Course: Software Engineering

Course Number and Section: 14:332:452

Group #5

Virtual Logic Lab

Report 3

Website: https://github.com/GeorgeM-K/onlinedb2

Submission Date: April 14, 2019

Group Members (8):
James Ramos, Andy Lee, Deeptanshu Murdeshwar, Christopher Basilio, Daniel Chan, Drew Koskinen, George Melman-Kenny, Willear Glimniene

Subgroup	Member	Contribution
A	James Ramos	- History of Work - Updated list of accomplishments - Updated Breakdown of Responsibilities - Updated Glossary of Terms - Updated system sequence diagrams (UC-5 and added diagram for FAQ pages) - Updated Traceability matrix and Domain Model for UC-19 - Updated Domain Analysis
A	Daniel Chan	- Formatted Report - Summary of Changes - Summary of Changes (Feedback) - New version of Class Diagram - Edited Class for Prelabs - Added Class for Postlabs - Updated System Sequence Diagrams for UC - 5 and 18 - Updated Gantt Chart - Updated History of Work
B	Drew Koskinen	- Updated Breakdown of Responsibilities - Updated Functional Requirements Specification - Updated Use Case Diagrams - Updated Traceability Matrix - Edited Project Management
B	Andy Lee	- History of Work - Summary of Changes - Updated Breakdown of Responsibilities - Updated Gantt Chart - Updated Glossary of Terms
C	George Melman-Kenny	- History of Work - Updated References - Updated Design of Tests - Updated User Design and Implementation
C	Christopher Basilio	- Updated Breakdown of Responsibilities (Programming) - Updated Gantt Chart - Design of Tests - History of Work

		- Contributed to Summary of Changes
D	Willear Glimniene	- History of Work and Breakdown of Responsibilities
D	Deeptanshu Murdeshwar	- Formatted report and outline - Updated History of Work - Updated Breakdown of Responsibilities - Updated Gantt chart - Added to Summary of Changes

Table of Contents:

Summary of Changes

---

1. The Screen Capture mechanic was not 100% able to be implement as originally planned. There is no screen sharing feature implemented. Pressing the buttons “\” and “s” simultaneously takes a screenshot of the current screen and then the user is able to email this picture to the instructor.
2. Lab 5 was not implemented.
3. The symbol keyboard for the chat was not implemented.
4. The partial credit grading system was implemented, however viewing which parts of the assignment that the student got correct or incorrect was not implemented f. The gradebook only shows the final score of the assignment.
5. Recommended “topics to review” was not implemented, this feature was not a high priority and other features took more time than expected.
6. Viewing statistics for specific parts of an assignment was no longer necessary since we only implemented a gradebook for the score of each assignment.
7. Quizzes are no longer graded and there just for student practice.

Summary of Changes (Based on Feedback)

After presenting demo 1, we got feedback from Professor Marsic as well as the TAs on what we could do to further improve our project. We used this feedback to shift our focus and try and implement or adjust based on those comments.

1. Request to add a FAQ page: Added a FAQ section that can be used by students and instructors and has a search field to find if a question was already answered.
2. Request to develop more labs: Not only did we create three labs from scratch, but we also incorporated the partial grading system developed initially for Lab 1.
3. Request to create forgot password button: Button to retrieve password was place into the login interface.

Customer Problem Statement

Problem Statement:

Education in any field can be significantly enhanced with the addition of laboratory experience. With proper laboratory work, students are able to put the theory and idea into practice. However, there is difficulty in providing labs for large classes of students. This is due to a number of factors including the number of people, logistics, faulty equipment, or financial restrictions of the schools/universities. Lack of physical resources can become troublesome for students because it will restrict their academic growth. In recent years technology has become a catalyst regarding academic development for many students. The accessibility and prevalence of technology has allowed students to use it as a supplement for training, lectures, homework, and more. A big step for integrating education into technology is to have the ability to conduct a lab experiment virtually through simulation. In an effort to bring about the idea of virtual laboratories, the goal of this project is to build upon existing virtual-lab projects from previous years. More specifically, our focus will be on Digital Logic Design Lab (DLD) experiments. The experiments from this course are very technical and require a lot of different parts (power source, breadboard, logic gates, LEDs, switches, etc.) However, some labs do not necessarily require physical interactions with the pieces to gain a full understanding of the material and working of the circuits. These labs could easily be completed in an online environment.

There are many target consumers of our planned system. The virtual digital logic design lab is to focus on the education obtained from a Digital Logic Design course, from a technical engineering track. However the tool should be utilized by more than just the students. Our focus is to make this a useable system for all parties involved. This includes the professor, the laboratory instructors, and the students themselves.

- Student perspective:

The growing number of students in my classes exceed the number of resources that my university can provide for laboratory work. These resources include power sources, bread boards, logic probes, and more. As a result, the quality of my education decreases in some ways. The lab experiments, which put theory into practice, are hit the hardest by the lack of resources. The lab equipment is limited, and even then the supplies available don't all work correctly. Due to the lack of equipment, bigger groups are needed. Larger groups prevent us from gaining much hands-on experience due to the fact that not too many hands can be working with the breadboard at the same time. The lab equipment can be dated, and inconsistent. Low quality equipment causes frustration and confusion. We waste time trying to figure out where we went wrong in the lab because our truth tables are not coming out how they should. We rebuild the circuit one or more times trying to find the source of our problem, only to realize that the breadboard is broken or something amongst those lines. The faulty lab equipment will also cause inaccuracy with my results.

Having only one Lab instructor per lab section results in us not receive enough help or time to complete the lab experiment. The instructor is spread too thin. Lab checkpoints that

require the lab instructor's approval end up wasting a lot of time for student learning, since we basically remain stagnant until we are given the go ahead. I would like to minimize the time wasted by waiting to receive help and approval from an instructor. I would also like the freedom to chat with my instructor in order to obtain the best approach for the experiment. Furthermore, I would prefer equipment with better accuracy however replacing the equipment in the laboratories will cost the school a lot of money, therefore I feel a simulated approach would be optimal in tackling this issue. Finally, I would not like to wait for my lab grade to come in and prefer to obtain direct feedback in the form of statistics. Some form of way to tell if I got the correct outputs would be ideal.

- Lab Instructor perspective:

One of the most significant problems that students face when it comes to lab courses stems from not being able to continually have a TA readily available and ready to answer questions. As an instructor, I believe most of my time is spent checking the circuits of many students so that they may move on to the next part of the lab, rather than answering necessary theoretical questions about the experiment. I would like to help each student during the lab so that they can apply their theoretical knowledge and integrate it into actual circuits. However, checking each person's work takes a lot of time and does not allow me to get to everyone's questions. Another problem that I face is due to the university's financial restriction. Financial restrictions limit the equipment that students use, in terms of quality and quantity. Instead of helping the students understanding of the material, I am stuck trying to figure out why their properly built circuit is not correctly working. We end up needing to scavenge the supply boxes to see what IC chips or logic gates are actually working. The equipment can be very unreliable and cause more problems. I would like to have reliable laboratory equipment and enough of it for each of the students in the class. This way there are not any students being carried by their partners, where one does all the work, and the others watch. I would like to have a program that would let students know if their experiments are correct automatically. If they are correct, the students can move on without being held back, and I can dedicate more attention to students that need it. In this way, those that are stuck can better communicate with me, perhaps via a chat system. I could also use the system to make announcements that may provide hints for the students. This would not only satisfy the student's curiosity, but it would also be beneficial to me when it comes to covering multiple groups questions' at the same time. Also, it may help if I am not restricted to the time of the lab. Having 24/7 access would allow me to help students anytime I am available on the system. I also find it that some students work more effectively at distinct times of the day, so this could better suit their needs. Since these labs are logic based, having a virtual lab that automatically grades the results of the experiment will allow for more time spent helping those that need the attention.

- Professor perspective:

The number of students in my classes has gotten too large to accommodate their individual academic needs adequately. I need the help of teaching assistants to help manage classes that are very large. Due to financial restrictions, I am not able to hire enough people to help keep up with the number of students. Putting too many people between the students and I reduces my awareness of which topics students are struggling with. I would like to know where and when students are struggling as early as possible. Being able to determine student problems earlier on will allow me to review topics more thoroughly before they are examined on those topics. A proper way to combat this may be to instantiate a feedback system where I can visualize the performance trends of students in real time; if I were to get a notification when students were repeatedly making mistakes, I could tune in to what the issue was and think of ways to better explain the material if need be.

Glossary of Terms:

Administrator	An "Actor" of this project that takes the role of viewing grades, create and delete users. Also referred to as "Teacher", "Professor", and "Instructor".
AND Gate	A boolean operator that gives the value one if and only if all the operands are one, and otherwise has a value of zero
Boolean	A binary variable, having two possible values, "true" and "false.".
Combinational System	A logical system with no memory for which the output depends only on the current input values
Full Adder	The full-adder circuit adds three one-bit binary numbers (C A B) and outputs two one-bit binary numbers, a sum (S) and a carry (C1). The full-adder is usually a component in a cascade of adders, which add 8, 16, 32, etc. binary numbers.
Gray Code	Gray code is an ordering of the binary numeral system such that two successive values differ in only one bit (binary digit).
Karnaugh Map	A diagram consisting of a rectangular array of squares, each representing a different combination of the variables of a Boolean function
LED (Light-Emitting Diode)	A semiconductor device that emits visible light when an electric current passes through it

Logic Diagram	Logic diagrams are diagrams in the field of logic, used for representation and to carry out certain types of reasoning.
Logical Sum	A logical term that is the Boolean OR of two or more variables
Logical Product	A logical term that is the Boolean AND of two or more variables
Maxterm	A Boolean expression resulting in 0 for the output of a single cell expression, and 1s for all other cells in the Karnaugh map, or truth table.
Minterm	A Boolean expression resulting in 1 for the output of a single cell, and 0s for all other cells in a Karnaugh map, or truth table. If a minterm has a single 1 and the remaining cells as 0s, it would appear to cover a minimum area of 1s.
NAND Gate	A Boolean operator that gives the value zero if and only if all the operands have a value of one, and otherwise has a value of one (equivalent to NOT AND).
NOR Gate	A Boolean operator that gives the value one if at least one operand (or input) has a value of zero, and otherwise has a value of one.
OR Gate	A Boolean operator that gives the value one if at least one operand (or input) has a value of one, and otherwise has a value of zero.
Protoboard	A board for making an experimental model of an electric circuit
Student	An “Actor” of this project that takes the role of making use of the laboratory simulations that is graded.
Switch	A device for making and breaking the connection in an electric circuit
Truth Table	A tabular list of all possible input combinations for a combinational system and the corresponding outputs
XOR gate	A digital logic gate that gives a logic one output when the number of true inputs is odd, and otherwise has a value of zero
Section	A class that has a unique name and contains grades for students that belong to that class
Class Statistics	Statistical variables such as mean, median, mode, standard deviation, etc. The statistical variables are calculated using the data from the gradebook
Instructor Home	A scene that the instructor can access. This scene allows the instructor to access the FAQ page, chat with students, create classes, delete classes, and access the class page.

Class Page	A scene that the instructor can access. This scene allows the instructor to view grades of students for a particular assignment, view the gradebook, view the class statistics, and a bar graph that displays the grade distribution of the class.
Scene	A unique environment that allows the user to interact with the user interface. When switching between scenes, variables and game objects within unity are unloaded and the next scene's game objects are loaded.
FAQ Page	A scene that both instructors and students have access to. Both instructors and students may use this page to find answers to questions are often asked. Instructors are also able to create and delete questions/answers from this page.
About Page	A scene that both instructors and students have access to. This scene provides insight regarding the lab course itself. More specifically information about the lab manual, instructor contact information, and course number.
Panel	A panel is a popup user interface that overlays the current scene.
Event (action in unity)	An "event" in this case is a state at which the program is running. More specifically the "event" could be something as simple as changing state from closed menu to a drop down menu
GameObject(unity element)	GameObjects are components in a project with specific properties and functions. The unity framework is focused facilitating the interaction between GameObjects.

System Requirements

Enumerated Functional Requirements:

General Requirements:

Number	PW	Requirements
REQ 1	4	The user shall be able to login as a student or a teacher, and this should result in them having access to the correct data set.

REQ 2	2	The teacher shall be able to set a lab to be worked on by a certain time.
REQ 3	5	The users shall be able to perform every lab normally performed in the course
REQ 4	4	The labs shall assign credit based on correctly placed components, correct logic, and correct component connection.
REQ 5	3	The labs will allow for manual re-grading by the teacher in the event of an error.
REQ 6	1	The system will allow for students to request a regrade with a specialized chat message
REQ 7	3	The system shall allow for the teacher to view all grades of students, but will only allow students to view their own grades
REQ 8	2	The system shall require students to use specified components to complete each lab
REQ 9	1	The system shall allow students to re-do labs after the due date for practice
REQ 10	4	The system will allow for the user to place multiple wires, without having to reselect wire each time.

Chat Subsystem Requirements:

Number	**PW**	**Requirements**
REQ 11	5	The system shall allow multiple lab instructors to simultaneously interact with students.
REQ 12	1	The system shall provide indicators to the lab instructor allowing to determine if an instructor is online
REQ 13	3	The system shall provide a chat history between an instructor and student
REQ 14	4	The system shall allow the instructor to make announcements to multiple students.
REQ 15	3	The system shall allow students to share screenshots of the lab with the instructor.
REQ 16	1	The system shall allow students to “screen-share” their lab with their

		instructor
REQ 17	2	The system shall allow students to utilize a keyboard of symbols (Ω, μ, etc.)

Class Statistics Requirements:

Number	**PW**	**Requirements**
REQ 18	3	The system shall allow an instructor to access multiple classes
REQ 19	5	The system shall show grades for all parts/topics of a lab
REQ 20	4	The system shall provide a recommended topic to review based on student grades
REQ 21	4	The system shall provide feedback to the instructor that students are scoring relatively low or high
REQ 22	3	The system shall provide a histogram of student score distribution
REQ 23	5	The system shall allow calculate the average, median, lower quartile range, upper quartile range, and standard deviation of the set of grades.
REQ 24	1	The system shall determine if the grade distribution is normal for cheat detection

Account System Requirements:

Number	**PW**	**Requirements**
REQ 25	5	The system will allow either an instructor or student to log in using a specific username and password set by said instructor and student.
REQ 26	4	The system will allow for students to keep track of grades received from every lab.
REQ 27	4	The system will allow for instructors to keep track of ALL students grades, as well as display class statistics like averages.
REQ 28	3	This system will allow for students and instructors to be able to change/recover password for their login.
REQ 29	2	This system will allow for specification of whether account is for a student or instructor when logged in.

REQ 30	4	The system will allow an administrative user to go back to the main menu from each sub-menu.
REQ 31	4	The system will allow the user to return to the main menu from the sandbox mode.
REQ 32	4	The system will allow students to join different classes, depending on their section
REQ 33	4	The system will allow teachers to create different classes for students to join
REQ 34	4	The system should time out an account if too many login attempts have occurred
REQ 35	4	The system should log out an account if inactive for more than a specified amount.

Smart Grading/Mini-Quizzes Requirements:

Number	**PW**	**Requirements**
REQ 36	3	The system will allow students and teachers to view the students' grades.
REQ 37	1	The system will allow teachers to view the graded virtual labs and edit the students' grades manually.
REQ 38	4	The system will analyze the students' virtual labs and award partial credit accordingly.
REQ 39	3	The system will calculate students' course grade using lab work, pre-lab, and miniquiz grades.
REQ 40	1	The system will allow teachers to assign mini-quizzes, pre-labs, and lab work.
REQ 41	2	The system will allow students to complete the mini-quizzes.
REQ 42	1	The system will enforce a time limit while students are taking the mini-quiz.
REQ 43	1	The system will store the student's grade in the account.
REQ 44	2	The system will not allow students to take the quiz after the deadline.

Enumerated Non-Functional Requirements:

General Requirements:

Number	PW	Requirements
REQ 45	2	The system should give feedback for every right or wrong action
REQ 46	3	The system should notify users of direct messages
REQ 47	1	The system should notify user when in screen sharing
REQ 48	2	The system should keep logs of all chats
REQ 49	3	The system should notify all students when teacher sends announcement
REQ 50	3	The system should notify teacher when student changes a section
REQ 51	2	The system will only grade labs done the first time (practice will not effect prior grades)
REQ 52	1	The teacher will be sent a different colored notification when a student asking to screen share

Class Statistics Requirements

Number	PW	Requirements
REQ 53	4	Provide a graph of the distribution of grades
REQ 54	4	Provide the min, max, and average of each class's grades
REQ 55	2	Provide a breakdown of each student's total lab grade

Account System Requirements:

Number	PW	Requirements
REQ 56	1	Users are provided feedback when logging in (login in failed, login successful, too many login attempts, timed out)
REQ 57	3	The system should display section number associated with user's input section
REQ 58	3	Graded labs are opened to users only during time frame teacher sets for each section

REQ 59	3	Answers for each user are saved once checked for correctness by system, to prevent cheating
REQ 60	5	Once time graded lab time frame has expired, the system will not allow students to begin the graded lab, and will open practice lab to students
REQ 61	3	All users will be sent a notification with 30 and 10 minutes left in graded lab time frame
REQ 62	5	All students in a section will have a visible timer for their graded lab time

Smart Grading/Mini-Quizzes Requirements:

Number	**PW**	**Requirements**
REQ 63	3	The students can begin the mini-quiz any time before the deadline.
REQ 64	3	The mini-quiz will end after time expires or after the deadline, whichever comes first.
REQ 65	4	Program will show logical process of determining output of logic functions.
REQ 66	5	Students will receive partial credit for each correct logic truth table component.
REQ 67	1	The grades will not be displayed to students until after the deadline to prevent possible cheating.
REQ 68	1	Administrator can review grades before releasing the grades for students to view.
REQ 69	2	Students will have access to practice quizzes.

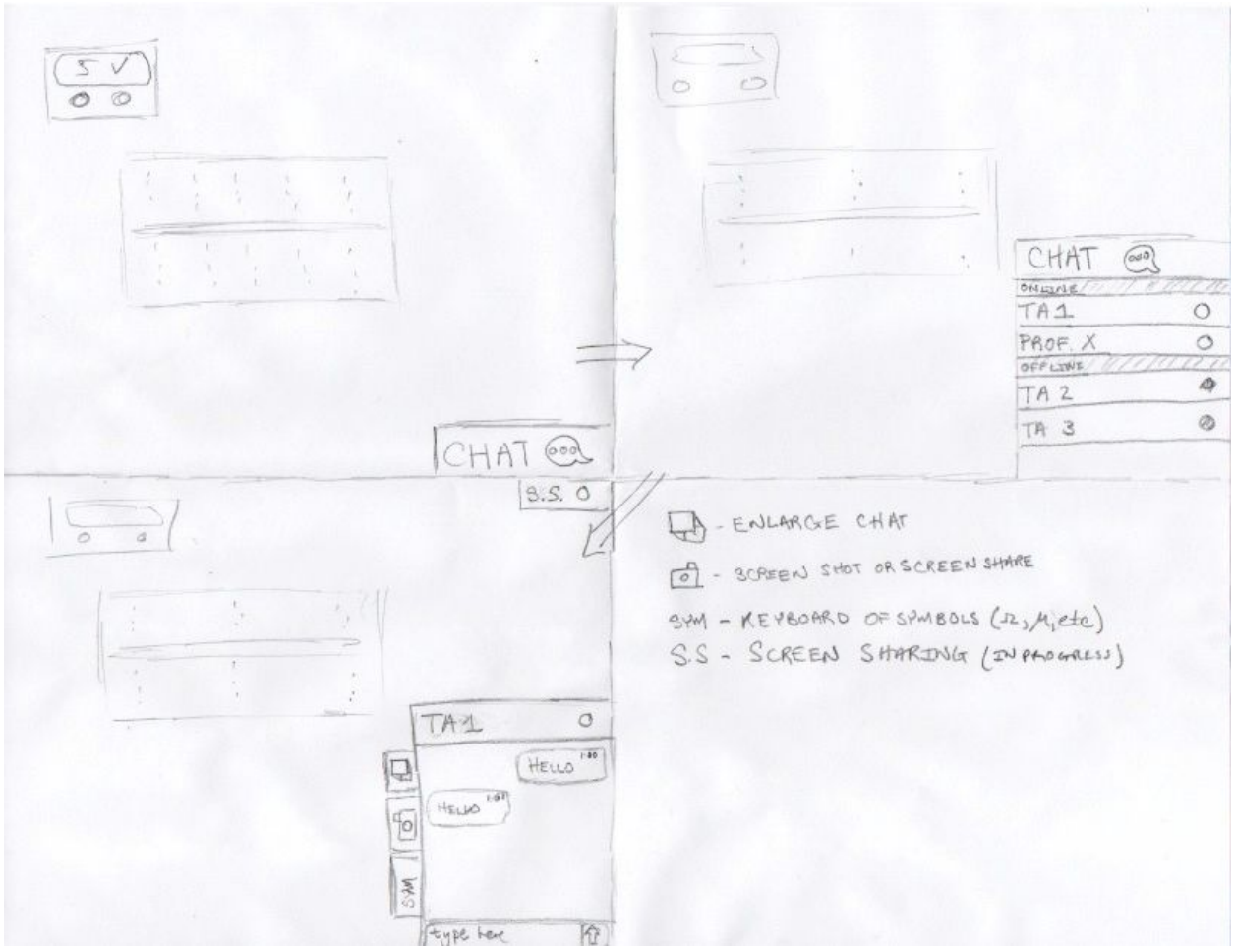
FAQ requirements:

Number	**PW**	**Requirements**
REQ 70	3	The students can click and view the FAQ page
REQ 71	5	Instructor can view FAQ page
REQ 72	3	The database takes most recent page from the instructor and stores it

