Spectrophotometer

Bio Lab Part #2

Bingbing Xu, Chao Han, Junwei Zhao, Xueyuan Song

Content

1. Customer Statement of Requirements ... 7
1.1 Problems ... 7
1.2 A Virtual Spectrophotometer ... 8
1.2.1 What Is the Spectrophotometer? ... 8
1.2.2 How does A Spectrophotometer work? ... 9
1.3 Problems Analysis ... 11
1.4 Objective of our design ... 11
2. Glossary of Terms ... 13
3. System Requirements ... 14
3.1 Enumerated Functional Requirements ... 14
3.2 Enumerated Nonfunctional Requirements ... 15
3.3 Acceptance Tests ... 16
3.4 On-Screen Appearance Requirements ... 17
4. Functional Requirements Specification ... 19
4.1 Stakeholder ... 19
4.2 Actors and Goals ... 19
4.3 Casual Description ... 19
4.4 Use Case Diagram ... 20
4.5 Tracebility Matrix ... 21
4.6 Fully-Dressed Description ... 21
4.6.1 UC-1 Switch-On ... 21
4.6.2 UC-2 AdjustBlank ... 22
4.6.3 UC-3 AdjustZero ... 22
4.6.4 UC-4 SelectWaveLength ... 23
4.6.5 UpdateInfoBoard ... 23
4.6.6 MeasureTubeValue ... 23
4.6.7 SetSampleValue ... 24
4.6.8 UC-8 Login ... 24
4.7 System Sequence Diagrams ... 25
5. Effort Estimation using User Case Points ... 30

5.1 Unadjusted Use Case Points (UUCPs) 30
5.2 Technical Complexity Factor(TCF) 31
5.3 Environment Complexity Factor (ECF) 32
6. User Interface Specification 33
6.1 Preliminary Design 34
6.1.1 Teachers' interface 35
6.1.2 Students' interface 35
6.2 User Effort Estimation 36
7. Domain Analysis 37
7.1 Domain Model 37
7.1.1 Concepts definition 37
7.1.2 Association Definitions 39
7.1.3 Attributes definition 40
7.1.4 Traceability Matrix 41
7.2 System Operation Contracts 41
8. Interaction Diagrams 43
8.1 ID1 SwitchON 43
8.2 ID2 AdjustZero 44
8.3 ID3 SelectWaveLength 47
8.4 ID4 NextButton 49
8.5 ID5 BackButton 50
8.6 ID6 OpenLid 50
8.7 ID7 InsertTube 52
8.8 ID8 CloseLid 54
8.9 ID9 AdjustBlank 55
8.10 ID10 OpenLid 57
8.11 ID11 RemoveTube 58
8.12 ID12 CloseLid 59
9. Class Diagram and Interface Specification 61
9.1 Class Diagram 61
9.2 Data Types and Operation Signatures 63
9.2.1 Account 63

9.2.2 Login 64
9.2.3 AccountStorage 64
9.2.4 AccountChecker 64
9.2.5 TeacherOperationController 64
9.2.6 LabObject 65
9.2.7 PowerSwitch 66
9.2.8 IndicateLamp 66
9.2.9 Dial 66
9.2.10 TestTube 67
9.2.11 WaveDial 67
9.2.12 Needle 68
9.2.13 Lid 68
9.2.14 SampleHolder 68
9.2.15 SpectroState 69
10. System Architecture and System Design 71
10.1 Architectural Styles 71
10.2 IdentifyingSubsystems 72
10.3 Mapping Subsystems to Hardware 72
10.4 Persistent Data Storage 73
10.5 Hardware Requirements 74
11. User Interface Design and Implementation 75
11.1 GUI overview 75
11.1.1 Power Switch 76
11.1.2Adjust Zero Control Dial 77
11.1.3 Wavelength Selection 77
11.1.4 Blank adjustment 77
11.2 GUI Design 78
11.2.1 The use of cross-platform color 78
11.2.2 The design of application graphics 78
11.2.3 The use of graphics for product identity. 79
11.3 Ease-of-use Design 79
11.4 Navigation Path 79

12. Design of test 81
12.1 Use Cases will be tested 81
12.2 Unit Testing test cases 85
12.3 Test Coverage 87
12.4 Integration Test Strategy 88
13. History of work 89
13.1 Problems and Progress Report 89
13.2 Plan of Work 90
Reference 91
Summary of Changes 93

1. Customer Statement of Requirements

The spectrophotometer is a machine used to measure the concentration of a substance in a solution by passing light of a specified wavelength through it. A photocell at one end receives the transmitted light and an analog meter displays the percent transmittance of light received by the photocell. Many substances absorb light and transmit light of specific wavelengths within the ultraviolet (200 – 400 nm), visible (400 – 700 nm), and near-infrared (700 – 1000 nm) regions of the electromagnetic spectrum. According to Beer's law, the amount of light absorbed by a medium is proportional to the concentration of the absorbing material or solute present. In order to use this instrument, the correct wavelength of light must be chosen, so that the light is absorbed by the substance contained in a test tube when it tries to pass through the solution (the more substance there is, the less light will pass through and vice versa). The main purpose of the lab is to familiarize students with the spectrophotometer and its use. The students first calibrate the instrument using a reference solution of known concentration. The concentration of unknown samples is calculated according to the transmittance value of the reference solution.

In a typical measurement of the absorbance of light, a beam of light is passed through a transparent tube containing the absorbing material. The intensity I of light reaching a detector is compared to the intensity Io reaching the detector in the absence of the absorbing material. The relationship between the absorbance A, the concentration c of the absorbing substance, and the path length ℓ of the light through the substance is given by the Beer-Lambert law:

$$A = e\,\ell\,c$$

where e, called the molar absorption, is a fundamental property of the absorbing material.

The key point is that the absorbance (defined as log [Io/I]) is proportional to concentration.

In reflecting on the usefulness of the spectrophotometer simulation, the students enjoyed the individual and repetitive practice they could engage in. They envisioned that this experience would help them with their lab practical exams as well as fine-tune their skills in operating the spectrophotometer in the actual lab.

1.1 Problems

Biology is a scientific field which is closely related to experiments. However, a traditional biology class in high school or college always faces to the following problems, such as:

1. Experimental devices and resources are always limited. Students, who want to do experiments in Laboratory, may need to make reservations in advance. Therefore, it's not an easy opportunity to do a bio experiment in laboratory.
2. Some bio-processes or bio experiments take relatively long time, and nobody can speed it up in the real world. For instance, some machines, before using them, take an hour to be

warmed up, and take another hour to get result. For some simple experiment, it's kind of waste of time. However, without knowing how to operate the machine, students may feel confused.

3. For some experiments, a small mistake may make the whole experiment incorrect. The students cannot fix the mistake even they know they have done something wrong. The only way to get a correct result is to do it again.
4. Sometimes, before doing an experiment, a student needs to prepare for it. For example, he needs to get familiar with how to use the experimental machine. In most cases, however, an instruction book is far away from enough.
5. It is difficult to test the students' experiments skills, because a bio experiment includes many steps. A correct result cannot guarantee all steps are correct. However, it is unrealistic that a teacher examines all steps of each student.
6. Physical machines always require constant maintenance. For some types of machines, their components are not manufactured anymore. In that case, they cannot be used, which is sort of waste of money.
7. For students in developing countries, or in universities that do not have sufficient fund to build labs, they lose the opportunity to do experiments.

If problems mentioned above are solved, students, teachers and educational institutes can get lots of benefits. Since computers are widespread, a computer application that simulates a machine is the simplest and cheapest way.

1.2 A Virtual Spectrophotometer

We propose to develop a specific virtual biology machine named spectrophotometer which can be used in biology teaching.

1.2.1 What Is the Spectrophotometer?

Figure1. 1 A spectrophotometer

The spectrophotometer (Figure 1.1) is an instrument which measures the amount of light of a specific wavelength which passes through a medium. Many substances absorb light and transmit light of wavelengths within the ultraviolet (200 - 400 nm), visible (400 - 700 nm) and near-infrared (700 - 1000 nm) regions of the electromagnetic spectrum. The extent to which a sample absorbs light depends strongly upon the wavelength of light. The absorbance spectrum(Figure 1.2) shows how the absorbance of light depends upon the wavelength of the light. The spectrum itself is a plot of absorbance vs wavelength and is characterized by the transmit light of wavelengths within the ultraviolet (200 - 400 nm), visible (400 - 700 nm) and near-infrared (700 - 1000 nm) regions of the electromagnetic spectrum. The extent to which a sample absorbs light depends strongly upon the wavelength of light. The absorbance spectrum(Figure 1-2) shows how the absorbance of light depends upon the wavelength of the light. The spectrum itself is a plot of absorbance vs wavelength and is characterized by the wavelength (λ_{max}) at which the absorbance is the greatest.

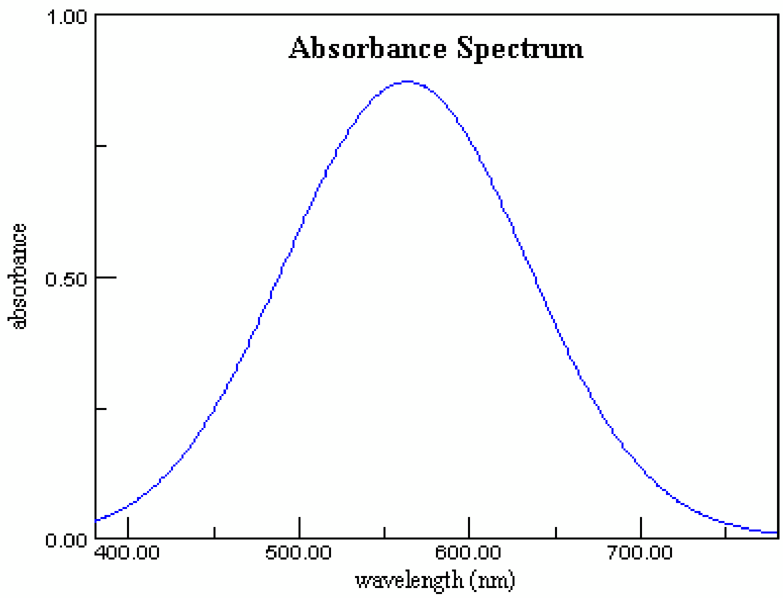


Figure1. 2 Absorbance Spectrum

According to Beer's law, the amount of light absorbed by a medium is proportional to the concentration of the absorbing material or solute present. Thus the concentration of a solute in a solution may be determined by measuring the absorbency of light at a given wavelength (λ_{max}).

In biology laboratories, for instance, the spectrophotometers are used to measure the concentrations of DNA, RNA or protein samples, λ_{max} of which are already known.

1.2.2 How does A Spectrophotometer work?

A spectrophotometer should include the following components (Figure 1-3):

- A light source
- A focusing device that transmits an intense straight beam of light
- A monochromator to separate the beam of light into its component wavelengths
- A device for selecting the desired wavelength

- A sample holder
- A photoelectric detector
- A meter to display the output of the detector

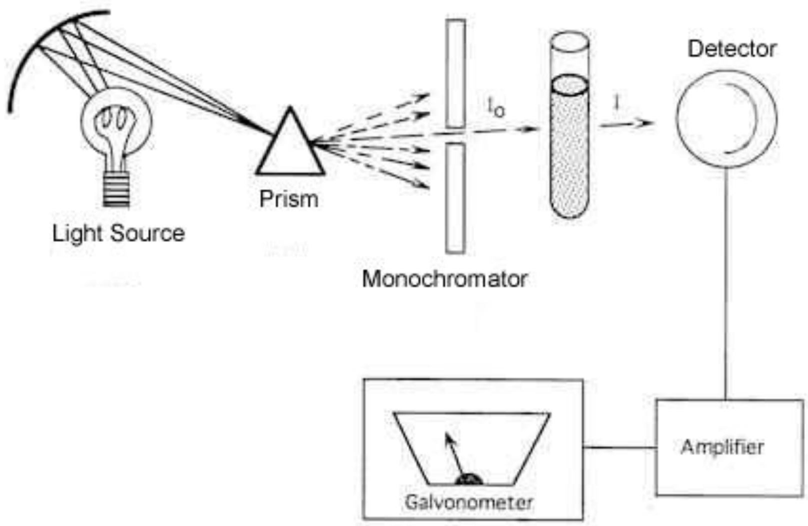


Figure1. 3 Machanism of the Spectrophotometer

The basic experimental procedure is as following:

- The intensity of light (I_0) passing through a blank is measured. The blank is a solution that is identical to the sample solution except that the blank does not contain the solute that absorbs light.
- The intensity of light (I) passing through the sample solution is measured.
- The experimental data is used to calculate two quantities: the **transmittance** (T) and the **absorbance** (A), as Figure 1-4 shows

$$T = \frac{I}{I_0}$$

$$A = -\log_{10} T$$

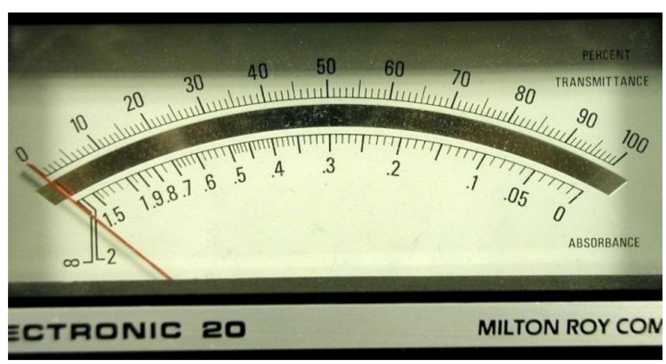


Figure1. 4 meter panel (upper line is Transmittance, bottom line is Absorbance)

• The concentration of an unknown solution can be determined by comparing the absorbance reading of the unknown to the standard curve.

The experimental result may be affected by the following factors:

• The operator doesn't choose the proper wavelength.
• The operator doesn't set a reference solution correctly.
• Unwanted light leaks into the sample.

1.3 Problems Analysis

Our design, a virtual lab, can easily solve the problems mentioned in Section 1.1.

From the viewpoint of cost, on the market, a spectrophotometer cost at least $1000, which causes limited laboratorial resources and devices. However, software can run on any computer, which means, taking almost everyone owns a computer, it is almost 0-cost if the software is free. Moreover, it doesn't need maintenance, and it needs fewer teachers in laboratory, which saves maintenance costs and human costs, especially, for institutes that has insufficient fund.

From the students' and teachers' point of view, it can make the experiment easier and save a lot of time. They don't need to go to Laboratory to do bio experiments. Students can preview or review haw to operate the bio machine in term of running the application on their computer. Obviously, it is more convenient. Also, software can implement lots of functions which are impossible in the real world. For example, most applications have functions like "undo" and "redo". And the process that takes one hour in the real world may finish in 1 second in the software.

Of course, there are many free so called virtual bio labs on the internet. However, they all have the several flaws such as lack of accuracy, no animation and only description and so on. Plus, the experiment using spectrophotometer cannot be found on the internet. After our design is accomplished, it will be free because of educational use. Therefore, it can be used widely, and thousands of students, teachers and institute will get benefits.

1.4 Objective of our design

The main purpose of the lab is to familiarize students with the spectrophotometer and its use. Generally speaking, the students first calibrate the instrument using a reference solution of known concentration. The concentration of unknown samples is calculated according to the transmittance value of this solution.

In reflecting on the usefulness of the spectrophotometer simulation, the students enjoyed the individual and repetitive practice they could engage in. They envisioned that this experience would help them with their lab practical exams as well as fine-tune their skills in operating the spectrophotometer in the actual lab. Overall, the students found the virtual labs to be an interface where they could learn and practice in spite of making errors. They also acknowledged that the simulations were realistic and helped in demonstrating certain processes that were not easily represented in the actual lab.

2. Glossary of Terms

Table 2.1 shows the terms used in this project.

Table 2. 1 Clossary of Terms

Term	Comments
Absorbance	The intensity of light at a specific wavelength that passes through a sample.
Blank/Transparent	The sample used to calibrate the zero end of the absorbance scale.
Blank Dial	Located on the left front side of the Spectrophotometer, used to set the meter reading to an infinite absorbance on the left side, used for very dark sample where no light passes through.
Concentration	The abundance of a constituent divided by the total volume of a mixture.
Light Production	Light of whichever wavelength that is set shone from the spectrophotometer through the sample.
Sample	The solution that needs to be measured.
Sample Holder	The slot on the top left side of the spectrophotometer that is where you place the test tube with the liquid that you want to find the absorbance of.
Spectrophotometer	A machine used to find the absorbance of a sample by shining a light through it of a specific wavelength.
Test Tube	Glass tube used to contain solution is open at the top and U shaped on the bottom.
Transmittance	The fraction of light in the original beam that passes through the sample and reaches the detector.
Wavelength Dial	The top screen on the Spectrophotometer that indicates at which wavelength, in nanometers, that the light is being shone.
Zero Dial	Located on the right side of the Spectrophotometer, used to set the meter reading to zero absorbance on the right side of the meter, used for when 100% of light is transmitted.

3. System Requirements

3.1 Enumerated Functional Requirements

In biology class, a virtual spectrophotometer application not only could help students obtain knowledge, but also activate their learning interests. The application can avoid the situation that several students share one spectrophotometer. With it, it is possible that students do not have to go to laboratory. Also, biology lab could reduce the cost by using this application.

From users' point of view, we list the requirements of a virtual spectrophotometer application (as shown in Table 3.1).

In Table 3.1, each requirement is assigned a unique identifier. The middle column shows the priority weight of each requirement, with a greater number indicating a higher priority.

Table 3. 1 System Requirements of the Virtual Spectrophotometer Application

Identifier	Priority	Requirements
REQ1	4	The system should have a power switch which allows the user to switch on the spectrophotometer, and an indicator lamp to show the power state.
REQ2	5	The system should allow users to select wavelength of the transmit light
REQ3	5	The system should allow the user to adjust the needle to the zero scale in terms of rotating the zero dial.
REQ4	5	The system should allow the user to adjust the needle to the infinite scale by rotating the blank dial.
REQ5	5	The system should have a meter that has two scales, transmittance and absorbance. The meter should show the result of operations.
REQ6	4	The system should have a test tube rack where the transparent solution and samples are put.
REQ7	5	The system should have a sample holder with a lid in spectrophotometer.
REQ8	3	The system should allow students to make mistakes during operating.
REQ9	1	The system should provide information board where teacher can post information.
REQ10	2	The system should allow the teacher to change concentration values of the samples by entering different values.
REQ11	3	The system should be able to differentiate students and teachers while login.

REQ1, REQ2, REQ3, REQ4, REQ5 and REQ7 make the spectrophotometer function like a physical one. REQ6 is make the software look like a workbench. Besides the spectrophotometer, other experimental instruments, such as test tubes, also need to be placed on the table. REQ8 makes the software like a real experiment, for, in real world, students are always making mistakes. REQ9 and REQ10 are requirements raised by teachers. They need to get access to the system to modify some values and post some information, which make software useful for the reason that teachers need to protect the students from memorizing the value of each sample.

3.2 Enumerated Nonfunctional Requirements

The non-function requirements contain the following aspects.

1. Functionality:

Our system should consider a reasonable and feasible data process method. For example, the experiment result presents over the scale or the result shows a negative data.

2. Usability

The application should be easy to use and contain instructions that make first user easy to operate.

3. Reliability

The system should consider the expected incorrect operations and deal with it during programming so that the application is robust for the user.

4. Performance

The application should possess a fast response to one operation. The computing speed, resource consumption are also the design aspects of programming.

Non-functional requirements are show in Table 3.2.

Table 3. 2 Non-functional requirements

Identifier	Priority	Requirements
REQ12	2	The system should not allow the result point over the scale. If the result is smaller than the smallest scale or larger than the largest scale, the pointer just points to the smallest or largest scale.
REQ13	3	The system should make sense. For instance, if the lid is close, the user cannot put the test tube in it or remove test tube from it. Likewise, if there already has a test tube in the sample holder, the user cannot put another one into it.
REQ14	2	The system should provide tool tips to help the user distinguish the dials and display correct operating process.
REQ15	3	The system should give the scientific result of the experiment.

REQ14 makes the first-time user easier to get familiar with the spectrophotometer. REQ15 means the system should give a higher or lower value (as a physical one does) according to the incorrect order of operations.

3.3 Acceptance Tests

- Acceptance test case for REQ1:

Ensure the indicator lamp is on and user can operate when the power switch is on.

Ensure the indicator lamp is off and operations on the spectrophotometer don't make its needle move.

- Acceptance test case for REQ2:

Ensure users can select and display wavelength of the transmit light during the experiment process.

- Acceptance test case for REQ3:

Ensure the user can adjust zero of the needle with zero dial during experiment process.

- Acceptance test case for REQ4:

Ensure the user can adjust blank with the blank dial during experiment process.

- Acceptance test case for REQ5:

Ensure the meter could show the result of operations, when operating the spectrophotometer.

- Acceptance test case for REQ6:

Ensure there are transparent solution and samples on the rack.

- Acceptance test case for REQ7:

Ensure the system has a sample holder with a lid where the samples or bland should be put in spectrophotometer.

- Acceptance test case for REQ8:

Ensure the system could still operate when user don't follow the right experiment steps during the process.

- Acceptance test case for REQ9:

Ensure the system provides an information board where display the information that posted by teachers in teachers' system.

• Acceptance test case for REQ10:

Ensure the user can change the value of each solution in term of entering its value.

• Acceptance test case for REQ11:

Ensure by entering different username and password, teachers and students can enter teachers' interface and students' interface, respectively.

3.4 On-Screen Appearance Requirements

The spectrophotometer is an instrument which measures the amount of light of a specified wavelength which passes through a medium. Our virtual device should simulate every aspects of the real spectrophotometer in order to hone their skill in the actual lab. The "measuring" behavior performs the calculations based on the density of the solution contained in the test tube and the light wavelength and sends a Transform Command to the instrument (Absorbance Meter) needle to display the wavelength. A dial knob can be rotated to set the desired value.

The "turning" behavior causes the "measuring" behavior to redo the measurement when a dial is rotated. Similarly, the lid "opening" behavior causes the "measuring" behavior to redo the measurement when the sample holder's lid is opened or closed. The interface should look like Figure 3.1.

