Software Engineering
Group Number Five
The Rutgers Virtual Biology Laboratory
Website: http://sjlu.github.com/Virtual-Biology-Lab
April 27th, 2012

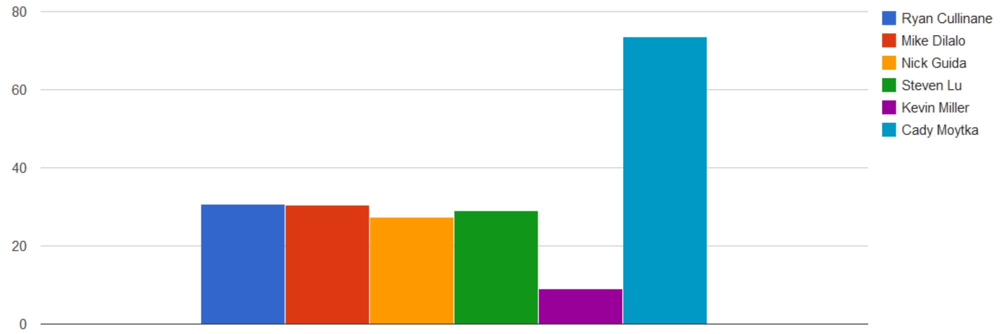
Ryan Cullinane, Michael DiLalo, Nicholas Guida, Steven Lu, Kevin Miller, Cady Motyka

Table of Contents

Participation Matrix 4
1. Summary of Changes 5
2. Customer Statement of Requirements 6
  2.1 Goal of Each Laboratory 6
  2.2 Unique Features 11
3. Glossary of Terms 11
  Laboratory 1 11
  Laboratory 3 13
  Laboratory 4 14
4. Functional Requirements Specification 15
  4.1 Enumerated Functional Requirements 15
  4.2 Stakeholders 15
  4.3 Actors and Goals 16
  4.4 Use Cases 19
    4.4.1 Casual Description 19
    4.4.2 Use Case Diagram 19
    4.4.3 Fully-Dressed Description 20
    4.4.4 Use Case Traceability Matrix 27
  4.5 System Sequence Diagrams 28
  4.6 Activity Diagrams 41
5. Nonfunctional Requirements 54
6. Effort Estimation using Use Case Points 54
  6.1 Unadjusted Use Case Points (UUCP) 55
    6.1.1 Unadjusted Actor Weight (UAW) 55
    6.1.2 Unadjusted Use Case Weight (UUCW) 55
  6.2 Technical Complexity Factor (TCF) 56
  6.3 Environmental Complexity Factor (ECF) 57
  6.4 Effort Estimation 58
7. Domain Analysis 58
  7.1 Domain Model 58
    7.1.1 Concept Definitions 58
    7.1.2 Association Definitions 61
    7.1.3 Attribute Definitions 64
    7.1.4 Traceability and Domain Model Diagrams 67
  7.2 System Operation Contracts 69
  7.3 Mathematical Model 73
8. Interaction Diagrams 74
  8.1 Interaction Diagrams 74
    8.1.1 Laboratories 1 and 4 74
    8.1.2 Laboratories 2 and 3 78
  8.2 General Interaction diagrams 87
9. Class Diagram and Interface Specification 91
  9.1 Class Diagram 91
  9.2 Data Types and Operation Signatures 93
  9.3 Traceability 96
10. System Architecture and System Design 99
  10.1 Architecture Styles 99
  10.2 Identifying Subsystems 99

10.3 Mapping Subsystem Hardware 102
10.4 Persistent Data Storage 102
10.5 Network Protocol 103
10.6 Global Control Flow 103
10.6.1 Execution Orderness 103
10.6.2 Concurrency 103
10.7 Hardware Requirements 103
11. Algorithms and Data Structures 103
11.1 Algorithms 103
11.2 Data Structures 105
12. User Interface Design 106
12.1 Preliminary Design 106
12.2 User Effort Estimation 111
12.3 User Interface Design and Implementation 113
13. Design of Tests 120
13.1 Unit Testing Test Cases 125
13.2 Test Coverage 128
13.3 Integration Testing Strategy 129
14. Project Management and History of Work 131
14.1 Merging the Contributions from Individual Team Members 131
14.2 First Iteration Milestones 131
14.3 Second Iteration Milestones 133
15. References 136

Participation Matrix



Responsibilities	Ryan Cullinane	Mike Dilalo	Nick Guida	Steven Lu	Kevin Miller	Cady Moytka	Totals	Possible Points
Totals:	30.65	30.45	27.35	28.95	9	73.6	100%	200
Project Managment	20%	20%	20%	20%	0%	20%	100%	13
1. Summary of Changes	10%	10%	10%	10%	10%	50%	100%	5
2 Customer Statement of Requirements	10%	10%	0%	0%	0%	80%	100%	6
2.1 Goal of Each Laboratory	0%	0%	0%	0%	0%	100%		
2.2 Unique Features	25%	25%	0%	0%	0%	50%		
3 Glossary of Terms	0%	0%	0%	0%	0%	100%	100%	4
4 Functional Requirements Specification	10%	20%	5%	7%	4%	54%	100%	30
4.1 Enumerated Functional Requirements	0%	0%	0%	0%	0%	100%		
4.2 Stakeholders	0%	0%	0%	50%	50%	0%		
4.3 Actors and Goals	0%	0%	0%	50%	50%	0%		
4.4.1 Casual Description	0%	0%	0%	0%	0%	100%		
4.4.2 Use Case Diagram	0%	0%	0%	0%	0%	100%		
4.4.3 Fully-Dressed Description	5%	20%	0%	0%	0%	75%		
4.4.4 Use case Traceability Matrix	0%	0%	0%	0%	0%	100%		
4.5 System Sequence Diagrams	30%	10%	0%	0%	20%	40%		
4.6 Activity Diagrams	0%	0%	50%	0%	0%	50%		
5 Nonfunctional Requirements	30%	0%	0%	70%	0%	0%	100%	6
6 Effort Estimation using Use Case Points	0%	0%	100%	0%	0%	0%	100%	4
7 Domain Analysis	0%	25%	25%	10%	0%	40%	100%	25
7.1.1 Concept Definitions	0%	10%	0%	0%	0%	90%		
7.1.2 Association Deffinitions	0%	0%	10%	0%	0%	90%		
7.1.3 Attribute Definitions	0%	50%	50%	0%	0%	0%		
7.1.4 Traceability and Domain Model Diagram	0%	0%	0%	0%	0%	100%		
7.2 System Operation Contracts	0%	50%	50%	0%	0%	0%		
7.3 Mathematical Model	0%	0%	0%	100%	0%	0%		
8 Interaction Diagrams	0%	40%	0%	0%	10%	50%	100%	30
8.1 Interaction Diagrams Labs 1 and 2	0%	90%	0%	0%	0%	10%		
8.2 Interaction Diagrams Labs 2 and 3	0%	10%	0%	0%	0%	90%		
8.3 Interaction Diagrams General	0%	0%	0%	0%	100%	0%		
9. Class DIagram	10%	20%	0%	0%	10%	60%	100%	10
9.1 Class Digrams	20%	20%	0%	0%	0%	60%		
9.2 Data Tpes and Operation Signatures	0%	10%	0%	0%	0%	90%		
9.3 Traceability	0%	0%	0%	0%	0%	100%		
9.4 Design Patterns	0%	0%	0%	0%	0%	100%	100%	10
9.5 Object Constraint Language Contracts	100%	0%	0%	0%	0%	0%	100%	10
10. System Architecture and System Design	5%	0%	0%	95%	0%	0%	100%	15
10.1 Architecture Styles	0%	0%	0%	100%	0%	0%		
10.2 Identifying Subsystems	100%	0%	0%	0%	0%	0%		
10.3 Mapping Subsystem Hardware	0%	0%	0%	100%	0%	0%		
10.4 Persistent Data Storage	0%	0%	0%	100%	0%	0%		
10.6 Global Control Flow	0%	0%	0%	100%	0%	0%		
10.7 Hardware Requirements	0%	0%	0%	100%	0%	0%		
11 Algorithms and Data Structures	0%	0%	0%	30%	70%	0%	100%	4
11.1 Algorithms	0%	0%	0%	40%	60%	0%		
11.2 Data Structures	0%	0%	0%	40%	60%	0%		
12 User Interface Design and Implementation	90%	0%	0%	10%	0%	0%	100%	11
13 Design of Tests	0%	0%	100%	0%	0%	0%	100%	12
14 History of Work and Current Status of Implementation	10%	10%	10%	10%	10%	50%	100%	5
15 Conclusions and Future Work, min 2 Pages	10%	10%	20%	25%	10%	25%	100%	
16. References	0%	0%	0%	0%	0%	100%	100%	

1. Summary of Changes

1. Renamed all of the figures, cited them in the references section and inserted internal citations.
2. Changed the customer statement of requirements to a) better reflect the importance of the biology and how the system interacts with the students and b) be named in order of their importance, so that the more important requirements have lower numbers. Also, changed the documentation so that they were always referred to as "REQ1" and so on.
3. Changed most of the use cases, and enumerated them in the casual descriptions section. We also made sure every use case is described and labeled here. Also, added these new use cases to the fully dressed descriptions.
4. Made corrections to the pre/post condition in the fully dressed descriptions of the use cases.
5. The test cases were moved from section four to section thirteen.
6. All of the diagrams were adjusted to account for the new use cases
7. The biology related domain model concepts were added to the use case-domain model tractability matrix.
8. New domain model diagrams were drawn.
9. Descriptions of the design principles used were added to the interaction diagrams.
10. The new sections effort estimation, design principles and the OCL contracts were added
11. Descriptions of the design principles used for the class diagrams were added.
12. Corrections were made to the system architecture section.
13. Adjusted the Design of tests to account for the new use cases .
14. Finished the user effort estimation sections.
15. The user interface design section was updated
16. History and future work section was updated.

2. Customer Statement of Requirements

The goal of the Virtual Biology Laboratory is to reinforce the concepts taught in a introductory biology course by allowing each student to complete an informative and and uniform laboratory. Each student will have to prepare slides and use machines to complete the laboratory as if they were completing all of the tasks in a real laboratory setting. The laboratories will focus on the subjects of mitosis, urinalysis, enzyme activity, chromosome structures, meiosis and genetics. While doing research on what the customer would be potentially interested in, it was discovered that the clearest way of presenting this information is to first show a diagram that represents the cell activity and then present a picture of a real cell and how that looks during the activity; this was the students will have an idea of what processes are going on and know how this is visible in a real cell.

2.1 Goal of Each Laboratory

<u>Laboratory One Cell Division</u>

The goal of the first Laboratory is to demonstrate all of the stages of the type of nuclear division known as Mitosis. This lab will start with a demonstration of all of the steps of Mitosis, this starts with the Interphase stage where the cell is preparing for division, the cell increases in size and duplicates its chromosomes. The next stage is Prophase, this is when the chromosomes begin to bound together at their centromeres. During this stage the nuclear membrane disintegrates too. Next is the Metaphase where the chromosomes line up on the equatorial plane of the cell. Next is the Anaphase where the chromosomes are pulled apart and to opposite sides of the cell. This is where the cell has separated into two identical copies of genetic material. The last stage is the Telophase, this stage is the reversal of prophase and the two newly created cells settle down and a new nuclear membrane forms.

The second part of this lab is going to involve the student preparing a slide to view the stages of mitosis like how they would in a real laboratory. The last part of this lab is going to show the student what the differences are between an animal cell and a plant cell. An animal cell is irregular shaped and has only a cell membrane while a plant cell is a fixed shape and has both a cell wall and a membrane.

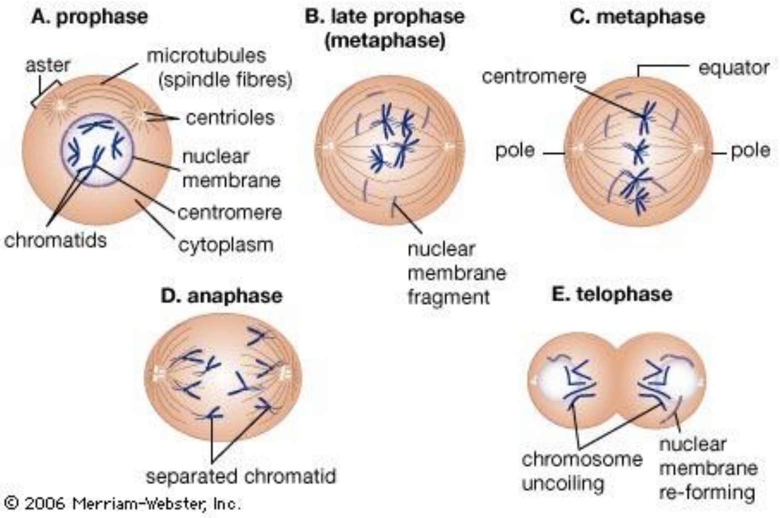
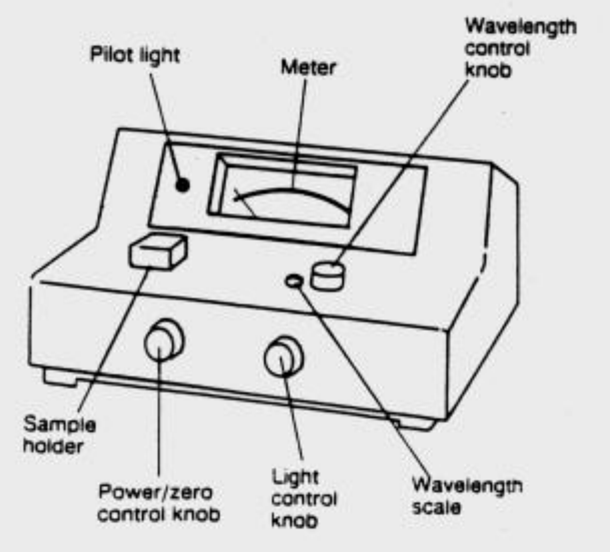


Fig. 2.1 This image shows the staged of mitosis that will need to be animated in order to explain in detail how each step occurs.

Laboratory Two Biological Molecules

The goal of the second laboratory is to learn how to use and calibrate the spectronic 20 spectrophotometer to figure out the absorbance of a substance. The absorbance is a measure of what size wavelength of light will pass through the test tube full of liquid without being absorbed. This machine uses a light and dark knob to set the zero and infinity ends of the absorbance scale. Once the machine has been set correctly using a 'blank' test tube, usually filled with distilled water, any test tube can be placed in the sample slot and the machine will immediately show the absorbance.

Once the basics of this machine are taught, the student will begin running urinalysis. First they will check for the pH of the urine, making sure it is between a 5 and an 8. Secondly they will check for Glycosuria, making sure the amount of glucose per milliliter is about 0.6. Next, they will ensure the patient doesn't have proteinuria, or an excess of 0.3 mg of albumin per milliliter. Lastly they will check for the color of the urine, making sure the patient doesn't have hemoglobinuria. The last part of this laboratory will involve the students experimenting with carbohydrate and protein chemistry. in this part the student will be given a prepared test tube of some mixture an have to use the Spec 20 to figure out what the concentration is based on some example readings.



The image to the right (**Fig. 2.2**) shows whats the spectronic 20 spectrophotometer looks like and how the buttons and screen are positioned on the machine. Most importantly is the sample holder on the top left side where the student will place the prepared test tubes to find their absorbance.

Laboratory Three Enzyme Activity

The third laboratory is similar to the second laboratory. This lab also uses the Specronic 20 to quantify enzyme activity under different conditions. Students in this lab will have to prepare 6 test tubes each with a different combination of buffer and enzymes. Once all of this mixtures have been prepared, the test tubes absorbance is found using the spec 20. Depending on which test tube the student is testing, there will be a very different reading of absorbance and this will illustrate how different types of reagents and substrates will react.

The second part of this lab involves the students preparing six test tubes again, each with a different amount of undiluted enzyme from 100 percent to six percent. Once this has been completed, the substrate is added to begin the reaction. The spec 20 will be used to find the absorbance initially and then again after five minutes. This will allow the student to see how the concentration of the enzyme affects how quickly the reaction begins and how long it takes to complete.

Laboratory Four Chromosome Structure, Meiosis and Genetics

The fourth laboratory is like and extension of the first lab. The focus is more on genetics and how genetic material is passed from parent cells to the daughter cells. A human has 48 chromosomes per somatic cell. Without meiosis, when a sperm and egg fuse to form a single zygote, there would be 4 copies of each chromosomes.

Meoisis ensures that when the gamete contains one representative of each homologous pair so that when they fuse, the resulting cell has the correct number of chromosomes In meiosis, the DNA is synthesized only once, but divides twice. It has the same same steps at mitosis, prophase, metephase, anaphase and telophase. Each of these stages occurs twice so there is, for example, meaphase I and metaphase II. In meiosis, prophase does not only include the chromosomes bounding together but also: Leptotene, Zygotene, Pachytene, Diplotene and Diakinesis. During lepotene the chromosomes become more visible and condense. Then in zygotene the homolgous chromosomes pair up. In pachytene the chromosomes form a tetrad and are fully lined up along their length. In diplotene the chromosomes begin to repel and finally in diakineses, the chromosomes become even shorter and the nuclear envelope breaks. This laboratory is going to show all of these steps to meiosis and how important it is to the development of living things.

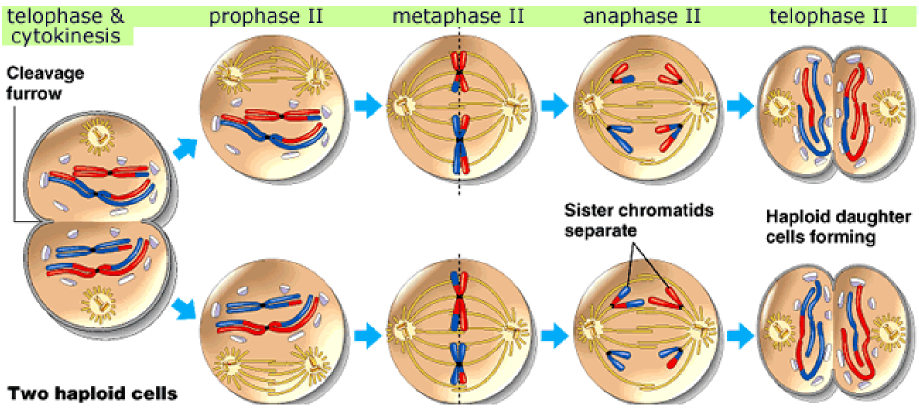


Fig. 2.3 This image above shows the different stages of meiosis. Just like with the staged of mitosis, this laboratory will have to animate exactly what happens during each of these phases.

In the second par of this laboratory, the students will be given a hypothetical chromosome spread. They are going to have to identify and number each of these chromosomes checking for any abnormalities. If there is an issue with the sex chromosomes there are four possible syndromes, either Turner syndrome, for an XO pair, Tripple X, Klinefelter syndrome for XXY or and XYY karyotype. Then the student would look for abnormalities in other chromosomes to indicate either Patau, Edwards and Down syndrome.

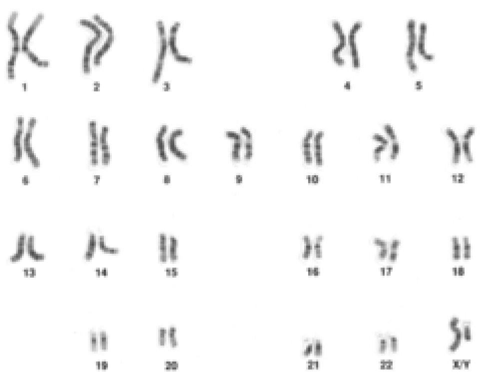


Fig. 2.4 The image to the left is that of a hypothetical karyotyping sheet similar to the one the students will have to check for chromosome abnormalities.

Below are four tables that explain in great detail what steps the user must take to complete each laboratory. These steps must be taken in the correct order in order for the experiment to be fully completed. The system will guide the user through all of these steps with enough freedom for them to make mistakes, but not with so much freedom that the entire laboratory could be done incorrectly. In some experiments, like the different tests in laboratory two, the user will be able to complete two to three steps after a mistake before the system indicates to them in some way that a mistake has been made. For other experiments, like the stages of mitosis in laboratory one, the system will instantly tell the user that they have made a mistake and will call for the student to fix that before they can move onto the next step. There will also be check points within the laboratory so that if a student decides to log out in the middle of an experiment, they will only have to complete from the last check point on and not begin again from the start.

The Following four charts show exactly what steps must be taken by the user when implementing the laboratories. These charts also shows the places where possible check points can be placed in the laboratories by dividing them into parts. If the laboratories are separated into parts, then a student can stop, log out and return later to complete a lab without having to restart it from the begining.

Laboratory One: Cell Division
Prepare slide: 1. Remove bottom 2 to 3 mm of the the onion root. 2. Place 2-3 drops of dye solution over the tip. 3. Heat gently on warm hotplate for one minute. 4. Press down on the cover slip, squashing the sample. 5. Look at slide and compare to sample to figure out which stage of Mitosis it is in. View stages of Mitosis and types of cells: 1. Drag components to show Interphase 2. Drag components to show Prophasee 3. Drag components to show Metaphase 4. Drag components to show Anaphase 5. Drag components to show Telophase 6. Show Plant Cells, Have user label parts 7. Show Animal cells, Have user label parts

Laboratory Two: Biological Molecules
Testing for ph balance 1. Place 9 squares of pHYDRION paper in a row 2. One drop of pH 7.0 on one square, pH 4.0 on four squares and pH 10.0 on four squares. Compare all of these colors to the guide. 3. One drop of each persons urine on each of the 6 square left. Record results. Assay for Glycosuria 4. Make BLANK test tube with 1 mL of water and 1 mL of DNS 5. Make STANDARD test tube with 1 mL of glucose and 1 mL of DNS 6. Make 6 test tubes each with 1mL of a patients urine and 1 mL of DNS 7. boiling bath for five minutes 8. Add 8 mL of distilled water and seal each tube. 9. Set Spec20, set at 540 nm, and find the absorbance of each tube. Record Results Assay for Proteinuria . 10. Add 1 mL of each patients to 6 different test tubes

11. Add 4 mL of biuret reagent to each.
12. Set Spec20, set at 540 nm, and find the absorbance of each tube.
13. Compare these results to the graph of the absorbance of different amounts of protein

Assay for Hemoglobinuria

14. observe each sample for brownish-red presence, or hemoglobin

Laboratory Three: Enzyme Activity

1. Prepare BLANK test tube
2. Add 4 mL of buffer and 1 mL of pNPP solution to test tube 1
3. Add 4 mL of buffer to test tube 2
4. Add 3 mL of buffer and 1 mL of pNPP solution to test tube 3
5. Add 2 mL of buffer and 1 mL of pNPP solution and sodium to test tube 4
6. Add 2 mL of buffer and 1 mL of pNPP solution and inorganic phosphate to test tube 5
7. Add 2 mL of buffer and 1 mL of pNPP solution and phenyl phosphate to test tube 6
8. Start reaction by adding 1.0 mL of enzyme to tubes 2 trough 6. Cover and mix.
9. Place tube 3 in the Spec 20, at 415 nm, and wait till the absorbance equals between 3.0 and 4.0, record the time that has elapsed.
10. then quickly read all the other tubes from 6 to 1, rereading tube 3
11. Prepare BLANK test tube
12. Add 2.0 mL of undiluted enzyme into test tube 1.
13. Add 1 mL of distilled water and 1 mL from test tube 1 into test tube 2.
14. Add 1 mL of distilled water and 1 mL from test tube 2 into test tube 3
15. Add 1 mL of distilled water and 1 mL from test tube 3 into test tube 4
16. Add 1 mL of distilled water and 1 mL from test tube 4 into test tube 5
17. Add 1 mL of distillled water to test tube 6
18. Add 3.0 mL of buffer, pH 8 to each test tube (except blank), mix thoroughly
19. To start the reaction: add 1.0 mL of substrate to each test tube and mix thoroughly
20. Immediately determine absorbance A using the spec 20 at 415 nm
21. redetermine the absorbances for each tube after 5 minutes

Laboratory Four: Meiosis

1. Animate Prophase I: Leptotene, Zygotene, Pachytene, Diplotene, Diakinesis
2. Transition to Metaphase I
3. Transition to Anaphase I
4. Transition to Telophase I
5. Transition to Prophase II
6. Transition to Metaphase II
7. Transition to Anaphase II
8. Transition to Telophase II

VBL will provide you with a hypothetical chromosome spread

9. Student clicks to circle every chromosome
10. Identify the chromosomes from the guide, and write the number under it
11. Identify if there are any abnormalities listed below

• XO = Turner Syndrome • XXX = Tripple-X • XXY = Klinefelter Syndrome • XYY = XYY Karyotype	• Trisomy 18 - Edwards Syndrome • Trisomy 21 - Down Syndrome • 14/21 translocation - normal carrier for down syndrome/inherited down syndrome

• Trisomy 13 - Patau Syndrome	

2.2 Unique Features

The most important goal of the virtual biology laboratory is to reinforce what the students are learning in their lecture. Second to that, this program is going to provide another opportunity for the students to show the instructor what sort of effort they are putting into learning this information. Since everyone will have to register to use this, the professor will have a list of who has worked on the laboratories. If he or she chooses, the program will also compile grades for each of the students. They will have the option to either make the laboratory grade participation based, where if the student completes the lab they will get credit, or based on the percent of questions answered correctly. Each laboratory will have a short quiz on the information taught, and the professor can decide if the student grade should be based on a single score or the best grade out of three. This will allow for a student to prepare for an upcoming real laboratory, and show their instructor that they are taking the time to learn the material better.

This program will also have useful tools for organizing a class. The professors will be able to sort all of the registered students into their correct section and instantly see how a student did on a quiz after it is taken. There will also be a section for addition information. In this area the instructors will be able to add more information for the class. They will be able to share links to other articles they'd like the students to read or provide information on due dates or class announcements or even provide study guides.

3. Glossary of Terms

Laboratory 1

Term	Comments
Anaphase	Chromosome move from the equator to the poles. Fig. 2.1
Centromeres	The region of the chromosome found in the center where sister chromosomes almost touch during mitosis. Fig. 2.1
Chromosomes	Organized structure of DNA protein found in a cell's nucleus Fig. 2.1
Cover Slip	Thin piece of plastic that overs the material that you place on a slide before viewing it with a microscope. Fig.3.2
Cytokinesis	When a single cell splits itself into two cells through cytoplasm division.
Diploid	A type of nuclei that has two sets of genes on two sets of chromosomes. 2n
Dye Solution/Stain	Biological stain that increases one's ability to see small structures within the cell. Fig. 3.2
Equator	Invisible line splitting the cell in half Fig. 2.1
Haploid	A type of nuclei that has one set of genes on one set of chromosomes. 1n
Homologous	Identical strands of chromosomes.
Hot Plate	The machine used to heat material.
Interphase	The time before mitosis. Fig. 2.1
Metaphase	Chromosomes appear aligned on the equatorial plane of the cell Fig. 2.1

Microscope	The machine used to view material on a molecular level.
Mitosis	A kind of nuclear division. A sequence of events by which the nuclear material of one cell is distributed into two equal parts. Fig. 2.1
Nucleus	The control center of the cell that contains the cell's DNA. Fig. 2.1
Nuclear membrane	Envelope around the nucleus that encloses all of the genetic material. Fig. 2.1
Nucleolus	The part of the nucleus,made of proteins and nucleic acid that does not have a membrane around it. Fig. 3.1
Onion Root Tip	Has a root cap on the very bottom, immediately over this is the meristematic reason, where cell division occurs, and the region of elongation. Fig. 3.2
Plant Cell	Usually rectangular or at least a fixed shape cell, has a cell membrane and a rigid cell wall. Fig. 3.1
Pole	Two exist on either side of the cell,divided by the equator. Fig. 2.1
Prophase	Stage of mitosis, chromosomes first appear as darkly stained areas and later as distinct entities spread throughout the nucleus. Nuclear membrane disappears and the nucleolus is gone. Fig. 2.1
Sample Cell	The prepared demonstration slide that shows exactly what a newly prepared slide should look like. Fig. 3.1
Slide	Thin piece of glass that is used to hold material that will be observed with a microscope. Fig. 3.2
Telophase	The darkly stained chromosomes are visible in both daughter nuclei. Fig. 2.1

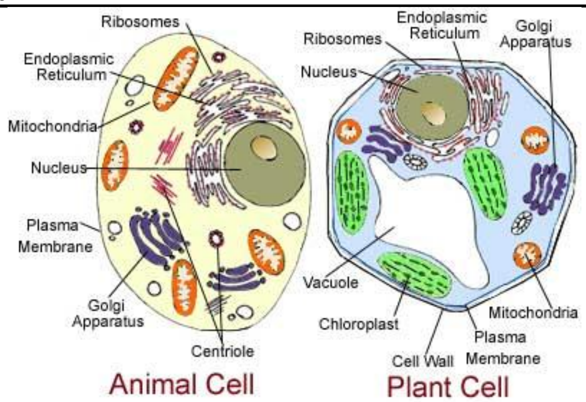


FIG.3.1 A depiction of the difference between animal and plant cells.

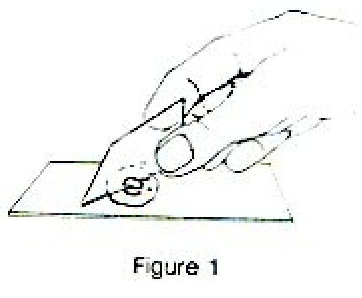


FIG. 3.2 A representation of what preparing a slide would look like.

Laboratory 2

Term	Comments
Absorbance	Or optical density, is the intensity of light at a specific wavelength that passes through a sample.
Blank	The sample used to calibrate the zero end of the absorbance scale.
Dark Control	Located on the left front side of the Spec 20, used to set the meter reading to an infinite absorbance on th left side, used for very dark sample where no light

	passes through. Fig. 2.2
Glycosuria	When urine has a high level of glucose, anything over 0.6 mg glucose/milliliter.
Hemoglobinuria	When there are high levels of free hemoglobin in the blood, usually a very low number.
Light Control	Located on the right side of the Spec 20, used to set the meter reading to zero absorbance on the right side of the meter, used fro when 100% of light is transmitted. Fig. 2.2
Light Production	Light of whichever wavelength that is set shone from the spec 20 through the sample. Fig. 2.2
pH	Measure of the acidity or basicity of a solution.
pHYDRION paper	Brand of compound pH indicators. Indicates the pH of a liquid dripped on it by changing color.
Proteinuria	An excess of serum proteins in the urine. Normal urine contains about 0.3 mg of albumin per millimeter.
Sample Holder	The slot on the top left side of the spec 20 that is where you place the test tube with the liquid that you want to find the absorbance of. Fig. 2.2
Spectronic 20	B&L Spectronic 20 Spectrophotometer, tool used to find the absorbance of a sample by shining a light through it of a specific wavelength. Fig. 2.2
Test Tube	Glass or plastic tube used to contain liquid, is open at the top and u shaped on the bottom.
Urinalysis	An array of tests preformed on urine,for medical diagnosis.
Wavelength Control	The top screen on the Spec 20 that indicates at which wavelength, in nanometers, that the light is being shone. Fig.2.2

Laboratory 3

Term	Comments
Buffer	Liquid with a pH of 8.0, added to the test tubes in the enzyme experiment.
Catalyze	Reagent that change the rate of the chemical reaction but are not used up in that reaction themselves.
Enzyme	A protein that increases the speed of rate of chemical reactions.
Inhibitor	A substance that binds to an enzyme and decreases the enzyme’s activity
Inorganic Phosphate	H3PO4, one of the resulting chemicals when pNPP reacts with water, the other is pNP. Used as an inhibitor in the enzyme experiment.
Phenyl Phosphate	Used as an inhibitor in the enzyme experiment.
Phosphate	A group of enzyme that catalyze the removal of phosphate groups from other molecules. Alkaline phosphate is used as the enzyme in this experiment.
pNP	p-Nitrophenol, one of the resulting chemicals when pNPP reacts with water. The other is inorganic phosphate. Yellow in color
pNPP	p-Nitro-Phenyl Phosphate, the substrate used in the enzyme reaction experiment. Colorless
Rate	Change in some quantity per time.

Sodium	Inhibitor added to some test tubes in the enzyme experiment
Substrate	A substance that is acted on by an enzyme

Laboratory 4

Term	Comments
Diakinesis	Part five of Prophase I, chromosomes become shorter, still repel and the nuclear envelope breaks. Fig. 2.3
Diplotene	Part four of Prophase I, chromosomes repel. Fig. 2.3
Karyotyping	Tool used for counting and viewing chromosomes. Fig. 2.4
Klinefelter Syndrome	A abnormality in the number of sex chromosomes, XXY. Fig. 2.4
Leptotene	Part one Prophase I, chromosomes start to become visible and condense. Fig. 2.3
Meiosis	Type of cell division necessary for sexual reproduction, unlike mitosis, these chromosomes recombine. Fig. 2.3
Pachytene	Part three of Prophase I, Chromosomes are fully synapsed along length, forming a tetrad. Fig. 2.3
Tetraploid	Having four copies of chromosomes Fig. 2.3
Tripple-X	A abnormality in the number of sex chromosomes, XXX. Fig. 2.4
Turner Syndrome	A abnormality in the number of sex chromosmes, XO. Fig. 2.4
XYY Karyotype	A abnormality in the number of sex chromosomes, XYY. Fig. 2.4
Zygote	A single cell that develops into a new organism Fig. 2.3
Zygotene	Part two of Prophase I, Homologous chromosomes begin to pair. Fig. 2.3

4. Functional Requirements Specification

4.1 Enumerated Functional Requirements

Number	PW	Requirement
REQ1	5	The system shall provide the student with the correct lab space when they select an experiment to perform. When they choose the first or fourth lab, the system shall generate a space for them to prepare a slide, and use the microscope. When the student selects laboratories two or three, the system shall provide a space with the appropriate test tubes, beakers and the Spec20 spectrophotometer.
REQ2	5	The system shall allow for a student to prepare a slide by cutting the tip off of an onion root, adding drops of dye solution and then covering this with the cover slip. The system shall then allow the student to view this under a microscope. All of these steps will be animations prompted by the student.
REQ3	5	The system shall allow the student to drag and place components in the correct space. This includes dragging the parts of the cell into the correct space within the cell membrane or dragging the beaker to the test tube that the student wants to fill with that liquid.
REQ4	4	The system shall simulate the calibration and use of the Spectronic 20 spectrophotometer. This includes inserting a blank or sample test tube, adjusting setting and then reading the wavelength shone through the sample test tube.
REQ5	4	The system shall provide a hypothetical chromosome karyotype and allow for the students to review this karyotpye and click on the irregular chromosomes, this will draw a circle around the abnormalities and show a description of what genetic disorder this abnormality would cause.
REQ6	4	The system shall have correct and incorrect ways to complete the experiments. The student will have to complete all of the steps correctly, with some little room for error, or else there will be a message saying that the step they just took was incorrect.
REQ7	3	The system shall allow the students to register, log in, view their grade, change account settings, and complete five virtual biology laboratory experiments.
REQ8	3	The system shall allow the professors to register, log in, create and edit sections, add information to be shared over the addition information section and view the grades of all students associated with their sections.
REQ9	2	The system shall instantly record the grade the student received, based on the requested way of scoring the professor has chosen, and updating both the students grade book and the statistics the professor sees on the class.
REQ10	1	The system shall give a quiz after the student’s completion of each laboratory, asking the student to correctly answer 5 to 10 questions.

4.2 Stakeholders

There are two main stakeholders in the Virtual Biology Laboratory. The first is the professor for the course and second are the students currently enrolled in that course. The professor's main purpose for this program is to provide more learning opportunities for his or who students and receive real time reports on student lab completions and quiz grades. The professor also should be able to add/remove students from his or her section. The student's purpose for this program is to reinforce what they have learned in lecture and laboratory, and to be able to take lab quizzes to be graded on their performance.

The needs of the stakeholders is one of the most important aspects of software engineering. The ability to communicate technical aspects of the project to stakeholders with different academic backgrounds is also key. Some requirements and needs of stakeholders must also be inferred, especially when a particular group of stakeholders is made up of a large, diverse population (in our case, the "Student". In reality, every student using the program would have their own wants and needs but it is highly unlikely all of these can be met, so we must generalize). The tables below show each stakeholder and their respective requirements/important use cases.

Stakeholder 1: Professor
Requirements (Summary): Ability to use the Virtual Biology Laboratory as a class supplement; to create a class section and add/remove students from it; receive updates of student progress and quiz grades; ability to add new lab modules.
Requirements (Specific): REQ8, REQ9
Important Use Cases: UC10 (ViewStudentResult), UC11 (StudentEditor), UC12 (CassEditor), UC13 (ViewClassResult)

Stakeholder 2: Student
Requirements (Summary): Ability to use the Virtual Biology Laboratory as a learning supplement; to sign-up to a specific section; perform labs and lab quizzes upon completion; check personal grade summary.
Requirements (Specific): REQ1, REQ2, REQ3, REQ4, REQ5, REQ6, REQ7, REQ10
Important Use Cases: UC1 (SelectLab), UC2 (PlaceItem), UC3 (Cut), UC4 (UseHotplate), UC5 (UseMicroscope), UC6 (PourItem), UC7 (UseSpec20), UC8 (Label), UC9 (TakeLabQuiz), UC10 (ViewStudentResult), UC11(StudentEditor)

Details regarding the specific use cases can be found in section 4.4.

4.3 Actors and Goals

There are two primary actors (human) and one secondary actor (virtual) in the Virtual Biology Lab. The two primary actors are the student and the professor.

The goal of the student is to learn about various biology topics through interactive lab demos (modeled in Flash) and use this knowledge to complete graded quizzes based on these demos. The student can also check his grades immediately after taking a quiz using a Gradebook applet.The professor can use the system to monitor student activity and view/adjust their grades.

The secondary actor in the lab is the MySQL database that holds all the information including student data and grades. It's primary goal is to interface with the main website and provide information to the primary application and its users when needed. It should also be able to save any user information that is edited by a human actor.

Below is a table showing the goals and use cases corresponding to each actor. The student and professor actors have largely similar goals/requirements as compared to the stakeholder tables in the last section, but here we will specify each goal and which use case(s) fulfill that goal.

Actor (Type)	**Goal**	**Use Cases**
Student (Human)	To select a lab to perform.	UC1 (SelectLab)
Student (Human)	To interact with a lab by moving and manipulating virtual objects.	UC2 (PlaceItem), UC3 (Cut), UC6 (PourItem), UC8 (Label)
Student (Human)	To perform measurements on virtual objects in the labs.	UC4 (UseHotplate), UC5 (UseMicroscope), UC7 (UseSpec20)
Student (Human)	To take a lab assessment quiz and receive a grade.	UC9 (TakeLabQuiz)
Student (Human)	To view completed assessment grades.	UC10 (ViewStudentResult)
Student (Human)	To sign up for a class section by inputting student information.	UC11 (StudentEditor)
Professor (Human)	To create and manage a class section.	UC12 (ClassEditor)
Professor (Human)	To edit individual student details.	UC11 (StudentEditor)
Professor (Human)	To view an individual student's grades/lab progress.	UC10 (ViewStudentResult)

Professor (Human)	To view an entire section’s grades/lab progress.	UC13 (ViewClassResult)
Database (Virtual)	To allow student grades saved in the MySQL database to be viewable by a human actor.	UC10 (ViewStudentResult), UC13 (ViewClassResult)
Database (Virtual)	To save changes made by a human actor to the database.	UC11 (StudentEditor), UC12 (ClassEditor)

4.4 Use Cases

4.4.1 Casual Description

Below is a chart that provides a short description for each use case. Each one will be explained in more detail in the full-dressed description section.

UC1	Select Lab: Allows the user to tell the system which laboratory to begin
UC2	Place Item: Moves a component into another space (examples: placing an item onto a slide, putting a slide in the microscope, placing a test tube in the sample holder of the spec 20)
UC3	Cut: Cut a part off of an item
UC4	Use Hotplate: Set temperature and time, heat up an item
UC5	Use Microscope: Load a slide, and zoom in or out and view the image
UC6	Pour Item: Pour the contents from beaker or test tube into another test tube, slide or pH paper
UC7	Use Spec20: Adjust the light, dark or wavelength control and read absorbance
UC8	Label: Add a label to the selected item
UC9	Take Lab Quiz: generates a quiz of random questions for a student after they finish a lab
UC10	View Student Result: Lets a student or professor view the students progress
UC11	Student Editor: Allows student to register their information in their class
UC12	Class Editor: Allows professor to register all the students in a class with the virtual biology laboratory
UC13	View Class Result: Lets a professor view the entire class' results

UC1 select lab, UC10, UC11, UC12 and UC13 are all optional for the student or professor to select. UC2 through UC7 are all sub use cases. These use cases must be completed within a specific laboratory so they are included by the Select lab use case, UC1. The UC8 is an optional use case that can performed within one of the laboratories, this is why it extends UC1, select laboratory.

UC1 through UC9 can only be completed by a user with a student account. Similarly UC12 and UC13 can only be completed by a user with a professor account. UC10 and UC11 can be completed by either a student or a professor. The data base interacts with the system for UC10 through UC13. In the next section, the use case diagram illustrates all of these relations between users and use cases, and in the section after that there are more detailed descriptions of all of the use cases.

4.4.2 Use Case Diagram

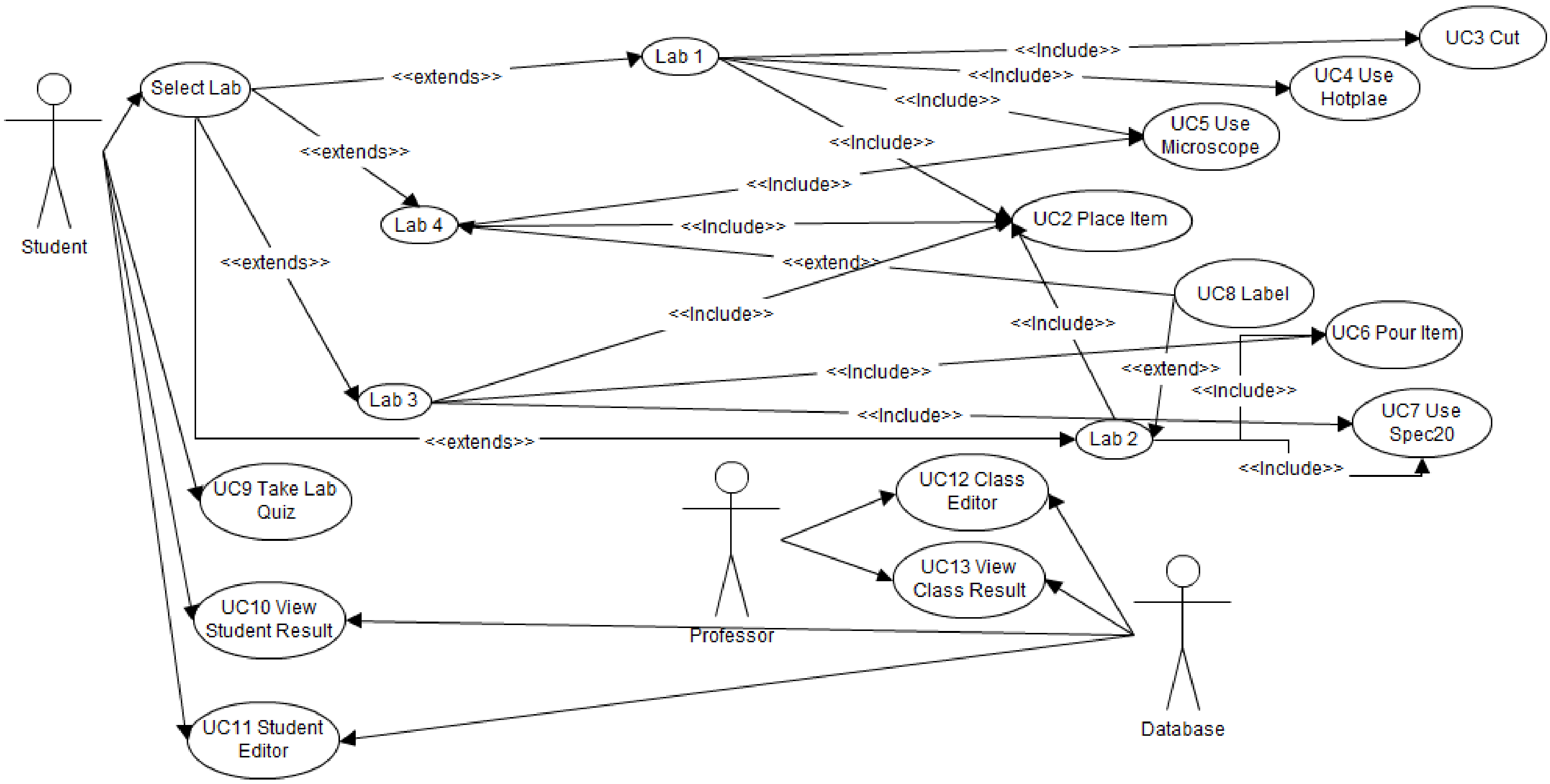


Fig. 4.1 Use Case Diagram

4.4.3 Fully-Dressed Description

UC1 Select Lab

Related Requirements: REQ1 REQ6
Initiating Actor: Student
Actor's Goal: The actor will select the laboratory that they wish to work on
Preconditions: User is at the home lab screen (not working on a lab), and has available labs to choose from.
Postconditions: The laboratory has been selected and the system has generated the laboratory space.