REQ 73	2	Administrator can add or remove questions

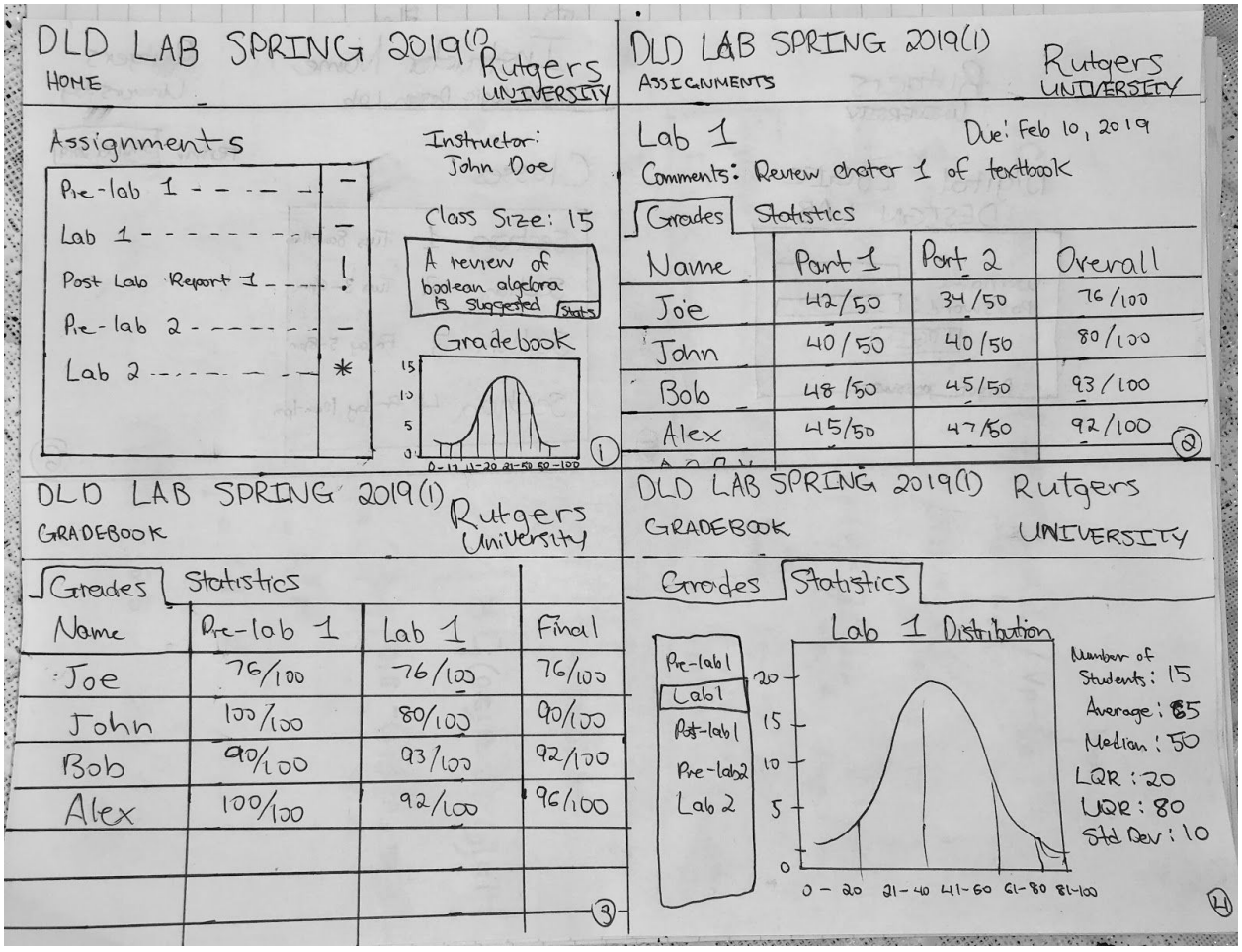
On-Screen Appearance Requirements:

Chat Subsystem UI (Rough Sketch):



This sketch shows 3 instances of the chat system. The first (top left corner) shows the placement of the chat and how it appears in the bottom right corner of the screen when first entering the lab. The second (top right corner) shows what appears when first clicking the chat. It opens up to a display of the various instructors with which you can chat with, as well as whether or not they are online. The third shows what appears when beginning a chat with one of the instructors as well as various tools such as a keyboard for symbols, a camera option to take a screenshot or share your screen with the instructor, as well as an enlarge button to enlarge the chat. In the top right corner of the screen, it shows that screen sharing is in progress.

Class Statistics Subsystem UI (Rough Sketch):



This sketch shows the class statistics subsystem with six different snapshots. Users, student or instructor, will log in through a page shown by snapshot 5. Once logged into the system, the instructor can select a term from a drop down box, or select a section in snapshot 6.

After selecting a section, the instructor will be redirected to a class homepage illustrated by snapshot 1. On the home page, instructors have a view of their assignments in a scrollable box. The assignments on this page will be complemented by symbols of three types: -, !, and *. The "-" symbol will represent a normal assignment with no alerts. The "!" symbol represents an assignment with relatively low scores. The "*" symbol represents an assignment with relatively high scores. On snapshot 1, we can also see a box the gives the instructor a recommended task. The is also a clickable gradebook with a snapshot of the overall grade distribution of the class. Snapshot 2 represents the assignment page after an assignment is selected from the home page. Snapshot 3 represents the gradebook page after gradebook is selected from the home page. On snapshots 2 and 3, the instructor has different tabs named grades and statistics. The grades tab will show each student's overall assignment grade on the gradebook page. The grades tab will show each student's sub-assignment score for a specific assignment on the assignments page. Snapshot 4 shows what the instructor will see when they click on statistics in either the gradebook or assignment page. Statistics will show the distribution along with other calculated variables. The instructor will have the option to select any assignment or sub-assignment to show statistics for said assignment.

Functional Requirements Specification

Stakeholders:

The current stakeholders of our system include Students, Instructors (Professors/Teachers), and College institutions as a whole.

Student: The primary goal for the student is to have the ability to complete assigned lab experiments and the ability to be able to communicate any questions they have to their assigned lab instructor or professor via chat module. They should also be responsible for taking the pre/post assessments to ensure they are ready for the experiments. Our software would benefit them because it will prevent the hassle of trying to figure out if their circuit isn't working because of something they did wrong or due to faulty equipment.

Instructor (Professor, Teacher): The teacher will need to be able to enhance the learning of the students by viewing statistics and grades to allow them to gauge where students fall behind. They will be able to communicate with the students during their time on the system, check the grades for each students that have completed the labs, and be able to manage any new or existing users. With our software they will also benefit from being able to answer lab queries without having to be at the lab with the student. If the students have questions they can quickly reference their lab to see where they are struggling, rather than the professor having to visit each lab in person.

College institutions: Colleges will not need to spend much money on providing students with the required equipment to perform the labs, and do not have to worry about upgrading all the equipment every so often because this could be implemented through software updates.

Actors and Goals:

- ❏ Student (Initiating): to perform digital logic experiments and get results at the end of each labs using logic equipment such as encoders, decoders, multiplexers … etc.

- ❏ Instructor(Initiating): to present assignments (labs) and obtain grades for all the students in the class. Also they should be able to have live interactions with the students through chat. Finally they should be able to modify assignments and grades.

- ❏ Database (Participating): SQL relational database that keeps all the data, including usernames, passwords, chat history, statistics, and grades.

- ❏ Chat (Participating): to facilitate communication between Actors, and store chat history in the database. To also have the ability to screen share so the Instructors can better assist with the assignment.

- ❏ Camera (Participating): Facilitates the cheat detector to insure that students are doing their own assignments.

Use Cases:

Casual Description:

Use Case	Name	Description	Requirement
UC-1	Login	Allows the actors to login and show them their options depending on whether they are a student or instructor	REQ1, REQ 25, REQ 29
UC-2	InstructorLab Control	Allows instructors to modify and alter grading of the labs	REQ2, REQ5, REQ7,
UC-3	ManagementEquipment	Fix equipment placement issues from the previous group. Allows students to perform the lab normally with the equipment provided in the dropdown menu	REQ4, REQ3, REQ8, REQ10

UC-4	Continuous Access	Simulation availability is 24/7 so that users are allowed to return to completed labs for practice	REQ9
UC-5	ChatSystem	Allows complete, real time interaction between student and instructor for assistance with the material.	REQ 6, REQ 11, REQ 12, REQ 13, REQ 14, REQ 15, REQ 16, REQ 17
UC-6	Assignment Sub-Score Statistic	Allows the instructor to view the score composition of each assignment. Notable statistics will be displayed for the instructor to analyze the data.	REQ 18, REQ 19, REQ 22, REQ 23, REQ 53, REQ 54, REQ 55
UC-7	Gradebook Statistic	Allows the instructor to view the overall score of each student for each assignment. Notable statistics will be displayed for the instructor to analyze the data.	REQ 22, REQ 23, REQ 53, REQ 54, REQ 55, REQ 36, REQ 37, REQ 27, REQ 43
UC-8	Multiple Classes	Allows the instructor to pick any number of classes the he/she may have during the term. The class page will give a suggested topic to review and high/low score indicators.	REQ 18, REQ 19, REQ 20
UC-9	LoginInquiries	Allows the user to retrieve their password if they have forgotten it, either via email or by answering a few preset questions answered at the creation of their account.	REQ 28, REQ 34, REQ 56
UC-10	CreateClasses	Allows the instructors to create different classes for students in different sections and allows them to move students around as necessary.	REQ 32, REQ 33, REQ 50
UC-11	LogOut	Logs out the user. Depending on current page, if timed out of activity, returns to main menu. If timed out on main menu, will log out completely.	REQ 2, REQ 35, REQ 42, REQ 60, REQ 64
UC-12	CurrentStatus	Allows the students to see how they are currently doing in class. With all their assignment grades summed up to give an idea of where their grade lies.	REQ 7, REQ 19, REQ 23, REQ 26, REQ 27, REQ 36, REQ 39, REQ 43, REQ 54,

			REQ 55, REQ 58, REQ 67
UC-13	Grading	The software automatically grades the students, but allows for the teacher to manually adjust any grades if an error occurred in the process.	REQ 4, REQ 5, REQ 6, REQ 7, REQ 19, REQ 22, REQ 23, REQ 37, REQ 38, REQ 45, REQ 53, REQ 54, REQ 55, REQ 66
UC-14	CreateAssignments	Allows the instructor to create mini quizzes, pre-labs to test students understanding.	REQ 40, REQ 41
UC-15	Notifications	Notifies user of whatever event has occurred.	REQ 6, REQ 12, REQ 14, REQ 21, REQ 42, REQ 45, REQ 46, REQ 47, REQ 49, REQ 50, REQ 52, REQ 56, REQ 61
UC-16	History	Keeps the history of major actions taken, like logging in, attempts at logging in, chat history, attempts at labs, etc.	REQ 1, REQ 4, REQ 5, REQ 9, REQ 13, REQ 22, REQ 27, REQ 34, REQ 41, REQ 43, REQ 48, REQ 51, REQ 56, REQ 59, REQ 65
UC-17	Quizzes	Allows students to test their understanding of the material, and allows them to acquire extra practice.	REQ 63, REQ 64, REQ 65, REQ 69
UC-18	ScreenCapture	Allows students to send pictures of their screen through the chat system and allows the system to record the actions the student takes during their experiment.	REQ 15, REQ 16, REQ 47, REQ 52
UC-19	FAQs	Allows both parties to view the FAQ page, Instructors can modify the page to reflect most important questions	REQ 70, REQ 71, REQ 72, REQ 73

(updated):

Use cases implemented by the time of Final Demo	Use cases not implemented for Final Demo
UC-1 (login)	UC-14 (Create Assignments)
UC-2 (Instructor Lab Control)	UC-16 (History)
UC-3 (Manage Equipment)	
UC-4 (Continuous Access)	
UC-5 (Chat System)	
UC-6 (Assignment Sub-Score Statistic)	
UC-7 (Gradebook Statistic	
UC-8 (Multiple Classes)	
UC-9 (Login Inquiries)	
UC-10 (Create Classes)	
UC-11 (Log Out)	
UC-12 (Current Status)	
UC-13 (Grading)	
UC-15 (Notifications)	
UC-17 (Quizzes)	
UC-18 (Screen Shot)	
UC-19 (FAQ's)	

Use Case descriptions(for not used cases):

UC-14(Create Assignments)-

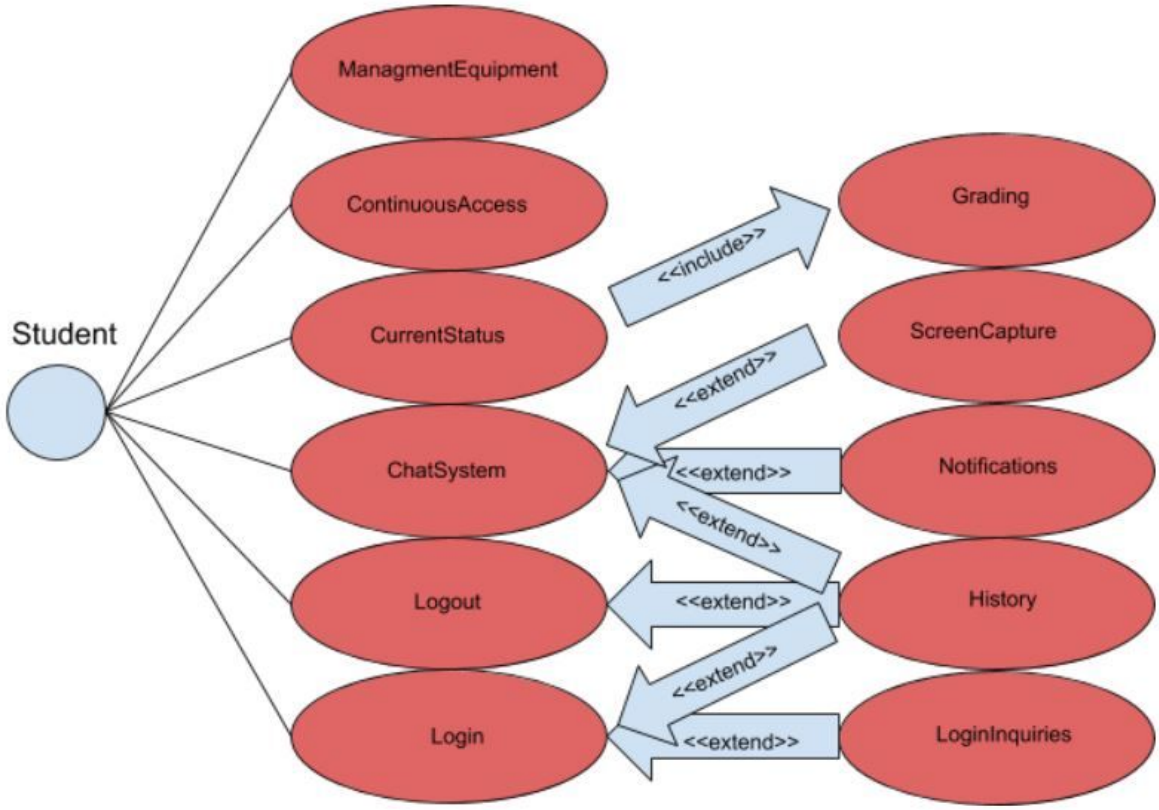
Create assignments was intended to be a feature to help Instructors create a personalized quiz or prelab to allow students to develop class skills prior to performing the lab. The only reason this entire feature was not fully implemented is because the quizzes and prelabs didn't have a specific interface to actually go in and change labs. Currently the prelabs and quizzes are all hard coded in because we simply ran out of time to implement this feature, but it definitely would not have posed much of a threat to create if a future group wanted to push this new feature.

UC-16(History)-

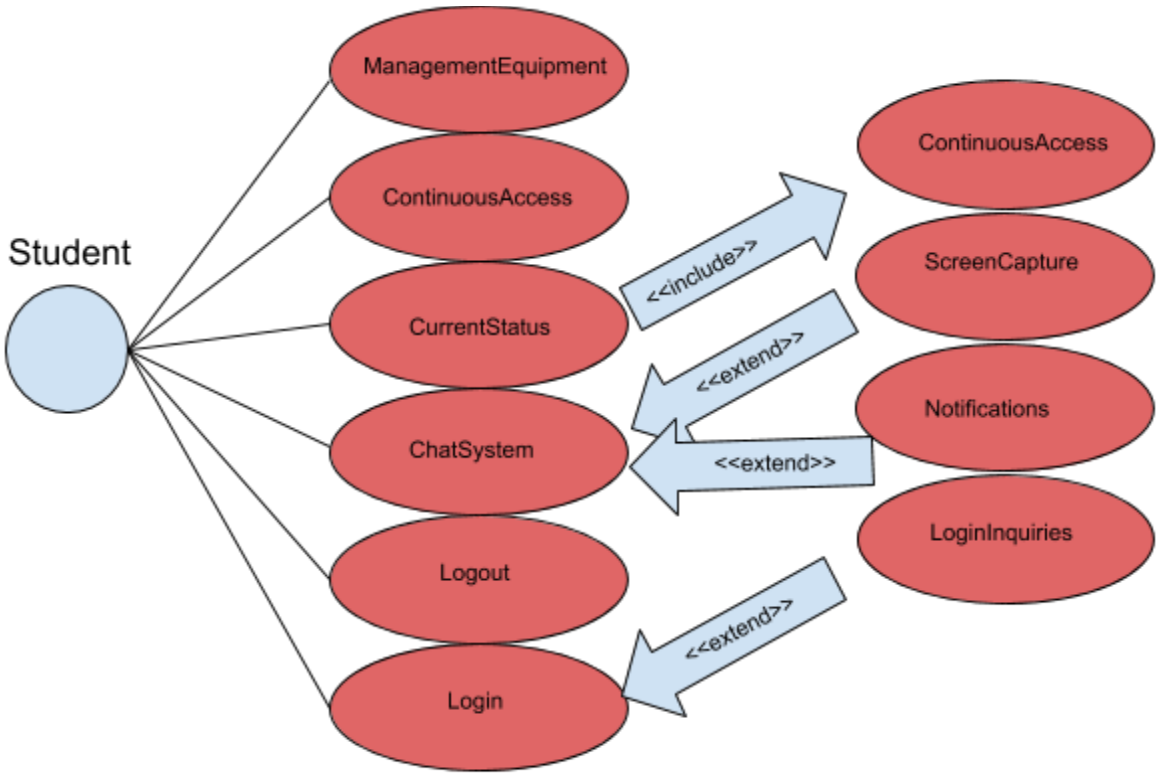
History was intended to be a feature to allow Admins more access of features allowed within the lab like login, logout, chat history, and lab details such as saving lab submission answers for each student. A few of these components actually are implemented in our code but since the lab detail portion of saving lab submission answers was not fulfilled I deem this Use Case partially not used. This feature would be very important for regrading students who had tech issues or general problems with the lab and needed a regrade. This regrading option is still available for Instructors to use but with this History feature implemented further they would have definite proof if a student completed the lab correctly and was graded wrong or if they just want points back because they were just wrong.

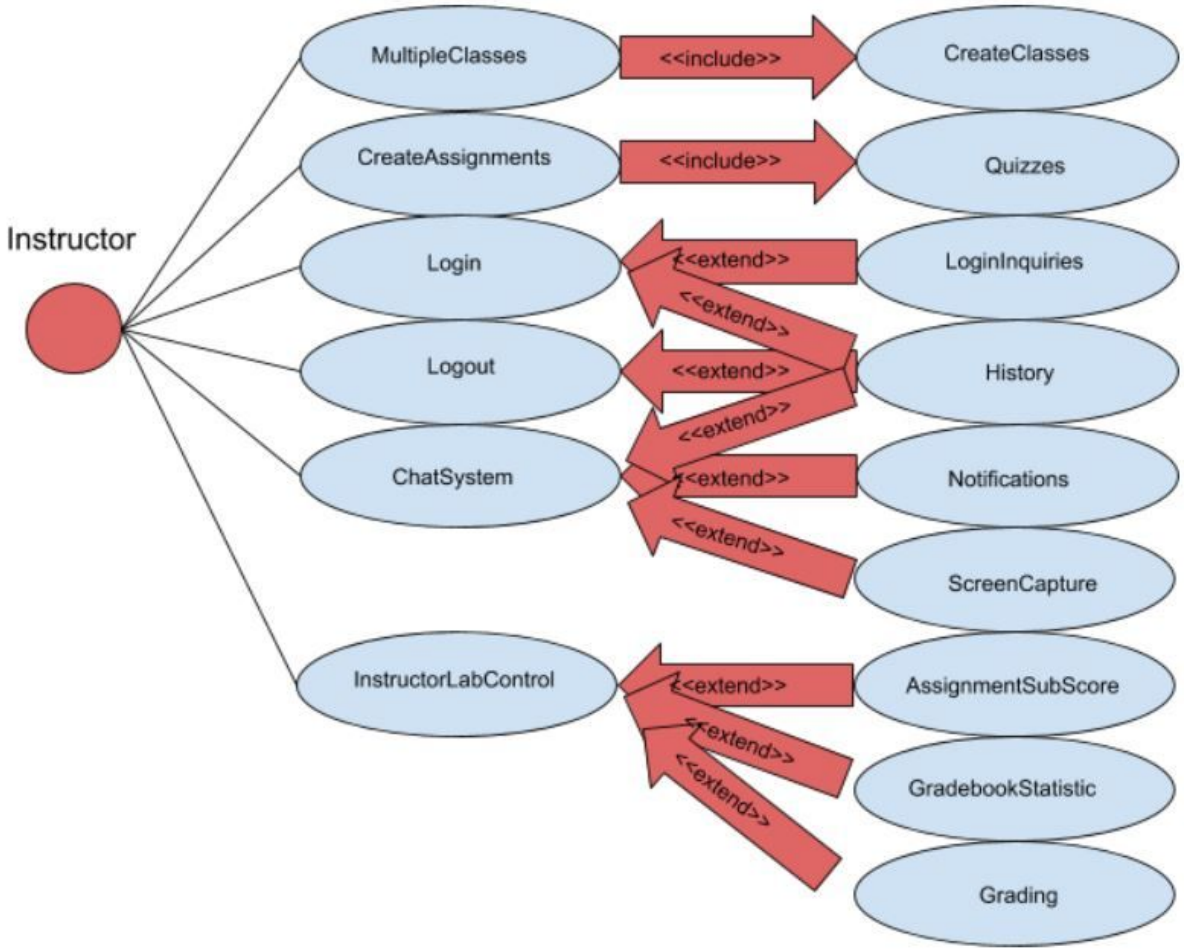
Use Case Diagram:

The following diagrams show the use case diagrams of the student, instructor, and the chat. It shows how each actor interacts with specific use cases related to them as well as how those use cases interact with other use cases.

