Functional Specifications:

Minerva Registration System

Project Members: Nicole Boshans, Mc Ali Jean Marie, Tyler Parris, Neville Ponde

Project Leader: Nicole Boshans

Last updated:10/22/19

Systems Design and Implementation

Professor Gupta

Weekly Log Sheet

Date	Time	Mc Ali Jean Marie (917)-651-4733	Nicole Boshans (516)-439-7290	Tyler Parris (516)-234-2152	Neville Ponde (347)-557-3683	Topic Discussed
9/3	3:50pm-5:20pm	P	P	P	P	Formed groups. Decided on a schedule
9/5	3:50pm-5:20pm	P	P	P	P	Proposal
9/10	2:40pm-3:40pm	A	P	P	P	Proposal/ Functional Specifications
9/10	3:50pm-5:20pm	P	P	P	P	Functional Specifications
9/12	2:40pm-3:40pm	P	P	P	A	Functional Specifications
9/12	3:50pm-5:20pm	P	P	P	P	Functional Specifications
9/17	2:40pm-3:40pm	P	P	P	A	Functional Specifications
9/19	2:40pm-3:40pm	A	P	P	A	Functional Specifications
9/24	3:50pm-5:20pm	P	P	A	P	Functional Specifications
9/26	2:40pm-3:4	P	P	P	p	Functional Specifications

	0pm					
9/26	3:50pm-5:20pm	P	P	P	P	Functional Specifications
10/1	2:40pm-3:40pm	A	P	P	A	Functional Specifications
10/1	3:50pm-5:20pm	P	P	P	P	Functional Specifications
10/3	2:40pm-3:50pm	P	P	P	A	Functional Specifications
10/3	3:50pm-5:20pm	P	A	P	P	Functional Specifications
10/8	2:40pm-3:50pm	P	P	P	P	Database
10/8	3:50pm-5:20pm	P	P	P	P	Database
10/10	2:40pm-3:50pm	A	A	P	P	ERD
10/10	3:50pm-5:20pm	P	P	P	P	ERD
10/15	3:50pm-5:20pm	P	P	P	P	
10/17	3:50pm0-5:20pm	P	P	P	P	

Note: P = Present, A = Absent

Table of Contents:

1.Functional Summary:

The system is a web-based registration system called "Minerva Registration System" which can be accessed from any web browser. This system will be used to help register for classes as well as perform a degree audit. The system should be made so that any institution can use it and is not specified for a specific institution. Due to the amount of features modern school register systems have, the main purpose will be to keep them all accessible to all types of administration, faculty, researchers, and students. In order to make it fully operational, any number of users may be signed on at the same time. The system will be targeted for student use but as stated earlier will be usable for faculty, administrators and researchers. In order to keep things simple, we will target reliability for students, faculty and researchers while targeting security for administrators. If more security is needed, we will add a timeout function where the system will log the user out after a certain amount of time has passed if the user has not taken any actions. The physical environment will be for computer desktops and laptops but a mobile version may be developed for on-the-go use later. The system can still be accessed via phone, but not through an app.

1.1 System Actors:

The **administrator** can access all of the features that the students and faculty have access to. This is because the administrator grants those features and can modify them if need be. Because the administrator has the same functions and more, they are considered the top of the hierarchy when it comes to the school register system. They will use the system occasionally as the amount of administrators are low. Being the only user with all of these functions, the administrator access is only granted to these specific users and would be dangerous to the system if any non-administrator had control over these functions. Thus, the administrator log in process

would be a two-step verification in order to make sure that only the administrator has access and no outside parties can obtain control. The administrator would also need to provide an ID number if they wish to add/drop/update a student's schedule.

The **faculty** will be able to check the classes they are teaching, check their student roster, keep track of attendance, access a student's information such as email and address, and communicate with other faculty members. The faculty will also have a function to send requests to the administrators. This will consist of opening up an email to send to administration. The faculty usage will be occasional as they will be mostly checking their roster and choosing which classes they wish to teach. Just like the student they only need to provide their username and password although an ID will be required if they wish to see a student's grade for their specific course.

Students should be able to add, drop, and search for classes.A student with six or less credit hours will be known as a part-time student while a student with six to twelve credit hours will be known as a full-time student. A student would be able to apply for a major (such as the Math program or English program). These programs would require classes to make up the criteria that a student must complete to acquire that part of their degree. They should be able to view the classes they are missing for their major(s). Students should be able to obtain and view their transcripts. A student may also have academic holds that may prevent them from registering or dropping a class. The student will have access to the system any time the system is up and running. Because the amount of students will eclipse the amount of faculty and administrators, the student user interface will be used more frequently a month before and during school is open. The access permissions for a student is their user created name and password.

A **researcher** can look at all data related to catalogue, classes and holds. This includes class credits, class times, professors, grades and amount of time it takes students to graduate. Recherches can not have access to any personal information of the systems users. That includes id numbers, emails, phone numbers, birthdates,ecta. The amount of researchers may also be low, so use may be low. The researcher only needs to provide a username and password.

In order to implement the project, we will be using JavaScript/CSS, Java, PHP and SQL. To get the project running on a website, we will be using Javascript/CSS implementation that will focus on website design, structure, and navigation through programmable elements. This will require implementing an applet in order to keep things moving as the user goes through the project. SQL will be used to develop and maintain a database for the project. The database should store information such as: logins, catalogue years, catalogue semesters, catalogue course selections , catalogue class times, catalogue classrooms, catalogue professors, catalogue prerequisites, and catalogue holds. The relational database would need to store information per institution and for each of said institutions programs and classes as well. Another datatable may need to be developed to hold student information such as degree requirements and personal information.

2. Human Interface:

We expect the users need to login in to the system. The login page will prompt the user to state whether or not they are logining in as an administrator, a faculty member, student, or researcher. This is to ensure that proper personnel are allowed certain actions.

2.1 Administration

When logining in as an administrator, the user will be able to create the class catalog. This consists of a list of all the classes able to be taken at the institution along with a description of each of the class, credit hours, lecture hours, levels (undergraduate or graduate), department, prerequisite requirement and course attributes. The user then can pick classes from the catalogue to be available during the current or next semester. For example, to add a class, it must be categorized under a specific year and time such as Fall 2019. The class being added will then have certain information that has to be available. This information being: CRN number, instructional method, subject, course number, section number, credit hours, title, days, times, section capacity, section remaining, instructor, date(that being the date of the beginning and end of the semester), course attributes and location. When adding classes, an error message may appear if two classes are in the same room at the same time, if one professor is assigned to teach two classes at the same time and if classroom occupancy can not hold class size. Also, the user can remove classes being offered. The user can place holds on student accounts. This would consist of using the student's id to place a hold on the account. The information given should be hold type, the start date of the hold, the end date of the hold, the reason for the hold and the certain processes that the hold effects. The user can then look up students information when given the students id number. Such information being first and last name, email, address, transcript, date they started at the institution, and the date they left the institution. The user can change a student's grade for a course up to next semester plus two weeks.The users will assign students an advisor and determine how many students a single advisor can have.

