Graduation Requirements

Simplifier

ICS370

Fall 2020

Bradley Taylor, Rene Ntumnui,
David Qual

Elaboration Phase III

Table of Contents & Status Report	
Project Component	**% Done**
1. Introduction	100
System Concept	
2. Business Case and Project Vision	100
3. Stakeholders	100
Business Process Requirements	
4. Business Process Analysis	100
System Requirements	
5. System Actors	100
6. Functional Requirements	100
7. Non-Functional Requirements	100
8. Mandated Constraints	100
9. Relevant Facts and Assumptions	100
Design Diagrams	
10. Updated Use Case Diagram	100
11. Updated Domain Model	100
12. Updated Class Diagram	100
13. Updated System Sequence Diagrams	100
14. Updated Sequence Diagrams	100
15. Updated Activity Diagrams	100
16. Updated State Machine Diagram	100
Appendices	

Communication Log					
Date	Communication Topic	Comm Type	Participants		
			Team Members	Client Reps	Other
Deliverable 1					
9/10/20	Phase 1 Preliminary Mtg	Zoom	Bradley, Rene, David		
9/12/20	Demo Prep	Zoom	Bradley, Rene, David		
9/16/20	Demo	Zoom	Bradley, Rene, David		
Deliverable 2					
10/9/20	Phase 2 Preliminary Mtg	Zoom	Bradley, Rene, David		
10/13/20	Demo Prep	Zoom	Bradley, Rene, David		
10/15/20	Phase 2 Demo	Zoom	Bradley, Rene, David		
Deliverable 3					
11/8/20	Phase 3 Preliminary Mtg	Zoom	Bradley, Rene, David		
11/10/20	Demo Prep	Zoom	Bradley, Rene, David		
11/11/20	Phase 3 Demo	Zoom	Bradley, Rene, David		
Deliverable 4					
11/30	Phase 4 Preliminary Mtg	Zoom	Bradley, Rene, David		
12/5	Demo Prep	Zoom	Bradley, Rene, David		
12/9	Final Demo	Zoom	Bradley, Rene, David		
Final Project Deliverable, Visual Simulation and Presentation					

Table of Contents

Introduction

In this document we describe the requirements for the Graduation Simplifier System. We first describe the system concept and vision, followed by a description of actors, system boundary and scope of the system. We then provide more detailed requirements with use cases, followed by a description of non-functional requirements for the system. Finally, we provide a data model and other information management specifications. Please refer to Appendix A for a glossary of terms and acronyms used in this document.

Business Case and Project Vision

The average bachelor's degree requires 120 college credits. However, the average recipient completes 134 credits before graduating. There are several factors that contribute to this phenomenon. We decided to explore those factors and determine what can be done to help with the problem.

One reason that students take extra classes is because they fail to take advantage of advising. Most, if not all universities, offer extensive academic advising services. However, advisors can be overloaded on cases and utilizing their services can take a great deal of time and effort. Thus, many students fail to follow through.

Other factors involve changing majors and/or transferring to different schools. Requirements can change from school to school and from program to program. This can cause great confusion and ultimately contributes to students taking unnecessary classes.

Degree audits can be very difficult to read. There are categories laid out in the audits, but often there are conditions listed that become very complicated. The wording of these conditions can almost resemble legalese, making them hard to interpret.

In addition to the factors mentioned above, I recently sat through student introductions for a college course. While giving their introductions, four separate students stated that they did not know when they would complete their major. It was apparent that they did not grasp the requirements to graduate with a degree.

Our desire is to create a simple application that will allow a student to better understand the requirements to graduate a given course of study. This system will have a simple user interface that can be accessed over the internet. There will be no appointments to make with advisors and no complicated degree audit to read. The scope will be programs at Metropolitan State University.

Stakeholders and Requirements

Introduction

This requirements report has been prepared to support our development of a software solution intended to be used by students that attend Metropolitan State University. We intend to show that this project will be beneficial across a variety of spectrums. These spectrums include, technical, financial, economic, and political. We will also explore the potential market value of this project.

Section 1 – Technical Factors and Requirements

We considered the technical feasibility of the project from two different viewpoints. This allowed for us to get a full picture of whether the project could be completely funded from start to finish.

The first was project development and infrastructure. After researching several tools and integrated development environments, we found that we could utilize freeware in some instances and low-cost subscription SaaS products in others.

The second viewpoint required us to consider the technology of all potential end users. We determined that all of the students that attend Metropolitan State have access to laptop or desktop computers. Many have their own devices, in other circumstances, the University has computer workstations available for student use.

In summary, each of the products that will be needed by developers are supported by the current technical resources that we have on hand, no additional hardware will need to be purchased. Also, there will be no costs associated with end users, which is ideal. This will allow us to stay within our allotted budget for development and infrastructure, a topic that will be discussed in more detail in the following section.

Section 2 – Financial Factors

Software development projects often involve numerous technical and operational expenses, particularly in the early phases. For this undertaking, we have gone to great lengths to avoid that potential pitfall.

Our project staffing consists of three individuals working together as a team. As mentioned in the previous section, all of the needed hardware for the project is in hand.

We do expect to incur minimal expenses related to software, this may come in the form of licensing or subscription fees related to development and hosting of the solution.

Section 3 – Economic Factors

The previous section explored the costs associated with developing this solution, this section will focus on the ability of the solution to break even or become profitable in the future.

We do anticipate a considerable incubation period for this software feature before it becomes widely used and appreciated. This is largely due to factors beyond our control. The University follows yearly cycles of enrollment, registration, and course completion and evaluation. These cycles repeat throughout Fall, Spring, and Summer semesters. Graduation requirements are not always considered daily but tend to coincide with the aforementioned semester cycles. This phenomenon will contribute to the incubation period.

The next section of this report will explore additional factors that may contribute to a potentially lengthy break-even period. We understand that this is not always ideal for a software project. However, the nature of our project will afford us the ability to allow several semesters to elapse as students and advisors grow in their appreciation of this powerful tool.

Section 4 – Political Factors

In a previous section, we mentioned that there could be a considerable period of time that elapses before this system becomes widely used. Our study on political feasibility helped us to form that belief. We have, however, concluded that none of the factors considered were harmfully significant or insurmountable.