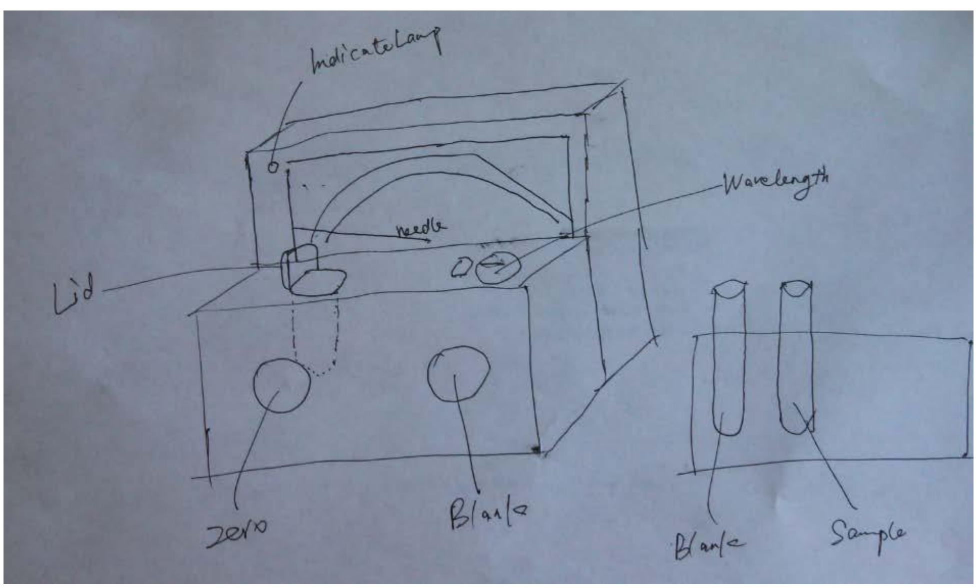


Figure3. 1 Interface Sketch

If this device is implemented so that the student is strictly guided through the process, step-by-step without a chance of making mistakes, then the system may not have much pedagogical value and may be boring to use. Because this is an instructional system, it should allow the students to demonstrate the mastery of the course material. Therefore, we should allow the student to make mistakes during the instrument operation. The students should be allowed to follow "blind alleys" until he or she realizes that they made a mistake and need to backtrack to an earlier point and continue the lab execution in the right direction. Of course, if the device usage is made too difficult, then the students may lose interest in using it. It is important to strike the right balance between a rigid linear process and a complex web of pathways and "dead alleys."

One option is to ask the students initially to set the "difficulty level" before they start the laboratory exercise. Another option is to keep track of students' progress and if the device somehow can detect that the students are not making progress and are becoming frustrated, then offer help and guide the students out of the "blind alley" where the students are currently stuck in.

For teachers who want to send message to students, the information board should be offered on our system interface. Through the login system, teachers can easily enter into the administration interface. We provide teacher an interaction with students through this interface. Teachers can use this system to upload announcements and type in the sample value. Announcements and tips would help them with their lab practical exams as well as fine-tune their skills in operating the spectrophotometer on the information board.

4. Functional Requirements Specification

4.1 Stakeholder

The stakeholders of our system are **Student** and **Teacher**.

4.2 Actors and Goals

Student: to operate the spectrophotometer, to get experimental result and to familiarize with using the machine.

Teacher: to post information and to modify some experimental data.

Meter: to display the result of the experiment.

IndicateLamp: to indicate the power state of the spectrophotometer.

4.3 Casual Description

Table 4-1 shows the casual description of all Use Cases. There are four columns in the following table. The first column is the Use Case Number, and Second and the third column are Name and casual description of each Use Case. The last column is the requirements that each use case covers.

Table 4. 1 Use Case Casual description

Use Case	Name	Description	Requirements
UC-1	**SwitchOn**	To make the machine start to work	REQ1
UC-2	**AdjustBlank**	To adjust the needle on the meter to the full scale	REQ4, REQ5, REQ8, REQ12, REQ14, REQ15
UC-3	**AdjustZero**	To adjust the needle on the meter to 0 scale	REQ3, REQ5, REQ8, REQ12, REQ14, REQ15
UC-4	**SelectWaveLength**	To select wavelength of the light	REQ2, REQ8, REQ14
UC-5	**UpdateInfoBoard**	To change the content of the information board	REQ9
UC-6	**MeasureTubeValue**	To get the value of tube	REQ7, REQ9, REQ11, REQ13, REQ14
UC-7	**SetSampleValue**	To set the sample value	REQ10
UC-8	**Login**	To allow user to login	REQ11

4.4 Use Case Diagram

Figure 4.1 and 4.2 are the use case diagram of the whole program, which show the relationship between actors and Use Cases clearly. The application is divided into two parts: Student(Figure4.1) and Teacher(Figure 4.2). In order to clearly illustrate the problem, in Figure 4.1 and 4.2, we just show two actors: Student and Teacher.

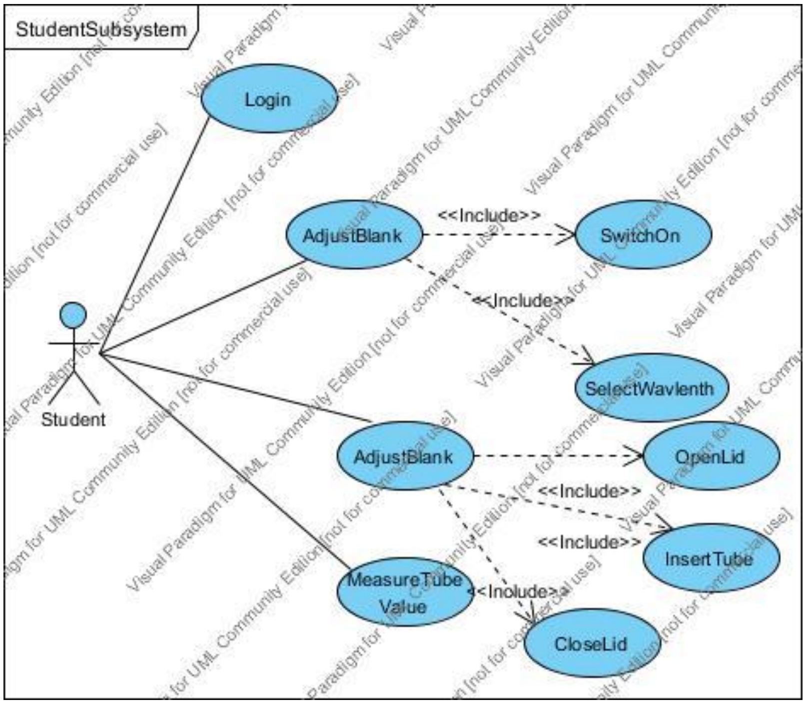


Figure 4. 1 Student Subsystem Diagram

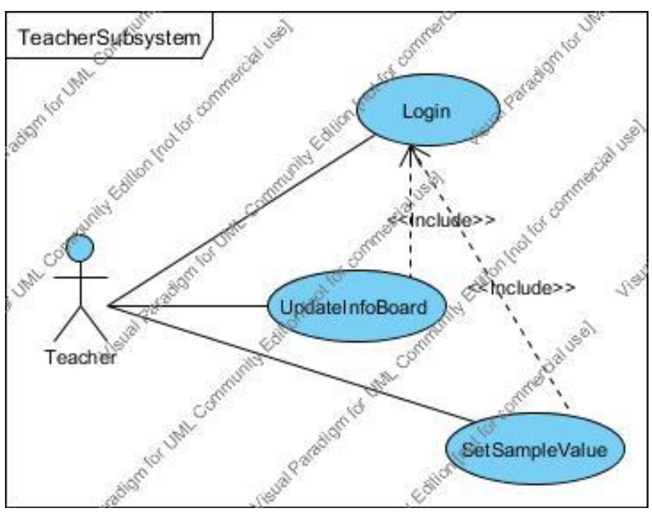


Figure 4. 2 Teacher Subsystem Diagram

4.5 Tracebility Matrix

Table 4-2 is the Traceability Matrix Table. PW is short for Priority Weight. Max PW is the highest PW of requirement. The Total PW is the sum of all requirements' PW of each Use Case.

Table 4. 2 Traceability Matrix

	UC-1	UC-2	UC-3	UC-4	UC-5	UC-6	UC-7	UC-8
REQ1	X							
REQ2				X				
REQ3			X					
REQ4		X						
REQ5		X	X					
REQ6								
REQ7						X		
REQ8		X	X	X				
REQ9					X	X		
REQ10							X	
REQ11						X		X
REQ12		X	X					
REQ13						X		
REQ14		X	X	X		X		
REQ15		X	X					
Max PW	4	5	5	5	1	5	1	1
Total PW	4	20	20	10	1	10	1	1

Note that all requirements, functional and non-functional requirement, are included in the Table 4-2, because these non-functional requirements are indispensable for this software. What we can conclude from it is that the first and most important step we need to do is to make the virtual device like a real one. It means that it must look like a real one and users can get result by operating it. Therefore, Use Cases related to teachers are less important than UCs of students.

4.6 Fully-Dressed Description

4.6.1 UC-1 Switch-On

Use Case UC-1:	SwitchOn
Related Requirements:	REQ1
Initiating Actor:	Student
Actor's Goal:	To make the spectrophotometer work, and get the power

	light on.
Participating Actors:	PowerSwith, Meter
Preconditions:	• The power light is off
	• The meter cannot move.
Postconditions:	The power light is on, and the machine starts to work.
Flow of Events for Main Success Scenario:	
→	1. **Student** turns the power switch to the State "ON".
←	2. **System** (a) signals to the **Student** the machine status, (b) signals to LightSwitch to turn the light on.
←	3. **System** signals to the **Needle** to be able to move.
→	4. **Student** starts to operate the spectrophotometer

4.6.2 UC-2 AdjustBlank

Use Case UC-2:	AdjustBlank
Related Requirements:	REQ4, REQ5, REQ8, REQ12, REQ14, REQ15
Initiating Actor:	Student
Actor's Goal:	To adjust the pointer to the largest scale.
Participating Actors:	Blank Adjust Dial
Preconditions:	• The pointer points to the zero position.
	• The transparent solution is in the cuvette chamber.
Postconditions:	The pointer point to the largest scale position.
Flow of Events for Main Success Scenario:	
→	1. **Student** rotates the dial.
←	2. **System** pointer of the scale points to the largest scale position.

4.6.3 UC-3 AdjustZero

Use Case UC-3:	AdjustZero
Related Requirements:	REQ3, REQ5, REQ8, REQ12, REQ14, REQ15
Initiating Actor:	Student
Actor's Goal:	To calibrate the pointer position strictly to direct to zero scale.
Participating Actors:	Zero Adjust Dial
Preconditions:	The pointer doesn't point to zero position.
Postconditions:	The pointer point to zero.
Flow of Events for Main Success Scenario:	
→	1. **Student** controls the dial.
←	2. **System** pointer of scale points to the zero position.

4.6.4 UC-4 SelectWaveLength

Use Case UC-4:		SelectWaveLength
Related Requirements:		REQ2, REQ8, REQ14
Initiating Actor:		Student
Actor's Goal:		Select the desired wavelength of light.
Participating Actors:		Meter
Preconditions:		Desired wavelength is not selected
Postconditions:		Desired wavelength are selected and displayed in namometer appears in the window.
Flow of Events for Main Success Scenario:		
→	1. Rotating the wavelength selection knob.	
←	2. The corresponding value of length will display on the namometer.	
→	3. Continuing to rotate the knob until a desired wavelength.	
←	4. The value of desired wavelength will display on the namometer.	

4.6.5 UpdateInfoBoard

Use Case UC-5:		UpdateInfoBoard
Related Requirements:		REQ9
Initiating Actor:		Teacher
Actor's Goal:		To upload information and display on the virtual devices.
Participating Actors:		N/A
Preconditions:		• Default information is displayed on information board
		• Teacher has logged in the system.
Postconditions:		Students can see the information posted by teacher when they log in the system
Flow of Events for Main Success Scenario:		
→	1. **Teacher** enters words in the information board.	
←	2. The information board displays the information.	

4.6.6 MeasureTubeValue

Use Case UC-5:	MeasureTubeValue
Related Requirements:	REQ7, REQ9, REQ11, REQ13, REQ14
Initiating Actor:	Student

Actor's Goal:	To get the value of tube.	
Participating Actors:	N/A	
Preconditions:	The test tube value is unknown.	
Postconditions:	Students can read the test tube value from the meter.	
Flow of Events for Main Success Scenario:		
→	1.**Student** select the test tube and put it in the sample holder.	
←	2. The test tube value shows in the meter.	

4.6.7 SetSampleValue

Use Case UC-5:	SetSampleValue
Related Requirements:	REQ10
Initiating Actor:	Teacher
Actor's Goal:	To set the sample value.
Participating Actors:	N/A
Preconditions:	The sample values are not set.
Postconditions:	The sample values change to the values wanted.
Flow of Events for Main Success Scenario:	
→	1.**Teacher** set the sample values in teacher interface
←	2. The sample values are changed to the values teacher set.

4.6.8 UC-8 Login

Use Case UC-8:	Login
Related Requirements:	REQ11
Initiating Actor:	Teacher, Student
Actor's Goal:	Access to the system
Participating Actors:	N/A
Preconditions:	System displays a login screen.
Postconditions:	Users log in their interfaces respectively.
Flow of Events for Main Success Scenario:	
→	1. User enters their username and password.
←	2. **System** displays main menu on screen.
Flow of Events for Main Extensions:	
→	1. User enters invalid username or password.
←	2. **System r**eject user from accessing the system.

4.7 System Sequence Diagrams

This section is System Sequence Diagrams for each Use Case. They show the events that external actors generate, their order, and possible inter-system events.

First, the teacher and student GUI sequence diagrams (Figure 4.3 and 4.4) are given to provide a overview of our virtual spectrophotometer application.

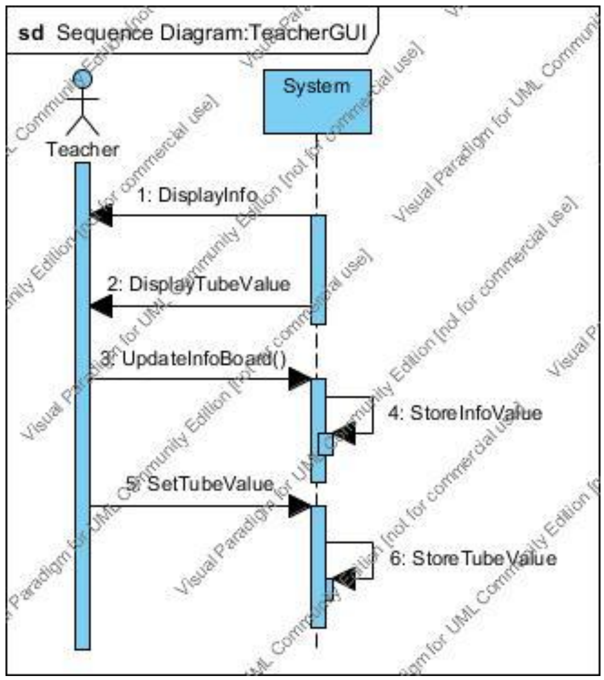


Figure 4. 3 TeacherGUI Sequence diagram

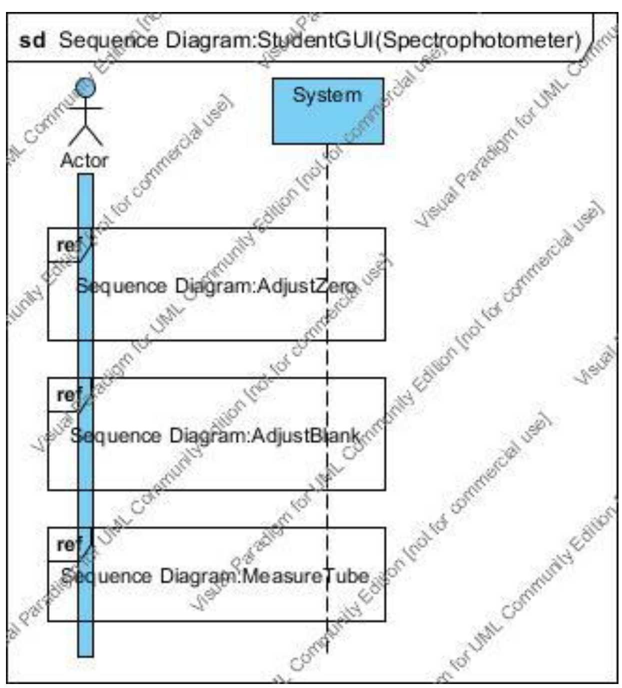


Figure 4. 4 StudentGUI Sequence Diagram

The system Sequence Diagram of UC-2 is as Figure 4.3.

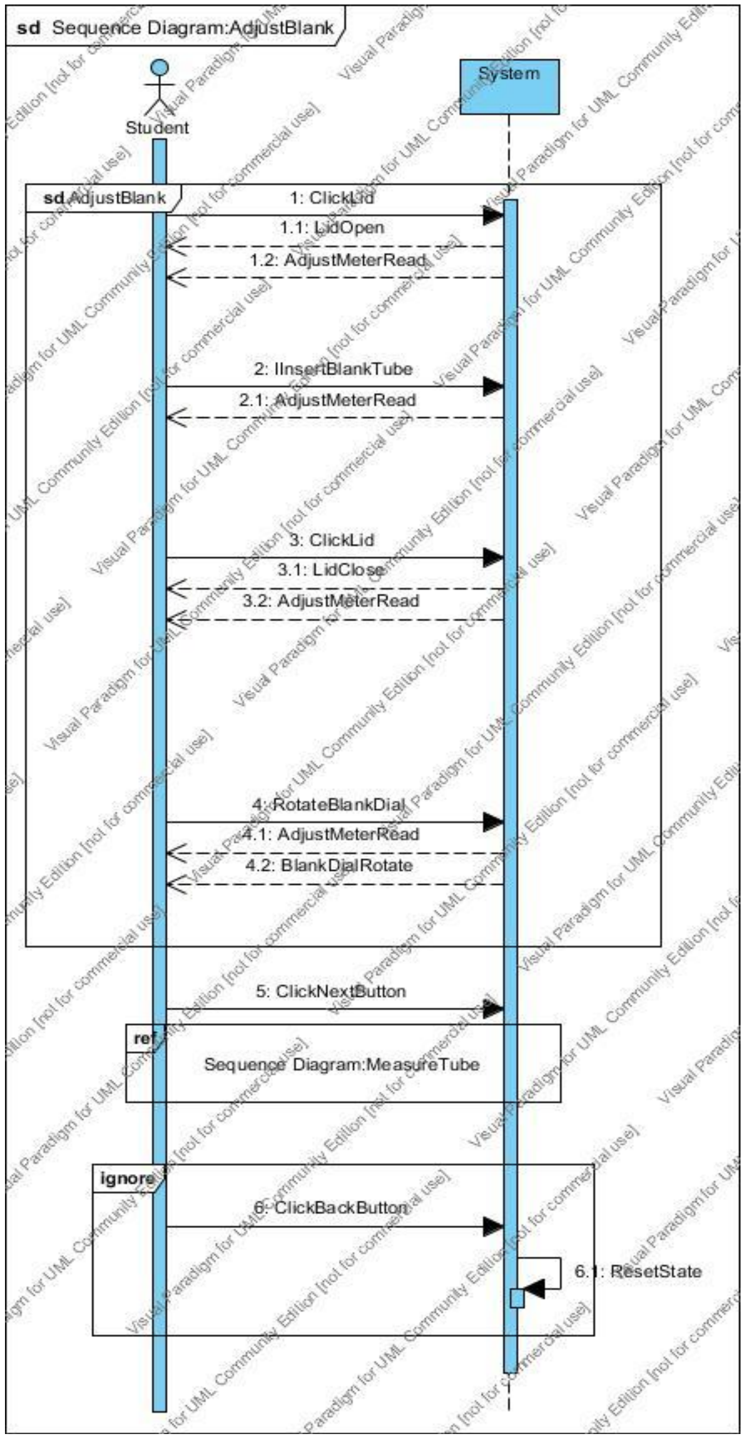


Figure 4. 5 Sequence Diagram: AdjustBlank

The system Sequence Diagram of UC-3 is as Figure 4.4.

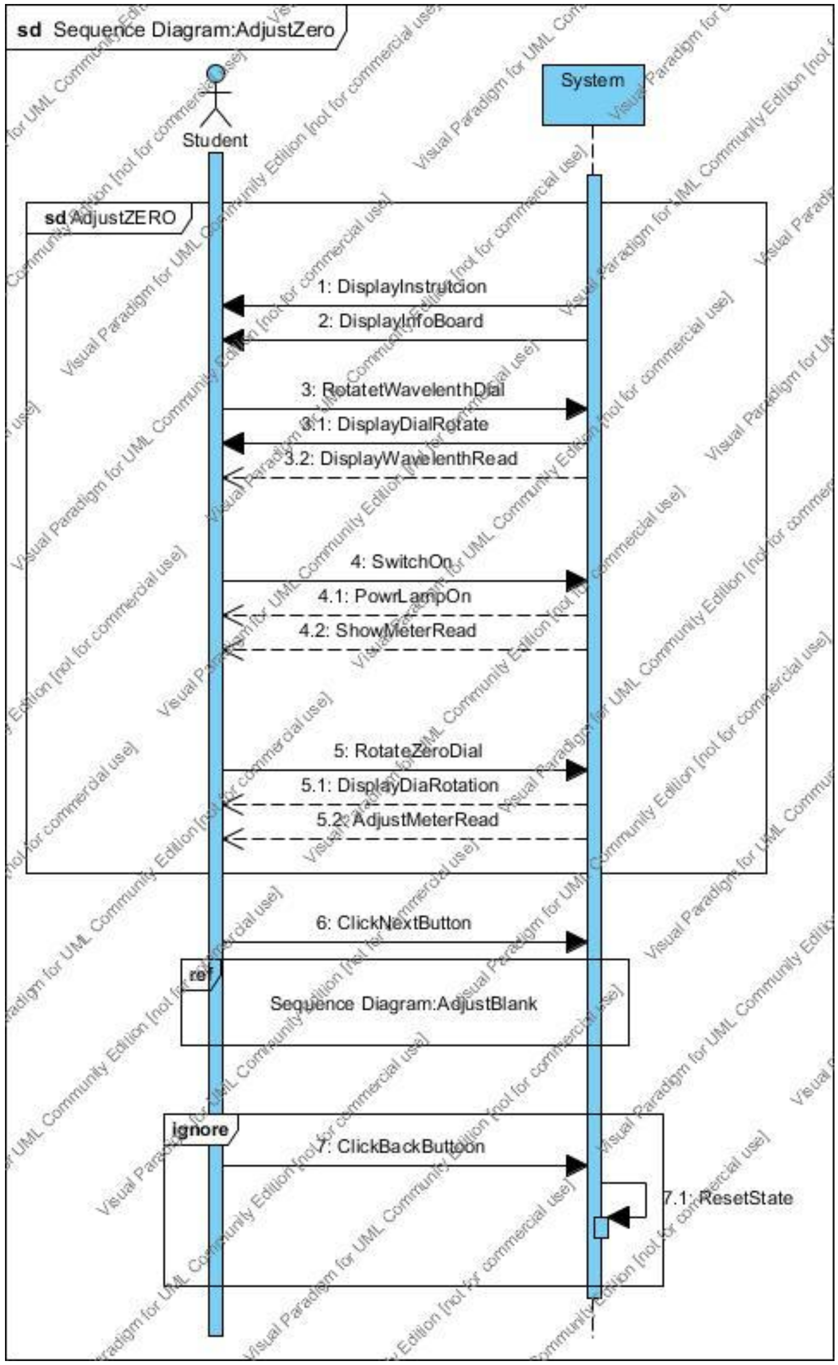


Figure 4. 6 Sequence Diagram: AdjustZero

The system Sequence Diagram of UC-6 is as Figure 4.5.