2.2 Faculty

When logging in as a faculty member, the user will be able to view the current classes they are teaching. This consist of a list of the classes the user is teaching as well as the class roster for each class being taught. Once the class has started, the professor will be able to input grades and attendance. The professor has to have final grades in by a week after the final class. After a week the professor can no longer change grades. A professor can look up a list of the students they are advising. Professor is part of a department, thus a professor can only advise and teach in there department. A professor who is also administration (eg. Department Chairman) assigns classes to the professors and is able to view any professors classes in the department.

2.3 Student

When logging in as a student, the user will be able to look up classes/register for classes. When looking up classes, the user has to decide which semester they want to look up. The same applies for when registering for classes, but there will be constraints of only being able to view the current registration semester(s). The user then can look up classes by subject, course number, title, instructional method, credit range, instructor, attribute type, time and days. The courses will then be listed and sectioned off by subject. When registering for a course, the same search parameters will be given. When a student registers for courses, they have to activate the check box next to the class and hit the register button. There may be error messages such as: "class added conflicts with other class", "class prerequisite not met, can not add due to too many credits" and "not able to register due to holds on account". The student can drop a course. This can be done up to two weeks after classes start. To drop a class a student will have to check the check boxes next to the classes they are currently enrolled in which they want to drop. Errors that may occur when dropping class is that the minimum credit is no longer met. The student will be

able to declare a major/minor. The student will enter information such as: whether or not this is for a dual major, is this a request for a minor, and the major/minor name. An error may occur if the student already has two majors or if the student already has a minor. The student can request a transcript. This will give the student a list of all the classes and grades they took. An error may occur if there is a hold on the account.

2.4 Researcher

When logging in as a researcher, you can view previous courses. Much like how the student can look up classes, the user has to decide which semester they want to look up and then they can search by the same specifications listed previously. Researchers have access to all current and previous semesters. Researchers in addition can view the class roster and grades. The researcher can not view the students ids when looking at the class roster. A researcher can look at when students entered the institution and when they left the institution.

Since the system is a web-based system there will be no input from a terminal. The GUI will be where we obtain our input. The GUI will restrict certain input from being entered. Mostly the GUI will consist of drop down boxes and clickables to reduce invalid input. For security reasons, we need to vet all input.

To the users, the system will allow the user to do multiple things. The user just has to input the proper information being prompted and the system will output the desired information/action the user wanted. The information given will be in a visual manner such that it is easily readable by the user. Most output the system will produce will be in a table format. Error messages will be displayed in red text stating Error:"Error message".

2.5 Explicitly Required/Forbidden

2.5.1 Visitor

Explicitly Required

- Can view catalog/master schedule
- Can view home page

Explicitly Forbidden

- No login status

2.5.2 Administrator

Explicitly Required

- Makes(Add/drop courses from the) master schedule
- Can search the master schedule
- Assigns an advisor to a student.
- Can add/drop a section.
- Can search for open rooms.
- Can remove a student from a section.
- Can view a student's schedule.
- Places/ removes holds.
- Can change a student’s major.
- Can assign major requirements
- Can view/update a student’s personal information such as address
- Can send messages to the student or faculty.

- Can access a student's transcript.

Explicitly Forbidden

- Cannot cancel a course without informing the registered students
- Cannot change ID or email

2.5.3 Faculty

Explicitly Required

- Can view class roster and their teaching schedule
- Can add and update attendance for their courses
- Can request that a student may be removed from their course if the student is believed to have not met the prerequisites score.
- Can search for open rooms.
- Can view student holds.
- Can view their class schedule.
- Can view their student advisement list
- Can view a student class standing whether they are: Freshman, Sophomore, Junior, and Seniors.
- Assign grades in a given time window
- Can view transcript

Explicitly Forbidden

- Cannot view student personal information such as: Date of Birth, Financial Aid Status, Social Security Number

- Cannot drop students from their courses themselves.
- Cannot register students in their courses.
- Cannot teach two courses at the same time/day

2.5.4 Student

Explicitly Required

- Can search the catalog and master schedule
- Can search the master schedule by department, course name/number, prof name and date and time.
- Can register for course sections.
- Can check prerequisites of a course
- Can look up advisors.
- Can check requirements for major.
- Can view transcript.
- Can view holds.
- Can do degree audit

Explicitly Forbidden

- Cannot register for a course without the proper prerequisites.
- Cannot change grades.
- Cannot remove holds.
- Cannot take two courses at the same time/day

2.5.5 Researcher

- Can view the catalog.
- Can view class information such as: course credit, professors, number of students taking/taken the class, number of empty set in the class, etc.
- Can view course grades.
- Can view how long it takes a student to graduate.
- Can view holds.
- Can view class attendance.

Explicitly Forbidden

- Cannot register for classes.
- Cannot view students personal information such as: student id, email, phone number, birthdates, ecta.
- Cannot add classes.

3. Sample Planned Interaction (Menu)

3.1 User: Administration

Displays log in page.

$Inputs jdoe@gmail.com, password1234 and selects user type Administration .$

Displays Administration home page

$Selects add classes for semester$

Display screen which includes a choose semester text and input box.

$Selects Spring 2020$

Displays screen which includes an outline of a class with input boxes.

$Inserts 1234, Lecture, CS, 4100, 001, 4.0, Artificial Intelligence, MW, 1:00pm-2:30pm, 25, 25, Park, 8/26-12/20, NA and NAB 0111$

Inserts the data to the database. Displays a confirmation of adding the class.

$Logs out$

Displays log in page.

3.2 User: Faculty

Displays log in page.

$Inputs jdoe@gmail.com, password1234 and selects user type Faculty.$

Displays Faculty home page.

$Selects enter grades.$

Obtains information from the database to give a list of all the classes Jane Doe is teaching.

$Selects Intro to Java$

Obtains information from the database to give a list of all students enrolled in Intro to Java

$Selects John Doe and enters the grade A$

Updates the database so that John Doe's transcript has the grade A for Intro to Java. Displays a confirmation of entering the grade.

$Logs out$

Displays log in page.

3.3 User: Student

Displays log in page.

$Inputs jdoe@gmail.com, password1234 and selects user type student.$

Displays Students home page

$Selects Register for classes$

Display screen which includes a choose semester text and input box.

$Selects Fall 2019$

Displays screen which includes various text and input boxes based on the different ways to look up classes. The searches are limited to Fall 2019.

$Selects Computer Science$

Obtains information from the database to give a list of all classes with the subject Computer Science.

$Selects Intro to Java$

Obtains information from the database to give a list of all days and times in which include Intro to Java.

$Selects the Intro to Java being offered at 8:00am-9:40pm on Monday and Wednesday.$

Inserts the class to the students Transcript in the database. Updates the database for Intro to Java so that the seats available is decreased by one. Displays a confirmation message stating the class was added successfully.

