Classic Education Center - Course Enrollment System

ISOM 3260 (Team 205)

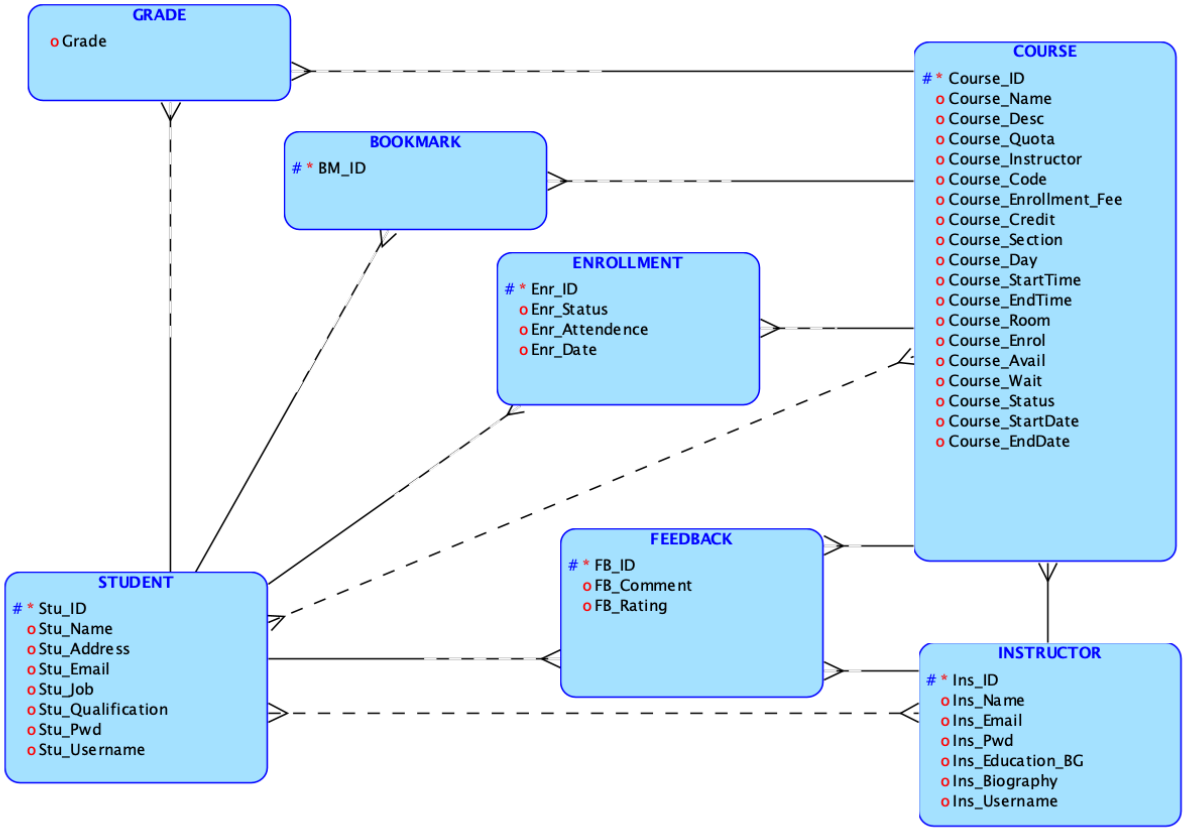
CHEN, Hsuan-ching (Cathy) 20825951
WONG, Wing Sum (Susanna) 20769937
AUVE, Karine 20755687
CHUNG, Ho Man (Alex)
MAK, Ming Hei (Ming) 20762563

1. Introduction to the Project [Karine]

This essay elaborates how Classic Education Center, a company that offers working adults professional training services, develops a system for enrolling students in courses. As IT consultants, we built a system revolving around SQL statements that retrieve information from the database. The requirement definitions, including the data and process requirements; data models and dictionary; and the system's current limitations, are the main topics of the essay.