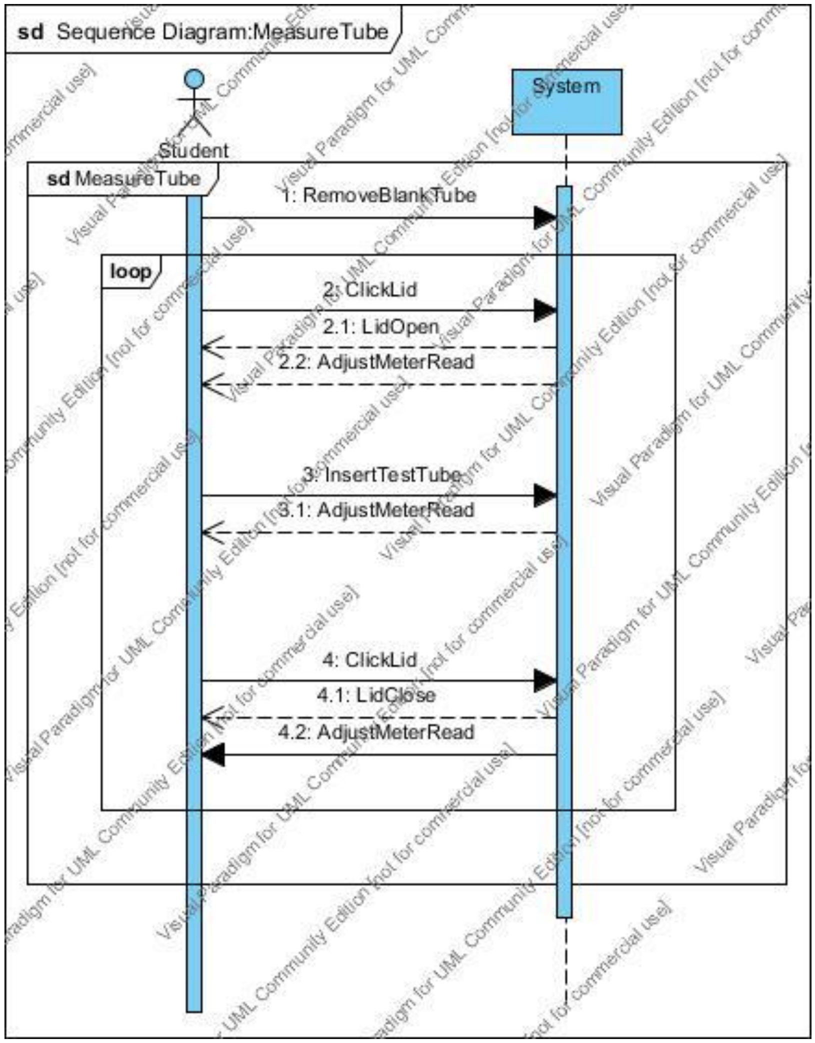


Figure 4. 7 Sequence Diagram: MeasureTubeValue

The system Sequence Diagram of UC-8 is as Figure 4.6.

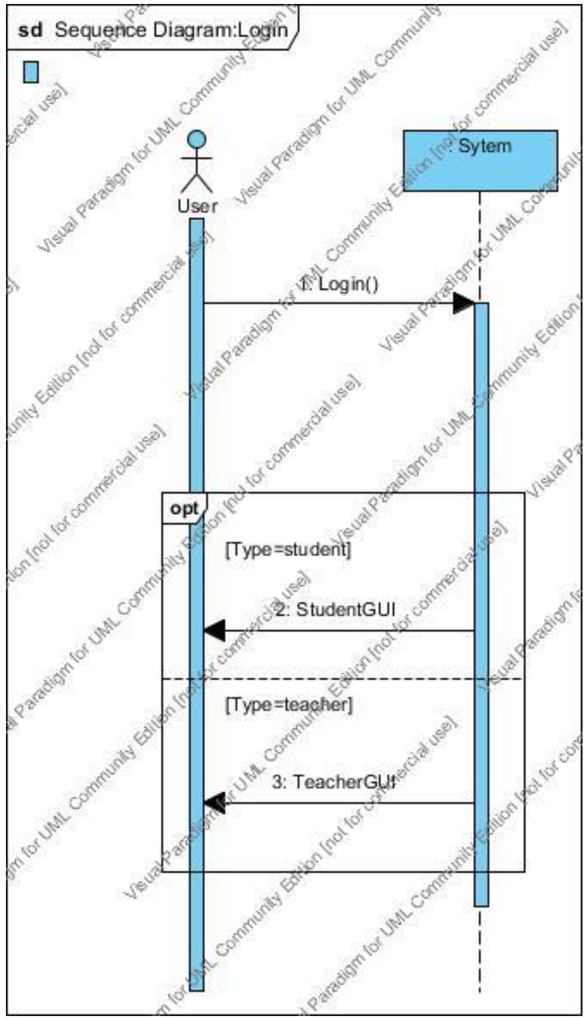


Figure 4. 8 Sequence Diagram: Login

5. Effort Estimation using User Case Points

The *Use Case Points (UCP)* method provides the ability to estimate the person-hours a software project requires based on its use cases. The UCP method analyzes the use case actors, scenarios, nonfunctional requirements, and environmental factors and abstracts them into an equation.

In order to calculate UCP, The variables: Unadjusted Use Case Points (UUCP), The Technical Complexity Factor (TCF), and The Environment Complexity Factor (ECF) should be known.

5.1 Unadjusted Use Case Points (UUCPs)

Unadjusted Use Case Points (UUCPs) are computed as a sum of those two components: The Unadjusted Actor Weight (UAW) and The Unadjusted Use Case Weight (UUCW).

Table 5.1 shows the ratings for the actors in the case study of spectrophotometer, for which the actors are described before. The UAW is calculated by totaling the number of actors in each category, multiplying each total by its specified weighting factor, and then adding the products.

Table 5. 1 Unadjusted Actor Weight (UAW)

Actor name	Description of relevant characteristics	Complexity	Weight
Student	Student is interacting with the system via a graphical student interface	Complex	3
Teacher	Teacher is interacting with the system via a graphical teacher interface	Average	2
PowerSwitch	PowerSwitch is another subsystem which interacts with main system through a defined API	Simple	1
Needle	Same as PowerSwitch	Simple	1
Dials	Same as PowerSwitch	Simple	1
Meters	Same as PowerSwitch	Simple	1
Lid	Same as PowerSwitch	Simple	1
Sample	Same as PowerSwitch	Simple	1
Sample holder	Same as PowerSwitch	Simple	1
Information Board	Information board is a subsystem updating information through a protocal	Average	2

UAW(Spectrophotometer) = 6*Simple + 2*Average+1*complex = 13

Unadjusted Use Case Weight (UUCW) is calculated by tallying the use cases in each category, multiplying each count by its specified weighting factor (Table 4.3), and then adding the products. Table 5.2 shows the details of UUCW

Table 5. 2 Use case classification for the case study of Spectrophotometer control

Use case	Description	Category	Weight
SwitchOn (UC-1)	To make spectrophotometer start to work	Simple	5
AdjustBlank (UC-2)	To adjust the needle on the meter to the full scale	Average	10
AdjustZero (UC-3)	To Adjust the needle on the meter to 0 scale	Average	10
SelectWaveLength (UC-4)	To Adjust the needle on the meter to 0 scale	Average	10
UpdateInfoBoard (UC-5)	To Adjust the needle on the meter to 0 scale	Average	10
OpenLid (UC-6)	To open the lid	Simple	5
CloseLid (UC-7)	To close the lid	Simple	5
SelectTestTube (UC-8)	To choose one of the test tubes from the rack	Simple	5
InsertTestTube (UC-9)	To insert the selected test tube into the chamber	Simple	5
CheckChamber (UC-10)	To remove the test tube in chamber	Simple	5

The UUCP is computed by adding the UAW and the UUCW. Based on the scores in Table 5.1 and Table 5.2, the UUCP for our case study project is *UUCP* = *UAW* + *UUCW* = 13 + 70 = 83.

5.2 Technical Complexity Factor(TCF)

Technical Complexity Factor(TCF)-Nonfunctional Requirements

Table 5. 3Table Technical complexity factor for case study of Spectrophotometer

Technical factor	Description	Weight	Perceived Complexity	Calculated Factor (Weight*Perceived Complexity)
T1	Distributed system (running on multiple machines)	2	3	6
T2	Reusable design or code	1	3	3
T3	Users expect good performance (including the response time)	1	3	3
T4	Reliability (to allow operation when execute wrong steps)	1	2	2
T5	Easy to use is very important (including the operation and interface display)	0.5	5	2.5

T6	Easy to change (to add new features or modify existing ones) minimally required	1	1	1
Technical Factor Total:				17.5

Wi = weight of i_{th} technical factor (Table 5.1)
Fi = perceived complexity of i_{th} technical factor (Table 5.2)
TCF = 0.6+(0.01*17.5) = 0.775

5.3 Environment Complexity Factor (ECF)

Environment Complexity Factor (ECF)

Table 5. 4 Environmental complexity factors for the case study of Spectrophotometer

Environmental factor	Description	Weight	Perceived Impact	Calculated Factor (Wight*Perceived Impact)
E1	Beginner familiarity with the UML-based development	1.5	1	1.5
E2	Some familiarity with application problem	0.5	2	1
E3	Some knowledge of object-oriented approach	1	2	2
E4	Beginner lead analyst	0.5	1	0.5
E5	Programming language of average difficulty will be used	-1	3	-3
Environmental Factor Total:				2

ECF = 1.4+2*(-0.03) = 1.36
Calculating the Use Case Points (UCP)
UUCP = 83
TCF = 0.775
ECF = 1.36
UCP = UUCP*TCF*ECF = 86.1955

6. User Interface Specification

A spectrophotometer should be:

1. Power switch to turn the instrument ON or OFF
2. Test tubes (or, “cuvettes”), contained in a rack; each cuvette contains a different solution
3. Sample holder (or, “chamber”) for holding the test tube to measure the concentration of its substance
4. Dark dial for calibrating the zero illumination
5. Light dial
6. Absorbance meter with two scales (transmittance and absorbance), a needle, and a pilot lamp to indicate when the instrument is powered. We assume an analog spectrophotometer with a needle pointer instead of a digital/numeric readout.
7. Wavelength dial for setting the color of the illumination light

There are some other parts of a physical spectrophotometer that will not be simulated in the virtual instrument, because their operation will not be explicitly visible to the user:

a). A light source shines the selected wavelength light from the spectrophotometer through the sample
b). A focusing device that transmits an intense straight beam of light
c). A monochromator to separate the beam of light into its component wavelengths
d). A photoelectric detector

Figure 6.1 shows the specifications of our system user interface. Through our system, students can see the overall system configuration as they see in the real lab.

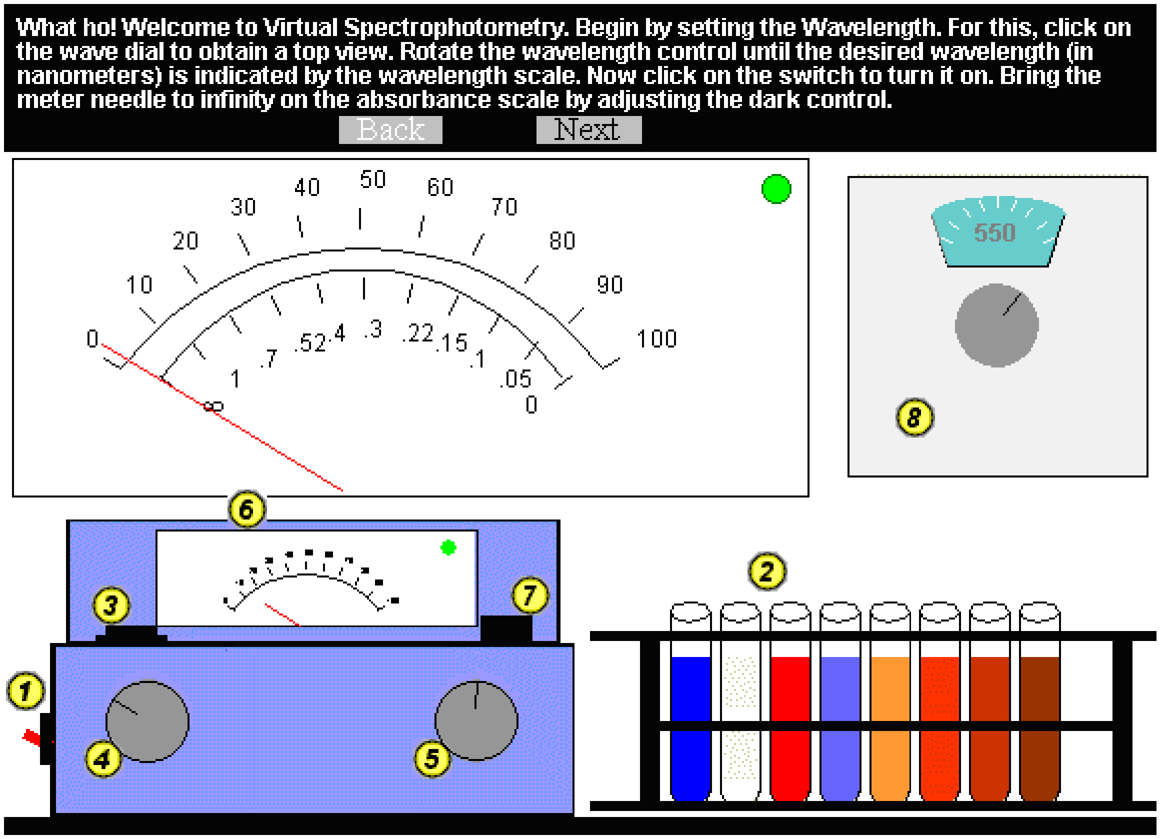


Figure 6. 1 Screen snapshot of the spectrophotometry virtual laboratory.

The parts in Figure 6.1 are listed following: ① on/off switch, ② rack with test tubes with solutions, ③ sample holder for the test tube, ④ zero control dial, ⑤ light control dial, ⑥ meter with needle and pilot light (magnified view shown on top), ⑦ wavelength control dial for setting the color of the illumination light, ⑧ magnified view of the wavelength dial.

6.1 Preliminary Design

In order to get the correct experiment result, students should do experiments step by step via our system: As you click on the software, you will see our login interface (Figure 6.2) at first glare.

Figure 6. 2 Login Interface

6.1.1 Teachers' interface

On the above screen, you can choose teachers' account or students' account. Then after valid your username, password and verification code enter a main system interface.
If you are a teacher, you will see the following picture (Figure 6.3)

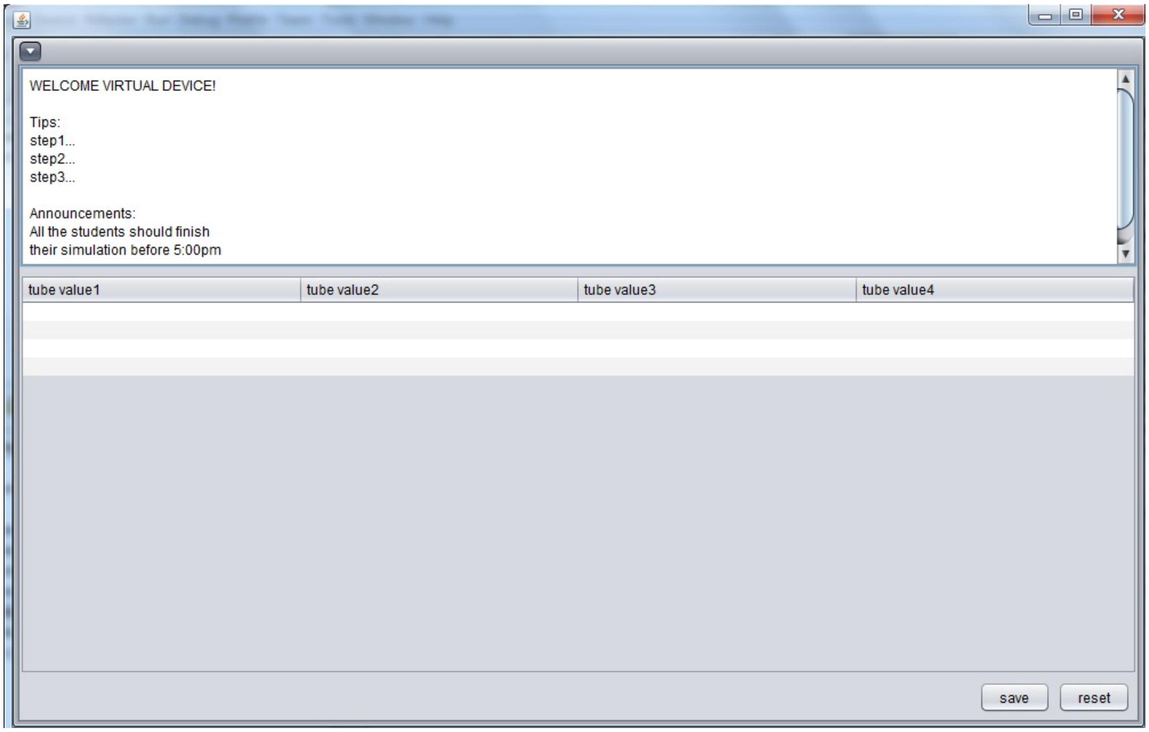


Figure 6. 3 Teacher's Interface

1) As a teacher, you can type in the tips and announcement. Students will see these tips and announcements in students interface
2) Also, you can type in the tube value, this value can be used for testing the results.

6.1.2 Students' interface

If you are a student, you will see the figure 6.1

1) Turn on the power switch (labeled as 1 in Figure 6.1). We assumed that the machine has been warmed up.
2) With no cuvette in the chamber, a shutter cuts off all light from passing though the cuvette chamber. Under this condition therefore, the machine may be adjusted to read infinite absorbance (zero% transmittance) by rotating zero adjust knob (labeled as 4 in Figure6.1). *Do not touch this knob again during the rest of the following procedure.*
3) Select the desired wavelength of light at which absorbance will be determined by rotating wavelength selection knob (labeled as 7 in Figure 6.1) until the desired wavelength in nanometers appears in the window. A nanometer, formerly millimicron, equals 10^{-9} meter.
4) Fill the blank (labeled as 2 in Figure 6.1) cuvette with the solvent used to dissolve specimen (often distilled water). Polish to clean, insert into the cuvette chamber (labeled as

3 in Figure 6.1), aligning mark to front. Close chamber cover.

5) Rotate blank adjust knob (labeled as 5 in Figure 6.1) to adjust absorbance to read zero from the meter (labeled as 6 in Figure 6.1).

6) Remove blank cuvette.

6.2 User Effort Estimation

Assume that the virtual device is currently open. Student need know following step to navigate the interface:

i). Choosing student or teacher account and entering the username, password and verification number then click login.

ii). If you are a teacher you can click the text part to upload information or click to enter the test value, while if you are a student, you will see following steps.

iii). Begin by setting the wavelength. For this, click on the wave dial to obtain a top view, rotate the wavelength control until wavelength is indicated by the wavelength scale. Click on the switch to turn it on. Bring the meter needle to infinity on the absorbance scale by adjusting the dark control.

Event: "SwitchOn"

Event "Adjust_to_zero"

iv. Click on the sample holder to open it. Choose the test tube with the transparent solution and place it in the holder. Then close the sample holder clicking it suit. Rotate the light control until the meter indicates zero on the absorbance scale. This procedure calibrates the instrument relative to the blank.

Event:"Openlid"

Event: "CloseLid"

v. Open sample holder and drop the transparent test tube back into the rack. Click on any of the other sample test tubes and drop it into the sample holder as before. Take down the readings of transmittance and absorbtion only after closing the sample holder. Repeat the above process for all the test tubes. Close the window after you are done.