$Logs out$

Displays log in page.

3.4 User: Researcher

Displays log in page.

$Inputs jdoe@gmail.com, password1234 and selects user type Researcher.$

Displays Researcher home page.

$Selects view holds.$

Obtains information from the database to give a list of all holds students currently have.

$Logs out$

Displays log in page

4. Database Information

4.1 Database Business Rules

- Faculty members are a part of one department.
- A department has many faculty members.
- Students can have no more than two advisors.
- Faculty members can advise many students.
- A faculty member can be a department head(Admin) .
- A school catalog has many departments.
- A department is apart of one school catalog.
- A department can have many majors.
- A major can belong to only one department.
- A school catalog has many courses.
- A courses belongs to only one school catalog.
- A course can have many prerequisites.
- A prerequisite can be for many courses
- Faculty members can teach many sections.
- A section can be taught by only one faculty member.
- An enrolled section is apart of one transcript.
- A transcript can have many enrolled sections.

- A student can enroll for many sections
- An enrolled section can be taken once by a student.
- A building can have many rooms.
- A room belongs to only one building.
- A course can have many sections.
- A section can belong to only one course.
- A section has only one building.
- A building can hold many sections.
- A section can have one room
- A room can have many sections
- A section can have only one period.
- A period can have many sections.
- A period has only one year.
- A year can have many periods.
- A period can only have one semester.
- A semester can have many periods.
- A period can have one time.
- A time can have many periods.
- A period can have one day.
- A day can have many periods.

4.2 Entity Names and Attributes

1. User
    a. User ID (PK)
    b. First Name
    c. Last Name
    d. Email Address
    e. Phone Number
    f. Password
        i. Administrator
            1. Administrator ID (FK,PK)
            2. Department ID (FK)
            3. Office Number
            4. Title
        ii. Student
            1. Student ID (FK,PK)
            2. Student Academic Standing
            3. GPA
        iii. Faculty
            1. Faculty ID (FK,PK)
            2. Faculty Rank
            3. Department ID (FK)
            4. Office Room
            5. Faculty Office Hours
        iv. Researcher
            1. Researcher ID (PK)
            2. Research List

2. Department
    a. Department ID (PK)
    b. Department Name

c. Department Email

d. Department Location

e. Department Phone

f. Department Secretary

g. Department Chair

3. Major

a. Major ID (PK)

b. Major Name

c. Department ID (FK)

4. Major Requirements

a. Major ID (FK)

5. School Catalog

a. Viewable

b. Department ID (CK, FK)

c. Course ID (CK, FK)

d. Course Name

e. Credit Numbers

f. Prerequisites

g. Room Number

h. Description

6. Course

a. Course ID (PK)

b. Course Name

c. Course Description

d. Course Credit Amount

e. Department Name (FK)

7. Prerequisite

a. Prerequisite ID (PK,FK)

b. Course ID (PK)

8. Section

a. Section ID (PK)
b. Course ID
c. Section Number
d. Room ID
e. Faculty ID
f. Time Slot ID

9. Building

a. Building ID (PK)
b. Building Name
c. Building Address
d. Room Amount

10. Room

a. Room ID (PK)
b. Room Number
c. Room Capacity
d. Building ID (FK)
e. Room Type
    i. Lecture Room
        1. Room Number
        2. Building Name
    ii. Lab Room
        1. Room Number
        2. Building Name
    iii. Office Room
        1. Room Number
        2. Building Name

14. MasterSchedule

a. SemesterYear ID (PK)
b. Semester
c. Year

16. Enrollment

a. Student ID (FK)

b. Section ID (FK)

17. History

a. Student ID (FK)

b. Section ID (FK)

c. CourseDump

d. Semester/Year

18. Attendance

a. Attendance ID (PK)

b. Student ID

c. Section ID

d. Date

e. IsPresent

19. Holds

a. Hold ID(PK)

b. Hold Type

20. Hold Status

a. Hold ID(PK)

b. StudentID(PK/FK)

c. Hold Status

21. Undergraduate

a. Student ID(PK/FK)

b. Major Name(FK)

c. Undergrad ID(PK)

22. Graduate

a. Student ID(PK/FK)

b. Major Name (FK)

c. Grad ID(PK)

23. Undergrad Full Time

a. Credits Number

b. Credit Total

c. Undergrad ID(PK/FK)

24. Graduate Full Time

a. Credits Number

b. Credit Total

c. Grad ID(PK/FK)

25. Undergrad Part Time

a. Credits Number

b. Credit Total

c. Undergrad ID(PK/FK)

26. Graduate Part Time

a. Credits Number

b. Credit Total

c. Grad ID(PK/FK)

27. Advisors

a. Faculty ID

b. Student ID

28. Master Schedule

a. SemesterYear ID

b. Year

c. Semester

29. TimeSlot

a. TimeSlot ID

b. Day

c. Start Time

d. End Time

30. Period

a. PeriodID

31. Day

a. DayID

32. Minor

a. MinorID

b. DepartmentID

4.3 Relationship Names

- **User** is the parent of **Admin**, **Student**, **Faculty** and **Researcher**.
- **Student** has a one to many relationship with **Enrollment,** a one to many relationship with **Hold Status** and is the parent of **Undergraduate** and **Graduate**
- **Undergraduate** is the parent of **Undergrad Part Time** and **Undergrad Full Time**
- **Graduate** is the parent of **Graduate Part Time** and **Graduate Full Time**
- **Faculty has** a many to one relationship with **Department** and a one to one relationship with **Section**
- **Department** has a one to many relationship with **Faculty**, a many to one relationship with **School Catalog**, a one to many relationship with **MajorRequirements**
- **Building** has a one to many relationship with **Room** and a many to one relationship with **Section**
- **Room** has a many to one relationship with **Building** and is the parent of **OfficeRoom, LectureRoom** and **LabRoom**

- **Course** has a one to many relationship with Prerequisite, a many to one relationship with **Section**, a one to many relationship with **MajorRequirements** and a many to one relationship with **Department**
- **Prerequisite** has a many to one relationship with **Major Requirement.**
- **Major Requirement** has a one to many relationship with **Prerequisite** and a many to one relationship with **Course.**
- **Section** has a many to one relationship with **Course**, a one to one relationship with **Faculty**, a many to one relationship with **Building,** a many to one relationship with **MasterSchedule**, a one to one relationship with **TimeSlot,** and a one to many relationship with **Enrollment**
- **MasterSchedule** has a one to many relationship with **Section**
- **History** has a one to one relationship with **Enrollment**
- **Enrollment** has a many to one relationship with **Student**, a many to one relationship with **Section,** and a one to one relationship with **History**
- **Holds** has a one to one relationship with **Hold Status**
- **Hold Status** has a one to one relationship with Holds and a one to many Relationship with **Student**
- **TimeSlot** has a one to one relationship with **Section**
- **Attendance** has a many to one relationship with **Enrollment**
- **School Catalog** has a one to many relationship with **Course.**
- **Period** has a one to many relationship with **TimeSlot**
- **Day** has a one to many relationship with **TimeSlot**