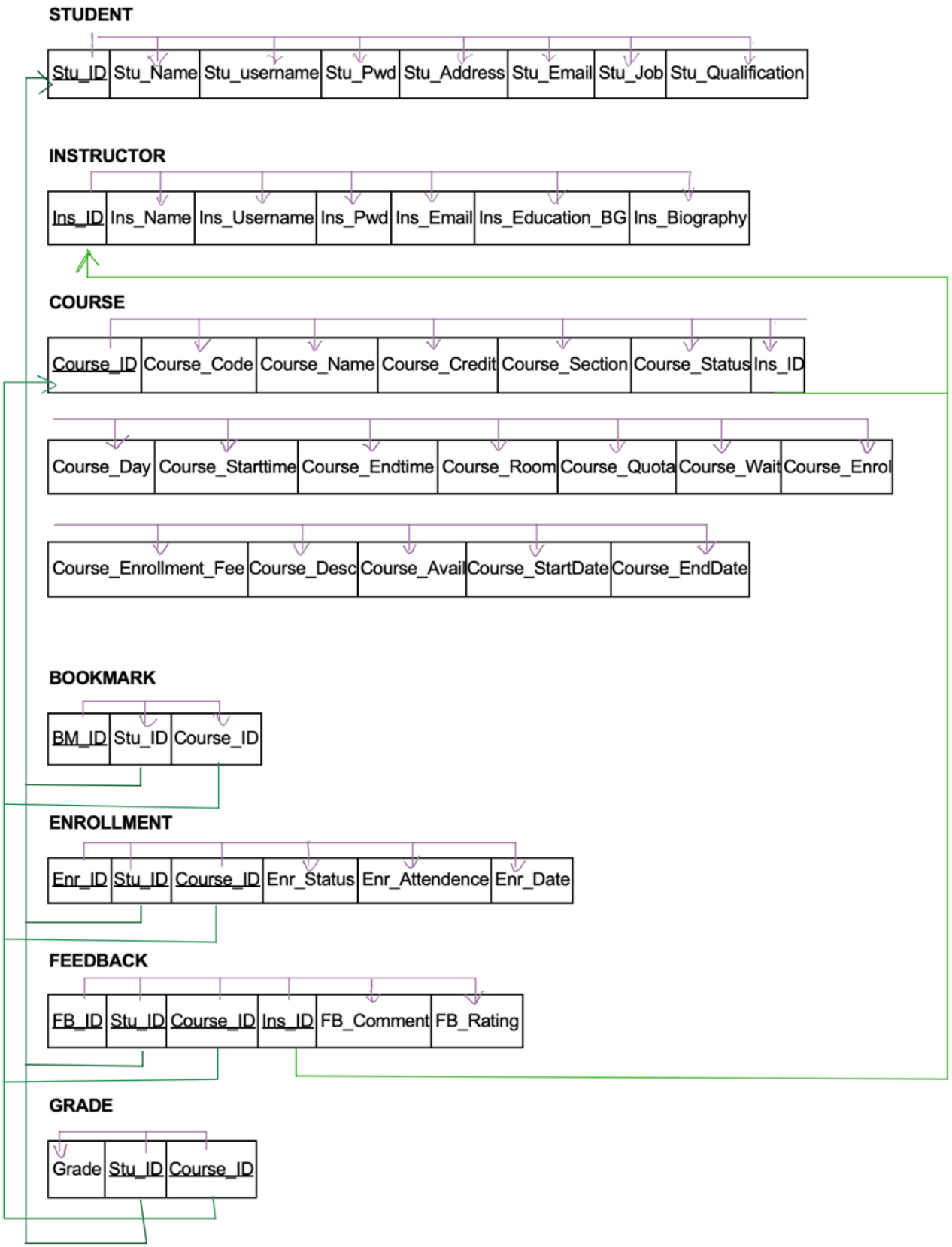
The page displayed first is the login page for instructors and students; there is also a register button. Only after a successful login, can students go to the “Student Home” page, edit their personal information, explore available courses, look up courses using keywords, check course descriptions, bookmark courses, register for courses, examine their enrollment history, and rate and comment on courses after they have passed them. For the “Instructor Home” page, the course rosters containing their student’s information, student feedback, and course ratings are all accessible for instructors to view. Managers can access a Manager Dashboard that generates reports and perform data analysis for monthly and daily income, enrollment, and instructor/course data summaries; as well as keep student/instructor/course details, check real-time course feedback response rates, create class rosters, and maintain refund lists for canceled courses. If the login was unsuccessful in the beginning, students and instructors can click forget password.