Event: "Adjust_to_blank"

Events:"setTestTube"and"removeTestTube"

7. Domain Analysis

7.1 Domain Model

Domain Model is as Figure 7.1 shows

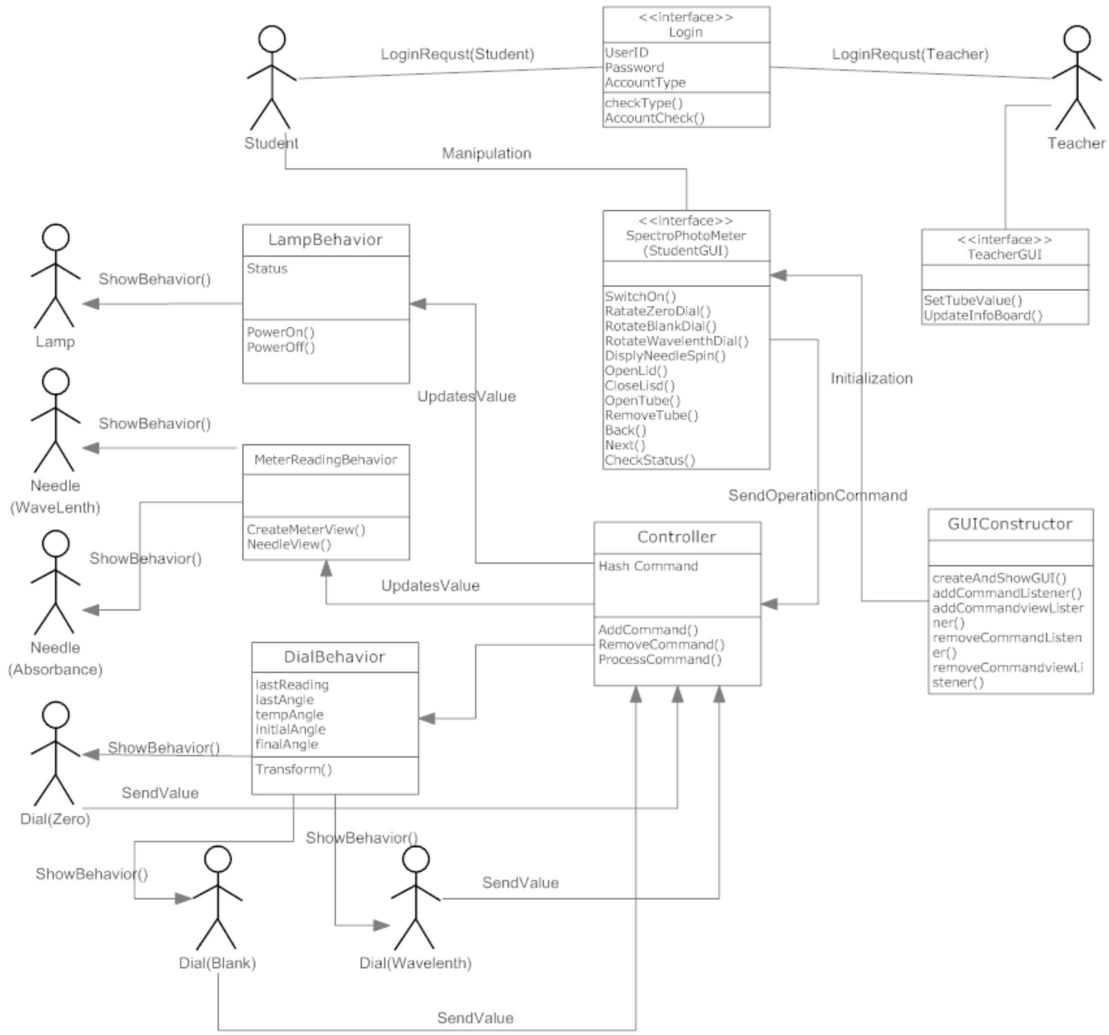


Figure 7. 1 Domain Model

7.1.1 Concepts definition

Table 7.1 lists the responsibilities and the worker titles (concept names) to whom the responsibilities are assigned. And each responsibility of a concept has a type “D” or “K” denotes doing vs. knowing, respectively.

Table 7. 1 Responsibility description for the Virtual Spectrophotometer

Responsibility Description	Type	Concept Name
Provide a virtual spectrophotometer to student which will give them real experience when manipulating it.	K	StudentInterface
Provide a interface for teacher to update information board and set up the test tube values.	D	TeacherInterface
Construct the virtual spectrophotometer	D	GUIConstructor
Receive the user manipulation and store them into a hash command table then process them	K	Controller
Receive the result from controller then update the MeterReading Behavior (rotate angle).	D	MeterReadingBehavior
Receive the result from controller then update the Dial Behavior (rotate angle).	D	DialBehavior
Receive the result from controller then update the Lamp status (on or off).	D	LampBehavior

Users can see our virtual Spectrophotometer from our interface. It is the only way for users to interact with our system. It will display all the reasonable respond to the user's action.

OperationSelector can detect users' command and then figure out the operation of the device. And send corresponding data to DataArchiver.

DataArchiver will receive and collect the data sent by the operation selector. These data can be regarded as the status of different knobs. As we mentioned above, our system will allow students to make mistakes, so they can do any step whenever they want. Considering these aspects, we will give a reasonable respond on our system when students make mistake. However, in the real world, operation sequence and different combination of knobs status will have an influence on the final results. So the Data controller will arrange the data in a right order which can show the real sequence of different operations. Then the data can be sent to database connection.

Database connection can connect the data controller with actor database and Data Calculator. In this concept, data can be updated when receiving the data from data controller. And put these data forward to the actor database and data calculator.

When collecting the data from database connection, the data calculator will compared the data with the reference data. Reference data stand for the all the combinations of operation sequences and knob status. Through this comparison, the system could exactly know the operation sequence and different knob status. Then the system will give a corresponding reaction to the actor meter and actor indicator lamp.

7.1.2 Association Definitions

Association Definitions is as Table 7.2 shows

Table 7. 2 Association Definitions

Concept pair	**Association description**	**Association name**
Student→Login	User behaves an operation (turn on device, adjust zero/ blank, select/insert/remove	OperationRequest
Teacher→Login	Record the user operation, and send it to DataArchiver	OperationDataDeliver
Teacher→TeacherInterface	Generate the update request to DatabaseConnection	DataUpdateRequest
GUI	Update and Retrieve data from Database	DateTransaction
DatabaseConnection →ResultCalculator	Send the retrieved information to ResultCalculator	DataRetrieve
ResultCalculator →Interface	Deliver the result to Interface	DeliverResult
Interface→User	Show the result to user	DisplayResult
Interface →IndicateLamp	Show the IndicateLamp result(shining or blank)	ShowIndicateLampResult
Interface→Meter	Show the Meter result(the angle needle spins)	ShowMeterResult
TeacherOperation→ DataArchiver	Send updated information and sample value	TeacherOperationRequst
ResultCalculator→MeterDisplay	Send the variable to calculate the angle spin on screen	Send the variable to calculate the angle spin
MeterDisplay→UserInterface	Display the spin result on screen	DisplaySpinResult

7.1.3 Attributes definition

Attribution Definitions is as Table 7.3 shows.

Table 7. 3 Attribution Definitions

Concept	Attributes	Attribute Description
StudentInterface	SwitchOn()	Turn on the machine
	RotateZeroDial()	Rotate knob to do zero adjustment
	RotateBlankDial()	Rotate knob to do blank adjustment
	RotateWavelength()	Rotate knob to select desired wavelength
	DisplayNeedleSpin()	Display the needle spin on the screen
	OpenLid()	Display the needle spin on the screen
	CloseLid()	Display the needle spin on the screen
	OpenTube()	Insert the tube to sample holder
	RemoveTube()	Remove the tube from sample holder
	Back()	Click to go next step
	Next()	Click to go last step
	CheckStatus()	Check the system status
LampBehavior	PowerOn()	Change the lamp status to 'ON'
	PowerOff()	Change the lamp status to 'OFF'
DialBehavior	IsRotated()	Determine whether the knob has been rotated
TeacherInterface	SetTubeValue()	Set the test value of the tube
	UpdateInfoBoard()	Set the content of Information Board
GUIconstructor	CreateAndShowGUI()	Create the show the interface
	AddCommandListsener()	Add the Listener
	RemoveCommandListener()	Remove the listener
Controller	Tranform()	Calculate the vector
	Getbound()	Calculate the position
MeterReadingBehavior	CreateMeterView()	To show the meter on the screen
	NeedleView()	To show the need on the screen

7.1.4 Traceability Matrix

Traceability Matrix is as Table 7.4 shows

Table 7. 4 Traceability Matrix

Use Case	PW	Interface	OperationSelector	DataArchiver	Database Connection	DataCalculator	TeacherOperation
UC-1	4	X	X				
UC-2	20	X	X	X	X	X	
UC-3	20	X	X	X	X	X	
UC-4	10	X	X	X	X	X	
UC-5	1	X					X
UC-6	10	X	X				
UC-7	10	X	X				
UC-8	7	X	X				

7.2 System Operation Contracts

Operation SwitchOn is the first step. Table 7.5 shows the system operation contact for switch on.

Table 7. 5 System Operation Contract for SwitchOn

Operation	SwitchOn
Preconditions	• IndicatorLampStatus= false (off) • VirtualMachineStatus=false(off) • The meter keep stable in initial position
Postconditions	• IndicatorLampStatus=true(on) • VirtualMachineStatus=true(on) • NeedleStatus ≠ Null

After switching on the machine, users start to do the simulation step by step. We design our system according to what happens in the real world. In other words, our system makes students feel that they are in a real lab. They can do everything they can do in the real lab by using our software. For example, they don't have to following the correct scenario of the experiment, even though what they do will result in an incorrect value. Table 7.6, Table 7.7

and Table 7.8 show the system operation contracts for different knobs. These contracts are in a correct order of using spectrophotometer.

Table 7. 6 System Operation Contracts for AdjustBlankTable

Operation	**AdjustBlank**
Preconditions	• NeedleStatus=0 • IsSampleHolder= 0 (transparent solution)
Postconditions	• NeedleStatus=Full scale

Table 7. 7 System Operation Contracts for AdjustZeroTable

Operation	**AdjustZero**
Preconditions	• NeedleStatus ≠ Null
Postconditions	• NeedleStatus=0

Table 7. 8 System Operation Contracts for SelectWaveLength

Operation	**SelectWavelength**
Preconditions	• Rotating the wavelength selection knob.
Postconditions	• Corresponding wavelength is selected.

Teachers can upload information and set concentration for all samples. The following Tables, Table 7.9 and Table 7.10 are system operation contracts for these two UCs.

Table 7. 9 System peration Contracts for UpdateInfoboard

Operation	**UpdateInfoboard**
Preconditions	• Teachers login the system successfully LoginTeacherSystem=1
Postconditions	• Information board display on the screen.

Table 7. 10 System Operation Contracts for SetSampleValue

Operation	**SetSampleValues**
Preconditions	• Teachers login the system successfully LoginTeacherSystem=1
Postconditions	• Information board display on the screen.

Following contract (as shown in Table 7.11) is related to the test tube.

Table 7. 11 System Operation Contracts for MeasureTestTube

Operation	**MeasureTestTube**
Preconditions	• The IndicatorLamp is on • The spectrophotometer is calibrated • The sample holder is empty
Postconditions	• IndicatorLampStatus is on • The sample holder has a sample in it • The needle points to the sample value in the meter

8. Interaction Diagrams

8.1 ID1 SwitchON

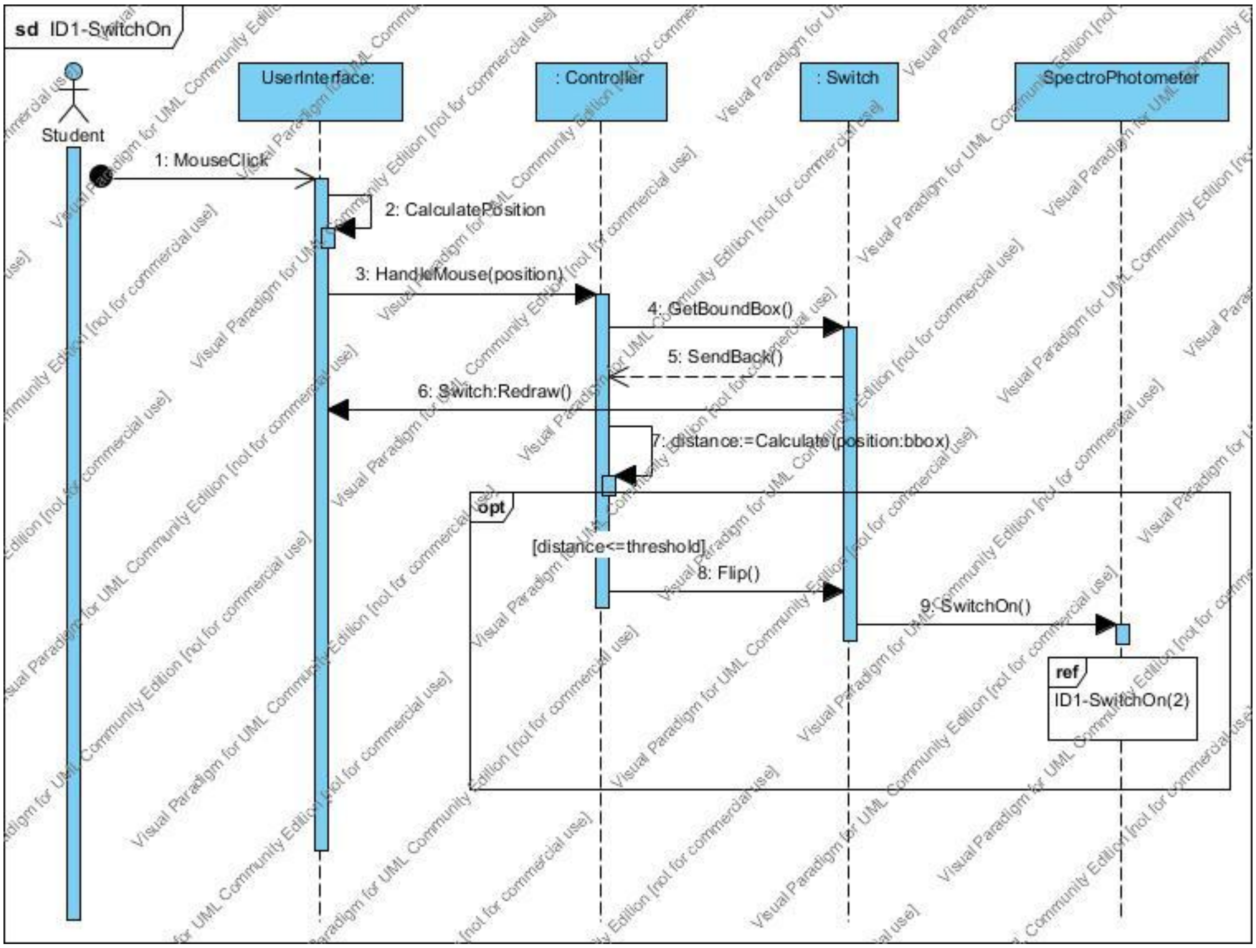


Figure 8. 1 Interaction diagram of SwitchOn

In UC-1 (Figure 8.1), the interaction with the system turns on by user clicking the mouse in the neighborhood of the power switch. The system detects a mouse click, measures its distance from the center of the power switch, and if it is less than a given threshold, it activates the switch. When the switch object is activated, it in turn activates the pilot lamp to indicate that the instrument is “on”, and enables interaction with other parts of the spectrophotometer.

Figure 8.2 refers to a fragment shown in the Figure 8.1. We consider the operation of the power switch as an event and define different states of the system. In 8.2 we assume that the spectrophotometer’s initial state is OffEmptyClosed

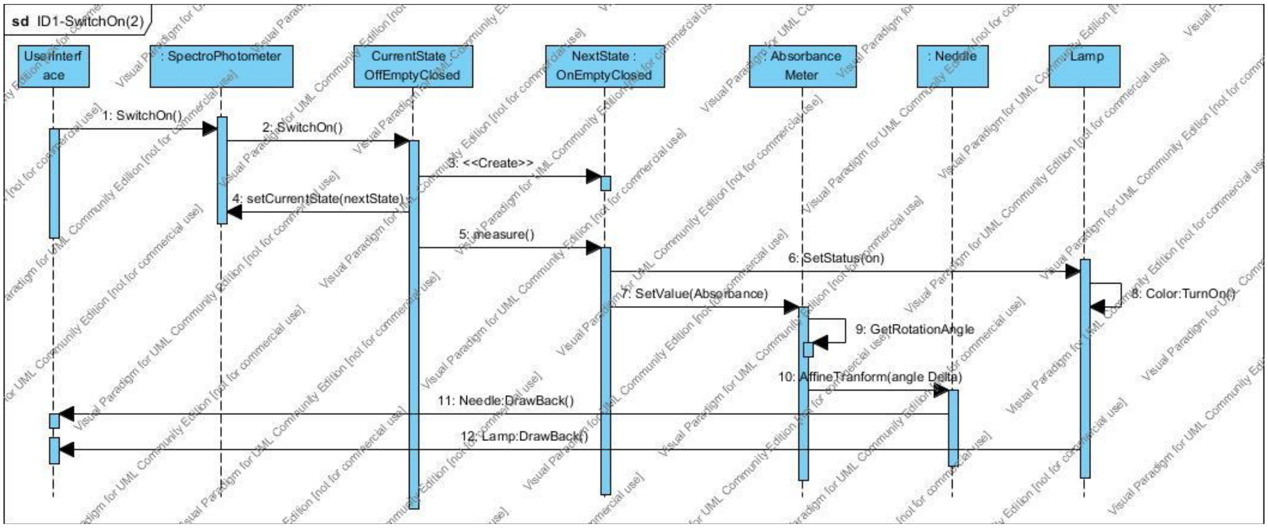


Figure 8. 2 Interaction diagram of SwitchOn (2)

8.2 ID2 AdjustZero

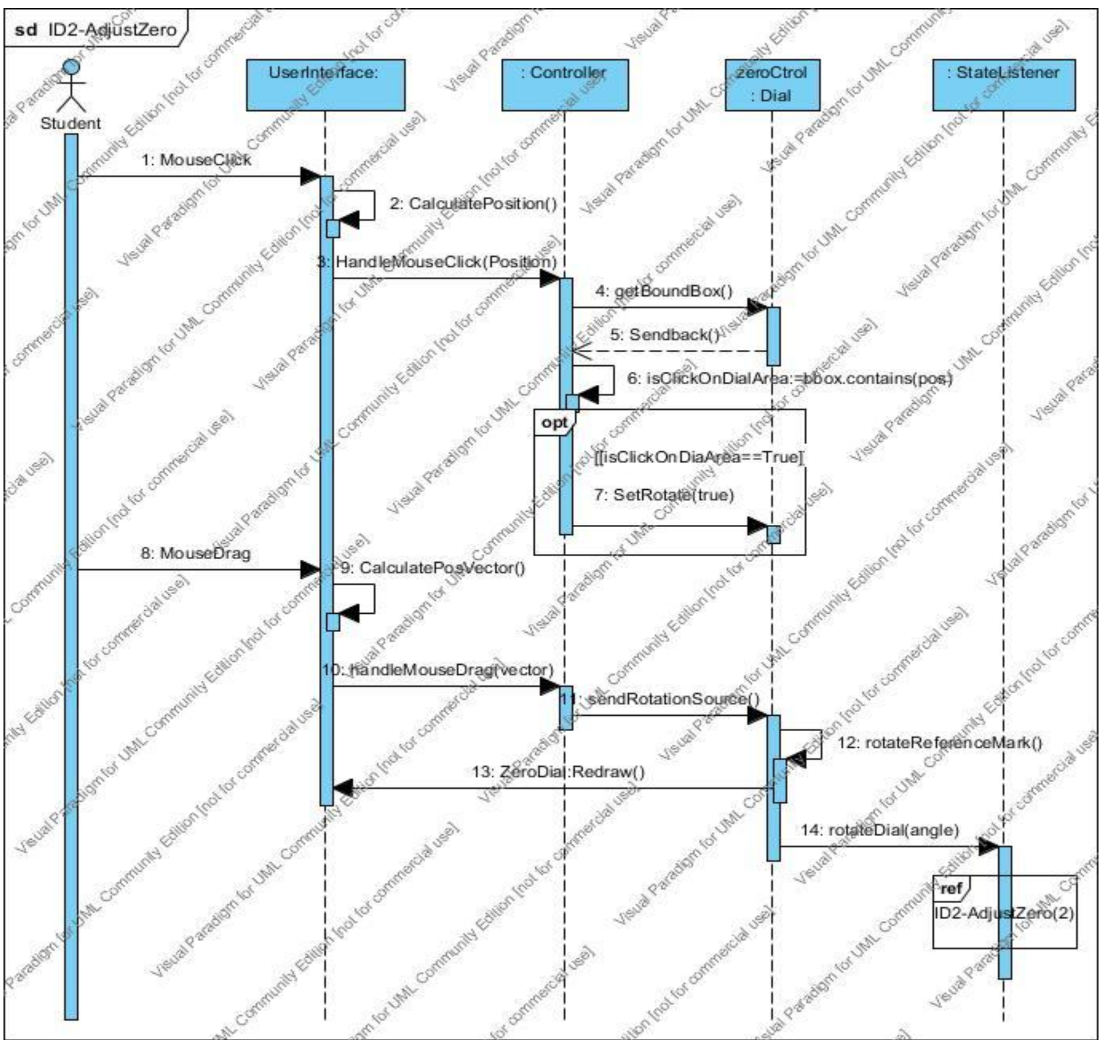


Figure 8. 3 Interaction diagram of AdjustZero

Figure8.3 describes the class behavior of zero adjustment. By detecting the position of mouse click and Mouse drug, the controller will calculate the Position and the translation vector. The controller will regard this operation as 'rotation'. When the zero dial is rotated, this causes the spectrophotometer to redo the measurement. The "measuring" behavior performs the calculations based on the density of the solution contained in the test tube and the light wavelength and sends a command to the AbsorbanceMeter needle to display the wavelength.

Figure 8.3 shows the sequence diagram of object interactions. The process starts with the user clicking in the dial area and dragging the mouse cursor in the desired direction. The system detects if the click is within the dial's area and if so, sets the dial attribute `isRotated` to `true`. This is necessary so that the dial object knows that the subsequent mouse-drag events should be properly processed.

When a mouse-drag event arrives, the dial object reacts by notifying all of its listener objects that it has been transformed, i.e., rotated. When the mouse is dragged, we calculate the vector by which the mouse cursor changed its location. Next, the Controller calculates the rotation angle from this vector and calls a rotation transformation on the Dial object. The dial first rotates the reference mark line to show its new rotation angle. Next, it notifies other objects ("listeners") to perform the measurement behavior.