- **Minor** has a many to one relationship with **Department**

4.4 Database Relation Schema

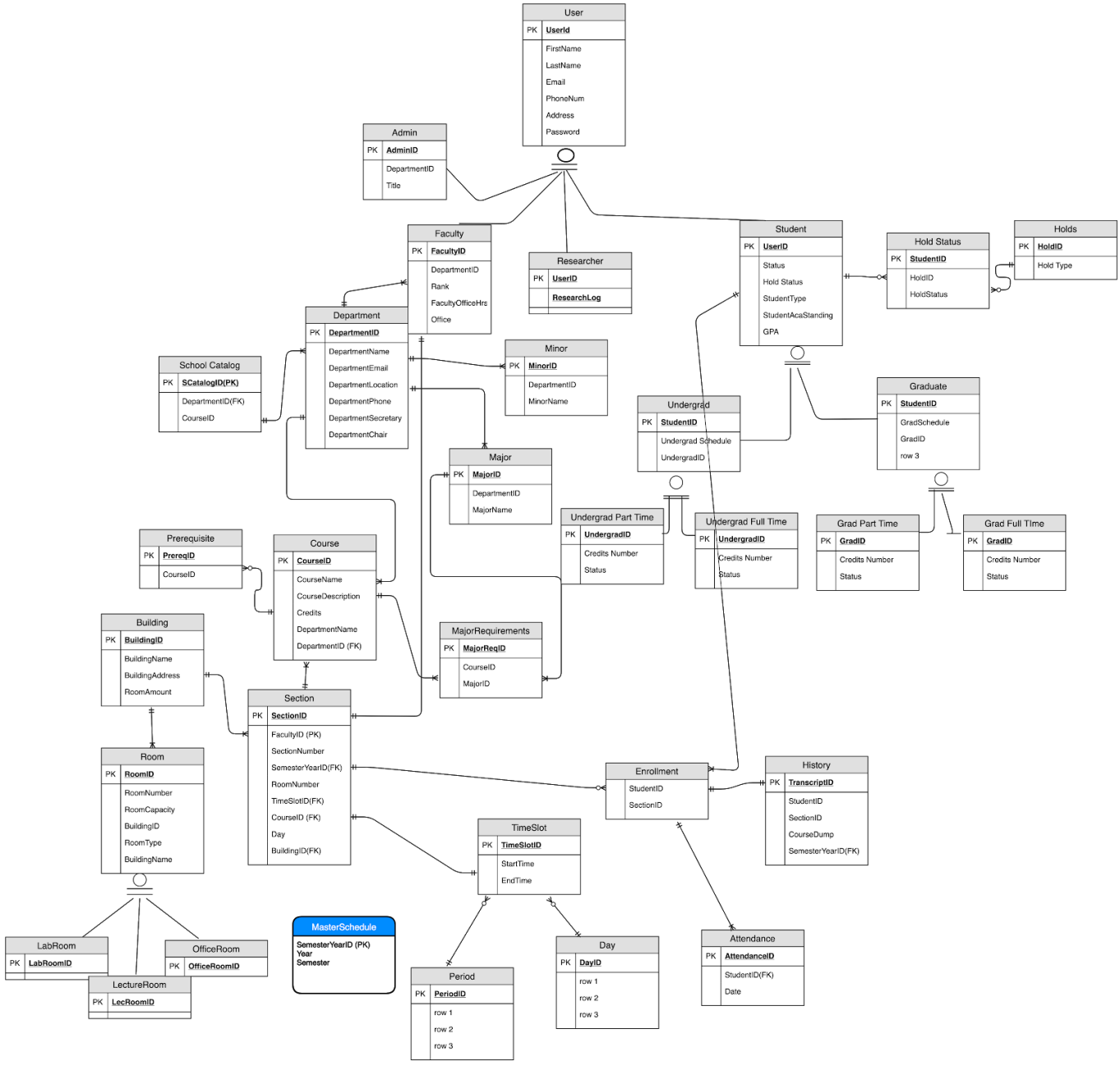
- User(**UserID (PK)**, FirstName, LastName, Email, PhoneNum, Address, Password, UserType)
- Admin(**AdminID (FK)**, OfficeNum, Title)
- Faculty(**FacultyID (FK)**, DepartmentID (FK), Rank, OfficeHrs, Office)
- Student(**StudentID(FK)**, Type, Standing, GPA)
- UnderGrad(StudentID(FK), Major Name(FK))
- UnderGradPT(StudentID(FK), SemesterYearID (FK), CreditNum, CreditTotal)
- UnderGradFT(StudentID(FK), SemesterYearID (FK), CreditNum, CreditTotal)
- Grad(StudentID(FK), Major(FK))
- GradPT(StudentID(FK), SemesterYearID (FK), GradID(PK/FK), CreditNum, CreditTotal)
- GradFT(StudentID(FK), SemesterYearID (FK), GradID(PK/FK), CreditNum, CreditTotal)
- Hold_Status(StudentID(FK), HoldID(FK), HoldStatus)
- Holds(HoldID(PK), HoldType)
- Researcher(**ResearcherID(FK),** ResearchList)
- Department(**DepartmentID(PK)**, DeptName, DeptEmail, DeptChair (FK), DeptLocation(FK), DeptPhone, DeptSecretary, DeptChair)
- Major(**MajorID(PK)**, DepartmentID(FK), MajorName)
- MajorRequirements(**MajorReqID(PK)**, MajorID(FK), CourseID(FK))

- SchoolCatalog(**DepartmentID(FK)**, **CourseID(FK)**, Viewable)
- Course(**CourseID(PK)**, CourseName, CourseDescription, CreditAmt, DeptName, DepartmentID(FK))
- Prerequisite(**CourseID**(FK)**, PrereqID(PK)**)
- Section(**SectionID(PK)**, CourseID(FK), FacultyID (FK), Room ID(FK), SectionNum, TimeSlotID (FK), SemesterYearID(FK), Day)
- Building(**BuildingID(PK)**, BuildingName, BuildingAddr, RoomAmt**)**
- Room(**RoomID(PK)**, BuildingID(FK), RoomCap, RoomNum, RoomType)
- LectureRoom(LecRoomID(FK), RoomNumber)
- LabRoom(LabRoomID(FK), RoomNumber)
- OfficeRoom(OffRoomID(FK), RoomNumber))
- MasterSchedule(**SemesterYearID(PK)**, Semester, Year)
- Enrollment(StudentID(FK), SectionID(FK))
- History(**TranscriptID(PK)**, StudentID(FK), SectionID(FK), CourseDump, SemesterYearID(FK))
- Attendance(**StudentID (FK)**, SectionID(FK), Date, IsPresent)
- Advisor(FacultyID(FK), StudentID(FK))
- TimeSlot(TimeSlotID(PK), DayID, PeriodID, StartTime, EndTime)
- Period(**PeriodID(PK))**
- Day(**DayID(PK))**
- Minor(**MinorID(PK)**, DepartmentID)