2. Conceptual Data Model [Karine]



There are three strong entities: STUDENT, INSTRUCTOR, and COURSE. The four associative entities are GRADE, BOOKMARK, ENROLLMENT, and FEEDBACK. We are assuming that each student can optionally make feedback, enrollments, bookmark and receive grades, and vice versa for each course. The reason for it being optional is that certain student may have just recently registered to this system without engaging with the courses yet; while some course's start date has not begun yet. An instance of these four associative entities must correspond to one course and one student. Feedback is the only associative entity that also connects with instructors, and instructors can receive numerous of them, but each feedback is for one instructor only. There is not a manager database as we are assuming that there is only one single manager for the course enrollment system. Students can enroll in many courses, while courses may have many students (zero if the course has not begun enrollment yet); instructors can have many students and must teach at least one course; and all courses have one instructor.

3. Logical Data Model [Karine & Susanna]



4. Data dictionary [Susanna]

You should describe all tables including their attributes, primary keys, and foreign keys.

Entity: STUDENT							
Attributes	**Datatype**	**PK**	**FK**	**Length**	**NULL**	**Description**	**Constraint**
Stu_ID	Number	X		8	FALSE	Store student's ID, an unique attribute	
Stu_Name	Varchar2			50	FALSE	Store student's Name	
Stu_username	Varchar2			20	FALSE	Store student's Username	
Stu_Pwd	Varchar2			20	FALSE	Store student's Password	
Stu_Address	Varchar2			50	FALSE	Store student's Address	
Stu_Email	Varchar2			20	FALSE	Store student's Email	LIKE '%@%'
Stu_Job	Varchar2			50		Store student's Job	
Stu_Qualification	Varchar2			50		Store student's Qualification	

Entity: ENROLLMENT							
Attributes	**Datatype**	**PK**	**FK**	**Length**	**NULL**	**Description**	**Constraint**
Enr_ID	Number	X		8	FALSE	Store enrollment ID, an unique attribute	

Stu_ID	Number		X	8	FALSE	Store student ID, an unique attribute	
Course_ID	Number		X	8	FALSE	Store course ID, an unique attribute	
Enr_Status	Varchar2			20	FALSE	Store enrollment status	IN ('enrolled', 'waitlist', 'dropped', 'waiting for refund', 'refunded')
Enr_Attendence	Varchar2			20	FALSE	Store enrollment attendance	
Enr_Date	Date				FALSE	Store Enrollment Date	

Entity: FEEDBACK						
Attributes	**Datatype**	**PK**	**FK**	**Length**	**NULL**	**Description**
FB_ID	Number	X		8	FALSE	Store feedback ID, an unique attribute
Stu_ID	Number		X	8	FALSE	Store student ID, an unique attribute
Course_ID	Number		X	8	FALSE	Store course ID, an unique attribute
Ins_ID	Number		X	8	FALSE	Store instructor ID, an unique attribute
FB_Comment	Varchar2			100		Store feedback commnet
FB_Rating	Number			2		Store feedback rating (5-25)

Entity: COURSE							
Attributes	**Datatype**	**PK**	**FK**	**Length**	**NULL**	**Description**	**Constraint**
Course_ID	Number	X		8	FALSE	Store course ID, an unique attribute	
Ins_ID	Number		X	8	FALSE	Store instructor ID, an unique attribute	
Course_Code	Varchar2			10	FALSE	Store course code	
Course_Name	Varchar2			30	FALSE	Store course name	
Course_Credit	Number			2	FALSE	Store course credit	>= 0
Course_Section	Varchar2			20		Store course section	
Course_Day	Varchar2			20		Store course day	
Course_Starttime	Varchar2			20		Store course starting time	LIKE '__:___M'
Course_Endtime	Varchar2			20		Store course ending time	LIKE '__:___M'
Course_Room	Varchar2			20		Store course instructor	
Course_Instructor	Varchar2			20		Store course instructor	
Course_Quota	Number			3		Store course quota	