Flow of Events for Main Success Scenario:
-> 1. Student requests to perform the next laboratories that they have to complete.
<- 2. System generates all of the supplies that the student would need for that laboratory.
Flow of Events for Extension (Alternate Scenario):
-> 1. Student requests to perform one the laboratories out of order
<- 2. System checks that student's progress, find that they did not complete the laboratory before
3. System redirects user back to main page

UC2 Place Item

Related Requirements: REQ2, REQ3, REQ4
Initiating Actor: Student
Actor’s Goal: To move one item from one space to another. Some times this is to add an item on top of another (slide cover over dye, dye over root, root over slide) or to fill a holder with that item (slide in microscope’s slide holder, test tube in test tube holder or test tube in spec20’s sample holder).
Preconditions: The laboratory space is generated and the item to be selected is movable
Postconditions: The selected item exists in a different space and has acquired new characteristics

Flow of Events for Main Success Scenario:
-> 1. Student selects an object
-> 2. Student drags that object to another space.
<- 3. The system recreates that object in the new space.
<- 4. If this space requires the object to change properties, the system will change them.
Flow of Events for Extension (Alternate Scenario):
-> 1. Student selects an object
-> 2. Student drags that object to another space.
<- 3. The object cannot be placed in that new space, and it is left in its original space

UC3 Cut

Related Requirements: REQ2, REQ6
Initiating Actor: Student
Actor’s Goal: To cut a part off of one object, get rid of the unusable part and leave only the smaller part.
Preconditions: The laboratory space is generated and the item to be selected is able to be cut
Postconditions: The selected item has been cut where the user selected, the larger portion is discarded and the smaller portion remains in its place.

Flow of Events for Main Success Scenario:
-> 1. The user selects the knife.
<- 2. The system changes the users cursor into an image of a knife.
-> 3. The user drags his/her cursor over the line on the object the user wishes to cut.
<- 4. The system splits the object in half and gets rid of the larger portion and returns the user’s cursor to normal

Flow of Events for Extension (Alternate Scenario):
-> 1. The user selects the knife
<- 2. The system changes the users cursor into an image of a knife
-> 3. The user drags his/her cursor over the line on the object the user wishes to cut.
<- 4. The user has tried to cut something that cannot be cut, the system leaves the object the same and returns the user’s cursor to normal

UC4 Use Hotplate

Related Requirements: REQ1, REQ2, REQ3
Initiating Actor: Student
Actor's Goal: To take an item and increase it's temperature to a set number.
Preconditions: The Hotplate is on and an item has been placed on it
Postconditions: The item's temperature has increase.
Flow of Events for Main Success Scenario:
-> 1. The user selects the optimal temperature
<- 2. The hot plate increases the object temperature and changes its appearance if necessary
Flow of Events for Extension (Alternate Scenario):
-> 1. The user selects a less optimal temperature
<- 2. The hot plate increase the objects temperature and changes its appearance to burnt.
<- 3. The system regenerates the object in its original space with its original tempurature and the system notifies the user to try a lower tempurature.

UC5 Use Microscope

Related Requirements: REQ1, REQ2, REQ3
Initiating Actor: Student
Actor's Goal: To zoom in on a slide and view it at that magnification
Preconditions: A slide is held in the slide holder of the microscope
Postconditions: The user only sees the zoomed in image of that slide

Flow of Events for Main Success Scenario:
-> 1. The user selects the magnification at which they would see the contents of the slide best
<- 2. The system changes the view from that of the lab space to that of the magnified slide
Flow of Events for Extension (Alternate Scenario):
-> 1. The user selects the a less optimal magnification
<- 2. The system changes the view from that of the lab space to that of an unclear image of the magnified slide
<- 3. The view returns to the lab space and the system notifies the user to try a larger or smaller magnification.

UC6 Pour Item
Related Requirements: REQ4, REQ6 **Initiating Actor**: Student **Actor's Goal:** To add the contents from one vessel to a second. (Beaker to test tube, test tube to test tube, and test tube to pH paper) **Preconditions**: The laboratory space has at least one thing to pour from and one to pour into. **Postconditions**: The correct contents are held in the pored and poured into vessels. **Flow of Events for Main Success Scenario:** -> 1. The user selects a pourable object -> 2. The user drags this objects to the object it wants to pour into <- 3. The system draws more liquid in the new object and less liquid in the original <- 4. The system draws the pourable object again in its original space **Flow of Events for Extension (Alternate Scenario):** -> 1. The user selects a pourable object -> 2. The user drags this objects to the object it wants to pour into <- 3. The system recognizes that the new object cannot have anything more poured into it and notifies the user of this <- 4. The system draws the pourable object again in its original space

UC7 Use Spec20
Related Requirements: REQ1, REQ3, REQ4 **Initiating Actor**: Student **Actor's Goal:** To adjust the controls and read the absorbance of whatever is in the sample holder of the Spec20 **Preconditions**: The Spec20 is on, calibrated and there are knobs available for the student to adjust **Postconditions**: The controls have been adjusted and the screen shows the absorbance of whatever is in the sample holder. **Flow of Events for Main Success Scenario:** -> 1. The user enters the wavelength they wish to read at <- 2. The system displays that as the new wavelength <- 3. The system gets the information about the absorbance from the test tube <- 4. The system displays the absorbance on the spec20 **Flow of Events for Extension (Alternate Scenario):** -> 1. The user enters the wavelength they wish to read at <- 2. The system displays that as the new wavelength <- 3. The system fails to get the information about the absorbance from the test tube <- 4. The system displays an error on the spec20

UC8 Label

Related Requirements: REQ5
Initiating Actor: Student
Actor's Goal: To select a portion of an image and add a label to it, like numbering a chromosome on a karyotype.
Preconditions: There is an image where a student can click that will generate a text field where that student can label it
Postconditions: That text field contains what the student wrote.

Flow of Events for Main Success Scenario:
-> 1. The user clicks on an image
<- 2. That image can be labeled, The system generates a text box
-> 3. The user enters the label
<- 4. The system checks that this label is correct and no longer lets the user edit it
Flow of Events for Extension (Alternate Scenario):
-> 1. The user clicks on the image
<- 2. The system realizes that that image cannot be labeled and notifies the user

UC9 Take Lab Quiz

Related Requirements: REQ10
Initiating Actor: Student
Actor's Goal: The actor will receive a randomized quiz on the material presented in a sub-module for grading.
Preconditions: The student has completed the associated lab but not yet completed the quiz
Postconditions: The system database contains the grade this student received for this quiz

Flow of Events for Main Success Scenario:
-> 1. Student performs laboratory and has completed it
<- 2. System redirects student to webpage with randomized quiz questions pertaining to the particular experiment they have just completed
-> 3. Student fills out quiz and submits for grading
<- 4. System records grade in gradebook database
Flow of Events for Extension (Alternate Scenario):
-> 1. Student performs the laboratory and has completed it
<- 2. System redirects student to webpage with randomized quiz questions pertaining to the particular experiment they have just completed
-> 3. Student fills out quiz and decides to leave page before submitting or submits without all answers filled in
<- 4. System prompts user that any unfinished questions will be graded, asks if they want to submit/leave page or resume answering quiz
-> 5. Student makes choice

UC10 View Student Result

Related Requirements: REQ7, REQ8, REQ9

Initiating Actor: Instructor and Student
Actor's Goal: View the grade on every lab that a particular student has completed
Preconditions: The class and individual student has registered. The student has completed the lab. The instructor has selected to assign grades.
Postconditions: The student's individual grades appear for the professor to view and make adjustments if necessary.

Flow of Events for Main Success Scenario:
-> 1. The student or instructor requests to see the students grades
<- 2. The system shows the gradebook for that particular student
Flow of Events for Extension (Alternate Scenario):
-> 1. The student or instructor requests to see the students grades
<- 2. The system shows the gradebook for that particular student, but no grades are in the database. System notifies user that there are no grades in the database.

UC11 Student Editor

Related Requirements: REQ7 REQ8
Initiating Actor: Student
Actor's Goal: Register their name and ruid in the correct section number. Used to add and remove specific students.
Preconditions: The instructor has provided the information on who should be in each section
Postconditions: The system database contains the information about this student and shows that they have successfully registered.The student can view their grades and complete labs/quizzes

Flow of Events for Main Success Scenario:
-> 1. The student inputs their netID, username and password
2. The system checks that this netID corresponds with a section that the instructor has registered
<- 3. The system notifies the student that he/she registered successfully and that they can begin completing labs
<- 4. The system makes a change to the professors gradebook indicating that the student has successfully registered
Flow of Events for Extension (Alternate Scenario):
-> 1. The student inputs their netID, username and password
2. The system checks that this netID corresponds with a section that the instructor has registered.
<- 3. The system notifies the student that he/she is not enrolled in any section and informs the student that registration has failed.

UC12 Class Editor

Related Requirements: REQ7 REQ8
Initiating Actor: Instructor
Actor's Goal: Associate the instructor and specified students to a new section
Preconditions: Professor is on appropriate page to create new section, has list of all available kids to select from.
Postconditions: The database contains information about the new section

Flow of Events for Main Success Scenario:
-> 1. Instructor inputs the section number and student NetID's to associate with it
<- 2. The system database is updated to contain all of this information
<- 3 The system notifies the students that they can now register

Flow of Events for Extension (Alternate Scenario):
-> 1. Instructor inputs the section number and student NetID's to associate with it
<- 2. System detects that the instructor has left out either the section number or the student's NetID and prompts instructor to re-enter data for that student.

UC13 View Class Result

Related Requirements: REQ7 REQ8 REQ9
Initiating Actor: Instructor
Actor's Goal: View the grades of every student in a particular section
Preconditions: The class and every student in it had registered. Students have completed the labs. The instructor has chosen to assign grades.
Postconditions: All available student lab information is displayed for professor. Professor can view the given quiz/lab results and make adjustments if necessary.

Flow of Events for Main Success Scenario:
-> 1. The instructor requests to see the class grades
<- 2. The system shows the gradebook and statistics on each section
Flow of Events for Extension (Alternate Scenario):
-> 1. The instructor requests to see the class grades
<- 2. The system shows the gradebook, but no grades are in the database. System notifies user that there are no grades in the database.

4.4.4 Use Case Traceability Matrix

Req't	PW	UC1	UC2	UC3	UC4	UC5	UC6	UC7	UC8	UC9	UC10	UC11	UC12	UC13
REQ1	5	x			x	x		x						
REQ2	5		x	x	x	x								
REQ3	5		x		x	x		x						
REQ4	4		x				x	x						
REQ5	4								x					
REQ6	4	x		x			x							
REQ7	3										x	x	x	x
REQ8	3										x	x	x	x
REQ9	2										x			x
REQ10	1									x				
Max pw	5													
Total pw	36	9	14	9	15	15	8	14	4	1	8	6	6	8

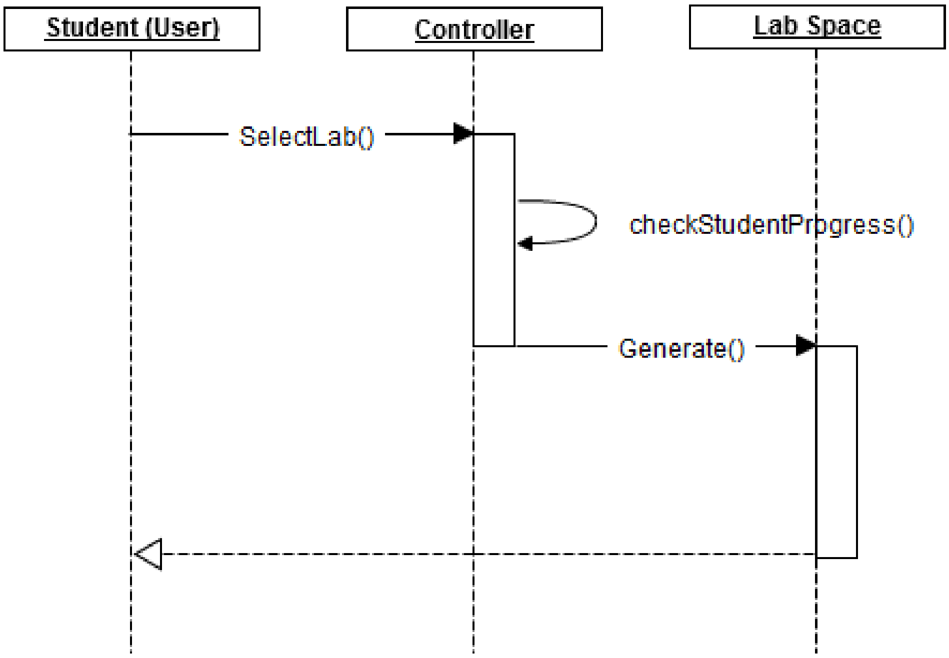
The matrix above shows the relationship between the function requirements and the use cases. This matrix also uses the priority weight of the functional requirements to show the weight of all of the use cases. By looking at the bottom row, one could see that UC4 and UC5 are the most important use case and that UC9 is the least important use case in relation to the user statement of requirements.

4.5 System Sequence Diagrams

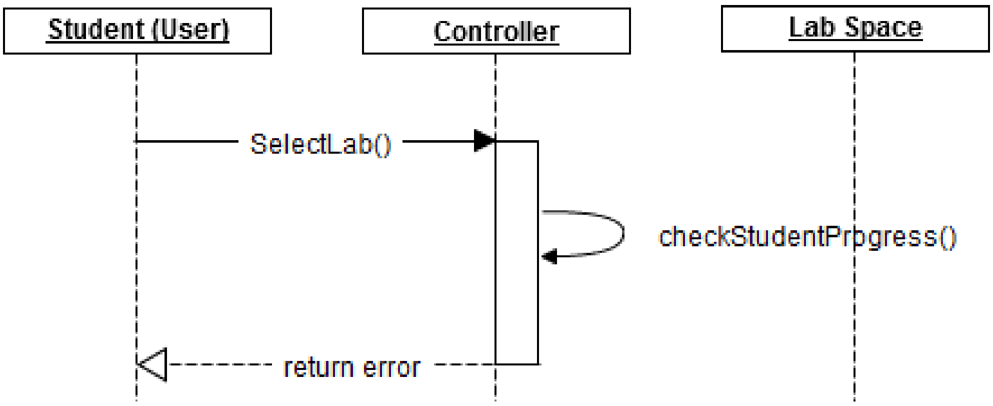
For many of the Sequence Diagrams you will see a method called "resetLabSpace()". This method is comparable and used in place of the Undo/Redo design principle. Since each of these actions are know to be reversible, we can simply make the call to the function to restore the workspace to the appropriate stage of the program.

UC1 (Select Lab) Sequence Diagrams:

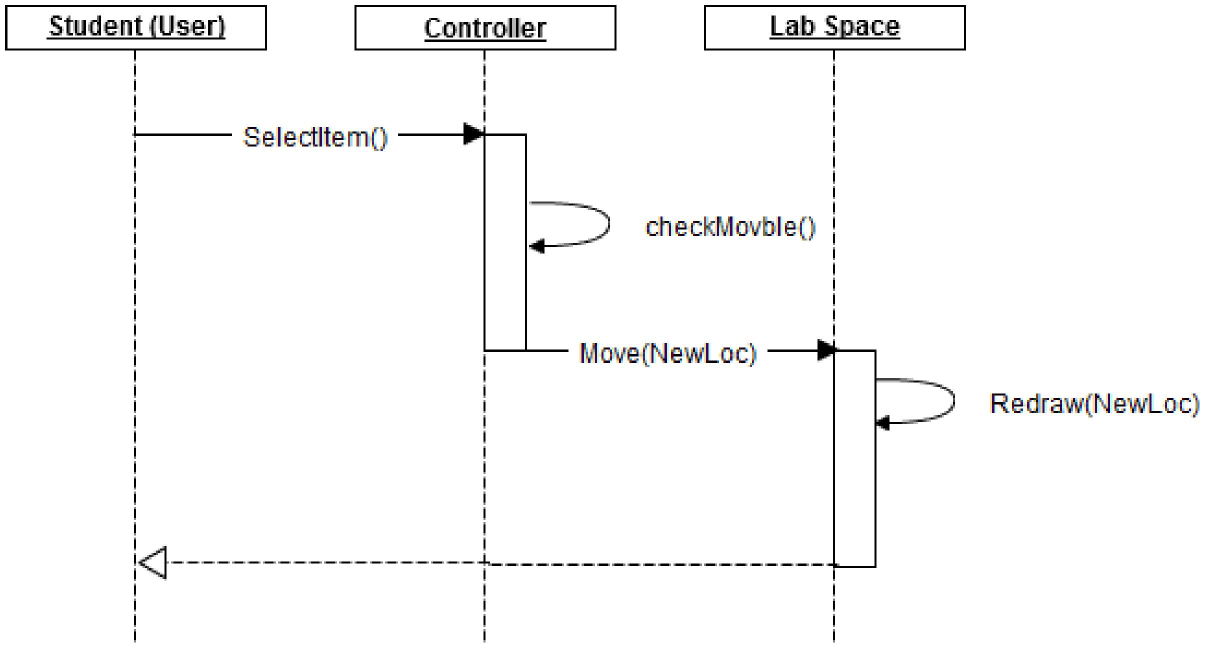
Success Case:



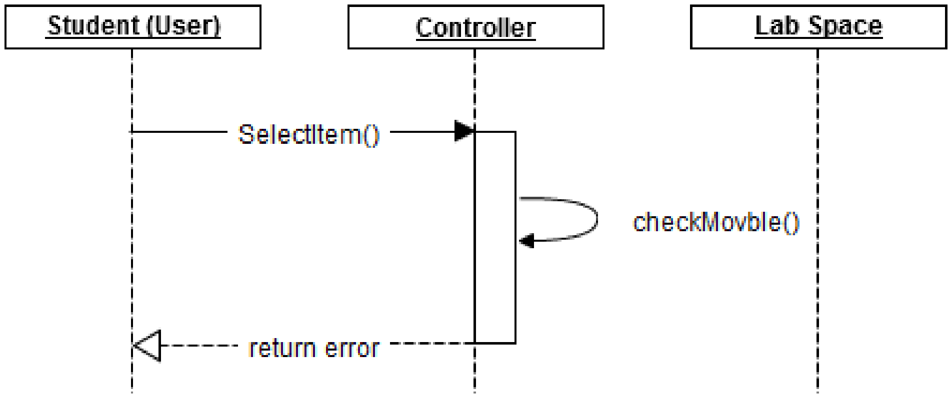
Alternate Case: (For when student is trying to complete a laboratory when they had not completed all of the earlier laboratories)



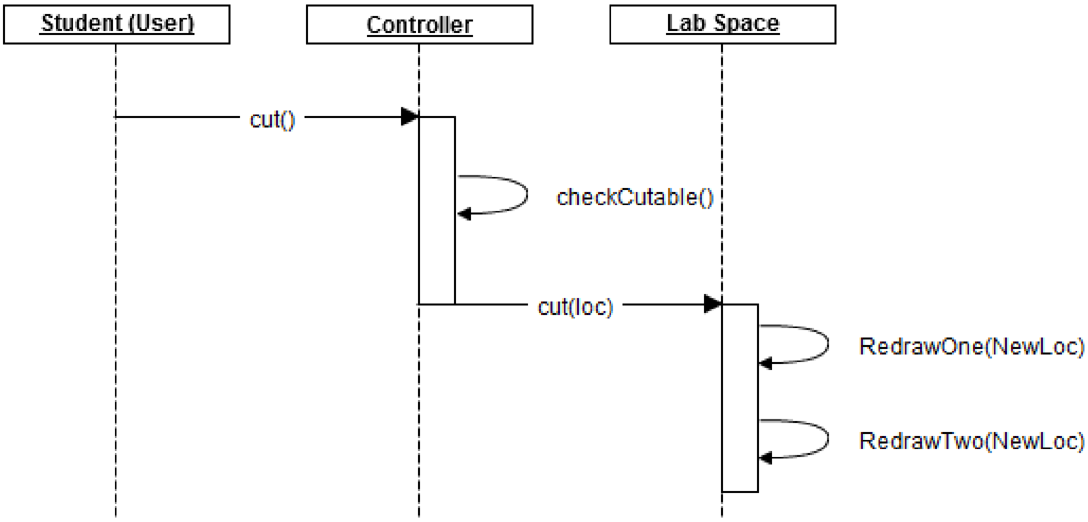
UC2 (Place Item) Sequence Diagrams:
Success Case:



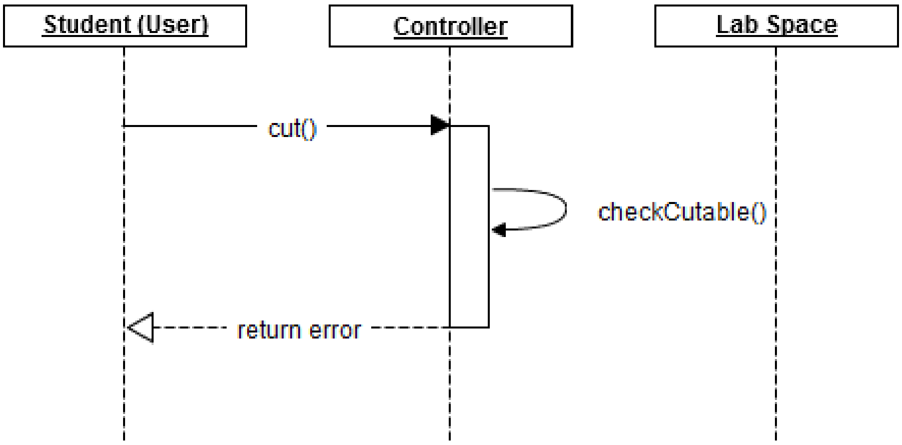
Alternate Case (For when the object the student selects is not movable):



UC3 (Cut) Sequence Diagrams:
Success Case:

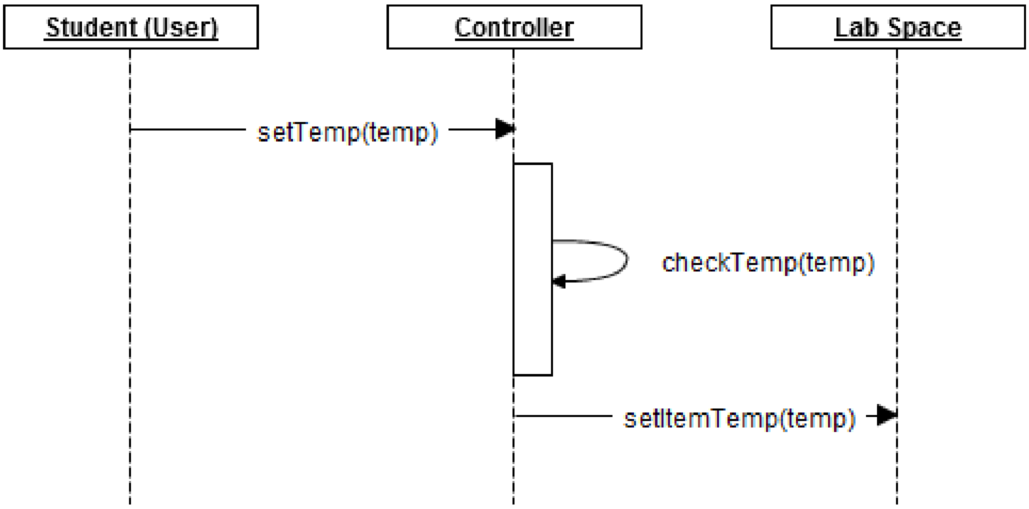


Alternate Case (For when object is not cuttable):

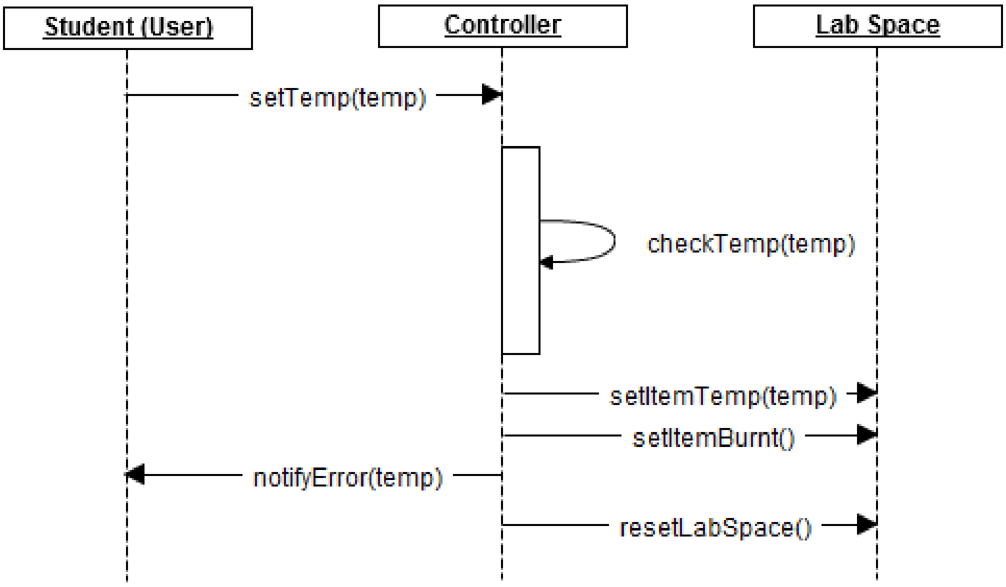


UC4 (Use Hotplate) Sequence Diagrams:

Success Case:

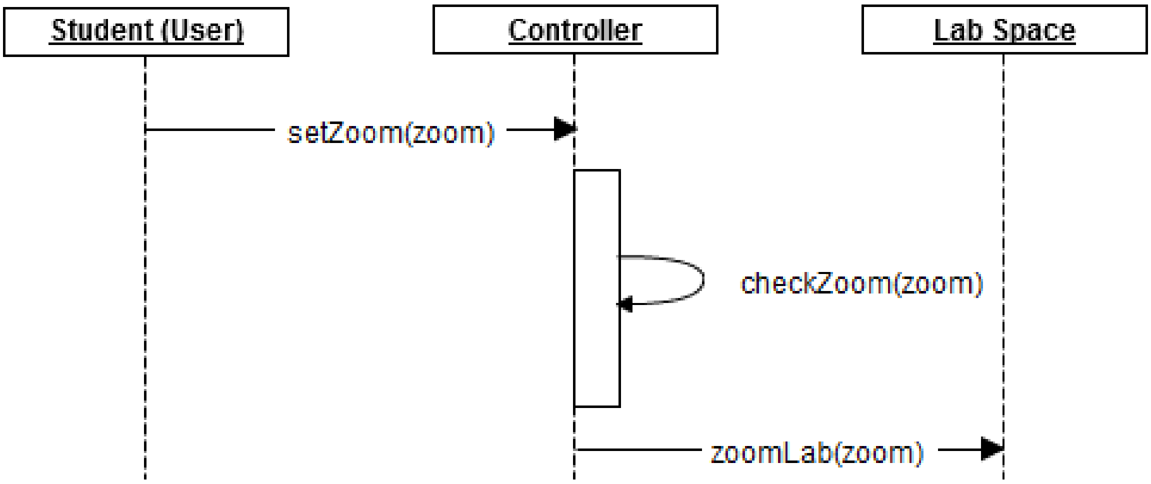


Alternate Case:

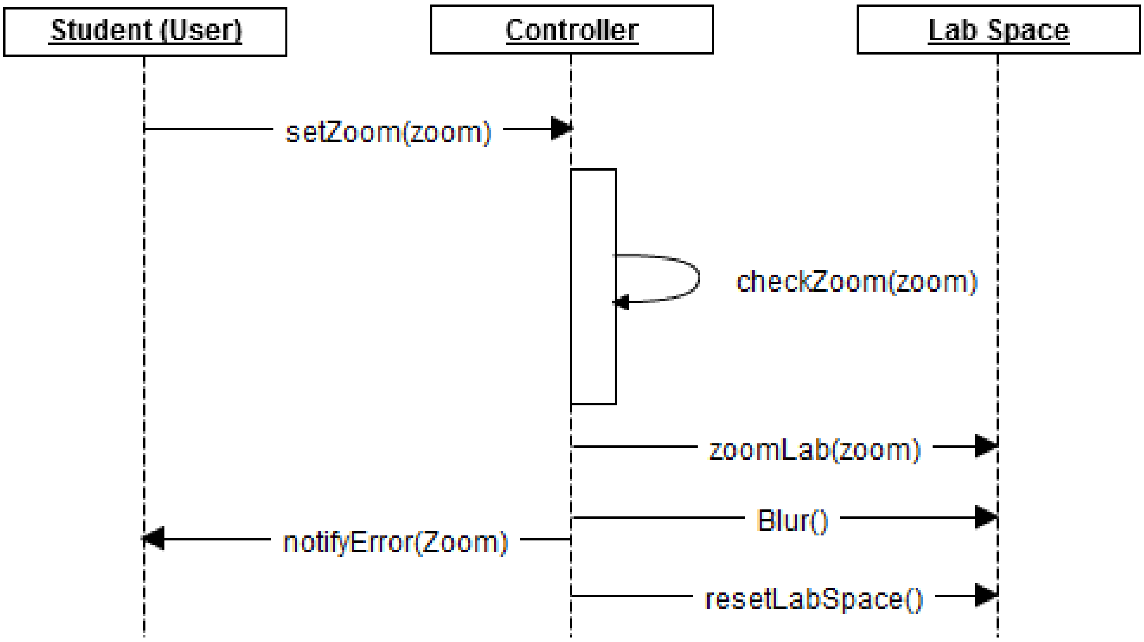


UC5 (Use Microscope) Sequence Diagrams:

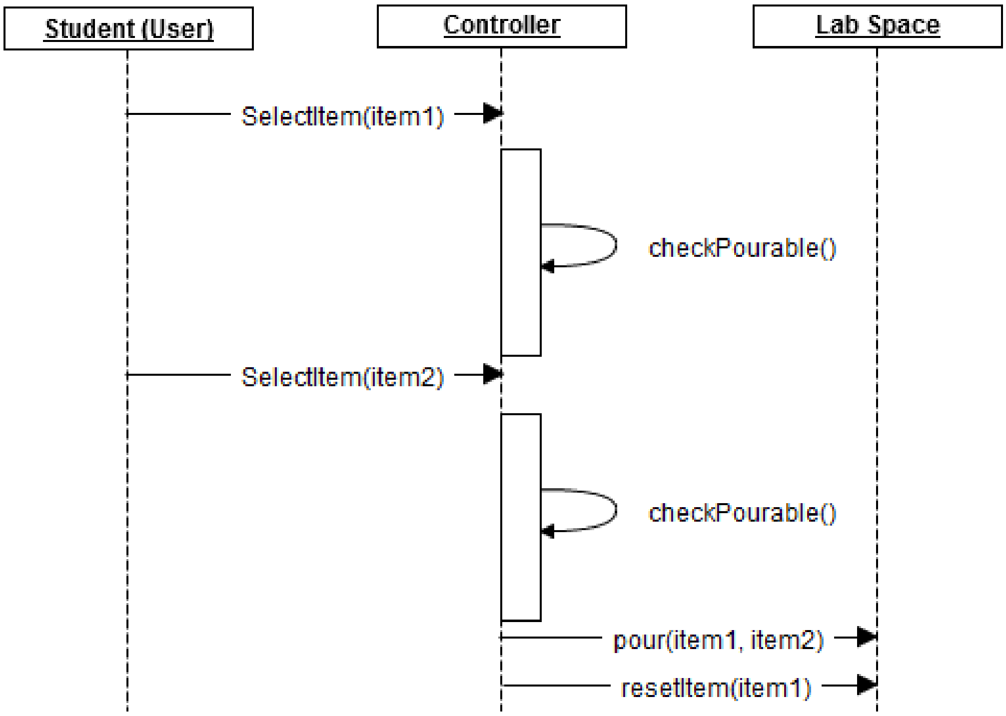
Success Case:



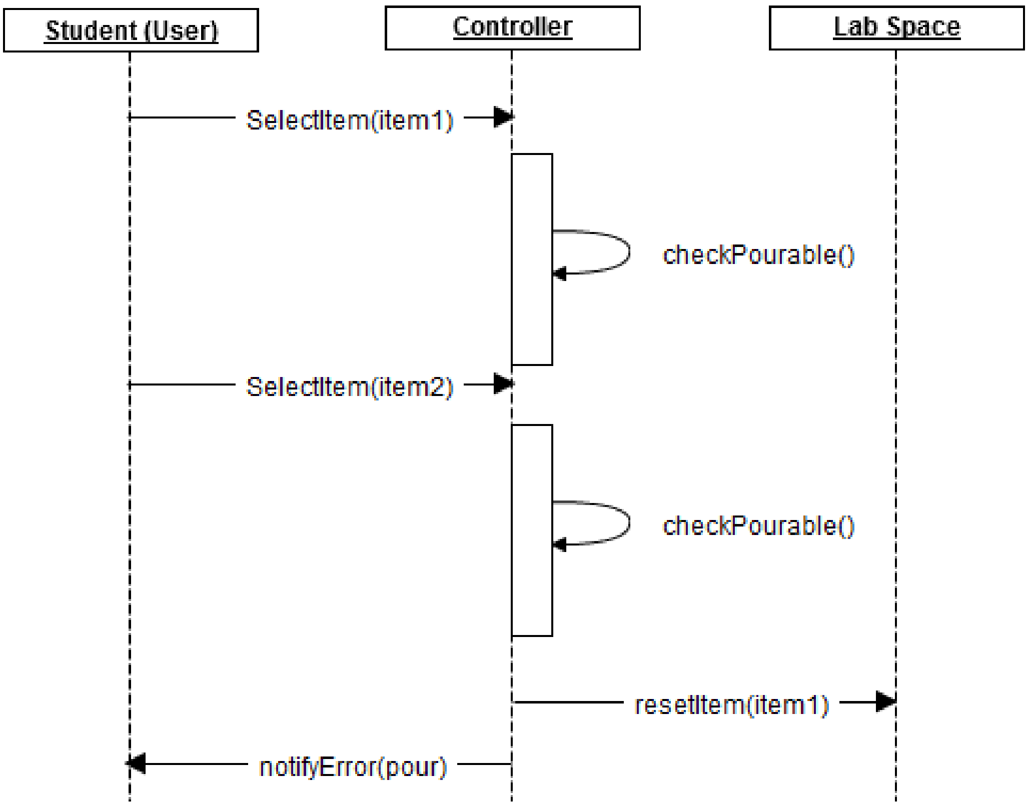
Alternate Case:



UC6 (Pour Item) Sequence Diagrams:
Success Case:

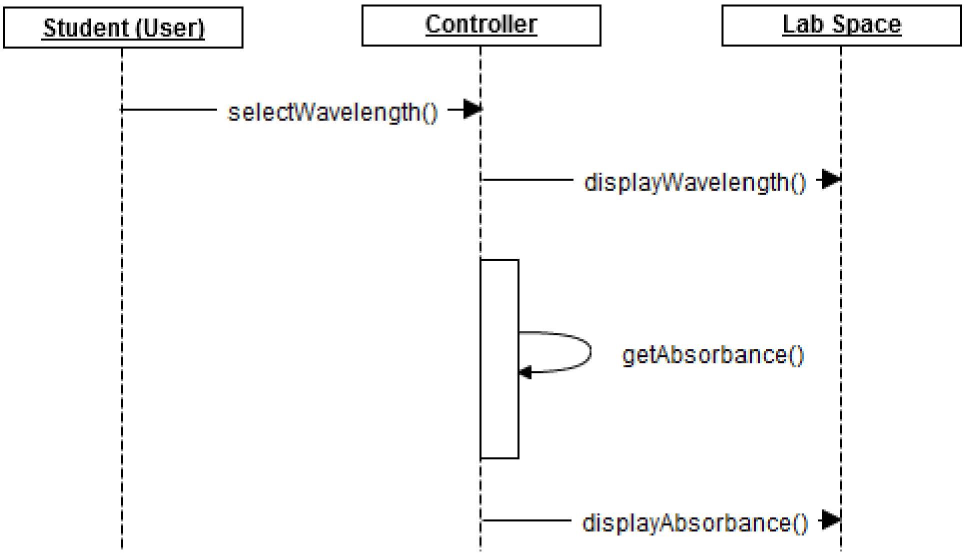


Alternate Case:

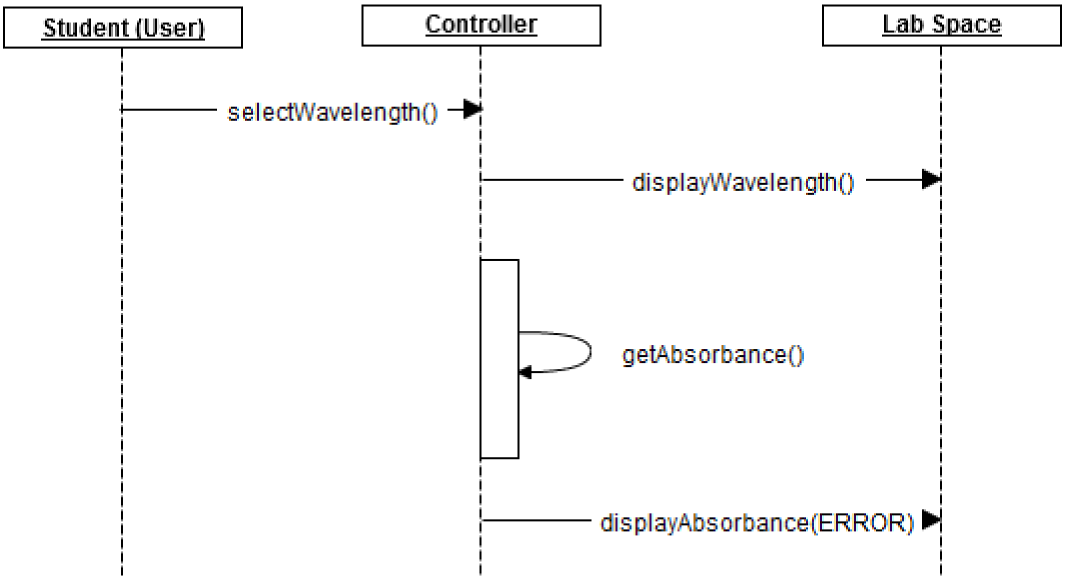


UC7 (Use Spec20) Sequence Diagrams:

Success Case:

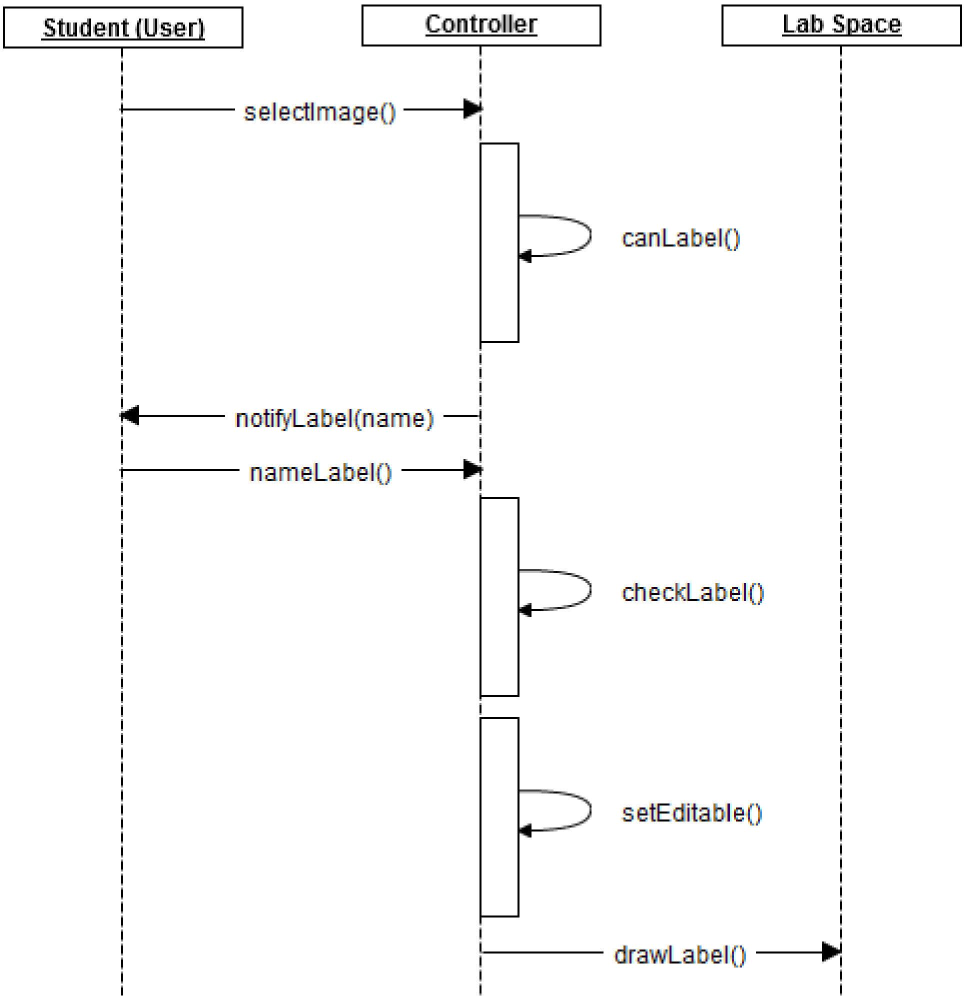


Alternate Case:

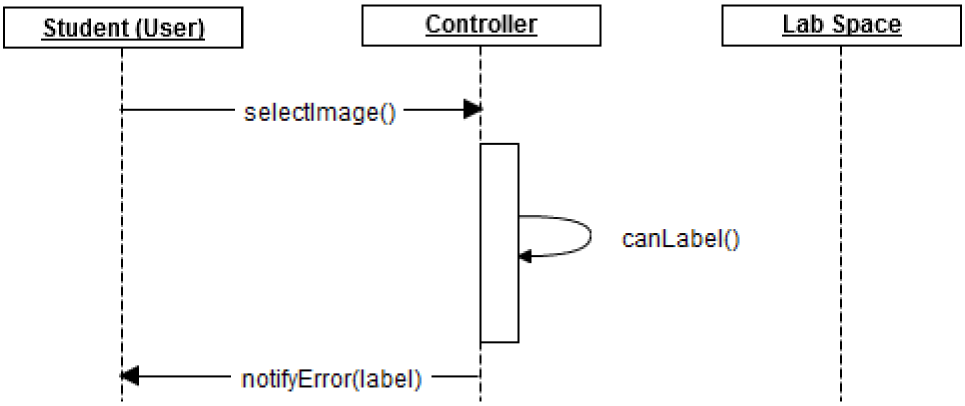


UC8 (Label) Sequence Diagrams:

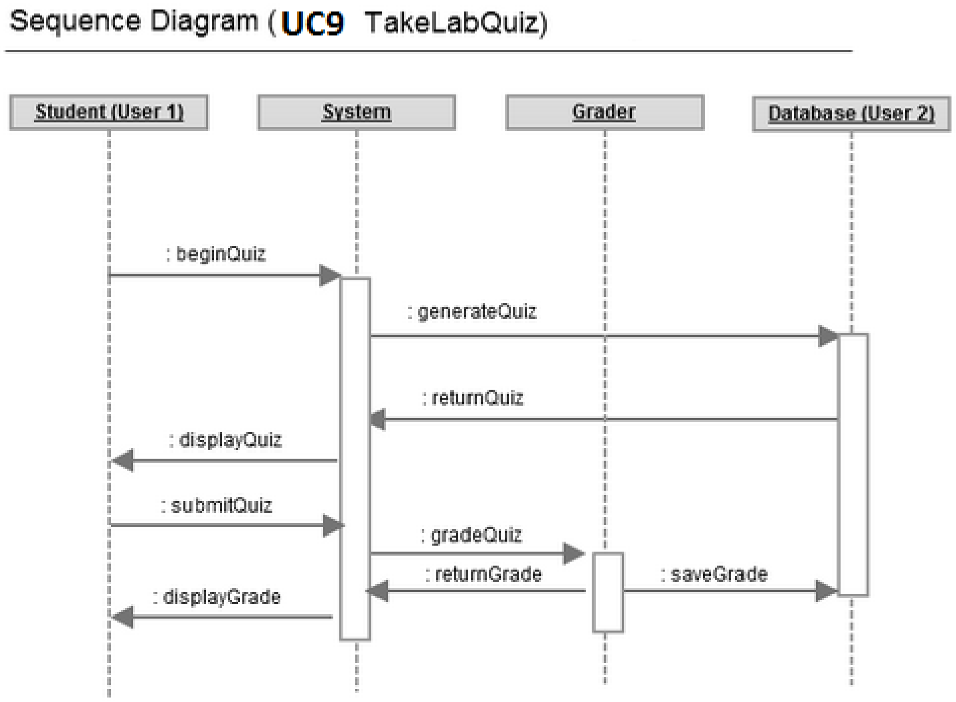
Success Case:



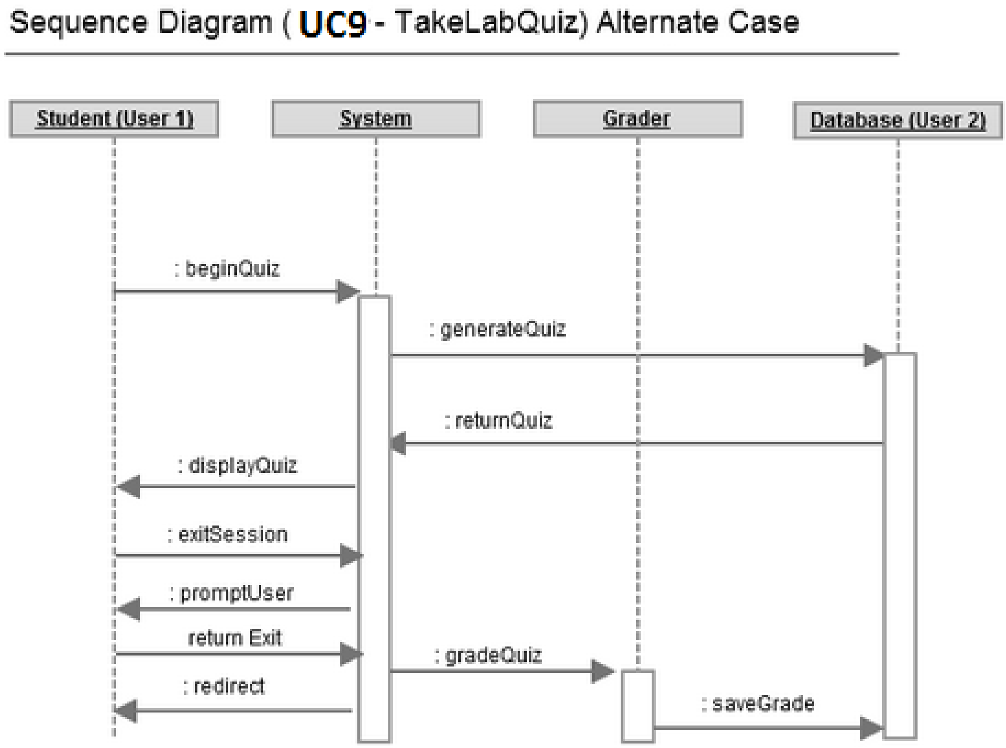
Alternate Case:



UC9 (Take Lab Quiz) Sequence Diagrams:
Success Case:

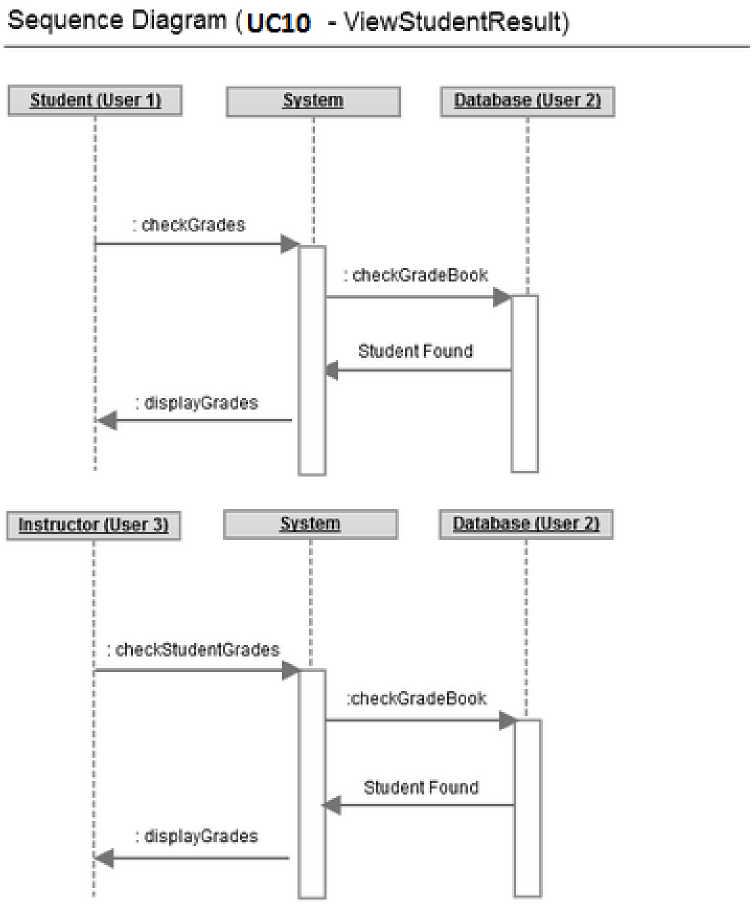


Alternate (Failure) Case:

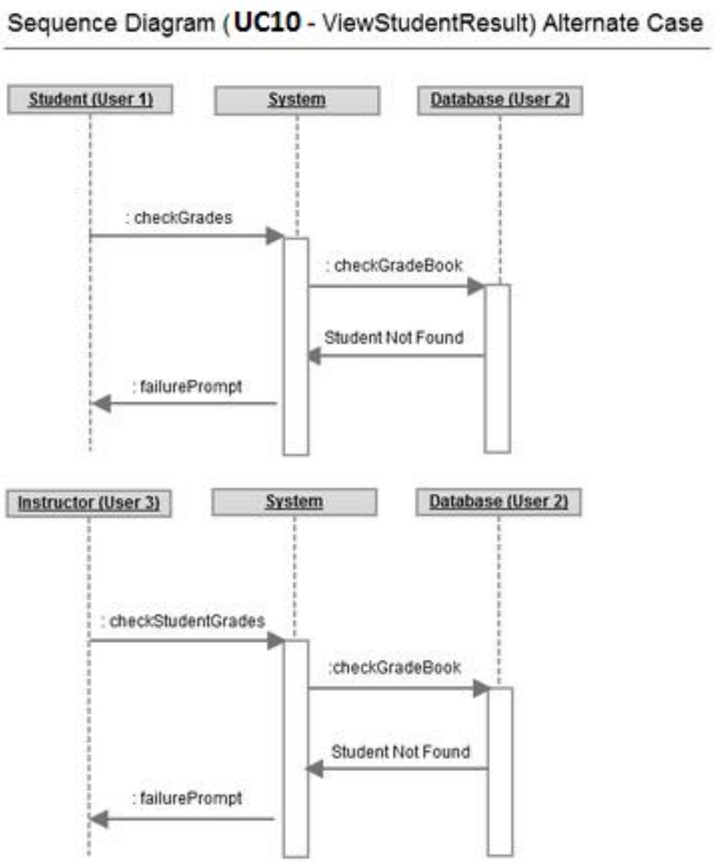


UC 10 (View Student Result) Sequence Diagrams:

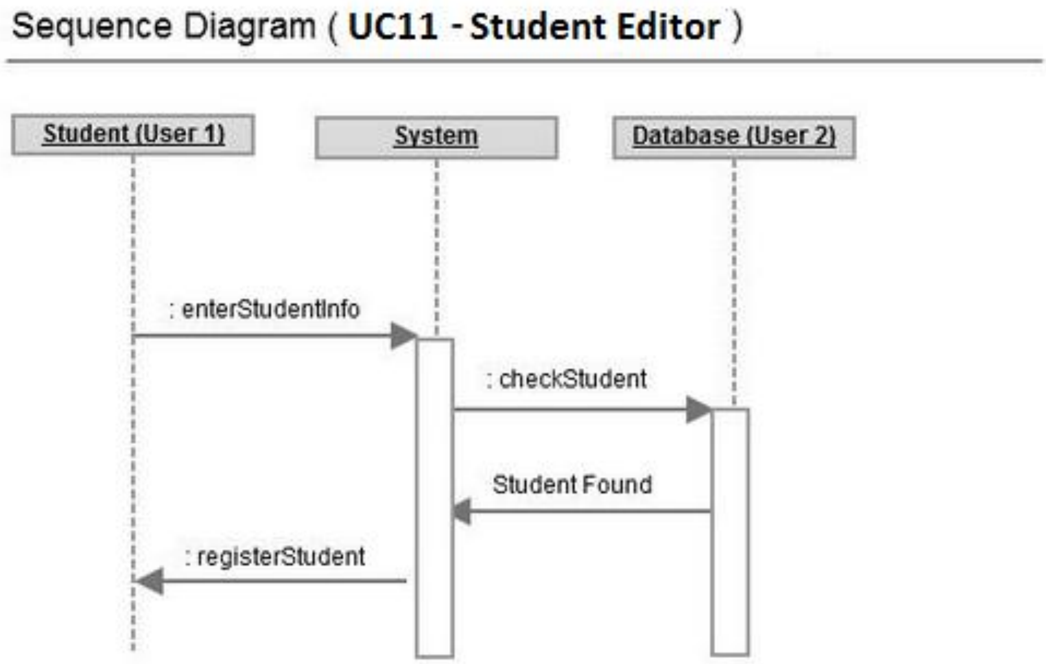
Success Case:



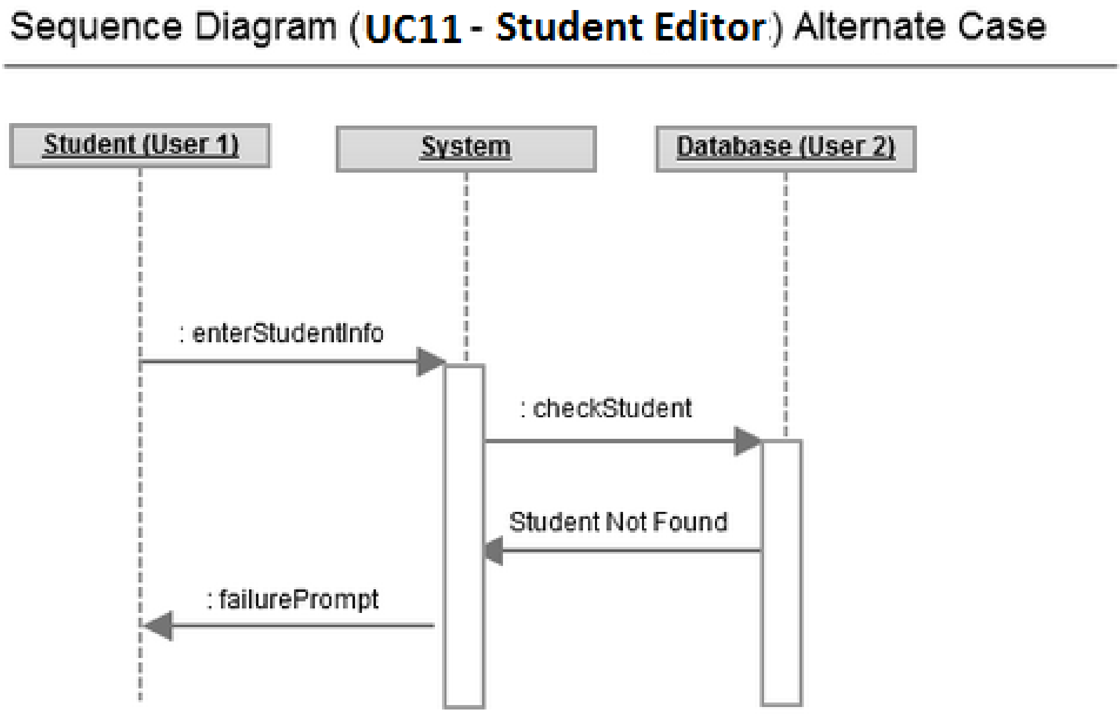
Alternate (Failure) Case:



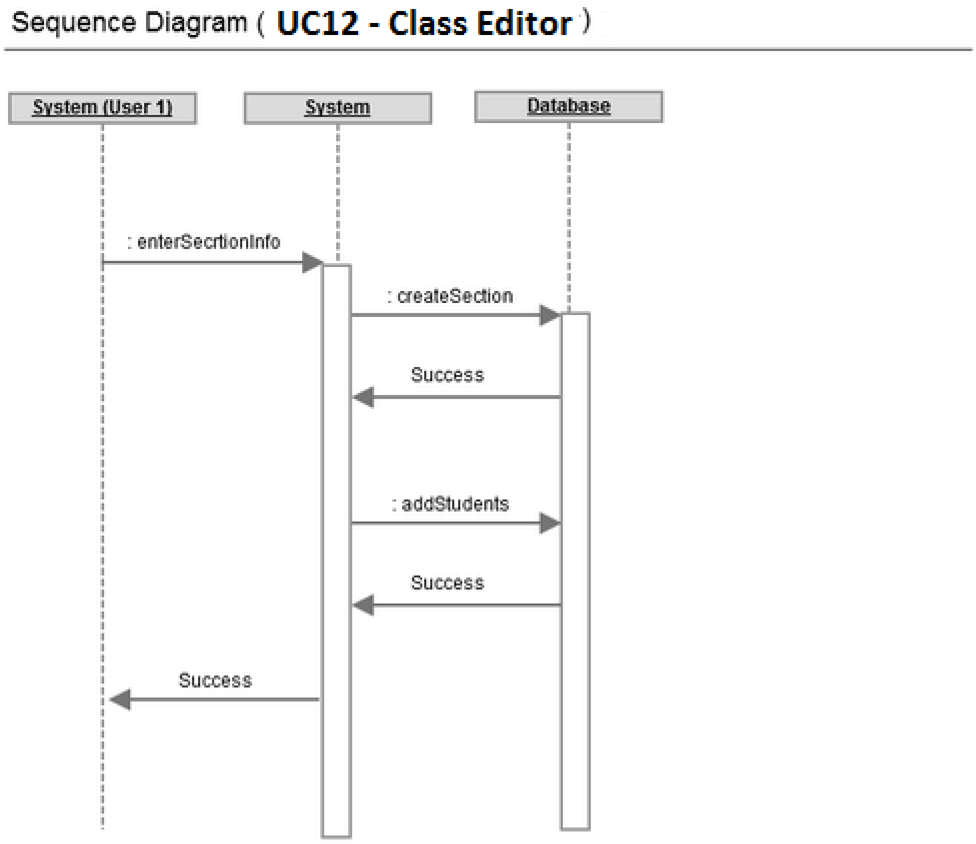
UC11 (Student Editor) Sequence Diagrams:
Success Case:



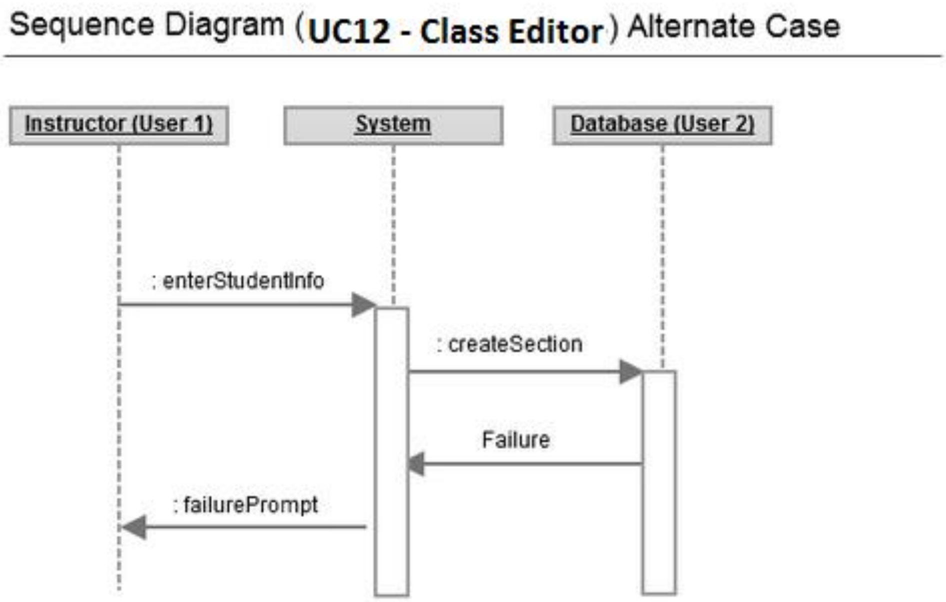
Alternate (Failure) Case:



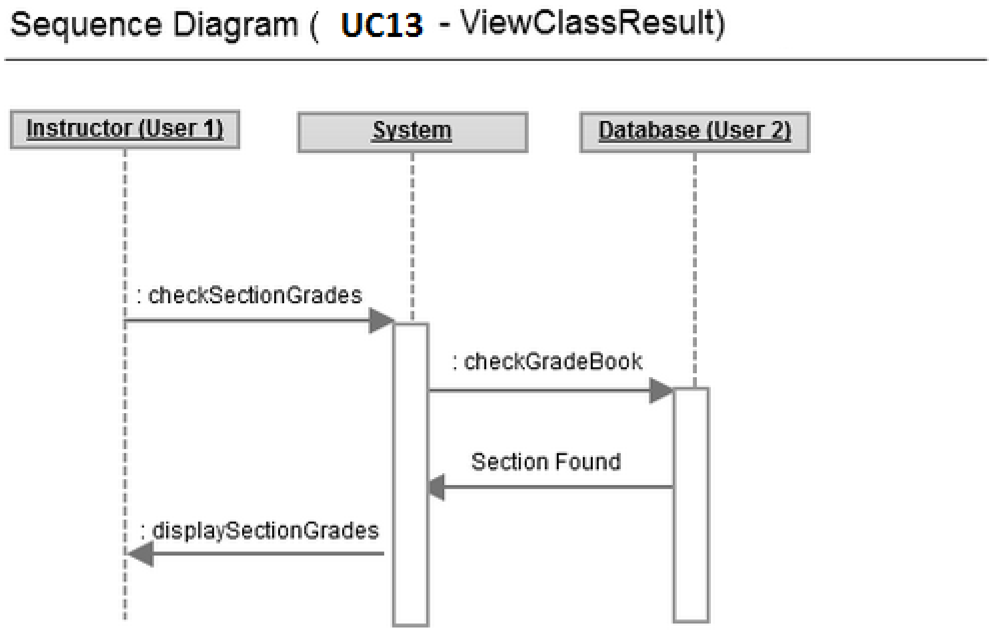
UC12 (Class Editor) Sequence Diagrams:
Success Case:



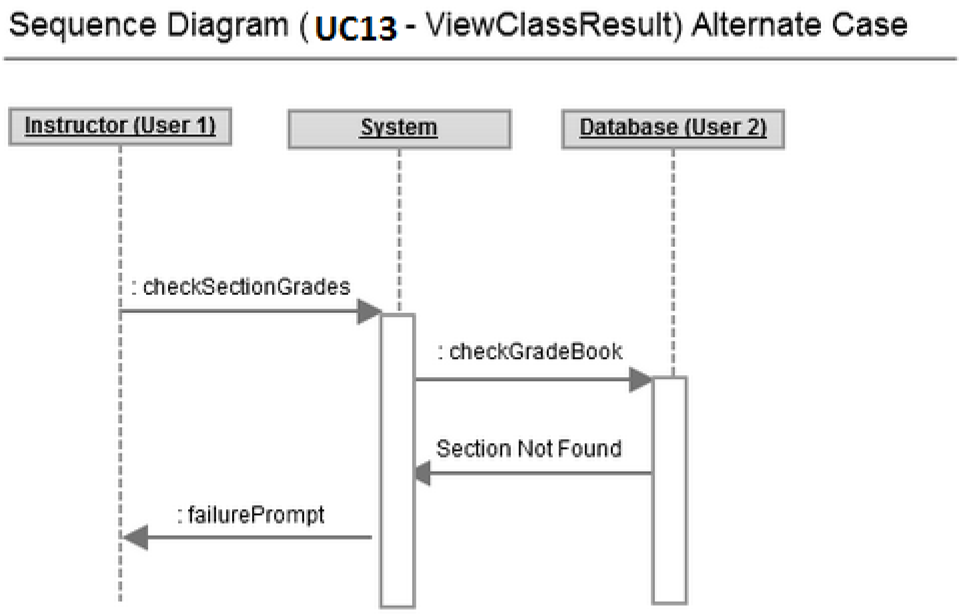
Alternate (Failure) Case:



UC 13 (View Class Result) Sequence Diagrams:
Success Case:

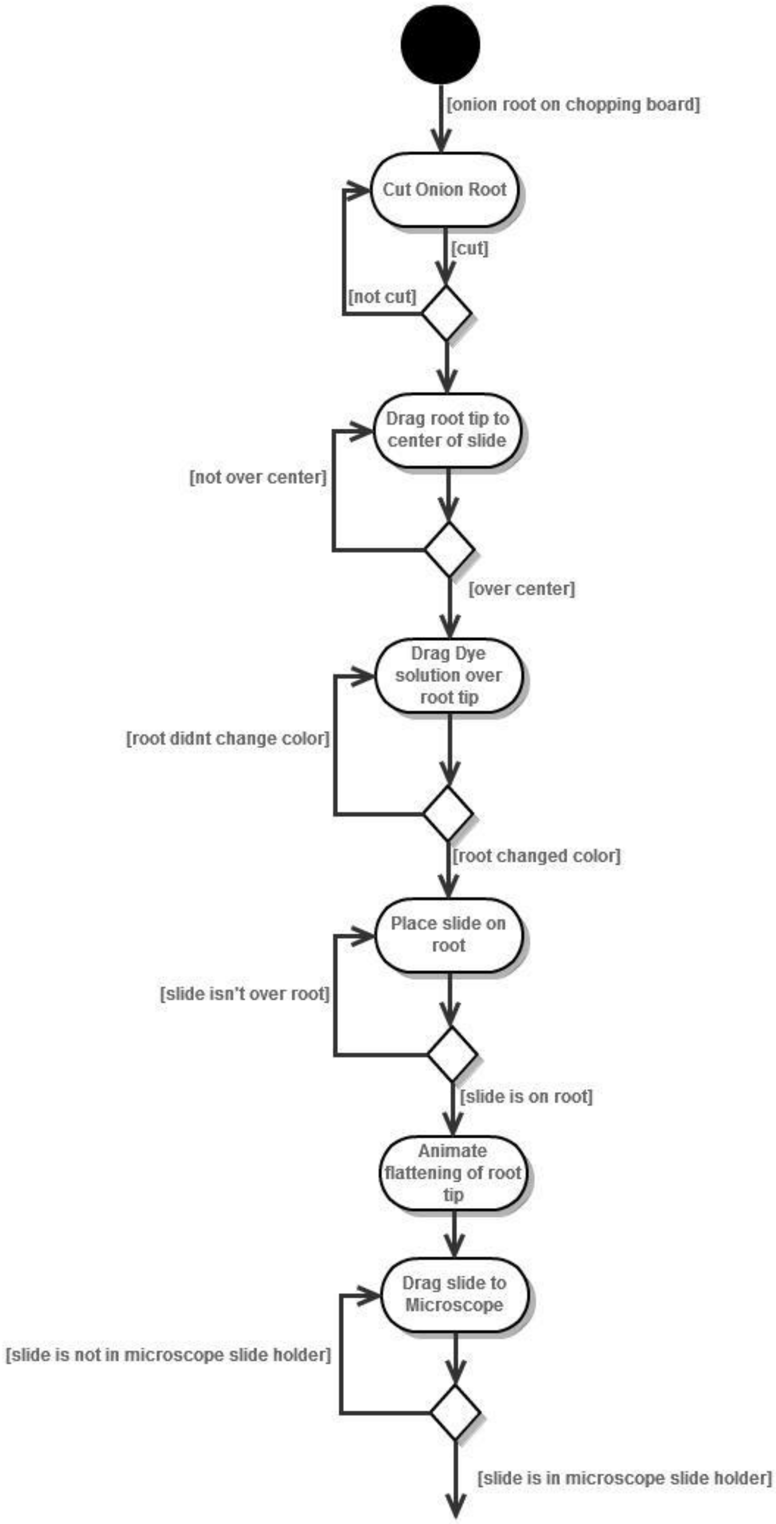


Alternate (Failure) Case:



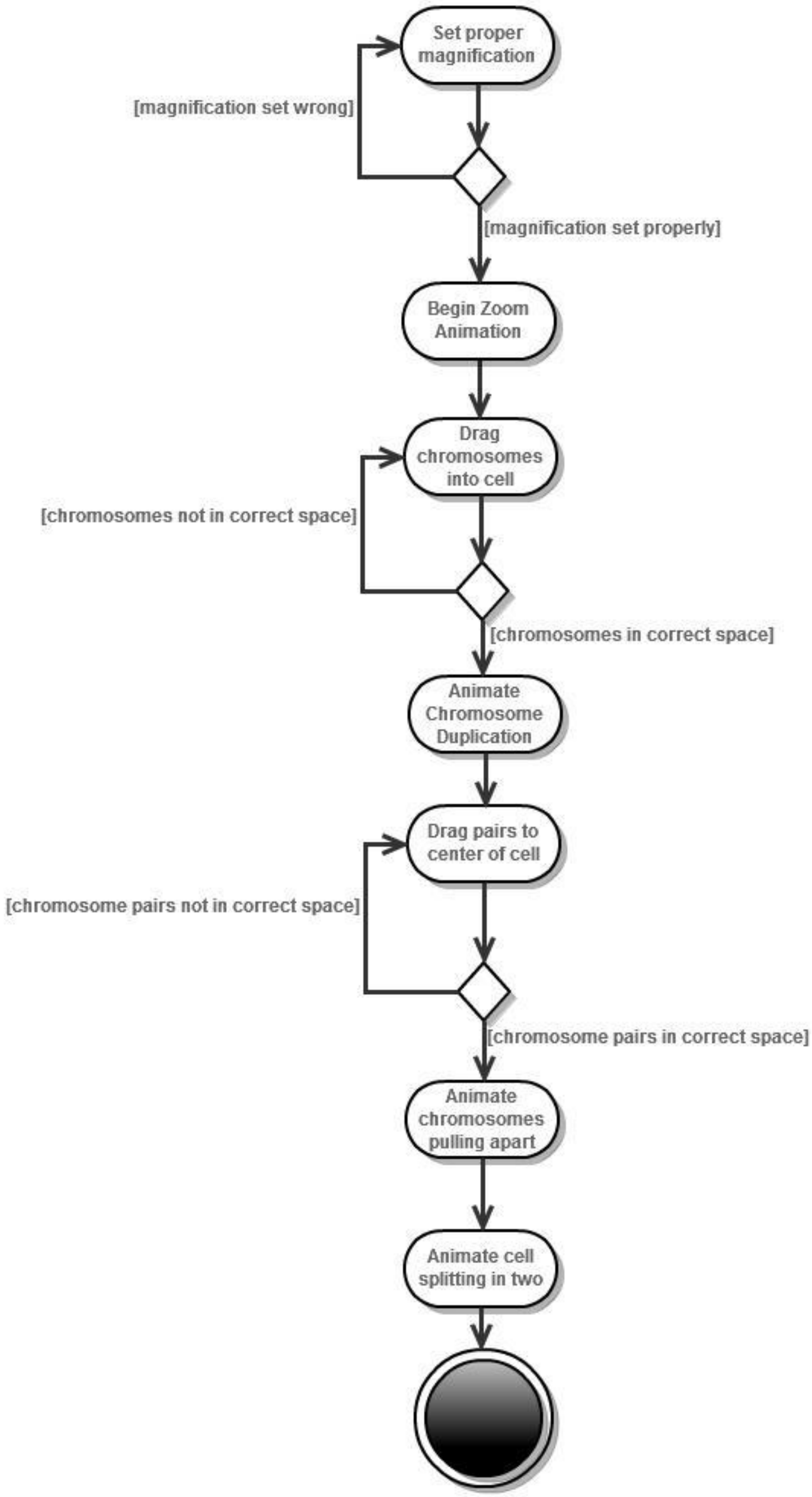
4.6 Activity Diagrams

Laboratory One Cell Division Activity



Diagram

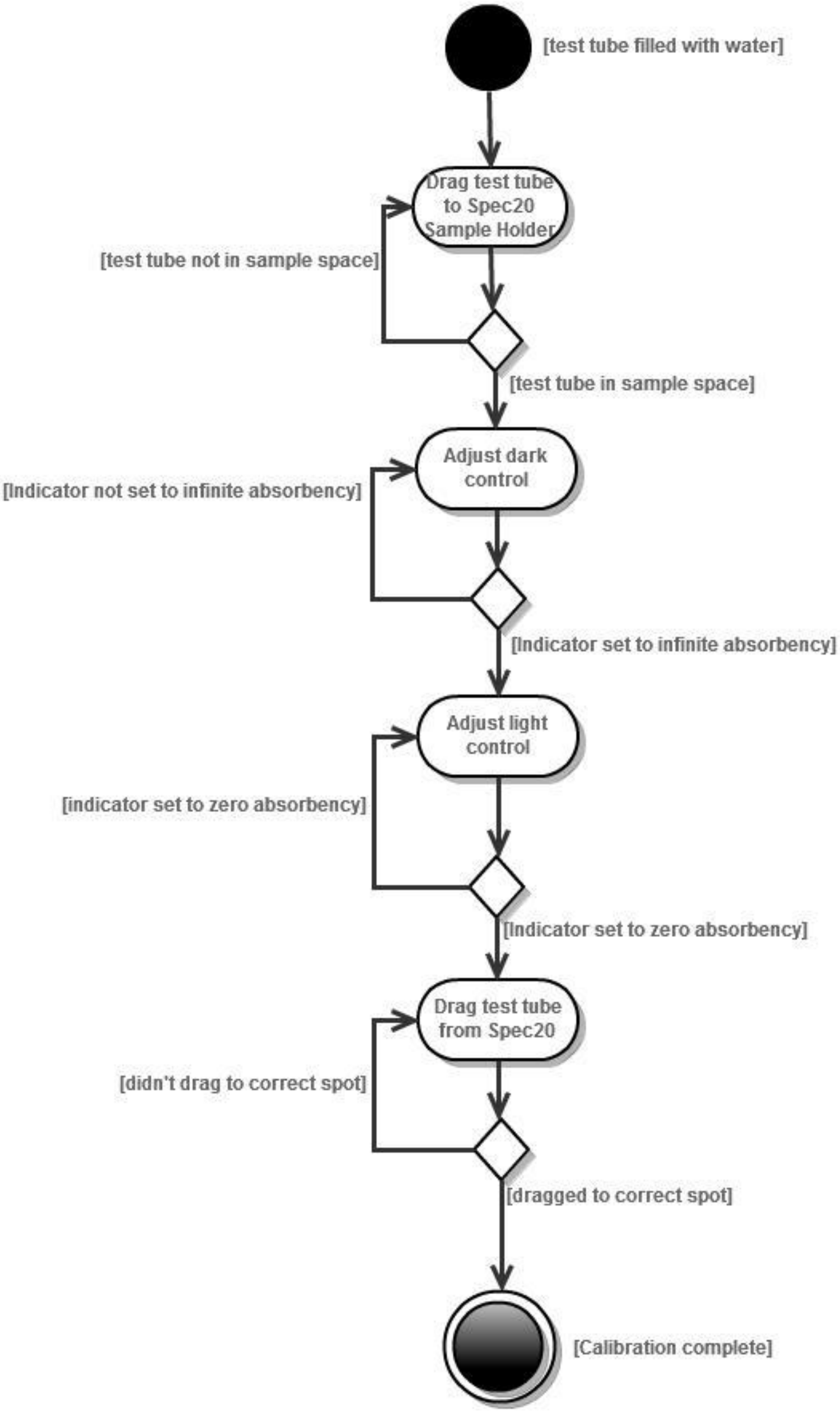
Continued from previous



page

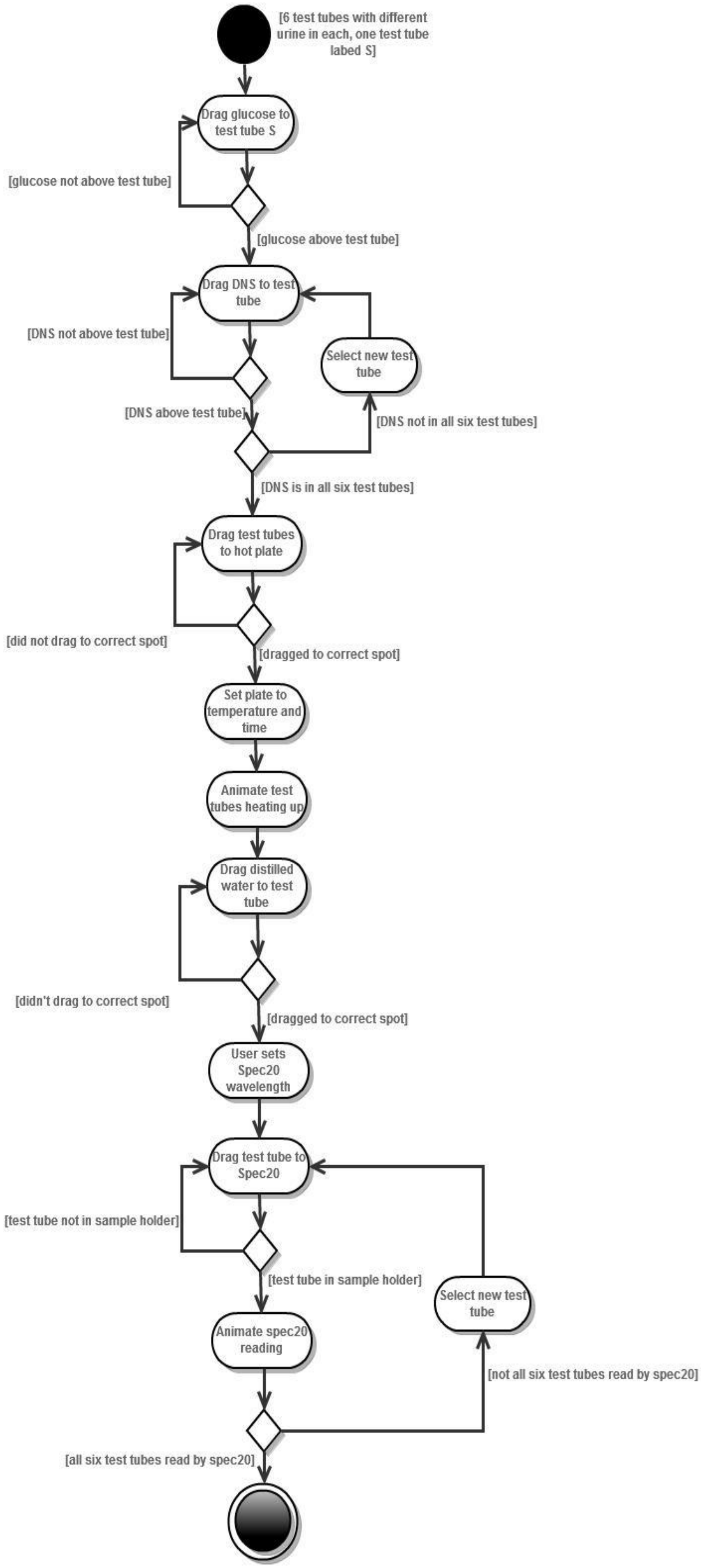
This diagram shows all of the steps involved in the first laboratory, it consists of two parts, first the user using the virtual laboratory to prepare a slide, place it in the microscope, setting the magnification and then viewing the slide. While viewing the slide the user will be able to interact with the chromosomes and view the animations of a cell going through all of the different stages of cell division.

Laboratory Two Biological Molecules Activity Diagram



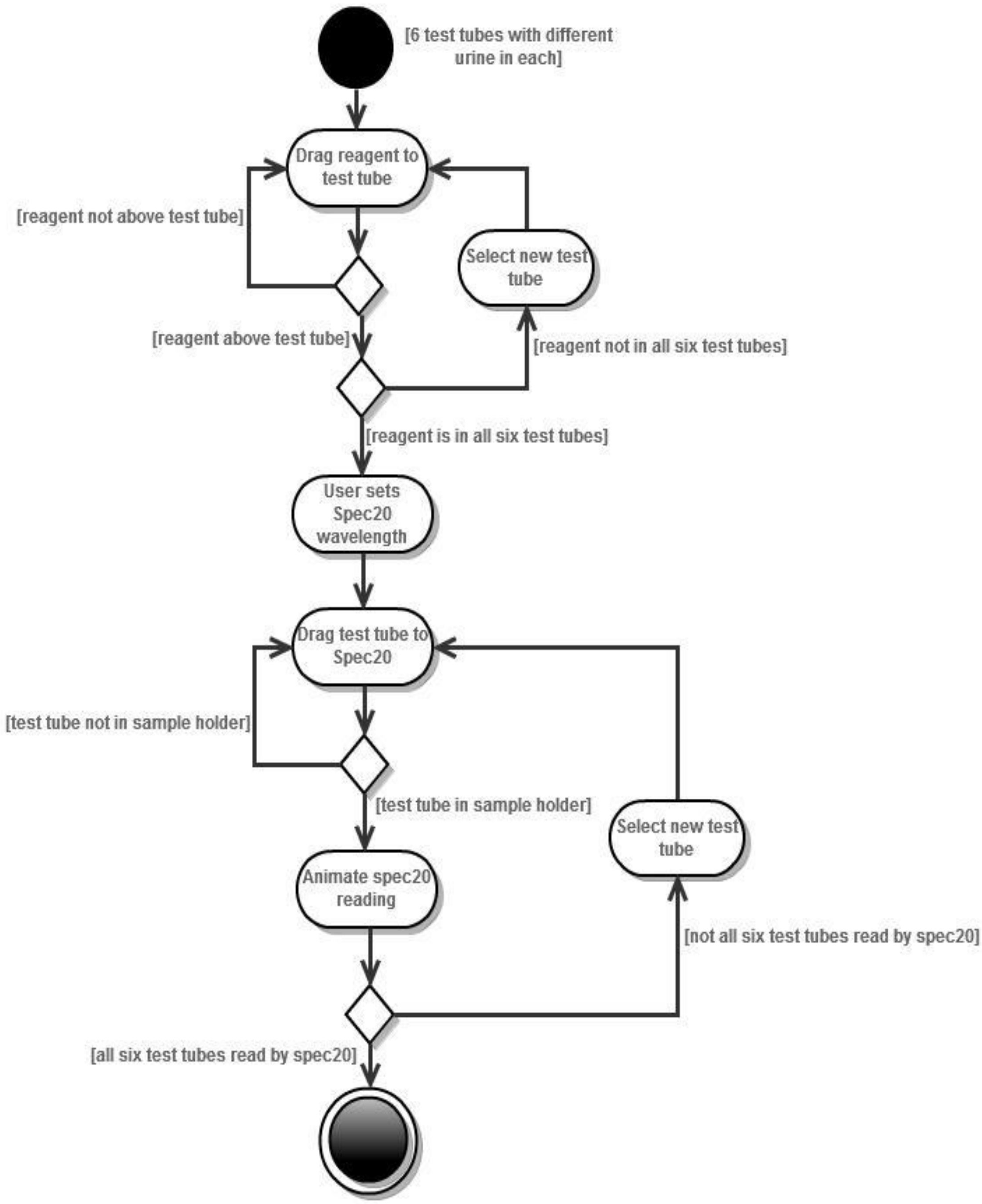
The activity diagram shown on the previous page explains the states that are gone through when calibrating the Spec20 Spectrophotometer. This process will be used at the beginning of the second and third laboratories to set the Spec20 to the correct zero and infinite absorbancies.

The parts to this lab are shown in separate state diagrams to make the flow of the laboratory more easy to understand. Each part below will be repeated for all 6 test tubes but the state diagrams are only shown once. Since the goal is to compare the properties of the components of each test tube, as the lab progresses, a table will be filled in with the observed results so that the student does not need to keep track. Below, in the first part is the state diagram for testing the pH of each patients urine. This involves just the student dragging each test tube to the pH paper and then the animation of the pH paper changing color to indicate the pH of the urine.

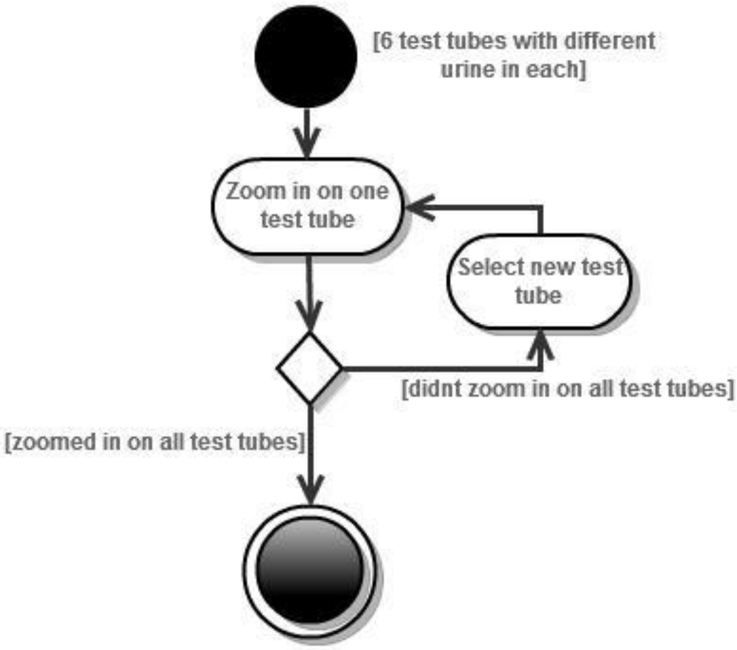

[6 test tubes with different urine in each, one test tube labed S]
Drag glucose to test tube S
[glucose not above test tube]
[glucose above test tube]
Drag DNS to test tube
[DNS not above test tube]
Select new test tube
[DNS above test tube]
[DNS not in all six test tubes]
[DNS is in all six test tubes]
Drag test tubes to hot plate
[did not drag to correct spot]
[dragged to correct spot]
Set plate to temperature and time
Animate test tubes heating up
Drag distilled water to test tube
[didn't drag to correct spot]
[dragged to correct spot]
User sets Spec20 wavelength
Drag test tube to Spec20
[test tube not in sample holder]
[test tube in sample holder]
Select new test tube
Animate spec20 reading
[not all six test tubes read by spec20]
[all six test tubes read by spec20]

This diagram on the previous page shows the second part to the laboratory where the user will add liquids to the test tubes, place them on the hot plate to heat up and then use the spec 20 to read each test tubes absorbance. Since not every step must be completed 6 times, there is a not on the side as to which items must be repeated by the user

The state diagram below shows the third interaction between the user and the Spectronic 20 machine, this time testing for the level of protein in the urine. It is very similar to the previous section of the second laboratory.