4.5 Entity Relational Diagram

https://drive.google.com/file/d/1B8U0cbIIe0CBwpc9-GISsUnnYKnTan-u/view?usp=sharing

(Use Add-ons in the toolbar to upload draw.io ERDs)



5. Diagrams

5.1 Use case diagram(s)

USE CASES TO ADD

Visitor:

UC 17	View Main Page
Primary Actor(s)	Viewer, Admin, Student, Faculty, Researcher
Trigger	Accessing the website
Pre-conditions	User is not on the website
Post-conditions	User is on the website and can see the main page information.
Main Scenario	1. User goes to minervauniversityedu.online
Extensions	User wants to view undergrad 1. User clicks on the Admissions 2. User clicks on Undergraduate User wants to view grad 3. User clicks on the Admissions 4. User clicks on Graduate User wants to view Academic Calendar 5. User clicks on the Admissions 6. User clicks on Academic Calendar User wants to view Master Schedule 7. User clicks on the Admissions 8. User clicks on Academic Calendar 9. User chooses the semester they want to view User wants to view Course Catalog 10. User clicks on the Admissions 11. User clicks on Course Catalog 12. User chooses the semester they want to view a. User can choose to view Art courses only b. User can choose to view Biology courses only c. User can choose to view Business courses only d. User can choose to view Computer Science courses only e. User can choose to view English courses only f. User can choose to view Humanities courses only g. User can choose to view Math courses only

	h. User can choose to view Medical courses only i. User can choose to view Music courses only User wants to view Departments 13. User clicks on the Departments a. User clicks on Art i. User click on Graphic Design to view the pdf file b. User clicks on Biology i. User click on Biochemistry to view the pdf file ii. User click on Biogenetics to view the pdf file c. User clicks on Business i. User click on Accounting to view the pdf file ii. User click on Business administration to view the pdf file d. User clicks on Computer Science i. User click on computer information systems to view the pdf file e. User clicks on English i. User click on English B.A to view the pdf file f. User clicks on Humanities i. User click on Humanities B.Ato view the pdf file g. User clicks on Math i. User click on Mathematics to view the pdf file h. User clicks on Medical i. User click on medicine to view the pdf file ii. User click on Psychiatry to view the pdf file i. User clicks on Music i. User click on Music to view the pdf file User wants to view president 14. User clicks on the About 15. User clicks on President User wants to view staff 16. User clicks on the About 17. User clicks on Staff User wants to view mission 18. User clicks on the About 19. User clicks on Mission User wants to view IT department 20. User clicks on the About 21. User clicks on IT department
Priority	High
Special Requirements	

UC 18	Login / Register a user
Primary Actor(s)	Admin, Student, Faculty. Researcher
Trigger	Clicking Login
Pre-condition	User is not logged in
Post-condition	User is logged in and directed to proper user homepage.
Main Scenario	1. User enters email 2. User enters password
Extensions	User Registers itself as a user 1. User clicks on the link that says " Do you not have an account? Sign up here." 2. User is directed to the sign up page 3. User enters First name 4. User enters Last name 5. User enters phone number 6. User enters address 7. User enters email(username) 8. User enters password 9. User is then redirected to login
Priority	High
Special Requirements	

Admin:

Student:

Split Add/Drop

Search Courses

Display Catalog

Check Advisor

Change Major

Display Major

Display Minor

Look up holds

Look up transcript

Degree audit

View Grades

UC 1	Add and Drop Course
Primary Actor(s)	User, Administrator
Trigger	User wants to add/drop a course to their schedule
Pre-conditions	User is registered in the school system AND no holds are on the student's account
Post-conditions	Course is successfully added to the student's schedule
Main Scenario	2. User logs-in to the system 3. System returns a successful log-in message and brings the user to the home screen 4. User selects Add/Drop/Update Course button 5. System brings the user to the Semester Year page a. User selects a year from the dropdown menu b. System brings the user to the Course Subject page c. User selects the subject the course is in d. System lists the courses that are offered that year in the subject the user has selected e. User selects the course they wish to add f. System brings the user to the course's description page along with a list of prerequisites and requirements

	g. User checks the checkbox and adds the course h. System checks if the user has no holds, has the appropriate prerequisites, has space to add the course and that there are no time conflicts. If all of these are valid, the course is added to the student's schedule 6. User selects drop course a. System brings up the user's courses b. User selects a checkbox next to the course they wish to drop c. System returns a confirmation prompt that they wish to drop the course d. User selects yes e. System returns the user's schedule without the dropped course
Extensions	If user wishes to add prerequisite as a corequisite, 1. Administrator logs-in to the system 2. System returns a successful log-in message and brings the user to the home screen 3. Admin accesses the student's schedule with the student's ID number 4. System displays the student's schedule 5. Admin selects Add/Drop/Update course button 6. System brings the user to the Course subject page 7. Admin selects the subject that the course is in 8. System lists the courses that are offered that year in the subject the user has selected 9. Admin selects the course they wish to add 10. System brings the user to the course's description page along with a list of prerequisites and requirements 11. Admin checks the checkbox and adds the course 12. System checks if the user has no holds, has the appropriate prerequisites, has space to add the course and that there are no time conflicts. If all of these are valid, the course is added to the student's schedule 13. Admin selects add/drop/update course button and goes into the class with the prerequisite 14. …. 15. System brings the user to the course's description page along with a list of prerequisites and requirements 16. Admin checks the checkbox and tries adding the course 17. System returns an error message stating that the student has not fulfilled the prerequisite and asks for the administrators ID number in order to add the course 18. Admin inserts their ID number and confirms to add the course as a corequisite

	19. Part 12 is repeated
Priority	High
Special Requirements	If the course is part of a degree requirement, student has their major listed as the subject's intended students [need to word this better later]
Exceptions	a. Class is already on the Student's Schedule i. If the class is already on the schedule, the student cannot add the course due to redundancy. This includes classes with the same section number and the same course number. b. Time conflict i. If there exists another class on the schedule with the same timeslot and day(s), the student cannot add the class. c. Error due to not having prerequisite completed i. If the student doesn't have the appropriate prerequisite, the student cannot add the course. The admin can add the prerequisite as a corequisite under the student's request. d. Error due to hold on account i. If the student has a hold on their account, they cannot add courses to their schedule
Open Questions	

UC2	Search course
Primary Actor(s)	Student, Administrator
Trigger	User wants to search for a course
Pre-conditions	User is registered in the system
Post-conditions	User successfully found the course they searched for using the parameters the wish to find it by
Main Scenario	1. User logs-in to the system 2. System returns a message that the login was successful 3. User selects the search for courses 4. System brings the user to the Semester/Year page 5. User selects the appropriate Semester/Year 6. System brings up the search page with the dropdown menus for department, faculty name and subject

	7. User selects the appropriate parameters for their search 8. System returns a list of courses that matches the users parameters 9. User finds the course with the parameters they wished for. Prerequisites will be displayed as well
Extensions	User wishes to add the course to their schedule, 1. User finds the course they wished for and selects a checkbox to add it to their schedule 2. System checks if the user has no holds, has the appropriate prerequisites, has space to add the course and that there are no time conflicts. If all of these are valid, the course is added to the student's schedule
Priority	Medium
Special Requirements	User has the space to add the course if needed. Faculty also has to have the course available in order to have it under the search
Open Questions	Can more parameters or an advanced search be added?