Course_Enrol	Number			3		Store course enrollment number	
Course_Avail	Number			3		Store course available place(s)	= Course_Quota - Course_Enroll
Course_Wait	Number			3		Store course waitlist number	
Course_Desc	Varchar2			100		Store course description	
Course_Enrollment_Fee	Number			5		Store course enrollment fee	
Course_Startdate	Date					Store course starting date	
Course_Enddate	Date					Store course ending date	
Course_Status	Varchar2			10		Store course status	IN ('open', 'closed', 'cancelled')

Entity: INSTRUCTOR							
Attributes	**Datatype**	**PK**	**FK**	**Length**	**NULL**	**Description**	**Constraint**
Ins_ID	Number	X		8	FALSE	Store instructor's ID, an unique attribute	
Ins_Name	Varchar2			50	FALSE	Store instructor's name	
Ins_Username	Varchar2			20	FALSE	Store instructor's username	

Ins_Pwd	Varchar2			20	FALSE	Store instructor's password	
Ins_Email	Varchar2			20	FALSE	Store instructor's email	LIKE '%@%'
Ins_Education_BG	Varchar2			50	FALSE	Store instructor's education background	
Ins_Biography	Varchar2			100	FALSE	Store instructor's biography	

Entity: GRADE						
Attributes	**Datatype**	**PK**	**FK**	**Length**	**NULL**	**Description**
Grade	Number			2	FALSE	Store grade
Stu_ID	Number		X	8	FALSE	Store student ID, an unique attribute
Course_ID	Number		X	8	FALSE	Store course ID, an unique attribute

Entity: BOOKMARK						
Attributes	**Datatype**	**PK**	**FK**	**Length**	**NULL**	**Description**
BM_ID	Number	X		8	FALSE	Store bookmark ID, an unique attribute
Stu_ID	Number		X	8	FALSE	Store student ID, an unique attribute
Course_ID	Number		X	8	FALSE	Store course ID, an unique attribute

5. Functional Requirements [Everyone]

Actor	Page (L1)	Page (L2)	Buttons on page (L2)	Page(L3)	Functions
	Login	Register as student			database= STUDENT/ INSTRUCTOR/ MANAGER
	Login		Login		login()
M1		Maintain	Search		retrieve_customer() retrieve_instructor() retrieve_course() Maintain: insert, edit, delete
M2		Review_M_1	Search		retrieve_feedback()
M3		Review_M_2	Generate		retrieve_roster()
M4		Review_M_3	Search		retrieve_customer_refund()
M5		Display	Display		show_mon_rev()
					show_new_enr()
					show_topcourse()
					show_topins()
					show_num_enr()

					show_avail_course()
I1	Home_I	Review_I	View Course		retrieve_instructor_course()
I2			View roster		retrieve_roster()
I3			View Feedback and Rating		retrieve_feedback() retrieve_avg_rating()
S1	Register		Register		register_as_student()
S2	Login / Reset (forgot pw)		Login		retrieve_login_info() reset_login_info()
S3	Home_S	Account Information	(with userform) Confirm		update_personal_info()
S4	Home_S	Course	x		display_course()
			Search		search_course_kw()
S5			Add Bookmark		add_bookmark_course()
S6			Bookmark (page direction)	View/Delete Bookmark	display_bookmark() delete_bookmark()
S7	Home_S	Enrollment cart	Enroll		enroll_course()
S8			View Enrollment		display_enrolled_course()
S9			Review course	Give rating/ comment	retrieve_course() rate_course() comment_course()

6. Conclusion [Cathy]

We have completed the basic functional requirements. We created a multiple-layers system, and the first page shown is “Login”, which is the entrance for all users, including students, instructors, and managers. After login, we direct them to their own Homepage. Meanwhile, we store their ID, username, and password in global variables for further usage, ensuring the consistency of all post-login functions and avoiding the information safety issues of retrieving data according to users’ input.

