**Functional Requirements:**

1. The system shall allow a user to add a Student Grade record to database.
   1. The system shall display a main menu to the user.
   2. The user shall select “Add Record” button to add a new record.
   3. The system shall display a New Grade Record form to the customer.
   4. The user shall fill the form.
      1. The user shall enter student id, course prefix, course number, grade, year and semester.
      2. The user shall click “Submit” button after entering the data.
      3. The system shall check if any of the data is missing.
         1. If any of the data is missing, the system shall display a message to the user and repeat Step 1.4.
         2. If all the data are correct, the system shall continue.
   5. The system shall verify if the record is already in the database.
      1. The system shall try to retrieve a record with the data except for “Grade.”
         1. If a record is retrieved, the system shall display an error message to the user, and check if the user wishes to update the record.
            1. If the user wishes to update the record, the existing record will be updated. (Refer to R2)
            2. If the user wishes not to go further, the system shall display a message to the user and return to main menu.
         2. If no record is retrieved, the system shall continue.
   6. The system shall add the new record to the database.
   7. The system shall recalculate the overall GPA for the student.
   8. The system shall update the overall GPA in the database for the student.
   9. The system shall display a message and return to main menu.
2. The system shall allow a user to edit a Student Grade record in the database.
   1. The system shall display a main menu to the user.
   2. The user shall select “Search Existing Grades”
   3. The system shall display a form to query the grade records.
      1. The form shall contain the following optional fields (NR3):
         1. The form shall contain the year.
         2. The form shall contain the semester.
         3. The form shall contain the class code.
         4. The form shall contain the student name.
         5. The form shall contain the grade.
   4. The system shall run a query on the database based off of the fields provided in the form.
      1. If the query returns 30 or less results, the system shall display the records in a list for the user.
      2. If the query returns greater than 30 results, the system shall display a message stating that the search needs to be more specific.
         1. The system shall return to 2.3, retaining any fields that were entered previously.
   5. The user shall select the entry that they wish to edit.
   6. The system shall display the following options:
      1. The system shall display an “Edit” Button.
      2. The system shall display a “Delete” Button.
   7. The user shall select the “Edit” Button.
   8. The system shall display an Edit Grade Record form.
      1. The system shall prepopulate each field for student id, course prefix, course number, grade, year, and semester.
      2. The user shall edit any necessary fields for their correction.
      3. The user shall click “Submit” button after changing their entry.
      4. The system. shall check if any of the data is missing.
         1. If any of the data is missing, the system shall display an error message to the user and return to 2.7.2.
         2. If all the data is correct, the system shall continue to the next step.
   9. The system shall recalculate the overall GPA for the student.
   10. The system shall update the overall GPA in the database for the student.
   11. The system shall display a success message and return to the main menu.
3. The system shall allow a user to delete a Student Grade from the database.
   1. The system shall display a main menu to the user.
   2. The user shall select “Search Existing Grades”
   3. The system shall display a form to query the grade records.
      1. The form shall contain the following optional fields (NR3):
         1. The form shall contain the year.
         2. The form shall contain the semester.
         3. The form shall contain the class code.
         4. The form shall contain the student name.
         5. The form shall contain the grade.
   4. The system shall run a query on the database based off of the fields provided in the form.
      1. If the query returns 30 or less results, the system shall display the records in a list for the user.
      2. If the query returns greater than 30 results, the system shall display a message stating that the search needs to be more specific.
         1. The system shall return to 3.3, retaining any fields that were entered previously.
   5. The user shall select the entry that they wish to delete.
   6. The system shall display the following options:
      1. The system shall display an “Edit” Button.
      2. The system shall display a “Delete” Button.
   7. The user shall select the “Delete” Button.
   8. The system shall display a message asking for confirmation and warning that the delete is permanent.
   9. The system shall delete the record from the database.
      1. If there is any error in the deletion or connection to the database, the system shall display an error message.
      2. The user shall close the error message.
      3. The system shall return to 3.5.
      4. If there was no error, the system shall continue to 3.9.
   10. The system shall recalculate the overall GPA for the student.
   11. The system shall update the overall GPA in the database for the student.
   12. The system shall display a success message and return to the main menu.
4. The system shall allow a user to print a Student Grade Record from the database.
   1. The system shall display a main menu to the user.
   2. The user shall select “Print Student Grade Record” from the main menu.
   3. The system shall display an input for the ID of the student.
   4. The system shall query the student by ID and all associated courses.
   5. The system shall display the values in the following format:
      1. Student Name, ID, and Overall GPA on the first line
      2. Each additional line shall contain 1 course with the course prefix, number, year, semester, and grade received.
      3. The system shall display a confirmation prompt to verify the information.
         1. If the user confirms, the information shall be printed.
            1. Once printing is complete, the system shall continue to 4.7.
         2. If the user selects “Go Back” the system shall return to 4.3
   6. The system shall display a success message.
   7. The user shall close the message.
   8. The system shall return to the main menu.
5. The system shall be able to parse excel files in the location and formats described in NR1 and NR2 to create new database entries.
   1. The system shall display a main menu to the user.
   2. The user shall select “Excel Import” from the main menu.
   3. The system shall register all folders that follow the format given in NR2 in the column of a 2-dimensional list of strings.
      1. The system shall register each file within a given folder that follows the format given in NR1 in the row of a 2-dimensional list of strings, where column corresponds to the correct parent folder
   4. The system shall query the database for a list of Year/Semester entries i.e. Fall 2019, Spring 2020 etc.
      1. For each Year/Semester combination that is found, the system shall check if that Year/Semester combination is found in the folder structure that was ingested into the column of the 2-dimensional array from 5.3.
         1. If the Year/Semester combination matches a column in the 2-dimensional list, the system shall query the database for a list of courses for that given Year/Semester combination.
            1. The system shall compare each returned course in the list against the row in the 2-dimensional string list that corresponds to the Year/Semester combination.
            2. If the name of a course returned from the database matches the name of a course found in the row, the entry shall be removed from the list as it has been parsed previously.