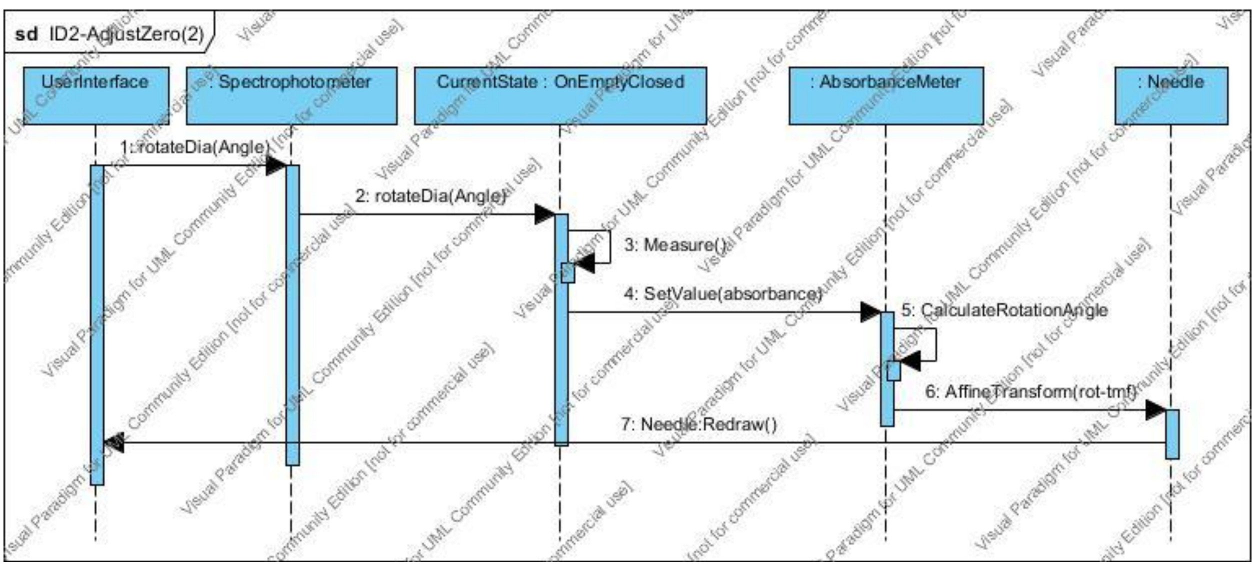


Figure 8. 4 Interaction diagram of AdjustZero (2)

Note that the diagram in Figure 8.4 refers to a fragment shown in Figure 8.3. Based on the calculated value of absorbance, the SpectroPhotometer object calculates by how much AbsorbanceMeter should rotate its needle to display the measured wavelength and calls the Affine`Transform()` method with rotation transformation parameter.

Note that there is a design issue with this distribution of responsibilities. On one hand, we want to keep object communications simple, so they just call the Affine`Transform()`

method on each other. On the other hand, the SpectroPhotometer object is assigned the responsibility to calculate by how much AbsorbanceMeter's needle should rotate. A better assignment of responsibilities would be that the absorbance meter calculates the needle's rotation angle. However, but in this case the calculated measured wavelength needs to be communicated to the AbsorbanceMeter using a different method—a new method needs to be introduced on the AbsorbanceMeter.

8.3 ID3 SelectWaveLength

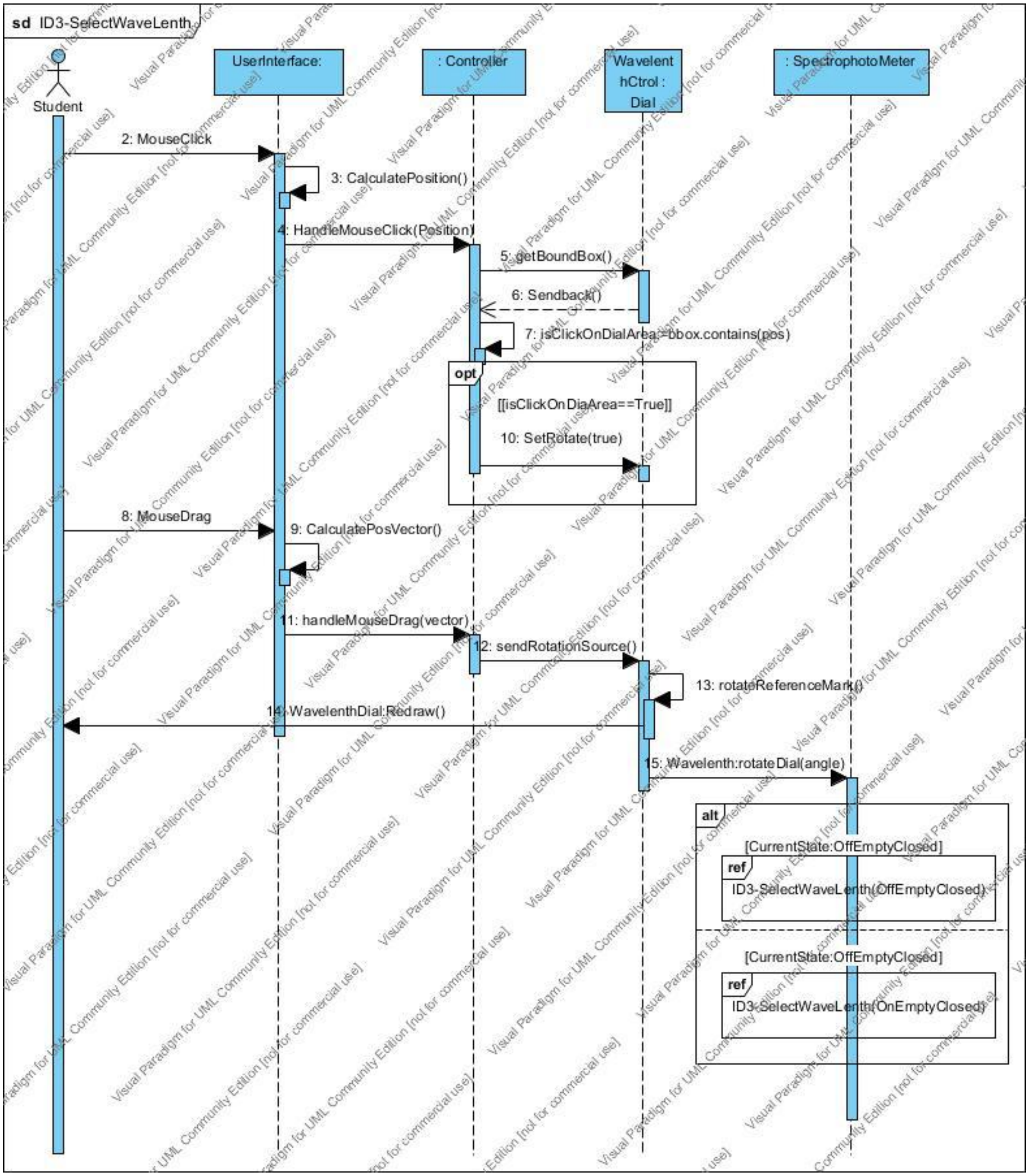


Figure 8. 5 Interaction diagram of SelectwaveLength

Figure 8.5 shows the sequence diagram of wavelength selection. The process starts by user clicking in the dial area and dragging the mouse cursor in the desired direction. The system detects if the click is within the wavelength dial's area and if so, sets the wavelength dial attribute isRotated to true as same process in Figure 8.3.

Following interaction diagram shows the wavelength selection under two different spectrophotometer states, they are OffEmptyClosed and OnEmptyClosed.

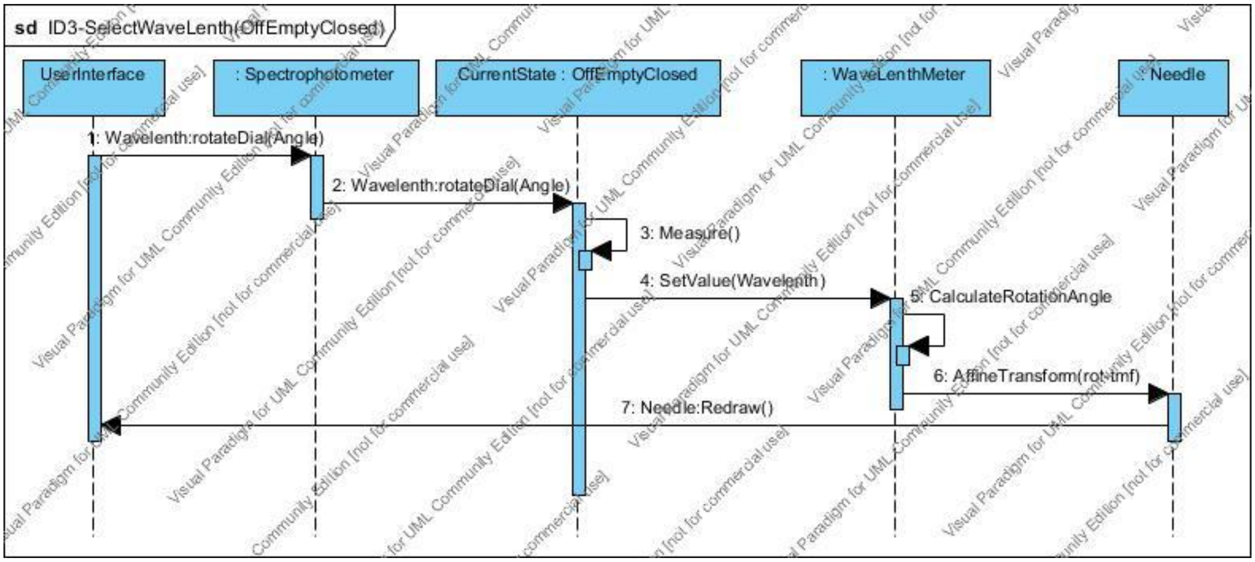


Figure 8. 6 Interaction diagram of SlectwaveLength (OffEmptyClosed)

Figure 8.6 shows the interaction diagram when the system's initial state is OffEmptyclosed. This process starts with the user clicking in the dial area and dragging the mouse cursor in the desired direction, because the system's initial state is OffEmptyClosed, by AffineTransform, the needle will reacts the action without measurement.

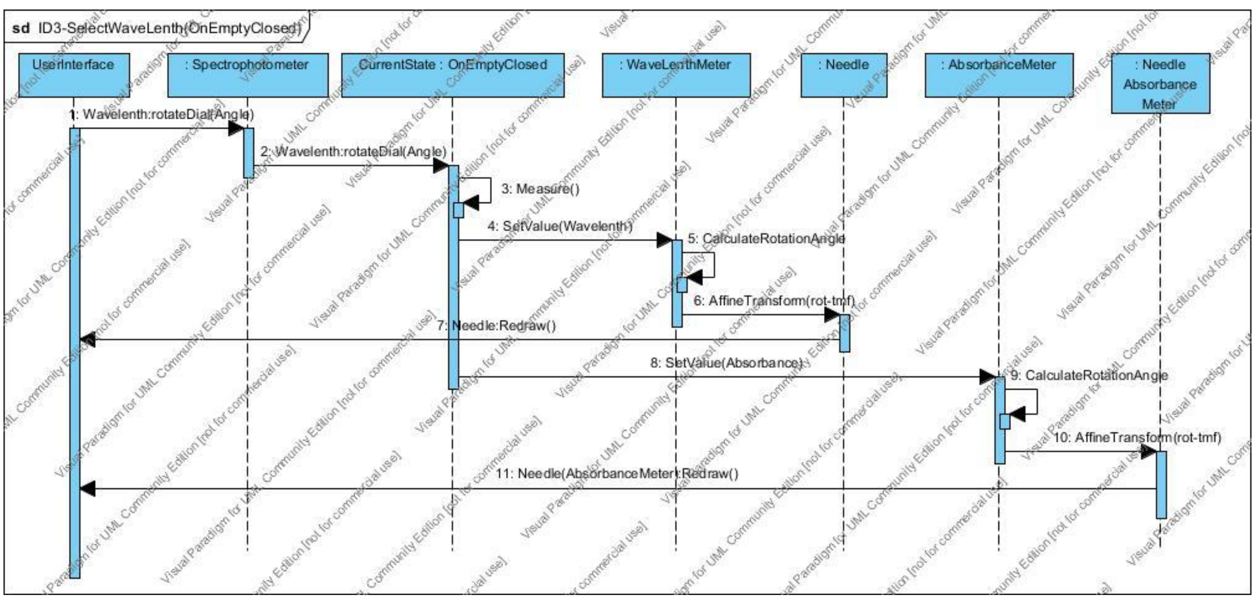


Figure 8. 7 Interaction diagram of SelectWaveLength (OnEmptyClosed)

Under the state 'OnEmptyClosed', when the dial is rotated, this causes the spectrophotometer to redo the measurement. The "measuring" behavior performs the calculations based on the density of the solution contained in the test tube and lid status and sends a command to the AbsorbanceMeter needle to display the wavelength.

8.4 ID4 NextButton

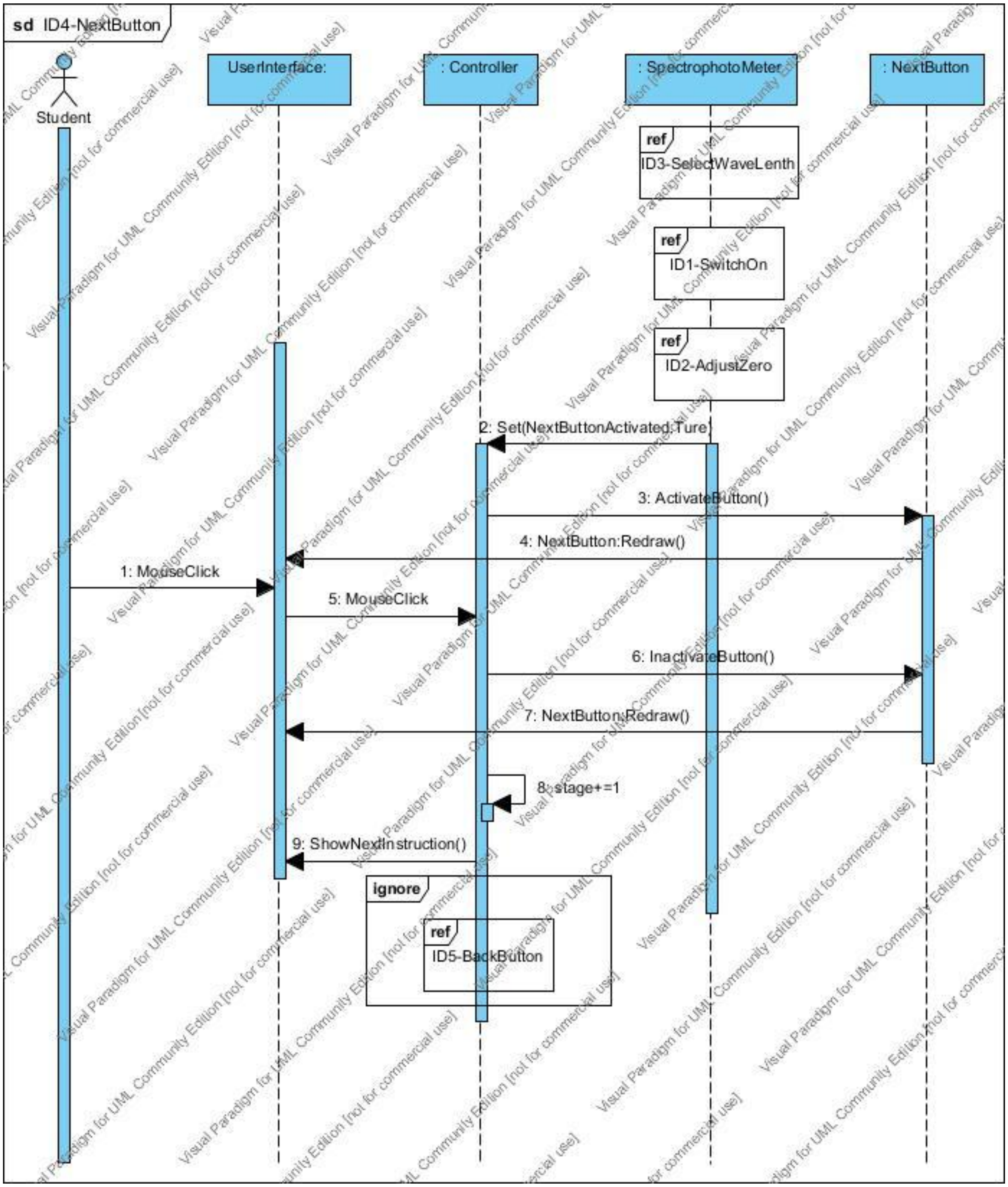


Figure 8. 8 Interaction diagram of NextButton

Figure8.8 shows the interaction diagram of Next Button, the process starts by click the next button, if all the operations are right, it will set NextButtonActive to be 'True'. Then, the interface will show the Next Instruction on the interface.

8.5 ID5 BackButton

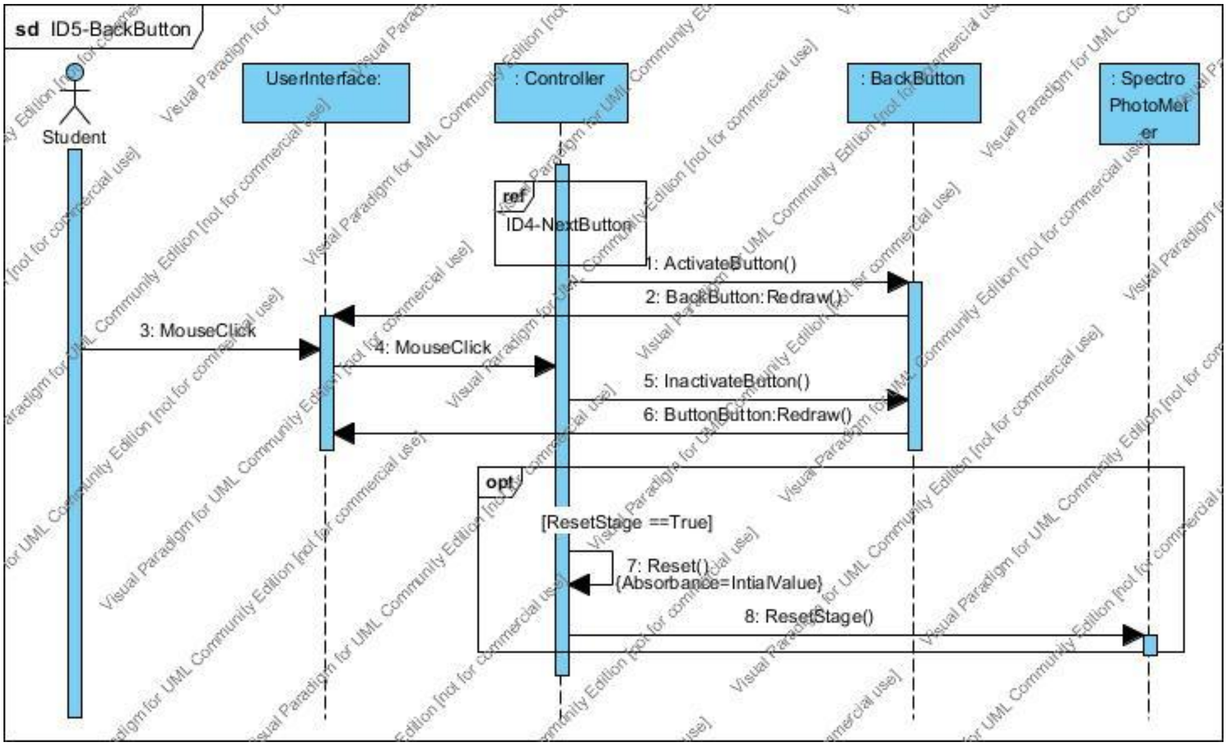


Figure 8. 9 Interaction diagram of BackButton

Figure8.9 shows the interaction diagram of Back Button, the process starts by click the back button, if all the operations are right, it will set ActivateButton() to be 'True'. Then, the interface will show the Next Instruction on the interface.

8.6 ID6 OpenLid

Following diagrams show the Openlid diagram, there are two situations when a user clicks to open the lid.

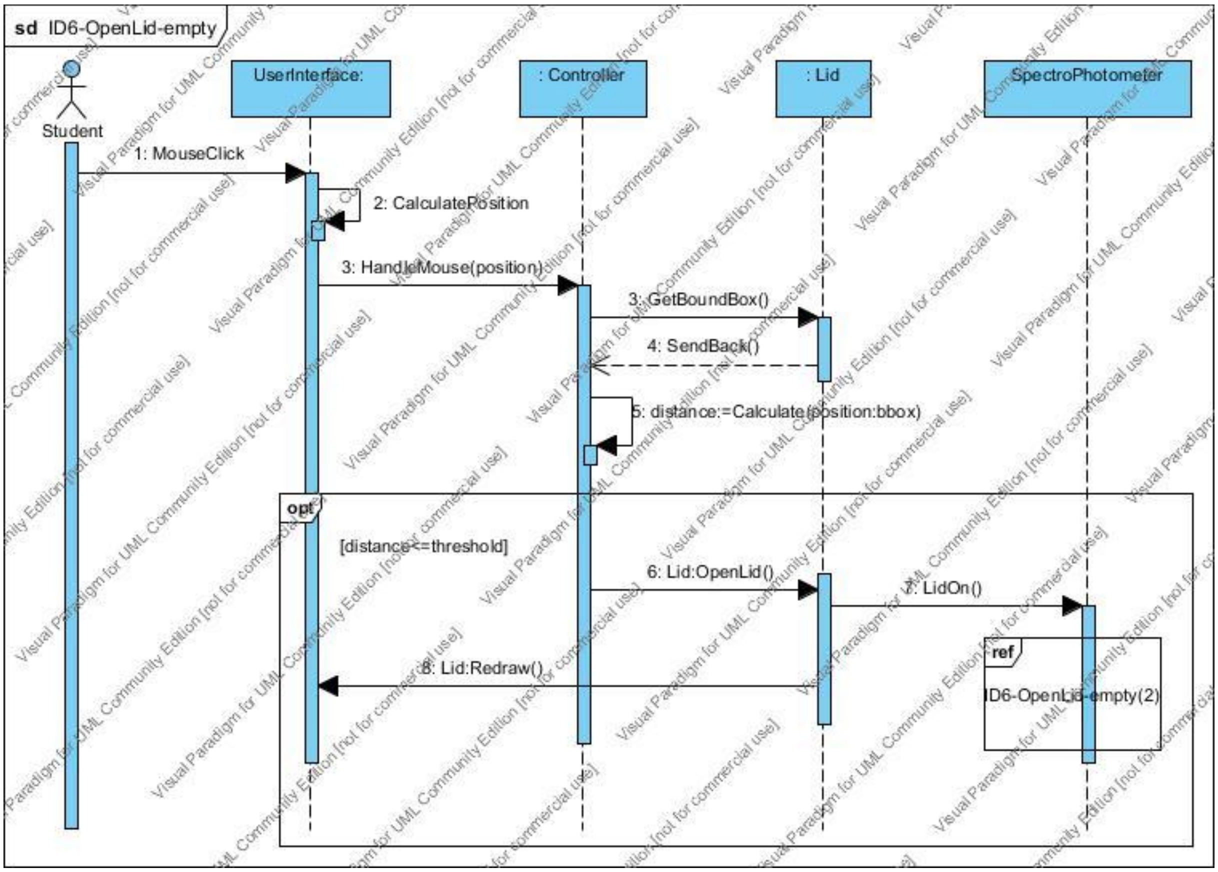


Figure 8. 10 Interaction diagram of OpenLid-Empty

The lid will be open without measurement, when the system is under the OffEmptyClosed state, after this process, the OffEmptyClosed state will turn to state OffEmptyOpen.

