**Detailed Design**

This detailed design will include the logical back-end organization of “The Force” thus far.

Subsystems

User management subsystem

. 1. **Detailed design diagram**

**Enrollment**

Student\_idStudent: INT

Sections\_Section: VARCHAR

Sections\_course\_Master\_list\_id : VARCHAR

**Student**

idStudent: INT

FirstName: VARCHAR

LastName: VARCHAR

PermanentCode:VARCHAR

Password: VARCHAR

Email: VARCHAR

1 1..\*

1

1

1  
 1..\*

**Transcripts**

Grade: VARCHAR

Completed: BOOLEAN

Semester: VARCHAR

**Addresses**

idAddresses: INT

address: VARCHAR

line2: VARCHAR

city: VARCHAR

province: VARCHAR

country: VARCHAR

zip\_code: VARCHAR

mailing: BOOLEAN

home: BOOLEAN

This subsystem includes all student information. It allows for the creation of an account on “The Force”. The student may first sign up, and then subsequently sign in with their credentials. Furthermore, the subsystem stores relevant information regarding their transcript and their student record. This allows the subsystem to communicate clearly with the course registration subsystem. The Student class will be references by “Enrollment” in the database, which in turn will be referenced by the “transcripts” class. An instance of the “transcript” class refers to a course the student has taken. Any superfluous information about the student that is not directly needed to enroll in a course is stored in a separate class called “Addresses” which references the “Student” class. The address table is included in the design for two reasons. The first reason is so the user can maintain his/her current address through this portal (stretch goal of the system). The second was to account for this system’s storage to draw information for other systems the school could use. This separation allows for easier manipulation of a user. The “enrollment” class is what will allow the user management subsystem to communicate with the course registration subsystem.

2. **Unit description**

**Class Student**

* idStudent (INT): stores the student id number
* FirstName (VARCHAR, String): stores the student’s first name
* LastName (VARCHAR, String): stores the student’s last name
* PermanentCode (VARCHAR, String) : stores the student’s permanent code
* Password (VARCHAR, String) : stores the student’s password for authentification purposes.

All above information is what is vital to the student’s ability to registering for courses and singing in.

**Class Addresses**

* idAddresses (INT): Because a student may possess many addresses, idAddresses stores an id number which may help refer to the same student without confusing the address instances.
* address (VARCHAR, String) : stores the student’s address
* line 2 (VARCHAR, String) : stores the student’s alternate address
* city (VARCHAR, String) : stores the student’s city
* province (VARCHAR, String) : stores the student’s province
* country (VARCHAR, String) : stores the student’s country
* zip\_code (VARCHAR, String) : stores the student’s zip code (postal code)
* mailing (BOOLEAN) : determines whether or not the given address is the mailing address to send any important documentation.
* home (BOOLEAN) : stores whether or not the address is a home address or not

**Class Enrollment**

* Student\_idStudent (INT) : stores a student’s id number. This allows the instance of “enrollment” to associate with a specific student.
* Sections\_section (VARCHAR, String) : stores the section the student is enrolled in.
* Sections\_course\_Master\_list\_id (VARCHAR, String) : stores the id number of the course the student is enrolled in from the database. This allows for the program to determine which courses have been completed or not.

**Class Transcripts**

* Grade (VARCHAR, String): stores the student’s grade in the given course
* Completed (BOOLEAN) : stores whether or not the student successfully completed the course or not.
* Semester (VARCAHR, String): stores the semester in which the student took the course.

Course subsystem

1. **Detailed Design Diagram**
2. 1..\*

**Section**

Section : VARCHAR

Semester : VARCHAR

Course\_Master\_List\_id : VARCHAR

**Course**

(course\_Master\_List)

Id : INT

Course\_code : VARCHAR

Number : INT

Description VARCHAR

Credits : INTEGER

Suggested\_Semester : INTEGER

**Prerequisites**

MainCourseID : VARCHAR

PrereqCoruseID : VARCHAR

1

1

1

**Enrollment**

Student\_idStudent: INT

Sections\_Section: VARCHAR

Sections\_course\_Master\_list\_id : VARCHAR

\*

\*

1..\*

**Timeslot**

Id : INT

Sections\_Section : VARCHAR

Sections\_Course\_Master\_List\_id : VARCHAR

Start : time

End : time

DOW : CHAR

The course subsystem contains all information regarding course numbers, IDs, and times. Having an entire subsystem where this information can be stored allows for a clean separation of tasks between the two subsystems. Although a student will enroll to a course from their point of view, the system sees it as an enrollment in a section. This is because different sections have different timeslot instances. This allows the system to simultaneously enroll the student while checking for time conflicts. It is important to note that any information regarding the course name and number starts at the course class. From there, you can even find the suggested semester. As you go further away from the course class, there is a “breadcrumb trail” that can always lead you back to it easily. Class Section contains an ID that matches to a class in the master list. Enrollment and timeslot classes have the same course name and number, although we append “Section\_” at the front, to force the path through the section class, without accidentally accessing every single section in the course. On top of this information, they contain the section ID in order to not pass through and get enrolled into the wrong section.

1. **Unit Descriptions**

**Class Enrollment**

* See “User management subsystem”
* This class serves as a “UI” between the two subsystems
* Sections\_Course\_Master\_List\_id (VARCHAR, String) : Refers to the id of a course. This id is first matched to a section along with Sections\_Section (VARCHAR, String) to match to a section the student is enrolled in.

**Class Section**

* Section (VARCHAR, String) : Stores the section number of a course the student is enrolled in.
* Semester (VARCHAR, String) : Stores the semester of the course the student is enrolled in.
* Course\_Master\_list\_id (VARCHAR, String) : Stores the ID of the course the student is enrolled in. This is matched to a specific ID in the database (Course class).

**Class Course**

* Takes the form of Course\_Master\_list in the database. Contains all information regarding courses.
* Id (INT) : Stores the ID number of a course.
* Course\_Code (VARCHAR, String) : Stores the course code in a string (4 letters)
* Number (INT) : Stores the course number (3 digits).
* Description (VARCHAR, String) : Stores the description of a course so that the student may know what they are chosing.
* Credits (INTEGER) : Stores the number of credits the course provides.
* Suggested\_Semester (INTEGER) : Stores the suggested semester of the year for the student (between 1 and 3). This allows to provide the student with a recommended course sequence.

**Class Prerequisites**

* MainCourseID (VARCHAR, String) : Stores the course name and number of the course the student is trying to enroll in.
* PrereqCourseID (VARCHAR, String): Stores the course name(s) and number(s) of the course(s) the student should have previously completed in order to enroll to “MainCourse”.

**Class Timeslot**

* ID (INT) : contains the student number of the student taking a course in that given timeslot.
* Sections\_Section (VARCHAR, String) : Stores the section number of the course of the timeslot the student is enrolled in.
* Sections\_Course\_Master\_List\_id (VARCHAR, String) : Stores the course name and number of the section of the timeslot the student is enrolled in.
* Start (Time) : the starting time of the timeslot.
* End (Time) : The ending time of the timeslot.
* DOW (CHAR): Day of the Week represented by a single character.