There are mainly three points we would like to improve. Firstly, user experience relates to consistency. As we distribute our work with buttons and tabs, not the actual functionality inside, it caused the inconsistency of layout of similar functions; for example, retrieving course information in the course catalog under “Student” and under “Instructor”, adding bookmarks of course and canceling course, etc. Deciding on a common layout throughout the system and grouping the related feature together will be the first step for further improvement as it plays an important role in user experience.

Secondly, user experience relates to the complexity of the system. One of the improvements will be simplifying the UI and consolidating multiple steps into a single action. Take the enrollment cart for example, users have to click 3-4 buttons to complete the enrollment for a single course, which may cause confusion and further lead to exhaustion. Another improvement is to add more functions; for instance, the current SFQ only reflect with comment and overall rating, which is too simple for instructors and the university to have a deeper understanding of students’ experience. We may add more sub-division, such as course content, instructors and TA’s teaching, grading, etc.

Thirdly, the flexibility of the database and program. In reality, the courses may not start in the same week and last for the whole semester. Some of them may not even have lessons regularly. However, our current database does not have corresponding columns to store data, and the program does not have the ability to deal with it, which makes our program less realistic. Also, the manager dard only shows the report of a pre-defined period, which is not convenient for the

manager to further analyze. Thus, we would like to add more columns to the database to accommodate different class schedules and add more advanced filtering options to the program.

7. Assumptions [Everyone]

1. Assume a user always using the same device to access the system, and each device belongs to one user only.
2. Assume students can only enroll in one course within one login
3. Assume courses start in September and February each year.
4. Assume the enrollment table is cleared every semester
5. Assume users' full name and address will not exceed 50 characters
6. Assume enrollment status will be updated after the semester starts
7. Assume the manager has local credentials that are stored locally but not in the database
8. Assume students can only give feedback to courses they have enrolled in
9. Assume students cannot drop a course nor request a refund once enrolled and paid
10. Assume only Classic Education Center is able to cancel the lesson with refund
11. Assume the verification of validity of credit card is done by the Credit Card Clearing House
12. Assume payment will be handled by the credit card company successfully
13. Assume enrollment status will be automatically updated followed by payment

8. Work Assignment Among The Team [Susanna]

Note: everyone must have a programming role

Name	Programming	Final report
CHEN, Hsuan-ching (Cathy)	Pages Flow Design Login S1: Register S2: Forgot Password S4: Course Catalog	Conclusion

	S5: Bookmark Courses S6: Manage Bookmarks Final Debugging	
WONG, Wing Sum (Susanna)	S1: Register S3: Account Information S7: Enrollment Cart S8: View Enrollment S9: SFQ	Logical data model Data dictionary
AUVE, Karine	M5: Manager Dashboard	Introduction ER Diagram Logical data model
CHUNG, Ho Man (Alex)	M2: View Feedback M4: Refund List	
MAK, Ming Hei (Ming)	M1: Account Maintenance M1: Course Maintenance M3: Class Roster I: Account Information I1: Course I2: Class Roster I3: Review	

Layer 1: 'Login', 'Register', "Forgot Password", "Student Home", "Instructor Home", "Manager Home"

Layer 2 Student:

'Account Information', 'Course'

Layer 3 Student:

'Course' → 'Course Catalog', 'Bookmarks', 'Enrollment Cart', 'View Enrollment', 'SFQ'

Layer 2 Instructor:

'Account Information', 'Course'

Layer 3 Instructor:

'Course' → 'Course', 'Class Roster', 'Review'

Layer 2 Manager:

'Dashboard', 'Account Maintenance', 'Course', 'Refund List', 'Feedback'

Layer 3 Manager:

'Account Maintenance' → 'Update/Delete Student', 'Insert Student', 'Update/Delete Instructor',

'Insert Instructor'

'Course' → 'View Course', 'Update/Delete Course', 'Insert Course'

'Feedback' → 'View Feedback'