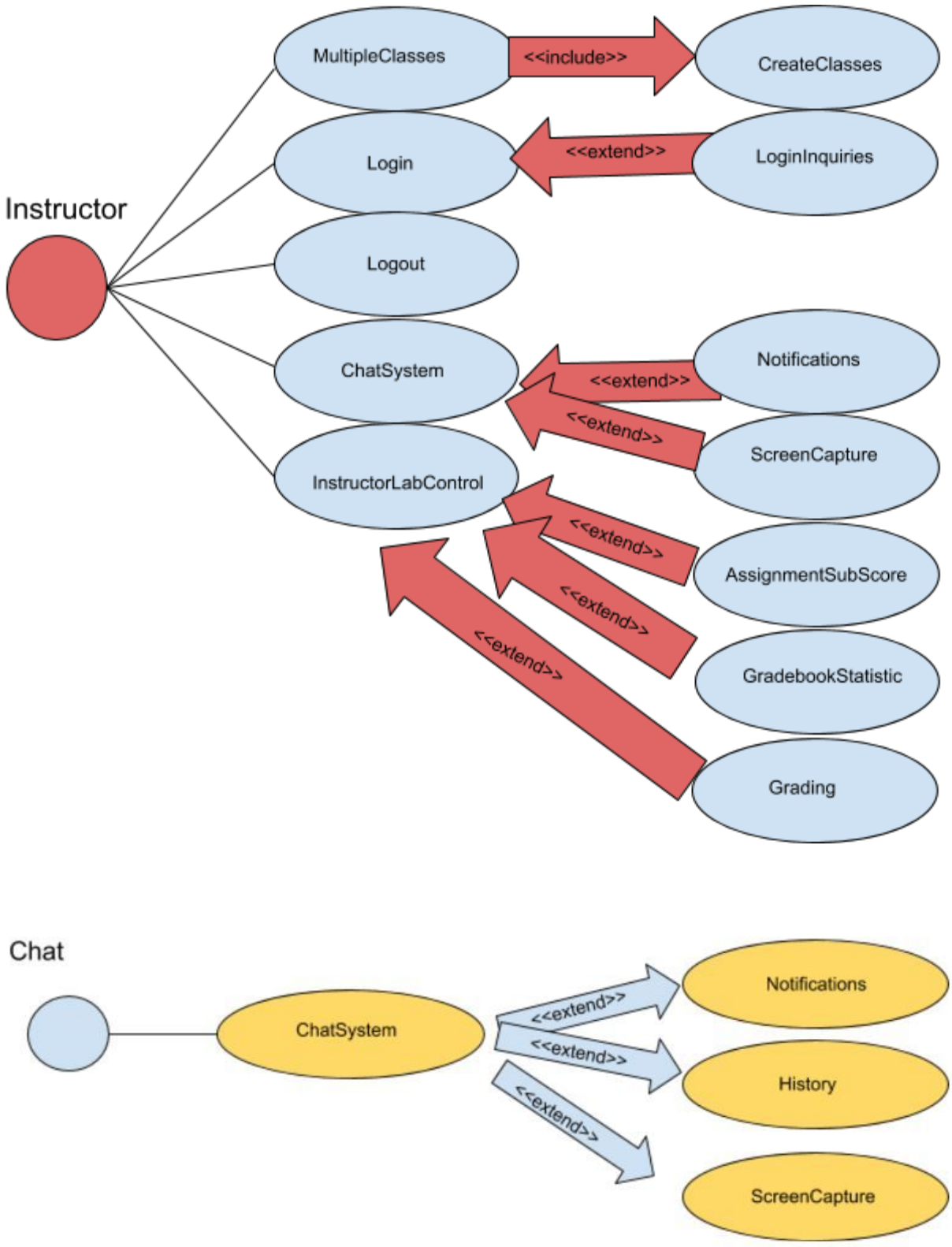


(Updated): Use Case Diagram

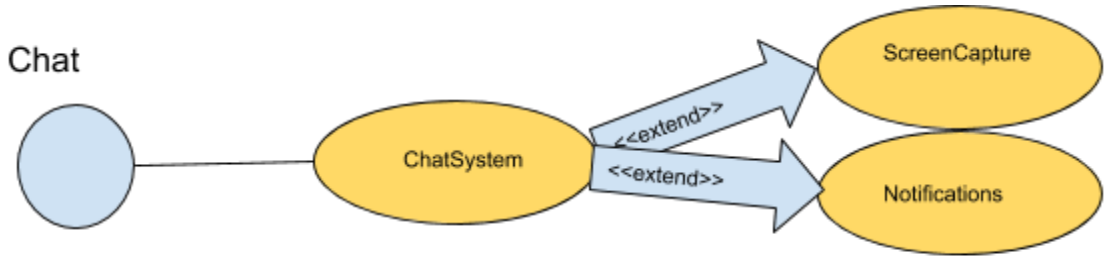

Student
ManagementEquipment
ContinuousAccess
CurrentStatus
ChatSystem
Logout
Login
<<include>>
<<extend>>
<<extend>>
<<extend>>
ContinuousAccess
ScreenCapture
Notifications
LoginInquiries



(Updated): Use Case Diagram



(Updated): Use Case Diagram



Traceability Matrix:

	PW	UC-1	UC-2	UC-3	UC-4	UC-5	UC-6	UC-7	UC-8	UC-9	UC-10	UC-11	UC-12	UC-13	UC-14	UC-15	UC-16	UC-17	UC-18	UC-19
REQ1	4	X															X			
REQ2	2		X									X								
REQ3	5			X																
REQ4	4			X										X			X			
REQ5	3		X											X			X			
REQ6	1					X								X		X				
REQ7	3		X										X	X						
REQ8	2			X																
REQ9	1				X												X			
REQ10	4			X																
REQ11	5					X														
REQ12	1					X										X				
REQ13	3					X											X			
REQ14	4					X										X				
REQ15	3					X													X	
REQ16	1					X													X	

REQ17	2					X													
REQ18	3						X		X										
REQ19	5						X		X				X	X					
REQ20	4								X										
REQ21	4															X			
REQ22	3						X	X						X			X		
REQ23	5						X	X					X	X					
REQ24	1																		
REQ25	5	X																	
REQ26	4												X						
REQ27	4												X				X		
REQ28	3									X									
REQ29	2	X																	
REQ30	4																		
REQ31	4																		
REQ32	4										X								
REQ33	4										X								
REQ34	4									X							X		
REQ35	4											X							
REQ36	3							X					X						
REQ37	1							X						X					
REQ38	4													X					
REQ39	3												X						
REQ40	1														X				
REQ41	2														X		X		
REQ42	1											X				X			
REQ43	1							X					X				X		

REQ44	2																		
REQ45	2												X		X				
REQ46	3														X				
REQ47	1														X			X	
REQ48	2															X			
REQ49	3														X				
REQ50	3										X				X				
REQ51	2															X			
REQ52	1														X			X	
REQ53	4						X	X					X						
REQ54	4						X	X					X						
REQ55	2						X	X					X						
REQ56	1									X					X	X			
REQ57	3																		
REQ58	3												X						
REQ59	3															X			
REQ60	5											X							
REQ61	3														X				
REQ62	5																		
REQ63	3																X		
REQ64	3											X					X		
REQ65	4															X	X		
REQ66	5												X						
REQ67	1											X							
REQ68	1																		
REQ69	2																X		
REQ70	3																		X

REQ71	5																		X
REQ72	3																		X
REQ73	3																		X

(Updated): Traceability matrix

	PW	**UC -1**	**UC -2**	**UC -3**	**UC -4**	**UC -5**	**UC -6**	**UC -7**	**UC -8**	**UC -9**	**UC -10**	**UC -11**	**UC -12**	**UC -13**	**UC -15**	**UC -17**	**UC -18**	**UC -19**
REQ1	4	X																
REQ2	2		X									X						
REQ3	5			X														
REQ4	4			X										X				
REQ5	3		X											X				
REQ6	1					X								X	X			
REQ7	3		X										X	X				
REQ8	2			X														
REQ9	1				X													
REQ10	4			X														
REQ11	5					X												
REQ12	1					X									X			
REQ13	3					X												
REQ14	4					X									X			
REQ15	3					X											X	
REQ16	1					X											X	
REQ17	2					X												
REQ18	3						X		X									
REQ19	5						X		X				X	X				
REQ20	4								X									
REQ21	4														X			

REQ22	3						X	X						X				
REQ23	5						X	X					X	X				
REQ24	1																	
REQ25	5	X																
REQ26	4												X					
REQ27	4												X					
REQ28	3									X								
REQ29	2	X																
REQ30	4																	
REQ31	4																	
REQ32	4										X							
REQ33	4										X							
REQ34	4									X								
REQ35	4											X						
REQ36	3							X					X					
REQ37	1							X						X				
REQ38	4													X				
REQ39	3												X					
REQ40	1																	
REQ41	2																	
REQ42	1											X			X			
REQ43	1							X					X					
REQ44	2																	
REQ45	2													X	X			
REQ46	3														X			
REQ47	1														X		X	

REQ48	2																	
REQ49	3														X			
REQ50	3										X				X			
REQ51	2																	
REQ52	1														X		X	
REQ53	4						X	X						X				
REQ54	4						X	X					X	X				
REQ55	2						X	X					X	X				
REQ56	1									X					X			
REQ57	3																	
REQ58	3												X					
REQ59	3																	
REQ60	5											X						
REQ61	3														X			
REQ62	5																	
REQ63	3															X		
REQ64	3											X				X		
REQ65	4															X		
REQ66	5													X				
REQ67	1												X					
REQ68	1																	
REQ69	2															X		
REQ70	3																	X
REQ71	5																	X
REQ72	3																	X
REQ73	3																	X

Fully-Dressed Description:

Use Case UC-1	Login
Requirements	REQ1, REQ 25, REQ 29
Initiating Actor	Student or Instructor
Actor's Goals	To be able to put an account name and password in, be recognized as either a student or an instructor, and be placed into the proper account type, giving them access to what is needed to complete whatever task that they have at hand.
Preconditions	Both the account name and password must be valid and in the database before allowing the account to gain access to the system.
Postconditions	User should have access to the labs, grades, and other valuable resources for such labs.
Flow of Events Student: →The student enters in proper username and password and clicks the login button ←System recognizes that the students credentials are valid →The student gains access to their labs (both completed and due), past labs, extra help, quizzes, a chat system, and a grade book. Instructor: →The instructor enters proper username and password and clicks the login button ←The system will recognize that the credentials are that of a instructor, thus giving the instructor more access to things like inputting/adjusting grades. →The instructor can now access all students labs, grades can be modified for the students, and the instructor is notified of any messages sent time him/her via a chat system.	

Use Case UC-2	InstructorLab Control

Requirements	REQ2, REQ5, REQ7
Initiating Actor	Instructor
Actor's Goals	To be able to look at the students lab and be able to make adjustments to grades accordingly.
Preconditions	Instructor account must be logged in. Students must already have already completed at least one lab to be able to modify grades.
Postconditions	Both the instructor and the student's grade books will be updated. This includes all statistics based on the student's current grades as well.

Flow of Events
→The instructor enters proper username and password and clicks the login button
←The system will recognize that the credentials are that of a instructor, thus giving the instructor more access to things like inputting/adjusting grades.
→The instructor can now access all students labs, grades can be modified for the students, and the instructor is notified of any messages sent time him/her via a chat system.
←The system gives the instructor a notification that student has sent them a message
→Instructor may receive student concern about lab grading being incorrect or for other reasons the student could not finish the lab entirely. Instructor will click on the gradebook button.
←The system updates page with student roster and corresponding lab grades.
→Instructor presses edit button.
←The system updates page giving access to manually overriding grades given.
→Instructor presses save button
←The system updates both the instructor's gradebook and the student's gradebook, also updating all statistics for the class average and students grade.

Use Case UC-3	ManagementEquipment
Requirements	REQ4, REQ3, REQ8, REQ10
Initiating Actor	Student
Actor's Goals	To be able to use the equipment needed for the lab a bit easier in regards to not having to click as much or having to deselect certain things in order to Re-select the same exact item. This was a problem developed in last years group.
Preconditions	Student must be logged in and have began a lab.
Postconditions	Student will accomplish lab with more ease than previous versions.

Flow of Events →The student will log into their account ←The system will recognize the student account information and notifiy student if there is a lab required for the student to do →The student will click on the lab that needs to get done. ←The system will send student to the lab page which consists of a bread board, several buttons to be used to return to main menu or get rid of items, and a drop down menu that consists of all the needed equipment for the lab. →The student will be able to create wires and plug them in wherever they need without having to constantly deselect and reselect the wire option in the drop down menu. Student will also be able to place down multiples of an item if need be/

Use Case UC-4	Continuous Access
Requirements	REQ 9
Initiating Actor	User
Actor's Goals	To be able to go back and use the application to redo past labs for practice if need be.
Preconditions	Logged into the system and is already passed the deadline of submitting the lab
Postconditions	User gains some insight by being able to practice the lab outside of the actual lab due date. This can help with upcoming quizzes,exams, and general knowledge on topics.
Flow of Events →The user logs in and begins a lab experiment at a time after the due date of the lab. ←System notifies user that the lab will not count for any credit and is strictly for practice →The student clicks on the "i understand" button and can now do the re do the lab for practice.	

Use Case UC-5	ChatSystem
Requirements	REQ 6, REQ 11, REQ 12, REQ 13, REQ 14, REQ 15, REQ 16, REQ 17
Initiating Actor	Student or Instructor
Actor's Goals	To initiate a chat between actors that will allow students to interact with

	instructors. To allow students to obtain general instructions regarding the lab from the instructors. To allow students to screen share with instructors for real time help constructing a circuit.
Preconditions	Student or instructor must be logged in. Students must be actively doing a lab at the time an instructor is available to chat.
Postconditions	A history of the chat should be stored in the database for a certain time.
Flow of Events Student: →The student logs in and begins a lab experiment ←The chat button displays in the bottom right corner of the display →The student clicks on the chat button and it opens a cat box that will show which instructor is online and available to chat. →The student sends a message and the instructor responds ←A short history of the chat is also displayed Instructor: →The instructor logs in and see which student is actively doing an experiment. ←The the system will display a chat option. →The instructor clicks on the chat button and it opens a cat box that will show which student is online and available to chat.	

Use Case UC-6	Assignment Sub-Score Statistic
Requirements	REQ 18, REQ 19, REQ 22, REQ 23, REQ 53, REQ 54, REQ 55
Initiating Actor	Instructor
Actor's Goals	To gather data on how well the students are understanding the topics that are being taught. To view the subscores of each assignment, and use statistical data to better analyze each student's scores.
Preconditions	Lab instructor must be logged in, and must already have classes being taught.
Postconditions	Any added/edited data must be saved.
Flow of Events Instructor: →The instructor logs in and selects a class that he/she is teaching ←The system will display a class homepage with a list of assignments and a gradebook	

→The instructor clicks on an assignment name.
←There are two tabs the instructor may view “Grades” or “Statistics”. The system will always default to the “Grades” tab, which display a spreadsheet of each students subscore for the assignment.
→The instructor selects the statistics tab.
←The graph of the class’ grade distribution is displayed along with the calculated average and other variables. A list of the sub-topics of the assignment is also displayed as options for the instructor to select from.
→The instructor selects a different sub-topic from the list
← The system switches to the set of data for the selected sub-topic. A graph of the grade distribution is displayed along with other statistical variables.

Use Case UC-7	Gradebook Statistic
Requirements	REQ 22, REQ 23, REQ 53, REQ 54, REQ 55
Initiating Actor	Instructor
Actor’s Goals	To gather data on how well the students are understanding the topics that are being taught. To view the scores of each student’s assignment, each student’s final grade, and their distributions.
Preconditions	Lab instructor must be logged in, and must already have classes being taught.
Postconditions	Any added/edited data must be saved.

Flow of Events
Instructor:
→The instructor logs in and selects a class that he/she is teaching
←The system will display a class homepage with a list of assignments and a gradebook.
→The instructor clicks on gradebook
←There are two tabs the instructor may view “Grades” or “Statistics”. The system will always default to the “Grades” tab, which display a spreadsheet of each students for the corresponding assignment.
→The instructor selects the statistics tab.
←The graph of the class’ grade distribution is displayed along with the calculated average and other statistical variables. A list of all the assignments are also displayed as options for the instructor to select from including a final grade option.
→The instructor selects a different assignment from the list
← The system switches to the set of data for the selected sub-topic. A graph of the grade distribution is displayed along with other statistical variables.

Use Case UC-8	Multiple Classes
Requirements	REQ 18, REQ 19, REQ 20
Initiating Actor	Instructor
Actor's Goals	To view the grades of his/her students. To view the grades separated by class.
Preconditions	Lab instructor must have already created an instructor account.
Postconditions	Any added/edited data must be saved. Past courses from different terms will be viewable.

Flow of Events
Instructor:
→The instructor logs in with a valid username and password
←The system will display a course page with a list of all the classes that he/she is teaching for the current term. Another option to pick from past terms that the instructor has taught is also selectable.
→The instructor clicks on term
←The system provides a drop down menu for the instructor to choose from. Only the terms for which he/she has taught will be displayed
→The instructor selects the current term
←The system displays all the classes for the selected term.
→The instructor selects a class from the list
← The system directs the instructor to the selected class' homepage

Use Case UC-9	Login Inquiries
Requirements	REQ 28, REQ 34, REQ 56
Initiating Actor	Any user
Actor's Goals	To be able to change their password in the case that they have forgotten it or if they think their account has been compromised.
Preconditions	User must have an account, must have provided an email address, and must have set security questions.
Postconditions	Must replace the old password with the new one.

Flow of Events User: →The user clicks on a button to retrieve his password. ←The system will display a page asking for the username or email address. →The user provides the username or email address. ←The system will ask one or more security questions that the user had set when the account was created. →The user answers the question(s). ←The system sends an email to the user with a link to change the password. →The user selects a new password. ← The system replaces the old password with the new one.

Use Case UC-10	Create Class
Requirements	REQ 32, REQ 33, REQ 50
Initiating Actor	Instructor
Actor's Goals	To create new classes/sections to partition students
Preconditions	Instructor must have made an account and have students
Postconditions	Have Students divided into multiple classes
Flow of Events Instructor: →The instructor clicks a button to begin creating a new class ←The system displays a page with a place to title the class, and lists the students currently with an instructor so he may begin partitioning them into classes. →The instructor selects all the students he needs to be in the new class. →The instructor confirms he wishes to make this new class with these students. ←The system migrates all the students selected into the new class, provides a notification to those students, and gives the students access to the coursework in the new class.	

Use Case UC-11	LogOut
Requirements	REQ 2, REQ 35, REQ 42, REQ 60, REQ 64
Initiating Actor	Any user
Actor's Goals	To be able to logout and allow another user to login afterwards.

Preconditions	User must be logged into an account.
Postconditions	Must require another user to enter a username and password to use the software.
Flow of Events User: →The user clicks on a button to logout. ←The system will display a message asking if the user is sure he wants to logout. →The user confirms he wishes to logout. ←The system will return back to the initial menu, and will require users to sign in to use the software.	

Use Case UC-12	Current Status
Requirements	REQ 7, REQ 19, REQ 23, REQ 26, REQ 27, REQ 36, REQ 39, REQ 43, REQ 54, REQ 55, REQ 58, REQ 67
Initiating Actor	Student
Actor's Goals	To be able to view results in the course
Preconditions	Student must be signed into an account
Postconditions	Must be able to resume normal activities after viewing
Flow of Events User: →The student clicks on a button to view his results in the class. ←The system will display a page listing results of each individual activity, with a cumulative score out of 100 shown at the bottom →The student clicks to go back to the main menu. ←The system returns to the main menu and allows for the student select any option from the menu	

Use Case UC-13	Grading
Requirements	REQ 4, REQ 5, REQ 6, REQ 7, REQ 19, REQ 22, REQ 23, REQ 37, REQ 38, REQ 45, REQ 53, REQ 54, REQ 55, REQ 66
Initiating Actor	Instructor

Actor's Goals	To calculate students' course grades and manually adjust or edit any errors
Preconditions	Instructor must be signed into an account
Postconditions	All grading and edits will be saved

Flow of Events
Instructor:
→The instructor clicks a button to view the grades of the students.
←The system displays a page listing every student's name and overall course grade.
→The instructor clicks on a student's name to view the grades.
←The system displays a page listing every grade for that student from the semester.
→The instructor clicks on a grade and has the option to adjust it.
←The system saves the adjusted grade, recalculates the course grade, and displays the updated grades.
→The instructor clicks a button to go back and view the grades of all the students.
←The system displays a page listing every student's name and overall course grade.
→The instructor clicks a button to go back to main menu.
←The system returns to the main menu and allows the instructor to select any option from the menu.

Use Case UC-14	CreateAssignments
Requirements	REQ 40, REQ 41
Initiating Actor	Instructor
Actor's Goals	To create mini quizzes and pre-labs to assign to students
Preconditions	Instructor must be signed into an account
Postconditions	The mini quiz or pre-lab will be saved and available for students to complete

Flow of Events
Instructor:
→The instructor clicks a button to create a new assignment.
←The system displays a page which gives the instructor the option to create a mini quiz or a pre-lab.
→The instructor clicks a button to select one of the options.
←The system allows the instructor to create the assignment.
→The instructor creates the assignment, sets a deadline, and clicks a button to publish.
←The system opens the assignment to students.

Use Case UC-15	Notifications
Requirements	REQ 6, REQ 12, REQ 14, REQ 21, REQ 42, REQ 45, REQ 46, REQ 47, REQ 49, REQ 50, REQ 52, REQ 56, REQ 61
Initiating Actor	System, Instructor
Actor's Goals	Communicate the appropriate information to the user
Preconditions	User is logged into an account
Postconditions	Notification must be marked as "seen" after viewing
Flow of Events System: ←An event causes the system generate a notification and sends it to the user. →The user clicks on the notification to view it. ←The system allows the instructor to create the assignment.The system displays the message to the user and marks the message as "seen". →The user can exit and return to main menu. ←The system allows the instructor to create the assignment.The system returns to the main menu and shows no pending notifications.	

Use Case UC-16	History
Requirements	REQ 1, REQ 4, REQ 5, REQ 9, REQ 13, REQ 22, REQ 27, REQ 34, REQ 41, REQ 43, REQ 48, REQ 51, REQ 56, REQ 59, REQ 65
Initiating Actor	System, Instructor
Actor's Goals	Keep a record of when user logs in, attempts at logging in, attempts at labs, chat history, etc.
Preconditions	Instructor is logged into an account
Postconditions	The records will be saved and available to view
Flow of Events →The student logs in, attempts to log in, attempts assignments, and logs out. ←The system saves this activity. →The instructor logs in and clicks on a student's profile to view activity.	

←The system displays the student's history to the instructor. →The instructor can choose to return to main menu. ←The system displays the main menu to the instructor.