UC3	Check major requirement courses/Change Major
Primary Actor(s)	Student, Administrator
Trigger	User wishes to check their major requirements
Pre-conditions	User is registered AND user has a declared major if a student
Post-conditions	User can see their major requirements
Main Scenario	1. User logs in to the register system 2. System returns a successful log-in message 3. User selects transcript/degree audit 4. System asks for user's ID 5. User enters ID 6. System returns an option for viewing major requirements or transcript 7. User selects view major requirements 8. System brings up the user's major along with requirements completed and courses that need to be completed

Extensions	User wishes to change their major 1. Admin logs in to the system 2. System returns a successful log-in message 3. User selects major requirements 4. System asks for student's ID 5. User enters ID 6. System brings up the user's major along with requirements completed and courses that need to be completed 7. User selects declared major change 8. System asks for the admin's ID 9. Admin enters ID 10. System brings a dropdown prompt asking for which major the student would like to change to 11. User selects new declared major 12. System returns a dialog prompt asking for confirmation 13. User selects yes 14. System returns the student's major requirement page updated with the new major requirements
Priority	Medium
Special Requirements	If changing major, user must be Admin. Button must not be available to Student users. Major Requirements is not the same as transcript, it will display the current classes to be taken for a major but can not be used as a transcript.
Open Questions	If major does not exist, what option does the student have? Are minors available?

UC4	Transcript/Degree audit
Primary Actor(s)	Student, Administrator
Trigger	User wishes to view their transcript/degree audit
Pre-conditions	User is registered AND user has a major declared AND user has no holds
Post-conditions	User has their transcript displayed
Main Scenario	1. User logs in to the system 2. System returns a successful login message 3. User selects transcript/degree audit 4. System asks for the User's ID

	5. User enters ID 6. System returns an option for viewing major requirements or transcript 7. User selects view transcript/degree audit 8. System brings up the print preview assuming the user has no holds on their account. If the user has a hold, the system will print a message that user must fulfill these hold conditions to view transcript. If not, 9. User prints out the transcript
Extensions	None
Priority	Medium
Special Requirements	Faculty must put in the grades before the semester ends if the user wishes to view their transcript after the semester ends
Open Questions	Will there be a cost to print the user's transcript?
Exceptions	a. User has a hold on their account i. If student has a hold on their account, they cannot print the transcript.

UC5	View grades and hold(s)
Primary Actor(s)	Student, Administrator, Researcher
Trigger	User wishes to view a student's grades or holds
Pre-conditions	User is registered AND user is currently registered in at least one course
Post-conditions	User can see the grades in one of their courses
Main Scenario	1. User logs in to the system 2. System returns a successful login message 3. User selects View grades/holds 4. System brings up a prompt asking to view grades or holds 5. User selects view grades a. System brings up semester/year page b. User selects the semester/year they wish to view c. System brings up a list of courses taken during that semester/year

	d. User selects which course they wish to view e. System brings up a list of grades that were entered by a faculty member along with the assignment name and date 6. User selects view holds a. System brings up any holds that the student may have along with an explanation
Extensions	Researcher wishes to view grades by course 1. User selects view grades 2. System asks for researcher's ID 3. User enters their ID 4. System brings up semester/year page 5. User selects the semester/year they wish to view 6. System brings up subject page 7. User selects appropriate subject 8. System brings up a list of courses 9. User selects appropriate course 10. System brings up a list of grades sorted by faculty name
Priority	Medium
Special Requirements	None
Open Questions	When will holds be updated when the student takes care of them?

UC6	View student roster
Primary Actor(s)	Faculty, Administrator, Researcher
Trigger	User wishes to view their student roster for a specific course
Pre-conditions	User is registered AND course exists
Post-conditions	User has a list of students in a specific course
Main Scenario	1. User logs in to the system 2. System returns a successful login message 3. User selects search course 4. System brings up 5. User selects appropriate semester/year 6. User select course at desired time and day.

	7. User submit selection of course of choice
Extensions	
Priority	Medium
Special Requirements	None
Open Questions	

UC 7	View respective course room and building
Primary Actor(s)	Faculty, Administrator, Researcher
Trigger	User wishes to view their classroom and building for a specific course
Pre-conditions	User is registered AND course exists
Post-conditions	User has a list of classrooms and buildings.
Main Scenario	1. User logs in to the system 2. System returns a successful login message 3. User selects search course 4. System brings up 5. User selects appropriate semester/year 6. User select building and room. 7. System displays a list of buildings with their classrooms that are hosting a specific course
Extensions	
Priority	Medium
Special Requirements	None
Open Questions	

UC 8	Message

Primary Actor(s)	Faculty, Researcher
Trigger	Faculty User & Researcher user wants to send a message
Pre-conditions	User is registered
Post-conditions	Admin receive message
Main Scenario	1. Faculty user logs into the systems 2. System returns successful login message 3. User view assigned courses 4. User views date and time of assigned courses 5. User request day off due to health emergency or family emergency four hours prior scheduled time for course. 6. User then await approval of request 7. Admin approve days off request 8. User then proceed to email students to advise them of cancelled class for specific time and date
Extensions	1. Researcher wants to send a message to Faculty. 2. Researcher wants to send a message to Researcher 3. Faculty want to send a message to researcher 4. Faculty wants to send a message to Faculty
Priority	Low
Special Requirements	None
Open Questions	

UC 9	View student attendance
Primary Actor(s)	Administrator, Researcher
Trigger	User wants to view a student's attendance record
Pre-conditions	User is registered AND course exists

Post-conditions	User has a list of students attendance
Main Scenario	1. User logs in to the system 2. System returns a successful login message 3. User selects search course 4. System brings up a dropdown menu for semester/year selection 5. User selects appropriate semester/year 6. System brings up a list of courses 7. User selects specific course at desired time and day. 8. System brings up a list of options to choose for the student 9. User submit selection of course of choice 10. System brings up a list of students that are enrolled in that course 11. User selects student of choice 12. System returns the number of unexcused absences recorded by the professor
Extensions	
Priority	Medium
Special Requirements	None
Open Questions	

UC 10	Add and delete student hold(s)
Primary Actor(s)	Administrator
Trigger	User has to notify or update a students hold.
Pre-conditions	Students is lacking/meet academic standards.
Post-conditions	A hold is set/removed to the students account.
Main Scenario	1. User logs in to the system 2. System returns a successful login message 3. User selects Add/Delete holds 4. System brings up an option to add or delete holds 5. User selects add hold 6. System brings up a prompt asking for the Student's ID number