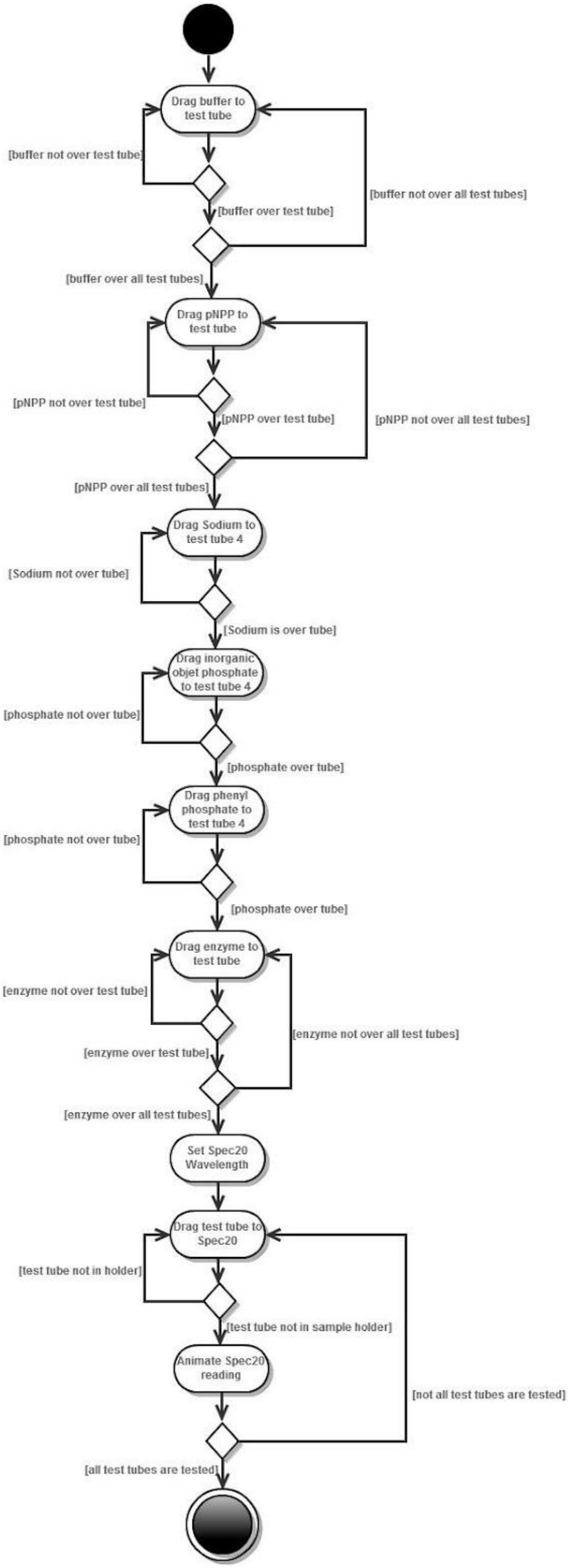


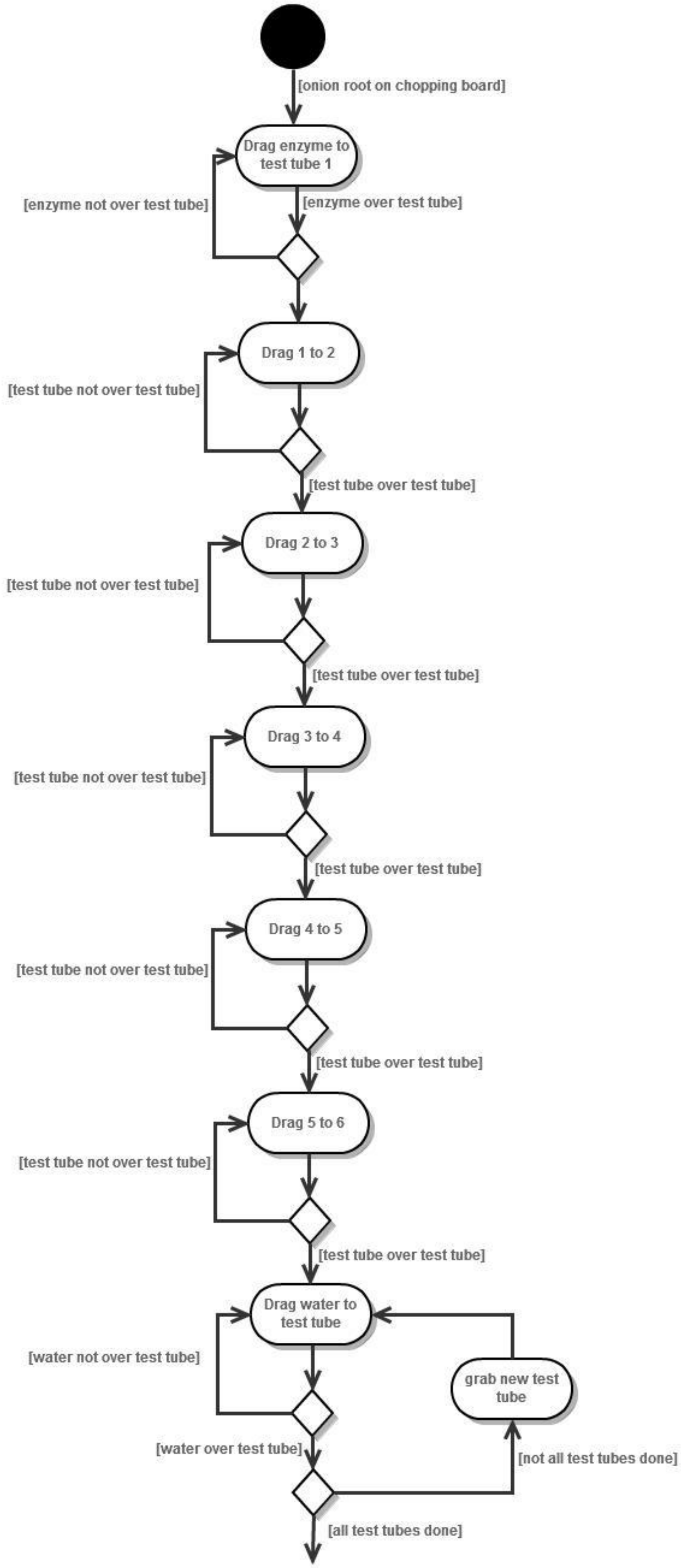
Below is the last activity diagram for the second laboratory, in this the system will just show the user each test tube and have them select which color it appears to be.



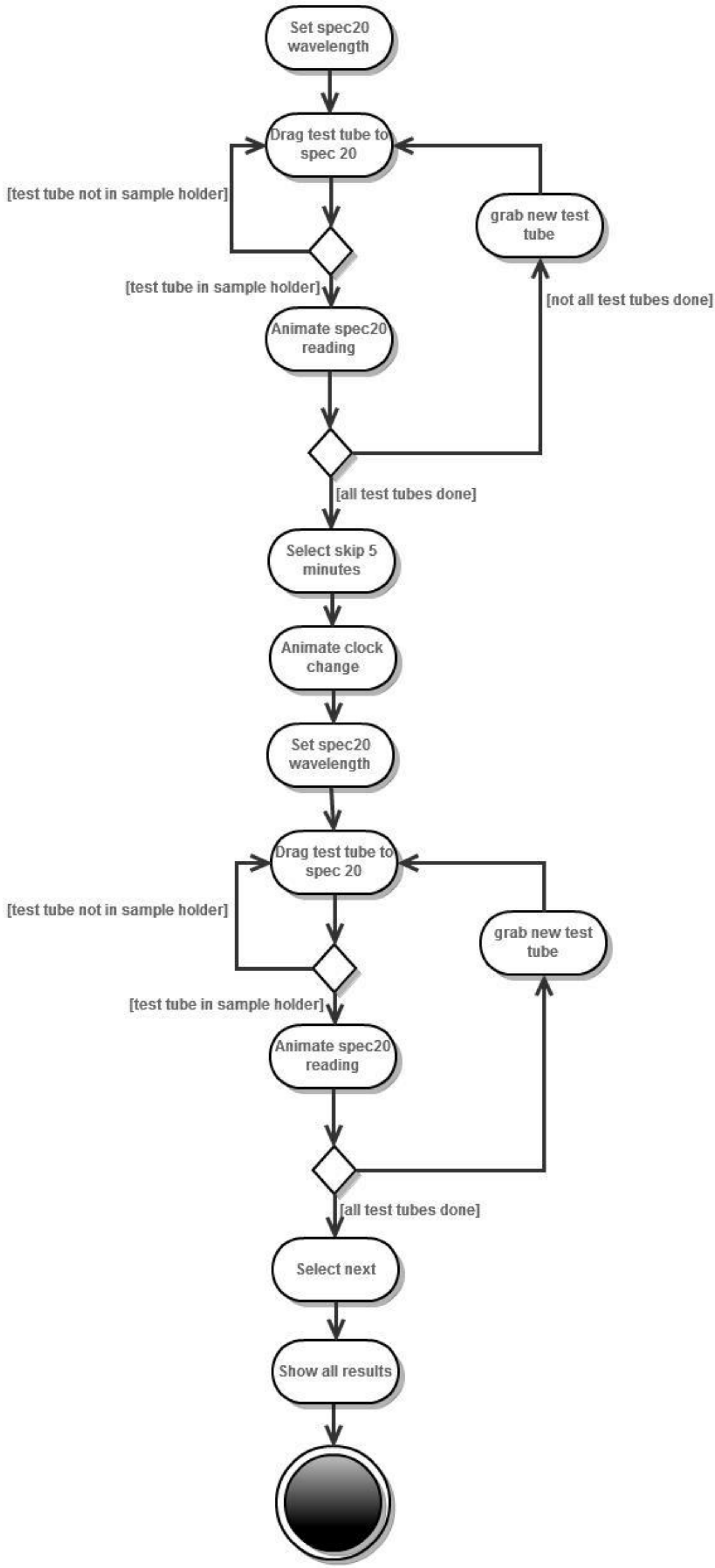
Laboratory 3 Enzyme State Diagram (next page)

The above diagram is the state diagram for the first part of the third laboratory. In this activity the user must use different liquids to fill each of the 6 test tubes, after this they must drag each test tube to the spec20 to get its absorbance and time before it reaches an absorbance of 3 or 4.

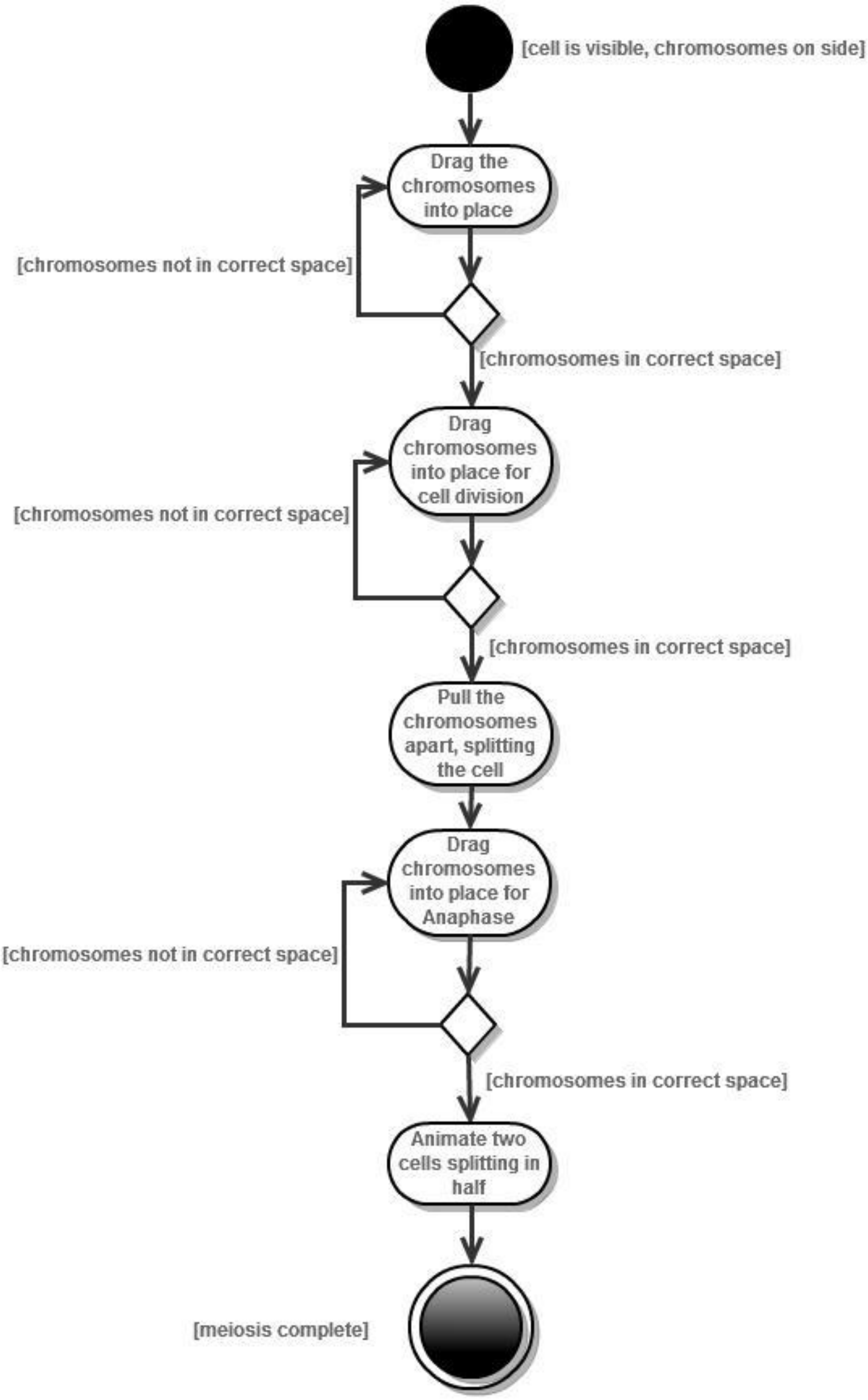

Drag buffer to test tube
[buffer not over test tube]
[buffer over test tube]
[buffer not over all test tubes]
[buffer over all test tubes]
Drag pNPP to test tube
[pNPP not over test tube]
[pNPP over test tube]
[pNPP not over all test tubes]
[pNPP over all test tubes]
Drag Sodium to test tube 4
[Sodium not over tube]
[Sodium is over tube]
Drag inorganic objet phosphate to test tube 4
[phosphate not over tube]
[phosphate over tube]
Drag phenyl phosphate to test tube 4
[phosphate not over tube]
[phosphate over tube]
Drag enzyme to test tube
[enzyme not over test tube]
[enzyme over test tube]
[enzyme not over all test tubes]
[enzyme over all test tubes]
Set Spec20 Wavelength
Drag test tube to Spec20
[test tube not in holder]
[test tube not in sample holder]
Animate Spec20 reading
[not all test tubes are tested]
[all test tubes are tested]

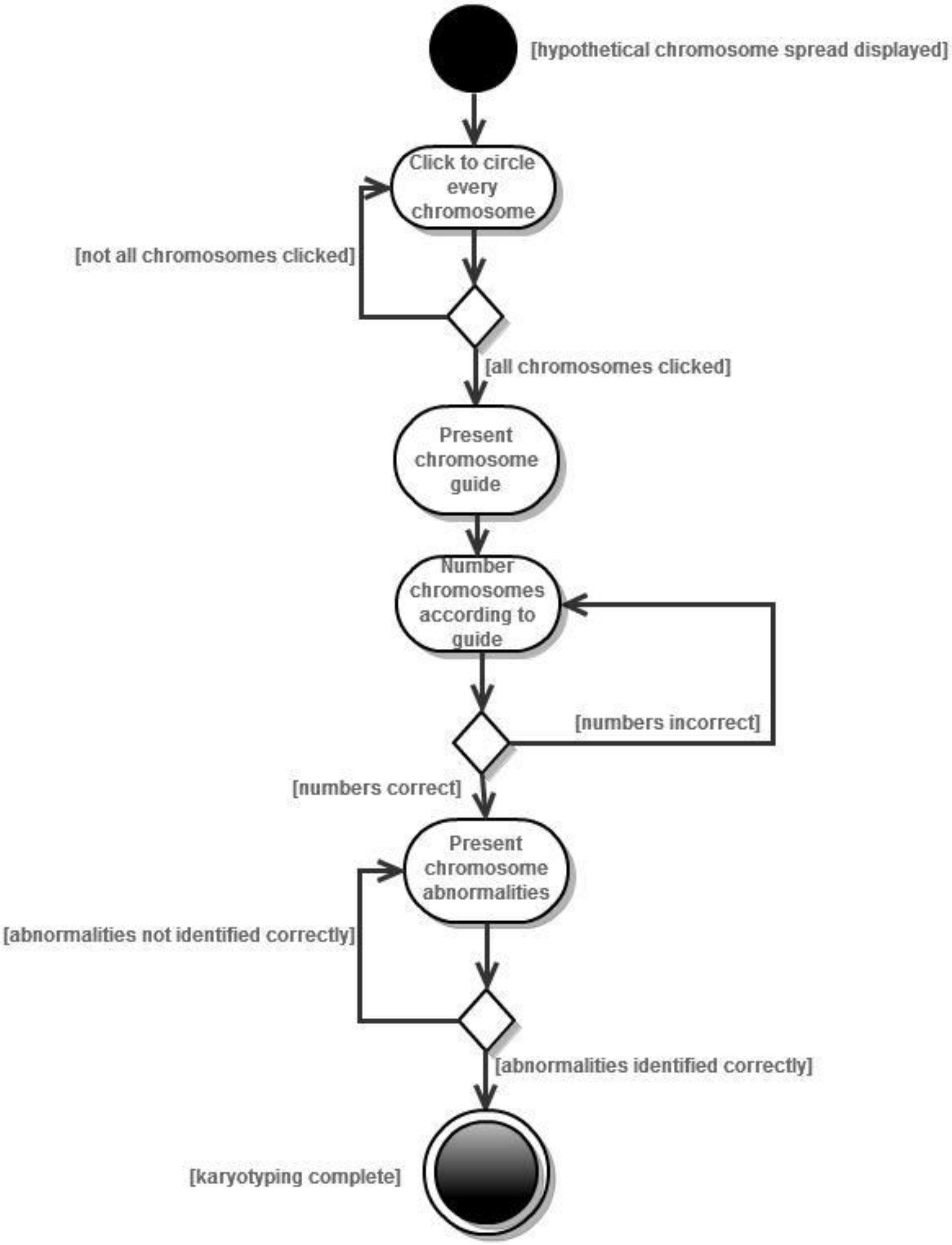

[onion root on chopping board]
Drag enzyme to test tube 1
[enzyme not over test tube]
[enzyme over test tube]
Drag 1 to 2
[test tube not over test tube]
[test tube over test tube]
Drag 2 to 3
[test tube not over test tube]
[test tube over test tube]
Drag 3 to 4
[test tube not over test tube]
[test tube over test tube]
Drag 4 to 5
[test tube not over test tube]
[test tube over test tube]
Drag 5 to 6
[test tube not over test tube]
[test tube over test tube]
Drag water to test tube
[water not over test tube]
grab new test tube
[water over test tube]
[not all test tubes done]
[all test tubes done]

The first part to this section of the laboratory involved diluting an enzyme so that the first test tube contains the most concentrated mixture of water and enzyme and the last test tube holds the least concentrated mixture of water and the enzyme. The user will accomplish this in the virtual biology lab by dragging each test tube to the next and emptying half of its contents into the next test tube. After this is complete the user must drag the distilled water to fill each of the test tubes. Below is the state diagram for the rest of this laboratory that involves entering each test tube into the Spec20 and reading the absorbancey, selecting to skip ahead 5 minutes and then re reading all of the absorbancies.


Set spec20 wavelength
Drag test tube to spec 20
[test tube not in sample holder]
grab new test tube
[test tube in sample holder]
[not all test tubes done]
Animate spec20 reading
[all test tubes done]
Select skip 5 minutes
Animate clock change
Set spec20 wavelength
Drag test tube to spec 20
[test tube not in sample holder]
grab new test tube
[test tube in sample holder]
Animate spec20 reading
[all test tubes done]
Select next
Show all results

Laboratory 4 Meiosis Activity Diagram





These two activity diagrams show the fourth laboratory, on meiosis. This lab has two parts, the first is very similar to the first laboratory because it shows the interactions between the user and the chromosomes within a cell in order to show the type of cell division called meiosis. In the second part is shows the interactions between the user and the Karyotype picture and how the system aids the user in finding and circling the chromosome defect in the picture.

5. Nonfunctional Requirements

Identifier	PW	Requirement
10	1	System should prohibit students from viewing other student grades, with clear differentiation between student and instructor. (Permissions)
11	3	System should randomize question order and answer order to prevent cheating from occurring.
12	5	System should store student grades into a secure database. Preferably MySQL.

The first nonfunctional requirement is key to providing security of the users information. If there is no differentiation between student and instructor, then any user can see the grades of all the students in the class, which is unacceptable in a school environment. This functionality is key to the professor because without it the quizzes cannot be used to grade the class overall and only provide the individual student with a metric of how they are doing int eh course.

Requirement 11 is important to improve the quality of the work the students produce. If the questions on the quizzes are not randomized in any way, the students can easily copy off one student who has gotten the answers after completing the quiz already. This can be taken a step further by implementing slight changes in the questions to give different answers while still testing the same material.

Similar to the first nonfunctional requirement, these grades need to be kept secure for two purposes. One is so that students can not view the grades of other students in the course, which is already protected on a basic level by the first requirement. The second and more applicable to the ever growing population of tech aware students. If the grades are stored in a simple text/spreadsheet file on the web then any user with basic web directory knowledge could access it, and possibly even change the grades. By placing this data into a secure database, the worry of students modifying grades is far less.

6. Effort Estimation using Use Case Points

The requirements and formula for Effort Estimation are listed here (copied from textbook):

The formula for calculating UCP is composed of three variables:

1. Unadjusted Use Case Points (UUCP), which measures the complexity of the functional requirements

2. The Technical Complexity Factor (TCF), which measures the complexity of the nonfunctional requirements

3. The Environment Complexity Factor (ECF), which assesses the development team's experience and their development environment

UCP = UUCP * TCF * ECF

Unadjusted Use Case Points (UUCPs) are computed as a sum of these two components:

1. The Unadjusted Actor Weight (UAW), based on the combined complexity of all the actors in all the use cases.

2. The Unadjusted Use Case Weight (UUCW), based on the total number of activities (or steps) contained in all the use case scenarios.

We will calculate the UUCP first, then the TCF, and then finally the ECF. After all of it is calculated, we will find the total Use Case Points (UCP).

6.1 Unadjusted Use Case Points (UUCP)

Here is where we will calculate the Unadjusted Actor Weight (UAW) and the Unadjusted Use Case Weight (UUCW)

6.1.1 Unadjusted Actor Weight (UAW)

Actor Name	Description of relevant characteristics	Complexity	Weight
Student	Student is interacting through a graphical user interface in order to complete nearly all of his tasks	Complex	3
Instructor	Instructor is interacting through a graphical user interface to see results and data of actor Student	Complex	3
Database	Database is another system interacting through a protocol	Average	2

UAW(Virtual Biology Lab) = 1 * Average + 2 * Complex = 1 * 2 + 3 * 2 = 8

6.1.2 Unadjusted Use Case Weight (UUCW)

Use Case	Description	Category	Weight
Select Lab	1 Initiating (and participating) actor (student) with 2 events	Simple	5
Place Item	1 Initiating (and participating) actor (student) with 4 events	Simple	5
Cut	1 Initiating (and participating) actor (student) with 4 events	Simple	5
Use Hotplate	1 Initiating (and participating) actor (student) with 2 events	Simple	5
Use Microscope	1 Initiating (and participating) actor (student) with 2 events	Simple	5

Pour Item	1 Initiating (and participating) actor (student) with 4 events	Simple	5
Use Spec20	1 Initiating (and participating) actor (student) with 4 events	Simple	5
Label	1 Initiating (and participating) actor (student) with 4 events	Simple	5
Take Lab Quiz	1 Initiating actor (student) with 2 participating (student and Database) with 4 events	Average	10
View Student Result	2 initiating actors (professor and student) with 3 participating (student, professor, database) with 2 events	Average	10
Student Editor	1 initiating actor (professor) with 2 participating (professor, database) with 4 events	Average	10
Class Editor	1 initiating actor (professor) with 2 participating (professor, database) with 3 events	Average	10
View Class Result	1 initiating actor (professor) with 2 participating (professor, database) with 2 events.	Simple	5

UUCW(Virtual Biology Lab) = 9 * Simple + 4 * Average = 9 * 5 + 4 * 10 = 85.

Total UUCP = UUCW + UAW = 85 + 8 = 93

6.2 Technical Complexity Factor (TCF)

Technical Factor	Description	Weight	Perceived Complexity	Calculated Factor (WeightxPerceived Factor)
T1	Distributed, Web-based system (due to view lab grades and student results)	2	3	2x3 = 6
T2	Lab information requires efficient and quick response times, however everything else (storing of scores and starting labs) only needs to be adequate, not exceptional	1	4	1x4 = 4
T3	End users expect good and smooth lab efficiency	1	4	1x4 = 4
T4	No huge internal processing other than calculating grades	1	0	1x0 = 0
T5	Very reusable - all labs are built on similar frameworks and integrated into modular website	1	1	1x1 = 1
T6	No installation required (besides flash)	0.5	0	0.5x0 = 0
T7	Very easy to use - web browser +	0.5	1	0.5x1 = 0.5

	mouse/keyboard			
T8	Very portable - usable on any computer with a web browser and flash installed	2	1	2x1 = 2
T9	Easily able to add or remove features due to modular design of lab assignments and students/classes	1	1	1x1 = 1
T10	Absolutely concurrent, allows multiple people to use/do the same lab at the same time	1	4	1x4 = 4
T11	Hashing of students passwords/logins, but other then that nothing to note	1	3	1x3 = 3
T12	Yes, anyone can access it assuming they have internet and a valid login	1	1	1x1 = 1
T13	No special training facilities required	1	0	1x0 = 0

Technical Factor Total: 26.5
Constant-1 (C1) = 0.6
Constant-2 (C2) = 0.01
TCF = 0.6 + (0.01x26.5) = 0.865. This results in a reduction of the UCP by 13.5%.

6.3 Environmental Complexity Factor (ECF)

Environmental Factor	Description	Weight	Perceived Complexity	Calculated Factor (WeightxPerceived Factor)
E1	Beginner familiarity with the UML-based development	1.5	1	1x1.5 = 1.5
E2	Decent amount of application solving experience	0.5	2	0.5x2 = 1
E3	Some knowledge of object oriented approach	1	2	1x2 = 2
E4	Beginner lead analyst	0.5	1	1x0.5 = 0.5
E5	Highly motivated with the occasional slacking	1	4	1x4 = 4

E6	Very stable requirements	2	5	2x5 = 10
E7	No part time staff	-1	0	-1x0 = 0
E8	Flash is an easy, but monotonous and tedious programming language	-1	3	-1x3 = -3

Environmental Factor Total: 16
Constant-1 (C1) = 1.4
Constant-2 (C2) = -0.03
ECF = 1.4 + (-0.03*16) = .92. Therefore, UCP will decrease by a factor of 8%.

6.4 Effort Estimation

Now that we have final values for our UUCP, TCF and ECF, our final UCP value is:
UCP = UUCP * TCF * ECF
UCP =~ 74

Now, to calculate our Effort Estimation value, we must multiply our UCP by a Productivity Factor (PF). For this project, our PF will be set at 28, which is on the higher end (assigned based on the experience of the group. Since we are Undergraduates, a 28 is the best for calculation). Therefore, our final duration value will be:

Duration = PF x UCP = 74x28 = 2072 hours.
2072 hours / 6 people = 345 hours/person.

As shown, our total duration is just over 2000 hours, and divided by each person in the group, we get ~350 hours/person. This is pretty reasonable since we've been doing this project since the beginning of the semester, and is decently accurate (probably a little high) with how things have actually been going.

7. Domain Analysis

7.1 Domain Model

7.1.1 Concept Definitions

<u>General Concepts</u>

Responsibility Description	**Type**	**Concept**

		Name
Coordinates actions with use cases	D	Controller
Displays information to the user through their web browser	K	Interface
Keeps track of all users in the system	K	UserDatabase
Opens the selected laboratory at the point that student left off	D	LabOpener
Keeps track of where each student is in every laboratory, what grades each of them have received	K	LabProgress
Finds the grades for the selected student or class section and displays them	D	GradeAccessor
Keeps track of all students registered to the class, their log in information and what section they belong in	K	ClassInfo
Provides the correct quiz for that laboratory, randomize questions	D	Quizzer
Includes a database of questions to be asked	K	QuizQuestions
Edits a preexisting section	D	ClassEditor
Edits a preexisting student	D	StudentEditor
Starts the simulation of the Spectronic 20 Specrophotometer	D	Spec20
Starts the simulation of the Microscope	D	Microscope
Starts the simulation of the laboratory space	D	LabSpace
Keeps track of simulated karyotypes of human chromosomes	K	Karyotype

Concepts for Laboratory One

Responsibility Description	**Concept Name**
Object, can zoom in to see cell shape and processes	OnionRootTip
Container that holds material	Slide
Liquid that colors in cell	DyeSolution
Machine used to heat up slide	HotPlate
Covers the slide	CoverSlip
Cell that exists within onion root	PlantCell
Example cell that shows mitosis	AnimalCell

Concepts for Laboratory Two

Responsibility Description	**Concept Name**
Container for liquids	TestTube

Subjects urine, what the student is running tests on	Urine
Tests the pH level of liquid	pHPaper
Concentrated glucose, used for comparison	Glucose
Concentrated protein, used for comparison	Protein
Water used for blank and to dilute liquids	DistilledWater

<u>Concepts for Laboratory Three</u>

Responsibility Description	**Concept Name**
Container for liquids	TestTube
Water used for blank and to dilute liquids	DistilledWater
Added to all of the test tubes, has a pH of 8.0	Buffer
A protein that increases the speed of rate of chemical reactions	pNPP
Liquid catalyst for a reaction	Sodium
Liquid catalyst for a reaction	Inorganicphosphate
Liquid catalyst for a reaction	Phenylphosphate
Water used for blank and to dilute liquids, used as a	DistilledWater

<u>Concepts for Laboratory Four</u>

Responsibility Description	**Concept Name**
Slide containing a cell that shows the stages of meiosis	Slide
Allows the student to circle the chromosomes on the karyotype	CircleChromosome
Allows the student to number the chromosome on the karyotype	NumberChromosome

7.1.2 Association Definitions

Concept Pair	Association Description	Association Name
Controller <-> Interface	Fetches and displays appropriate data to the user through the web interface based upon their input	Controls and Displays Web Page
Controller <-> GradeAccessor	Fetches a student or classes grades	Provides/Retrieves Data
Controller <-> LabOpener	Fetches data related to the selected lab	Provides/Retrieves Data
Controller <-> Quizzer	Fetches data related to the selected quiz	Provides/Retrieves Data
Controller <-> UserDatabase	Stores data related user accounts	Provides/Retrieves Data
Controller <-> ClassInfo	Processes actions related to user accounts	Provides/Retrieves Data
ClassEditor <-> ClassInfo	Creates, edits or deletes information about a specific class	Provides/Retrieves Data
StudentEditor<-> ClassInfo	Creates, edits or deletes information about a specific student	Provides/Retrieves Data
GradeAccessor <-> ClassInfo	Check to see if the correct person is trying to access grades	Verifies Data
GradeAccessor <-> LabProgress	Finds and provides the correct grade for that student	Provides/Retrieves Data
LabOpener <-> LabProgress	Finds out where the student last left off and records how much the student had completed when they stopped working on the lab	Provides/Retrieves Data
Quizzer <-> QuizQuestions	Finds and provides randomized questions for that laboratory	Provides Data
Quizzer <-> LabProgress	Records what grades students receives while completing the quizzes and if they successfully finished that lab	Provides/Retrieves Data
LabOpener <-> Spec20	Opens the simulation for the spec20 where the student will use the spec 20 machine to find the absorbance	Generates Simulation
LabOpener <-> Microscope	Opens the simulation for the microscope where the student will view the stages of mitosis and meiosis	Generates Simulation
LabOpener <-> LabSpace	Opens the simulation for the lab space, where the student will prepare slides and test tubes	Generates Simulation
LabOpener <-> Karyotype	Finds and provides a randomized picture of human chromosomes represented in a karyotype	Provides Data

Associations for Laboratory One

Concept Pair	**Association Description**
LabSpace <-> Slide	Student will put empty slide in lab space
OnionRootTip <-> Slide	Student will place the onion root tip onto the slide
DyeSolution <-> Slide	Student will drop dye solution onto slide
Slide <-> SlideCover	Student will cover slide with slide cover, squashing the onion root
Slide <-> HotPlate	Slide lays on hot plat, hot plate heats slide
Slide <-> Microscope	Slide goes onto microscope, microscope can be zoomed in and out
Microscope <-> AnimalCall	Microscope shows the structure of an animal cell going through mitosis
Microscope <-> PlantCell	Microscope shows the structure of a plant cell going through mitosis

Associations for Laboratory Two

Concept Pair	**Association Description**
LabSpace <-> TestTube	Student will place empty test tubes in lab space
Urine <-> TestTube	Students will specify how much urine to pour into test tube
Glucose <-> TestTube	Students will specify how much glucose to pour into test tube
Protein <-> TestTube	Students will specify how much protein to pour into test tube
DistilledWater <-> TestTube	Students will specify how much distilled water to pour into test tube
TestTube <-> pHPaper	Student will pour liquid from test tube to pH paper and pH paper will change color
TestTube <-> Spec20	Student will place test tube into sample slot of the Spectronic 20

Associations for Laboratory Three

Concept Pair	**Association Description**
LabSpace <-> TestTube	Student will place empty test tubes in lab space
TestTube <-> TestTube	Student will be able to drag one test tube to another and specify how much to add to the second from the first
Buffer <-> TestTube	Students will specify how much Buffer to pour into test tube
pNPP <-> TestTube	Students will specify how much pNPP to pour into test tube
Sodium <-> TestTube	Students will specify how much Sodium to pour into test tube
Inorganicphosphate <-> TestTube	Students will specify how much Inorganic phosphate to pour into test tube
Phenylphosphate <-> TestTube	Students will specify how much Phenylphosphate to pour into test tube
Enzyme <-> TestTube	Students will specify how much enzyme to pour into test tube
DistilledWater <-> TestTube	Students will specify how much distilled water to pour into test tube
TestTube <-> Spec20	Student will place test tube into sample slot of the Spectronic 20

Associations for Laboratory Four

Concept Pair	**Association Description**
LabSpace <-> Karyotype	Student will place empty test tubes in lab space
Slide<-> Microscope	Slide goes onto microscope, microscope can be zoomed in and out
CircleChromosome <-> Karyotype	Student can select a chromosome on the karyotype to circle
NumberChromosome <-> Karyotype	Student can select a chromosome on the karyotpye to number

7.1.3 Attribute Definitions

Concept	Attribute	Attribute Description
D1. Controller	Use case coordinator	Provides the medium between interface and execution of desired result
D2. Interface	Graphical Web Interface	Provides the medium between user and controller
D3. UserDatabase	Student Info	
	Student progress	
	Student Grade	
D4. LabOpener	Lab Opener	Opens the desired available lab
	Lab State	The state of the lab being opened is identical to how it was when previously closed
D5. LabProgress	Lab State	Maintains the actually state of the lab
	Grade	Contains the grade according to the state of the lab
D6. GradeAccessor	Grade	Contains the grade for each student
D7. ClassInfo	Lab Info	Contains information about each individual with respect to lab dates, sections and login
	Student Info	
D8. Quizzer	Quiz Database	Maintains the appropriate quiz questions pertaining to the lab
	Randomizer	Randomizes the quiz questions to be asked
D9 QuizQuestions	Quiz Content	Long list of possible questions pertaining to the lab
D10 ClassEditor	Class Content	Edits the content of a given section
D11 StudentEditor	Student Content	Edits the content of a given student
D12 Spec20	Wavelength(nm)	Depicts the wavelength of light being analyzed

	Device Control	Adjusts the light and dark parameters for measurment
	Data Holder	Holds sample in place for further examination
D13 Microscope	Device Control	Allows for optical zoom
	Data Holder	Holds sample slide in place for further examination
D14 LabSpace	Table	Holds cutting board or slide or any machine that the user is using
	Test Tube Holder	Shelf that holds any test tubes in the lab space
D15 Karyotype	Karyotype Content	Contains information regarding test DNA samples
D16 Onion Root Tip	Dye solution	changes the onion root from its original color. Cannot be seen under microscope without it
D17 Slide	Plant Cell	Slide can contain a plant cell
	Animal Cell	Slide can contain an animal cell
	Cover Slip	The cover on the slide, cannot be placed in the microscope without it
	Temperature	Is the temperature of the slide and its contents
D18 HotPlate	Device Control	changes the temperature of the hotplate
	Sample Holder	Holds the slide or the test tube that the user wants to heat up
D19 Test Tube	Urine	Test tube can contain urine
	Glucose	Test tube can contain glucose
	Protein	Test tube can contain protein
	Distilled Water	Test tube can contain distilled water
	Buffer	Test tube can contain buffer
	pNPP	Test tube can contain pNPP

	Sodium	Test tube can contain sodium
	Inorganic phosphate	Test tube can contain inorganic phosphate
	Phenylphosphate	Test tube can contain phenylphosphate
	Temperature	Is the temperature of the test tube and its contents
	Absorbance	Is the absorbance of the test tube and its contents
	pH	Is the pH of the test tube's contents
D20 pH Paper	pH	The pH of what was poured on it
D21 Circle Chromosomes	Circled	Contains if the chromosome was circled by the user
	Correct	Contains if that chromosome should be circled
D22 Number Chromosomes	numbered	Contains if the chromosome was numbered by the user

7.1.4 Traceability and Domain Model Diagrams

UC	PW	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	D11
1	9	x	x		x							
2	14					x						
3	9					x						
4	15					x						
5	15					x						
6	8					x						
7	14					x						
8	4					x						
9	1	x	x			x	x		x	x		
10	8	x	x	x			x	x				
11	6		x	x				x			x	
12	6		x	x				x				x
13	8	x	x	x			x	x				

UC	PW	D12	D13	D14	D15	D16	D17	D18	D19	D20	D21	D22
1	9											
2	14			x		x	x		x	x		
3	9			x		x						
4	15			x				x				
5	15		x	x			x					
6	8			x						x		
7	14	x		x					x			
8	4			x	x						x	x
9	1											
10	8											
11	6											
12	6											
13	8											

The two matrices above show the relationship between the 13 use cases and the 24 domain models. The first half of the domain models have a lot to do with the later use cases, the ones pertaining to using the grade book and student or class editor. The last domain concepts have more to do with the first few use cases, or the ones that have more to do with the biology and the actual implementation of the biology laboratories.

The domain model diagram, since it has so many concepts and relationships, has been split into two different partial domain diagrams that are connected by the lab space domain concept D14. This first diagrams, **Fig. 7.1,** shows that the user interacts with the system through the controller, which then calls on the correct domain concepts and tells the to either retrieve or provide data to another domain concept.

The second diagram, **Fig. 7.2** on the next page, shows the second partial domain model. This diagram connects to the first partial domain model diagram through the domain concept D14, the Lab Space. These domain concepts are more related to the biology laboratories than the gradebook and student profile concepts of the first diagrams. Depending on which laboratory is selected, the Lab Space domain concept D14 will generated the indicated number of domain concepts that it points to. The other relationships between these domain concepts are labeled on the diagram.

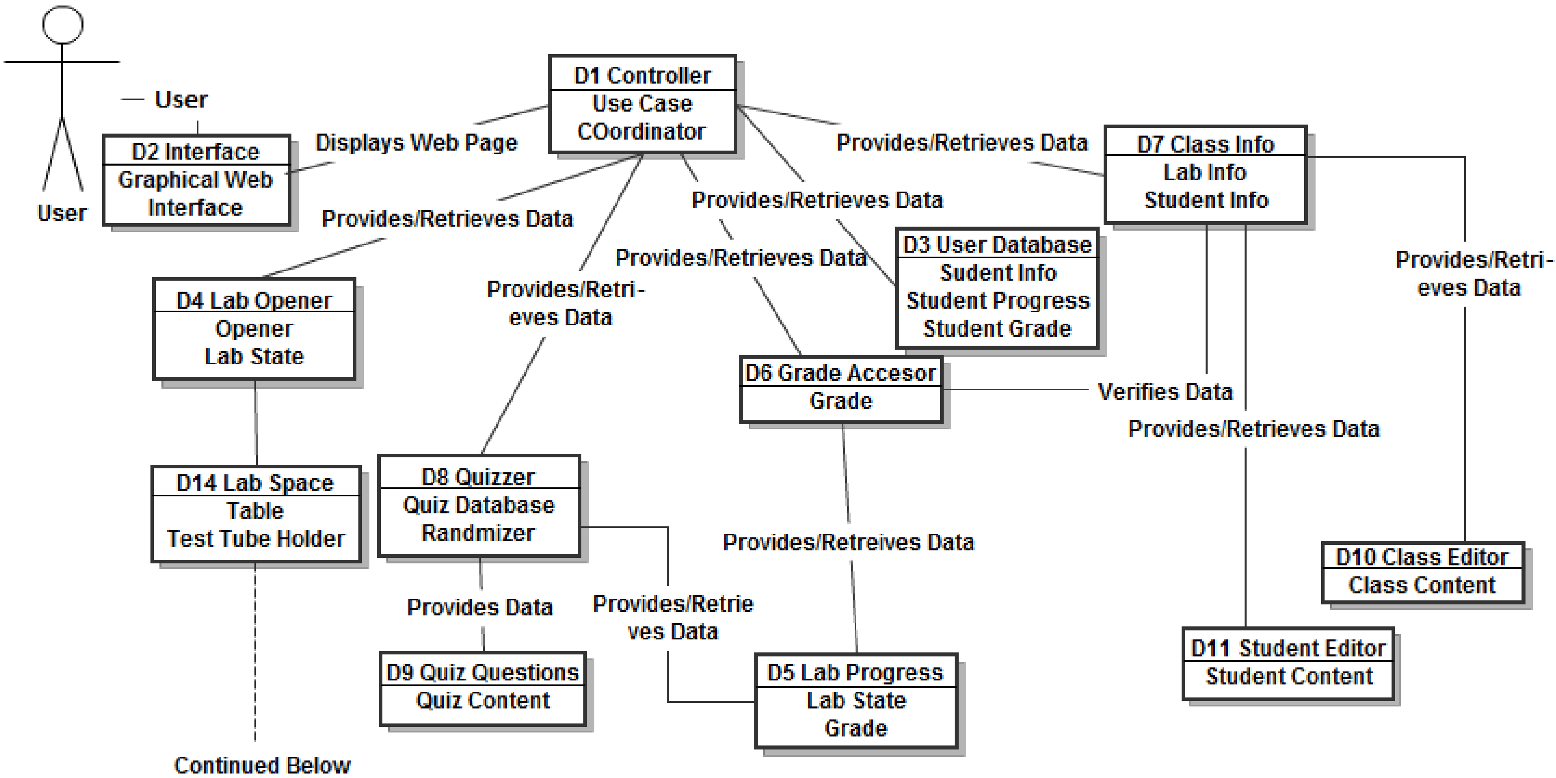


Fig. 7.1 First Partial Domain Model Diagram

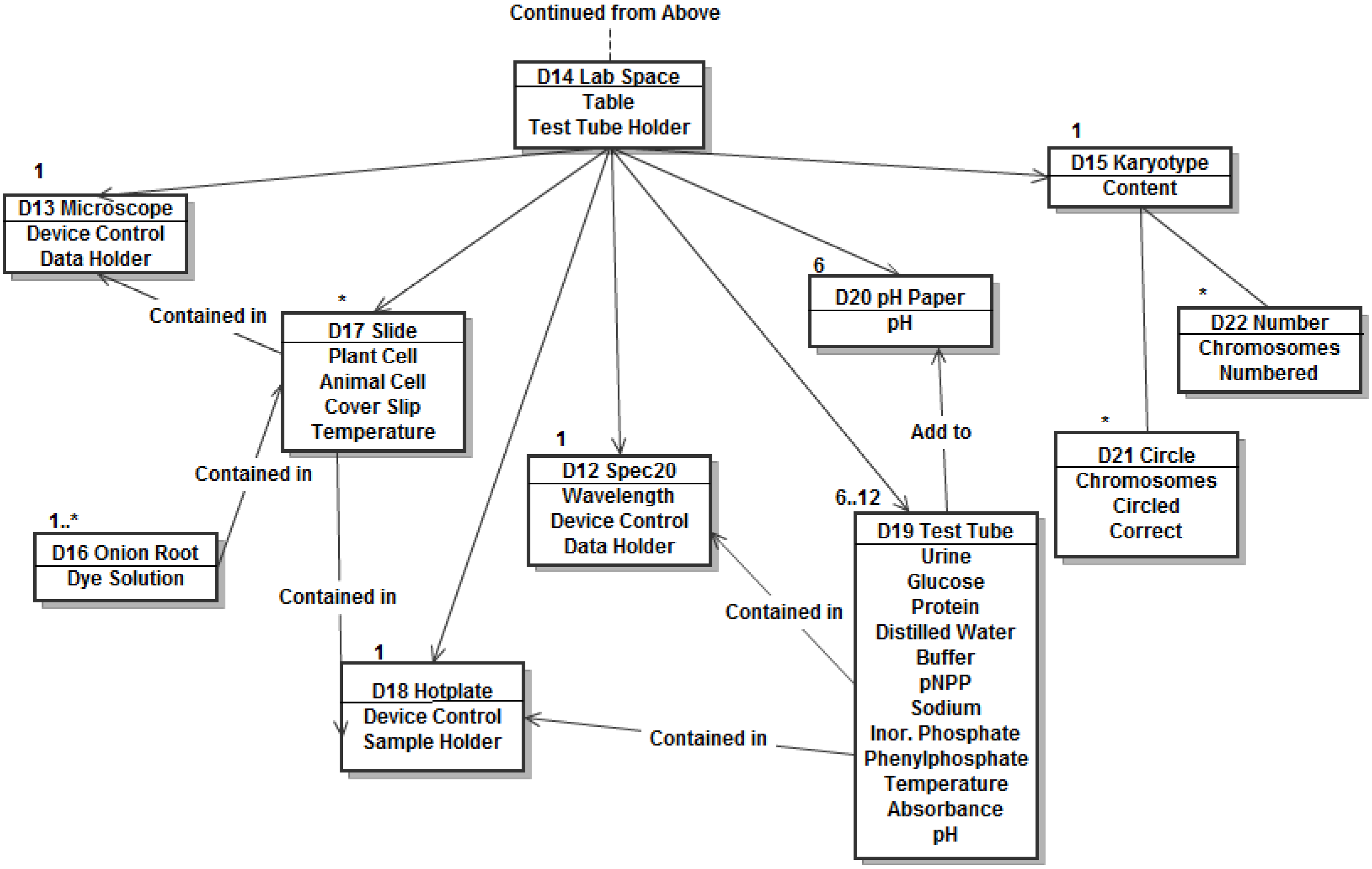


Fig. 7.2 Second Partial Domain Model Diagram

7.2 System Operation Contracts

Operation:	Select Lab **UC1**
Precondition:	• Student has not yet completed the lab • Student's lab progress is stored in UserDatabase (beginning of lab if no previous attempt was made or last completed task if lab was not finished in a previous attempt)
Postcondition:	• Student is not able to perform the lab • Student's lab progress is stored in UserDatabase (complete) • Student is able to take the quiz associated with the particular lab completed

Operation:	Place Item **UC2**
Precondition:	• User has not selected an item yet • Available objects to be selected are present
Postcondition:	• Object is grabbed by cursor and able to be moved around

Operation:	Cut **UC3**
Precondition:	• User has a free cursor • Available objects to be cut are present
Postcondition:	• Object is cut along desired line

Operation:	Use Hotplate **UC4**
Precondition:	• Hot plate is present in lab • No other object is currently on the hot plate • User has established a time and temperature for hot plate
Postcondition:	• Object is successfully on hot plate • User is able to see change in Object

Operation:	Use Microscope **UC5**
Precondition:	• Available microscope in lab • Nothing in slide holder of microscope
Postcondition:	• Object in slide holder of microscope • Set zoom buttons to appropiate magnification levels

Operation:	Pour Item **UC6**
Precondition:	• Object with liquid contents available • Another object which can have liquid poured on it/in it
Postcondition:	• Able to pour liquid onto/into new object • Updates properly new liquid level and appropriate reaction

Operation:	Use Spec20 **UC7**
Precondition:	• Available spec20 with nothing in test tube holder • Dark/light, wavelength, and read absorbance settings are correct
Postcondition:	• Properly display test tube in holder • Proper readings of the tested material

Operation:	Label **UC8**
Precondition:	• Available object which can have a label attached
Postcondition:	• Label placed on object • Appropriate name written on label

Operation:	TakeLabQuiz **UC9**
Precondition:	• Student has completed the lab associated with the selected quiz • Student has not yet attempted to take the quiz after completing the lab
Postcondition:	• Student is not able to take the quiz • Results are graded and stored in UserDatabase

Operation:	View Student Result **UC10**
Precondition:	• Student wishing to view is registered for class • Instructor wishing to view is in charge of class • Student has completed and submitted lab for grading
Postcondition:	• Grade assigned to student • UserDatabase updated with appropriate grade information

Operation:	Student Editor **UC11**
Precondition:	● Instructor has specified student to be in a section ● NetID has not already been registered
Postcondition:	● UserDatabase contains username and password for the student ● Student's NetID is recorded as being successfully registered in ClassInfo

Operation:	Class Editor **UC12**
Precondition:	● User is an instructor with administrative abilities ● Section hasn't already been registered yet
Postcondition:	● ClassInfo contains information regarding section time and availability ● UserDatabase contains student NetID's associated with the section ● Class is recorded as being successfully registered ● Current section is unavailable to be registered again

Operation:	View Class Results **UC13**
Precondition:	● Instructor in charge of class only one allowed to view ● UserDatabase contains all of the class grades
Postcondition:	● Instructor can see all of the grades to determine averages, curves, etc

7.3 Mathematical Model

For every question answered correctly counts as a correctly answered question. This total is then divided by the number of questions for the given quiz for that given lab. That calculation then shows as the user's grade for that given lab. Later after a user has fully completed all the labs, the system then takes care of calculating the average. The average is done by weighing each and every lab evenly, multiplying the available points for that lab by the grade for that lab. Adds all of them up accordingly and displays a final average.