Use Case UC-17	Quizzes
Requirements	REQ 63, REQ 64, REQ 65, REQ 69
Initiating Actor	Student
Actor's Goals	Attempt the quiz and submit after finishing or when time expires
Preconditions	Student is logged into an account
Postconditions	Quiz is submitted for automatic grading. Only instructor can access quiz.
Flow of Events Student: →The student logs into the student account. ←The system displays the main menu. →The student click on quizzes. ←The system displays the quiz menu showing a list of quizzes. →The student clicks on a quiz. ←The system displays the quiz and allows the student to complete the quiz while enforcing time limit. →The student finishes the quiz and clicks on submit button or allows time to expire. ←The system will close the quiz, submit results, and display the quiz menu.	

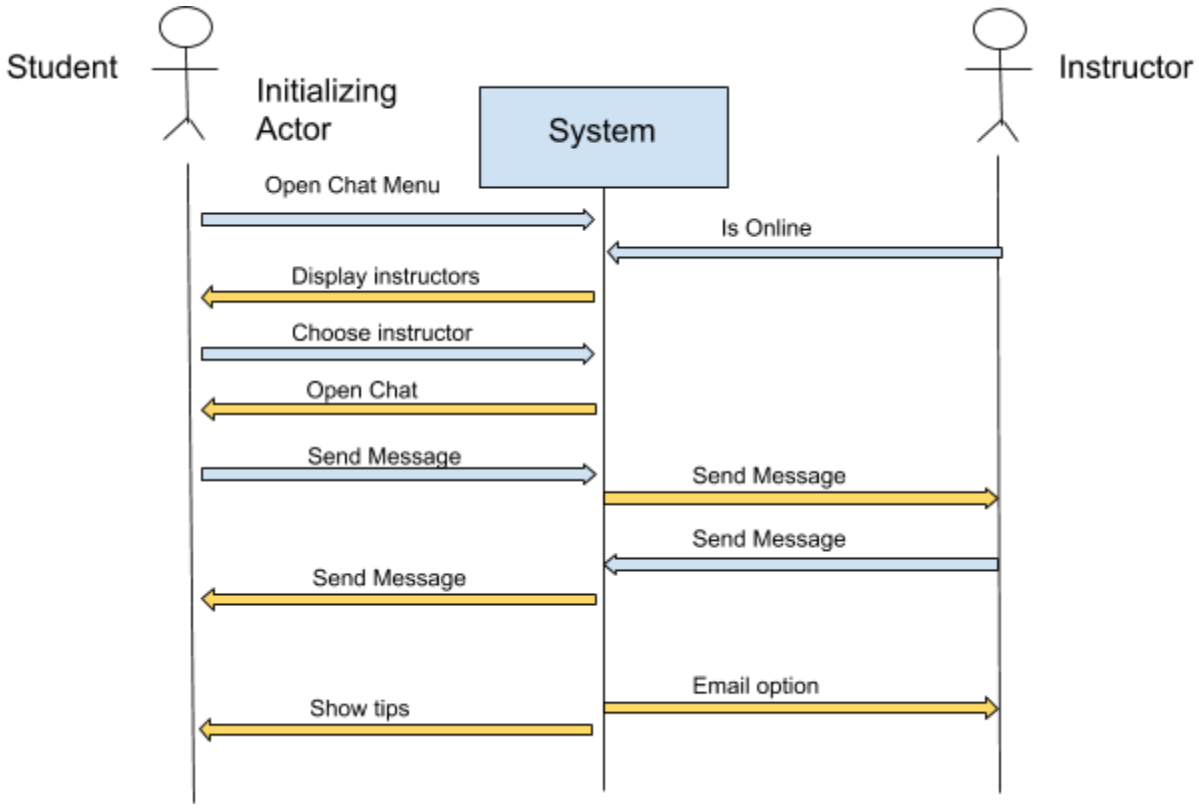
Use Case UC-18	ScreenCapture
Requirements	REQ 15, REQ 16, REQ 47, REQ 52
Initiating Actor	Student
Actor's Goals	Share screen with instructor whether it be through a lone image or through screen sharing
Preconditions	-Student is logged into an account -Instructor is online (Only for screen sharing)
Postconditions	Image is sent to instructor through chat

	Screen sharing is ended

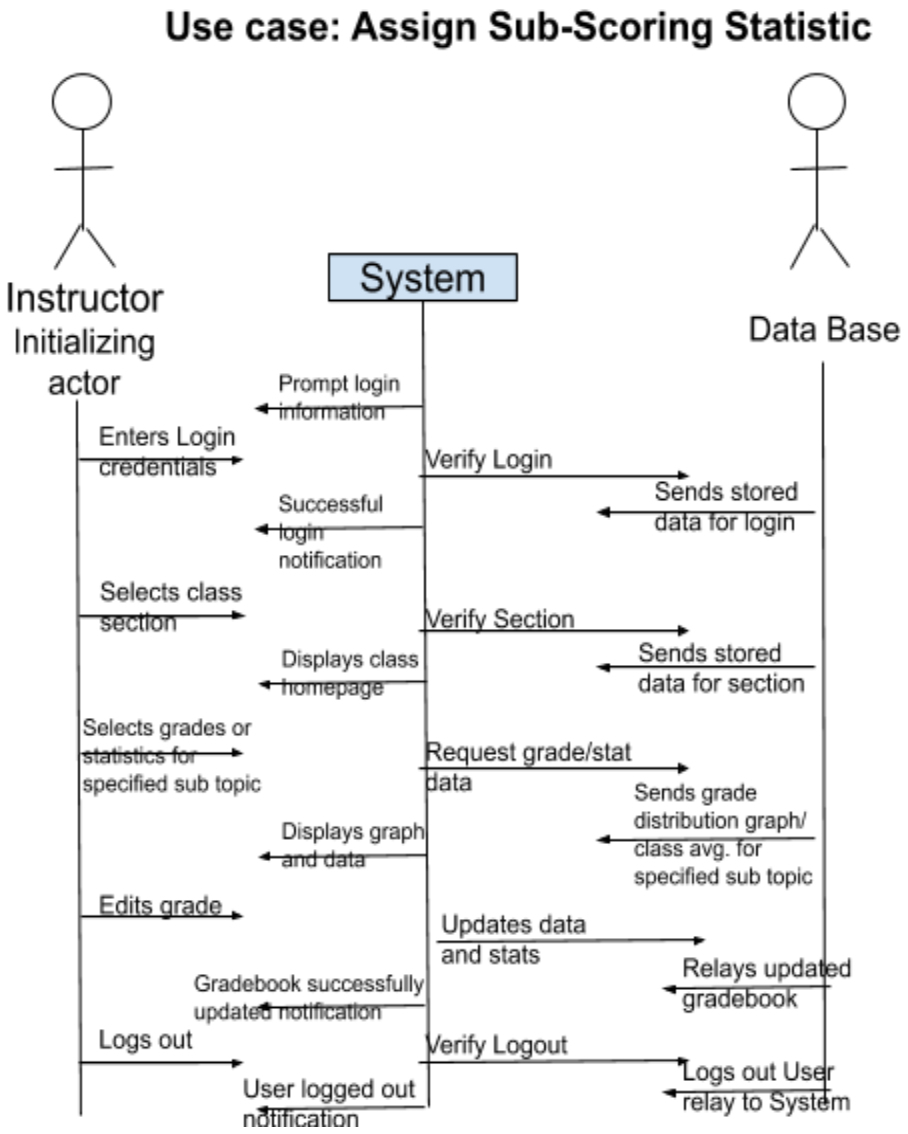
Flow of Events
Student:
→The student starts the lab.
←The system displays the experiment.
→The student clicks on chat menu.
←The system displays the initial menu, showing which instructors are online.
→The student starts a chat with an instructor.
←The system notifies the instructor.
→The student clicks on the camera button and chooses image or screen sharing.
←The system opens the camera menu.
Image:
→The student clicks on the camera capture button to take an image.
←The system captures and image of the screen and prompts the user asking if the image is ok.
→The student clicks “ok”.
←The system sends the image to the instructor.
Screen Share:
→The student clicks on the screen share icon.
←The system prompts the student asking if they want to screen share.
→The student clicks “ok”
←The system notifies the instructor of the screen share request.
→The instructor clicks “ok”
←The system shares the screen of the student in real time.

System Sequence Diagram:

UC-5 diagram

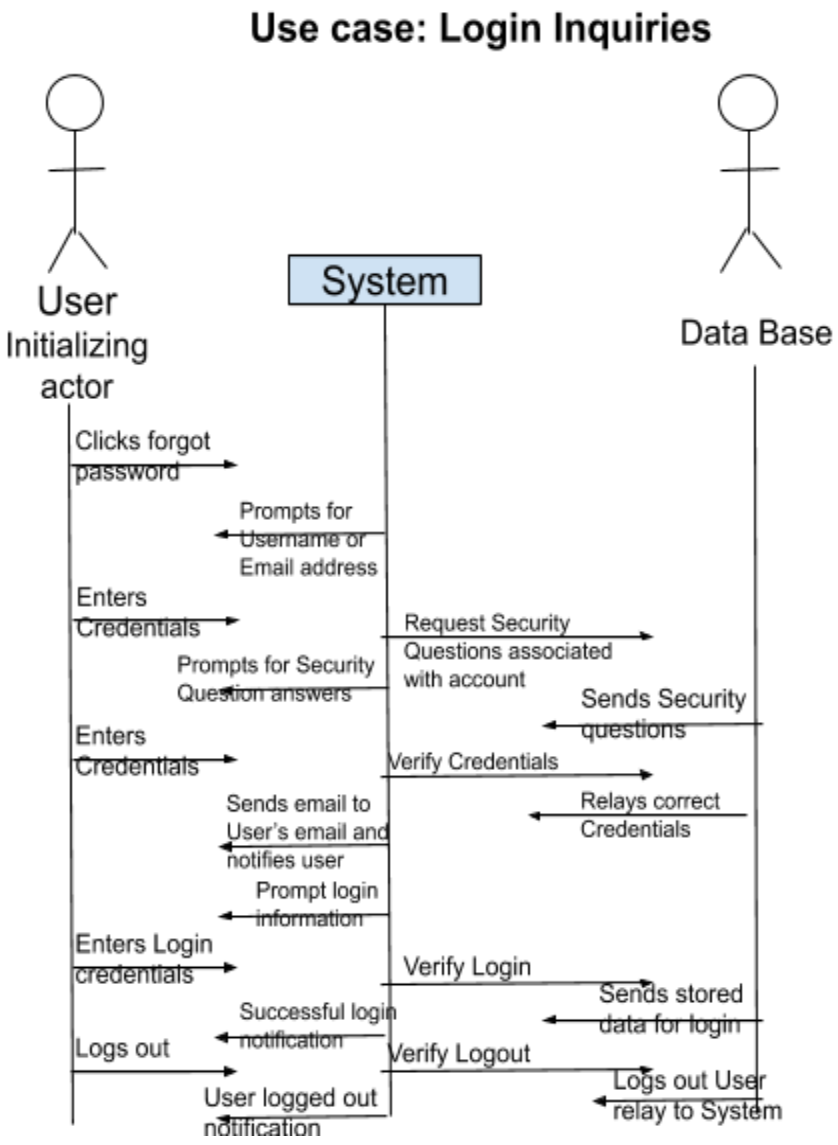

Student
Initializing Actor
System
Instructor
Open Chat Menu
Is Online
Display instructors
Choose instructor
Open Chat
Send Message
Send Message
Send Message
Send Message
Email option
Show tips

UC-6 diagram



Main Success Scenario

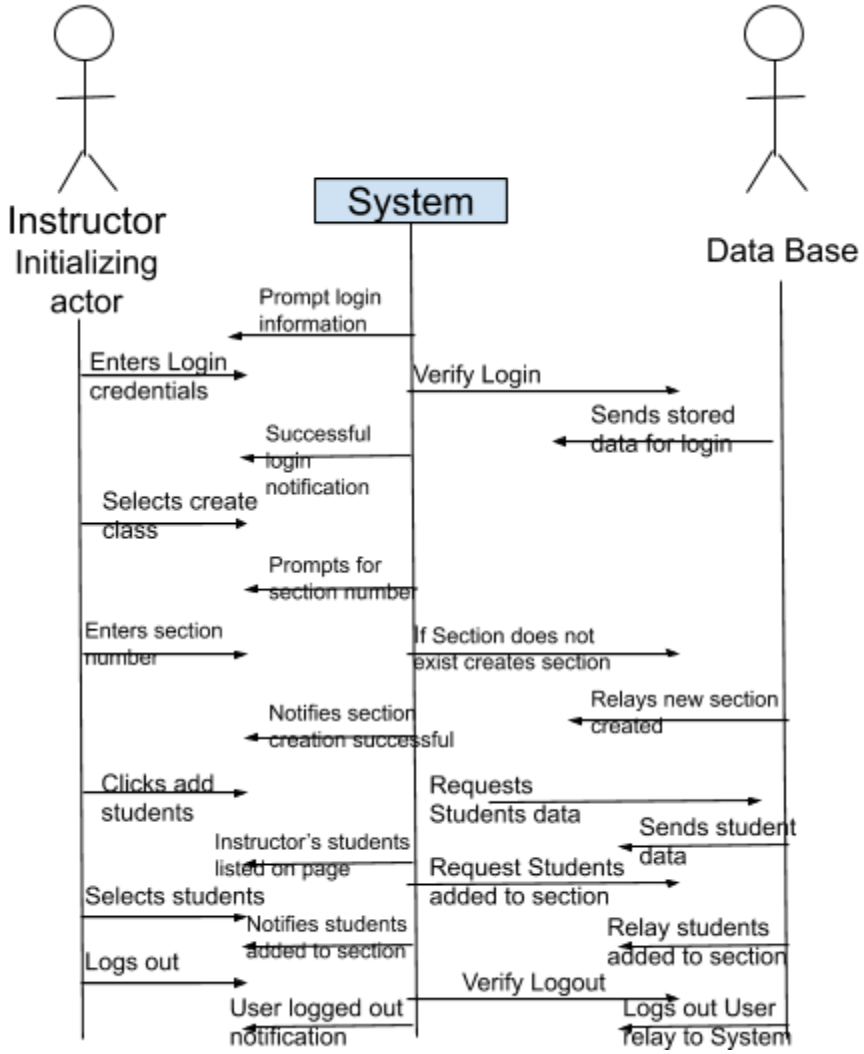
UC-9 Diagram



Main Success Scenario

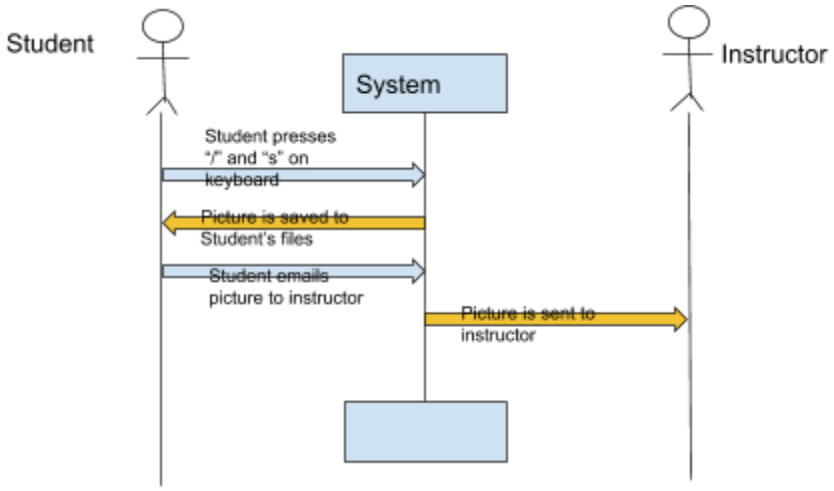
UC-10 Diagram

Use case: Create Class

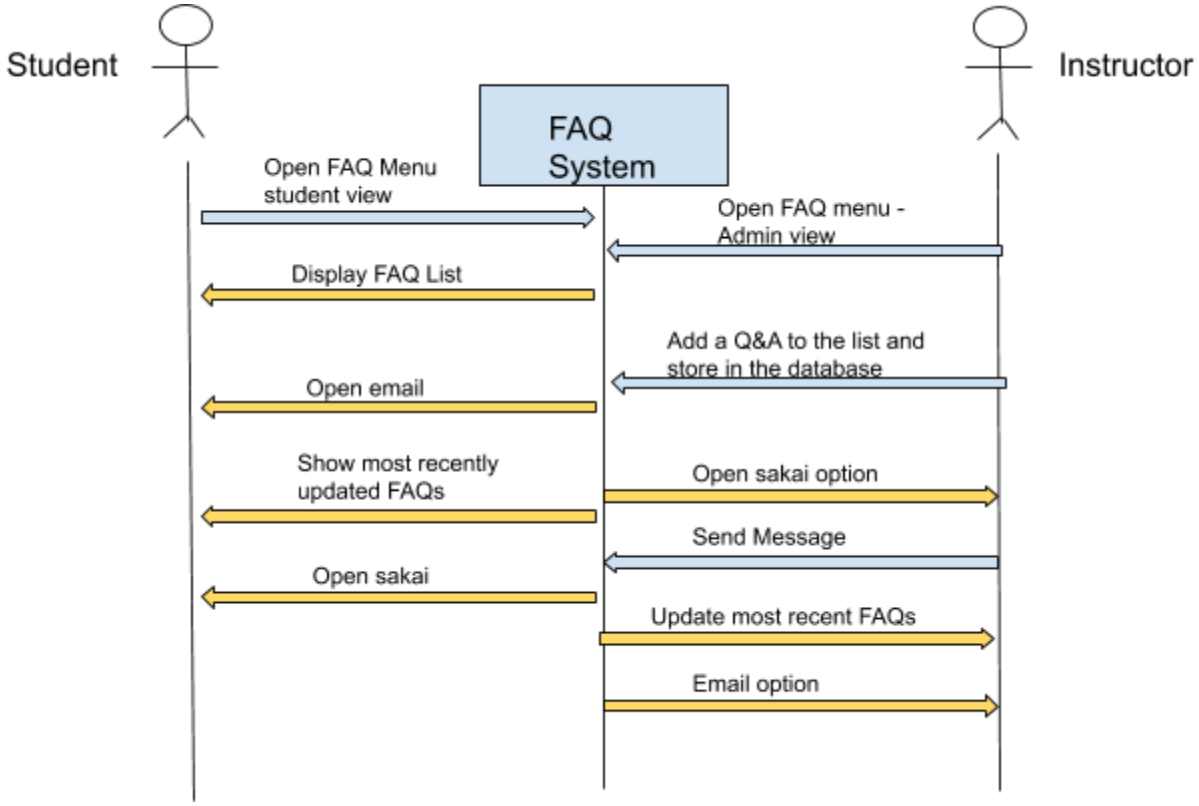


Main Success Scenario

UC-18 Diagram



UC -19 FAQ Diagram



User Interface Specification

User Effort Estimation:

Log In - Total variable presses + 1/0 clicks

a. Enter Username (variable length)
b. Enter Password (variable length)
c. Press enter/click log in

Manage Equipment - Total 3 Clicks

a. Click on drop down menu that holds all the equipment
b. Click on the equipment desired which places the equipment under the mouse
c. Click anywhere in the view to place the equipment

Chat - Total 2 clicks + 1 press Assumption: Student is already logged in

a. Click on Chat tab
b. Click on the text box on the bottom to enter text
c. Press enter

Class Homepage(Current term) - Total 1 click - Assumption: Instructor is already logged in

a. Click on class from list

Assignment Grades - Total 3 clicks - Assumption: Instructor is already logged in

a. Click on class
b. Click on Assignment from assignment list
c. Click on statistics tab

Gradebook Statistic - Total 3 clicks - Assumption: Instructor is already logged in

d. Click on class
e. Click on “Gradebook” from class homepage
f. Click on statistics tab

Login Statisistic - Total 4 clicks - Assumption: User has already established and account and security questions

a. Click on “Forgot Password”
b. Click “Enter” after entering username/email
c. Click “Enter” after entering answer to security question
d. Click “Enter” after entering new password

Logout Statistic - Total 2 clicks - Assumption: User is at the main menu:

a. Click on “Log Out”
b. Click “Yes” when asked if the user is sure he would like to log out”

Current Status Statistic - Total 2 clicks - Assumption: Student is at the main menu.

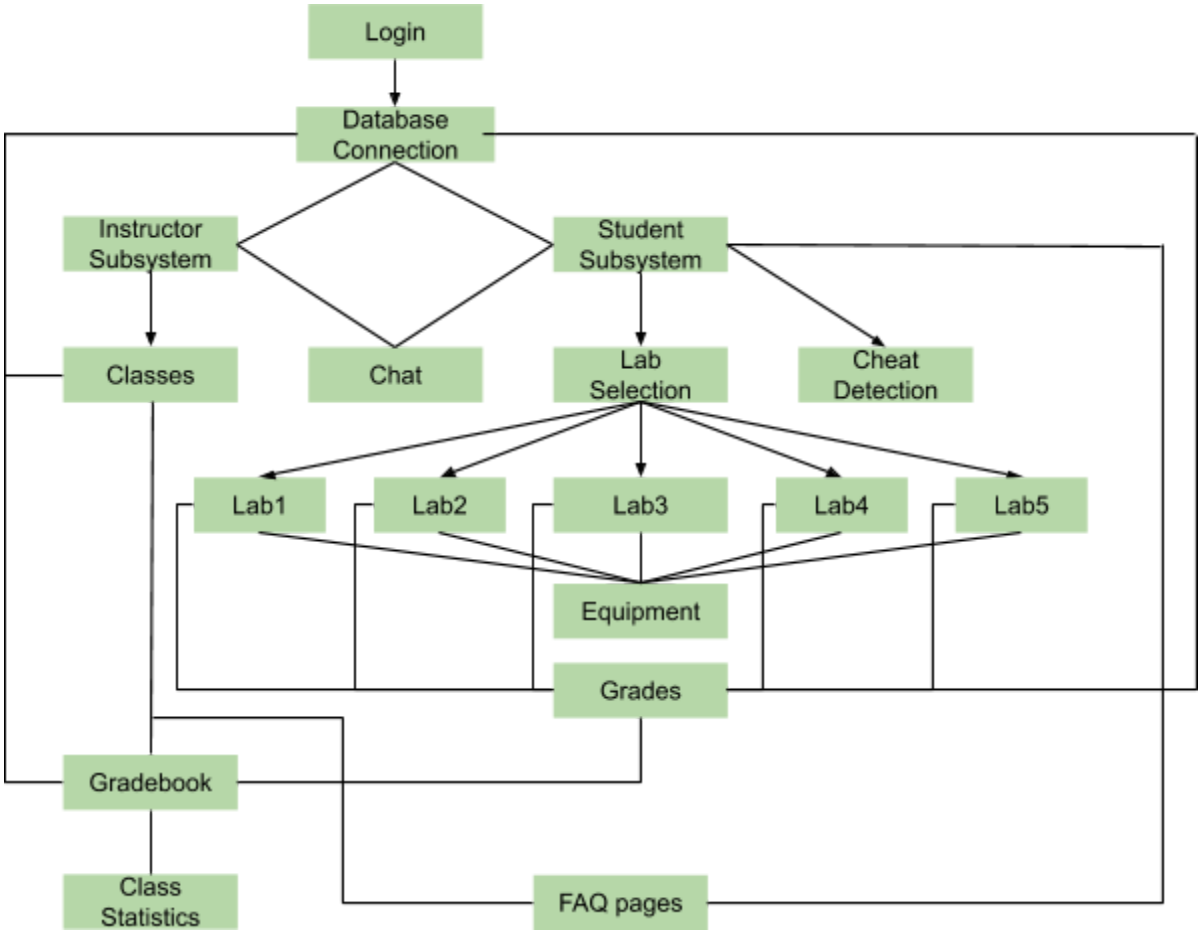
a. Click on “View Course Results”
b. Click on “Main Menu” to return to the main menu once finished.

