Independent Study	> 700	0

- b) A pie chart displaying the data:

Distribution of Students in Courses



7. Response Document

The response we received on the Solution Design document was “ The design document is very well done. It is detailed and clear. According to your design, it fulfills the project requirements”. As a result of this feedback, the workflow design did not change. The biggest change was the user interface was revised after the program was developed. The User Manual that was developed after the program code was finalized, gives the users instructions for using the modules and reflects the changes that were made.