The grading algorithm will look something of this formula:
Total grade = [q1(number of correct answers)/q1(number of questions)]*(100/(number of assignments))+...

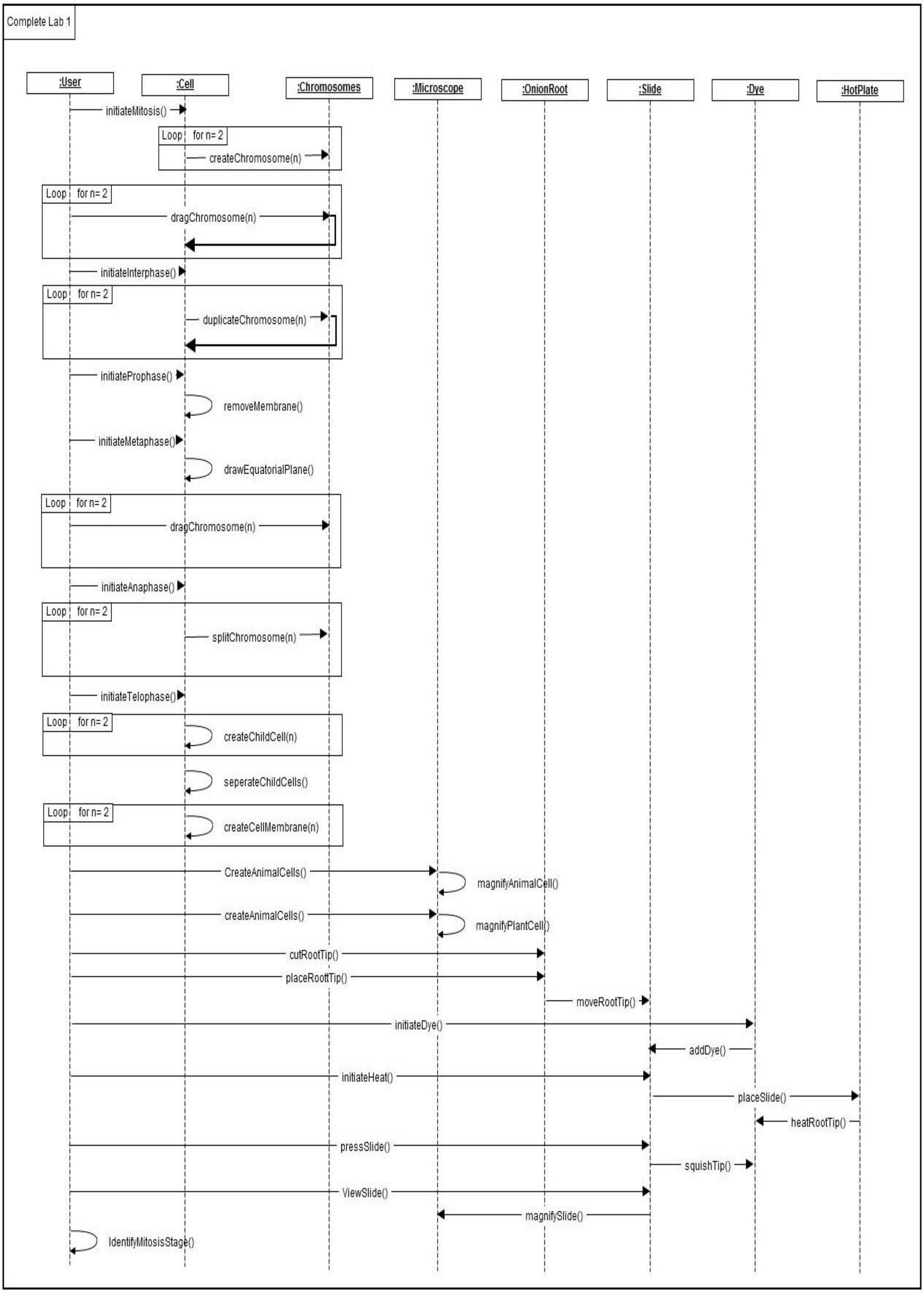
There are no other areas which require mathematical models in our project. Many of the graphical units that require mathematical models to generate will be done manually through our animations using Adobe Flash. Although it may not be perfect, it will simulate an environment appropriate enough for the lab's requirements. We can do this in our lab simulations because unless the values are correct, via comparison with known stored values, the lab will know how to handle the item. For example, the temperatures 200 and 250 will not require mathematical calculation to figure out how burnt and item is, simply to know that it is burnt.

8. Interaction Diagrams

8.1 Interaction Diagrams

8.1.1 Laboratories 1 and 4

On the previous page is the fragment interaction diagram for the first laboratory of the

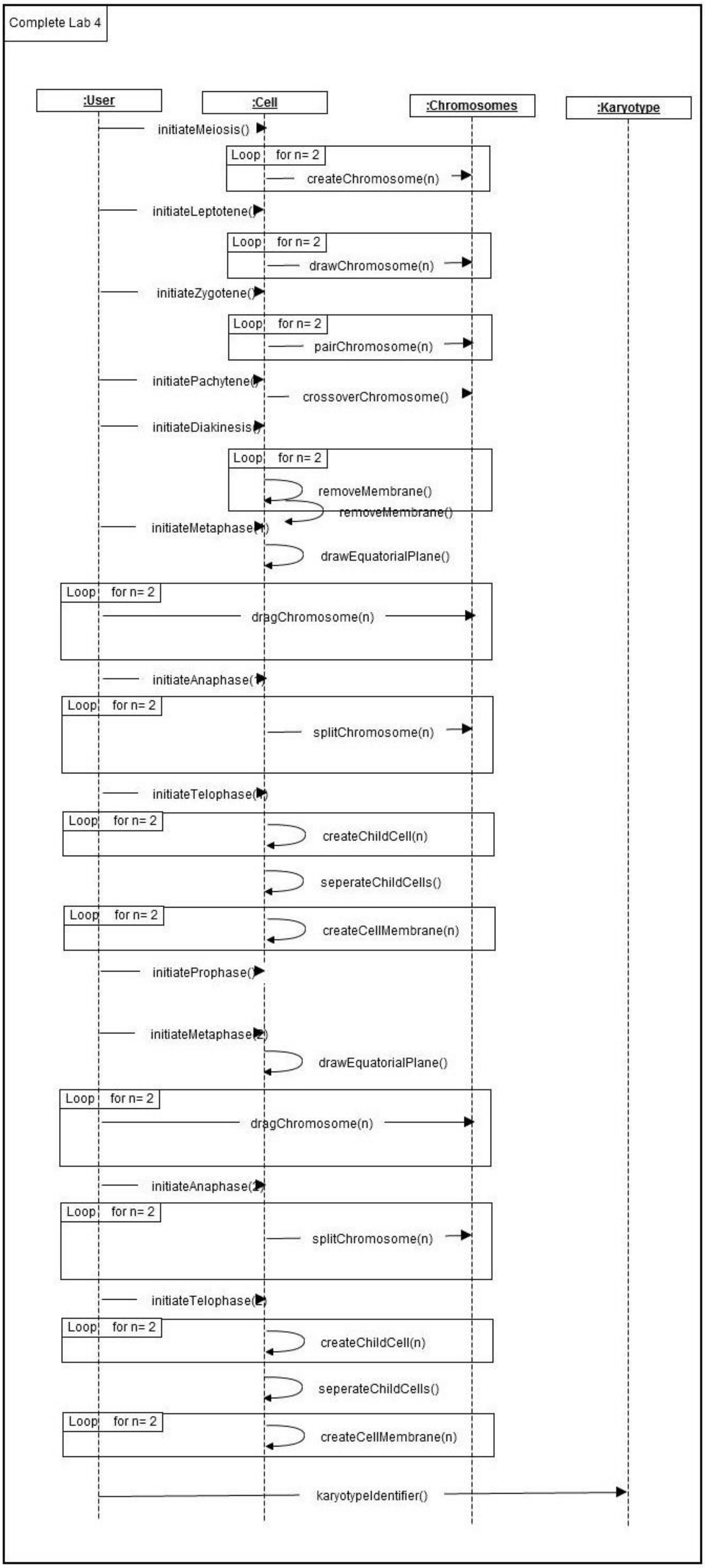


Virtual Biology Laboratory. In this laboratory the student will prepare a slide of an onion root and observe the process of cell division by mitosis. This diagram shows that first the system draws the chromosomes and then has the student drag them into the correct space. Then the user must initiate each stage of mitosis. Once prophase is initiated, the nuclear membrane disappears. Once metaphase is initiated, the equator is drawn across the cell. Each individual step is shown above. After this is finished, the interaction diagram shows that the student would then move onto creating the slide of the onion root.

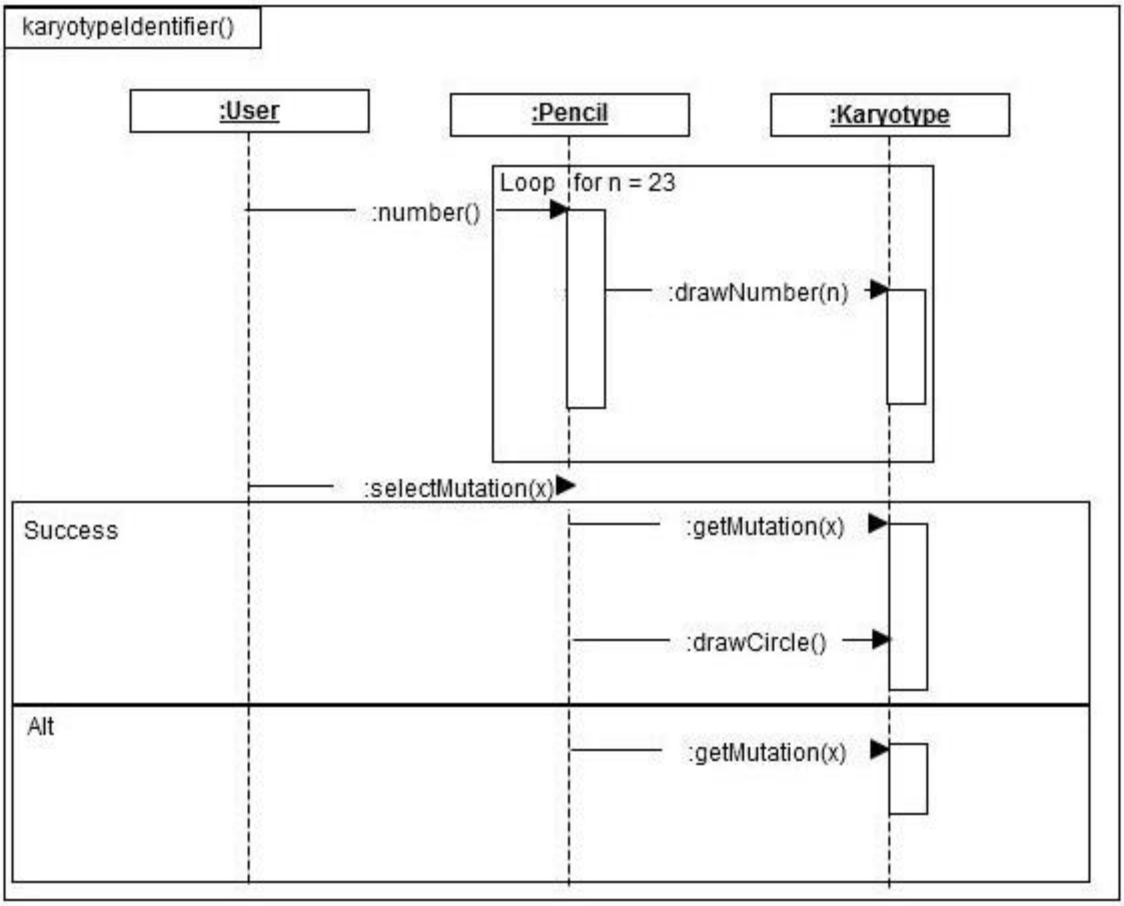
On the next page is the fragment interaction diagram for the fourth laboratory. The first part of this laboratory is to observe a different type of cell division called meiosis. The process of meiosis is similar to mitosis, but more of the steps are repeated in order to have the resulting cells have the correct amounts of genetic material. One of the differences between meiosis and mitosis, is that there are many more steps to prophase. Every interaction for the second part of lab four is shown in the karyotype interaction diagram fragment.

The design pattern used most by these two interaction diagrams are a task-oriented design. There is a huge focus placed on the user interaction with the system, this is shown by the user initiation and carrying out the transformation from one state of cell division to the next. Each goal of the user is broken down into a task, which is then broken down into a sub task. An example of this is how the user would initiate mitosis, and then the system would generate a cell with two chromosomes, and then the user would drag the two chromosomes to their correct spaces. The goal of the user is to complete the cell division known as mitosis, this is then divided into tasks: the user initiating which phase for the cell to enter, and then into sub tasks: the user moving the cell components into the correct spaces.

This seems like the optimal design for this part of the laboratory. There is still the use of objects with their own methods, but the way they interact is very dependant on the user’s input. This task-oriented design allows for the user to tell a certain component what to do, instead of the user telling a single component what to do and depending on that component to pass a message along. With this design there is tighter coupling since each component works together and knows almost the same things. This design is not entirely task-oriented because that would mean there was very little data being shared with the user, our system has more of a task oriented front end and a object oriented back end to allow for the user to complete frequent simple tasks, but have a comprehensive data center back end.



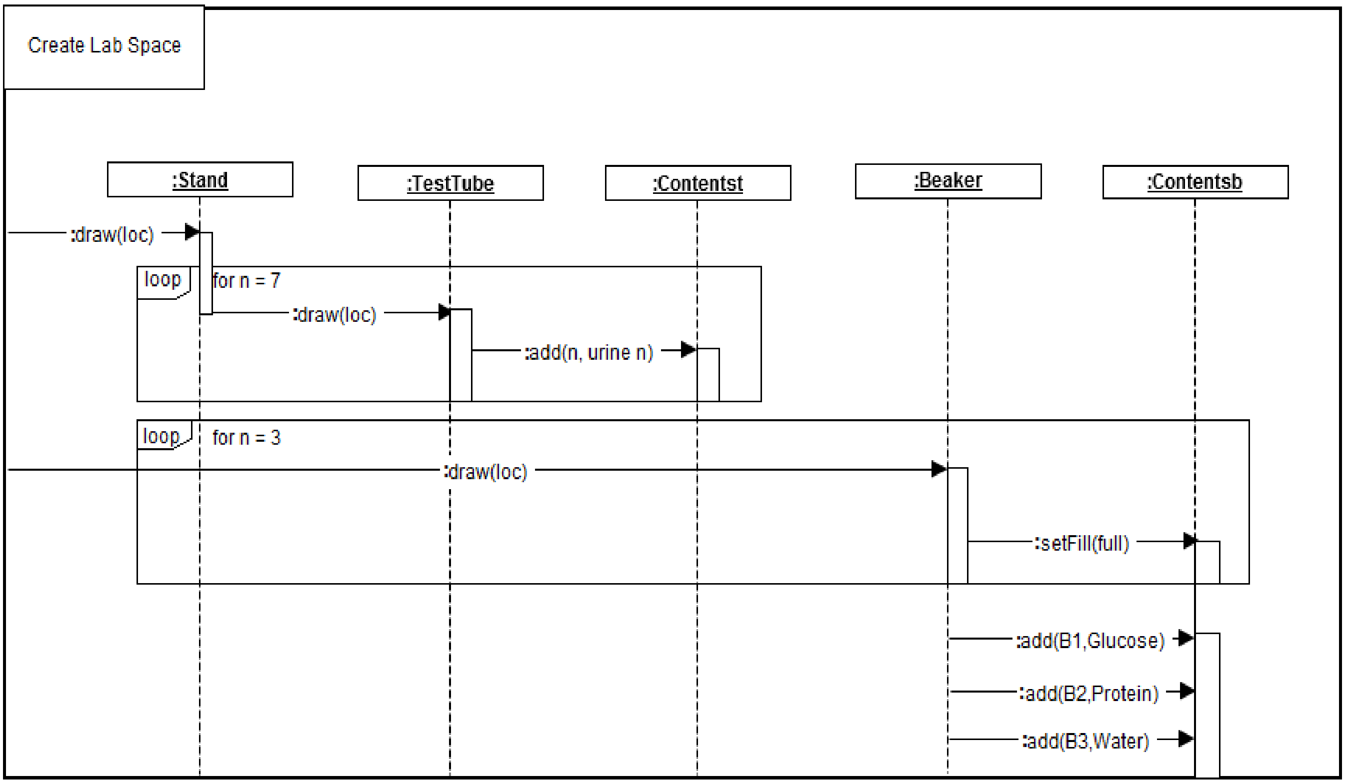
Complete Lab 4
:User
:Cell
:Chromosomes
:Karyotype
initiateMeiosis()
Loop for n= 2
createChromosome(n)
initiateLeptotene()
Loop for n= 2
drawChromosome(n)
initiateZygotene()
Loop for n= 2
pairChromosome(n)
initiatePachytene()
crossoverChromosome()
initiateDiakinesis()
Loop for n= 2
removeMembrane()
removeMembrane()
initiateMetaphase()
drawEquatorialPlane()
Loop for n= 2
dragChromosome(n)
initiateAnaphase()
Loop for n= 2
splitChromosome(n)
initiateTelophase()
Loop for n= 2
createChildCell(n)
seperateChildCells()
Loop for n= 2
createCellMembrane(n)
initiateProphase()
initiateMetaphase()
drawEquatorialPlane()
Loop for n= 2
dragChromosome(n)
initiateAnaphase()
Loop for n= 2
splitChromosome(n)
initiateTelophase()
Loop for n= 2
createChildCell(n)
seperateChildCells()
Loop for n= 2
createCellMembrane(n)
karyotypeIdentifier()



This interaction diagram fragment shows the last part of lab four. In this laboratory the student will be given a picture of a karyotype with human chromosomes. they will then number all of the chromosomes. After this has been shown, the student will have to select which chromosome has the mutation. If they select correctly the chromosome will appear circled. If they select incorrectly the system will return and error and ask them to select again.

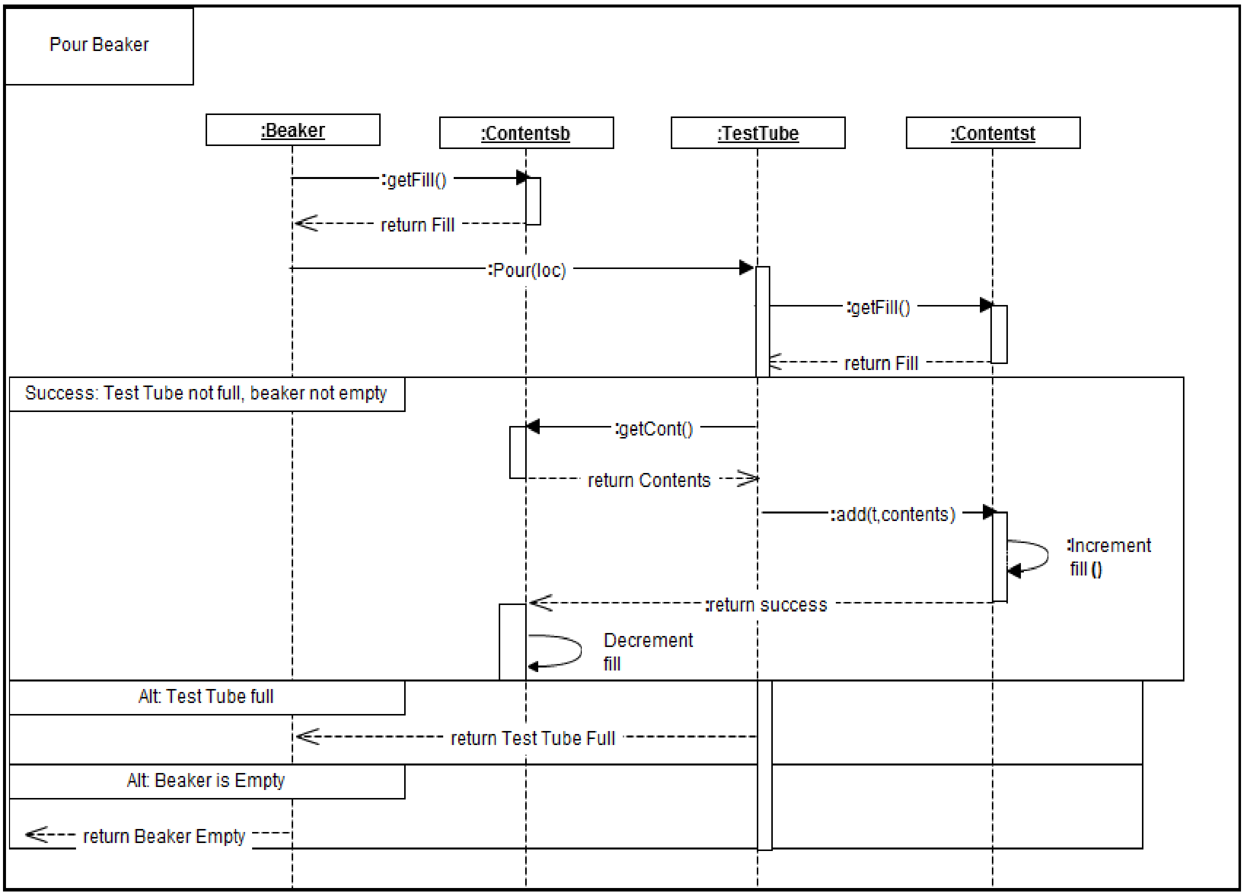
The design pattern shown in this interaction diagram is a publisher and subscriber design. The user tells the pencil object what he or she wishes to do and then that object passes the message on to the karyotype object. The user isn’t directly telling the karyotype what changes to make to itself, but instead it is using the pencil to speak indirectly to the karyotype object.

8.1.2 Laboratories 2 and 3

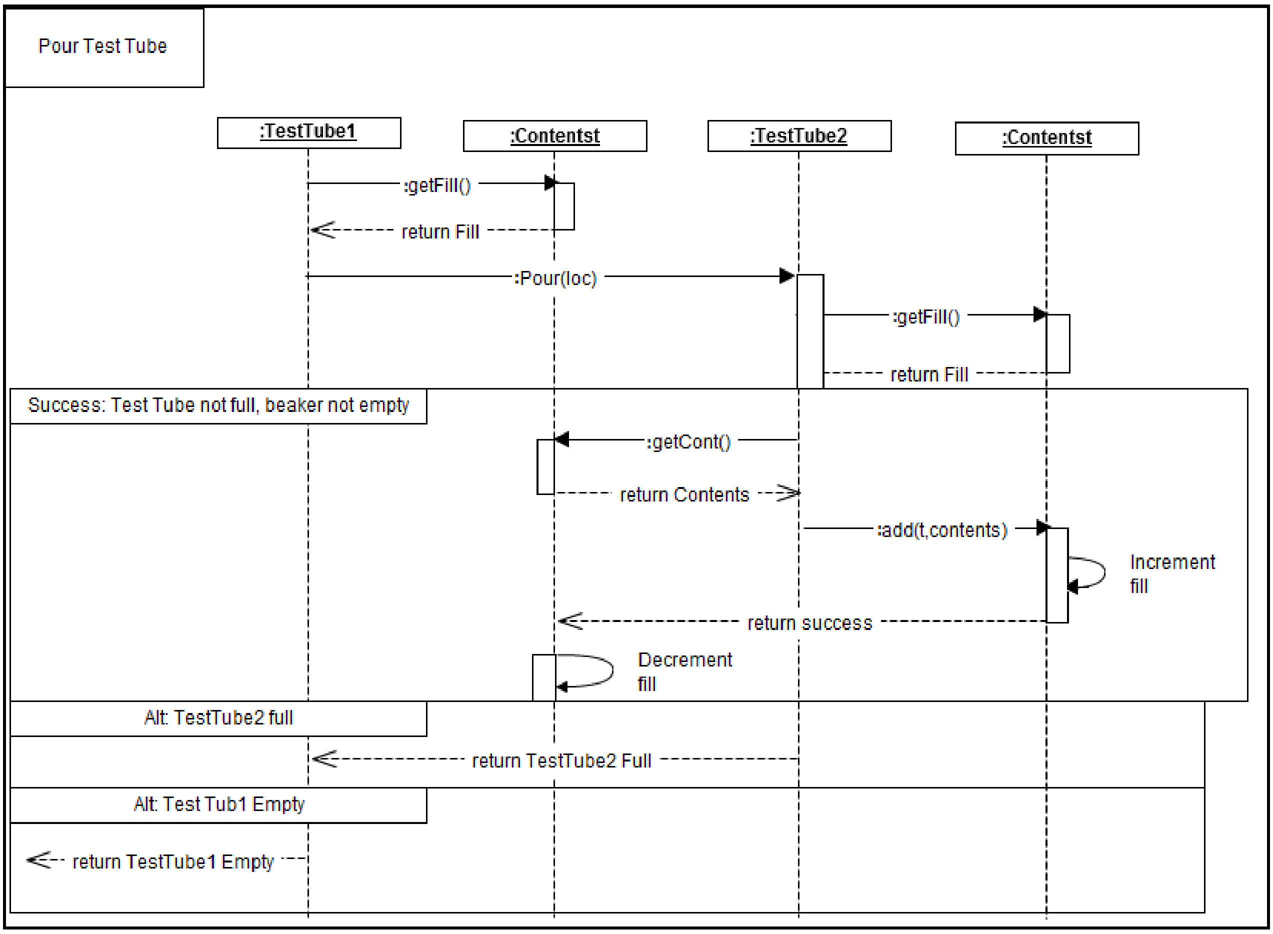


This is the interaction diagram fragment for creating the lab space for the second and third laboratory. In this part the system draws out the stand, test tubes and beakers, and then initializes the right test tubes with the correct liquids and initializes the beakers with the correct liquids.

The design pattern seen here is the decorator design pattern. The object “stand” doesn’t have much data held within it, just whether it is full or not. This object though acts as a decorator for the test tubes, it forwards on any initializing data that it needs to and the test tube forwards that to its contents. The same is true for the beaker. The user doesn’t interact directly with the contents of these objects, but relies on the stand, test tube and beakers to make the changes to the objects



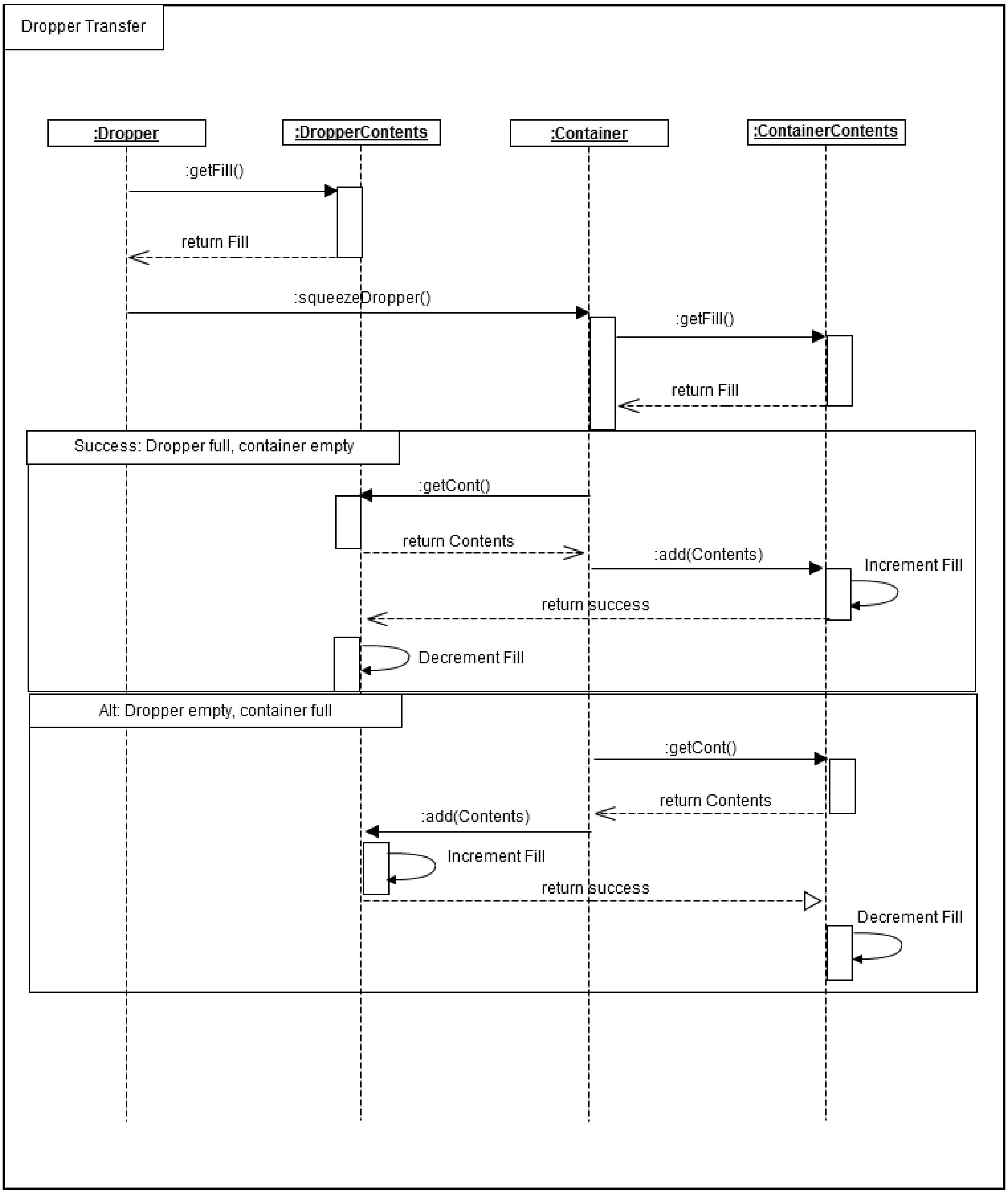
This interaction diagram fragment is the interactions of pouring a beaker into a test tube. It shows how the system check to see if either the beaker is empty or the test tube is full, and then how it transfers the contents from the beaker to the test tube, and then increments the fullness of the test tube and decrements the fullness of the beaker.



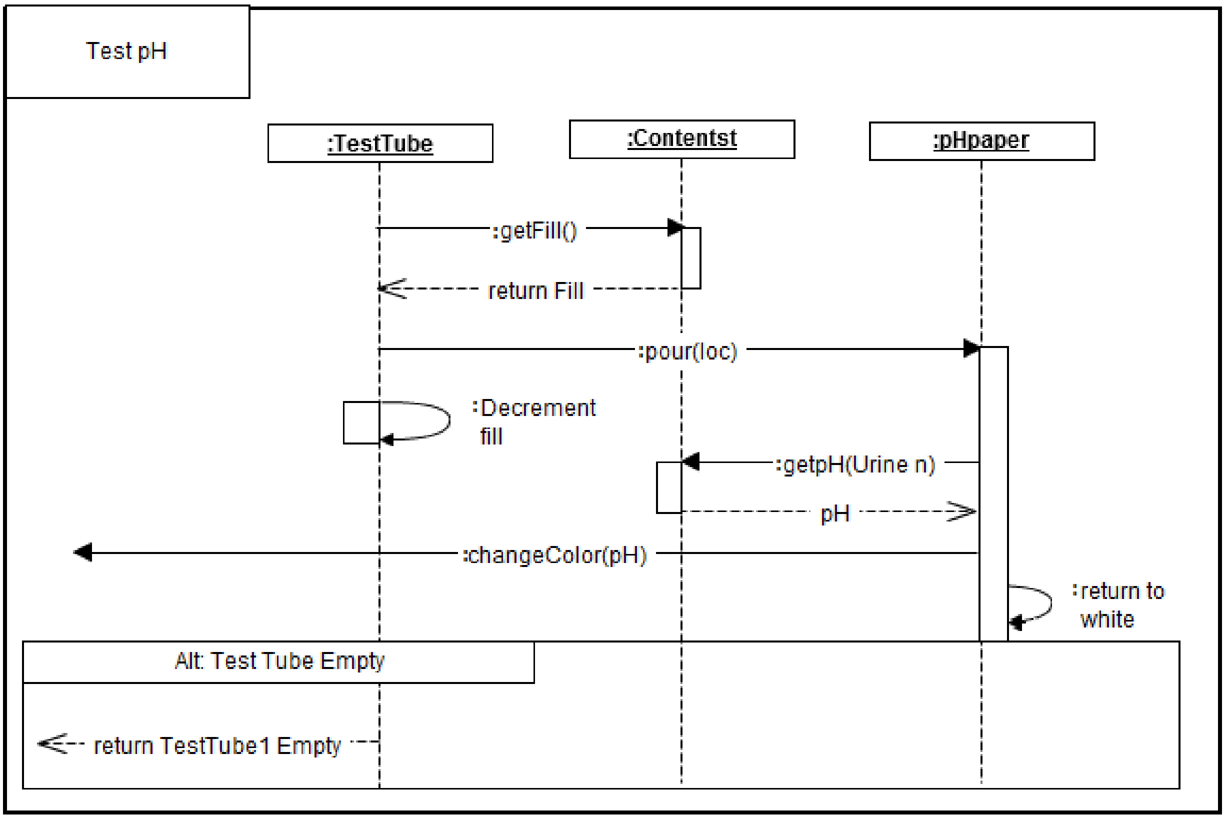
This is the interaction diagram fragment for pouring the test tube. In laboratory two, the test tube is poured onto the pH paper and in laboratory three, the test tube is poured into another test tube. this fragment shows how the system will check if either test tube is full or empty. Then once the contents of one test tube have been read by the other and added to its contents, the fullness of the second test tube is incremented and the fullness of the first test tube is decremented.

The previous two diagrams show the original design to pour a test tube or beaker into another test tube or beaker. In order to make the laboratories more fluid, we decided to change this design to use a dropper to move the liquid from one container to another. This works better for the implementation of our laboratory for two reasons. First off, we needed to use the dropper in different parts of the laboratory for distributing the dye solution and adding the urine to the pH paper, so the dropper needed to be created anyway. Secondly, the dropper will control exactly how much liquid the student is moving from one container to the other. This will allow the user to measure out the contents they are moving by how many drops they are putting in. Our original design did not leave any room for error, the system would just know to transfer the correct amount of liquid, now the student has to add the correct number of drops.

The new corrected interaction diagram, shown below, expresses the new dropper way of implementing the pour use case. The design pattern used in this is the state pattern. The two states of the dropper, test tube or beaker, are empty and full. If the dropper is empty than the user can click on a full test tube or beaker and change the state of the dropper to full. If The user uses the empty dropper to click on an empty test tube or beaker than nothing would happen. Once the dropper is in the full state, the student can use it to click on where they wish to distribute the contents.

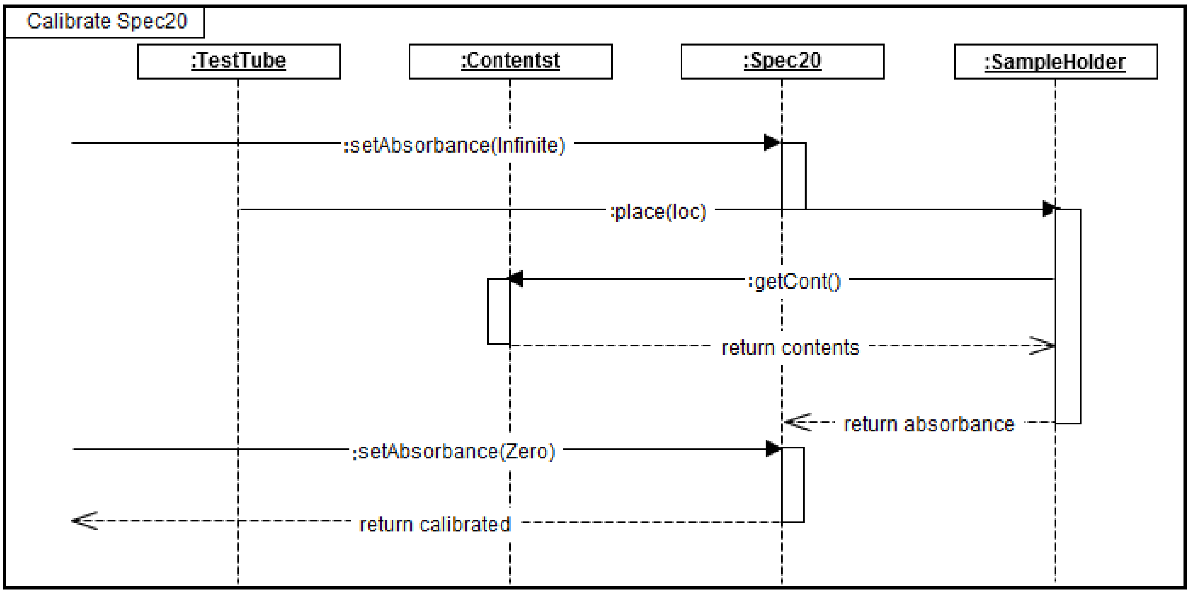


create and share your own diagrams at gliffy.com

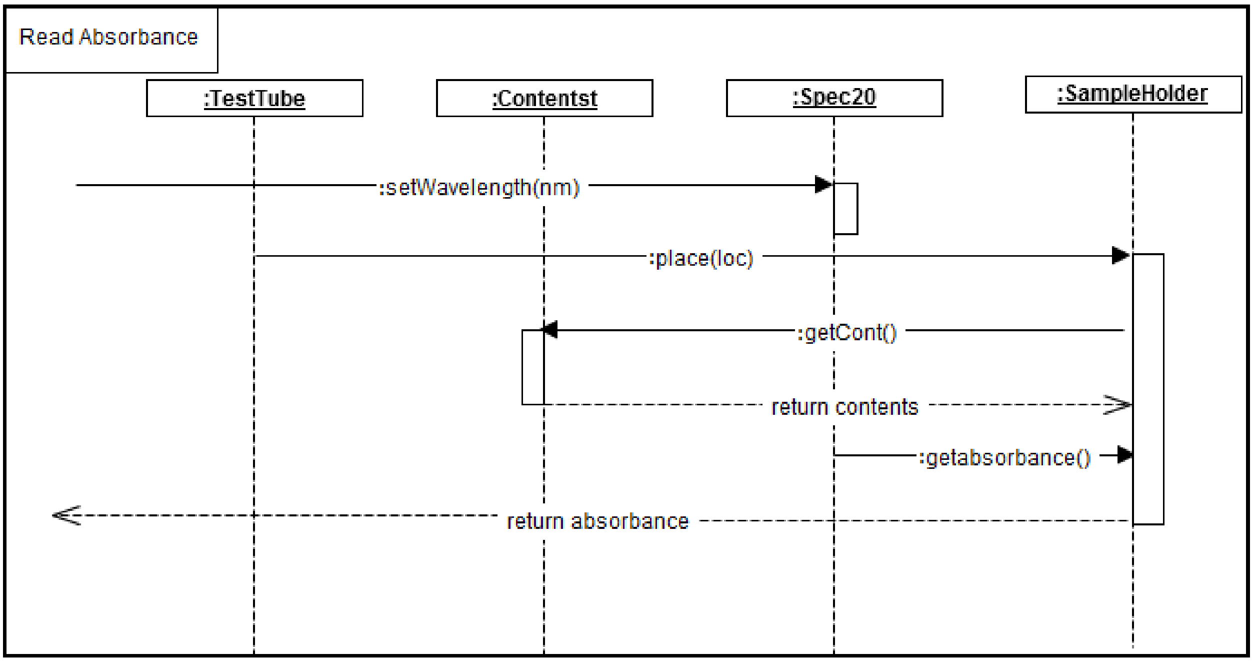


This interaction diagram fragment shows the interactions involved with testing the pH of the contents of a test tube. It hows how the test tube's contents are poured into the pH paper and then the pH paper accesses the number corresponding to the pH from the contents of the test tube. The pH paper then uses that number to change into the color corresponding to the pH number it received. The pH paper then returns to white.

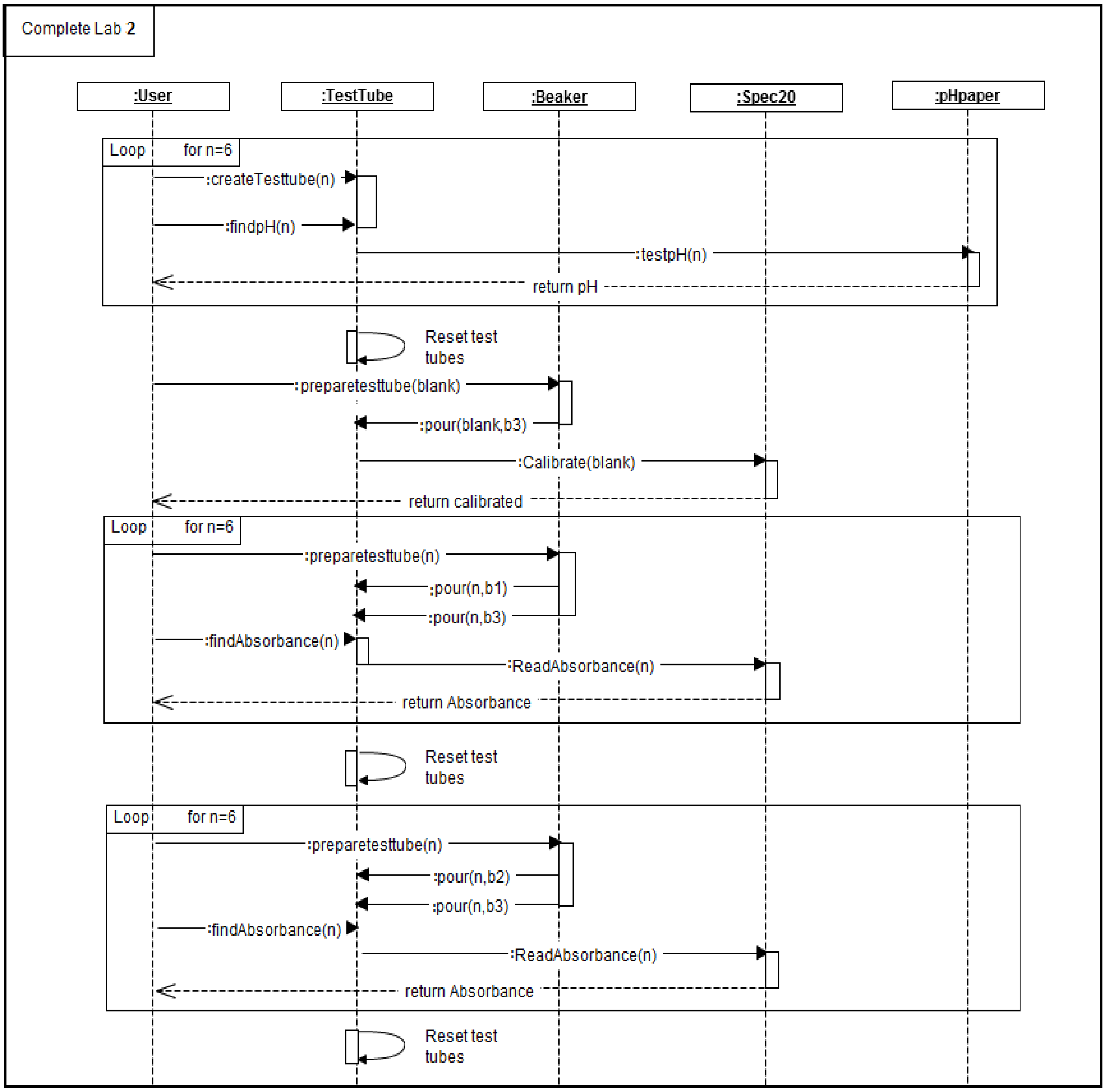
The design decisions of this part of the virtual Biology Laboratory, and the following two parts (calibrate Spec20 and Read Absorbency) was made to keep a balanced workload between all of the objects. There could be a problem with some of these parts since there is a relatively long chain of communication between the objects, but this needs to be done this way in order to change the outcome of the lab, or the re-usability of the code. In the future we could change the code so that each students test tubes would have different pH balances or different absorbency and the way the system is designed now, we would only have to make a few small changes in order to change the whole outcome of the entire experiment.