Our project will be designed with the intention of helping students and also advisors. We expect the latter to involve political implications. The reason being that any outside attempt to improve processes or procedures often implies that a weakness or deficiency currently exists.

Previous experience shows that attempts to improve processes or procedures within any group or department, often incurs initial backlash from stakeholders or actors. We anticipate that, in this case, the benefits will be significant enough that the any pushback will be short lived.

Section 5 – Market Factors

Our completed offering will have great potential within the Metropolitan State University system. We envision that students will find the solution to be intuitive and easy to use. It will negate the need to schedule countless appointments with their advisor and will provide peace of mind.

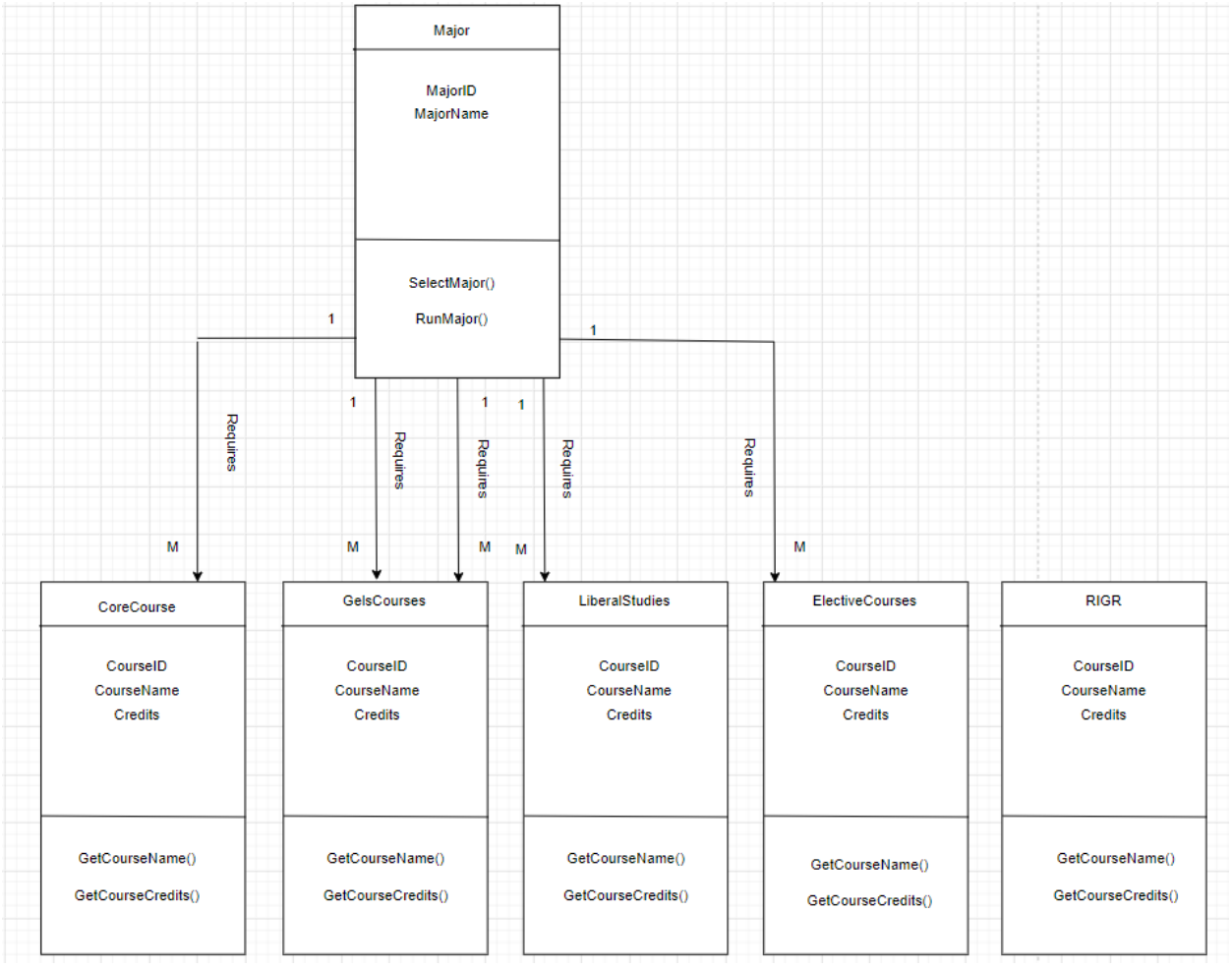
Advisors will also greatly benefit from this solution. Informed students will free up advisors to perform their essential duties and help them to manage their caseloads more efficiently.

Finally, we expect this solution to encounter demand from other colleges and universities. This simple application will improve student experience and advisor efficiency and thus will be coveted throughout the industry. This factor greatly mitigates the lengthy break-even period previously discussed.

GoF Design Patterns

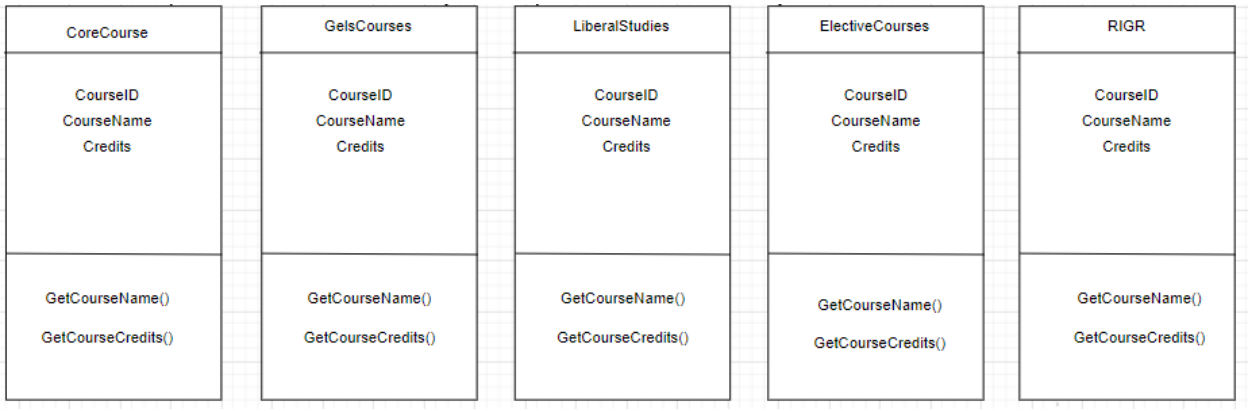
Composite Pattern

Composite patterns allow each object to function with the same interface. This allows clients to treat objects and their compositions the same way. We found this useful because we have several types of course requirements that are classes having functionality that is much the same as seen in the snapshot below:



High Cohesion

High cohesion is an evaluative pattern that attempts to keep objects appropriately focused, manageable, and understandable. High cohesion is generally used in support of low coupling.
The term cohesion is used to indicate the degree to which a class has a single, well-focused responsibility. Cohesion is a measure of how the methods of a class, or a module are meaningfully and strongly related and how focused they are in providing a well-defined purpose to the system.
Our use of this concept is evident when you look at the efficiency of our class methods demonstrated in the graphic below:



Appendix A

Users/Actors	Use Cases/Functionalities
Student	• Login • Select Major • Upload Transcript/Dars Report • View Results • Print Results • Download Results
Administrator	• Login • Upload CSV File

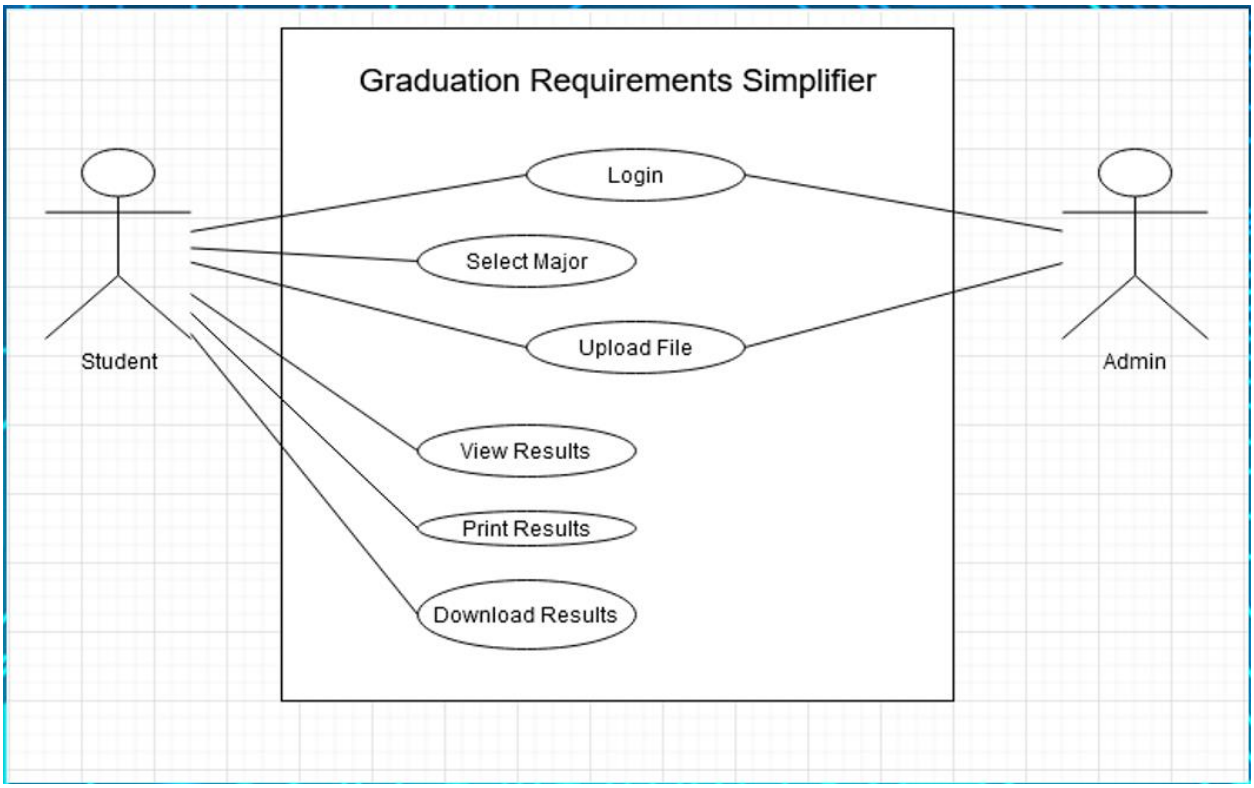
Student

The student will be the primary actor in our system. The student will have to be able to login to access the interface of the program. Upon gaining access to the interface, the student will need access to several program features. These features include the ability to select a major course of study, upload a transcript or Dars Report, run the program, and access the results. Additionally, the student will need to view, download, or print the results.

Administrator

There will be a system administrator that needs to interact with the system. The administrator will need to be able to login to access the administrative features of the program. Upon gaining access to the administrator interface, the instructor will need the ability to configure and maintain program majors and requirements.