	7. User submits Student's ID number. 8. System brings up a dropdown menu for specified holds 9. User selects hold 10. System asks for ID with confirmation 11. User enters ID 12. System updates Student account with appropriate hold
Extensions	User wishes to add a financial hold 1. User selects financial hold 2. System asks for financial reason and amount 3. User enters financial reason and amount 4. System adds the hold to the student's account
Priority	Medium
Special Requirements	None
Open Questions	

UC 11	Create course catalog
Primary Actor(s)	Administrator
Trigger	User has to create course catalog.
Pre-conditions	Catalog needs information to be added OR catalog needs to be created
Post-conditions	Updated Catalog is posted
Main Scenario	1. Admin User logs in 2. Successful login message displayed 3. Admin User access old catalog 4. System brings up old catalog template 5. Admin User checks all information on catalog against current information on the systems. User presses the edit button 6. System brings up the edit box 7. Admin user add new information onto catalog for accuracy of information

	8. Admin then email corresponding section of the department to the head of the department for final check for errors; User submits changes 9. System confirms the information has been changed and will be updated 10. Each Department give their input on the information on the catalog 11. Admin then make final upgrade and post the new catalog under its corresponding year and semester 12. System adds the new information to the database
Extensions	None
Priority	High
Special Requirements	None
Open Questions	

UC 12	View student information
Primary Actor(s)	Administrator
Trigger	User needs to look up a students information
Pre-conditions	Student is registered
Post-conditions	User has the students information displayed
Main Scenario	1. User logs in to the system 2. System returns a successful login message 3. User selects Student information 4. System brings up a prompt to enter student's id number 5. User submits students id number. 6. System brings up Student's personal information along with class schedule and holds if any
Extensions	none
Priority	Medium

Special Requirements	User has the student's permission
Open Questions	

UC 13	Assign Faculty to a course
Primary Actor(s)	Administrator
Trigger	User wants a Faculty member to teach a specific course section
Pre-conditions	Faculty has credentials to teach course AND Faculty is available for the specific section time AND Faculty approves of teaching the course
Post conditions	Faculty is registered as the course professor
Main Scenario	1. User logs in to the system 2. System returns a successful login message 3. User selects search course 4. System brings up the search page with the options to search by course name, faculty name or CRN/Section number 5. User enters CRN/Section number 6. System brings up the course page 7. User selects add course information 8. System brings up an edit box 9. User adds the Faculty member's full name and email address to the course 10. System saves the changes when finished
Extensions	User sends an email noting that the faculty member was added to the course as the professor
Priority	Medium
Special Requirements	CRN/Section number is a key value pair

UC 14	Course Creation
Primary Actor(s)	Administrator
Trigger	User wishes to create a course/section
Pre-conditions	Course is approved to be taught AND Faculty is available to teach it
Post conditions	Course is added to the course catalog
Main Scenario	1. User logs in 2. System returns a successful login message 3. User selects course catalog 4. System brings up the course catalog 5. User selects add course 6. System brings up a form 7. User fills out the course form with CRN/Section, description, time/day/semester/year, credit amount, roster size, and room/building number 8. System accepts form
Extensions	User wants to add a faculty member (See UC 13)
Priority	Medium
Special Requirements	CRN/Section is a key value pair

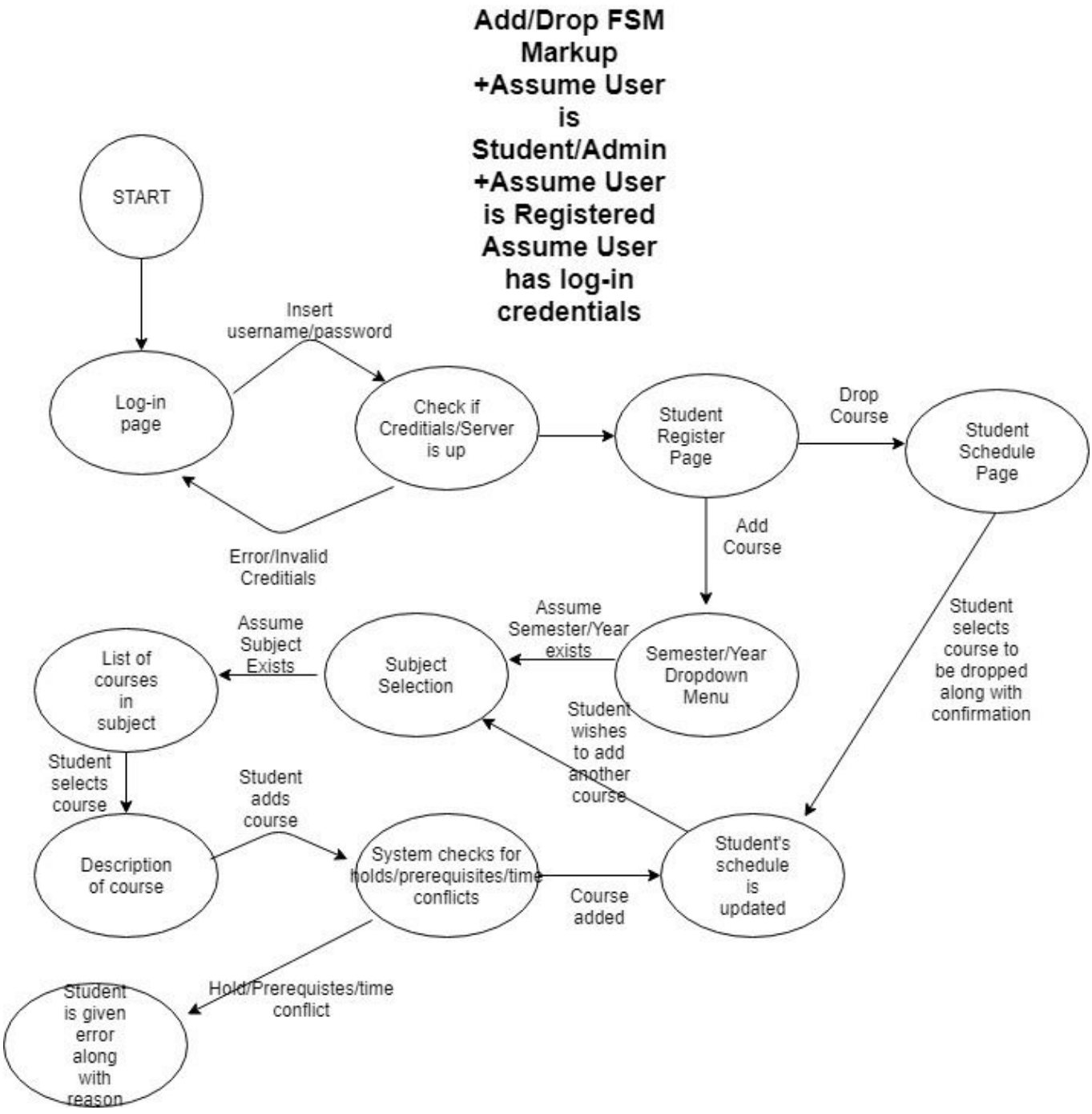
UC 15	View Schedule
Primary Actor(s)	Student, Faculty
Trigger	User wishes to view their schedule
Pre-condition	User is registered AND Student has at least one course registered
Post-condition	Student’s schedule is displayed