If there are no more courses remaining for the given year, the Year/Semester will be removed from the 2-dimensional list.

* + - * 1. If the name of a course returned from the database does not match any entries found in the row of the 2-dimensional list, it shall remain as it will need to be ingested by the system and added into the database.
      1. If the Year/Semester combination does not match a column in the 2-dimensional list, the system will continue to the next Year/Semester combination since none of the courses contained will have been ingested by the system yet.
    1. For each Year/Semester combination remaining in the 2-dimensional list, the software will create an entry for the Year/Semester combination in the database and continue to 5.4.2.1.
       1. The system shall read in the data contained in the excel sheet for the current course.
       2. The system shall parse each column, where each row is a unique entry.
       3. For each row, the system shall create a record in the database.
          1. If there are any errors in the table insertions, the system shall display an error message to the user that the excel import has failed.
  1. Once complete, the system shall display a message to the user stating that the import was successful, along with the number of courses imported and total number of student grades added to the database.
  2. The system shall return to the main menu.

**Non-functional Requirements:**

1. The system shall only accept excel files in the format “[Course Prefix][Number][Year][Semester]”.
   1. The columns of the excel file shall be the following:
      1. Name
      2. ID
      3. Grade
2. The system shall only check for excel files in folders that have the following naming format “Grades [Year][Semester]”.
3. The system shall display a main menu with the following options:
   1. Excel Import
   2. Add a Record
   3. Search Existing Grades
   4. Print Student Grade Report
4. The system shall keep all fields optional when searching existing records. This is to allow users to pull multiple records in case of a large number of edits or deletions being required.
   1. If only one entry needs to change, the user is assumed to be able to fill out all fields to pull exactly 1 record.
   2. The system shall limit returns to the user to 30 entries in the event that an entire class needs be edited or deleted. 30 entries are the assumed maximum class size.