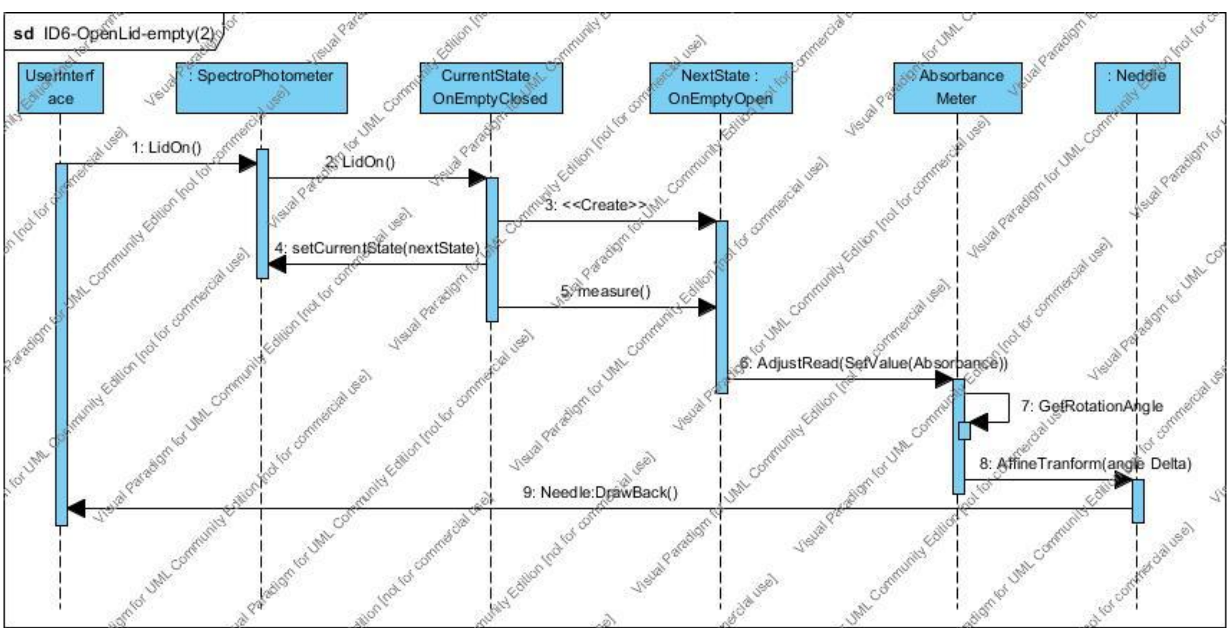


Figure 8. 11 Interaction Diagram of OpenLid (2)

When the system initial state is OnEmptyClosed, Openlid will change the system state to OnEmptyOpen. This causes the spectrophotometer to redo the measurement. The SpectroPhotometer object is assigned the responsibility to calculate by how much AbsorbanceMeter's needle should rotate. The absorbance meter calculates the needle's rotation angle and changes the needle position by AffineTransformation

8.7 ID7 InsertTube

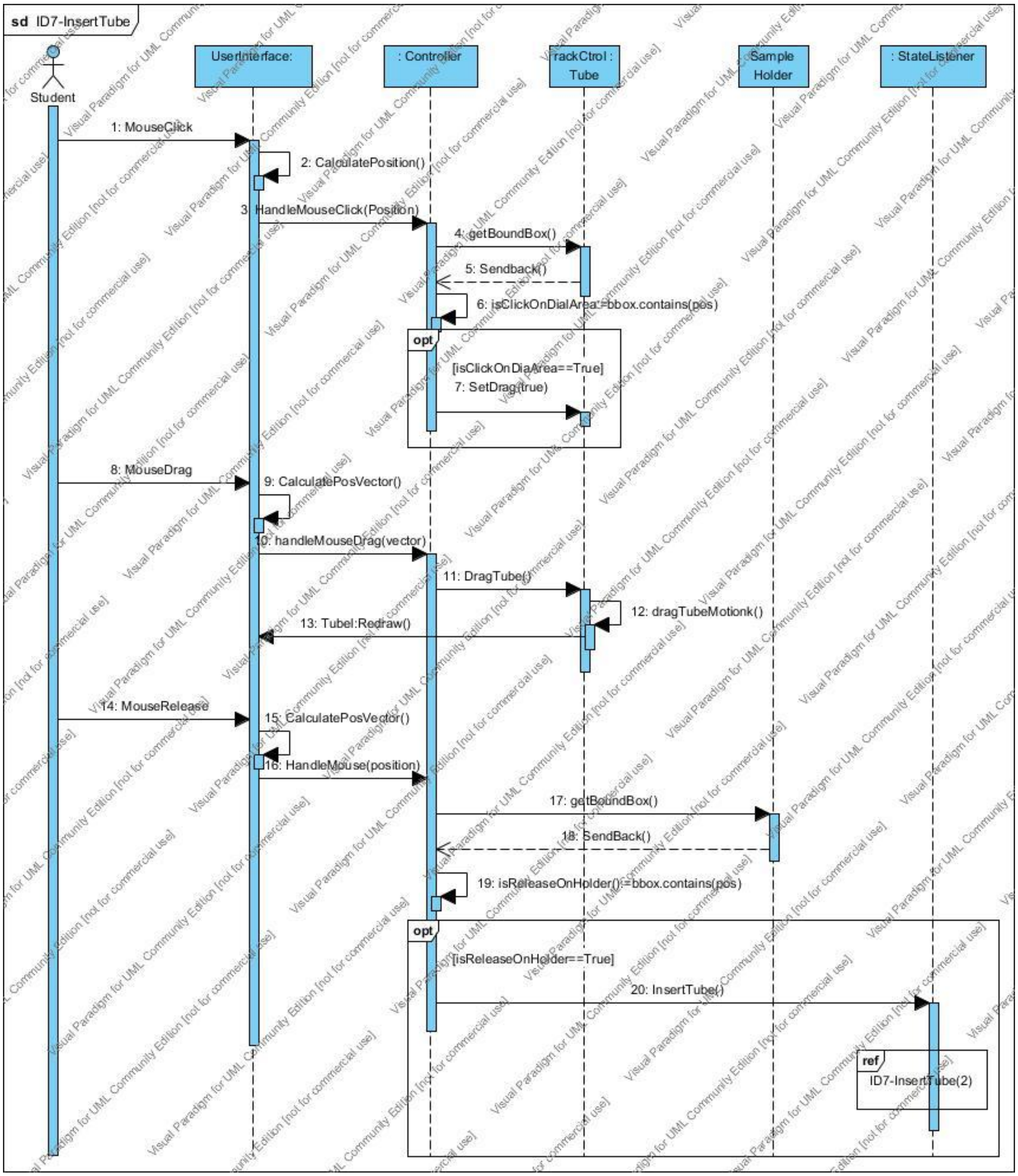


Figure 8. 12 Interaction diagram of InsertTube

The Userinterface calculates the mouse position within the laboratory window, the Controller checks whether the click position is within a threshold distance from sample holder. It will notify other listener to perform the measurement behavior.

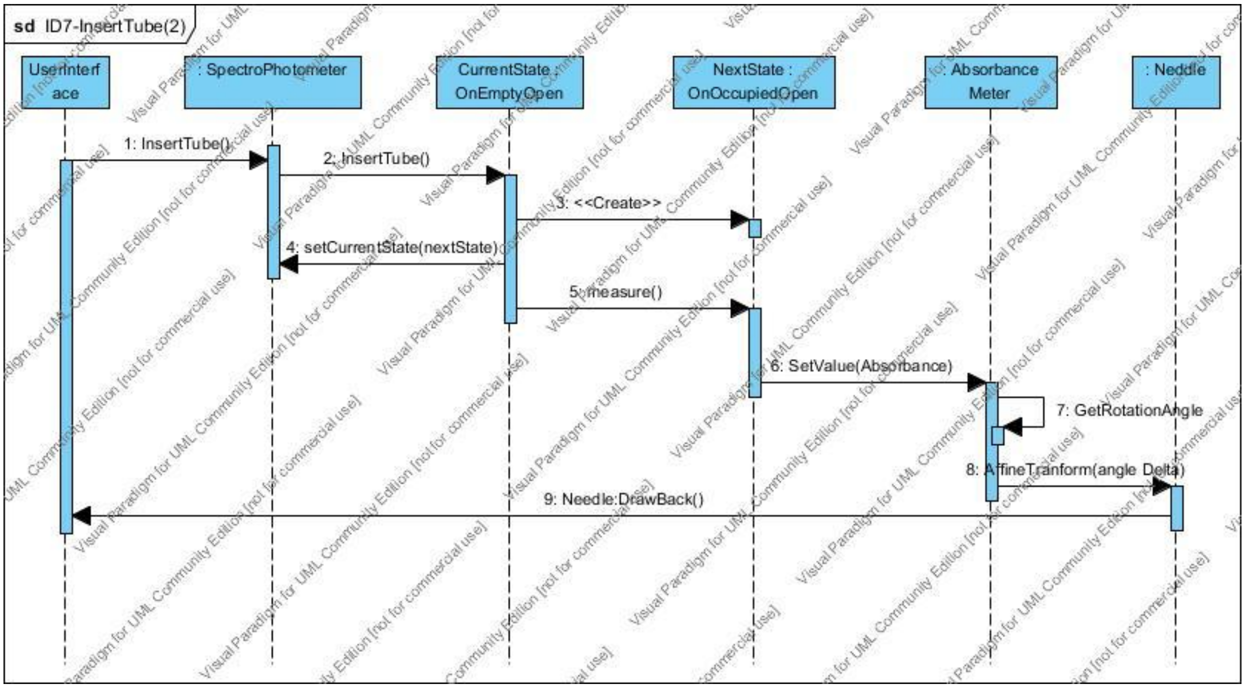


Figure 8. 13 Interaction diagram of InsertTube (2)

Figure 8.13 refers to a fragment shown in figure 8.12. The system's state OnEmptyOpen turns to state OnOccupiedOpen. Based on the state change, the SpectroPhotometer object calculates by how much AbsorbanceMeter should rotate its needle to display the measured wavelength and calls the `transform()` method with rotation transformation parameter.

8.8 ID8 CloseLid

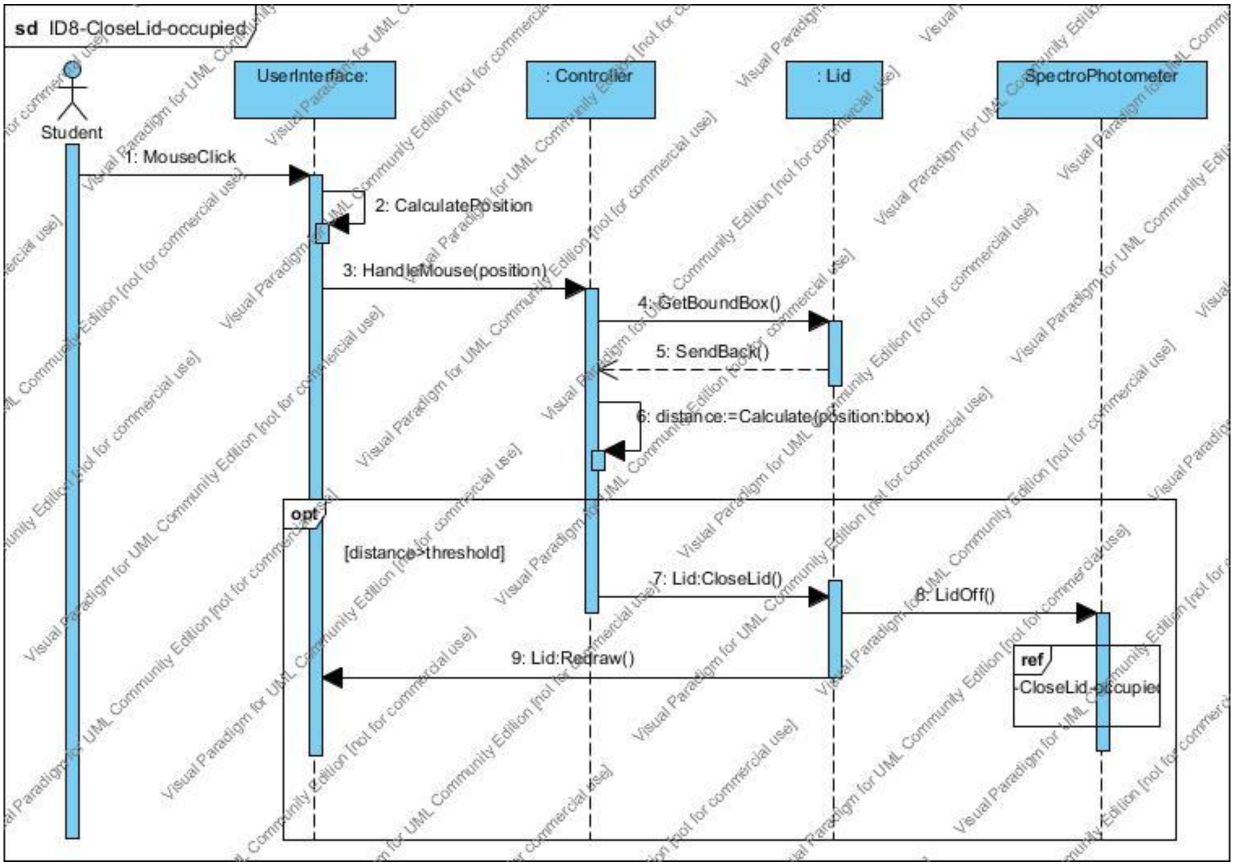


Figure 8. 14 CloseLid-occupied

Figure 8.14 is the diagram show the CloseLid when the sample holder with a test tube in it.

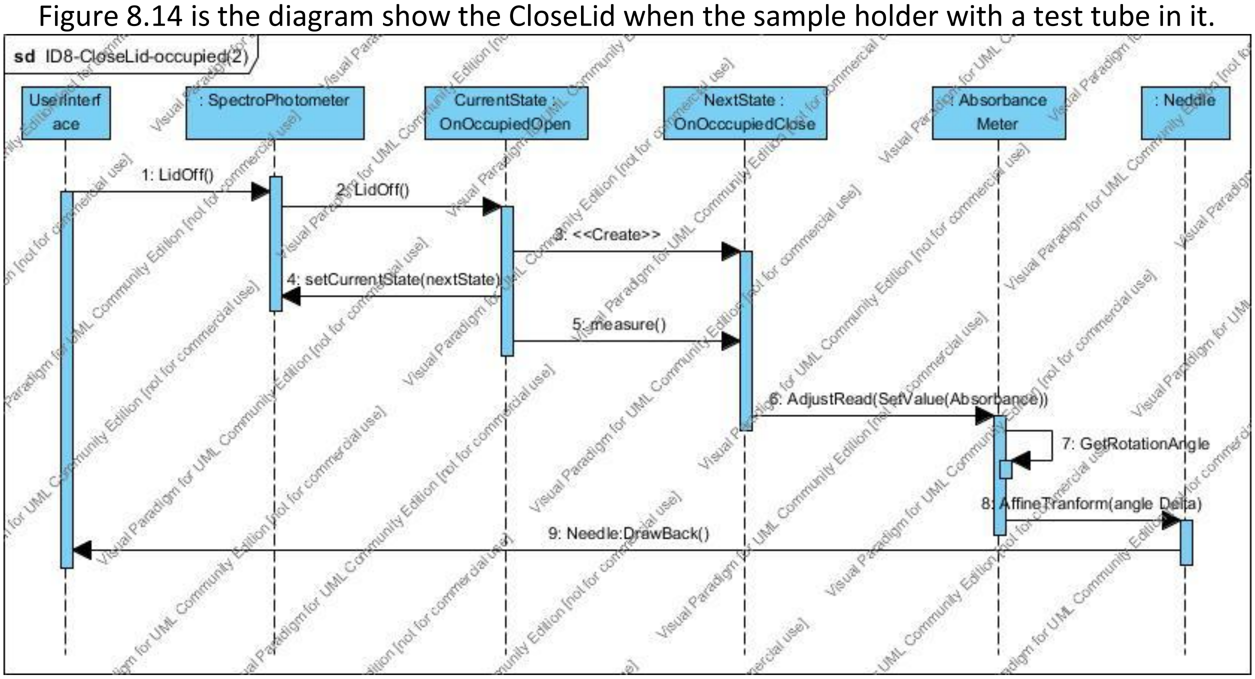


Figure 8. 15 Interaction diagram of CloseLid-occupied (2)

Figure 8.15 is the fragment of Figure 8.14. The system's state OnOccupiedOpen turns to state OnOccupiedClosed. Based on the state change, the SpectroPhotometer object calculates by how much AbsorbanceMeter should rotate its needle to display the measured wavelength and calls the `transform()` method with rotation transformation parameter.

8.9 ID9 AdjustBlank

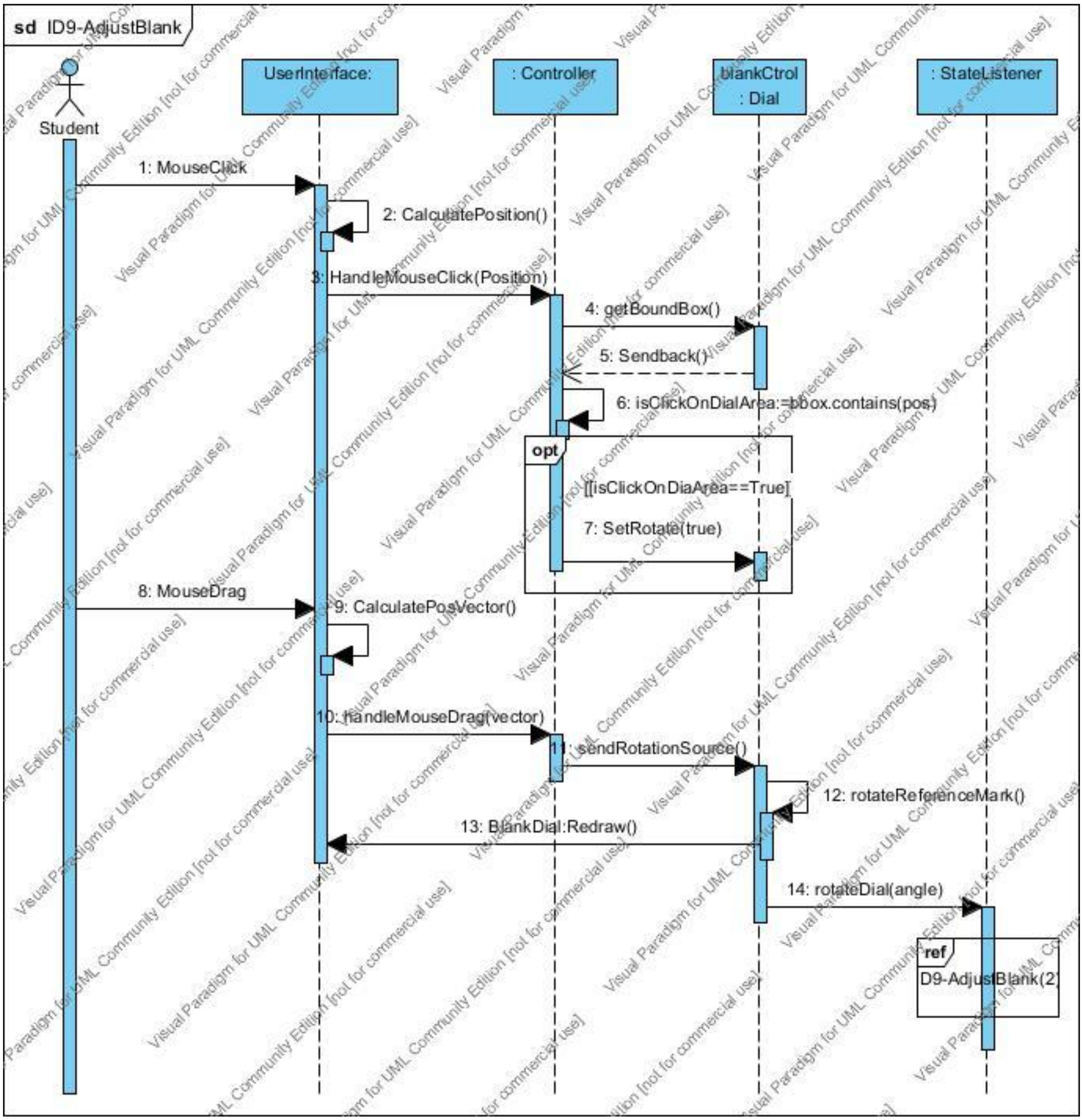


Figure 8. 16 Interaction diagram of AdjustBlank

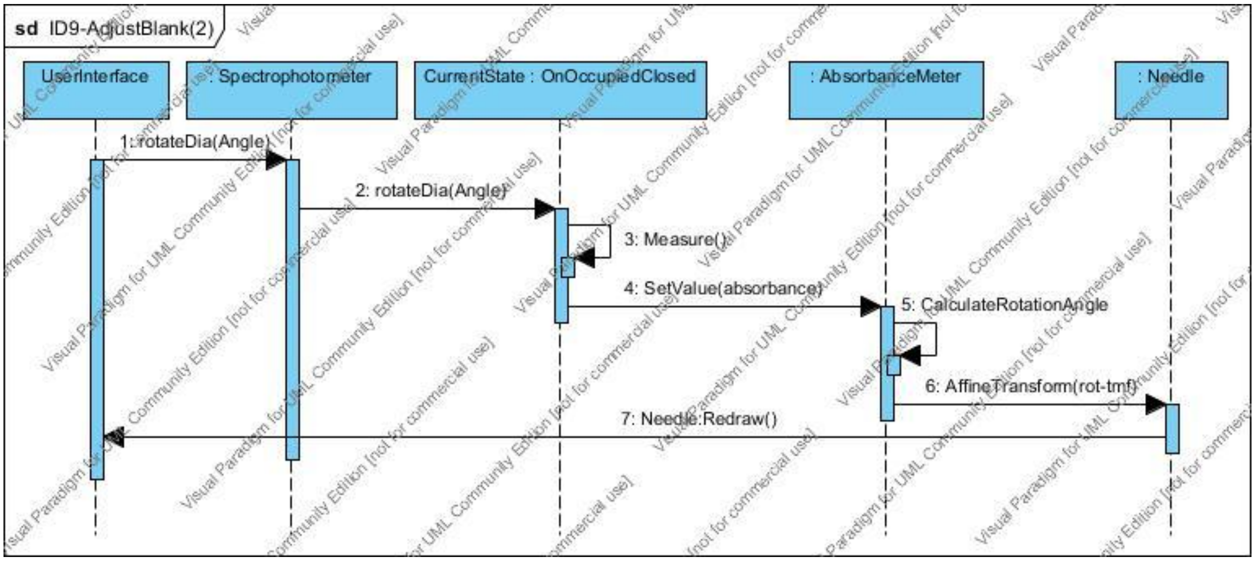


Figure 8. 17 Interaction diagram of AdjustBlank (2)

Figure 8.17 is the fragment of Figure 8.16. The process starts by user clicking in the dial area and dragging the mouse cursor in the desired direction. The system detects if the click is within the wavelength dial's area and if so, sets the wavelength dial attribute isRotated to true. Base on the Blank adjustment, other listeners will be notified a behavior measurement. Then the SpectroPhotometer object calculates by how much AbsorbanceMeter should rotate its needle to display the measured wavelength and calls the `transform()` method with rotation transformation parameter.