Main Scenario	3. User logs in to the system 4. System returns a successful log-in message 5. User selects view schedule 6. System returns a printable copy of the Student's schedule with the Course Name/Time/Day(s)/Building/Room Number
Extensions	User wishes to view detailed schedule 1. User selects view detailed schedule 2. Systems returns a printable copy of the Student's schedule with the Course Name/CRN/Section/Time/Day(s)/Credits/Semester/Faculty Name/Faculty Email/Building/Room Number
Priority	Medium
Special Requirements	Building/Room Number is a key-value pair.

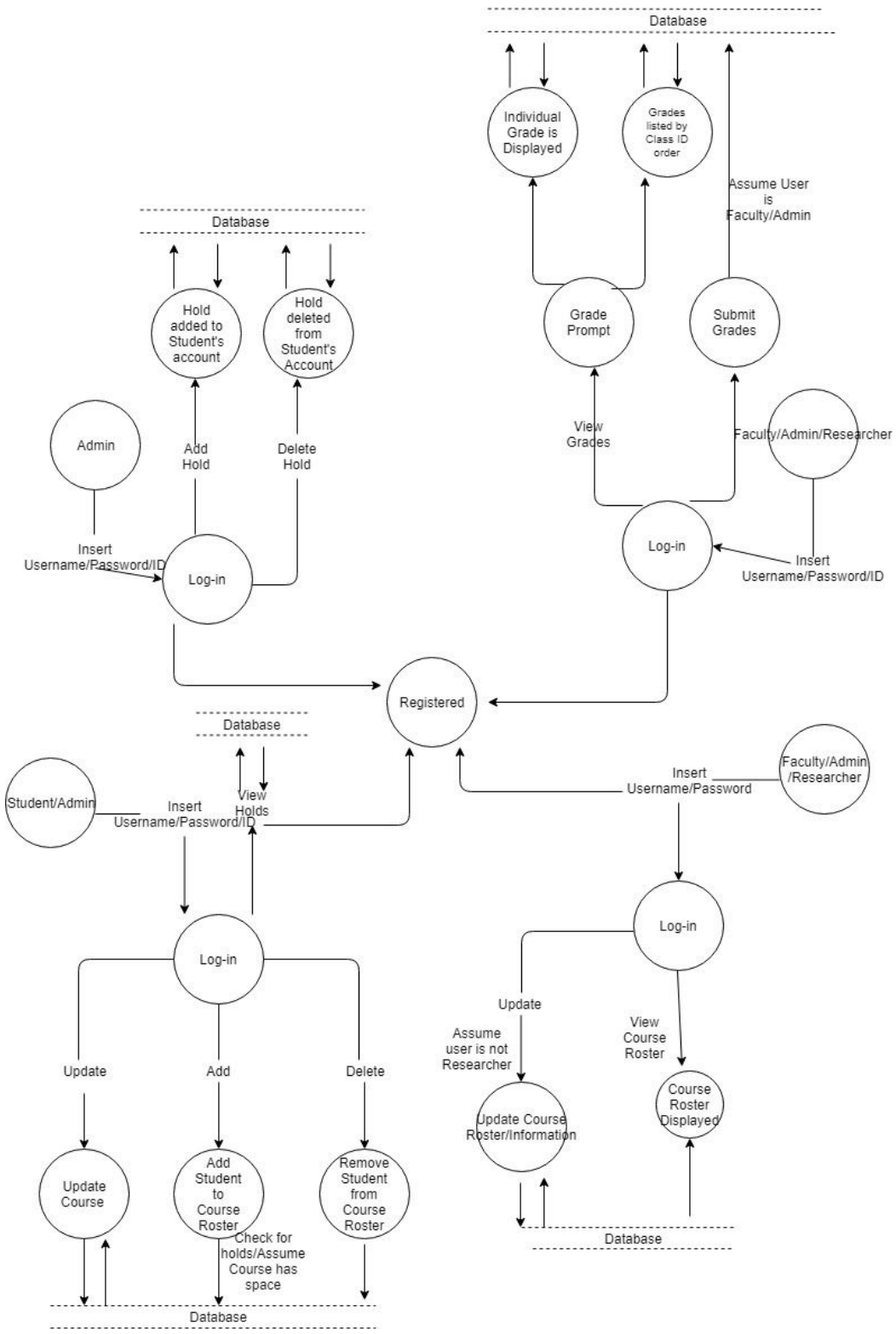
UC 16	Record Course Attendance
Primary Actor(s)	Faculty
Trigger	User wishes to record attendance for a course
Pre-conditions	Course exists
Post-conditions	Course attendance is recorded
Main Scenario	1. User logs in to the register system 2. System returns a successful login message 3. User selects view courses registered 4. System returns a list of courses being taught by the professor this semester 5. User selects appropriate course/section 6. System displays the course description and a list of students enrolled in the section 7. User submits attendance

	8. System saves attendance to the database
Extensions	None
Priority	Medium
Special Requirements	None

5.2 FSM Diagram Mark-up of adding courses



5.3 Data flow diagram mark-up of Add/Drop, Viewing Grades/Holds and Add/Delete Holds



6. Risks and Assumptions

- User information integrity.
- Can't allow any user but the professor to assign a grade to a student.
- Can't allow other users to view each other's personal information that can't be shared under specific circumstances.

7. Skills/Time usage

Most of the time used on this project will be used toward making sure the database works properly. That entails creating the database and thoroughly testing that all the information we need to obtain is easily. The rest of the time will be implementing the GUI so that we can visually view the data obtained from the database. Tyler and Mc Ali will be working the most on the database since this meets their skill sets the best. Nicole and Neville will be working on the GUI. Nicole will help out with the database when needed.

Version History:

1.1: Started Document

1.2: Added Functional Statements, Human Interaction, Sample Human Interactions

1.3: Added Use case diagram, FSM draft

1.4: Added Business Rules, Added UC2-6, updated UC1

1.5: Started to edit/format document, Added rest of UC

1.6 Finished Database information, diagrams and optional information

1.7 Fixed Errors, Added Explicit Required/Forbidden for business rules, fixed database schema

1.8 Updated Document w/ Database Information and use cases

1.9 Added more use cases (13-16)

2.0 Cleaned up Requirements/Forbidden and Database Entities

2.1 Cleaned up Requirements and Database Entities

2.2 Added ERD draft

2.3 Cleaned up ERD and Entity Attributes

2.4 Cleaned up ERD section and Entity Relations