Create Classes Statistic - Total 2 clicks + 1 click per student - Assumption: Instructor is at the main menu

a. Click on “Create New Class”
b. Move students into new class at one click per student
c. Click on “Confirm” to finalize the creation of the class

Domain Analysis

Domain Model:



Concept Definitions:

The table below lists the concept name, responsibility description, and its type. The concept's type is either doing(D) or knowing(K).

#	**Responsibility Description**	**Type**	**Concept Name**
Rs1	Provides a graphical user interface (GUI) for the user to access the application as a student or an instructor. The user's credentials are verified with the database.	K	Login
Rs2	Stores user information, grades, and classes, basically overall user data.	K	Database
Rs3	Provides a graphical interface for an instructor to chat with students, access classes, grades, and course statistics.	D	Instructor Subsystem

Rs4	Provides a graphical interface for a student to chat with instructors and interact with labs, quizzes and practice mini quizzes.	D	Student SubSyste m
Rs5	Establishes the connection that instructors and students are able to communicate with each other through.	D	Chat
Rs6	A list of all classes that instructors can access to view grades, assignments, and student information.	K	ClassesMe nus
Rs7	A list of all assignments and grades corresponding to each student in a specific class.	K	Gradebook
Rs8	Displays a graph of the grade distribution and statistical variables to the instructor using information from the gradebook.	D	Class Statistics
Rs9	Displays a graphical interface for students to choose labs from.	D	Lab Selection
Rs10	Allows students to simulate lab 1	D	Lab 1
Rs11	Allows students to simulate lab 2	D	Lab 2
Rs12	Allows students to simulate lab 3	D	Lab 3
Rs13	Allows students to simulate lab 4	D	Lab 4
Rs14	Allows students to simulate lab 5	D	Lab 5
Rs15	Virtual components that are configured to represent their realistic counterparts. Used in lab simulations.	D	Equipment
Rs16	Allows students to work on mini quizzes	D	Quizzes

Association Definitions:

Concept Pair	Association Description	Association Name

USER ←→ Login	Login receives user input and verifies user credentials with the database.	Conveys Request
USER ←→ Logout	Logout sends a request to logout user, (no 2 users can be logged in at the same time).	Conveys Request
Login←→ Database	Login checks to see what kind of user is login in.	Conveys Request
Logout←→ Database	Logout logs out user, regardless of if they're logged in (incase they're logged out already for whatever reason).	Conveys Request
Database ←→ Instructor GUI	User is verified as instructor, display instructor homepage.	Generates
Database ←→ Student GUI	User is verified as student, display student homepage.	Generates
Student GUI ←→ Database	Student saves work on the lab to resume at another time.	Conveys Request
Student GUI ←→ StudentSubSystem	Student chooses what task to work on.	Generates
StudentSubSystem ←→ Quizzes	Student decides to work on quizzes.	Generates
StudentSubSystem ←→ Labs	Student decides to work on labs	Generates
Labs←→Equipment	The active labs will allow students to interact with the equipment pertaining to the particular lab	Conveys Request
Lab 1,2,3,4,5 ←→ Quizzes	If student chooses to work on practice quizzes before starting the lab they can do so.	Generates
Labs←→CheatDetection	While the lab is active, allows the system to detect if the student is clicking on things outside of the GUI, to use outside material	Request notify
Student←→ Instructor	Student sends a request to share screens with the instructor for assistance in a lab.	Conveys Request
Student←→ Instructor	Student sends chat message/screenshot to instructor for assistance in lab.	Conveys Request

StudentSubSystem ←→chat	Student selects to chat with instructor	Generates
Instructor SubSystem←→Chat	The chat functionality is displayed in the instructor's GUI	Generates
Instructor←→ Database	Instructor clicks to view the grades of all students in specified section.	Conveys Request
Instructor ←→ Database	Instructor edits grade in gradebook for student in specified section.	Conveys Request
Instructor ←→ Database	Instructor specifies exact time when each class is able to access labs.	Conveys Request
Labs ←→ Database	When student attempts to access labs checks if class is in time frame to access labs.	Conveys Request
Chat ←→Database	Maintains history of chat in database	Conveys Request
FAQ←→Database	Maintains history of FAQs in database	Conveys Request
Labs ←→Database	Maintains history of lab progress	Conveys Request

Attribute Definitions:

Concept	Attributes	Attribute Description
Login	checkType()	Checks to see if the user is an instructor or a student
	authenticate()	Checks if the username and password match the record in the database
	launchGUI()	Launches the respective GUI for the student or the instructor

Lab	compareOutput()	Compares the output of the logic to the answer truth table, capable of assigning partial credit
	saveProgress()	Saves student's current work to database-accessible only by the instructor and the student
Equipment	getEquipment()	Gets the equipment pertaining to each lab ready for when the user is ready to place it on the breadboard
Report	EvaluateLab()	Evaluates the lab to generate a grade for the specified lab.
	EvaluateQuiz()	Evaluates the quiz to generate a grade for the specified quiz
	GetGradeFromLab( )	Returns grade from the specified lab for the student class
StudentGUI	startQuiz()	Allows student to take quiz.
	startLab()	Allows student to start a certain lab.
	startChat()	Allows student to initiate chat with instructor.
InstructorGUI	createQuiz()	Allows instructor to create quizes
	makeAnnouncement()	Allows instructor to make announcements to class.

	createClass()	Allows instructor to create class.
	manualGrade()	Allows instructor to modify grading if need be.
	getGrade()	Method to get a list of students' grades.
CheatDetector	retrieve()	Retrieves camera footage
	outsideClick()	Keeps checking to see if student is doing anything outside of the software.
Quiz	giveQuiz()	Administers quiz to student
	recordResults()	Records results of quiz
	checkAnswers()	Checks the users answers with the recorded actual answers.
Chat	enterMessage()	Allows user to enter message to the chat
Screen Share	requestScreenShare()	Method to request to share screens with another user
	AcceptDenyScreenShare()	Allows user to accept or deny request.
	endScreenShare()	Method to stop sharing screen
Classes	addStudent()	Method to add students to a class
	editClass()	Method to open the menu to make modifications to a class
	changeClassName()	Method to change a class's name
	editClassLabTime()	Specifies when lab is accessable to a class

History	chatHistory()	Allows the user to view chat history. For students, they can only view their history with the instructor. For the instructor, they can choose which chat history to view.

Traceability Matrix

Domain Model																					
Use Case	PW	Login GUI	Chat	Classes	Screen Capture	Lab 1	Lab 2	Lab 3	Lab 4	Lab 5	Equipment	Report	StudentGUI	InstructorGUI	Cheat Detector	Quiz 1	Quiz 2	Quiz 3	Quiz 4	Quiz 5	Database	FAQs
UC-1	11	X											X	X								
UC-2	8					X	X	X	X	X				X		X						
UC-3	15										X											
UC-4	1					X	X	X	X	X						X	X	X	X	X		
UC-5	20		X		X																	
UC-6	26					X	X	X	X	X			X									
UC-7	27													X							X	
UC-8	12			X										X								
UC-9	9	X																				
UC	11			X										X								

-10																					
UC -11	15											X	X								
UC -12	39																			X	
UC -13	42											X	X								
UC -14	3												X								
UC -15	28		X	X		X	X	X	X	X				X	X	X	X	X	X		
UC -16	41											X	X							X	
UC -17	12														X	X	X	X	X		
UC -18	6		X		X							X									
UC -19	15											X	X							X	X

Domain analysis text description:

UC-1: Login will use the data base and based credentials will load the Student GUI or the Instructor GUI
UC-2: Instructor uses the results from labs and prelabs to get a display of the students and their grades.
UC-3: Equipment(logic gates, LEDS, wire) is used and shared between all labs.
UC-4: The database connection is web hosted allowing the grades accessed and updated when needed.
UC-5: Chat is an established link between instructor and student.
UC-6: Grabs the grades from each lab
UC-7,13: Uses the database to store the grades
UC-8,9,10, 12: uses the admin page to create sections for students and view their corresponding grades
UC-11: logout is also a database-subsystem relationship that just returns the user to the login menu
UC-14: uses the student subsystem to display available quizzes
UC-19: The FAQ page pulls from the database and displayed for both user types.

System Operation Contracts:

Name: login()
Responsibility: Use the database to control account systems and decipher between admin and student
Cross-References: UC-1
Output: The system will present the user wit a UI asking them submit corresponding credentials.
Pre-Conditions: Both the account name and password must be valid and in the database before allowing the account to gain access to the system.
Post-Conditions:

-User should have access to the labs, grades, and other valuable resources for such labs.

Name: instructorLabControl()
Responsibility: Under the instructor account, manually control/modify the students registered as well as the grades for the students.
Cross-References: UC-2
Output: The system will present the user wit a UI allowing them to view active students registered for the lab and their respective grades
Pre-Conditions: Instructor account must be logged in. Students must already have already completed at least one lab to be able to modify grades.
Post-Conditions:

-Both the instructor and the student's grade books will be updated. This includes all statistics based on the student's current grades as well..

Name: managementEquipment()
Responsibility: From the available equipment, this operation responds to the user's click/drag
Cross-References: UC-3
Output: The system can introduce new object to the board and have them interact with each other
Pre-Conditions: Student clicks on and drags a piece to place on the protoboard
Post-Conditions:

-A new Equipment object is created and given the ability to interact with the
-protoboard Different parts interact with each other as they are moved around the board
-Location values for each object constantly change as they are moved around

Name: chat()
Responsibility: Control the live interaction between students and instructors.\ via active chat messaging system
Cross-References: UC-5

Output: The system will have its own small UI on top of the lab UI and will present the use with options such as starting a new conversation, screen sharing, and more.
Pre-Conditions: Student or instructor must be logged in. Students must be actively doing a lab at the time an instructor is available to chat.
Post-Conditions: A history of the chat should be stored in the database for a certain time.

Name: quiz()
Responsibility: Give the students a quiz upon the completion of each lab
Cross-References: UC-17
Output: A quiz is given to the user upon completion of a lab experiment
Pre-Conditions: The user completes and submits a lab
Post-Conditions:

-The user is presented with a quiz, typically the image of a Karnaugh Map or Truth table that must be solved

-Empty spaces must be filled with values, which will be compared to the correct values after submission

Mathematical Model:

Statistics Mathematical Model

Grade Average:

$$\frac{\left(\sum_{i=1}^{N} \frac{SG}{TG} * 100\%\right)}{N} = \text{Average Grade of Section (in \%)}$$

Variables

N = Number of students in section
SG = Student's grade for assignment
TG = Total grade for assignment
i = initializing number
(starts at first student in class)

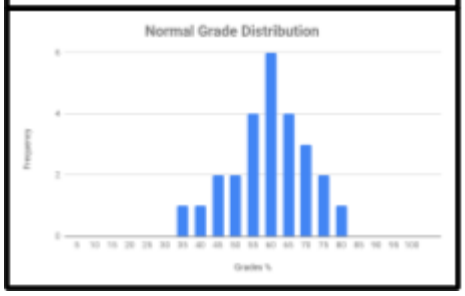
Statistics Mathematical Model

Graph Distribution:

The Frequency Function returns an array of how often the data value occurs within a range of interval values.
Frequency(data,classes)

Bin Array is simply used to create a range of data on the x-axis.

In a real example Bin Array would be more precise and would include every value from 0-100, instead of increments of 5.



Using Google Sheets, I was able to simulate what the Normal Grade Distribution would look like given this sample set of Student Grade Data.

The Normal Grade Distribution is easily found utilizing the Frequency formula and tools given within Google Sheets

The information for Frequency, Students, Grade% and Bin Array are given below.

Given any sample of student data we can easily determine a graph for the normal grade distribution using this function and data set.

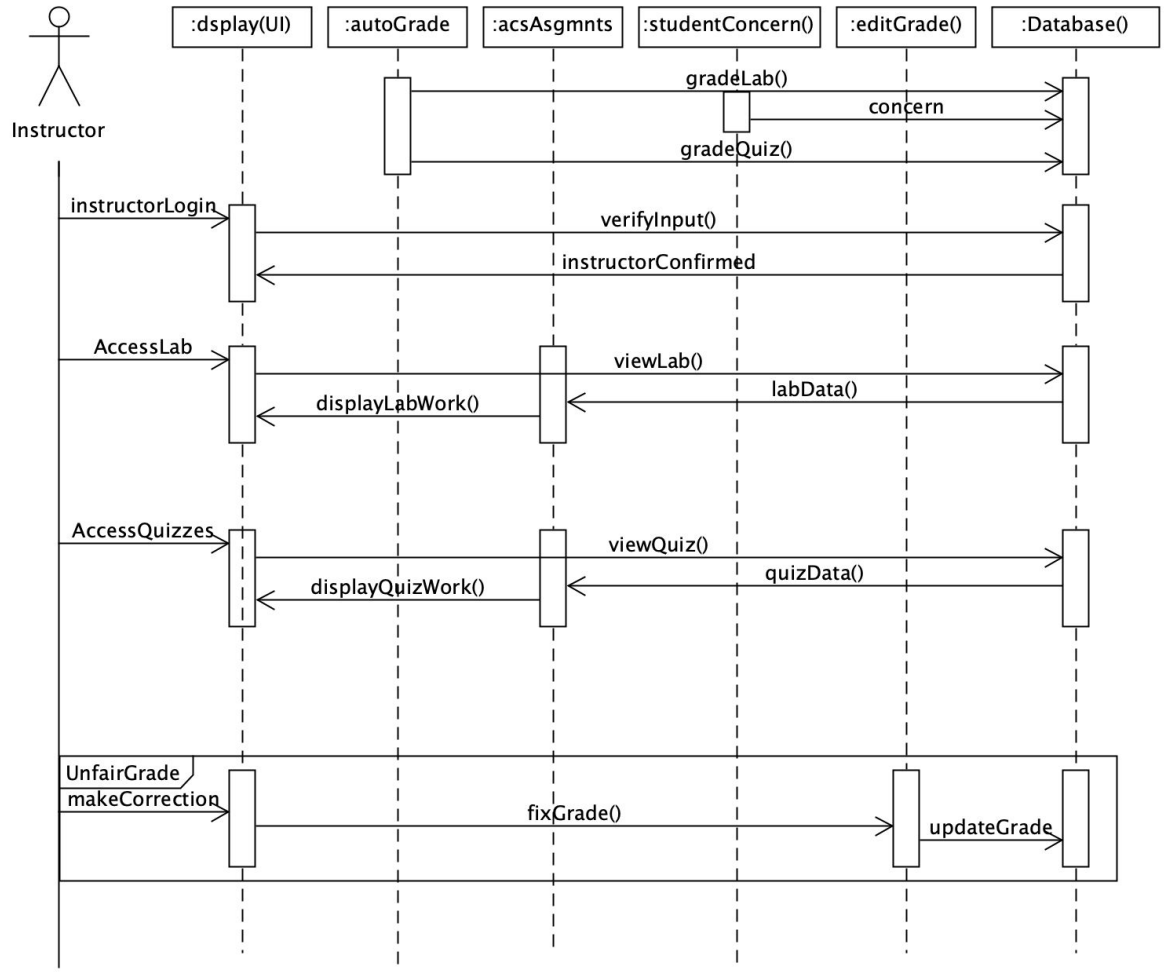
Student	Grade (%)	Frequency	Bin Array	N	53	3	65
A	45	0	0	O	65	2	70
B	48	0	5	P	29	1	75
C	73	0	10	Q	57	0	80
D	54	0	15	R	48	0	85
E	66	0	20	S	55	0	90
F	62	0	25	T	43	0	95
G	38	1	30	U	46	0	100
H	34	1	35	V	53	0	
I	51	2	40	W	58		
J	63	2	45	X	56		
K	68	4	50	Y	60		
L	36	6	55	Z	54		
M	48	4	60				

Project Size Operation

	Group A	Group B	Group C	Group D
UUCP	**90**	**90**	**125**	**100**
TCF	**0.900**	**0.905**	**0.89**	**0.85**
UCP=UUCP*TCF	**81.00**	**81.45**	**111.25**	**85**
Normalized UCP	**0.73**	**0.73**	**1**	**0.76**

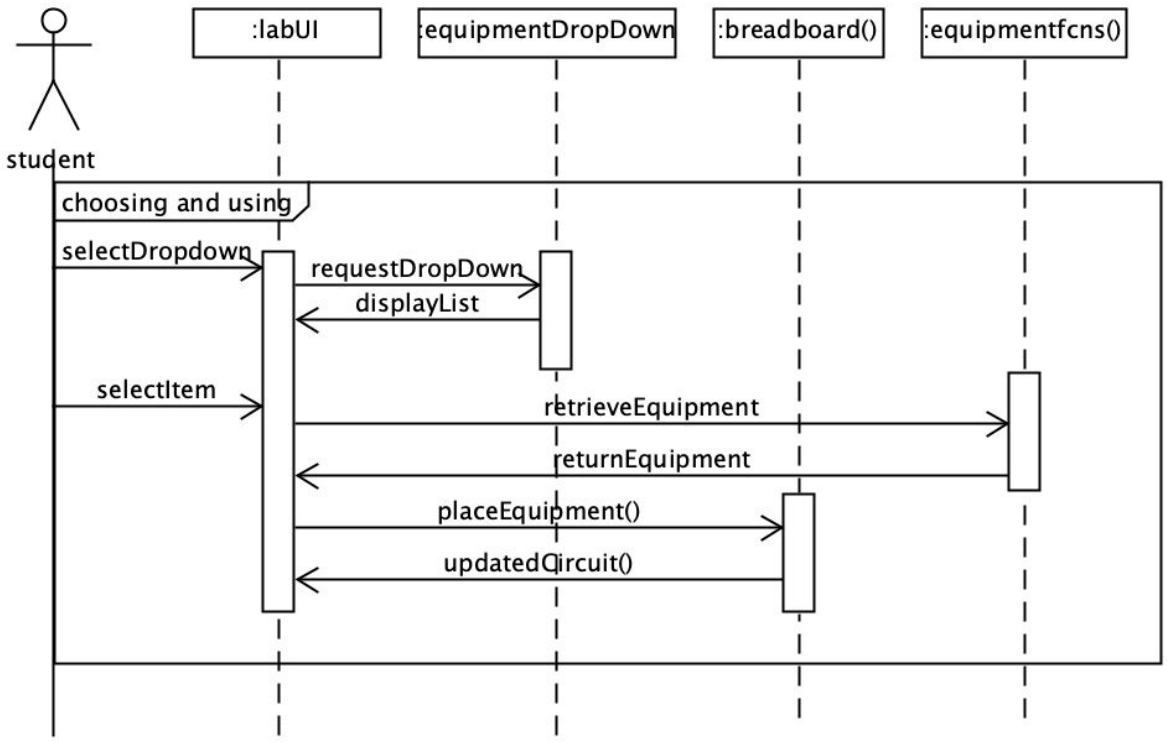
Interaction Diagrams

UC-2:Instructor Lab Control



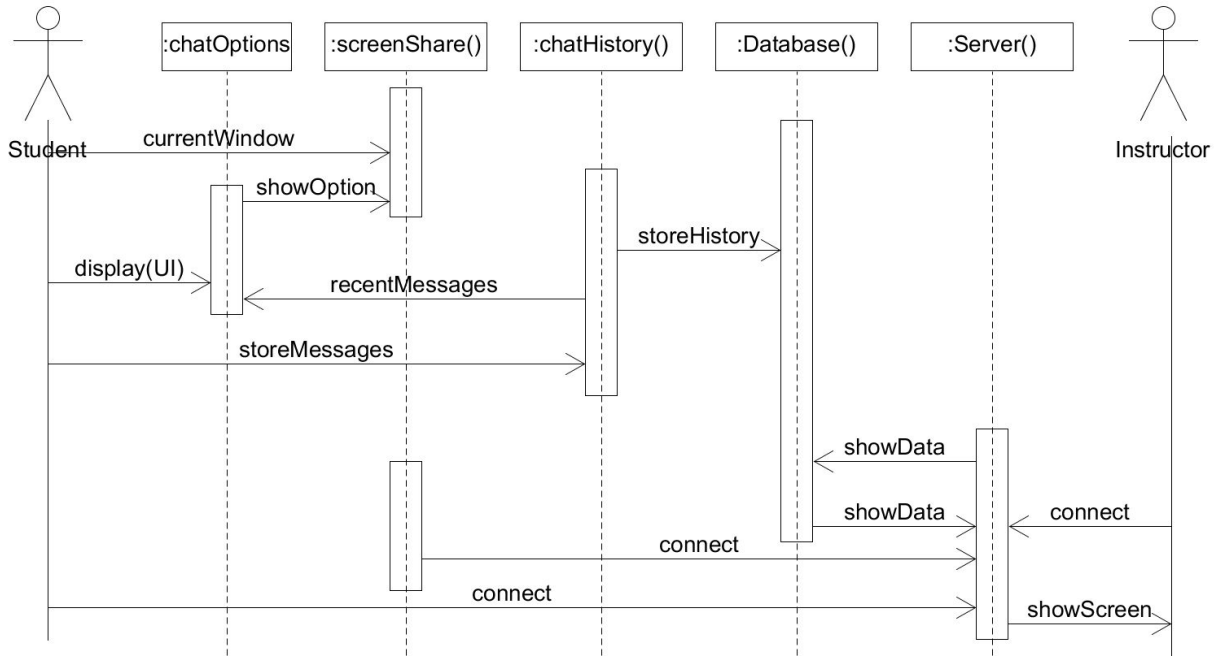
This diagram shows the sequence for an instructor to log in, choose to view students' labs or quizzes. Once they see the student's work progress, they can choose to edit any grading errors that might have occured, or give the student better partial credit if they see it fit.