This interaction diagram fragment shows the interactions involved with calibrating the spec20 spectrophotometer. It involved the user setting the infinite absorbance, and placing the bank test tube within the sample holder and checking the absorbance. Then the user must set the absorbance to zero and run it again, the machine then would return that it is calibrated.

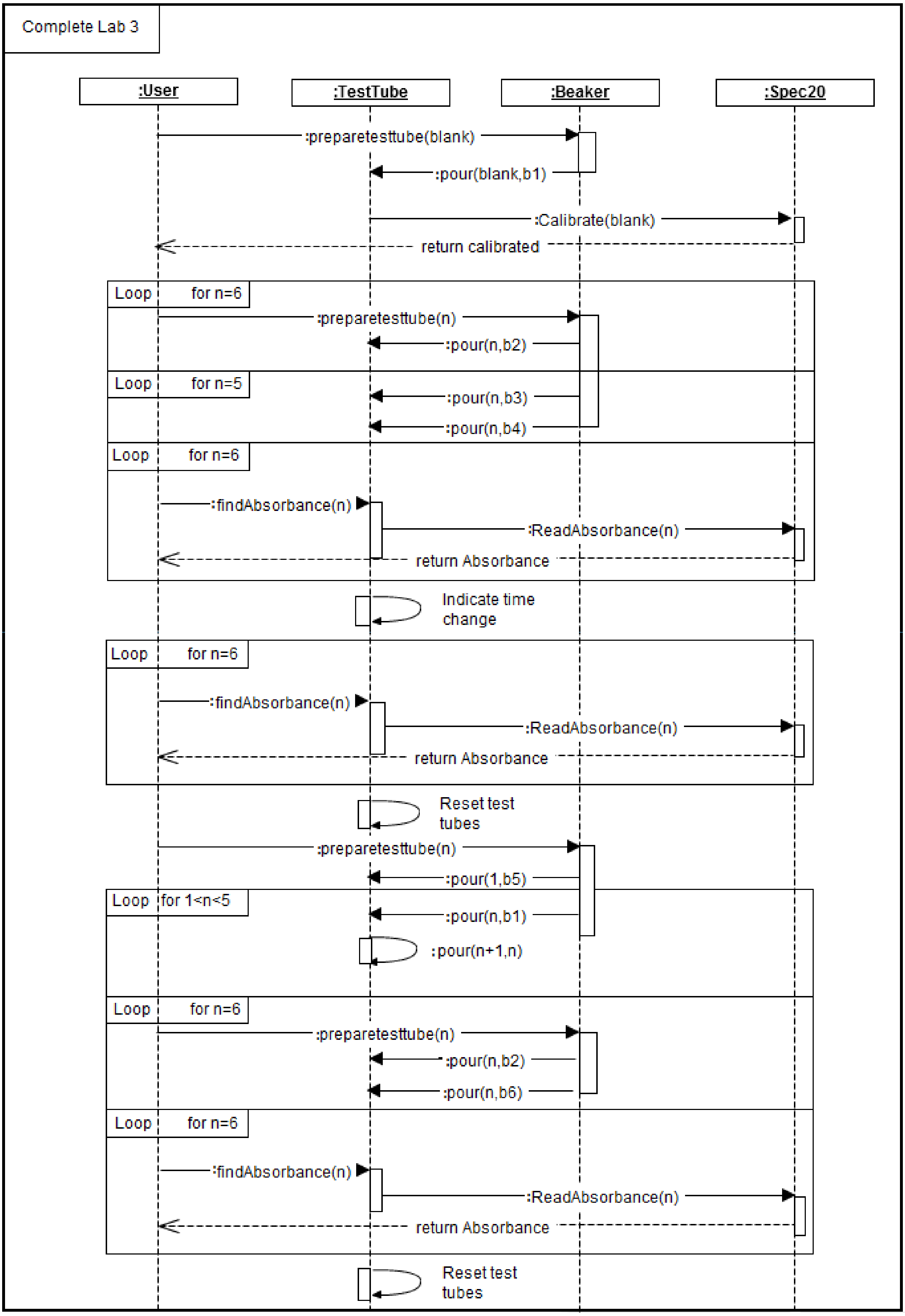


This is the interaction diagram fragment that shows how the user uses the system to read the absorbance of a test tube. If the Spec20 is calibrated, then the user can place a test tube within its sample holder. The spec20 would then get the contents of that test tube and look for its absorbance and then return that number that it found.



[REF: Create Lab Space, Pour Beaker, Test pH, Calibrate Spec20 and find Absorbance]

This is the full interaction diagram for the second laboratory: biological molecules. The user is adding to the test tubes, checking their pH and then checking for their absorbance. This interaction diagram uses the fragments shown above like creating the lab space, pouring the beaker, testing pH, calibrating the Spec20 and then finding the absorbance. If the user tried to do any of these steps in the wrong order, the system would return an error saying that the previous step had not been completed correctly.



[REF: Create Lab Space, Pour Beaker, Pour Test Tube Test pH, Calibrate Spec20, find Absorbance.]

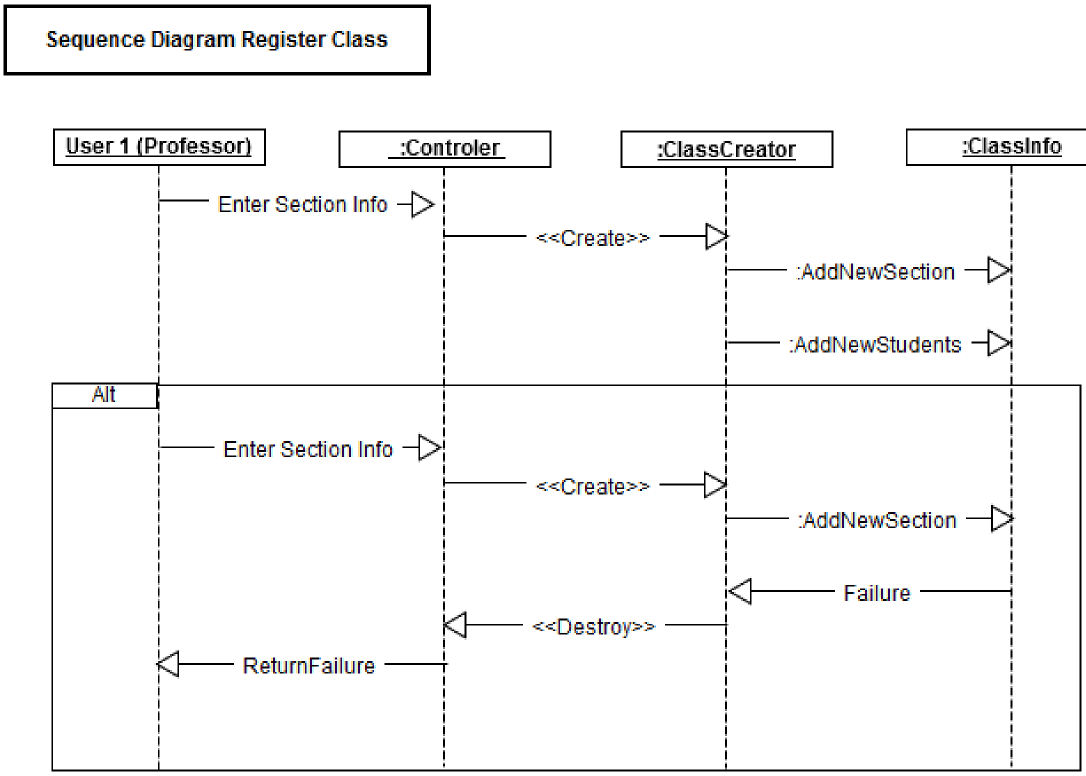
The interaction diagram on the previous page is for completing the third laboratory: Enzyme activity. It also uses the interaction diagram fragments as laboratory two, but there are some changes made in what the contents of the beakers and test tubes are. Just like in that laboratory though, the interactions between test tubes, beakers and the spec20 are the same.

The previous two interaction diagrams, for completing laboratories two and three, also show the task oriented front end and a object oriented back end. The user is going to move the test tubes around to interact with different objects in order to complete the tasks that he or she wishes to complete. The test tubes, beakers, pH paper and the spec20 all need o share data with each other in order to respond to the students commands. This is a good design because the system will always react to the command the students gives it in the same way, but the data shared between them can be easily changed without affecting those user/system interactions.

8.2 General Interaction diagrams

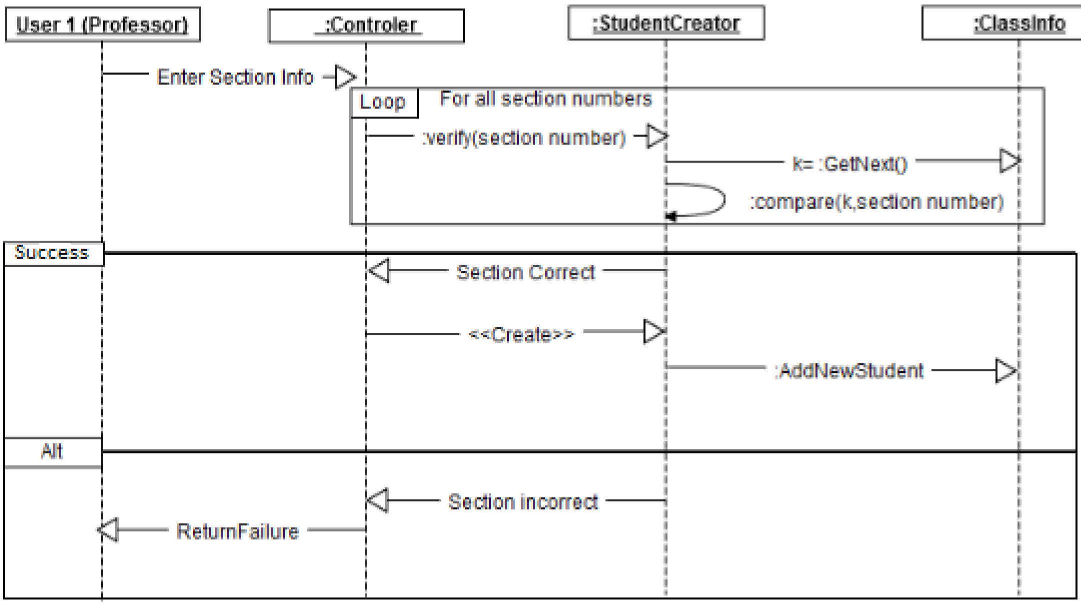
The interaction diagram displayed on the next few pages show the flow between the UI elements that are visible in the system. There are several grouped, main functions. These include, the users, the homepage, the main page and the navigation menu. Each of these elements have sub-elements in which are distinguished differently between the students and the professors. These are only a few of the back-end commands that our system will accomplish.

The design pattens expressed in the following four diagrams is an object-oriented design. A student is a member of a class so the information held in the class if very similar but more general than the information held about a student in that object. It also shows the command design principle. the controller acts as the invoker so each command the user puts in, the controller assigns the work to the correct object.

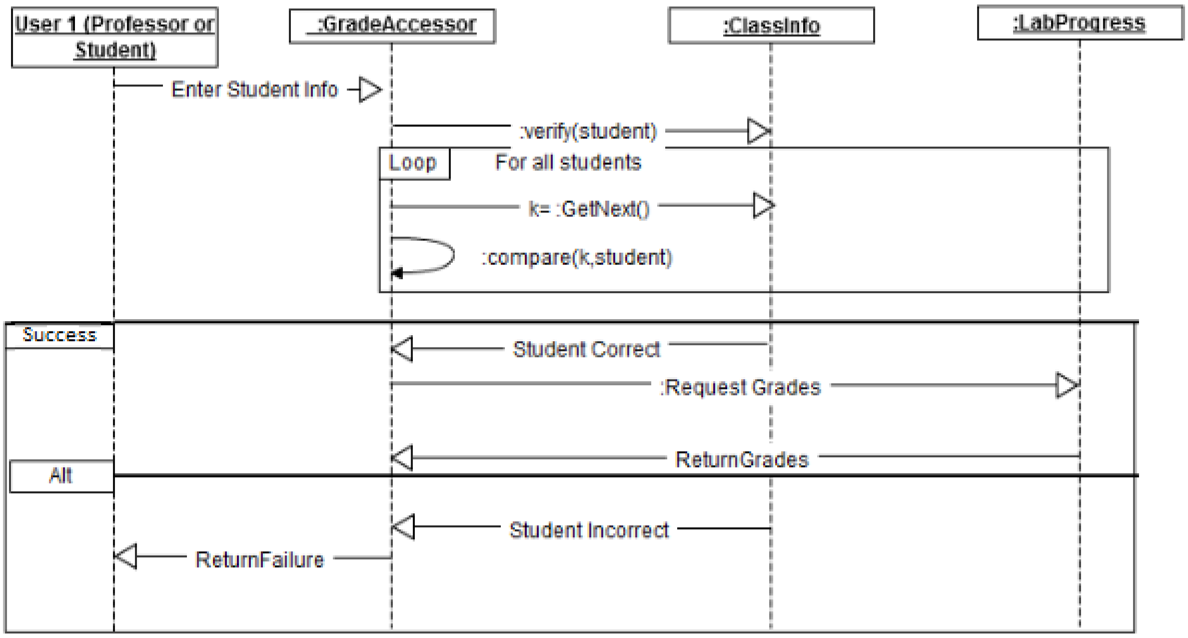


This diagram shows the reaction the system has to the use case RegisterClass.

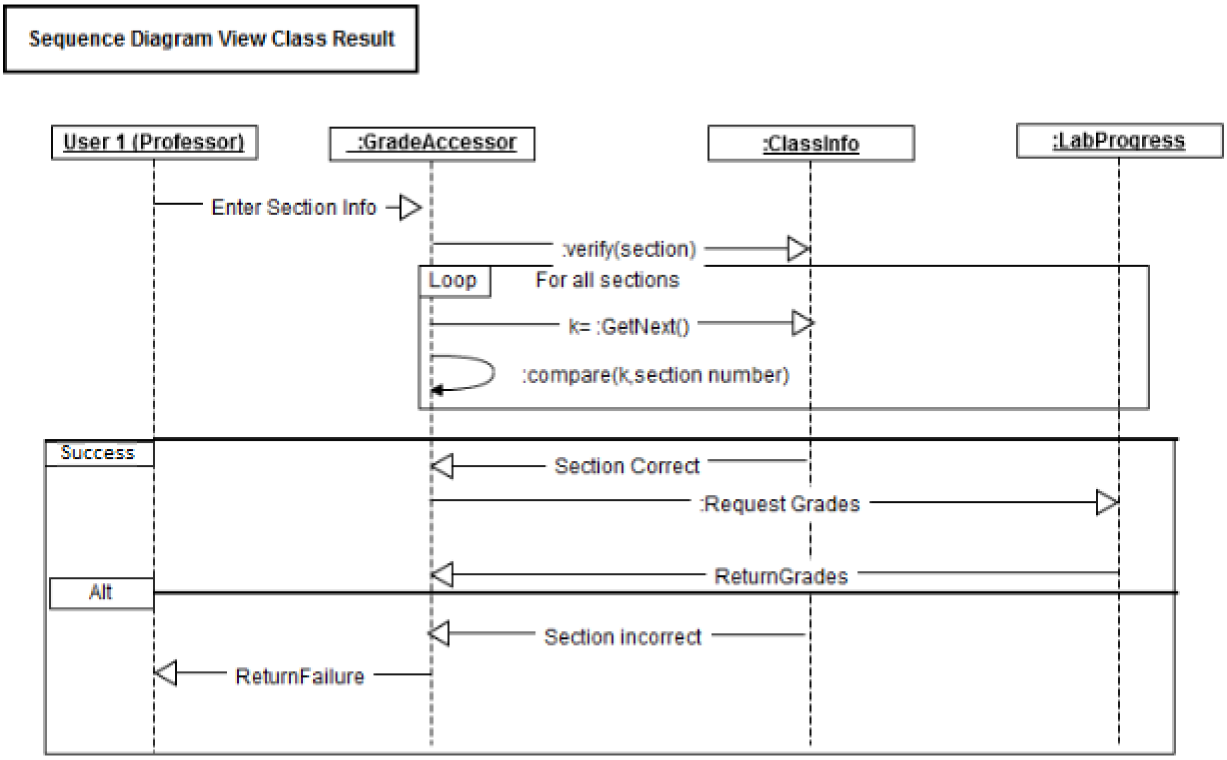
Sequence Diagram Register Student



Sequence Diagram View Student Result



This diagram shows the reaction the system has to the use case ViewSudentResult



This diagram shows the reaction the system has to the use case ViewClassResult

9. Class Diagram and Interface Specification

9.1 Class Diagram

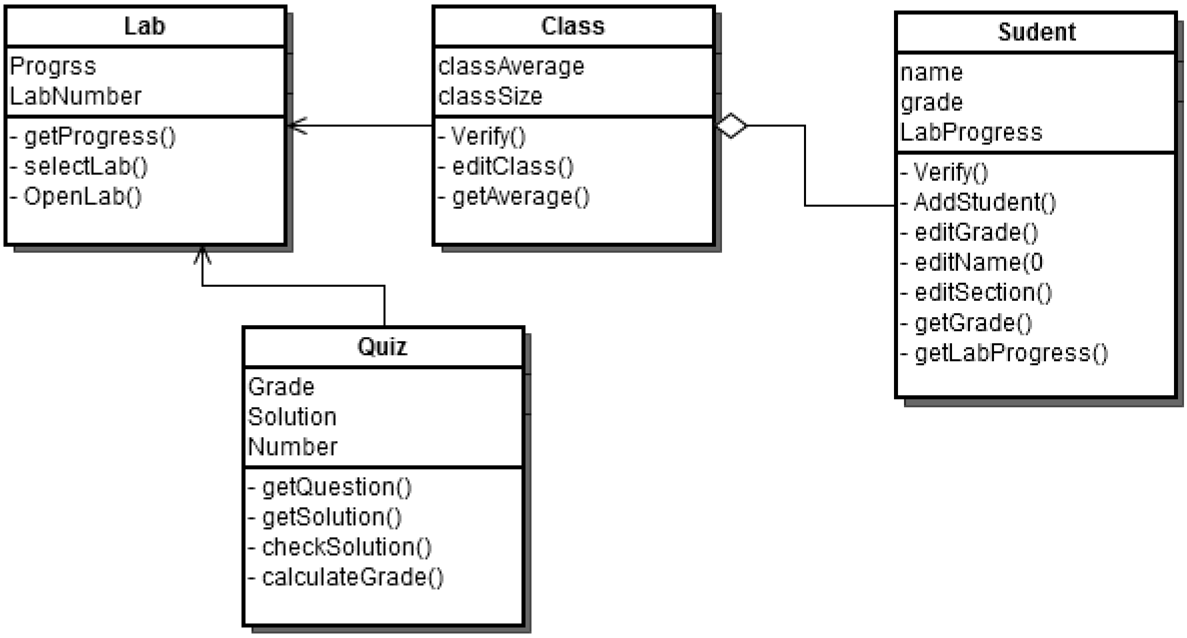


Fig. 9.1 Class Diagram for General Concepts

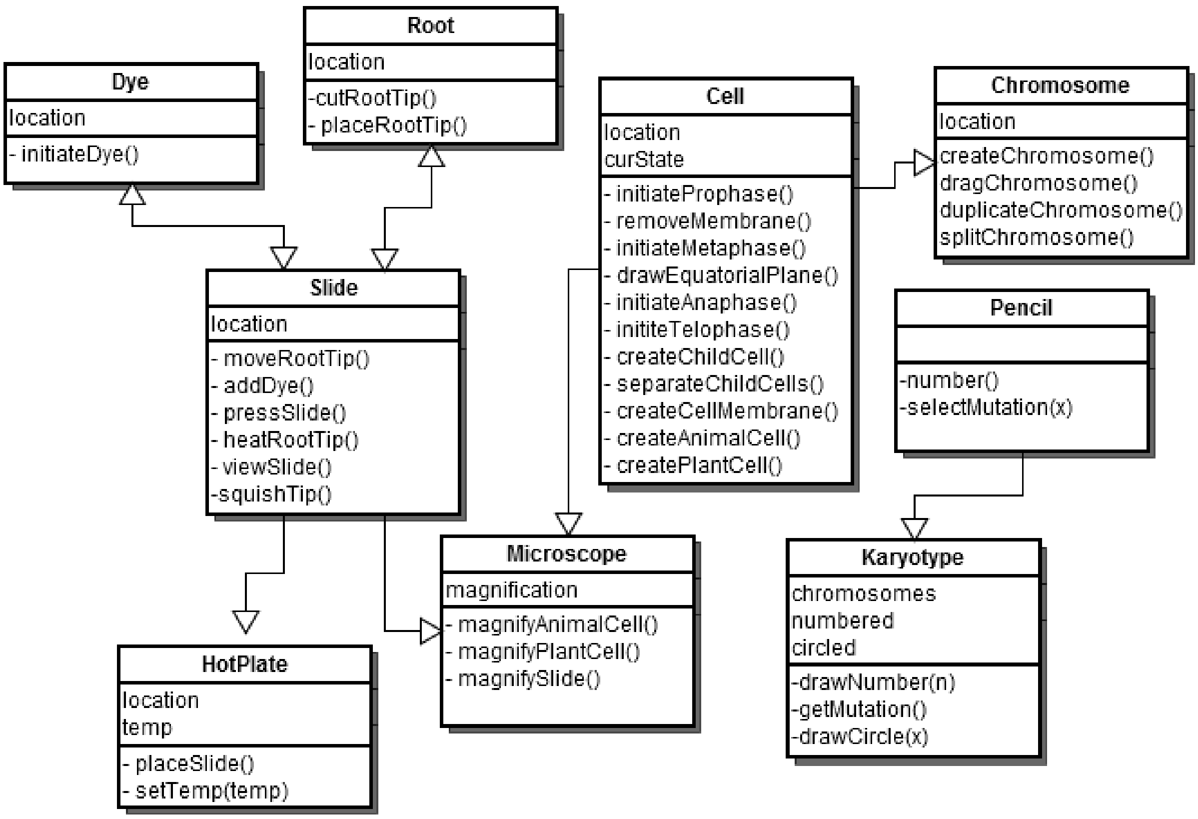


Fig. 9.2 Class Diagram for Laboratories One and Four, Cell Division and Meiosis

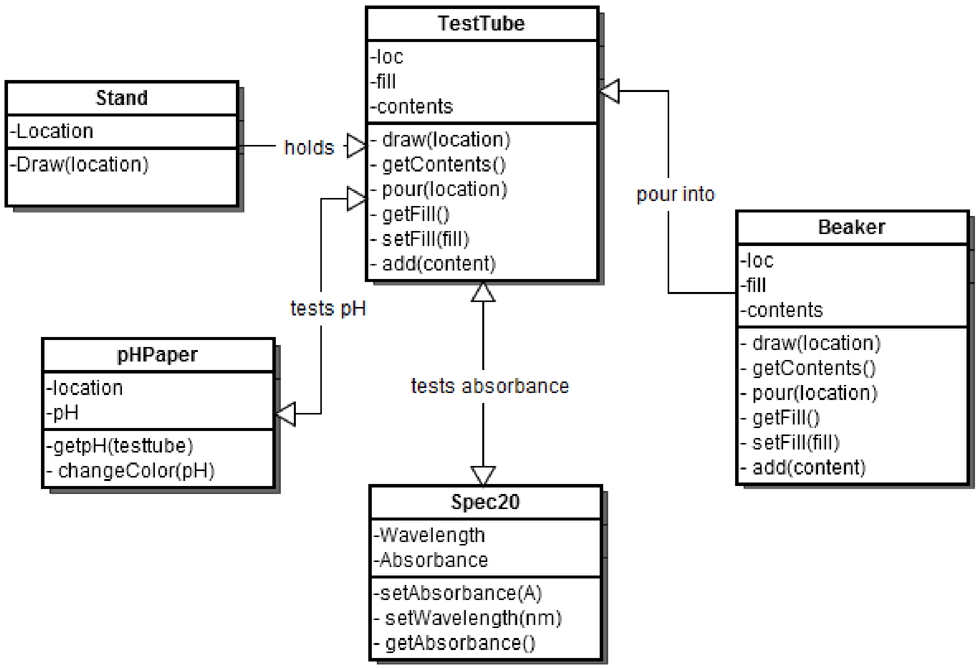


Fig. 9.3 Class Diagram for Laboratories Two and Three, Biological Molecules and Enzyme Activity

The class diagrams for the Virtual Biology Lab express two different main design principles. The first diagram shows a data-centric approach and the second two diagrams show a more task-oriented design. This is discussed in more detail in section 9.4. **Fig 9.1** shows the general class diagram for the Virtual Biology Laboratory. This is the class structure and relations for the student, class, quiz and Lab opener. Connected to the Lab class is **Fig 9.2** which is the partial Class diagram for laboratories one and two. This diagram represented the classes that are used when a student selected to complete the first or fourth laboratory. **Fig 9.2** is the partial class diagram for the remaining laboratories. This diagram shows the classes and relationships between them for when a student selects to complete the second or third laboratory. The controller connects all of these class diagrams together.

9.2 Data Types and Operation Signatures

	General Data Types and Operation Signatures
C1 Lab	• Progress- indicates how far along the lab a student is, keeps track of how many check points they have reached • labNumber- indicates which number lab the student has selected • getProgress() - returns how many checkpoints have been reached • selectLab(labNumber) - allows the user to select which lab they would like to complete • openLab(labNumber)- creates all of the objects needed for the user to complete the laboratory that he selected
C2 Student	6. name - the name of the particular student 7. grade - the grade that student has 8. labProgress - indicates how many laboratories the student has completed 9. verify(name) - checks to see if a particular student exists 10. AddStudent(name) - adds all of the information for a new student 11. editGrade(grade) - edits the grade the student holds 12. editName(oldName, name)- edits the name of the student 13. editSection(name, section)- moves the student into another section 14. getGrade(name)- returns the grade that particular student has 15. getLabProgress()- returns the progress that student has made
C3 Quiz	8. grade- keeps tally of the correct answers the student has given 9. number- keeps track of total number of questions asked 10. solution- keeps track of the solution to that particular question 11. getQuestion()- randomly generates a question 12. getSolution(x)- gets the solution to that question 13. checkSolution(y,x)- compares the students answer to the real solution 14. calculateGrade(grade, number)- calculates the students grade based on the number of questions correct out of all the questions asked
C4 Class	15. classAverage- indicates what the average grade is in that section 16. classSize- indicates how many students belong to that class 17. verify()- verifies that that section exists 18. editClass() - edits number of students in a class 19. getAverage() - takes the average of every students grade within the specific class

	Laboratories 1 and 4, Cell Division and Meiosis
C5 Cell	22. Location, integer 23. State, represents which state of mitosis the cell is in 24. initiateProphase() begins the process of prohpase 25. removeMembrane() takes away the image of the circle around the nuclear material 26. initiateMetaphase() begins the process of metaphase 27. drawEquatorialPlane() draws a line at the point of the equatorial plane in the cell 28. initiateAnaphase() begins the process of anaphase 29. inititeTelophase() begins the process of telophase 30. createChildCell() creates a second cell that looks the same as the original 31. separateChildCells() moves the child cell from the original 32. createCellMembrane() redraws the cell membrane arounf the original

	and the child cell 33. createAnimalCell() creates a representation of an animal cell 34. createPlantCell() creates a representation of a plant cell
C6 Chromosome	35. createChromosome() draws out a single chromosome within the cell 36. dragChromosome() allows the user to drage the chromosome to a new location 37. duplicateChromosome() draws a second chromosome that is the same as the first 38. splitChromosome() splits an existing chromosome in half
C7 Pencil	39. number() numbers each of the chromosomes on the karyotype 40. selectMutation(x) user selects which chromosome has a mutation in it
C8 Karyotype	41. numbered, integer keeps track of how many chromosomes are numbered 42. circled, integer, keeps track of how many chromosomes are circled 43. drawNumber(n) draws the number under the chromosome 44. getMutation() finds if the user selected the correct location for the mutation 45. drawCircle(x) draws circle around the chromosome with the mutation
C9 Microscope	46. Magnification, integer, represents how zoomed in the microscope is 47. magnifyAnimalCell(), magnifies the animal cell 48. magnifyPlantCell(), magnifies the plant cell 49. magnifySlide() magnifies whatever is in the slide
C10 Hot Plate	50. Temp, integer, sets the temperature of the hot plate 51. placeSlide(), moves the slide to the hot plate 52. setTemp(temp) sets the temperature of the hot plate
C11 Slide	12. Location, integer, represents the loation of the slide 13. moveRootTip() moves the onion root tip to the slide 14. addDye() adds dye to the contents of the slide 15. pressSlide() presses the slip cover onto the slide 16. heatRootTip() heats up the root tip when contained in the slide and placed on the hot plate 17. viewSlide() views the contents of the slide when it is placed on the microscope 18. squishTip() crushes the contents of the slide in order to have the slide cover lay flat.
C12 Dye	● initiateDye() initiates the change in color of the object that the user as added the dye to
C13 Root	● location, integer, shows the location of the root ● cutRootTip() removes one part of the root for the user to move ● placeRootTip() allows user to move the root from one space to another

	Laboratories 2 and 3, Biological Molecules and Enzyme Activity
C14 Test Tube	• location (integer) • contents, • fill (integer) • draw(location) draws test tube in a specified location • getContents() returns what is contained in the test tube • pour(location) adds the contents of the test tube to the location indicated, either pH paper or beaker • getFill() returns if the test tube is empty or full • setFill() sets the amount of the test tube that is full, incremented whenever something is poured in or out • add(content) adds a content to the list of what is in the test tube
C15 Beaker	• location (integer) • contents, • fill (integer) • draw(location) draws beaker in a specified location • getContents() returns what is contained in the beaker • pour(location) adds the contents of the beaker to the location indicated, into test tube • getFill() returns if the beaker is empty or full • setFill() sets the amount of the beaker that is full, decremented whenever something is poured out • add(content) adds a content to the list of what is in the test tube
C16 pH	• location (integer) • pH (integer 0 through 7) • getpH(testtube) returns the pH number to the contents of what is in the test tube • changeColor(pH) changed the colopHPaperr of the pH paper to show whichever level pH it is set to. Returns to white when set to 0
C17 Spec20	• Absorbance (integer) • Wavelength (integer) • setAbsorbace(A) sets the light and control settings for the spec20, set to 0 for the zero calibration and 1 for the infinite calibration • setWavelength(nm) set the wavelength of the wave that the spec20 is shining through the test tube in order to find its absorbance, in nanometers • getAbsorbance() returns the absorbance of the contents of the test tube being tested
C18 Stand	• Location (integer) • Draw(location) draws out the stand and prompts the drawing of 7 test tubes

9.3 Traceability

	C1	C2	C3	C4	C5	C6	C7	C8	C9	C 10	C 11	C 12	C 13	C 14	C 15	C 16	C 17	C 18
D1	X																	
D2	X																	
D3		X		X														
D4	X																	
D5		X		X														
D6		X		X														
D7		X		X														
D8			X															
D9			X															
D10				X														
D11		X																
D12														X			X	
D13					X	X			X		X							
D14								X	X	X					X			X
D15						X	X	X										
D16					X						X	X	X					
D17					X				X		X	X	X					
D18										X	X			X				

D19														X			X	X
D20														X		X		
D21						X	X	X										
D22						X	X	X										

All of the names of the domain concepts are very similar to their corresponding class concepts. D1 through D11 deal with the general aspects of the Virtual Biology Laboratory, these are things like the controller, student editor or class editor. These domain concepts correlate most closely with the class concepts C1 through C4.

The other domain concepts D12 through D22 relate mostly to the biology concepts in the virtual biology laboratory. This includes the microscope, spec20, test tube or beakers, all of which directly correlate to the class concepts ranging from C6 to C18. Within the biology concepts there are three major types of ideas, a machine, a holder and an item. The machines are the microscope and spec 20, the holders are the test tubes and beakers and the items are the cells and chromosomes. These concepts are expressed in both the domain and class concepts and in this matrix occupy the bottom right square of the matrix from D12 to D22 and C6 to C18.

9.4 Design Patterns

A major design pattern for the Virtual Biology Laboratory is the Undo/Redo principle. Since there are many options for the user to take that could ultimately be incorrect it makes sense that our system should take the user’s input, check to see if it is correct and then reset the system and return an error if the step taken was incorrect. An example of this is when the user needs to pour an item. If the user tries to pour a test tube into a beaker, which is not allowed in our system, the system needs to redraw the test tube in its original space, redraw the beaker without anything poured into it and return an error message to the user. This principle is shown mostly in the alternate flow of events in the sequence diagrams where it outlines what user inputs would return an error if done incorrectly.To account for this design principle we would add an “unexecute()” function within any of the classes that would ever need to be reversed and label the alternate case an option for when “reversible == true” where reversible would be equaled to true if the system checked and found the users input impossible.

The biggest design pattern change that was made to our design was to the pour test tube or beaker concept. Originally we had a system where the user would drag test tube 1 to test tube 2 to move some of the contents of test tube 1 to test tube 2. In order to make our system more fluid we decided to change this to the user using a dropper to move the contents from test tube 1 to test tube 2. This new design uses the state design pattern since the dropper will either be in the empty or full state. This is the optimal design for this use case because it allows for the user to make an error, to make the laboratories a little bit more challenging, and also makes the design more modular, where one could easily add more contents or types of containers without having to change any of the properties of the previous containers.

The class diagrams above express two different main design patterns. The first diagram shows a data-centric approach and the second two diagrams show a more task-oriented design. The benefits to having this design is how modular our virtual biology laboratory is because of it We can easily add new components or interactions between the user and the system without changing the parts of the program that keep track of the data. Similarly, we could change the data recorded by the system without affecting how the interactions between the system and the user are done. It would be very simple for someone to pick and choose which labs they wanted to include without making any major changes to the system.

9.5 Object Constraint Language Contracts

```
context Class inv:
    self.classAverage  >=0 and self.classAverage <=100


context Quiz inv:
    self.Grade >=0 and self.Grade <=100

context Student inv:
    self.Grade >=0 and self.Grade <=100

context pHPaper inv:
    self.pH >= 0 and self.pH <= 14


context pHPaper inv:
    let pH : Integer = self.pH->getpH(pH) in
    if isBase then
        pH >= 7
    else
        pH < 7
    endif

context Microscope inv:
    self.magnification >= 1 and self.magnification <= 100

context Spec20 inv:
    self.Wavelength >= 300 and self.Wavelength <= 800

context Hotplate inv:
    self.temp >= 22  and self.temp <= 250

context HotPlate inv:
        let temp : Integer = self.temp->set(temp) in
        if isBurnt then
```

```
        temp > 200
else
        temp <= 200
endif
```

Hypothetical method contained in each lab to check user input. If user input is wrong, halt lab and wait for them to restart the lab.

```
context Lab:: checkAnswer() post:
    self.answers = false (
        (haltLab = true)
    )
```

On a whole, our system's design does not have many unique OCL contracts. The majority of these restrictions are on the integer values of variables such as grades, temperature, magnification, and pH. Our system is based on user interaction, and does not have many complex calculations to check, rather compare the user input values to the proper values stored in the system. Above are some of the key contracts defining ranges for grades and how the system determines the values of certain attributes such as **isBurnt** and the color of the pH paper.

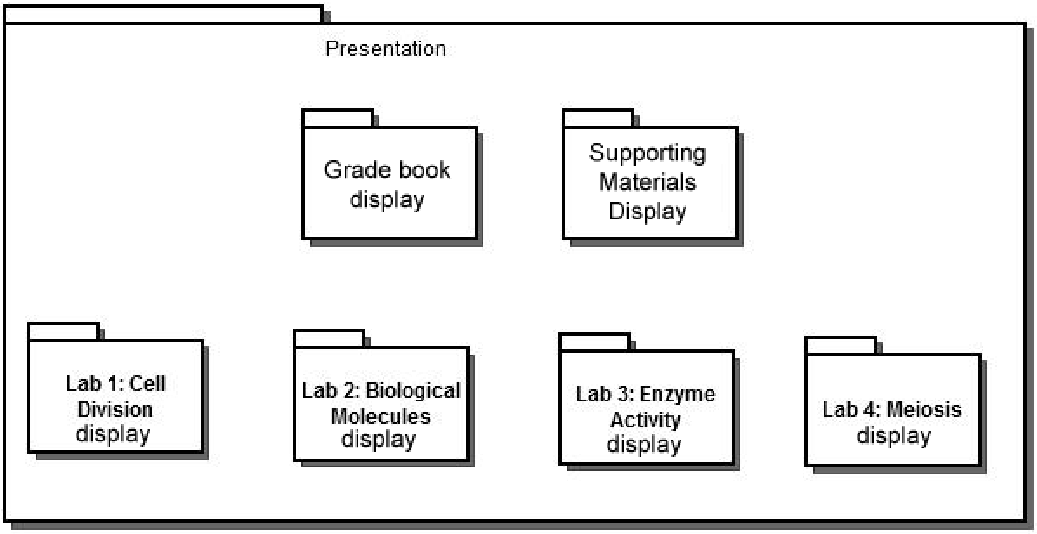
10. System Architecture and System Design

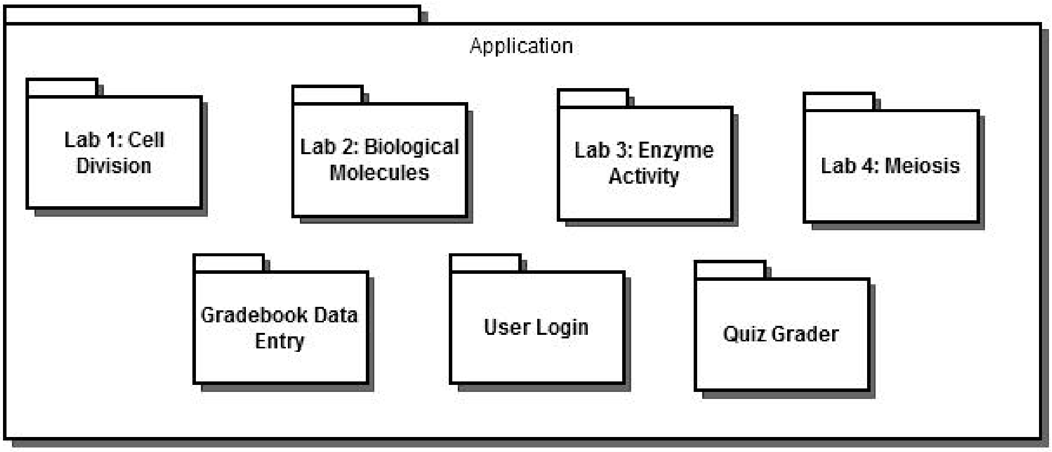
10.1 Architecture Styles

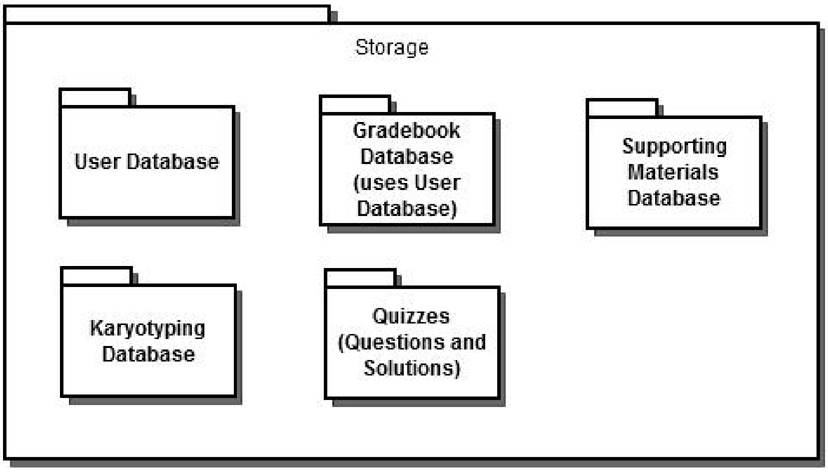
There are two major components and architectures used within our application. The first component being the website. We will be programming in a PHP environment built on top of a Linux server running a Apache, MySQL, and PHP (also typically known as a LAMP stack). We will be developing in a Model, View, Controller (also known as MVC) to help the structure and development of PHP code. Note that this will follow the standard server-client architecture where the server will distribute the necessary content to the user provided their input. Note that the server will also require a database architecture where it must input grade results and user information into. We will be relying on this to provide dynamic data.

The second component involves animation and the actual labs themselves. Here, we will use a standard UI architecture, using Flash, to display and show educational and hands-on material to the user which was first provided to them by the server-client architecture.

10.2 Identifying Subsystems





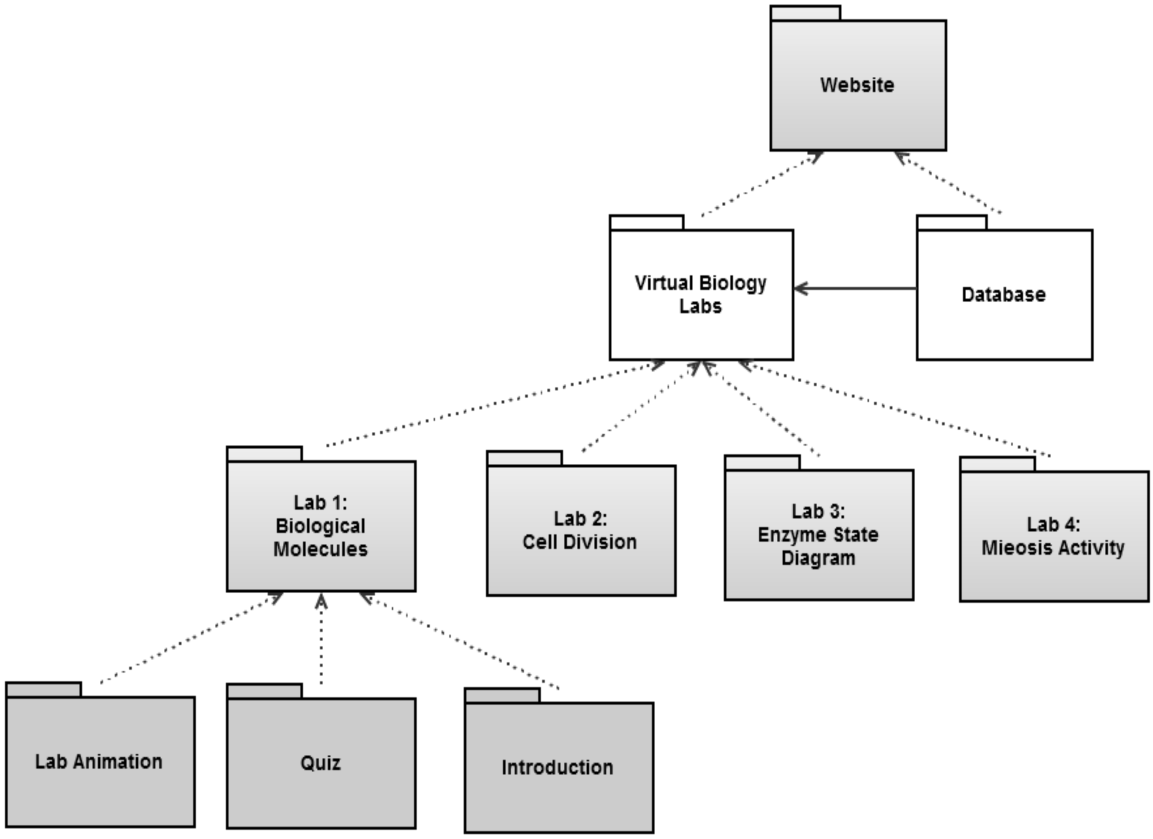


We have divided our subsystems into three different levels each with different purpose for the individual components. In our Presentation level, we handle purely the visual aspects of the system and how they will be displayed to the user. For example, we have our Flash Animations, which consists of the four lab reports and the corresponding quizzes.