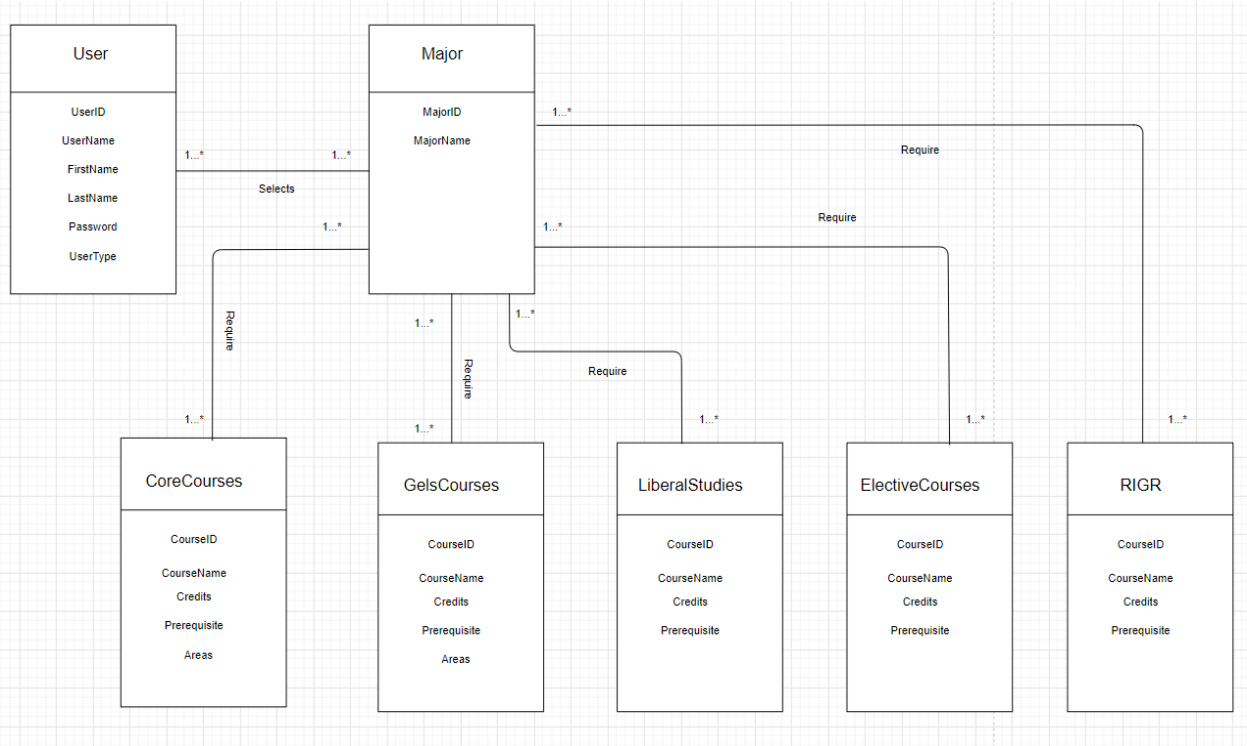
Use Case Diagram



Appendix B

Domain Model

The domain model represents relationships between the conceptual classes in our system.



Appendix C

Software Architecture Document

Issues:

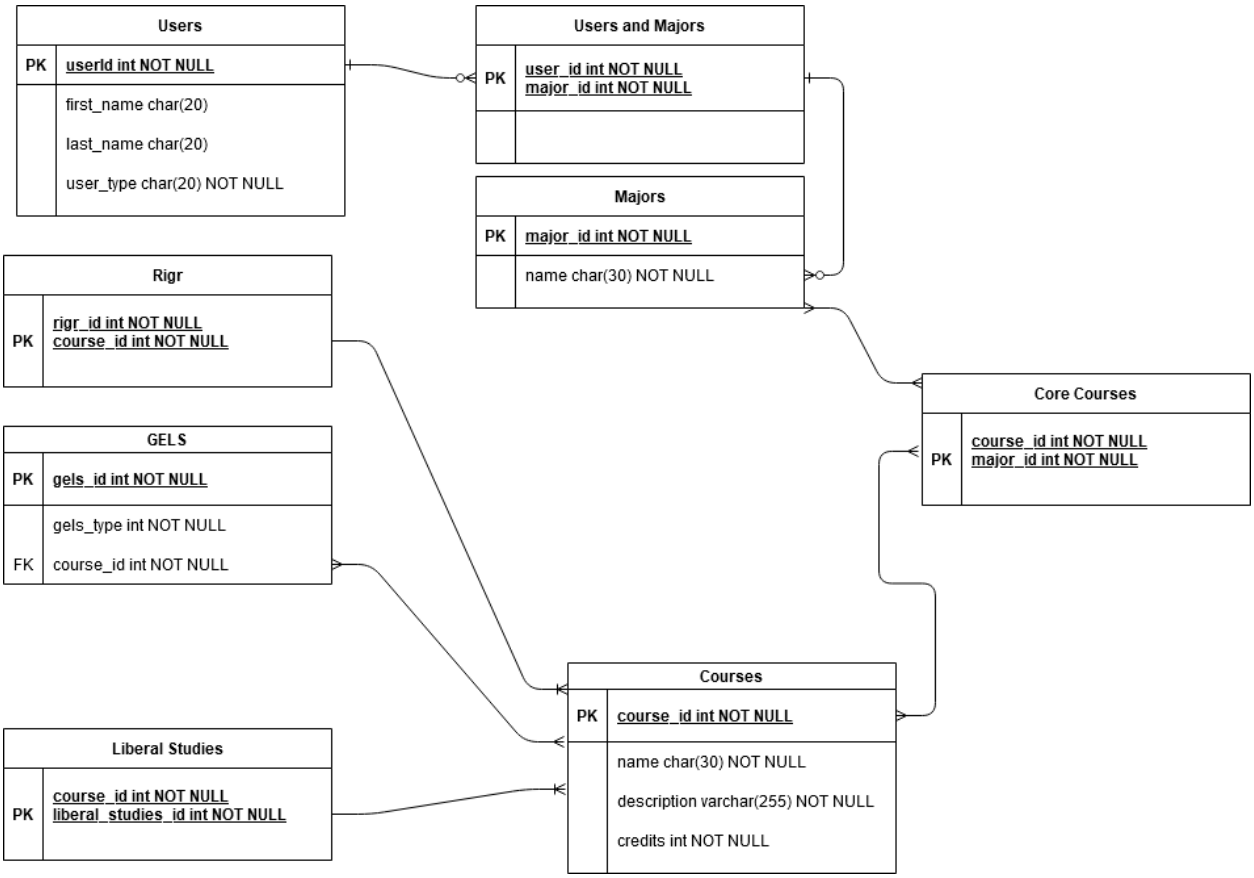
- Moving from python to a web app
- Generating and retrieving data
- Building databases
- Missing logic to build the fastest approach to completing a major
- Missing logic to ingest files
- Missing logic to handle searching
- Missing logic to handle logging in

Resolutions:

- Moving to from python to a web app has been successful, the UI has been fully migrated and there is now more data than was in the original prototype. This should be considered complete.
- Generating and retrieving data was in flight and will be an ongoing issue. Through the usage of existing helper code the generation of relevant data for the databases can be harvested and cleaned.
- Building the databases has been completed
- File ingestion
- Search logic
- Logging in to the system
- Fastest possible path to Major completion