UC-3 Management Equipment



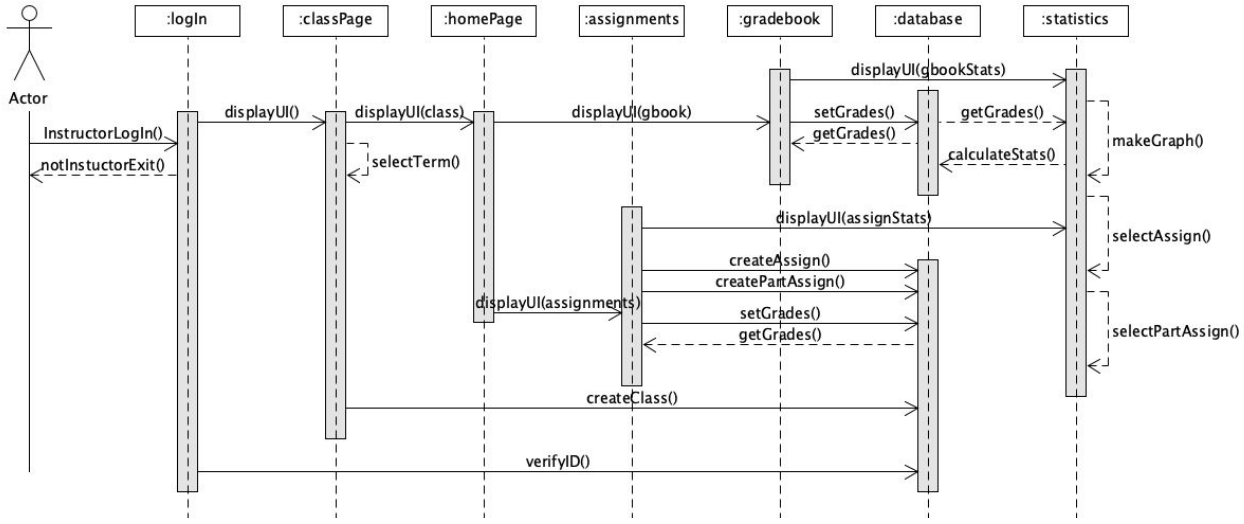
This sequence shows the process for a student to access the tools available in their lab. They can click on the drop down menu to access the equipment for that certain lab and they are able to drag it onto the breadboard.

UC-5: ChatSystem



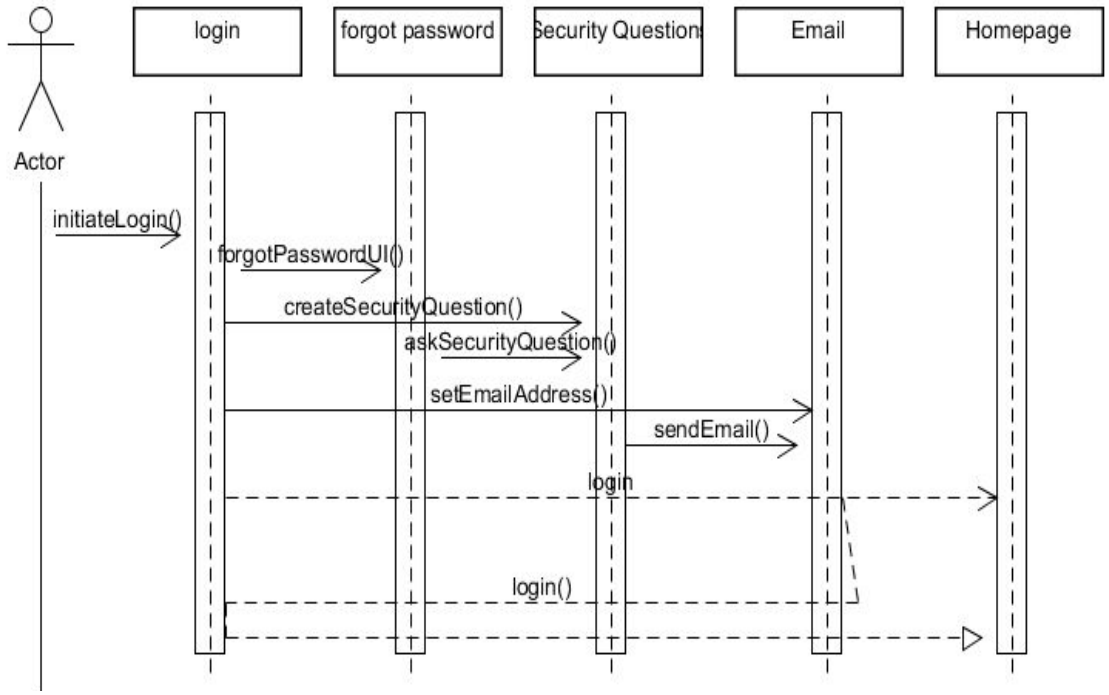
In the above sequence diagram, we could see the interaction between two users and the chat subsystem. The users, both student and instructor would have the ability to connect to the server and chat within the chat display. The subsystem also keep a brief history of their messages. It should also allow the student to show what he/she current has on their screen.

UC-7 Gradebook Statistic



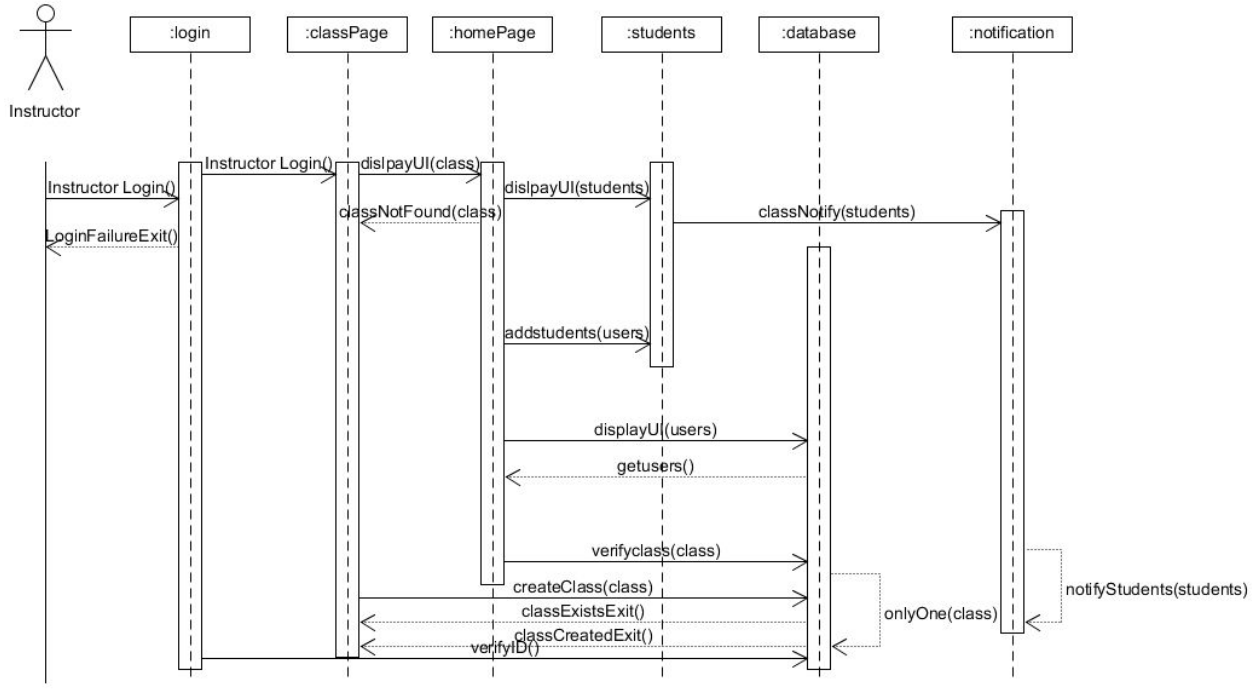
The sequence diagram above represents the interaction between a user and the gradebook statistic use case. In order for the user to access the system, the user must first have a verified instructor log in account. From the class page, the user would be able to select class from a different term, create new classes, or go to a selected class homepage. From the homepage, the user can access the gradebook, or access the assignments page. When accessing the gradebook, the user can view student assignment grades, or view the statistics tab. The statistics tab will display the grade distribution of the students and calculate statistics of the grades for the instructor.

UC-9 Login Inquiries



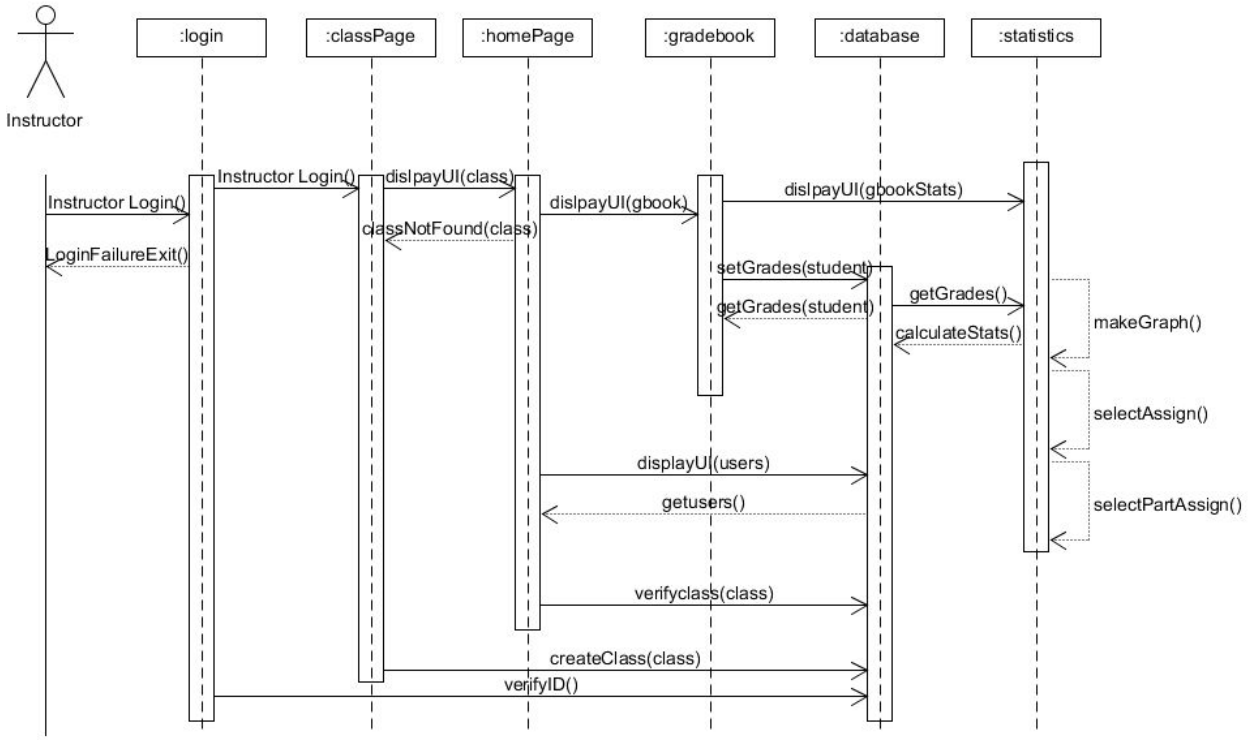
The sequence diagram above shows the interaction between the user and the Login Inquiries use case. The user has the chance to obtain their password in case they forget. The user would initiate the forgotPassword sequence. The user has the option of having the password sent to the user's email. Alternatively, the user can answer security questions that were set when the user created the account. If the answer to the security question is correct, the system verifies and provides the user with the password. Short communication chains and low degrees of connectivity are used between objects for healthy design.

UC-10 Create Class



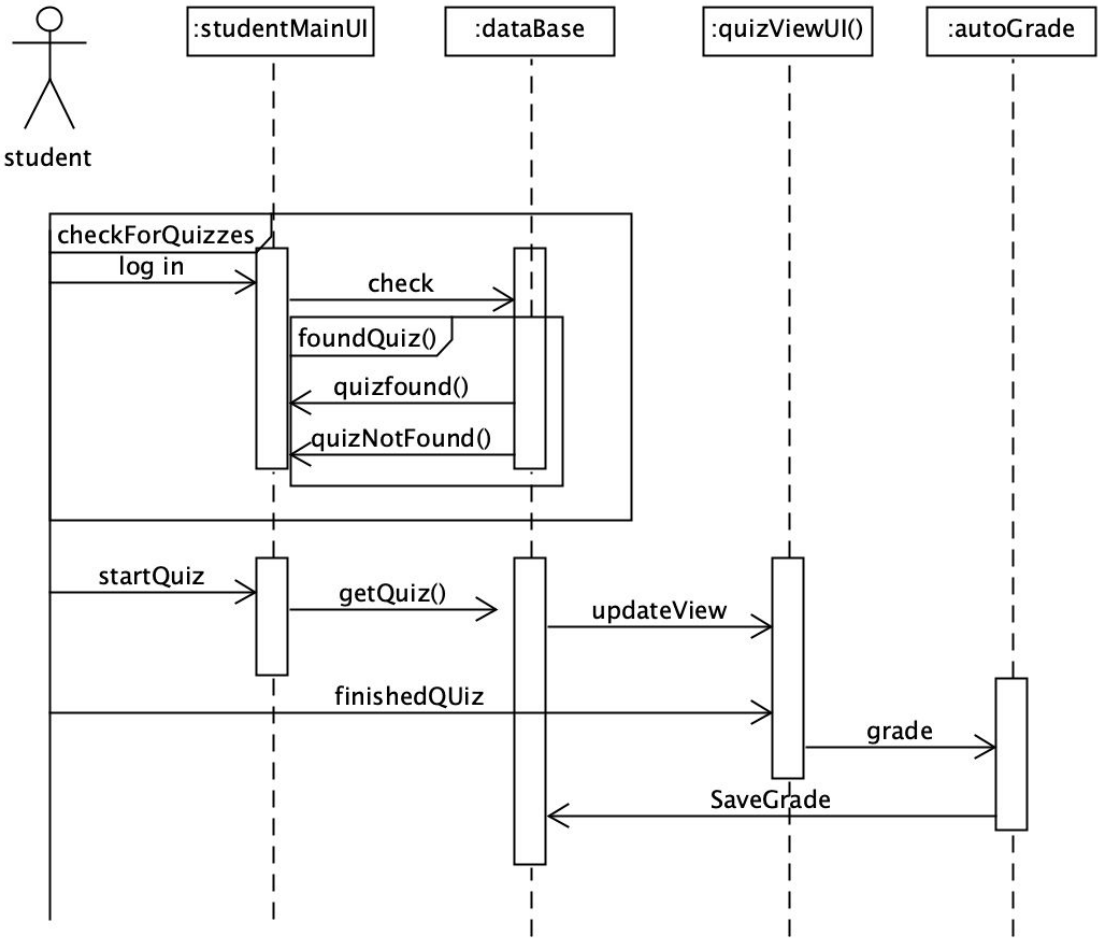
The sequence diagram above represents the relationship between an instructor and class creation. The instructor must first log on with a verified ID, then select createClass() on the classPage, which will prompt the instructor for a section number(class). The database will confirm that the class being created does not exist, and return either classCreated(success) or classExists(failure) exit. The instructor can then access the new class. Then from there the Instructor will be able to add students given user information from the database. They will then select the amount of student to be added. Once completed the notification will send a notification email to all students added to the class, informing them of their addition.

UC-13 Grading



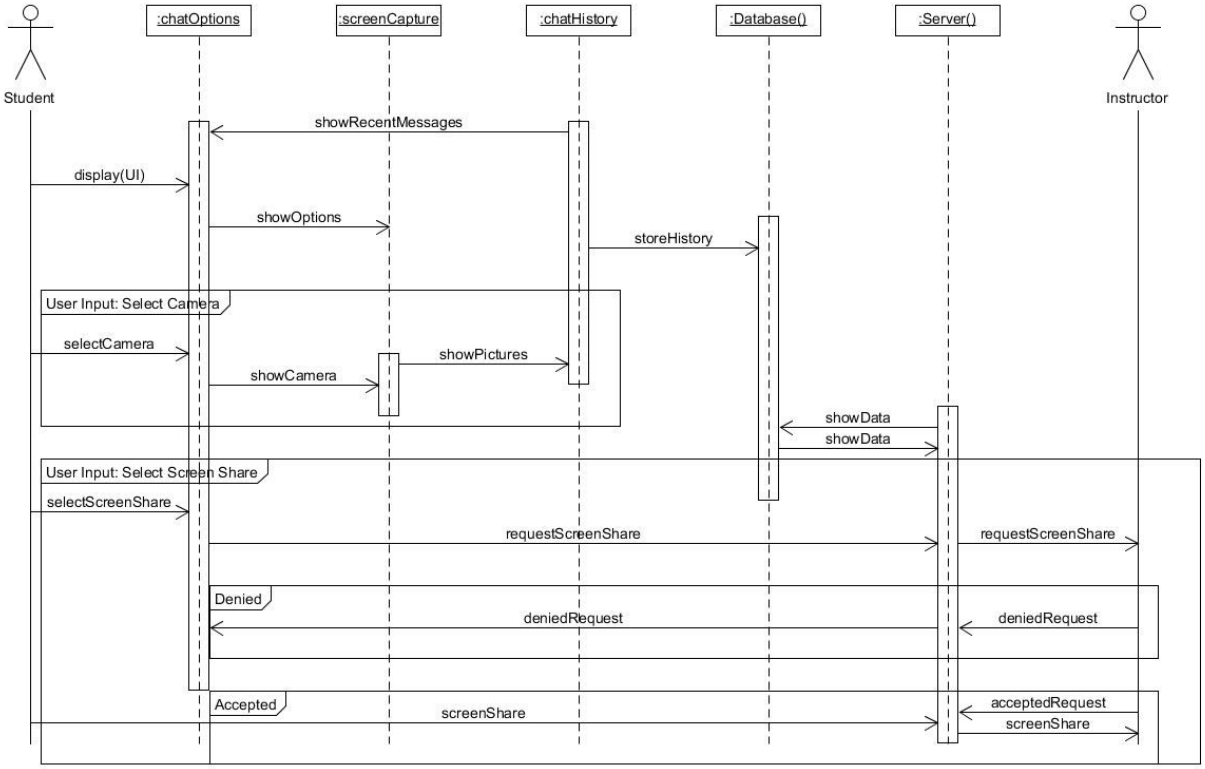
The sequence diagram above represents the relationship between an instructor and grading. The instructor must first log on with a verified ID. Then select a class section and display the gradebook. Then from there they can select a student's grades and edit the contents. The database will make this changes and notify statistics to update the graph averages and ect. The instructor can then check gradebook statistics if they choose.

UC-17 Quizzes



This shows the sequence of a student logging in and the system automatically checking for a new quiz. The student has the option to start the quiz and once they are done the grade is updated into the database.

UC-18 Screen Capture

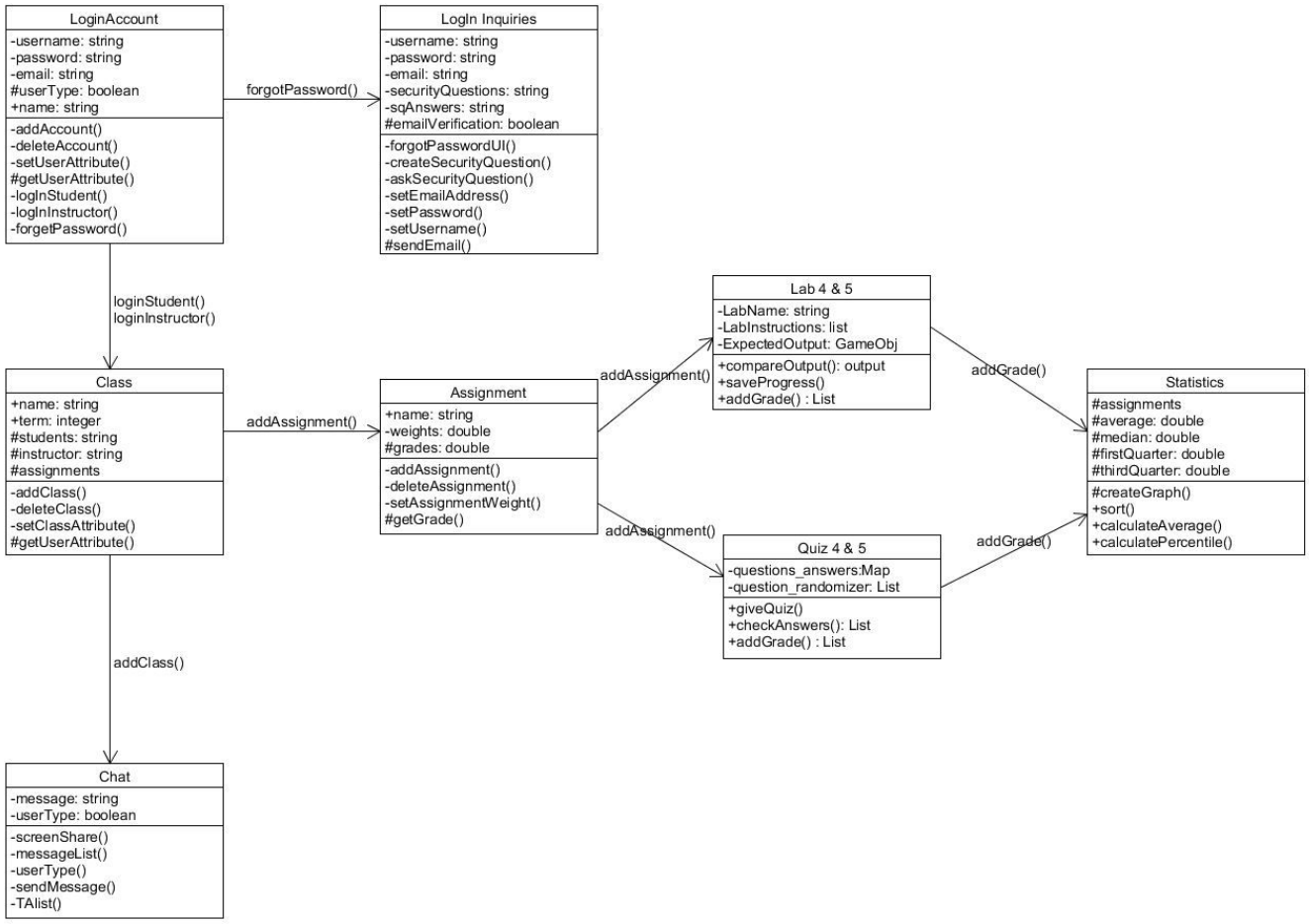


This shows the interaction diagram of the Screen Capture system. This essentially is a more detailed focus on the chat system but specifically the subsection which allows the user to share their screens whether it be through a picture or through live screen sharing with the instructor. This shows the two options of the camera and the screen sharing, as well as the two possibilities of sharing or not given the instructor input.