In our second level, the Application level, our subsystems will be described in terms of what they will be actually doing as a part of the system. Here we have descriptions for each of the labs and how they will react and respond to user input. A more specific example of an item from this subsystem would be in Lab 2: Biological Molecules, where the user deals with the spec20 instrument. This subsystem will layout how this instrument behaves to certain conditions and what it should inform the system the output is. Our application level also mentions quizzes which on the presentation level will be handled by the "Flash Animation".

Finally we have the Storage level which contains all of our databases and static information about items such as the lab methods and data, the correct solutions to quizzes, any content that a professor has uploaded for supporting materials, and where our user's information will be dealt with. The user data is by far the most important in this section because this will contain their progress in labs, which will be passed to the gradebook when certain checkpoints/requirements are met, and all of their identifying information so that they will be able to continue their work from where they left off each time they login.

Within the database itself, we store certain class data, permitted by a key value. When a new student comes to register, they must place in this key value in order to be accepted into a section. By registering, we store their username and password into a database, making sure that password is not stored as plain text (using md5 hashing). We also keep in store every lab they complete including the quiz grades they get. We access the necessary data for the user by their unique username only when needed. For example, we perform a search for the quiz grades for a unique username and pass it to the user's web browser.



Here we can see the subsystems that occur in our project. There are 4 main tiers for which the subsystems are organized. First is the main Website folder which contains all the subsystems. The second level contains the organizational layer consisting of the folder containing all the labs, and the database where the grades are stored. Below this is the division of each lab by itself, followed by the final layer containing the subsystems pertaining to each individual lab.

Starting from the second level of the diagram, we can see this is the lay that handles the interaction of the labs with the database on a superficial level, indicated by the arrow connecting Database to Virtual Biology Labs. This directory then contains the contents of each lab, which have been split by each lab, rather than by dividing them into the subsystems of “Lab animation, Quiz, Introduction”. The organization depicted above seemed to be much more logical because it allowed for you to navigate by content pertaining to certain Labs, rather than navigating based of the type of content and then finding the item in question. Each labs data is kept exclusive from other labs which also reduces the chance for accidental overwrites and reads from items in another lab.

Further, each lab is divided into three more subsystems, “Lab Animation”, “Quiz”, and “Introduction”. Starting with simplicity, the Introduction contains the basic requirements to update the introduction content for each lab and has no connection to outside subsystems. Next is Quiz, which contains the methods for the database to be updated with the grades, grade the quiz, and contains the actual questions and solutions to the quiz. Finally is the Lab Animation, which in our specific case only contains the Flash Animation file for the simulation to run. This file contains the means to generate the graphics, allow for user interaction, and all of the actual data to compare the simulation results with.

10.3 Mapping Subsystem Hardware

Our application requires both a server environment and a client environment. The server environment will pre-process hypertext using PHP given input from the user and information from the database. The server will be responsible for grading quizzes, showing and calculating grades and displaying the correct labs.

The client will require a web browser that is compatible with the latest standards of HTML, CSS, Javascript, and Flash. Preferably browsers such as Chrome, Firefox or Safari. The client will be in charge of displaying content served by the server properly, meaning, interpreting the HTML, CSS, Javascript and Flash data.

As per discussion with Parneet, we feel that this section of the report does not apply too our specific problem and would like to be able to assign a majority of the credit from this section to our interaction diagrams

10.4 Persistent Data Storage

Our application will require to store session data onto the client’s system and browser. This is better known as cookie data, which helps the server interpret who the request belongs to and where to grab proper information from the database. As mentioned, we will also require a database which will store user information such as name, grades, passwords, emails and more that are required for dynamic data computing for each individual user.

10.5 Network Protocol

The server and client will require standard HTTP ports and communication. This communication requires standard socket transfer, POST and GET request support. This type of data has been chosen due the nature of standard HTTP servers and Apache's schema.

Apache's scheme is a basic HTTP web server. We use this over other HTTP servers because how "stable" (it has been around for a really long time and is used in production environments). Apache's scheme is a a fixed based, child based server. It will serve requests on a first come, first serve basis and automatically recover itself if any crashes it encounters. We are not using event-based web servers due to its high-error rate and low-grade production use. Apache is also used since it is the easiest to setup across multiple environments especially for development use.

10.6 Global Control Flow

10.6.1 Execution Orderness

Our system is an event-driven system where a user can execute and perform labs in orders they wish to choose. However, there is a linear path in which the user must complete the labs. For example, a user can choose to browse a lab and choose to view their grades before taking a quiz.

10.6.2 Concurrency

Our application will be using multiple threads on the server-side level. Apache, the utility that serves HTTP and PHP requests, handles concurrent threads by spawning a new child process for each unique client request. Apache will also spawn its on PHP thread for each child process if required.

10.7 Hardware Requirements

Our system will require a server with standard resources. Of minimum a virtual machine with root access. It should at least have 512 megabytes of usable RAM and a minimum of 4GB disk drive. We will require Apache, PHP and MySQL installed on this server.

Our application will require clients to have a computer which can meet the minimum standards of running a web browser and the Flash plugin. We will require at least a screen that can run a resolution of 1024 pixels wide.

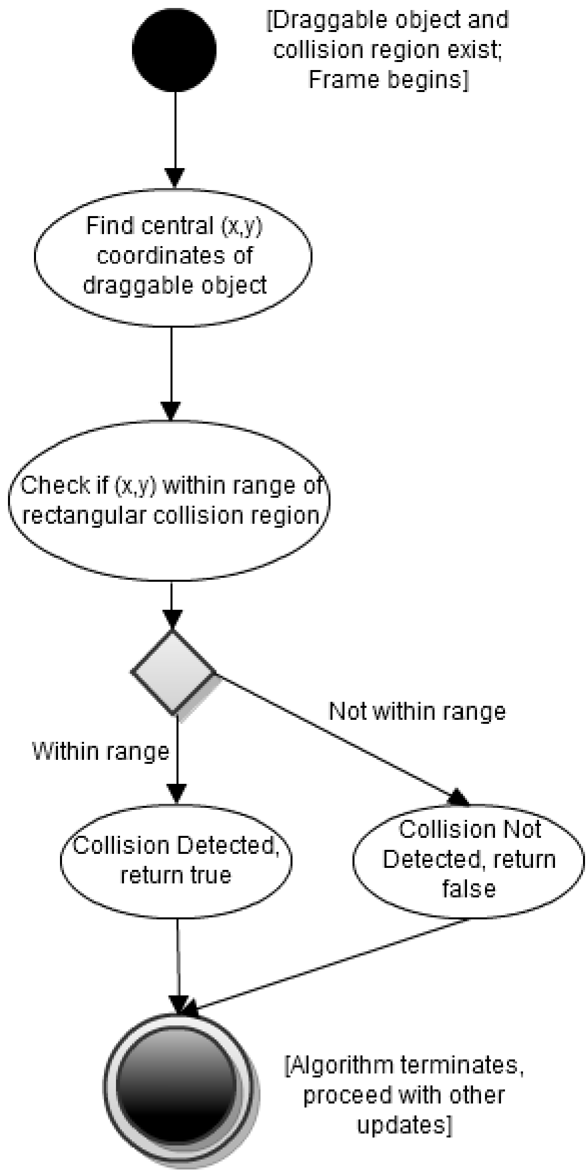
11. Algorithms and Data Structures

11.1 Algorithms

Although the goal of the virtual biology lab is to simulate real lab situations and phenomena, we use a very basic modeling scheme to present this to the user. Interaction with the lab demos is guided, meaning the system will wait for proper user input before moving to the next state. Actual biological and physical phenomena will not be mathematically calculated; instead there will be discrete changes in the properties of the system that simulate results you would expect in an optimal lab setting.

However, the most prominent example of algorithm use in our project would be our collision detection algorithm. Many of the labs require the user to move objects to certain rectangular regions, and these regions will be hard coded into the system by inputting the coordinates of their 4 corners. For example, in lab 1, the user is required to move two chromosomes into their proper positions in the cell. On every frame update, we will run a method that computes whether the chromosome is in the proper region of the cell.

First, we would need to calculate the central x and y coordinates of the chromosome object (which is being dragged by the user's mouse). This is preferable to the (x,y) coordinates of the mouse itself because it gives a more accurate reading on where the object actually is. We then use these values and check if they within a certain range (within the rectangular region), and if so, the system should detect the collision and proceed to the next step if applicable. This basic algorithm can be expanded to any object and any region we desire. The activity diagram is shown below:



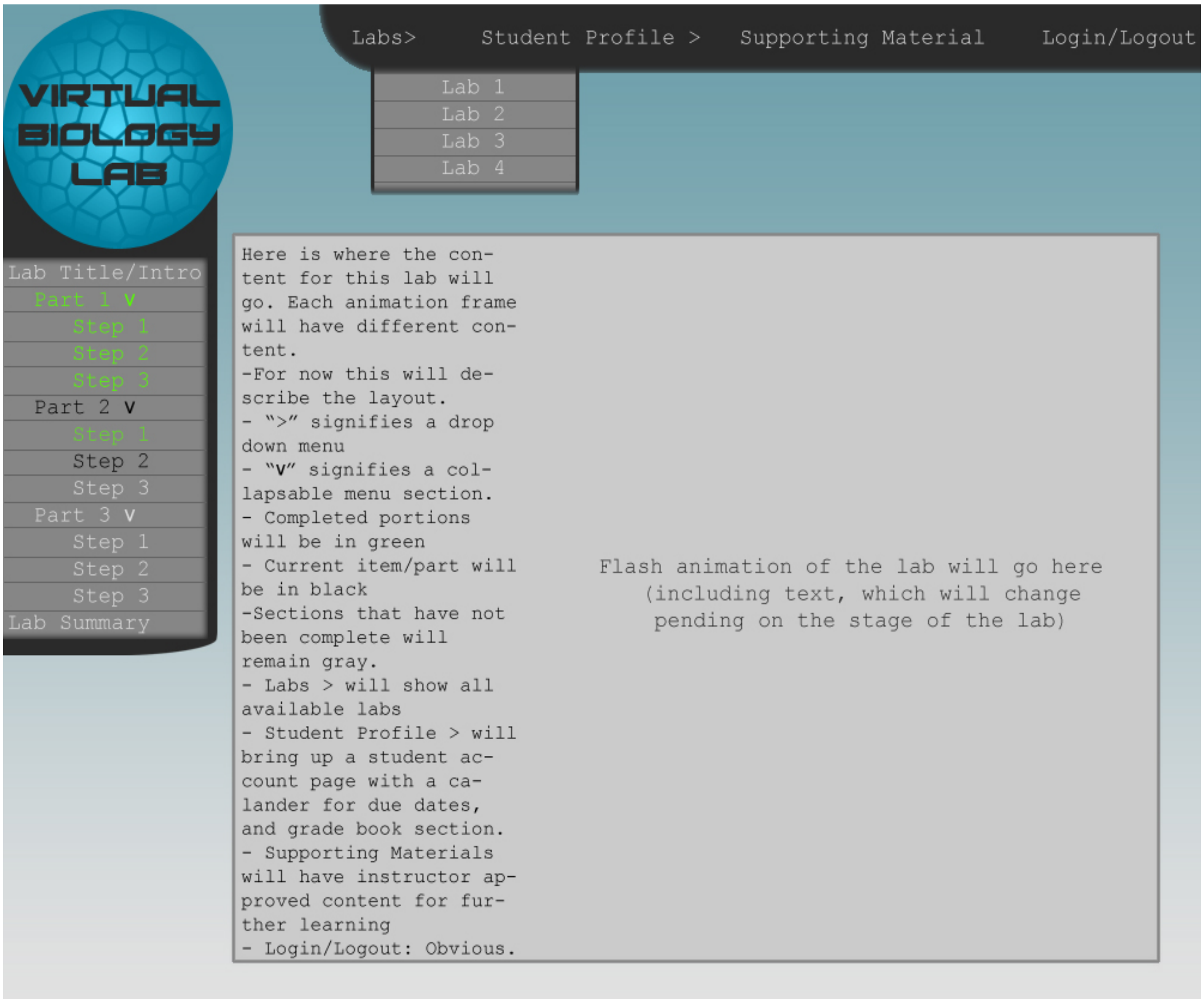
11.2 Data Structures

For calculating grades, obtaining class rosters and so on, we will be obtaining the data that has been associated from the database into an associative array. This associative array acts similarly to a hash-table, allocating all spaces and keys into memory, allowing us to handle performance in constant time. With these associative arrays, we are able to calculate averages, means and much more analytics for grades. We will also be using associative arrays into grading quizzes and so forth as defined from a POST operation from the web-browser.

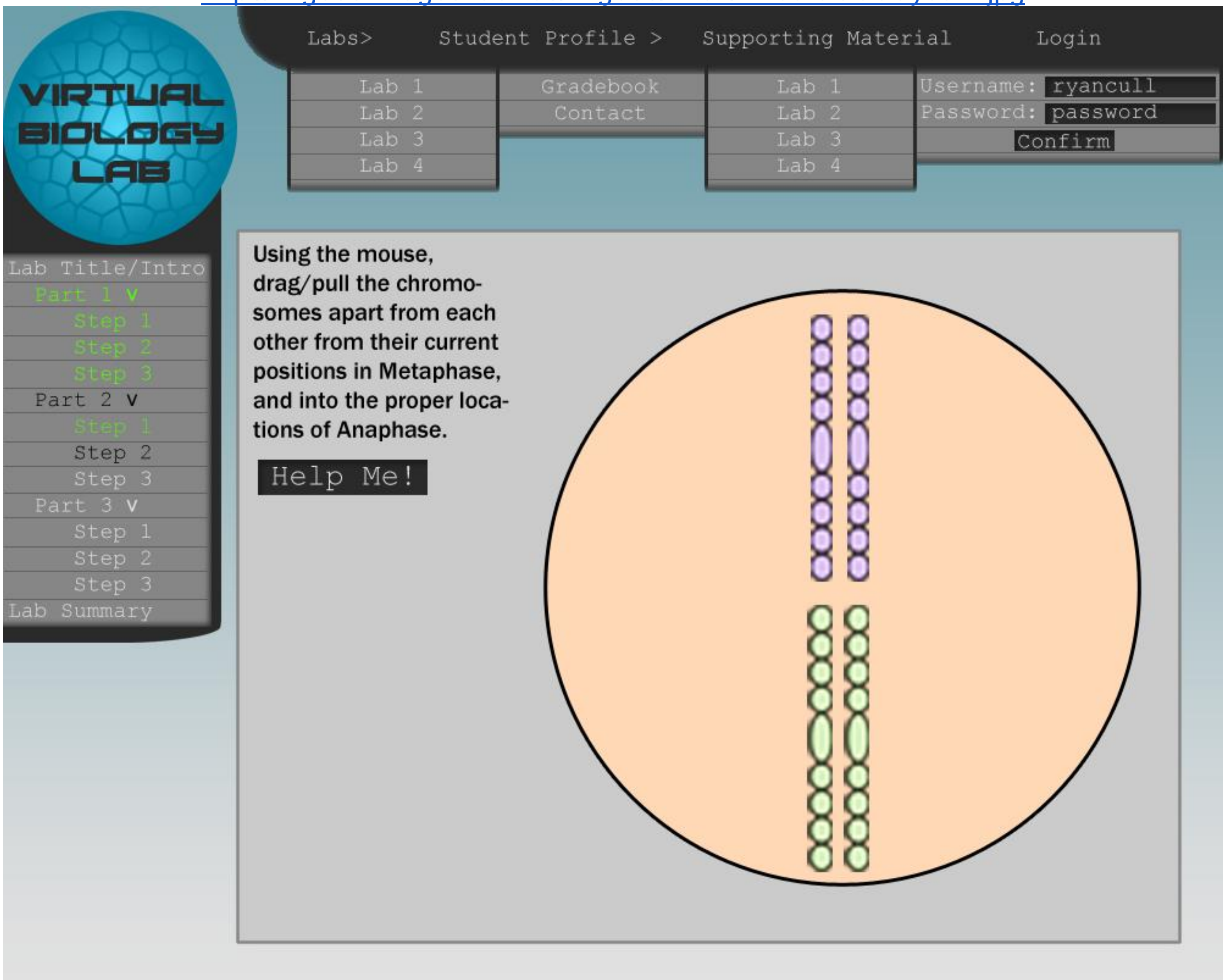
All other items, such as the labs and animations will not require any sort of complex data structures as they are formed more as a “movie” that runs in linearity. Also, we are unable to understand the technology of Flash as it is proprietary technology.

12. User Interface Design

12.1 Preliminary Design



A description of the functionalities is listed in the text of the layout design. To see a full size image go to http://img638.imageshack.us/img638/2407/virtualbiolablayout4.jpg



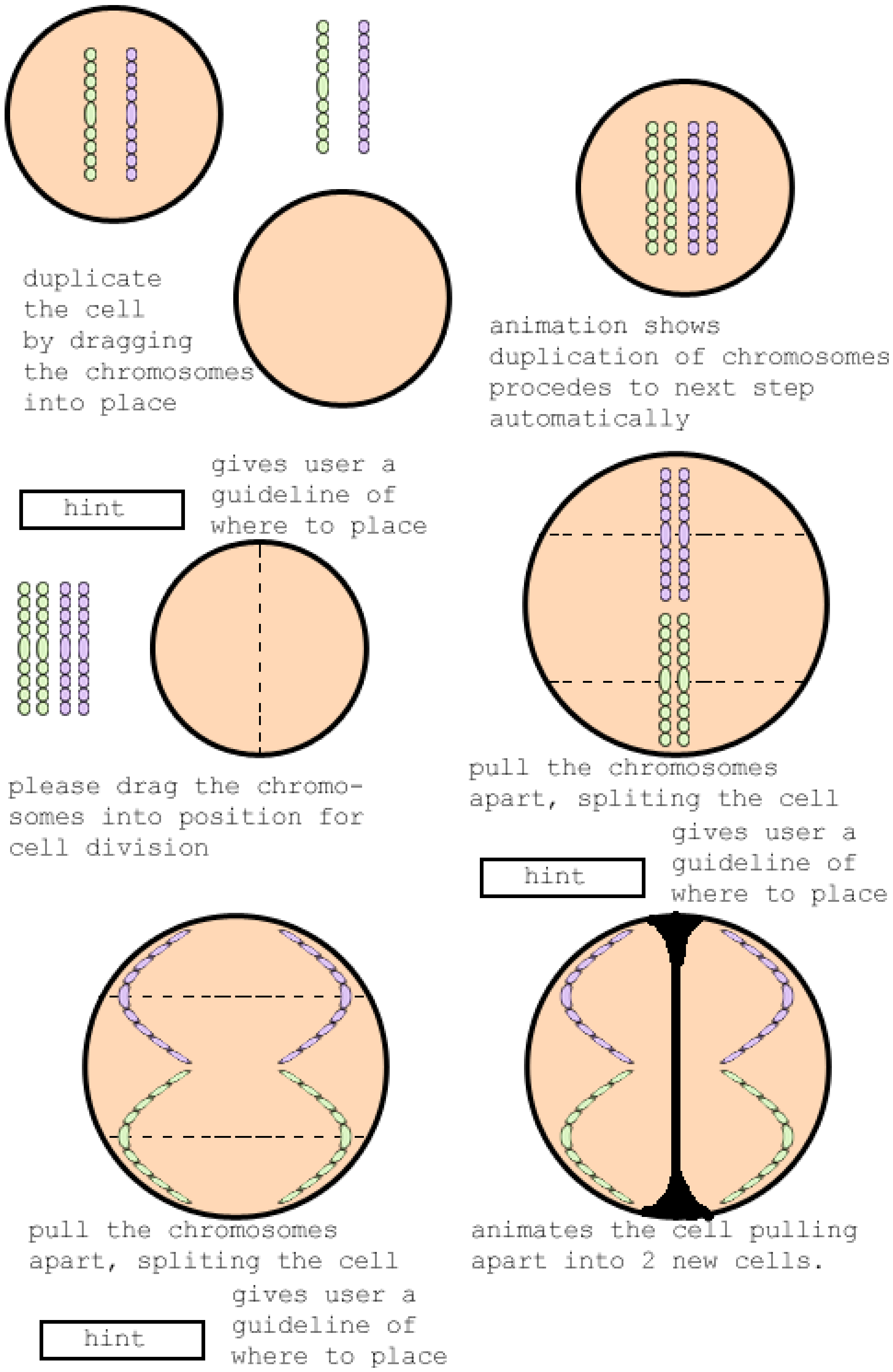
This image shows an example of what one of the steps of the lab may look like. The only difference between this and the final product is that the drop down menus will not always be in their expanded state. These drop downs are shown to make it easier to grasp the descriptions below in the following paragraph and Section 4.2.

For a general overview of the layout, we will have a central navigation bar located on the top which will be used to access to the main subsections of the application. Under Labs, the student will be able to find a drop down listing all the labs that are currently available on the site. Under student profile we will allow the student to view a grade book of the assignments that they have completed and how they did on them. In Supporting materials, the professor of the lab can choose to link to other content which can assist the students in their completion of the labs. Finally the Login/Logout will link to a simple page (or dropdown item) allowing the student/ professor to access their personal account. Further, professors will have an option when logged in to view different content than a student would. For example, “Student Profile” would be changed to give the teacher access to information about all the students who are registered for the course. Similarly in Supporting Materials there will be an option for the professor to add more content to the list.

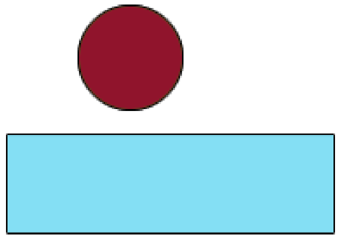
Here I have linked to a larger image showing the storyboard for the flash animation that the user will interact with for the Mitosis lab:

Lab 1:

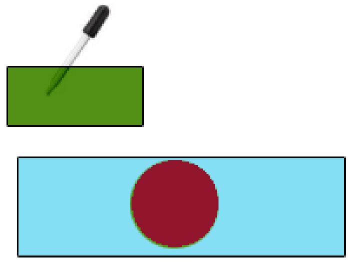
<u>Part 1:</u>


duplicate the cell by dragging the chromosomes into place
animation shows duplication of chromosomes procedes to next step automatically
hint
gives user a guideline of where to place
please drag the chromo-somes into position for cell division
pull the chromosomes apart, spliting the cell
hint
gives user a guideline of where to place
pull the chromosomes apart, spliting the cell
animates the cell pulling apart into 2 new cells.
hint
gives user a guideline of where to place

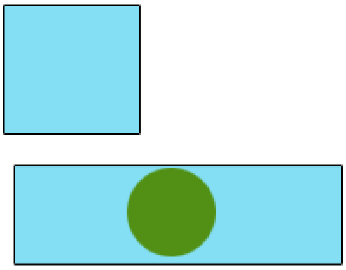
<u>Part 2:</u>



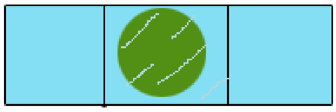
place the subject in the middle of the slide by dragging it.



apply the dye solution in order to observe the subject under the microscope

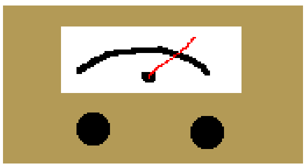


place the slide cover on to keep subject in place

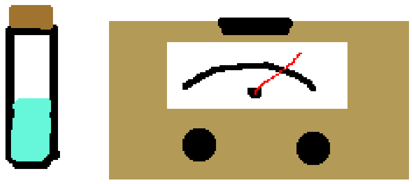


place the slide under the microscope and observe.

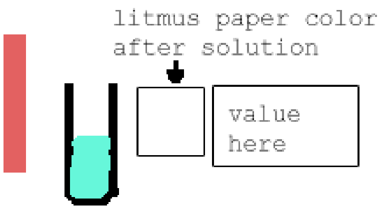
Lab 2:



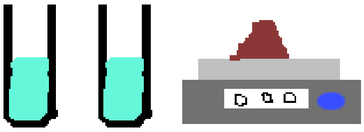
turn the knobs on the meter to calibrate. Alter the zero and infinity ends to get proper calibration (turn and tell if correct)



insert your "blank" tube filled with distilled water and verify results. Drag tube into slot on top of meter.



insert the litmus paper and select the pH value you read. (If wrong will tell them to repeat and look more carefully)

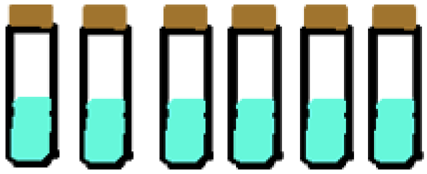


Here will be an animation showing the inspection of the Glycosuria, proteinuria, albumin, and hemoglobinuria (may later evolve to be interactive with some sort of scale to weight compuptnds and make proper mixture)

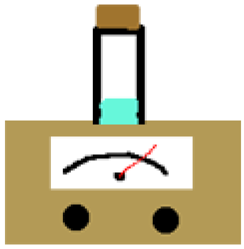
Lab 3:

<u>Part 1 & 2: processes will use same animation, but measure different values</u>

bottles of each solution will be shown to pick from
test tubes are labelled 1-6

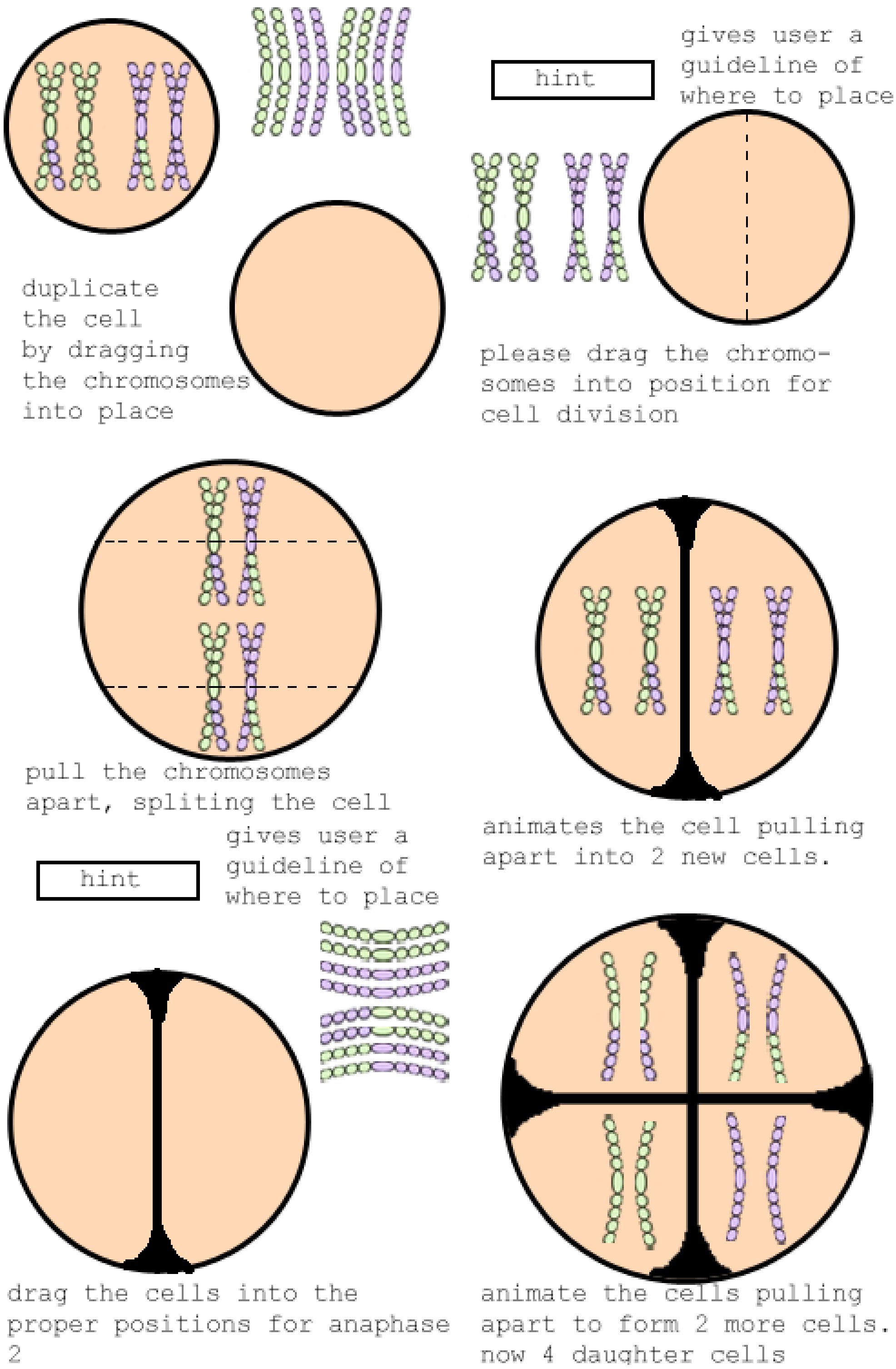


create the mixtures of the required amounts (will be listed in animation) by clicking the proper bottle and selecting the value to add



insert test tube required into the machine. start timer. (not sure about execution here) may just run timer for set duration as example

Lab 4:



12.2 User Effort Estimation

Login Process

1. Navigation: 2 mouse clicks
    - a. Click “Login” button
    - --- complete data entry as show below ---
    - b. Click “Confirm” button
2. Data Entry: 3 mouse clicks, keystrokes vary depending on login and password length
    - a. Click “username:” text field

b. Input account username in text field

c. Click “password:” text field

d. Input account password in text field

Logout Process

1. Navigation: 2 mouse clicks

a. Click “Logout” button

--- if a stage of a lab is in progress asks for confirmation ---

b. Click “Confirm” button

Student Profile

1. Navigation: 1 mouse click

a. Hover over “Student Profile” to enable dropdown

b. Click on the item which you would like to view

Sub-navigations:

Grade book

1. Navigation: varies based on user request

a. User can choose to view the overall grades for each lab (0 clicks)

b. User can choose a lab to expand and see subsections (varies)

Future items: will follow similar complexity as Grade Book

Lab Selection Process

1. Navigation: 1 mouse click

a. Hover over “Labs” to enable dropdown

b. Click on the lab title which you would like to work on

Lab Interaction

1. Navigation: varies on user choices

a. Can interact with the flash animation (clicks vary for each lab section)

b. Can select a specific section to work on (2 clicks)

i. Click Section Title, to expand content listed under it

(not necessary if section is already expanded)

ii. Select section (refreshes flash animation to proper step)

Professor Access

Grade book:

1. Navigation: varies on user choices

a. View Individual student (1 click, keystrokes vary)

i. Click “filter” button

ii. Enter Student Name

iii. Click “OK”

b. View by Assignment (1 click)

i. Click on assignments title (listed on side bar)

Professor Supporting Materials

<u>1. Navigation:</u> 1 click

a. Hover over Supporting Materials

b. Select “Add Material” from list

--- data entry as stated below ---

<u>2. Data Entry:</u> varies on user input

a. Select type of material from list (2 clicks, one to open, 1 to select)

b. Input Require information (undetermined at this point)

(most likely contains title, link to content. Keystrokes vary per entry)

12.3 User Interface Design and Implementation

Since our first report, our initial screen mock-ups have not changed drastically. One of the main changes is that some of the artwork has been improved from simple MS Paint sketches, to full fledged final presentation material. This can be seen in the demo of the first lab, which can be found at the end of Section 7.4.

In our demo you can see that there are almost two panels in our design. The top panel which has a lighter gray background is the directive panel. This area displays information to the user about the current stage of the lab and what the user should be doing in order to interface with the system. For example, our first “slide” of the animation instructs the user to “Drag and drop the chromosomes into place so they look like this:” accompanied by a diagram of the cell that the user needs to duplicate. Below this is a darker gray panel, which is the interactive portion of the interface. Here we render all the items that the user needs to complete the given instructions and allows them to follow the required steps. In our first slide, we can see the user is able to drag and drop the chromosomes to any location they desire in the frame (including the directive panel). Once the user lets go, the item being dragged will remain there and remain movable until the user has placed it correctly.

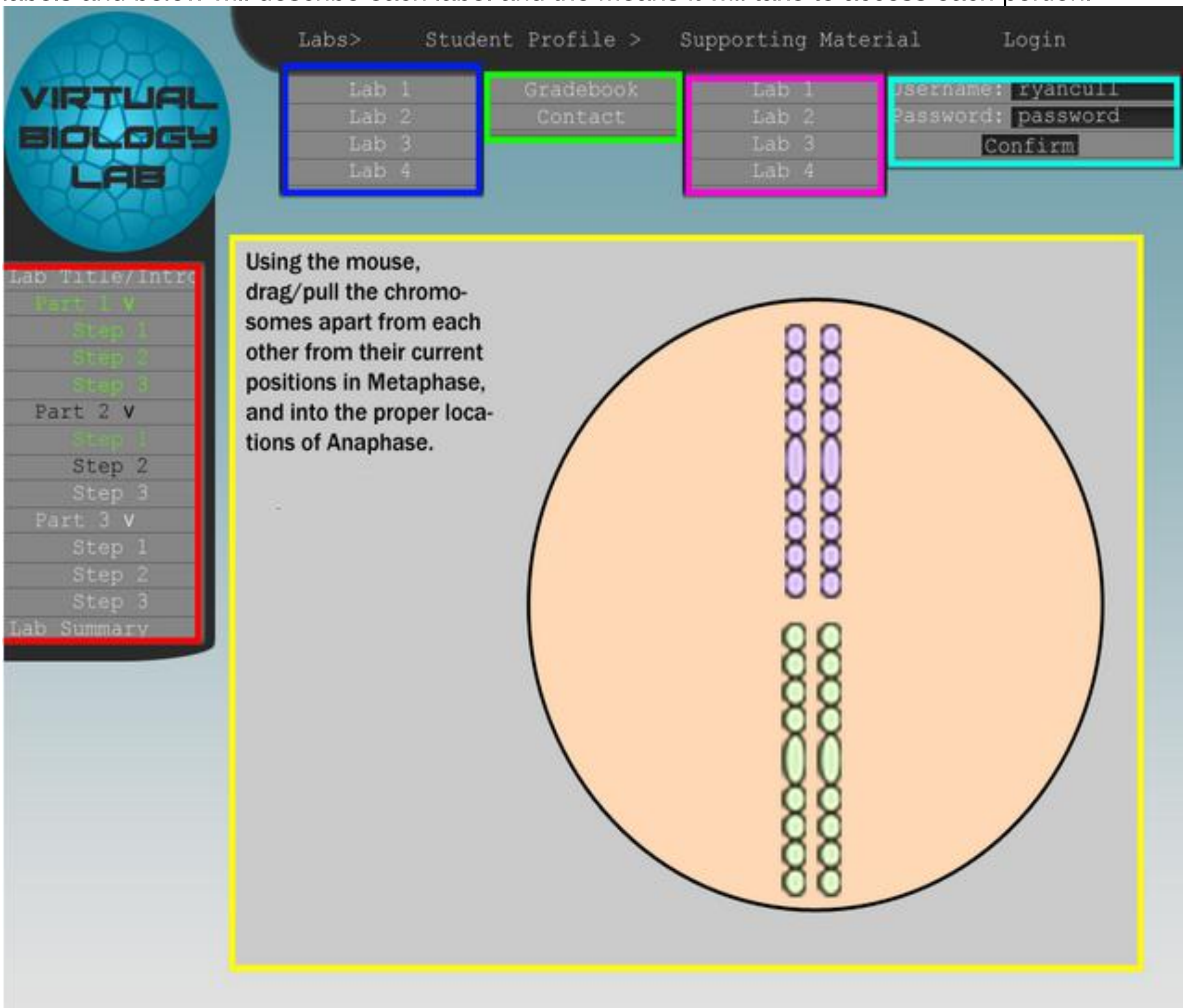
Due to precision issues, the user cannot be expected to place the chromosomes in a pixel-perfect position, so there is a margin of leniency where the system will automatically reposition the chromosome to the “accepted” position if the user is close. From here the user will continue to perform the proper actions until the system is satisfied with the results, at which point a the system will prompt the user to continue to the next slide in the lab. In which a similar user interaction will be repeated.

Our lab demonstrations will also be incorporated into a website with a fully functioning back-end that maintains user progress. This system allows for users to select which lab they would like to work on, look up what sections of each lab they have completed, and even find supporting materials for the lab assignments. These features are all extremely easy to access requiring only a handful of clicks in order to navigate to each menu. Each of these features is displayed directly on each page of the site, which grants this instant access and eliminates an immeasurable amounts of wasted time due to searching for links to pages.

Overall based on these factors, our system is proven to have a very high ease-of-use, while still maintaining a visually appealing and stimulating interface, making the user experience more enjoyable. This ease-of-access will be extremely easy to see in our first demo, which will include the web page that includes the individual lab simulations.

The design described below shows the functionalities of our initial ideas for implementation. Since then we have drastically altered our layout to a much more professional and simple design which is discussed following the description of the initial design concept.

For each of the requirements we have listed I have provided the following picture with labels and below will describe each label and the means it will take to access each portion.



Red: Here in the red section we have a listing of the current sections available in the lab that the user is working on. This information will very depending on what section the user has entered. For the “Labs” the system will display information similar to the current information, where every item with a “v” next to the name represents a drop down. This action requires 1 click to either expand the content below, or hide the content in that subsection. The other operation this navigation can support requires 1 click, where the system will refresh the page and bring the user to the requested section of the lab. For our other sections such as “Supporting Materials” we will present the main subsections for each lab, which will interact the same as the other drop downs (1 click) and a listing of the content that the professor shares (also 1 click). Gradebook will provide the same functionality of a single click interface, but each item will refresh the content showing the grades for the selected assignment.

Blue, Green, Pink, Cyan: These sections all have the same base functionalities. When the corresponding header is clicked, the content shown in the drop downs will appear. This means that each of these require a single click to access the content. To further access each sections content, the user will only be required one more click to refresh the page with the requested content on the page.

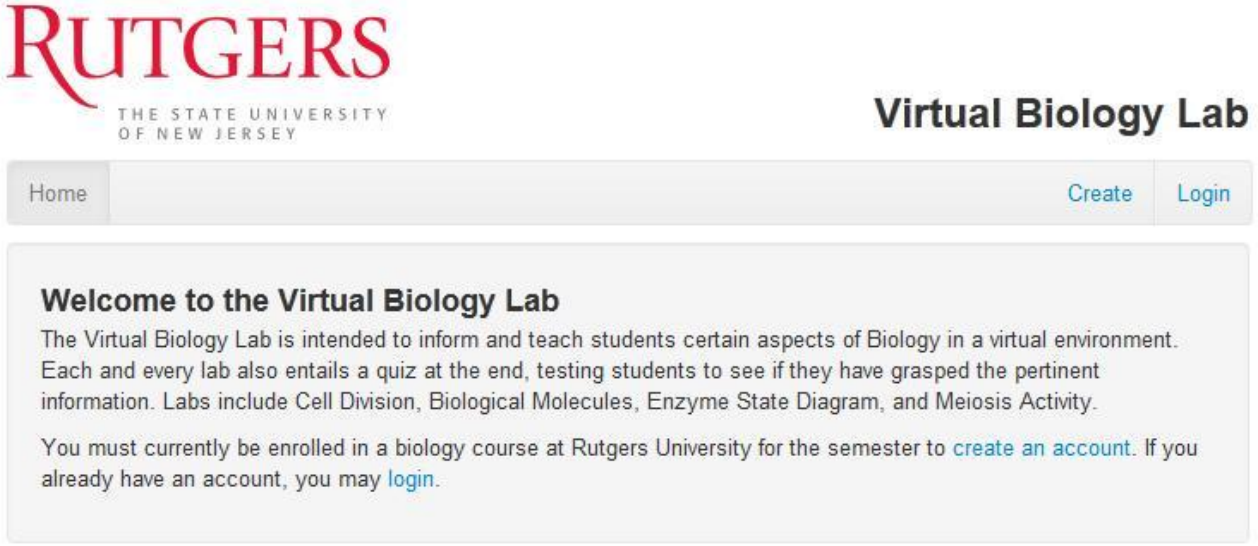
Cyan: While accessed by the same method as the other 3 sections in the previous paragraph, this section requires more interaction. As you can see in the linked animation here (http://www.mikedilalo.com/VirtualBiologyLab/Report_2/Animation.gif), I have shown the way to access the each header and view the drop down. This animation goes further by showing the instructions and the remainder of the animation. The amount of user input in this section will vary based on each user. All users will need to input their user name (generally ranging from 4-12 characters) and a password (minimum requirement of 6 characters, ranging from 6 - 12 keypresses). Then one additional input will be required, either a key press of the return key to confirm the information, or a click on the confirm button.

Yellow: This is the section that will require the most user interaction. While the user is performing the lab assignment, they will be instructed to perform multiple tasks involving user inputs. For example in our first lab demo, we will go slide by slide and quantify the minimal number of interactions required to complete the lab so far. First the user will be required to move the two chromosomes into the correct locations, (minimum 2 mouse clicks and dragging to the proper position and releasing). To proceed to the next step the user will need to click the “Next” button (total of 3 mouse clicks). On our next slide, the user will see an animation and be asked to click “Next” in order to proceed to the next step (total: 4 clicks). Again the user will need to drag two items into place (minimum 2 mouse clicks and dragging to the proper position and releasing) and continuing by clicking “Next” (total: 6 mouse clicks). Once again an animation will be played and the user will be asked to click next. Therefore in the lab so far, the user needs seven mouse clicks to properly interface with the lab, which is fairly simple given the subject matter being covered.

Final Design Implementation

Below is an elaboration of our current design implementation as can be seen in our project demos.

First we will start of with the main page of the site, the homepage:



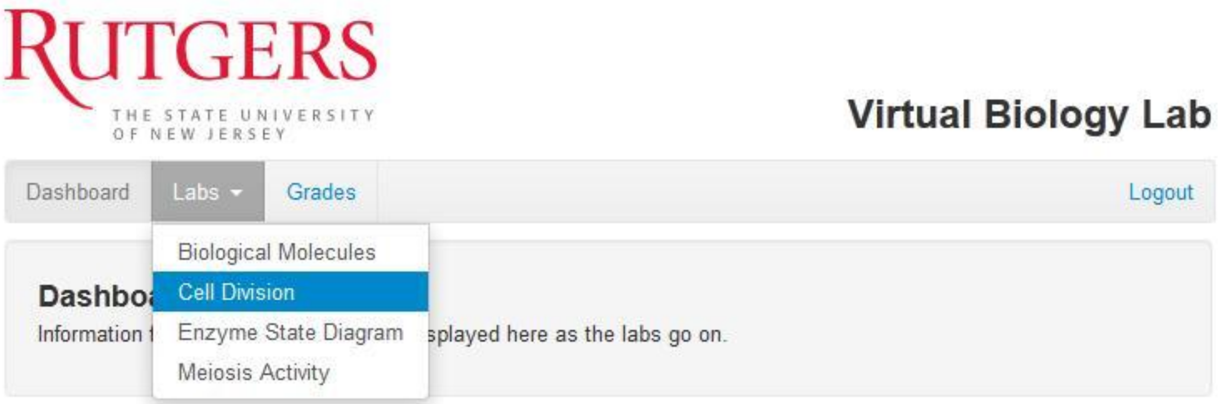
Here we can see the options to Create an account, Login to an existing account, and then a display of general course information. In this area the professor can post information such as due dates, announce upcoming course events, and anything else they consider important to the course as a whole. From this screen we have two navigation options, the first we will explore is “Create”.

Upon clicking the “Create” button, you will be brought to a new page where you are able to create your unique user account. After inputting the user fields shown in the picture below, the user will click the “Create Account” button and then see a visual confirmation that their account was created and requests that go to the Login page to access the account.