8.10 ID10 OpenLid

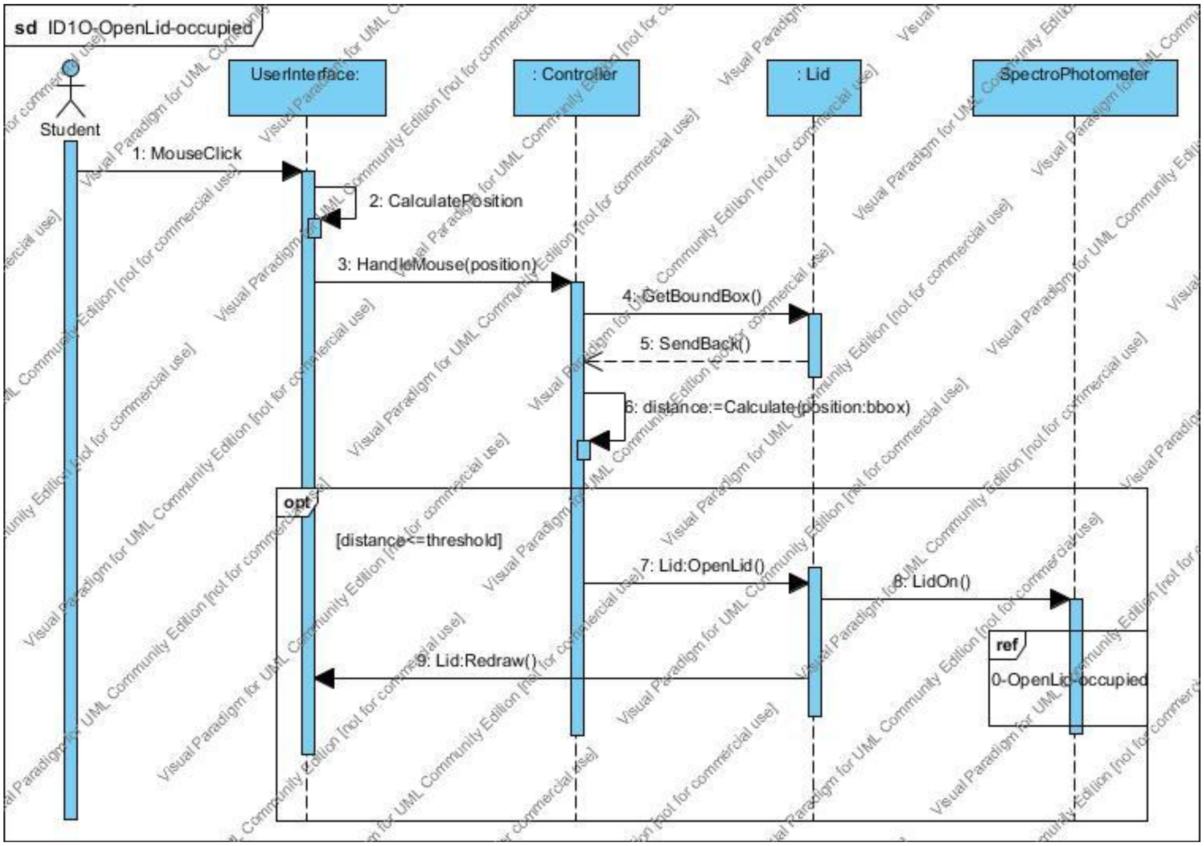


Figure 8. 18 Interaction diagram of OpenLid

Figure 8.18 is the diagram show the OpenLid when the sample holder with a test tube in it.

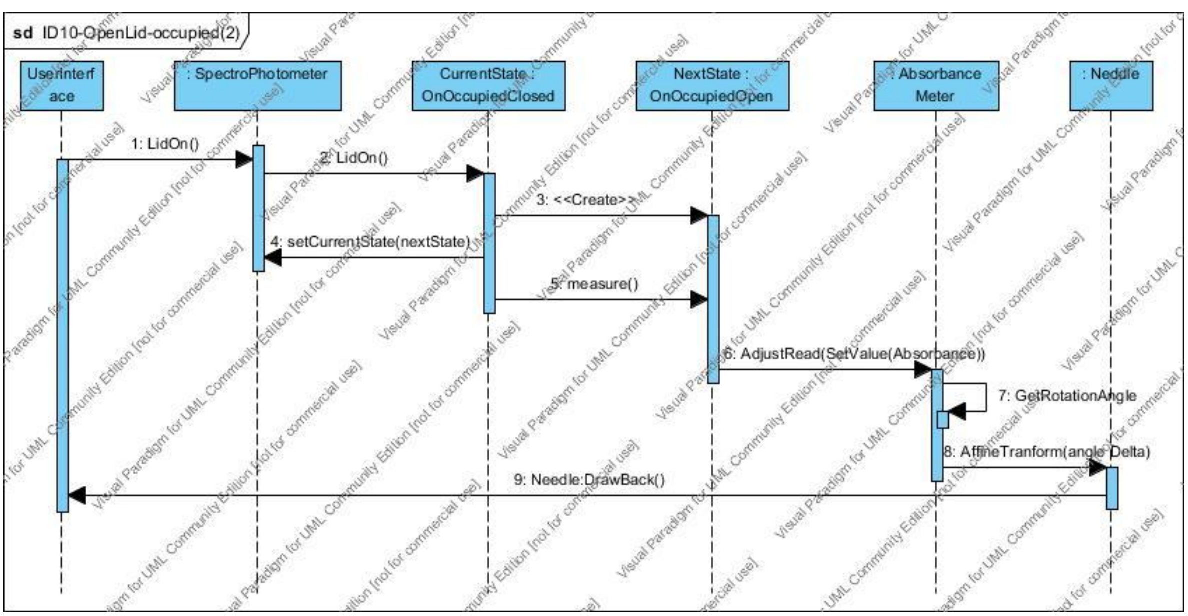


Figure 8. 19 Interaction diagram of OpenLid (2)

Figure 8.19 is the fragment of Figure 8.18. The system's state OnOccupiedClosed turns to state OnOccupiedOpen. Based on the state change, the SpectroPhotometer object calculates by how much AbsorbanceMeter should rotate its needle to display the measured wavelength and calls the `transform()` method with rotation transformation parameter as we mentioned before.

8.11 ID11 RemoveTube

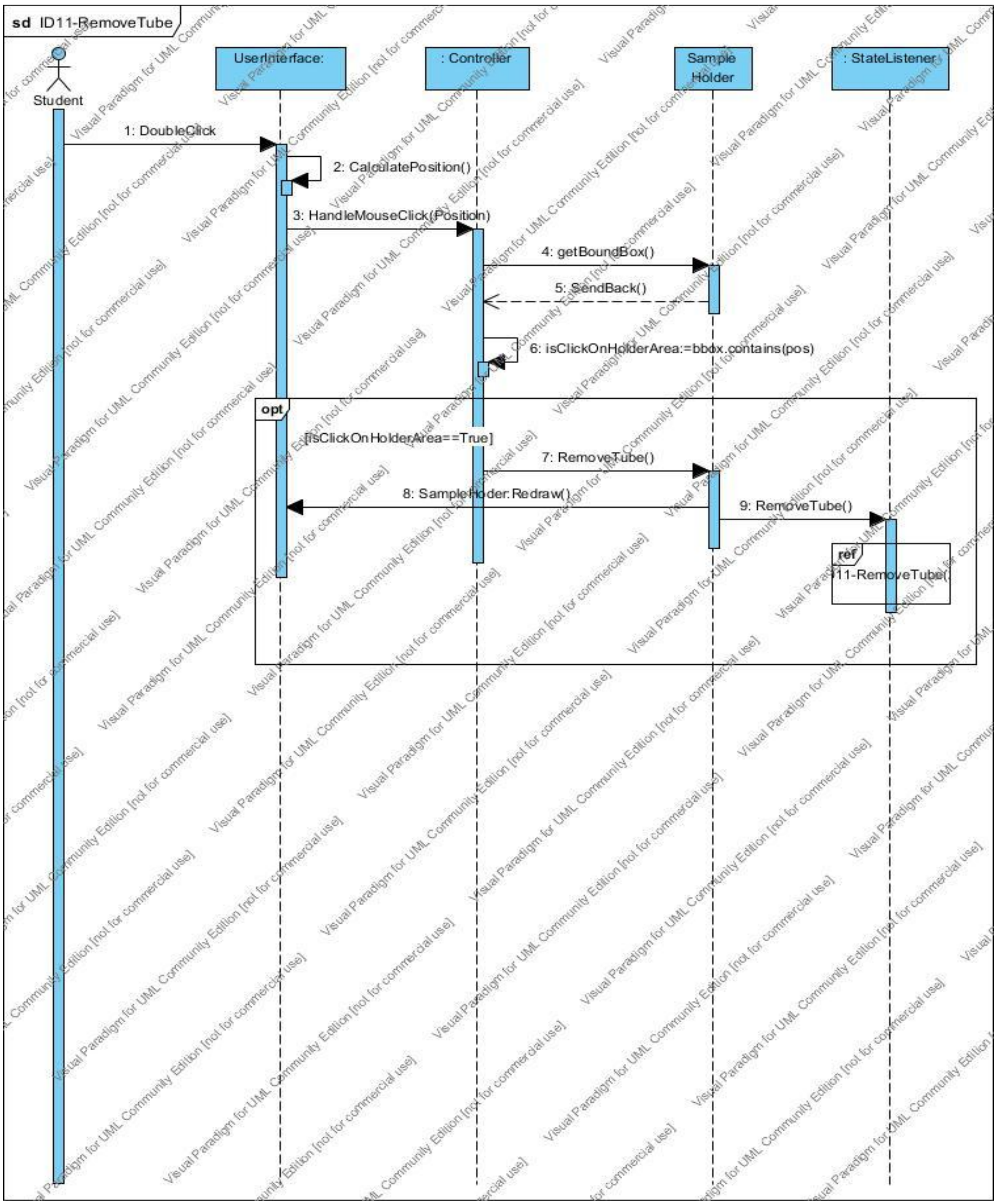


Figure 8. 20 Interaction diagram of RemoveTube

The Userinterface calculates the mouse position within the laboratory window, the Controller checks whether the click position is within a threshold distance from sample holder. If yes, it will notify other listener to perform the measurement behavior.

8.12 ID12 CloseLid

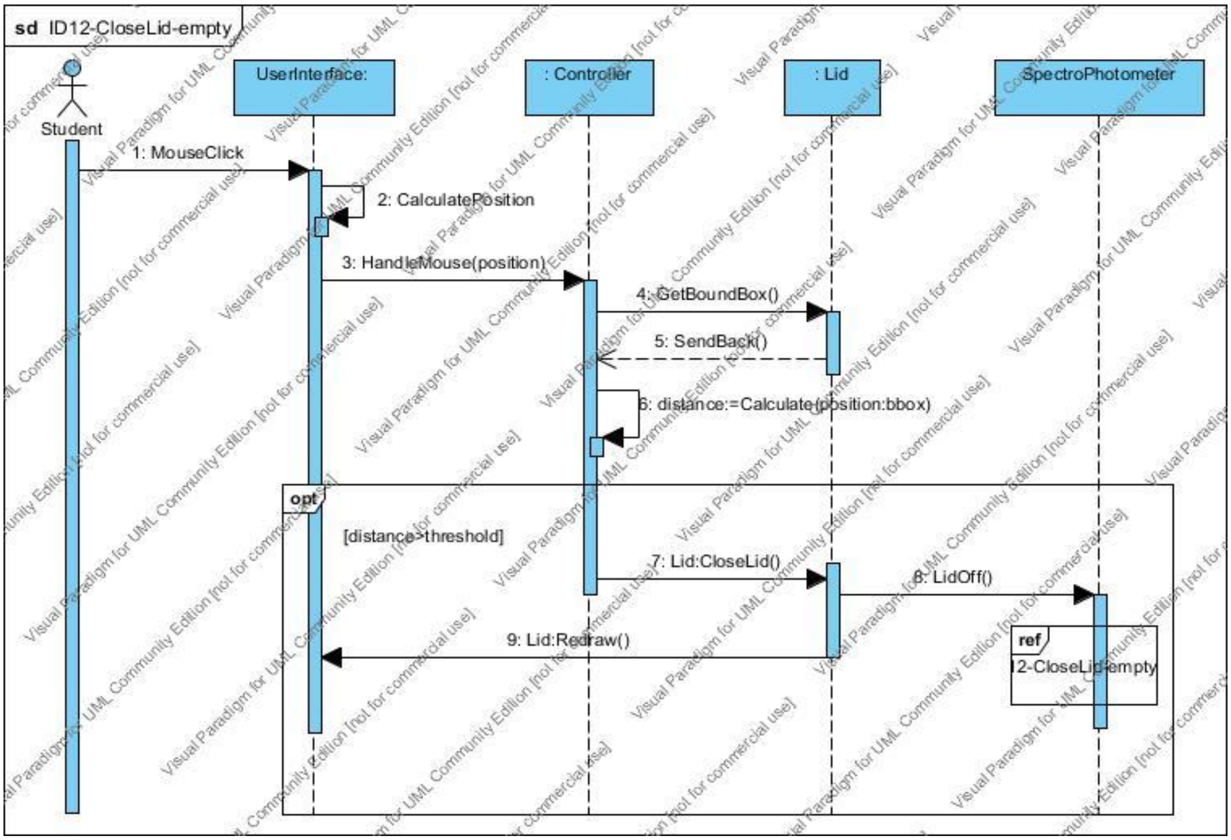


Figure 8. 21 Interaction diagram of CloseLid-Empty

Figure 8.21 is the diagram show the CloseLid when the sample holder is empty.

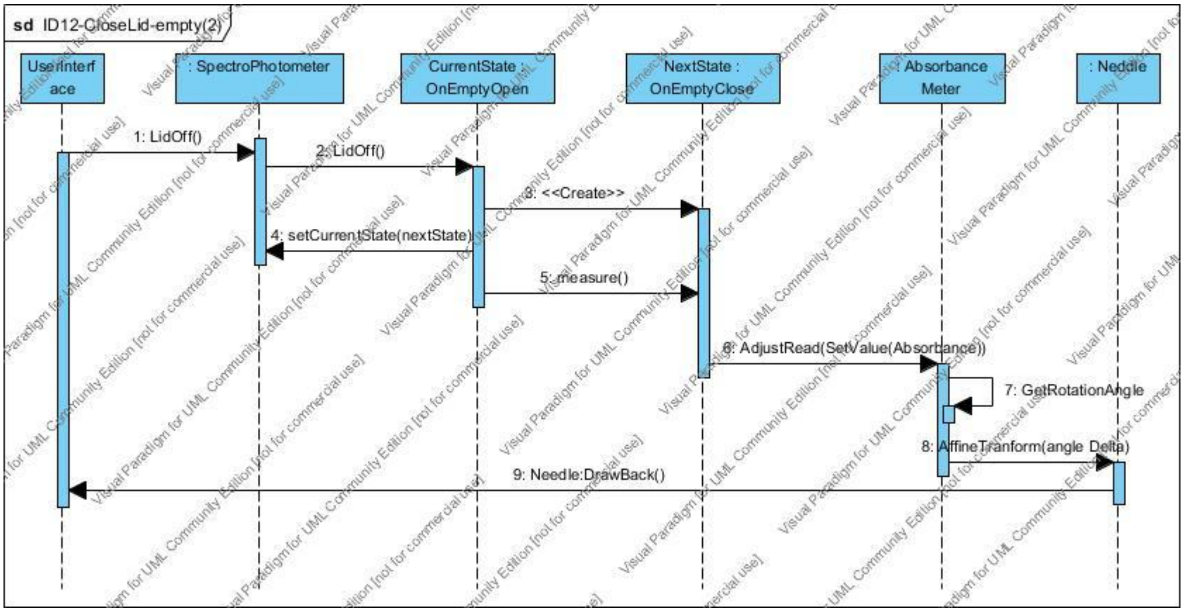


Figure 8. 22 Interaction diagram of CloseLid-empty (2)

Figure 8.22 is the fragment of Figure 8.21. The system's state OnEmptyOpen turns to state OnEmptyClosed. Based on the state change, the SpectroPhotometer object calculates by how much AbsorbanceMeter should rotate its needle to display the measured wavelength and calls the `transform()` method with rotation transformation parameter.

9. Class Diagram and Interface Specification

9.1 Class Diagram

The following diagrams are class diagrams of this software.

Figure 9.1 shows the login mechanism, and classes related to enter teachers' and students' interface.

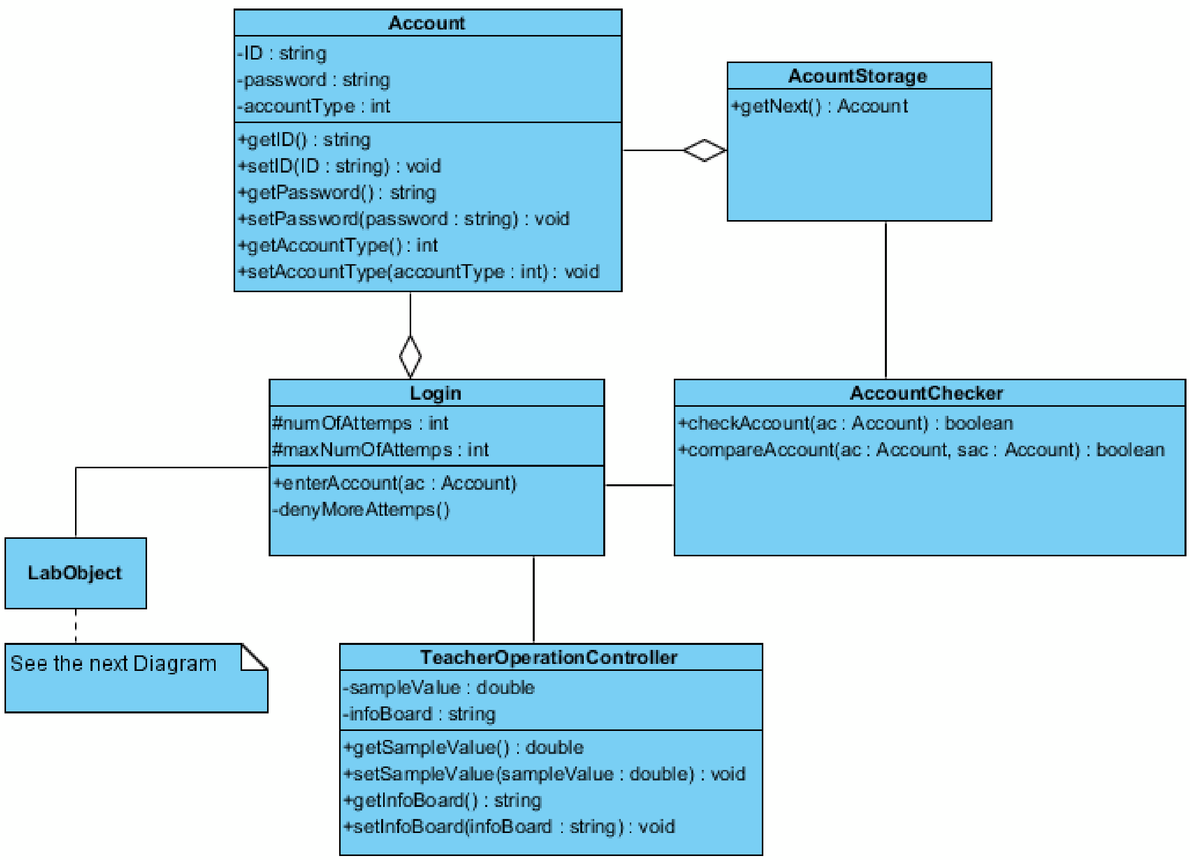


Figure 9. 1 Class diagram #1

Figure 9.2 shows the major components of spectrophotometer, which contain all classes of operations, and the relationship of all objects of the spectrophotometer.

