Milestone 1: The Semantic-Conceptual Model

COLLEGE DEPARTMENT DATABASE MANAGEMENT SYSTEM

Name: Denzill Loe

SFSU ID: 920606619

GitHub Username: Denzill7

Milestone / Version	Date
M1V1	03/15/2022

Table of Contents

Section 1: Project Description

This project is about creating a college department database management system. Its purpose is to correctly handle the data of the college departments, the faculty members and their courses of instruction, the students' names, school ID's, classes, and much more. Students will likely have classes that their name will keep a place in, and the database needs to correctly handle properly assigning these values as efficiently as possible to ensure a clean and efficient database.

This project helps to ensure better organization of all the responsibilities of all the faculty within the department and hence the department can focus on other concerns than managing the large volume of data involved with students adding or dropping classes. This keeps things on track and makes the college department much more efficient as a result.

The College Department Database Management System takes into consideration the student's satisfaction with the department. The database will keep track of classes that still need spots to fill, and it will keep track of any complaints filed by students in order to improve the available classes. Whenever there is a complaint or a notable lack of classes for a certain course, the database management system will help the faculty know to increase the amount of classes for that particular course.

Section 2: Use Cases

1. Use Case: Classes Not Being Filled

Actor: Department Chair (Carmen), Course, Class, Student

Description: Carmen is a chairperson of a computer science and engineering department at a university. She is upset because she is unable to properly enroll students into classes, resulting in some classes being only partially filled. She is now looking for a solution to better organize the remaining classes left by prominently displaying classes with low student numbers.

The College Department Database System is there to help Carmen by keeping track of the classes within the department and keeping track of the amount of students enrolled. The database system will be able to assess which classes are in need of filling.

2. Use Case: Lack of Classes for a Course

Actor: Instructor (Jose), Course, Class, Student (Denzill)

Description: Denzill, a student at a university, is trying to enroll in classes within his department in the university for the upcoming semester. Sometimes when trying to enroll in a certain class, the sections fill up so quickly that it’s impossible for a student to enroll in a timely manner. Students then struggle to obtain add codes from professors, causing untold stress on many students within the department.

The College Department Database Management System will help students search for all available sections for a given course and show them the availability of each class. If there are no classes that the student is able to enroll in, the database should give the student resources on how to contact a professor and get an add code for a given section. The database will make note of what courses students had difficulties enrolling in to properly evaluate the amount of sections needed for next semester.

3. Use Case: Students Adding Classes Late

Actor: Student, Instructor, Class, Course, Add Code

Description: Related to the above situation, students may find themselves unable to enroll in sections of a course before they fill up completely. The standard protocol is usually to find a way to attend the first class and ask the instructor for an add code to enroll in the section after the school semester has started.

If a student requests an add code from an instructor and they permit it, they can ask the database to give them a unique code for the student to submit and properly enroll their student account in the class.

4. Use Case: Advertising Events and Open Classes to Students

Actor: Department Ambassador, Students, Fliers

Description: The department wants to promote open classes to students within the university's department. However, the department struggles to find the best way to reach out to students during the busy school semester.

The College Database Management System can help the faculty by automatically sending out promotional events to the emails of members of the department.

5. Use Case: Promoting the Department to the University and Other Places

Actor: Chair (Carmen), Office Manager, Faculty, Social Media Account, Brochure

Description: Carmen runs a department at a university. She wants to give the department more exposure within the university and to outside students looking to pursue a college education. She has an office manager of the department operate a few social media accounts and set up flyers advertising events within the department to invoke outside interest. Furthermore, the ambassador prints out a bunch of brochures to hand out to the faculty of the department in case people want to learn more about the department.

Section 3: Database Requirements (Business Rules)

1. Student

A student shall be able to create at most one account.

A student shall have at least one major.

A student shall have at least one transcript.

A student shall be able to enroll in many classes.

A student shall have at most one account.