Appendix D

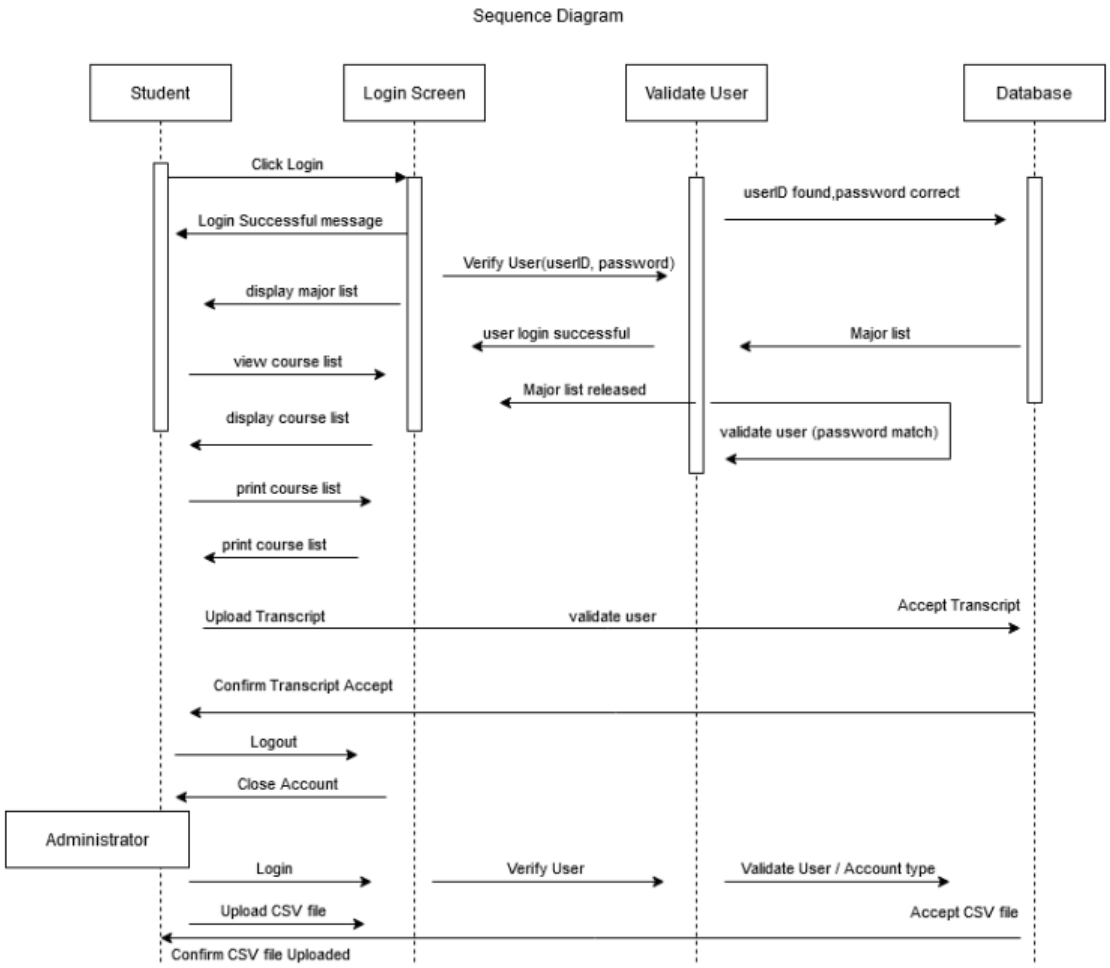
Data Model



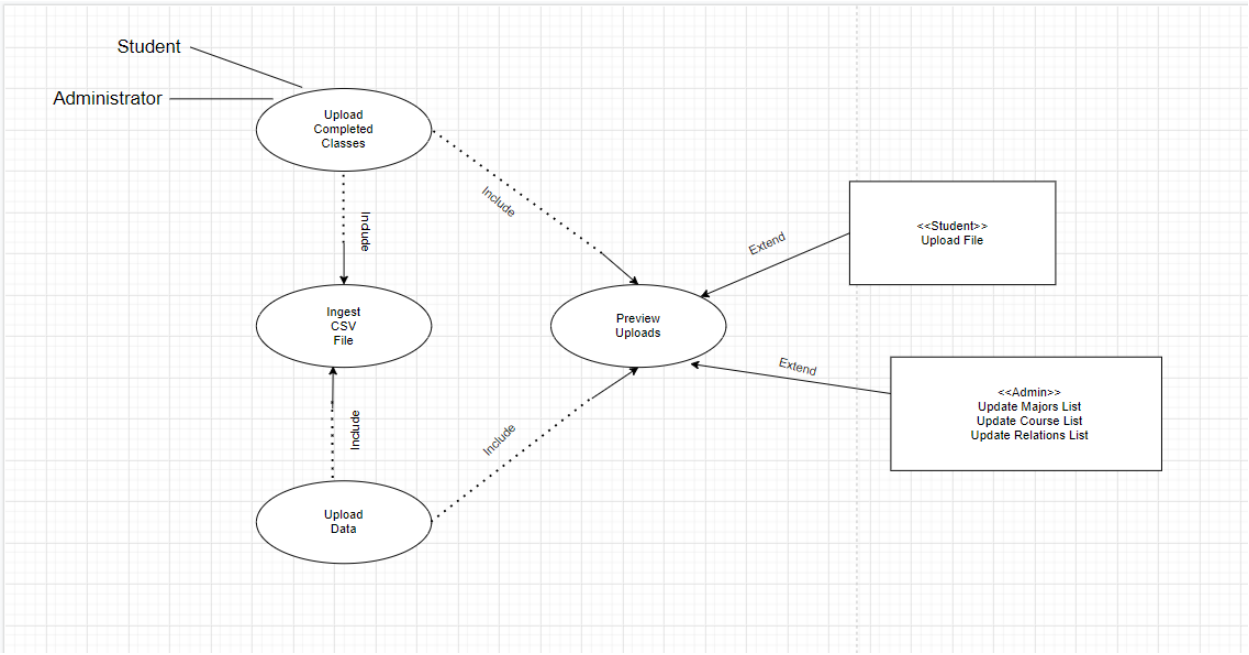
Appendix E

System Sequence Diagram

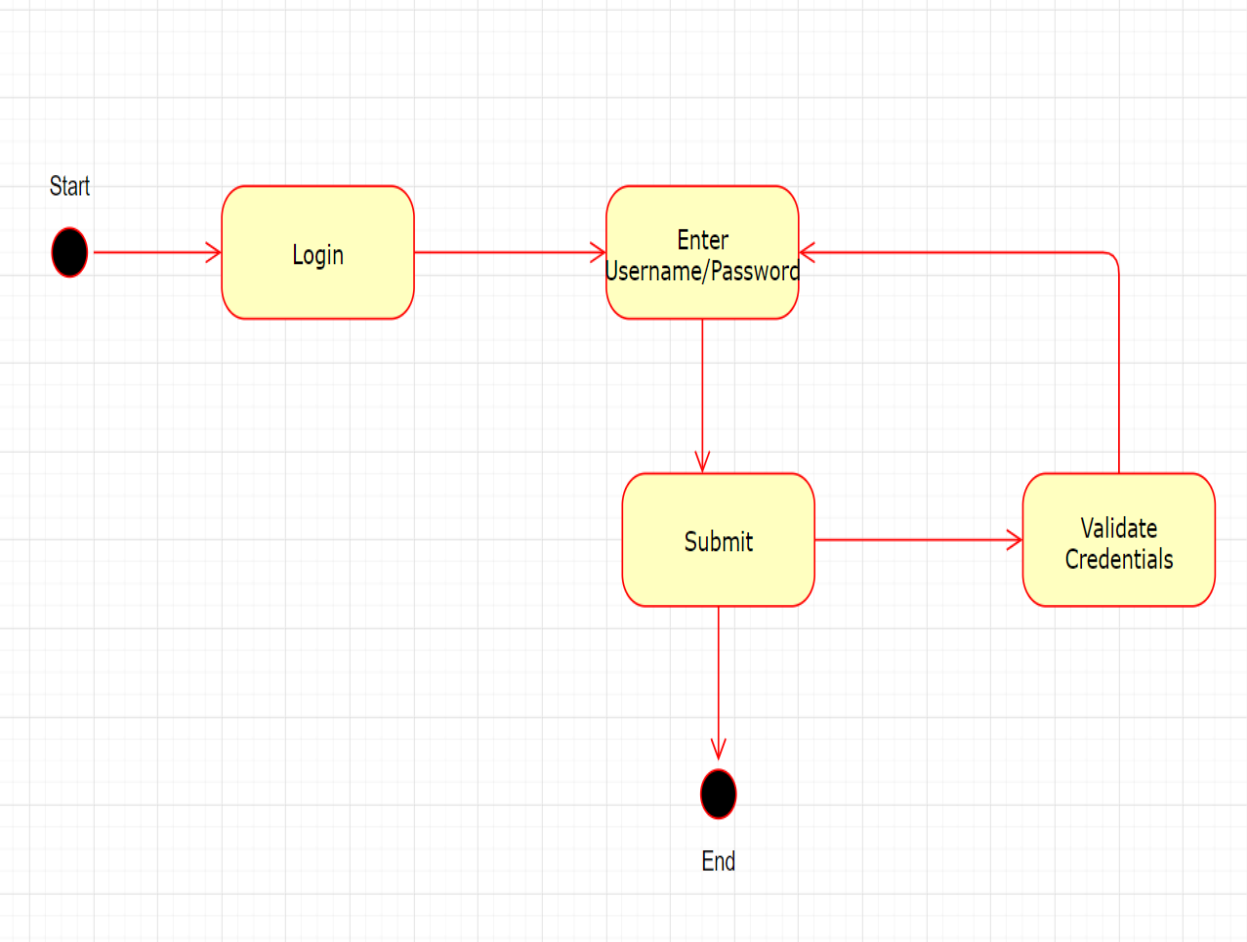
The system sequence diagram below describes how the actors interact with the system. It captures the input and outputs as actions and events occur within the system.