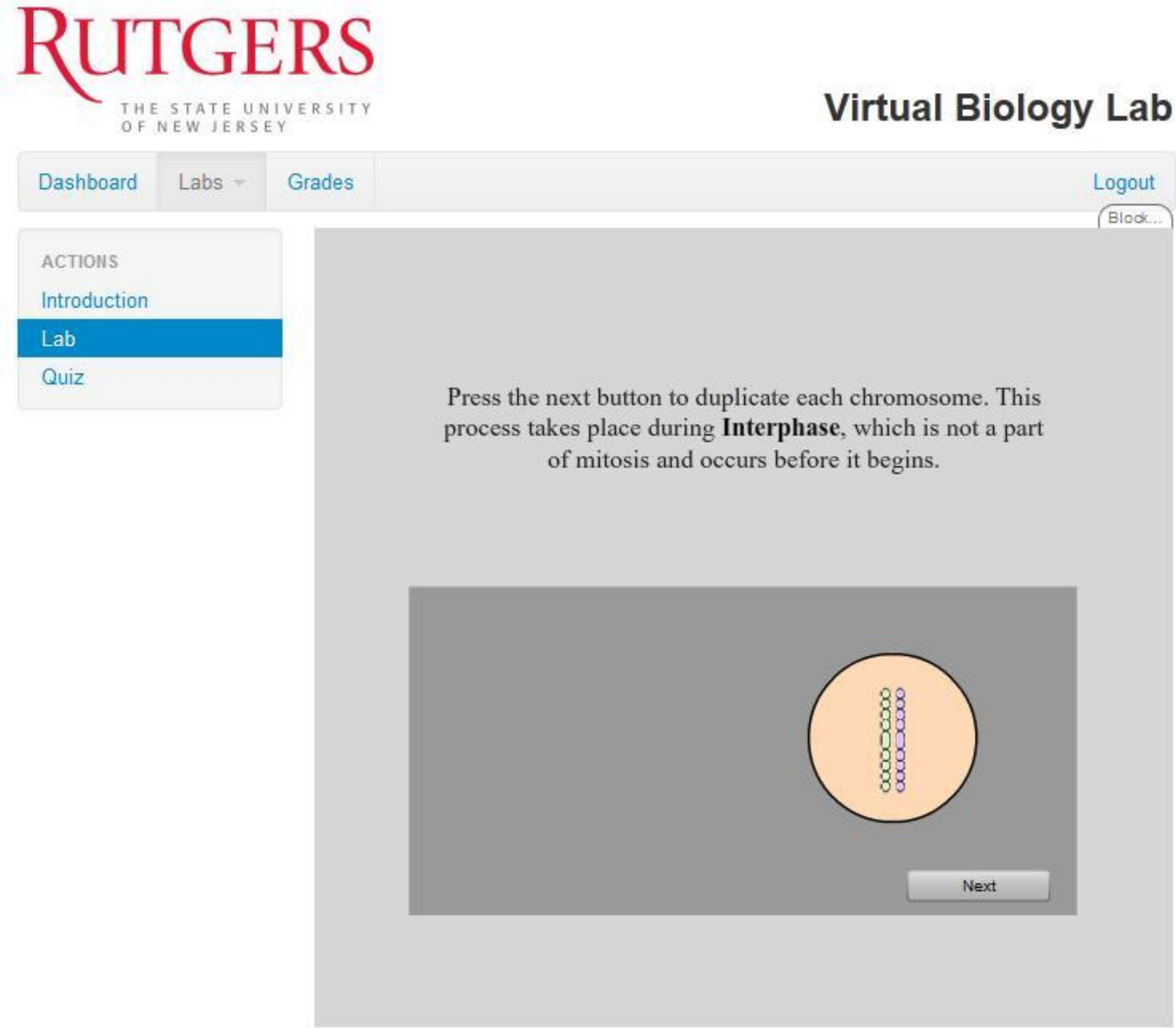
Now we will step into the Login screen, which is very similar to the Account Creation screen, but only requires your username and password and then logs you in to your account, while redirecting you to the
Dashboard.

When the user is redirected to the Dashboard, they are presented with three new options, “Labs”, “Grades”, “Logout”. Up to this point in our interface, the user has had to use approximately 2 mouse clicks and a variable amount of keystrokes to put in the user’s unique information. The most simple option that stems from this page is the single mouse click to “Logout” which signs the user out of their account and brings them back to the site home page. The next option goes to the “Gradebook” page. This page is one of the more complex item to implement, and was unable to be fully implemented to our initial vision. For our current implementation we were only able to track and monitor students grades on quizzes. Interfacing the lab implementations with the Gradebook proved to be more difficult than initially anticipated.

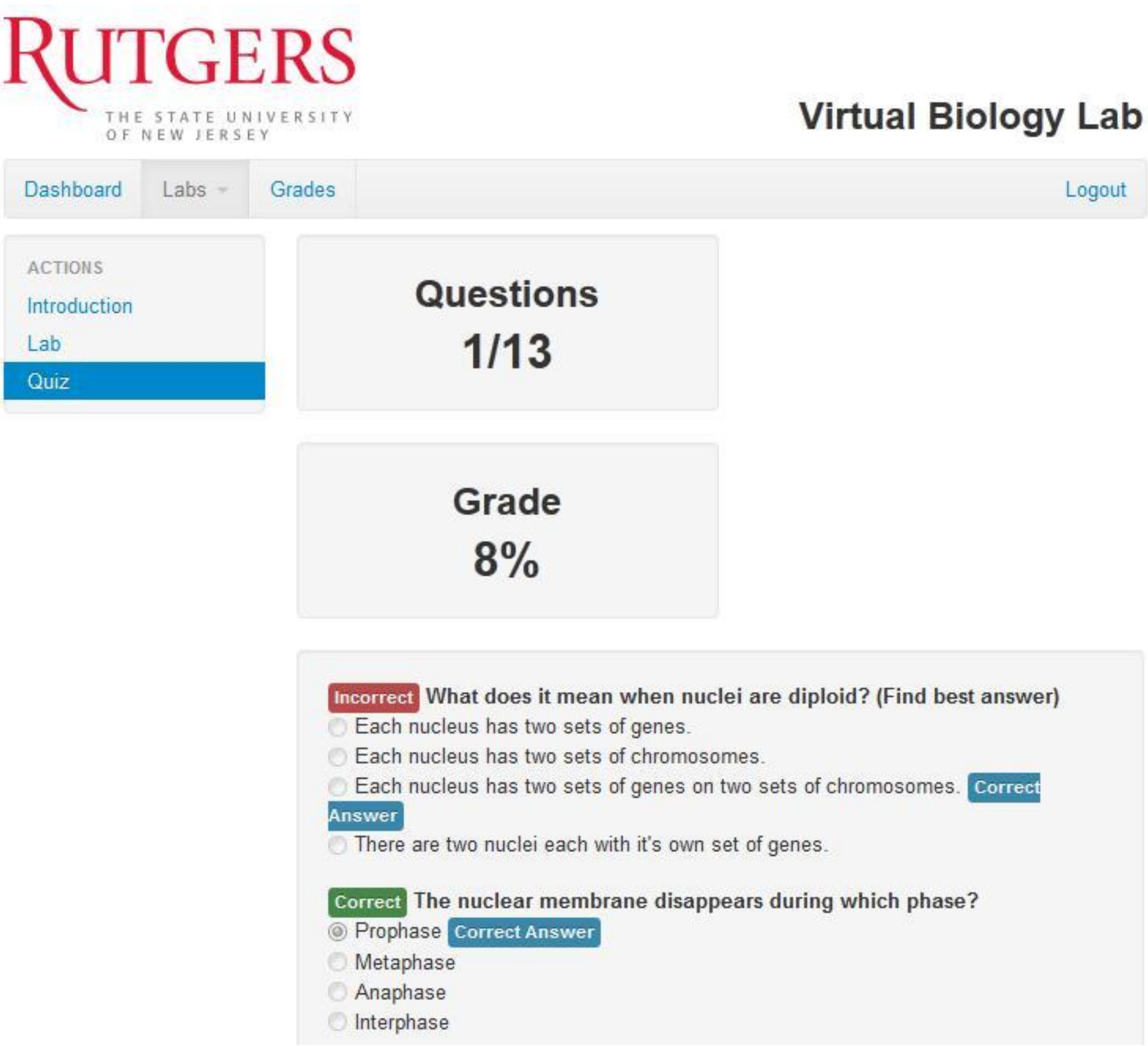
The third option for the user to select leads them to the selection of the 4 different labs. This item is a drop down menu as can be seen in the picture below and the user is able to select the lab in which they would like to work on. As you can see below, with a simple click on the “Labs” button, the user will see this drop down menu and have the selection they are hovering over be highlighted. In total the process to select a lab once logged in requires only two mouse clicks.



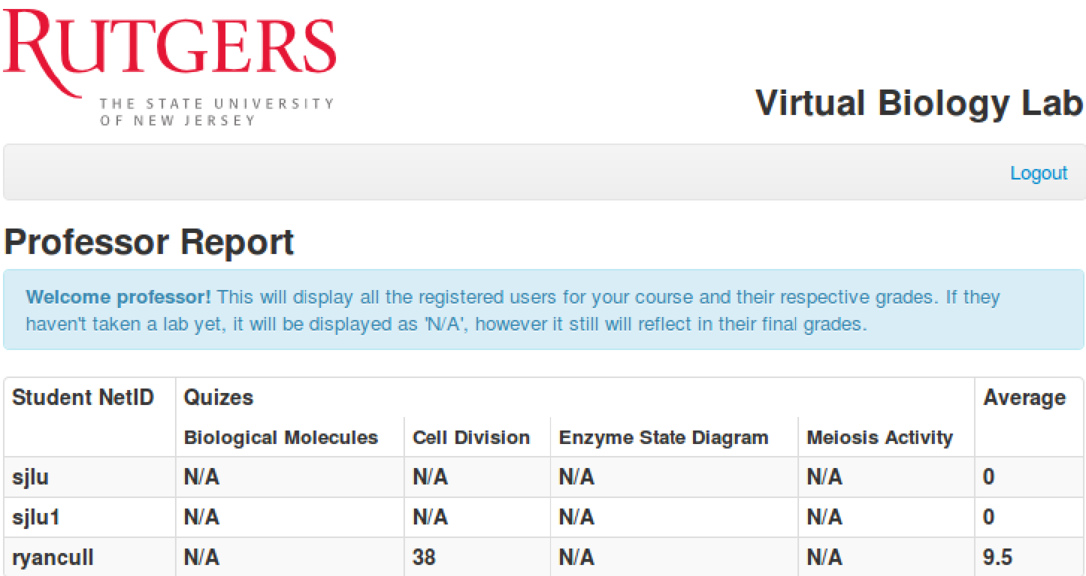
From here the user is brought to a page where they can see a new menu on the left side of the page which has links to the “Introduction”, “Lab”, and “Quiz” pages. The introduction page is very similar to the dashboard and displays a simple text box. This page is also the default page that comes up when you select a lab from the drop down menu. A second option to select is the active lab page which contains the panel that was described in our preliminary design with the directive panel and the interaction panel and is pictured below. Access to this page requires a single additional mouse click from the previous steps.



The final area of navigation on the web page is the quiz page. Here the user will be tested on their knowledge of the lab assignment and the topics involved. These grades will ideally be recorded by our system and appear in the grade book automatically, but as of this time it is not completely implemented. When a user fills out the quiz page they are presented with a very clean and simple way to see their results immediately after take the quiz. Each question requires a minimum of one mouse click per question and then another mouse to submit the user's final answers.



The only major alteration in these paths of navigation are when the user logging in uses a professor account. Upon logging into the professor account, the user is taken directly to the Gradebook page where it shows the grades for every student in the course. This can be seen below:



Student NetID	Quizes				Average
	Biological Molecules	Cell Division	Enzyme State Diagram	Meiosis Activity	
sjlu	N/A	N/A	N/A	N/A	0
sjlu1	N/A	N/A	N/A	N/A	0
ryancull	N/A	38	N/A	N/A	9.5

13. Design of Tests

Test-case Identifier: TC1a **Use Case Tested:** UC1, main success scenario (partially completed lab) **Pass/fail Criteria:** selects a Lab which is not yet complete **Input Data:** Lab Selection	
Test Procedure	**Expected Result**
Step 1: Selects a lab that the student has already completed	System pops up with an error; informs the student that they have already completed the selected lab
Step 2: Selects a lab that is not yet complete	System checks to see if student had previously begun the selected lab; sends student to the specified lab page wherever they had stopped previously
Step 3: Leaves lab before completion	System records the point at which the student left the lab

Test-case Identifier: TC1b **Use Case Tested:** UC1, main success scenario (new lab) **Pass/fail Criteria:** selects a Lab which is not yet complete **Input Data:** Lab Selection	
Test Procedure	**Expected Result**
Step 1: Selects a lab that the student has already completed	System pops up with an error; informs the student that they have already completed the selected lab
Step 2: Selects a lab that is not yet complete	System checks to see if student had previously begun the selected lab; sends student to the specified lab page at the beginning of the lab; upon completion, records the lab as complete in the database

Test-case Identifier: TC2 **Use Case Tested:** UC2, main success scenario (place item properly) **Pass/fail Criteria:** Successfully places object over proper area **Input Data:** Object to be moved	
Test Procedure	**Expected Result**

Step 1: Selects unmovable object Step 2: Selects desired object that the student wishes to move Step 3:Place Object in proper place	System displays nothing, doesnt allow user to grab unmovable object System displays the object is “grabbed” and movable through cursor movements System registers proper placement of object and continues with lab sequence

Test-case Identifier: TC3 **Use Case Tested:** UC3, main success scenario (cut item) **Pass/fail Criteria:** Successfully cuts object at the rights pot for the lap report **Input Data:** Knife to cut object	
Test Procedure	**Expected Result**
Step 1: Place the knife over desired area of object you want to cut and hold down on mouse button Step 2:Drag knife across desired object to cut Step 3: Use newly cut piece for further lab use	System shows the knife go into ‘cut mode’ System shows knife movement cutting object System will either accept the newly cut piece or pop up with an error stating that the piece is too small for use

Test-case Identifier: TC4 **Use Case Tested:** UC4, main success scenario (set hot plate), UC2 **Pass/fail Criteria:** Select proper time and temperature on hot plate for item **Input Data:** Time and temperature for hot plate	
Test Procedure	**Expected Result**
Step 1: Select proper temperature for hot plate Step 2: Select proper time for hot plate to remain at said temperature Step 3: move object onto hot plate for heating	System shows the hot plate’s new temperature System displays a timer for hot plate System shows interaction between hotplate and object being heated

Test-case Identifier: TC5 **Use Case Tested:** UC5, main success scenario (microscope), UC2 **Pass/fail Criteria:** Able to zoom in/out of an object	

placed on microscope slide **Input Data:** Object on slide, zoom parameters	
Test Procedure	**Expected Result**
Step 1: Use zoom buttons on microscope Step 2: Place object on to available microscope slide Step 3: Use zoom buttons to either zoom in or out of the object	System pops up with error, stating there is nothing on the slide System shows object on lab slide System adjusts magnification of object on slide

Test-case Identifier: TC6 **Use Case Tested:** UC6, main success scenario (pour item) **Pass/fail Criteria:** Pouring contents from one object (beaker, test tube) to another (test tube, slide, pH paper) **Input Data:** Contents to be poured	
Test Procedure	**Expected Result**
Step 1: Select appropriate container with the proper liquid contents Step 2: Attempt to pour container over incorrect object or non-interacting object Step 3: Move and pour contents from container over appropriate object	System registered control of container System will either register pouring into incorrect object (displaying an error message afterwards) or not display anything if user attempts to pour over non-interacting object System updates the new ‘wet’ object and continues with lab sequence

Test-case Identifier: TC7 **Use Case Tested:** UC7, main success scenario (spec 20 calibration), UC2 **Pass/fail Criteria:** Placing slide in Spec20 and adjusting the settings on the spec20 for the lab **Input Data:** Spec20 settings, slide	
Test Procedure	**Expected Result**
Step 1: Place slide into spec20 holder Step 2: Adjust the light/dark parameter of the spec 20 Step 3: Adjust the wavelength and read absorbance of spec 20	System show the slide in spec20 holder System displays the light/dark setting to user System displays the wavelength and read absorbance setting to user

Test-case Identifier: TC8 **Use Case Tested:** UC8, main success scenario (label) **Pass/fail Criteria:** Add label to desired object **Input Data:** Label	
Test Procedure	**Expected Result**
Step 1: Select label button Step 2: Attempt to label something which cannot have a label Step 3: Attempt to label something which can have a label	System acknowledges label button selected System pops up with an error; informs user that said item cannot have a label System places label over object, has user write desired information on label

Test-case Identifier: TC9 **Use Case Tested:** UC9, main success scenario **Pass/fail Criteria:** selects a Quiz which has not been completed yet but whose Lab has been **Input Data:** Quiz Selection	
Test Procedure	**Expected Result**
Step 1: Selects a quiz that the student has already completed Step 2: Selects a quiz associated with a lab which the student has not yet completed Step 3: Selects a quiz which the student has not yet taken that is associated with a lab which the student has successfully completed	System pops up with an error; informs the student that they have already taken the selected quiz System pops up with an error; informs the student that they have not yet completed the associated lab System sends student to the appropriate quiz page; presents the student with a randomized quiz; upon completion, grades the quiz; stores quiz results in the database; records that the student has taken the quiz

Test-case Identifier: TC10 **Use Case Tested:** UC10, main success scenario (Instructor) **Pass/fail** Criteria: Enters in a valid NetID for a student within one of their sections. **Input Data:** Student NetID	
Test Procedure	**Expected Result**

Step 1: Fail to select a NetID	System pops up with an error; informs the instructor that a NetID was not selected
Step 2: Selects a valid NetID of a student in another instructor's section	System pops up with an error; informs the instructor that the specified student is not in their section
Step 3: Enters a valid NetID of a student in one of the instructor's sections	System associates sends instructor to the specified student's gradebook page

Test-case Identifier: TC11 **Use Case Tested:** UC11, main success scenario **Pass/fail Criteria:** The test passes if the student enters in a valid NetID, username, and password **Input Data:** Student NetID, student username, student password	
Test Proceedure	**Expected Result**
Step 1: Fail to enter data in all every field	System pops up with an error; informs the student that all three fields are required
Step 2: Enter data in to each field with an invalid username or password (not alphanumeric between 7-12 characters)	System pops up with an error; informs the student that they have entered an invalid username or password
Step 3: Enter data in to each field with a NetID which does not match with any in the database	System pops up with an error; informs the student that their NetID is not in the database
Step 4: Enter data in to each field with a NetID which has already been registered	System pops up with an error; informs the student that their NetID has already been registered; gives the student the e-mail address of the instructor associated with their section
Step 5: Enter data in to each field with a valid NetID which has not yet been registered	System associates username and password with specified NetID; denotes student as registered; sends the student to a page confirming that the registration process has been completed successfully; logs student in

Test-case Identifier: TC12 **Use Case Tested:** UC12, main success scenario **Pass/fail Criteria:** The test passes if the instructor enters an acceptable number of students [1-1000] to an acceptable section (not previously created)	

Input Data: Student NetID(s), section number	
Test Procedure	**Expected Result**
Step 1: Enter an invalid number of students to add. (less than 1 or greater than 1000) Step 2: Enter in a valid number of students to add with an invalid section number. Step 3: Enter in a valid number of students to add and a valid section number.	System pops up with an error; informs the Instructor that the number of students entered is invalid System pops up with an error; informs the Instructor that they have entered an invalid section number System adds students and their associated section number to the database; sends the instructor to a page confirming that the students have been added

Test-case Identifier: TC13 **Use Case Tested:** UC13, main success scenario **Pass/fail Criteria:** Selects a valid section that is associated with the instructor **Input Data:** Section number	
Test Procedure	**Expected Result**
Step 1: Fail to enter a valid section number Step 2: Enters a valid section number associated with another instructor Step 3: Enters a valid section number associated with the instructor	System pops up with an error; informs the instructor that the section number is not valid System pops up with an error; informs the instructor that the specified section number is associated with another instructor System associates sends instructor to the specified section's gradebook page

The tests we will be describing in this section relate to what we feel are the most important and ubiquitous functions inherent to our software which are pivotal for the main successes of our use cases. Some of these test cases were enumerated on our first report, and will be further expanded upon in this report.

13.1 Unit Testing Test Cases

All tests will be initiated using a systematic approach, wherein multiple unique input values will be selected with each value engendering a different response from the software. This way, we will clearly be able to decipher how our software operates under normal (pass) and faulty (fail) input operation. I will list every important use case/submodule in this section, and then elaborate on the test to be used or elucidate as to why no test is needed.

PourBeaker/TestTube - This test will consist of affirming that moving a beaker or test tube over an appropriate object will empty the contents of the beaker/test tube onto or into the appropriate object.

Input	Action	Output
-Beaker/test tube full	Move beaker/test tube over appropriate object	Success: object now contains contents of beaker/test tube
-Beaker/test tube not full	Move beaker/test tube over appropriate object	Failure: beaker /test tube does not have any contents
-Beaker/test tube full	Move beaker/test tube over inappropriate object	Failure: Object does not interact with beaker/test tube

testpH - This test will consist of affirming that moving a piece of pH paper into an associated liquid will result in a change of color for the pH paper and be an accurate representation of the liquid being tested.

Input	Action	Output
-clean pH paper	Move pH paper over acid or base	Success: pH paper changes color reflecting acid or base
-already used pH paper	move pH paper over acid/base	Failure: pH paper already used
-clean pH paper	move pH paper over non liquid	Failure: pH paper needs to be testing in an associated liquid

calibrateSpec20 - This doesn’t need testing, as it only requires setting the settings of the Spec20. Obviously the output of the Spec20 needs to match the input, but that’s the only area of error which exists and can easily be identified/fixed during the lab.

ReadAbsorbance - This test will consist of confirming that the contents being tested in the spec20 are reading the correct absorbency.

Input	Action	Output
-full test tube in spec20	Read absorbancy	Success: Spec20 reads off correct absorbancy for chemical in test tube
-improper test tube/test tube not full	Read absorbancy	Failure: test tube does not have any contents and/or invalid test tube
-full test tube in spec20, wrong settings	Read absorbancy	Failure: incorrect or no absorbancy due to bad parameters

Create LabSpace - This does not need a test, again because of it's simplicity in the output for the given input. When a user requests for a new lab, the createLabSpace should always give him the correct lab with the correct materials. Failure of this will be seen immediately and able to be fixed without any extra testing needed.

Karyotype Identifier - This test will consist of confirming that the shown chromosome mutations have the appropriate disease name attached to it.

Input	Action	Output
-Chromosome mutation	Match up appropriate mutation name	Success: Correct chromosome mutation name
-Chromosome mutation	Match up incorrect mutation name	Failure: Incorrect chromosome mutation

RegisterClass - Test will consist of multiple different inputs and actions to the RegisterClass function to ensure proper creation. Possible tests with outcomes include:

Input	Action	Output
-Instructor login	Register new Class	Success: new class registered
-Student login	Register new Class	Failure: User doesn't have proper credentials
-Instructor login	Register existing Class	Failure: Class already exists

RegisterStudent - Test will consist of different attempts of registering students to a Bio section. Possible tests with outcomes include:

ASSUMED: STUDENT LOGIN

Input	Action	Output
-Correct section, non-full section, and NetID	Register Student	Success: You are now registered
-nonexisting section and NetID	Register Student	Failure: Section does not exist
- Correct section, but improper NetID	Register Student	Failure: Student is not in course
-Correct section and NetID, full section	Register Student	Failure: Section is full

ViewStudentResult - This function doesn't need a test, as it's behavior is very one dimensional and all that's important is that the information comes up when requested. Any fail case for this function would be a graphical bug in the interface/program which wouldn't need external testing.

ViewClassResult - Similar to ViewStudentResult, this function is very one dimensional and should only be viewable to the instructor. If the ViewClassResult graphical interface button were to appear for a student or the information were not to come up, then it would be a bug with the graphical interface which wouldn't require necessary external testing.

13.2 Test Coverage

The test coverage for this software is two fold: (1) Ensure that invalid inputs are dealt with quickly and gracefully, and (2) Ensure that there are no software coding faults which may present an invalid output or which may fail to execute correctly. In order to be certain that our software does what is expected of it, we have created a litany of tests which should encapsulate the majority of problems with our program. Virtually every method being tested has a combination of incorrect and correct inputs along with another combination of incorrect and correct actions to exhume any and all outputs available. We feel that these tests are sufficient (although could potentially be revised) to ensure proper operation under all inclement loads. As a brief example, let's quickly look at testpH and registerClass to discuss what is exactly covered under each.

testpH -
(1) The first of these tests is the one which leads to the success case, in which a blank piece of pH paper is used on the appropriate liquid (either acid or base) and the pH paper turns the correct color. This is the easiest and most obvious case, where everything goes smoothly and correctly
(2) The second test is a fail case, where we try to re-use a piece of pH paper on a new liquid (either acid or base). In this instance, the pH paper should maintain the same color and not change or morph colors due to the new liquid being exposed to it. This way, we guarantee that one cannot accidentally mix up the chemicals' resultant pH paper.
(3) The third test is a fail case, where we try to move a clean piece of pH paper over an inappropriate object (e.g empty test tube/beaker, spec20, microscope, heat plate, etc). In this instance, the pH paper should maintain it's color and form until properly exposed to the correct object. However, what this test does not correct is that if a user wanted to test test tube 1, but accidentally goes over test tube 2, the pH paper WILL change color reflecting the contents of test tube 2.

registerClass -
(1) the first of these tests is a pass case, where the proper credentials are presented (instructor login) and the instructor attempts to register a new section which does not already exist. Although there will be a software limitation on the number of classes which can be created, we didn't feel it to be necessary to test the cap due to realistic uses of the software which wouldn't lend itself to testing such superfluous things.
(2) the second of these tests is a fail case, where the improper credentials are presented (student login) and the button to register a class is unavailable. Obviously, a student should not possess power to create a new section, and the button should be unavailable to every individual who logs in with non-instructor credentials. This is pivotal to proper operation of our software.
(3) the third test is also a fail case, wherein by a user who logs in with the proper

credentials (instructor login) attempts to create a new section which already exists. The software will not allow an action, and present the user with an error stating "error: section already exists". This is to ensure that there are no duplicate sections being created/overwritten.

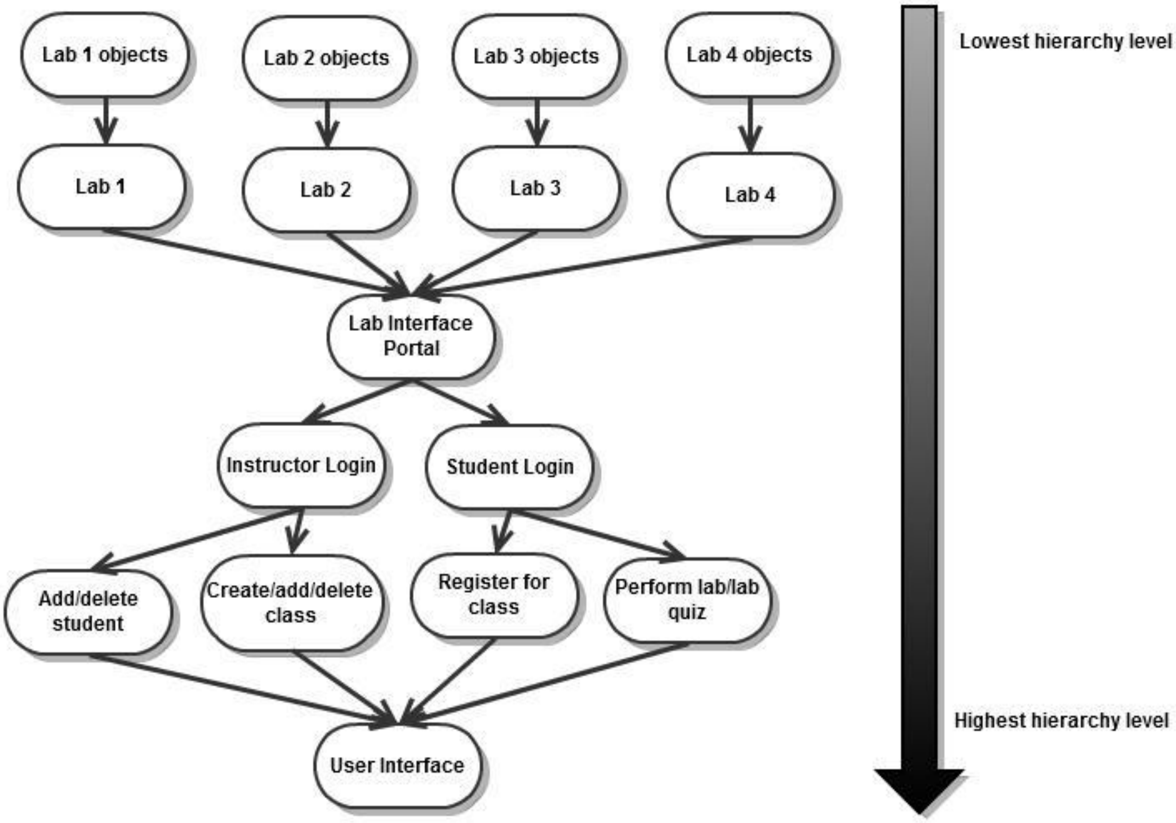
These are just two examples of the dozen or so methods which were tested, yet represent the logic and thought process which went into each method. Basically, we want strict adherence to software functionality and handling when presented with improper inputs. With those problems sorted, our software becomes much more stalwart and user friendly.

13.3 Integration Testing Strategy

Our integration testing strategy will implement the Bottom-Up integration path. This integration path requires the testing of low level, independent pieces on the hierarchy and working your way up until you reach the top piece, which interconnects all bottom pieces.

For us, the low level pieces will be the labs and the contents of the labs. We will work on each lab independently, ensuring that proper operation exists independent of everything else. Then, once the labs work sufficiently, we will begin to integrate them into our user interface, making sure that the user interface produces the appropriate lab, etc. From there, we will begin to add other user interface functionality, including the ability for students and instructors to create/register for sections and view given labs/quizzes/grades. Finally, all of it will be brought together with the user interface having the correct buttons and display for every individual part and allowing quick navigation between any two parts (e.g labs and grades).

We feel that this testing strategy is the most appropriate for our design, as lab functionality is a huge - yet independent - portion of our software. Therefore, although our strategy is somewhat bottom heavy, we can easily work our way bottom to top knowing that the lower pieces on the hierarchy are independent and correct. Here is a brief diagram summarizing everything stated:


Lab 1 objects
Lab 2 objects
Lab 3 objects
Lab 4 objects
Lab 1
Lab 2
Lab 3
Lab 4
Lab Interface Portal
Instructor Login
Student Login
Add/delete student
Create/add/delete class
Register for class
Perform lab/lab quiz
User Interface
Lowest hierarchy level
Highest hierarchy level

14. Project Management and History of Work

14.1 Merging the Contributions from Individual Team Members

To complete this project, we all created a google Document that allowed for everyone to update the same final report simultaneously. With this web document we were able to format and complete each section, upload all of the images of the use case diagrams, and leave comments on the document for any one that was reading it. This meant that every person's contribution to our report was in the same place from the beginning, and only a few formatting changes needed to be made before submitting the report just to ensure that everything was uniform and consistent.

For the reports our groups would assign sections to individual people and use all of the previous sections as a resource for finishing our own. Some members of the group were more involved with the design portions and some were more involved with the implementation. Everyone was in charge of keeping track of deadlines and choosing which section would be their focus for each report.

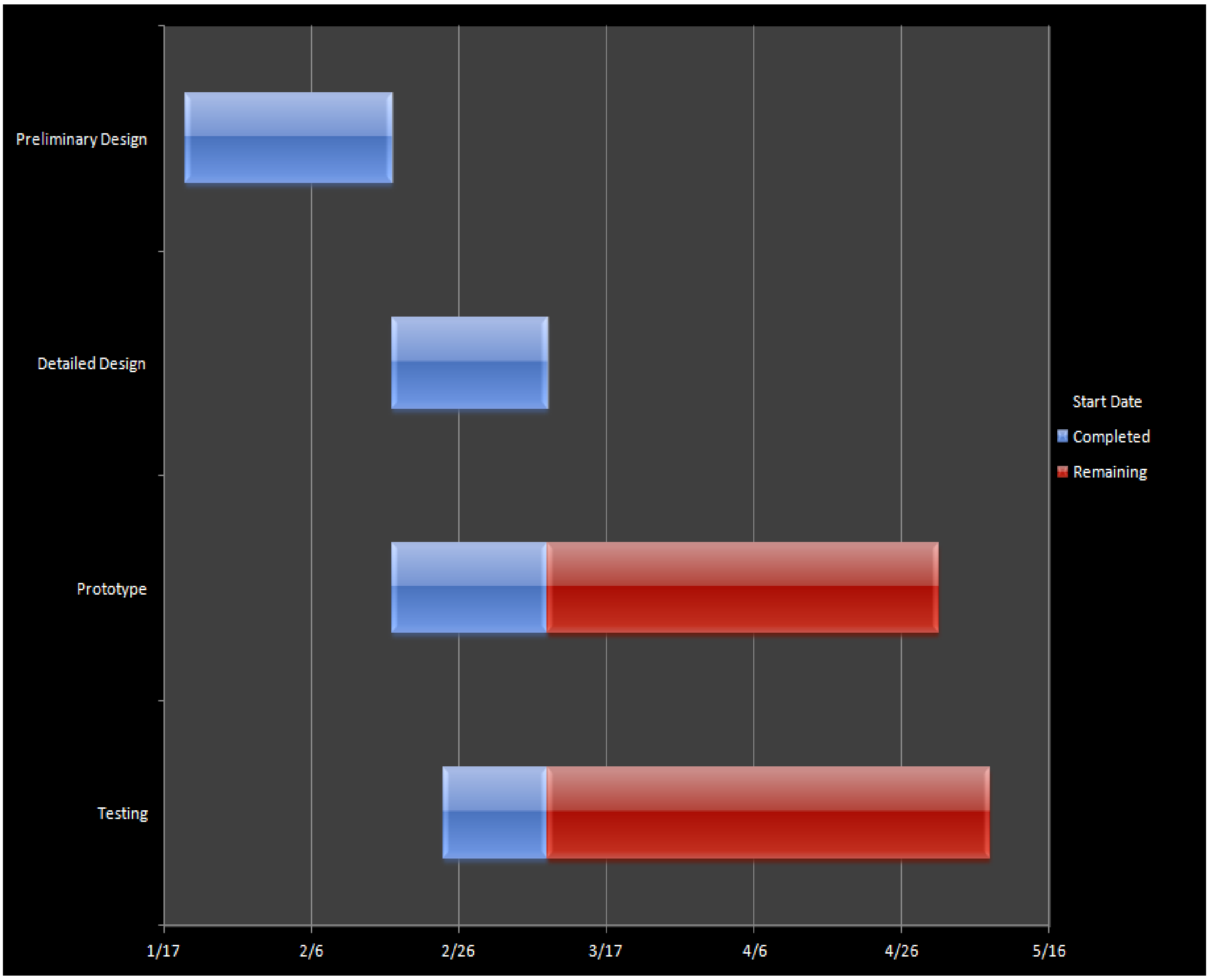
14.2 First Iteration Milestones

During the first iteration of the Virtual Biology Laboratory we placed too much importance on the complementing features and not enough on the value of the implementation of the actual virtual biology. Our initial design focused almost entirely on the logging in, creating classes, editing students and viewing the grade book. We did make some small changes during report one before we handed in the final copy. The first report was completed on time with our timeline but we knew that it would need a lot of changes for when it needed to be resubmitted for report three.

The list below shows the deadlines that our group set for ourselves at the time of the first report. Since the second report was due in parts, we did not follow this list exactly. It does show that our goals was to complete the reports and the demo with enough time to either send in portions for feedback or to go to office hours and ask for advice on how to better our design.

Fri Jan 20	All day	E-mail Professor Marsic with group members & project selection
Wed Jan 25	All day	Submit proposal for feedback
Fri Jan 27	All day	PROPOSAL DUE
Fri Feb 10	All day	submit report 1 rough draft for feedback
Wed Feb 15	All day	Submit revised report 1 for feedback
Fri Feb 17	All day	FIRST REPORT DUE (SPECIFICATION ONLY)
Fri Mar 2	All day	Submit report 2 rough draft for review
Wed Mar 7	All day	Submit revised report 2 for feedback
Fri Mar 9	All day	SECOND REPORT DUE (DESIGN ONLY)
Tue Mar 20	All day	Submit first demo for feedback
Fri Mar 23	All day	Submit revised first demo for feedback.
Tue Mar 27	All day	FIRST DEMO DUE
Fri Apr 13	All day	Compile third report

This second image shows the chart we constructed to show the overlap of the more detailed design, creating the prototype and testing. For the design, report two was handed in in three parts. We submitted all the parts of the report separately and once we received feedback, we changed it for the final submission of report two. The problems with the second report were similar to the problems with the first; we did not focus enough on the biology. An example of this is our interaction diagrams. Originally there only existed interaction diagrams for creating a student profile, viewing a grade book or for logging in. When the final report two was submitted, we added interaction diagrams that expressed how the system acted during the actual laboratory.

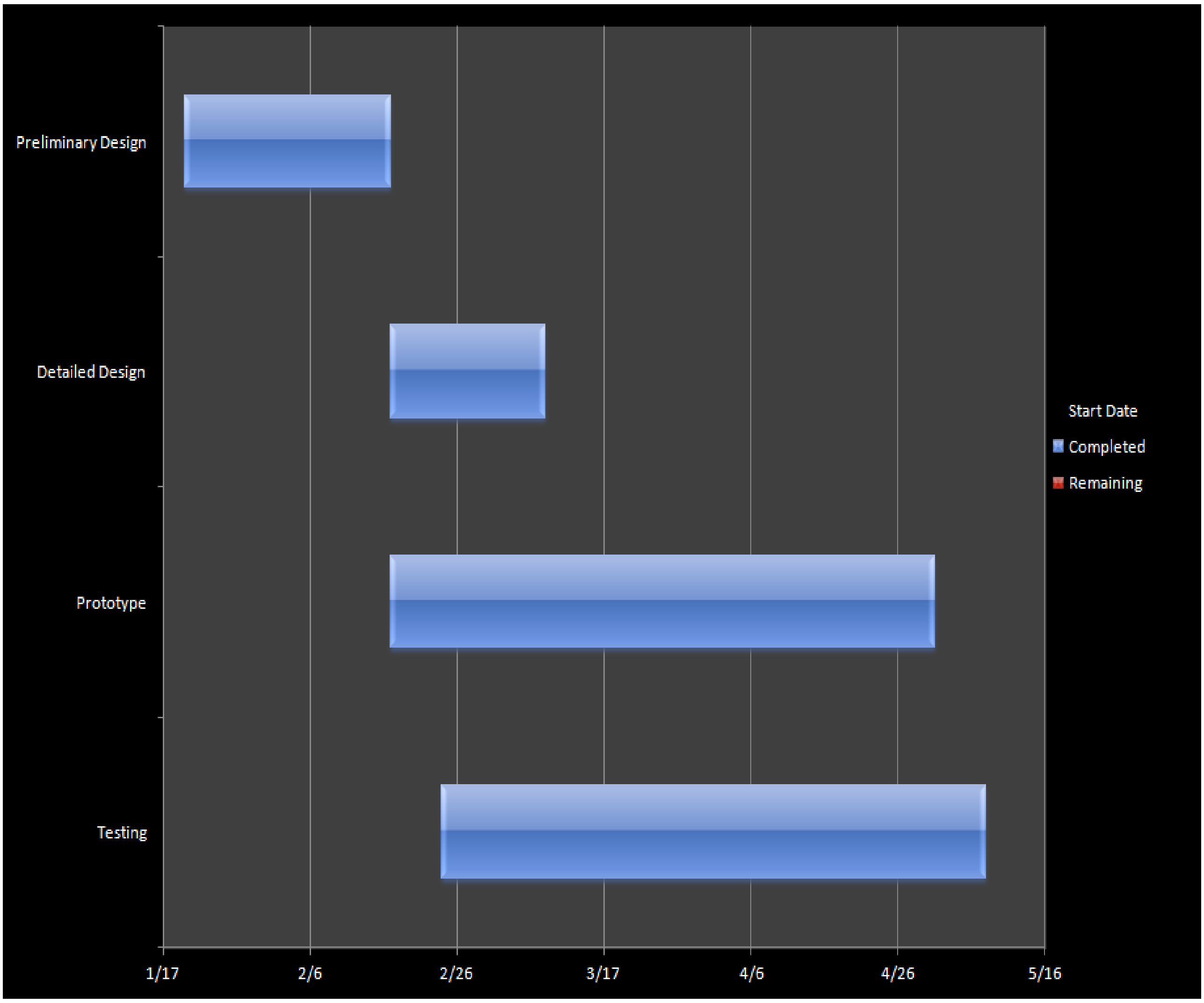


In demo one, we completed setting up our web environment and our framework built on top of PHP. We then completed most of the visual aspects of our web-end such as the login boxes, create account boxes and the menu bar. We also completed aspects such as the lab quizzes themselves, including the grading functions towards them. We were not able to complete all labs yet, but were able to complete most of the visual components of them (they weren't fluidly glued together yet). With the mentioned features, we were able to give a step by step view of the Virtual Biology lab and how it would look.

14.3 Second Iteration Milestones

For the third report, the second iteration of our project, there were many changes made to our design. First, we doubled the number of use cases, and adjusted all of the sections after that. Since there were new use cases, all pertaining to the interactions between the students and the system in the actual virtual laboratory, we had to add new casual descriptions, fully dressed descriptions, use case diagram, and sequence diagrams. To keep the report more focused on the biology we also changed the domain and class diagrams to show both the implementation of the laboratory and the extra features.

Since we had received detailed feedback on the mistakes made in the first two reports, a lot of time was spent making those small corrections to the paper. We labeled more of our diagrams, wrote more detailed descriptions about each image and traceability matrix, and made sure to use consistent names for each use case, domain concept and class. Since we knew that there were so many changes to be made, we began working on the third report the Monday after the second report was submitted.



This gantt chart shows the completion with our project as per our redefined goals for the final submission. Some of the elements that we had initially intended on implementing were not able to come into existence and were ultimately scratched from the project in order to put more emphasis into other areas of design and testing.

In demo two, we began to create more aspects of the website and the actual labs themselves. For example, the website now incorporates a database to store users, sections, grades and more. The actual labs themselves are also more fully incorporated and are actually usable. Though our final product may seem fluid, bugs may exist due the nature of agile development in our project.

By this stage of development, the system's design had been revised and fleshed out to the point that our group could start surveying possible methods of implementation. Prior to settling on Adobe Flash, we compared several similar products to find out which animation software would best suit our particular needs. Upon researching these developmental tools, our group ended up with two options: Adobe Edge, a new HTML5 multimedia platform or Adobe Flash, the industry standard for computer animation for the past decade. In the end, we chose to use Adobe Flash. Although it is not as compatible with certain devices and operating systems, it has a very reliable track record compared to Adobe Edge which was in beta at the time.

Although it is the most common computer animation software used today, Adobe Flash requires the purchase of a license for each computer it is installed on which can cost up to $700. In addition to this, software used for the creation of vector based graphics (Adobe Illustrator) can cost upwards of $600. Luckily, one member of our team had access to a computer with this software installed on it which is what was used to create the animation modules shown in the demos. Due to the constraints of the course as well as the limited resources available to us, we chose to choose interesting parts of each lab to create and show in the demo. Given more time and resources, every one of the labs could be created similarly according to the lab descriptions outlined througout the report.

14.4 Future Work

There are many features that can be added to the Virtual Biology Laboratory. In addition to the four Laboratories discussed in this report, there are an additional six laboratories that exist in the Rutgers Lab Manual. Most of these labs are about plants and photosynthesis, if someone were interested they could use the information provided about these laboratories and implement the other six labs. The structure of the cell would only need to change a little bit because a plant cell's wall is a lot more structured than an animal cell's wall, but the machines like the microscope and the hot plate can be totally reused.

Our next course of action, if we were to continue this project, would be to find an interested biology professor here at Rutgers who would be interested in acting as a consultant in perfecting our Virtual Biology Laboratory. We would check with him to make sure that each of our labs were biologically correct and placed importance on the correct concepts. We would also ask him or her for more quiz questions for each subject. Next we would implement the other laboratories in the manual or make up new laboratories to compliment the introduction to biology course. The professor would also supply us with supporting material to post on the website for the students.

Another addition that could be made to the Virtual Biology Laboratory is to implement the code in a different language than flash. This would allow the students to complete their assigned laboratories on their tablets of cellular phones. Enabling the students to perform the laboratories on their handheld devices will also offer the professor an opportunity to include the use of parts of the Virtual Biology Laboratory in the classroom. We could potentially add a feature that allows a student to log on and answer a question or insert a code given by the professor to count as their attendance for that days class.

15. References

Campbell, Neil, Jane Reece, Lisa Urry, Michael Cain, Steven Wasserman, Peter Minorsky, and Robert Jackson. *Biology*. 2nd Edition. Volume 1. San Francisco: Pearson, 2008.

General Biology 01:119:101- 2011 Laboratory Manual

Marsic, Ivan. "Software Engineering.", 16/01/2012. 1 Feb 2012. <http://www.ece.rutgers.edu/~marsic/books/SE/book-SE_marsic.pdf>.

Solomon, Eldra, Linda Berg, and Diana Martin. *Biology*. 8th Edition. Volume 2. Belmont: Thomson Brooks/Cole, 2008.

Fig 2.1 http://media.web.britannica.com/eb-media/95/72195-035-F4E0640E.jpg

Fig 2.2 http://www.chemistry.nmsu.edu/Instrumentation/spec20_dig.gif

Fig 2.3 http://kenpitts.net/hbio/8cell_repro/meiosis2.gif

Fig 2.4 http://wiki-images.enotes.com/thumb/5/53/NHGRI_human_male_karyotype.png/250px-NHGRI_human_male_karyotype.png

Fig 3.1 http://www.beyondbooks.com/lif71/images/00046824.jpg

Fig 3.2 http://www.microscope-microscope.org/activities/images/prep-slides-1.jpg

Adobe Product Buying Guide, 4/30/2012 <http://www.adobe.com/products/flash/buying-guide.html>