2. Instructor

An instructor is a faculty member.

An instructor shall be able to generate many add codes.

3. Faculty Member

A faculty member is an instructor, administrator, researcher, office manager, chancellor.

4. Course

A course shall have a course number.

A course shall have at least one class section.

5. Section (Class)

A class shall have many students

A class section can have only one add code.

A class section can be permitted to add with an add code.

6. Account

An account shall belong to one student.

An account shall have at least one payment method.

7. Payment Method

A payment method can be linked to many accounts.

A payment method can have many credit cards linked to it.

8. Credit Card

A credit card is linked to a payment method.

9. Major

A major shall be had by many students.

A major shall have many courses.

10. Department

A department shall have at least one faculty member.

A department shall have at least one department ambassador.

A department shall have at least one major.

A department shall have at least one department chair.

11. University

A university shall have at least one department.

A university shall have at least one address.

A university shall have a school name.

12. Transcript

A transcript can have many courses.

Many transcripts can belong to one student.

13. Administrator

An administrator is a faculty member.

14. Researcher

A researcher is a faculty member.

15. Office Manager

An office manager is a faculty member.

16. Add Code

An add code can only belong to one class.

An add code permits adding a class section

17. Chancellor

A chancellor is an instructor.

Section 4: Detailed List of Main Entities, Attributes, and Keys

1. Student (Strong)
    - school_id: key, numeric
    - name: composite, alphanumeric
        1. first
        2. last
    - dob: multivalue, timestamp
    - age: derived, numeric
2. Instructor (Strong):
    - name: composite, alphanumeric
    - num_courses: numeric
3. Faculty Member (Strong):
    - faculty_id: key, numeric
    - salary, alphanumeric
    - name: composite, alphanumeric
        1. first
        2. last
4. Course (Strong):
    - title: alphanumeric
    - prerequisites: multivalue, alphanumeric
    - units: numeric
5. Class Section (Strong):
    - section_id: key, alphanumeric
    - num_add_codes: numeric
    - num_students: numeric
6. Department (Strong)

- name: alphanumeric
-

7. University (Strong):
    - school_name: key, alphanumeric
    - address: composite, alphanumeric
    - founding_year: numeric
    - num_students: numeric
8. Account (Strong):
    - account_id: key, numeric
    - name: composite, alphanumeric
    - email: alphanumeric
9. Add Code (Strong):