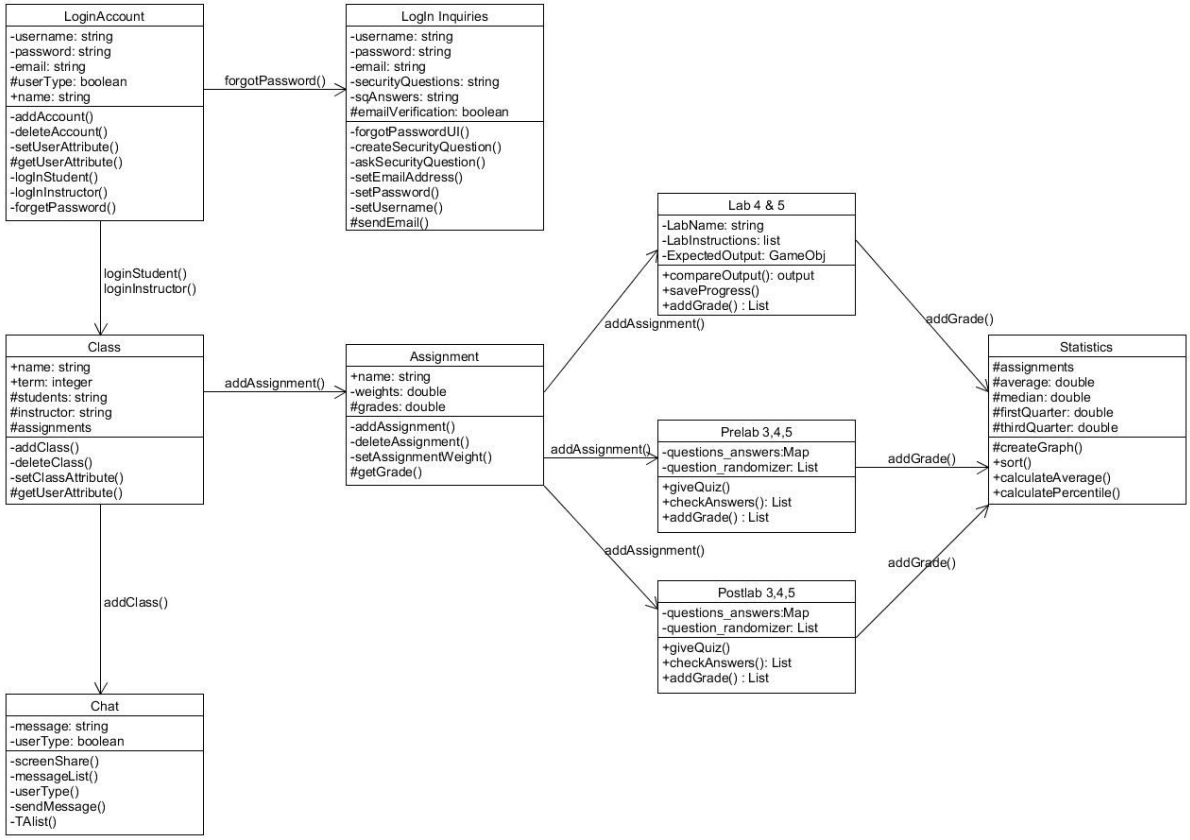
Class Diagram and Interface Specification

Class Diagram:

Below is the general class diagram for the new classes added to the program. (Report 2 version)



This is the updated version of the class diagram which takes into account both the prelabs and the postlabs.



Data Types and Operation Signatures:

Class Name:	LoginAccount	
Attributes:	-username: string -password: string -email: string #userType: boolean +name: string	-Stores unique username -Stores password -Stores verified email #Stores user type (student/Instructor) +Stores name of user
Operations:	-addAccount() -deleteAccount() -setUserAttribute() #getUserAttribute() -logInStudent() -logInInstructor()	-Creates new account -Deletes existing account -Set attributes of an existing account #Returns user attributes -Directs user to student page -Directs user to instructor page

	-forgetPassword()	-Directs user to password retrieval
Definition:	Manages the creation, deletion, and attributes of accounts.	

Class Name:	Class	
Attributes:	+name: string +term: integer #students: string #instructor: string #assignments	+Stores the name of the class +Stores the term of the class #Array of students of the class #Stores the name of the instructor #Stores an array of assignments
Operations:	-addClass() -deleteClass() -setClassAttribute() #getUserAttribute()	-Creates a new class -Deletes an existing class -Set attributes of existing class #Returns class attributes
Definition:	Manages the creation, deletion, and attributes of classes. Stores all students and assignments that are apart of the class.	

Class Name:	Assignment	
Attributes:	+name: string -weights: double #grades: double	+Stores name of assignment -Stores weights of assignment parts #Stores grades of assignment parts
Operations:	-addAssignment() -deleteAssignment() -setAssignmentWeight() #getGrade()	-Adds new assignment -Deletes existing assignment -Sets grades of assignments -Returns grades of assignments
Definition:	Manages the creation, removal, and scores of assignments	

Class Name:	Statistics	
Attributes:	#assignments #average: double #median: double #firstQuarter: double #thirdQuarter: double	#Stores all assignments in an array #Stores the calculated average #Stores the calculated median #Stores the calculated 25 percentile grade #Stores the calculated 75 percentile

		grade
Operations:	#createGraph() +sort() +calculateAverage() +calculatePercentile()	#Returns the image of the grade distribution +Sorts an array of integers +Returns the average of a given array of integers +Returns the percentile grade of a given percentage
Definition:	Manages and calculates the statistics of assignments.	

Class Name:	Chat	
Attributes:	-message: string -userType: boolean	-Stores the input message as string -Stores user type
Operations:	-screenShare() -messageList() -userType() -sendMessage() -TAlist() -tips()	-Allows the instructor to view the students screen -Displays the past 25 messages between a student and an instructor -sends input string -see available instructors -see helpful instruction for the lab
Definition:	Allows complete, real-time interaction and communication between student and instructor for assistance with the course material.	

Class Name:	LogIn Inquiries	
Attributes:	-username: string -password: string -email: string -securityQuestions: string -sqAnswers: string #emailVerification: boolean	-Stores a unique username -Stores a password -Stores a verified email -A string array of selected questions -A string array of answers to security questions #Boolean to check if the user has verified their email
Operations:	-forgotPasswordUI() -createSecurityQuestion()	-Displays the forgot password UI -Creates an array of security questions

	-askSecurityQuestion() -setEmailAddress() -setPassword() -setUsername() #sendEmail()	-Prompts user to answer security questions and stores the response -Set the current account's email address -Set the account's password -Set the account's username #sends an email to the user
Definition:	Guides the user through account management. Can email the user, verify the user's email, and store security questions/answers.	

Class Name:	Lab 3, Lab 4, Lab 5	
Attributes:	-LabName: string -LabInstructions: list -ExpectedOutput: GameObj	-Stores the name of the lab -Stores the instructions of the lab -Stores one solution of the lab
Operations:	+compareOutput(): output +saveProgress() +addGrade() : List	-Compares the circuit built by the student to the built in solution -Saves Student Progress -Adds the grade received for the lab to the gradebook
Definition:	Tells the Student what must be done on the lab. Is ale to record and log grades of students for labs 3, 4, and 5.	

Adjusted:

Class Name:	Prelab3, Prelab 4, Prelab 5	
Attributes:	-questions_answers: Map -question_randomizer: List	-Stores the answers to the questions asked -Will choose a fixed number of random questions from a question bank.
Operations:	+giveQuiz() +checkAnswers(): List +addGrade() : List	-Gives the quiz - Checks if the student has answered correctly -adds the grade to the grade book

Definition:	Pre-Lab quizzes to check student preparedness.

New:

Class Name:	Postlab3, Postlab 4, Postlab 5	
Attributes:	-questions_answers: Map -question_randomizer: List	-Stores the answers to the questions asked -Will choose a fixed number of random questions from a question bank.
Operations:	+giveQuiz() +checkAnswers(): List +addGrade() : List	-Gives the quiz - Checks if the student has answered correctly -adds the grade to the grade book
Definition:	Post-Lab quizzes to assess student's understanding of the lab.	

Traceability Matrix:

The Traceability Matrix listed below was split into separate tables to demonstrate the different domain concepts and displays a clear connection to each individual class that houses these concepts. The main reason we elected to isolate the tables was to make the table easy to read and cohesive as a chart.

Part 1:

Domain Concepts	**Classes**					
	LoginGUI	Database	StudentGUI	InstructorGUI	Chat	Report
Login	X					
Database Connection		X				
Admin Subsystem				X		
Student Subsystem			X			

Gradebook				X		X
Chat					X	
Statistics		X		X		X
Grades						X

Part 2:

Domain Concepts	**Classes**												
	Lab 1	Lab 2	Lab 3	Lab 4	Lab 5	Equipment manager	Q1	Q2	Q3	Q4	Q5	Screen Capture	Cheat Detection
Class Section				X	X								
Equipment				X	X	X							
Lab 4				X						X			
Lab 5					X						X		
Cheat Detection													X
Screen Capture												X	

Most of the Domain Concepts are mapped to a class mirroring our class diagrams. In part 2 of the traceability matrix we included all labs from the previous year as well as our two new incorporated labs 4 and 5. The columns with Q 1-5's stand for quiz 1-5's, we needed to reduce the length of the name to fit the space allowed. In part 1 of the traceability matrix we included Report as a class, for all concepts that need to record and relay important information to the students and instructors. This Report class encompasses mostly the grading, and statistical portion of the application, which is fitting for the description provided previously.

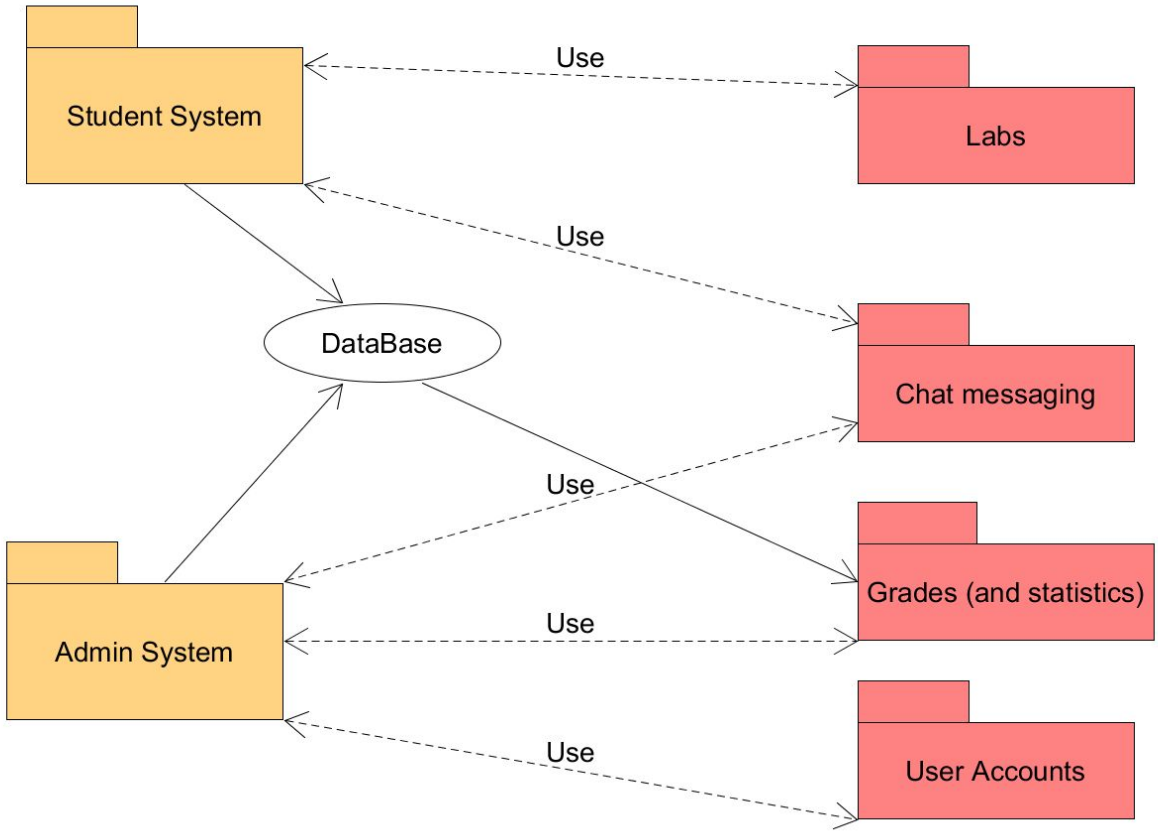
System Architecture and System Design

Architectural Styles:

The architecture style for our project is a mix of event-driven architecture with some essence of shared memory and database use. By definition the event driven architecture style is based on production, detection, consumption, and reaction to events. An “event” in this case is a state at which the program is running. More specifically the “event” could be something as simple as changing state from closed menu to a drop down menu. Most of the simulation for the virtual lab will be based off of interactive events where each button or draggable equipment accessed by any user, will trigger the state change and lead to some event that responds accordingly. This style suits this project the most because nearly all system procedures are reactions to the event changes initiated by the user. Event driven architecture is known to compliment the service oriented architecture in its network communication and use of servers/databases to send and receive information for events. This project would need the database centric architecture for events pertaining to user information. More specifically databases will be used, as a means to store information about each student and instructor, lab grades, quiz grades, usernames, and passwords.

The tool used to develop upon this architecture is the Unity Game Engine. The Unity Framework is based on an Event driven architecture. Most events in Unity are controlled ia a C# script and within each script is a Start method and Update method. These methods are what control the event loops. For more information on unity and its relation to this project’s architecture, we chose to look a its use of GameObjects and how these can be programmed as events. GameObjects are components in a project with specific properties and functions. The unity framework is focused facilitating the interaction between GameObjects. Each object can be controlled via a script. This script allows the object to extend or inherit properties as well as use a callback system whenever a user clicks a button or uses the mouse. This allows for a more cohesive simulation experience. Besides the support from developers and ease of use, this framework was chosen to extend the work of the prior group. They have provided the basics and the foundation, and we look to implement many modifications and improvements.

Identifying Subsystems:

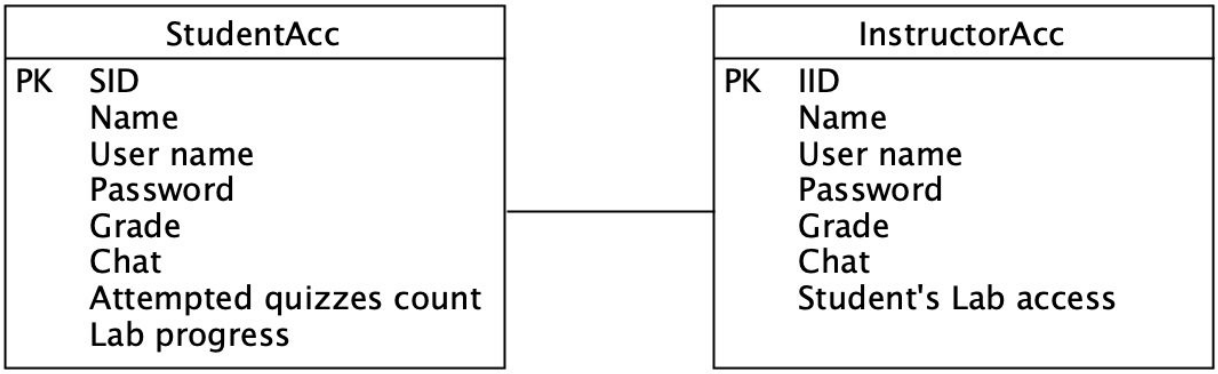


Description: The packages in the uml diagram are systems that are developed and organized in such a way so the communication between two main user systems is indirect and linear. Meaning there is no direct communication between users. Rather each package can run independent from one another.

Mapping Subsystems to Hardware:

Given that our subsystems for users will need to share and store information, we need to have client server interaction. More specifically the login data (for both student and admin), the grades, chat history, and lab instruction need to be accessed via a server based relational database. Besides the server hardware, the system is expected to run independently.

Persistent Data Storage:



The entire system is dependent on being able to save data that outlives a single execution of the system. From the moment you log in to the account, we have to keep a saved copy of your login information. Every event you take on the system will be recorded in the database to keep track of your progress in the course. Things from student lab progression and lab grades to things like chat history are stored. We use Amazon web service to store the data, which is a good reliable database

Network Protocol:
Currently in regards to the network protocol we are still attempting to figure out if last years Firebase SDK is the best connection between the C# code and the Unity Engine Framework. We are still looking through options however for now we will be sticking with the safety of last year's program. However this may be subject to change. Some of the other possible alternatives could be Parse which is an open source backend platform OR back4app which is a parse hosting platform.

Global Control Flow:

This system is event-driven. When the user logs in, they can do a few different things, such as start a lab, play around with the breadboard, or work on quizzes. The system might also be considered procedure driven due to the requirements of labs and quizzes. Students all have to do the same assignments in similar ways. The final circuits of the labs might look different, but they should ideally have the same results.
The system is also time dependent. The students are assigned the labs and quizzes, and they have to finish them in by the assignment’s due date. The labs themselves are not timed, but the quizzes are.

Execution Orderness:

Our system is event-driven, as users can create any circuit they wish to upon entering a lab or sandbox mode, and they are responsible for what they create. This means that there is more than one way to achieve a correct solution in a lab, as each logical component is consistent in its function, and can be built upon with other components. This means that the UI is also loop based, as it will update based on user action.

Timer Dependency:

The timer is used in two places. The first is the pre-lab quizzes, where the student can be timed if the instructor wishes, to ensure a good understanding of the material. If time runs out, the student will not be forced out of the quiz UI, but will receive a significant reduction in grade. This same concept applies to the lab, where the instructor can set an amount of time for the lab, and if the student does not complete the lab within this time limit, his/her grade will be substantially reduced.

Concurrency:

Students will be able to send messages while in the lab, without interrupting the lab itself. In addition to this, instructors will be able to view a students gradebook, even if the gradebook is updated due to the student completing a lab while the instructor is looking at it. It also means an instructor can add a student to a different class while the student is working on the lab without interruption.

Hardware Requirements:

OS support: Windows 7 SP1+, macOS 10.11+, Ubuntu 12.04+, SteamOS+
Graphics card: DX10 (shader model 4.0) capabilities
CPU: SSE2 instruction set support
Screen: minimum resolution of 720p
Minimum network bandwidth: 56k
Minimum hard drive space: 150 megabytes

Data Structures

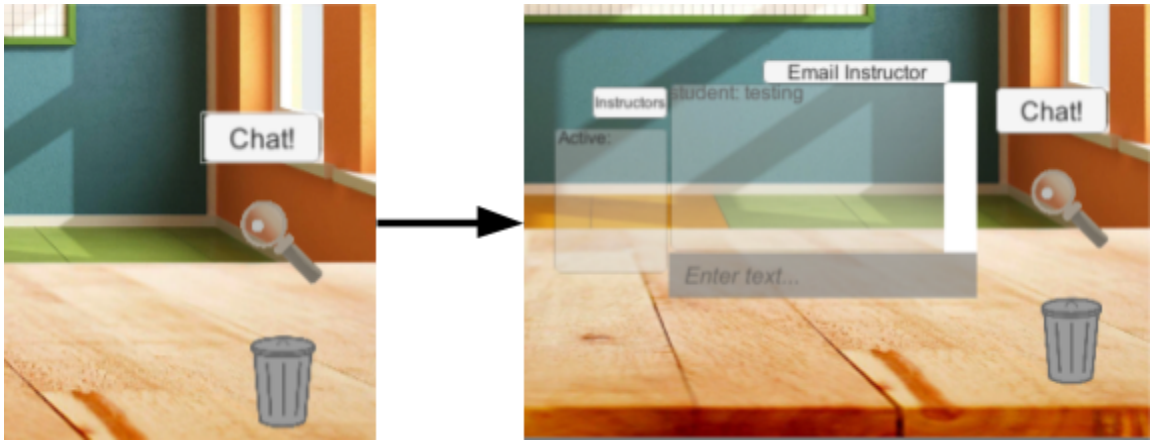
Data Structures:

1. We will be using a stack to save student grades for instructor viewing and will read/write from the stack storing names and grades when the instructor opens the menu to view/edit student grades.
2. We will be using an ArrayList to store classes under an instructors account. Arraylists and linked lists allow flexibility in the number of classes that an instructor can have. Our decision of Arraylists over linked lists stems from the time complexity of add, delete, and search functions. Searching for classes will occur much more often in our cases than adding or deleting classes. Since the time complexity of search for ArrayList is O(1), and O(n) for linked lists, we will use Arraylists to store classes.
3. We will also be using a linked list whenever a new account is created that the new account name and its password are referenced/connected when logging in. Essentially the username and password will be saved to a .txt file, but in order to fully connect the two from the login screen, they will need to refer to one another in our C# files.
4. For the chat system we will be using an Arraylist to maintain 25 viewable messages while the chat box is active. We plan to use a stack to add or subtract instructors to the instructor list as well.