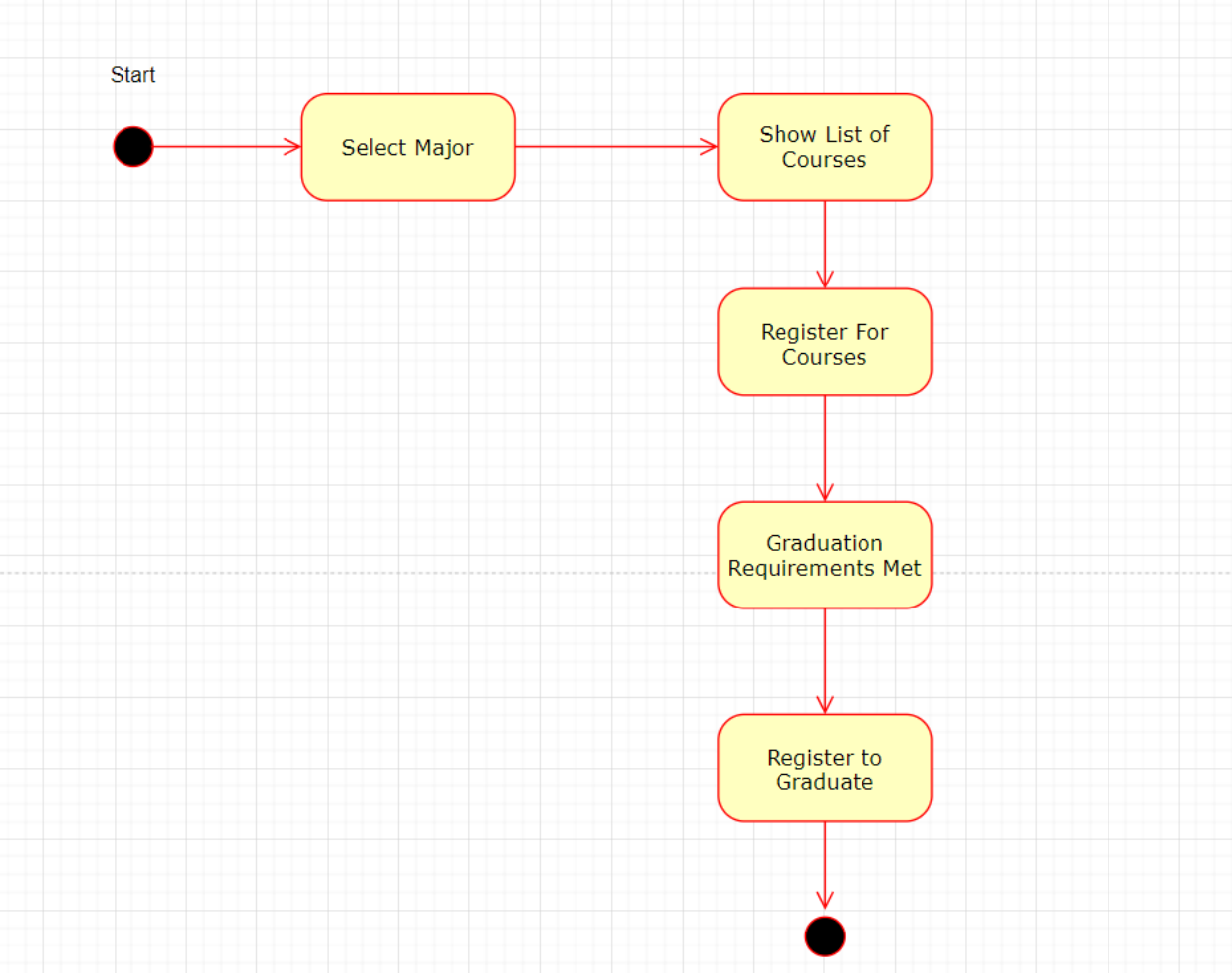
Activity Diagram 1



Activity Diagram 2



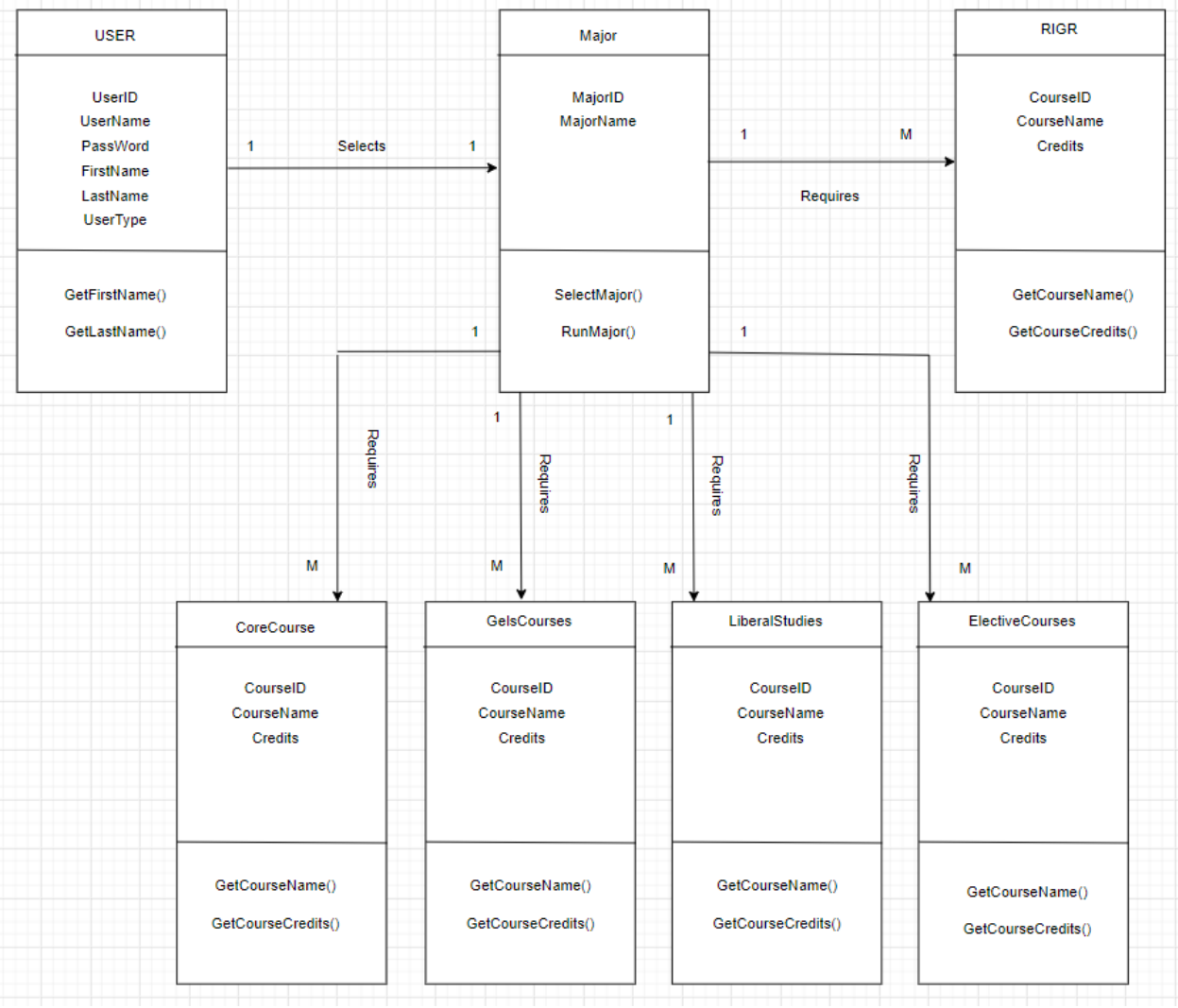
Activity Diagram 3



Appendix F

Class Diagram

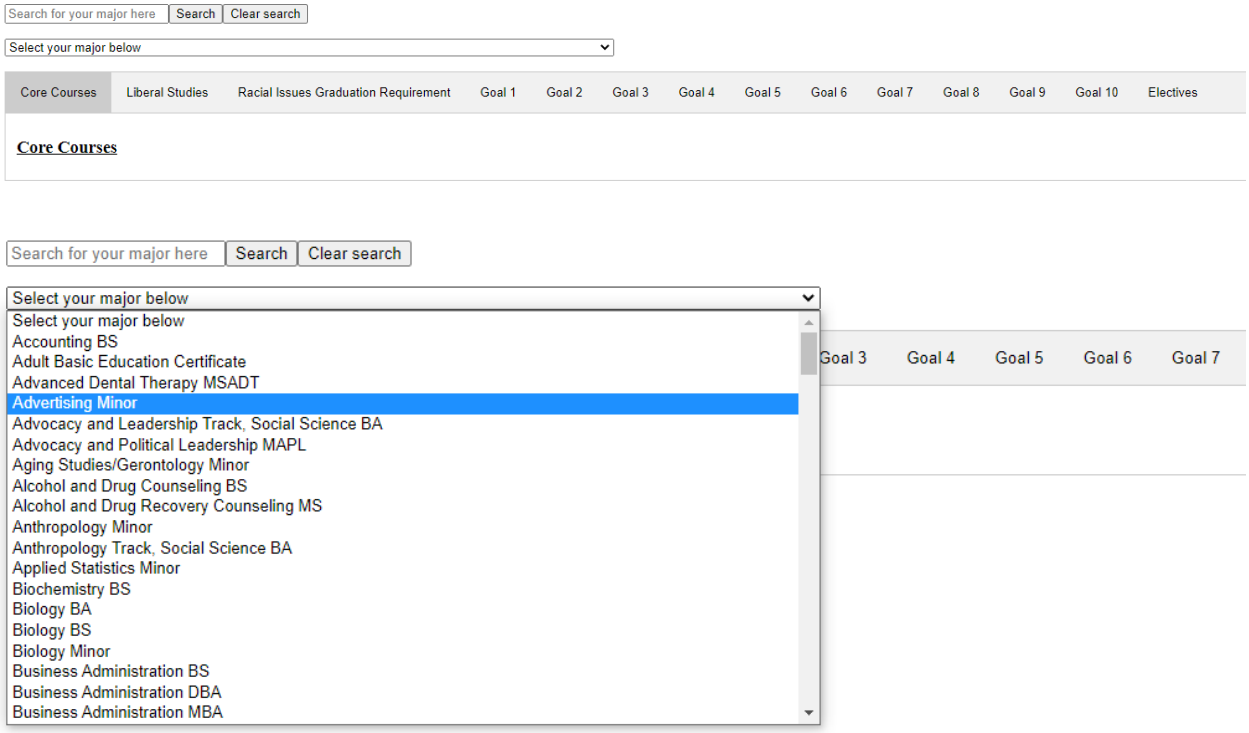
The class diagram below consists of the set of classes that need to be developed within our system. Each class appears with its attributes and methods. Relationships between classes are also defined.



Appendix G

Prototype information and Examples

Current code count including helper code is 1181 lines, and 182 files were generated.



Graduation Requirements Simplifier

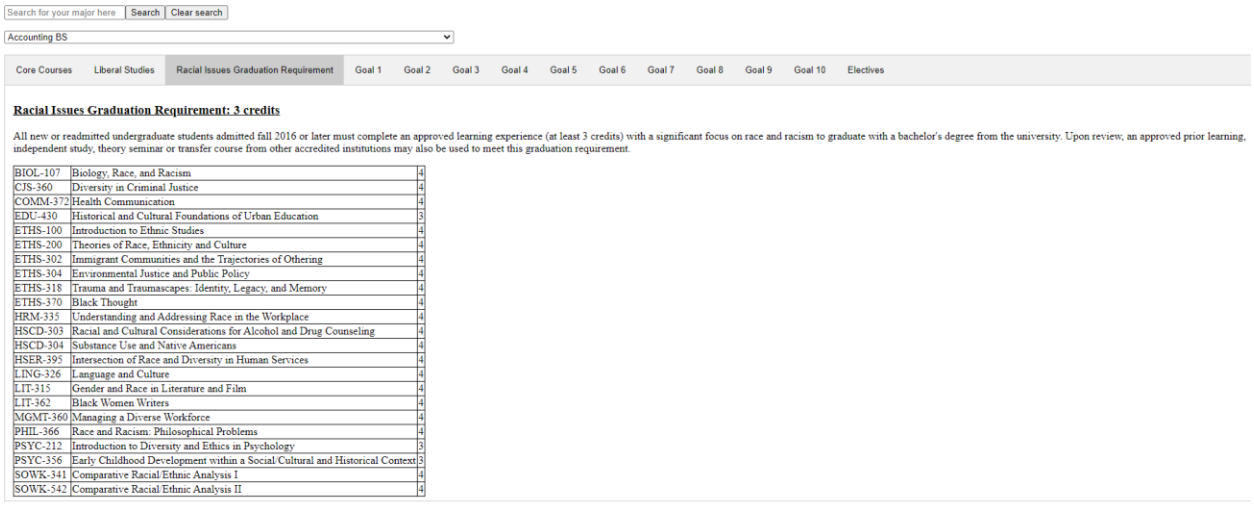
Search for your major here | Search | Clear search

Accounting BS

Core Courses | Liberal Studies | Racial Issues Graduation Requirement | Goal 1 | Goal 2 | Goal 3 | Goal 4 | Goal 5 | Goal 6 | Goal 7 | Goal 8 | Goal 9 | Goal 10 | Electives

Core Courses

ACCT 210 Financial Accounting
ECON 201 Macroeconomics
ECON 202 Microeconomics
MATH 115 College Algebra
MIS 100 Fundamentals of Information Technology in Organizations
STAT 201 Statistics I
DSCI 434 Introduction to Operations Management
FIN 390 Principles of Finance
MGMT 310 Management Principles and Practices
MIS 310 Principles of Management Information Systems
MKTG 300 Marketing Principles
ACCT 311 Intermediate Accounting I
ACCT 312 Intermediate Accounting II
ACCT 320 Strategic Management Accounting
ACCT 340 Accounting Information Systems
ACCT 510 Advanced Accounting
ACCT 512 Auditing
ACCT 520 Advanced Strategic Management Accounting
ACCT 530 Business Taxation
MGMT 499 Case Studies in Strategic Management
ACCT 515 Financial Statement Analysis
ACCT 515 Financial Statement Analysis
BLAW 310 Business Law: UCC and Contracts



Graduation Requirements Simplifier