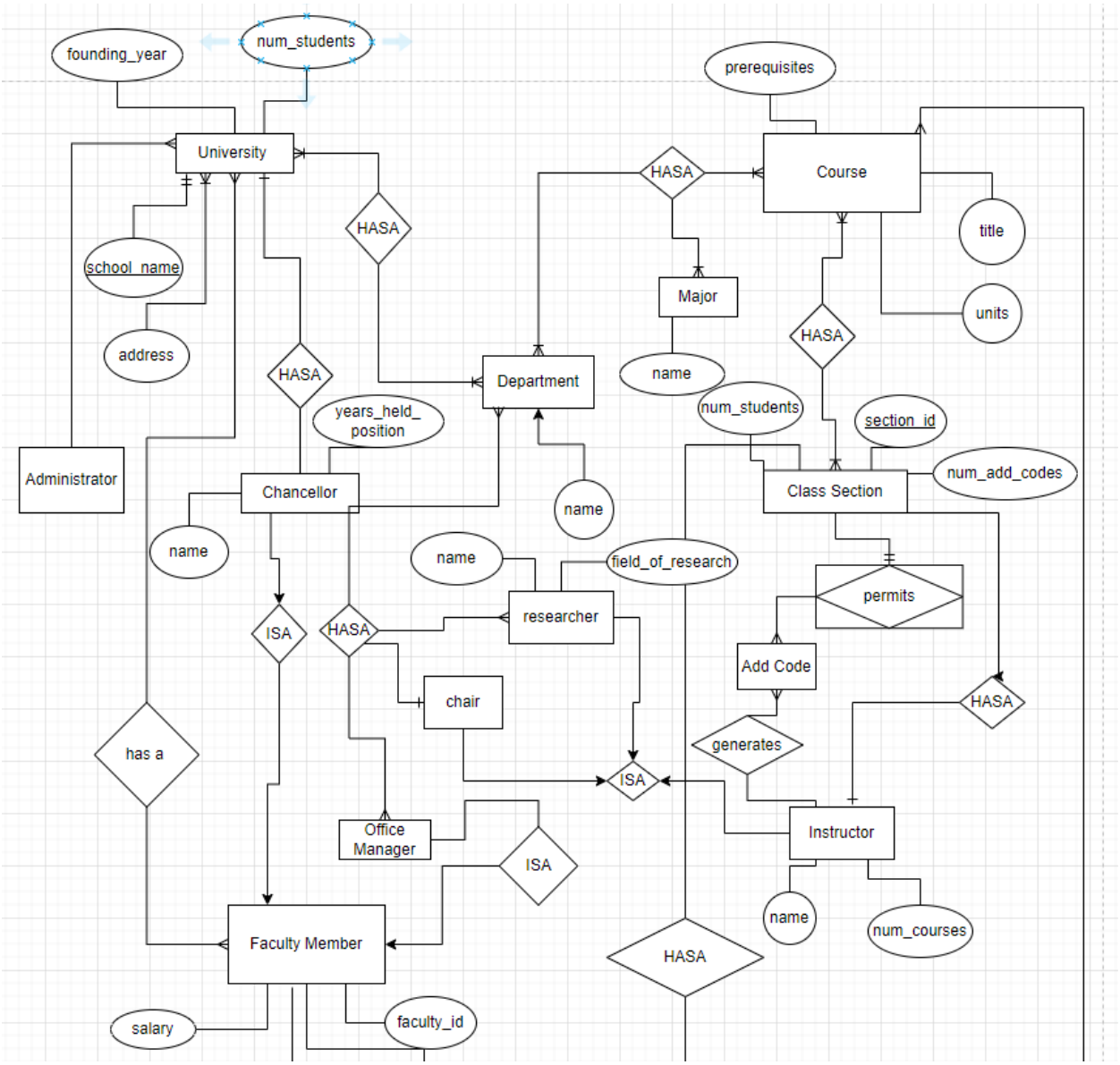
- code_id: key, numeric
- expiration_date: multivalue, timestamp