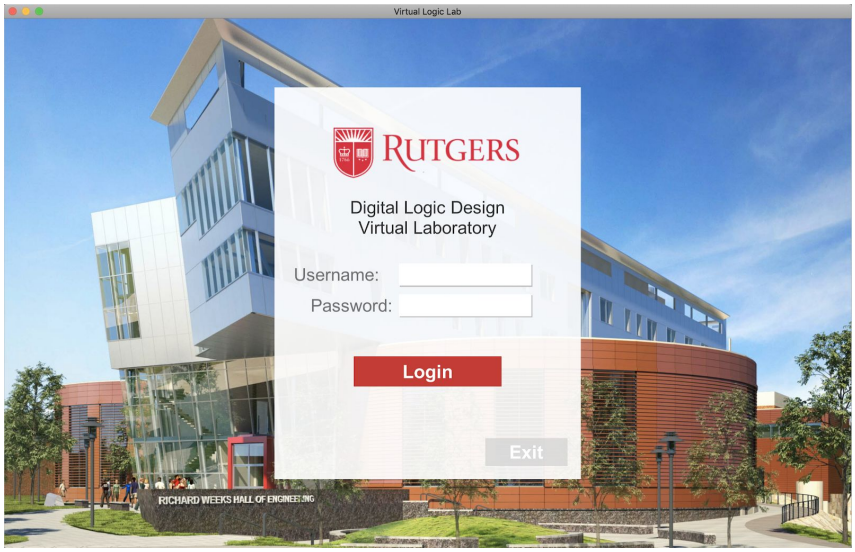
User Interface Design and Implementation

---

1. The initial screen mock-ups for the chat subsystem has changed slightly since the first report. Ease of use is slightly increased in that with one click the user should see the interface and all information necessary to input a message into the chat. In the future, while networking is currently under development, the user shall also see the list of available instructors that are online at that moment.

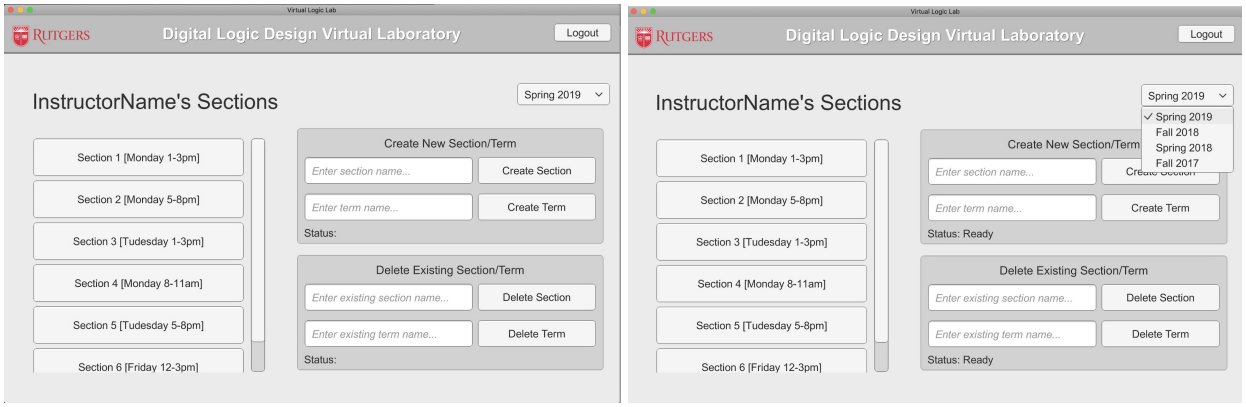


2. The login menu will only change slightly, with the addition of the option to create an account there. Also, there will be options added in the instructor menu to create classes and to view/edit student grades.

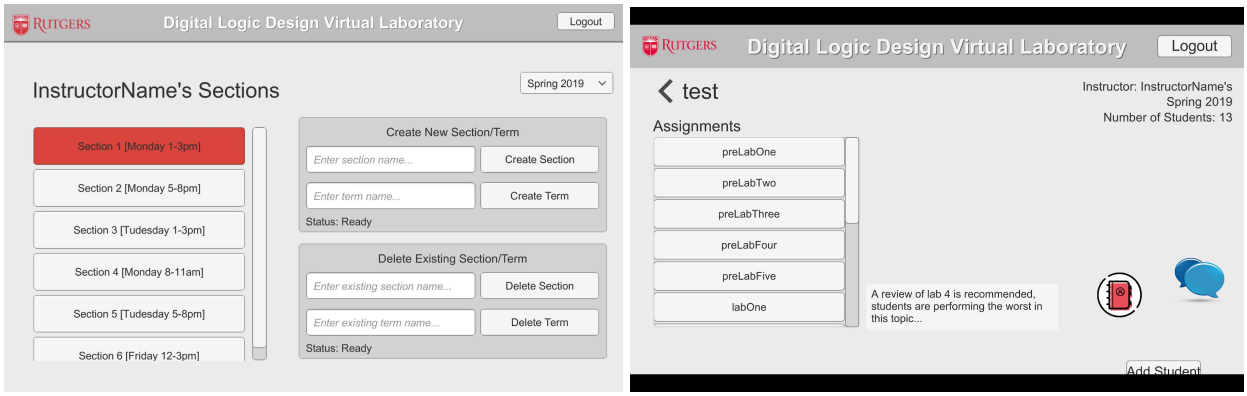


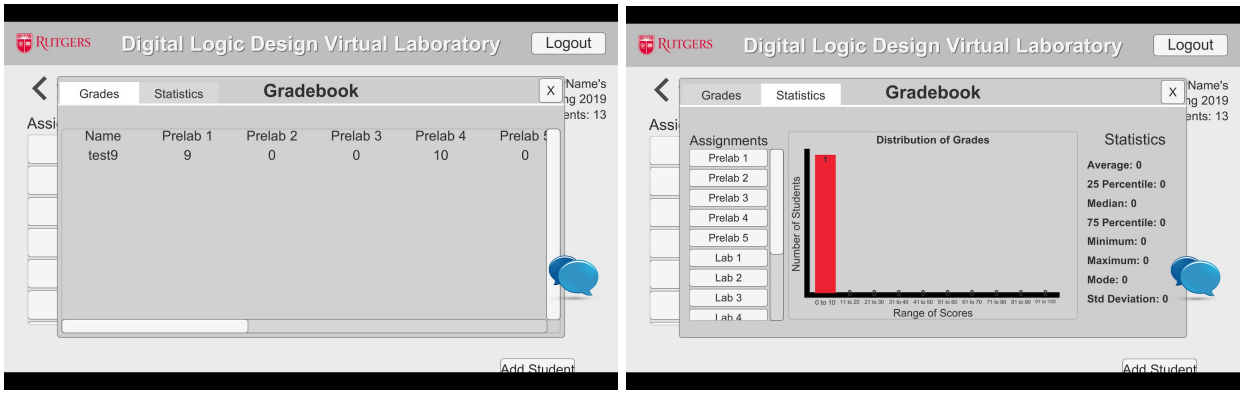
3. The instructor home page can be accessed with a verified instructor username and password. On the instructor page, the instructor can access class pages from the scrollable list on the left. On the right is the logout button, term dropdown selection, create

section/term buttons, and delete section/term buttons. When creating or deleting sections/terms, the instructor will receive feedback on their actions. Creating will result in a “success,” “Name already exists,” or “Please enter name” status. Deleting will result in a “success,” “Name does not exist status,” or “Please enter name” status.

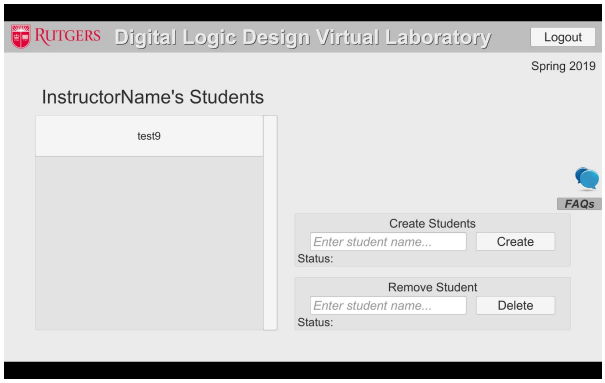


4. The class page can be accessed by clicking on a section from the list. On the class page there is a list of assignments, which will bring you to a grade list showing the grades of all students in the class on that assignment, shown in the picture in the bottom left. Clicking on the “Statistics” tab in the gradebook will show you the bar graph of all student scores for that assignment, as shown in the picture in the bottom right.

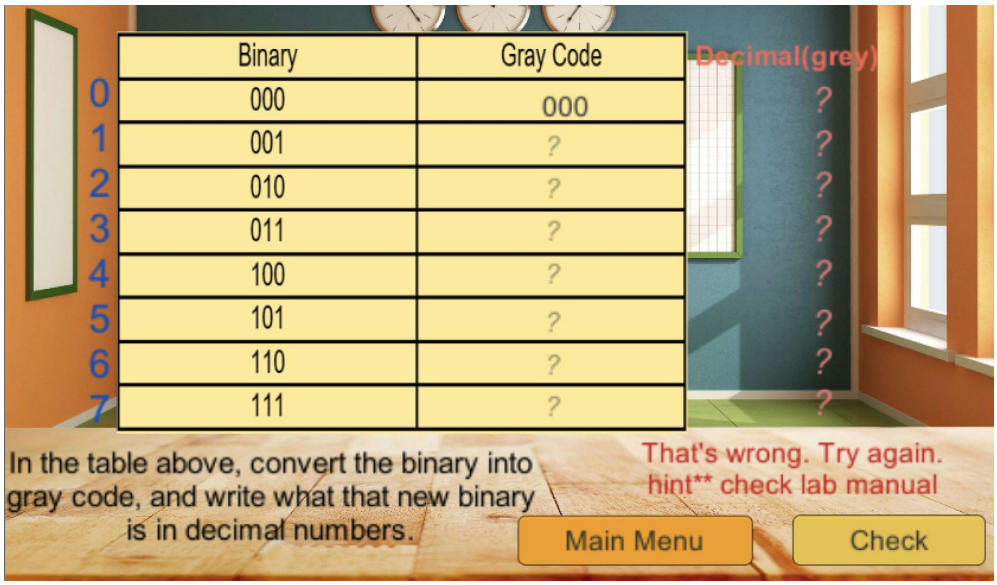




5. The Add Students page can be accessed by clicking on the “Add Student” in the bottom right of the class page. This will allow the instructor to add students by email to their section, and will cause their grades to be shown in the class page.



6. The prelabs inform you of your mistake(s) and gives hints to help you figure out where you went wrong.



Design of Tests

UI Testing

Action	Output
The user clicks on the Chat button to open the initial UI	SUCCESS: The user will see available options to chat (chat box, input fields, Active

	Instructor option, and Email) FAILURE: The button does not respond to the user's click event
The user Inputs a message into chat	SUCCESS: The user will be able to input a message into the chat system and see their respective user status (student or admin) displayed FAILURE: The user cannot type into the chat box. Message does not appear
The user clicks on Instructors list	SUCCESS: The list appears and the user can view the panel with active list FAILURE: The list does not appear, or the panel does not open
The user clicks to view student grades in instructor mode	SUCCESS: The gradebook opens showing student names and their respective grades. FAILURE: The gradebook does not open or does not match student names with grades successfully.
The user clicks to create an account	SUCCESS: The options to create an account will open. FAILURE: The list does not appear, or the panel does not open
The instructor clicks to create a class	SUCCESS: The menu to create a class appears, as does a list of students. FAILURE: The menu does not open.
The instructor clicks to delete a section	SUCCESS: Section with the name typed in by instructor gets deleted. FAILURE: Section does NOT get deleted or The wrong section gets deleted.
The instructor clicks to add a student	SUCCESS: The student gets added to the section.

	FAILURE: The student does not get added to the section
The instructor clicks to remove a student	SUCCESS: The student gets removed from the section. FAILURE: The student does not get removed from the section
The student clicks the “Finish” button when lab is completed.	SUCCESS: Students personal grade gets sent to gradebook and now has updated grade for the lab that was completed. FAILURE: Lab does not get graded and student can only return to main menu.
The instructor enters in a problem and solution into the FAQ page.	SUCCESS: FAQ page is updated with the question and solution that was entered. Students get updated FAQ page. FAILURE: FAQ Page fails to update.
The instructor loads the FAQ Page	SUCCESS: FAQ page connects to the database and shows all questions and answers that have been set by instructors FAILURE: FAQ Page fails to show all questions and answers.
The student loads the FAQ page	SUCCESS: FAQ page connects to the database and shows all questions and answers that have been set by instructors FAILURE: FAQ Page fails to show all questions and answers.

Project Management

Merging the Contributions from Individual Team members

Currently, since this is not the full final copy of our report, aside from our team members working to make sure our groups do not create repetitive interaction diagrams and descriptions there has not been much. However, as individual subsets, we have all began to work on our

respective responsibilities to the project. Be it brainstorming how to get one of the new use cases done or actual hard coding; each group is in different stages.

Project Coordination and Progress Report

The use cases that have currently been implemented include:

Use Case 1 - Login - Has been implemented already, although the previous group had this done, it was imperfect in the sense that none of the progress saved and there was only one student account. This issue has been fixed.

Use Case 3 - Management Equipment - Is currently being worked in the works. There was a glitch with the object selection. Whenever the user grabbed an object to use in the virtual lab, once it is placed, and the same object is to be used again, the user would have to deselect the said object and reselected it to use again. The wire object was our first focus because it will be the most used item; having to select a new wire every time would be tedious. This has been fixed, other materials will follow.

Use Cases 2 and 5 - Instructor Lab Control and ChatSystem - Are both the big portions of two subgroups, so they are about to end the brainstorming phase on how to actually tackle and implement this and begin coding it and running tests through it soon.

Use Case 11 - LogOut - Has also been implemented in relation to the first Use Case. The user can now log out from their account and be returned to the login screen.

While this may not be as much as we initially wanted, We feel that within the next two weeks a lot more will get done simply because the workload of this past week outside of this class was extremely hectic. In all honesty, to get this amount of work done with the limited time that we had is impressive in itself. Now that midterm season 1 has just ended, the priority of this project has shot right back up to where it was initially and will get much work done very soon. Regardless, this being Part 1 of the Second report, we would hope it is not too bad of a problem only because this is a mere fraction of what we expect the final product to be.

The process for this upcoming week will include going one by one down the UC list and trying to tackle them as a unit unless they are part of a specific subgroup, in which case, it would be the group's responsibility.

What is functional? What is Being tackled

As we are closely approaching the second demo, much has changed since our first demo. The addition of the database and connection to a network allows for a lot of incorporation of everybody's ideas. Creating the scenes on Unity for each individual group is not the most difficult in the world, however being able to connect these scenes into the database and allowing information to be passed from one person's application to another is crucial for a lot of our ideas. Our prototype is still functioning and nothing has broken. We expect the incorporation of the gradebook and chat system to be fairly straightforward as well as create new labs for the software itself. We were given great suggestions like a FAQ section for lab questions which we also take into consideration after our skeleton is finished.

History of Work

Christopher Basilio and George Melman-Kenny

Throughout this project, there were many accomplishments in terms of account management. After the initial database was connected, we were able to constantly connect many parts of this application to it. From the FAQ page, to the grading, to the section creation, and login details. Everything was manually setup into the database and made so that the application would have to pull from it to work. This succeeded in making the entire project a lot more legitimate/viable for actual in school usage. The development of the partial grading was also a responsibility that was taken on. George focused on Lab 1 while Christopher focused on Lab 2. Both labs are fully functional and susceptible to the partial grading system that was set up.

James Ramos and Daniel Chan

We are responsible for working on implementing the Chat subsystem; mostly GUI development and design tailored for each lab in the student subsystem and the admin system. A main feature is to allow the student to input a string into the the input fields and have their corresponding username display alongside their message. Another implemented feature is chat-history which allows the chat to save up to 25 messages without using extra space. Each lab now has a chat button, that when clicked, will allow the student to enter a message to send to the instructor. We also implemented a tips feature, tailored for each lab. The goal of this feature was to provide the user with a subtle hint in the event they are stuck on the lab. This will reduce the required amount of communication needed between student and instructor. We also incorporated a screenshot feature, where the user presses "/" and "s" on their keyboard and it saves the picture to their files, which they can then email to their instructor. In the event the student has a bigger concern, we also implemented a smaller feature (button) that the user can use to email the professor/instructor directly. In addition to the chat subsystem we also worked on the general about page for students that provides ample information about running the simulation as well as

general course information. We successfully implemented ways to mitigate the need to use the chat box with the help of the FAQ and About pages. The student can now use these resources to answer many questions that are asked by other students. If the questions are more specific, there is also the Tips feature to guide them through the lab experiment if needed. We also implemented the entirety of Lab 4, which includes both a prelab and postlab quiz, as well as the lab itself. The XOR gate that was created by the previous group was required for this lab, however, there were was a bug in the error that made it so the chip did not produce any output for some of the gates. We fixed this bug and included the chip in the lab.

Andy Lee and Drew Koskinen

We have implemented the multiple class system and class page. The multiple class system allows instructors to manage different sections of labs during the semester. Instructors are able to create and delete sections. The multiple class feature will allow instructors to have a single account for all of their classes. Each section will have its own class page. The class page has been created, but new features such as the gradebook will be implemented soon. We planned to have the multiple courses feature finished and start the class page for demo 1, and were successful. Our next steps are to connect with the database so that we can implement the course statistics feature. Along with the course statistics feature, we will work on implementing more labs for students.

Willear Glimniene and Deeptanshu Murdeshwar

In the beginning, time was mainly utilized for familiarizing ourselves with Unity and exploring the project. Then, we updated the grading system to allow the user to reattempt the prelabs before doing the main lab. This gives the student the chance to correct their work and understand their mistake. The student will be shown which answer is wrong but not how many so they must check each value in the truth table to verify accuracy. Students will lose 1 point for each incorrect attempt. This grading system was implemented into prelab 1 and 2. We also created prelab 3 as a prelude to Lab 3. Prelab 3 includes this grading system. Then, we added a practice quizzes section which includes 4 timed quizzes. This can be accessed directly from the student subsystem's main menu.

List of Accomplishments:

- Increased menu usability
- Created Account Login System
- Created Account Creation System
- Added basic security features to the Database

- Created the base of the chat system
- Created about page for more information regarding the simulation/course logistics
- Created Tips feature in every lab to reduce communication required between instructor and student.
- Created the Section creation feature
- Created the Section delete feature
- Created the Class page
- Created multiple class per instructor account feature
- Created quizzes section with time limit
- Added a frequently asked questions page for both instructor and student. The page can be modified by the instructor (can add or remove questions).
- Added and implemented the logic necessary to run through lab experiment 4. More specifically half adder logic using AND and XOR gates.
- Added partial grading for all of the lab experiments.
- Added prelabs and postlabs that also utilize the partial grading feature.

Current Gantt Chart

Future of Work

The project itself has a good base and structure. For the future of this project, new features could be implemented as well as the current features could be tweaked and polished. For the Chat System, we were unable to implement the screen sharing feature given that the focus later shifted to adding more lab experiments and implementation. As for labs, the current labs could be polished, like for instance the grading aspect for each of the labs could be perfected or the prelabs and postlabs could be made into something more in depth than it already is. More labs could be implemented as well as more equipment like transistors or even an oscilloscope. The entire project itself can be adapted into other lab classes in the ece department such as Principles of Electrical Engineering I and II, Electronic Devices, Digital Electronics, etc. This would require the implementation of transistors, resistors, capacitors, oscilloscopes, etc. as well as the behaviour of these devices. These devices behave a lot differently than the basic logic of the circuits in our labs.

Breakdown of Responsibilities

Christopher Basilio and George Melman-Kenny

Programming:

Currently we are responsible for editing the Login UI to make it more accessible to people. This

includes fixing a “back to main menu” button that was incorporated incorrectly. As well as create a method that will allow the creation of new accounts with ease rather than physically having to right login information into a txt doc, we want to make it easier for a user to do it within the .exe itself if possible.

In addition to this we are also responsible for connecting this project to the internet, allowing multiple users to be online and interact with the software itself. We’ll begin by trying to get the gradebook to work, and from their see how to incorporate the chat system. (UC -1, UC-5, UC-11)

Update: We have successfully connected the project to a database and have tested the usage of it through the creation of different login accounts. Next up would be to incorporate the chat system onto the database as well as create an easily accessible gradebook through the database. This will be a fairly straightforward task now that the the actual connection has been made.
Aside from this we have began the creation of one of the labs, which is the responsibility of each of the groups in this project.

Andy Lee and Drew Koskinen

Programming:
Responsible for the instructor subsystem and class statistics. The instructor subsystem includes the “Class” and “Term” classes. These classes will allow the instructor to create, delete, and go to different class sections for their laboratories(UC-8, UC-10, UC-11). The class statistics portion will allow the instructor to view grades or assignments and partial credit towards said assignments. Class statistics will allow the instructor to analyze the level of progress that each section of their classes are performing at(UC-6, UC-7). A major part of our subsystems utilize the graphical user interface, we will take advantage of Unity’s resources for this. To utilize the class statistics functions, a connection to a database must first be implemented. Once a database is created, we will have to establish a connection to retrieve data.

Update:

We were able to implement the class page along with many of the class statistics’ core features. The class page now has a gradebook panel that has a gradebook tab and statistics tab. The gradebook tab is a scrollable list that contains student’s grades for each assignment. The statistics tab shows the distribution of grades for a particular assignment in the form of a bar graph. The statistics tab also has calculated statistical variables for the selected assignment. Each subgroup was also responsible for implementing a new lab, we were responsible for implementing lab 3. As well as designing new sprites implemented to lab 3 and various Logic gate bug fixes associated with lab 3.

Willear Glimniene and Deeptanshu Murdeshwar

Programming:

We are responsible for creating a smart-grading system for the quizzes/prelab and labs. This system needs to be able to award partial credit for students' grades. We are also looking to implement a way for instructors to create their own assignments and publish them. These assignments are then to be graded by the grading system. The students will be able to attempt the quizzes in order to test their knowledge. (UC-13, UC-14, UC-17). We will complete these tasks through Unity and scripts will be written in C# programming language.

We plan on also implementing an "About" page to hold miscellaneous information regarding the lab course. Though this is not a use case, this can be useful for users looking for more information.

Project Management:

Shared responsibility for making sure that the project schedule is followed and that deadlines are met.

References

1. Professor Marsic's Website, Report 3 of 2018 Virtual Lab attempt
https://www.ece.rutgers.edu/~marsic/books/SE/projects/ViBE/2018-g10-report3.pdf

2. Professor Marsic's Website, Software Engineering Report
https://www.ece.rutgers.edu/~marsic/Teaching/SE/report1.html

3. Professor Marsic's Website, Digital logic design Lab
https://www.ece.rutgers.edu/~marsic/Teaching/DLD/lab.html

4. UML Basic notations
https://www.tutorialspoint.com/uml/uml_basic_notations.htm

5. UML Class Diagrams
https://www.uml-diagrams.org/class-reference.html

6. Hardware requirements for unity Player

https://unity3d.com/unity/system-requirements

7. Unity DataBase Connections
https://www.youtube.com/playlist?list=PLTm4FjoXO7ndjPE8JXrJ9MjMfsJv956Qm
https://www.youtube.com/playlist?list=PL5KbKbJ6Gf99mcmE1ptsn0oXO1_vnKDlS