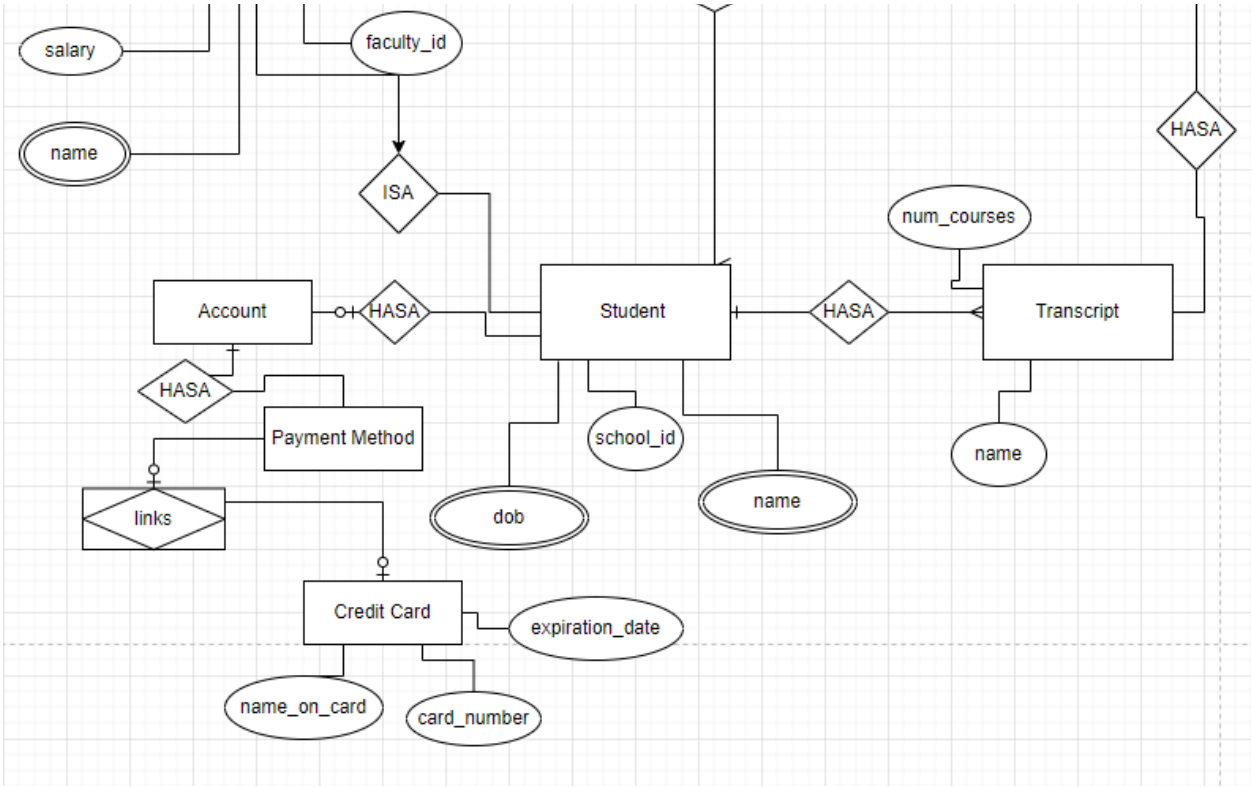
10. Researcher (Strong):
    - name: composite, alphanumeric
    - field_of_research: alphanumeric
11. Administrator (Strong):
    - name: composite, alphanumeric
12. Transcript (Strong):
    - name: composite, alphanumeric
    - numCourses: numeric
13. Major (Strong):
    - name: alphanumeric
14. Payment Method (Strong):
15. Credit Card (Strong):

- name_on_card: composite, alphanumeric
    1. first name
    2. last name
- card_number: key, numeric
- expiration_date: multivalue, timestamp

16. Chancellor (Strong):
    - name: composite, alphanumeric
        1. first name
        2. last name
    - years_held_position: numeric

Section 5: Entity Relationship Diagram (ERD)




salary
faculty_id
name
ISA
HASA
num_courses
Account
HASA
Student
HASA
Transcript
HASA
Payment Method
school_id
name
links
dob
name
Credit Card
expiration_date
name_on_card
card_number

Section 6: Testing Table

Rule	Entity A	Relation	Entity B	Cardinality	Pass/Fail	Error Description
1	Student	create	Account	0-to-1	Fail	Student could create multiple accounts
2	Student	has	Major	M-to-1	Pass	
3	Student	has	Transcript	1-to-1	Pass	
4	Student	enrolls	Class	1-to-M	Pass	
5	Student	has	Account	1-to-1	Pass	
6	Instructor	generate	Add Code	1-to-M	Pass	
7	Course	has	Section	1-to-M	Pass	
8	Class Section	has	Student	1-to-M	Pass	
9	Account	has	Payment Method	1-to-1	Pass	
10	Payment Method	linked	Account	M-to-1	Pass	
11	Credit Card	linked	Payment Method	1-to-1	Pass	
12	Major	has	Student	1-to-M	Pass	
13	Major	has	Course	1-to-M	Pass	
14	Departmen t	has	Faculty Member	1-to-M	Pass	
15	Departmen t	has	Major	1-to-M	Pass	
16	University	has	Departmen t	1-to-M	Pass	