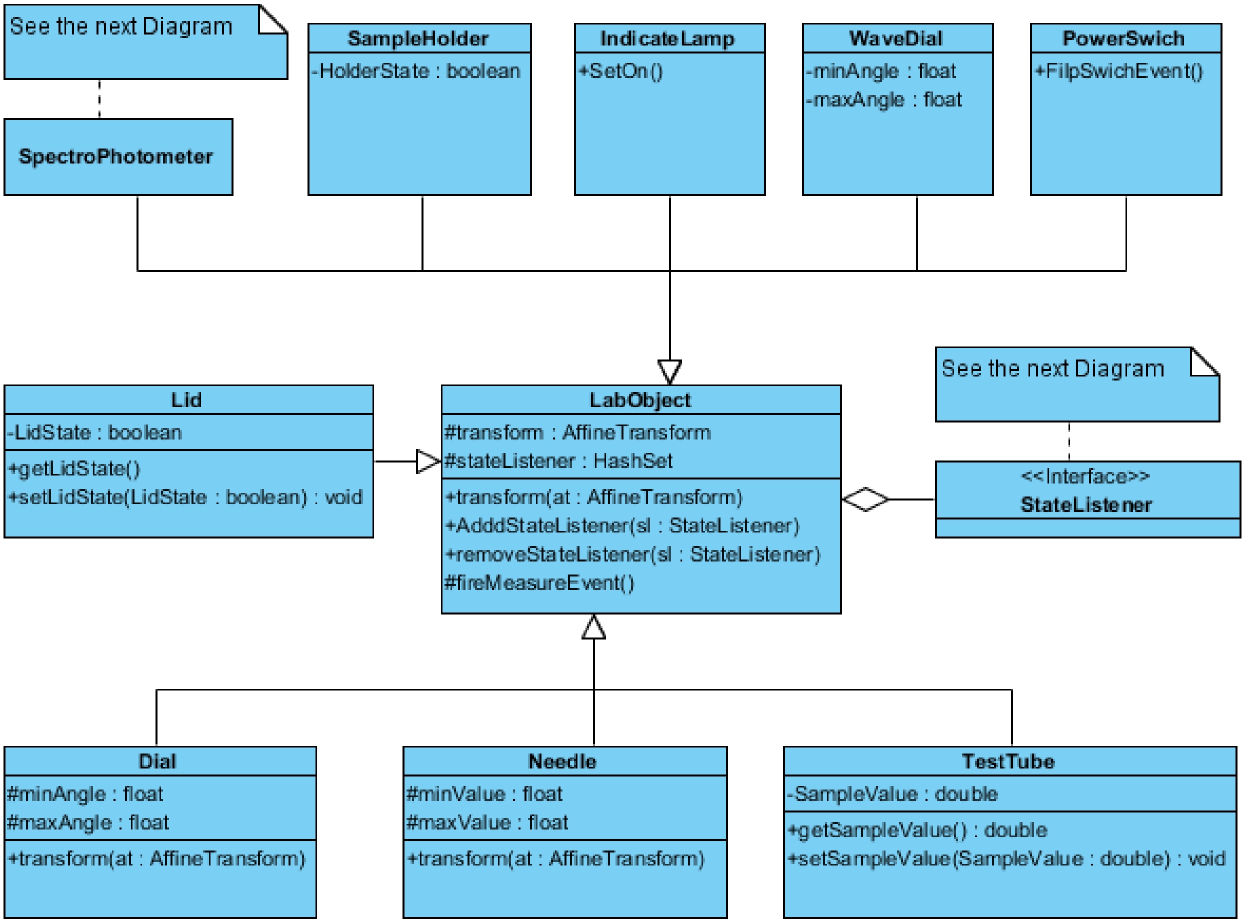


Figure 9. 2 Class diagram #2

Figure 9.3 is the internal structure of spectrophotometer. It includes the part that store states of components. And by using these states, the software can compute the result of the experiment.

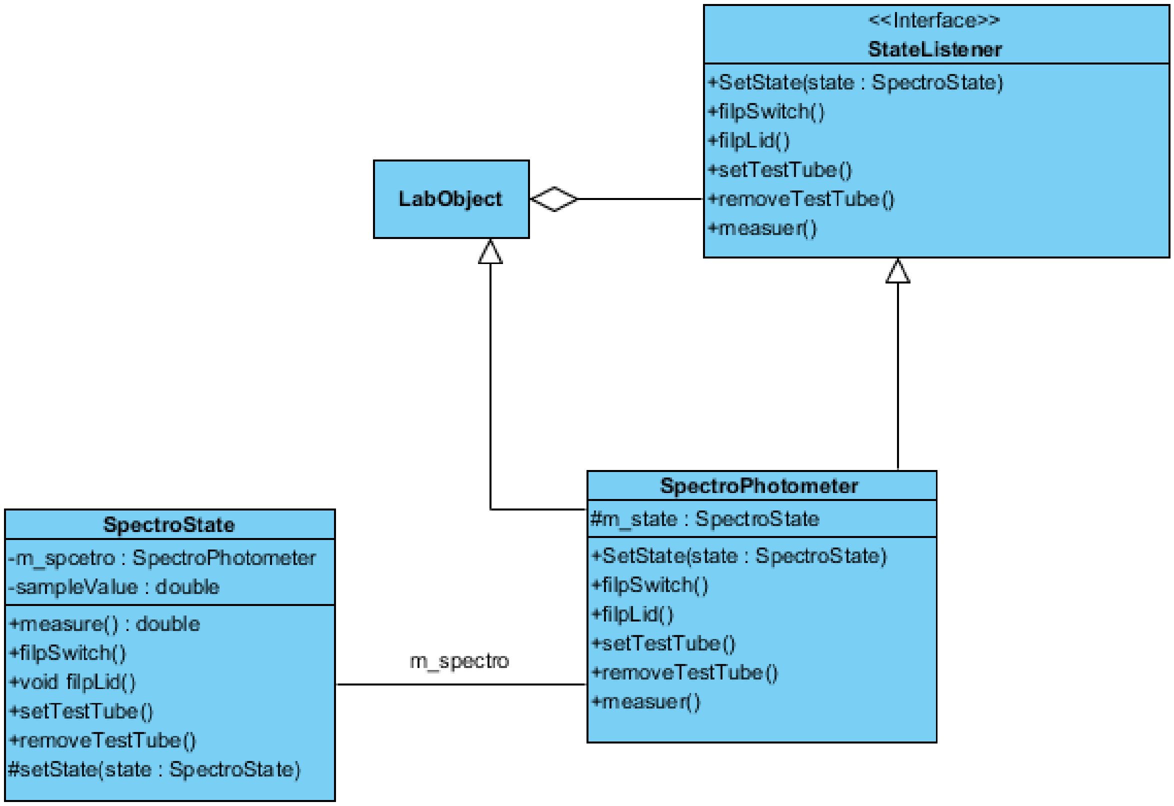


Figure 9. 3 Class diagram #3

9.2 Data Types and Operation Signatures

9.2.1 Account

An Account class (as Figure 9.1 shows) contains three parts, an ID, a password and an accountType. Because all attributes are private, the operations of this class are all Getters and Setters which are used to get and modify these attributes, respectively.

Class Name:	Account	
Attributes:	-ID: string	Account ID
	-password: string	password related to Account ID
	-accountType: int	types of account
Operations:	+getID():string	Getters and Setters
	+setID(ID): void	
	+getPassword():string	
	+setPassword(Password: string)void	
	+getAccountType():int	
	+setAccountType(accountType: int):void	

9.2.2 Login

This class (as Figure 9.1 shows) is used for login.

Class Name:	Login	
Attributes:	#numOfAttempts: int	Number of login Attempts
	#maxNumOfAttempts:int	the max number of login Attempts
Operations:	+enterAccount(ac: Account)	input ID, password and select account type
	-denyMoreAttempts()	system refuses account that reaches max attempts to login

9.2.3 AccountStorage

This class (as Figure 9.1 shows) is used to authorize account. It returns the next Account in the list of Account.

Class Name:	AccountStorage	
Attributes:		
Operations:	+getNext(): Account	move to the next Account and return it

9.2.4 AccountChecker

It is the class (as Figure 9.1 shows) that checks for the account validity. The mechanism of authorization is compare the Account input by user with each Account stored in the List. If there is an Account exists in the AccountStorage, it returns 1, otherwise, it returns 0.

Class Name:	AccountChecker	
Attributes:		
Operations:	+checkAccount(ac:Account):Boolean	check the format of the Account input
	+compareAcccount(ac:Account,sac:Account):boolean	compare the Account user input with Account got from AccountStorage

9.2.5 TeacherOperationController

This class (as Figure 9.1 shows) is designed teachers who can modify some value and information of the system.

Class Name:	TeacherOperationController	
Attributes:	-sampleValue: double	The sample's concentration value
	-infoBoard:string	information that will be posted on the infoBoard
Operations:	+getSampleValue():double	Getter and Setter
	+setSampleValue(sampleValue:double):void	
	+getInfoBoard():string	

	+setInfoBoard(infoBoard:string):void	

9.2.6 LabObject

This is an abstract base class for all parts of the spectrophotometer (shown as Figure 9-2). This is a graphical figure base class that can be manipulated by mouse pointer. The manipulations include click and drag. A dragging manipulation is converted into an affine transformation that will be applied to the corresponding figure.

Class Name:	LabObject	
Attributes:	# transform : AffineTransform	Represents the current affine transformation of the object, relative to its non-transformed prototype.
	# stateListeners : HashSet	Listener objects interested in state changes of this lab object.
Operations:	+ transform(at : AffineTransform)	Applies an affine transformation to this object.
	+ addStateListener(sl : StateListener)	Adds the listener slto the list of listeners.
	+ removeStateListener(sl : StateListener)	Removes the listener sl.
	# fireMeasureEvent( )	Calls the method measure()on all listener objects.

All objects are in a non-transformed prototype, which means zero translation, zero rotation, and scale is 1 for both x and y dimension (i.e., resized to 100%). To change the object's translation, rotation, or scale (resizing), just call the method `transform()` with the appropriate transformation parameter.

```
protected transient HashSet stateListeners;

protected void fireMeasureEvent()

{

    HashSet listeners = (HashSet) stateListeners_.clone();

    for (Iterator i = listeners.iterator();

        i.hasNext(); ) {

        ((StateListener)i.next()).measure();

    }

}
```

9.2.7 PowerSwitch

Extends LabObject

This object is on/off switch (shown as Figure 9.2). It turns on or off the IndicateLamp when the spectrophotometer is turned on or off (see section 9.2.8). When the switch is “OFF”, the device cannot work, which means that the indicateLamp is off, and, in any case, the needle on the meter cannot move.

Class Name:	PowerSwitch	
Attributes:		
Operations:	on mouse click	When mouse pointer is clicked on this object, this method calls fireFlipSwitchEvent().
	# fireFlipSwitchEvent( )	Calls the method flipSwitch()on all listener objects.

The list of listeners used in the method `fireFlipSwitchEvent()` is inherited from the base LabObject.

9.2.8 IndicateLamp

Extends LabObject, shown as Figure 9.2.

The indicateLamp (also known as ON/OFF indicator) indicates when the spectrophotometer instrument is turned on (using the on/off switch). It is a part of the absorbance meter. The light is shown as a green oval in the upper right corner of the absorbance meter scale.

Class Name:	indicateLamp	
Attributes:		
Operations:	+ setOn( )	Turns the green light ON or OFF.

9.2.9 Dial

Extends LabObject, shown as Figure 9.2.
There are two dial knobs on the front side of spectrophotometer:
(a) The zero-control dial (also known as “AdjustZero”).
(b) The light control dial (also known as “AdjustBlank”).

Class Name:	Dial	
Attributes:	# minAngle : float # maxAngle : float	Dial's minimum rotation angle. Dial's maximum rotation angle.
Operations:	+ transform(at : AffineTransform)	Applies a rotation transformation.

A knob is shown graphically as a figure consisting of a circle and a line, which represent the knob and the reference mark.

The operation `transform()` allows the user to rotate the knob and set it in the desired position. This operation first calls the operation `transform()` on LabObject (its superclass), and then it informs all the listeners that a measurement should be performed for the new position of the dial. The operation `fireMeasureEvent()` is described in Section 9.2.6.

The code transform(at : AffineTransform):

```
{
   super.transform(at);
   fireMeasureEvent();
}
```

Behavior "measuring" performs the calculations based on the solution's density and the light wavelength and calls the `transform()` method on the instrument needle (described in Section 9.2.12) to display the wavelength. It also turns on or off the pilot lamp when the spectrophotometer is turned on or off. The knob "turning" behavior causes the "measuring" behavior to redo the measurement when a dial is rotated. Similarly, the lid "opening" behavior causes the "measuring" behavior to redo the measurement when the sample holder's lid is opened or closed (described in Section 9.2.13).

9.2.10 TestTube

Extends LabObject

It has an attribute named SampleValue (shown as Figure 9.2), which can be modified by teacher in teachers interface.

Class Name:	TestTube	
Attributes:	-SampleValue:double	Concentration of the sample in the test tube
Operations:	+getSampleValue():double	Getter and Setter
	+setSampleValue(SampleValue: double):void	

9.2.11 WaveDial

Extends LabObject

This object (shown as Figure 9.2) is the wavelength control dial for setting the color of the illumination light. Because this object is normally seen in a side projection but the main view is from the top, there will also be a magnified view of the wavelength dial (shown as top view).

Class Name:	WaveDial	
Attributes:	# minAngle : float # maxAngle : float	Dial's minimum rotation angle. Dial's maximum rotation angle.
Operations:		

9.2.12 Needle

Extends LabObject

This class is the object that displays result in term of spinning the needle by calculation, shown as Figure 9-2.

Class Name:	Needle	
Attributes:	-minAngle:float	shows the smallest value on the meter
	-maxAngle:float	shows the largest value on the meter
Operations:	+transform(at: AffineTransform)	spin the needle

9.2.13 Lid

Extends LabObject

The Lid has two states open and close. The state is highly related to the correctness of result. When the lid is open, measuring the concentration will cause the incorrectness because some day light may leak in to the sample holder. The states of the lid are changed between open and close by clicking the lid.

Class Name:	Lid	
Attributes:	-LidState: boolean	the lid is open or close
Operations:	+getLidState():boolean	Getter and Setter
	+setLidState(LidState: boolean):void	

9.2.14 SampleHolder

Extends LabObject

This object is the sample holder for the test tube. It is a hole where the test tube is inserted for measuring the light absorbed by the substance contained in the test tube. There is also a lid on top of the sample holder (See Section 9.2.13). Several facts should be observed:

• The sample holder can be either empty or hold one test tube
• The holder's lid can be open (lifted) or closed
• The test tube cannot be inserted or removed when the lid is closed
• The measurement should not be correct while the lid is open, because the external light will interfere with the measurement light that illuminates the test tube.

Class Name:	SampleHolder	
Attributes:	-HolderState:boolean	the states of the sample holder, open or close
Operations:		

9.2.15 SpectroState

The following events can be identified for user interaction with the spectrophotometer:

1. Flip the power switch ON or OFF Event: "flipSwitch"
2. Flip the lid of the sample holder OPEN or COSED Event: "flipLid"
3. Rotate the dial knob
   Event "measure"
4. INSERT or REMOVE the test tube to/from the sample holder
   Events: "setTestTube" and "removeTestTube"

   There are three state variables that define the spectrophotometer state:
   1. Power switch value: ON or OFF
   2. Sample holder's lid value: OPEN or CLOSED
   3. Sample holder's occupancy value: OCCUPIED or EMPTY

Based on these state variables, we define the following states (Table 9.1) of the spectrophotometer:

Table 9. 1 States of Spectrophotometer

State Name	State Variable Values	Java Class
OnEmptyClosed	{Switch=ON, Sample-holder=EMPTY, Lid=CLOSED}	OnEmptyClosed.java
OnEmptyOpen	{Switch=ON, Sample-holder=EMPTY, Lid=OPEN}	OnEmptyOpen.java
OffOccupiedClosed	{Switch=OFF, Sample-holder=OCCUPIED, Lid=CLOSED}	OffOccupiedClosed.java
OffOccupiedOpen	{Switch=OFF, Sample-holder=OCCUPIED, Lid=OPEN}	OffOccupiedOpen.java
OffEmptyClosed	{Switch=OFF, Sample-holder=EMPTY, Lid=CLOSED}	OffEmptyClosed.java

OffEmptyOpen	{Switch=OFF, Sample-holder=EMPTY, Lid=OPEN}	OffEmptyOpen.java
OnOccupiedClosed	{Switch=ON, Sample-holder=OCCUPIED, Lid=CLOSED}	OnOccupiedClosed.java
OnOccupiedOpen	{Switch=ON, Sample-holder=OCCUPIED, Lid=OPEN}	OnOccupiedOpen.java

Class Name:	SpectroState	
Attributes:	m_spectro : SpectroPhotometer number: double	 The number shown by default is 5.67

Operations:	+ measure( ) : double
	+ flipSwitch( ) // abstract
	+ flipLid( ) // abstract
	+ setTestTube( ) // abstract
	+ removeTestTube( ) // abstract
	# setState(state : SpectroState)
		Returns the current value of "number".

Note: this is an abstract class. Any specific state has a class that extends the SpectroState in order to get the result.

```
protected  void setState (SpectroState state)
{
        m_spectro.setState(state);
        m_spectro.measure();
}
```

10. System Architecture and System Design

10.1 Architectural Styles

The main components in our application are the interface and the data processing part. The interface is designed for displaying a virtual Spectrophotometer, providing all the operation dials and returning results of the experiment. While the data processing part is responsible for the user's commands and running the right subsystem model. The architectural styles of our application use service-oriented architectures. Figure 10-1 gives the detailed illustration.

d

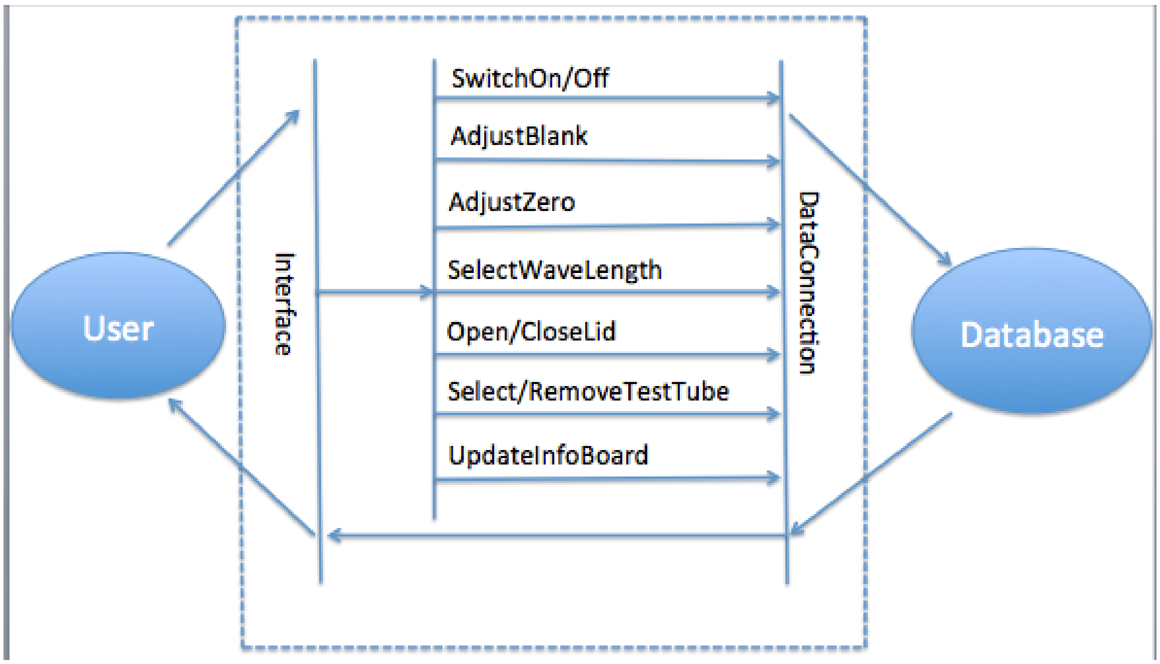


Figure 10. 1 Service-Oriented Architectural Styles of Virtual Spectrophotometer

10.2 IdentifyingSubsystems

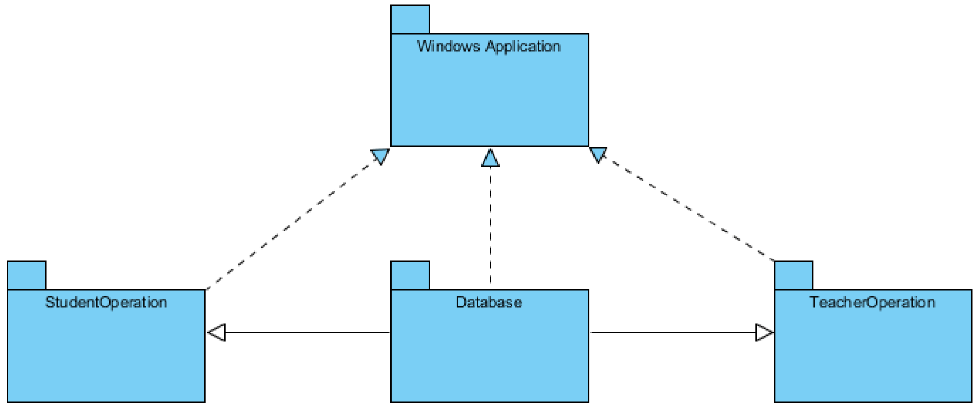


Figure 10. 2 Subsystems

Figure 10.1 shows the subsystems in our project. They are StudentOperation, Databse and TeacherOperation.

The first subsystem is StudentOperation Subysystem. It presents a real spectrum like interface to students. Aiming at make student become more familiar with spectrum, this subsystem allows students have operations like switch on, adjust to blank, insert test tube and so on. As for its purpose, it is not a linear process system. It allows students make mistakes which will result in inaccurate result of test sample.

The second subsystem is named Database. As its name, its main function is to maintain the data interaction between interface and database. When students behave like (open/close lid, insert/remove test tube), the information (lid status tag, tube status tag and so on) will be updated in database via this subsystem. Furthermore, this subsystem deliver teacher updated information (like information board and sample value) as well.

The third subsystem is TeacherOpeartion. It allows teacher to update information board which displays tips for students. Also teacher can set up sample value in this subsystem, so that they can see whether student can use this virtual device in a correct way.

10.3 Mapping Subsystems to Hardware

Our system need to store data into database in order to ensure the normal interaction between users and interface. The database can be divided into two parts which are login database, experiment database.

Login database provide a way to valid the username and password of both student and teacher, and update the system status which decide next interface will be students or teachers interface.

When students log into the students interface, they will see our virtual device and do the experiment steps. Then these steps will generate a corresponding data, after that these data will be sent to the experiment database and update the database value. DataCalculator will get this data from experiment database. By comparing the data table, the DataCalculator will find the final result which was influence by sequence of operation. As a teacher, when he login our system, he will see the administration interface. On this interface he can change the value of our test tube meanwhile he can update the database immediately.

We choose flat (table) model as our database model. The flat (or table) model consists of a single, two-dimensional array of data elements, where all members of a given column are assumed to be similar values, and all members of a row are assumed to be related to one another. In our login database Columns are for account type, ID and password (shown as table 10-1). Each row would have the specific password associated with an individual user. Columns of the table often have a type associated with them, defining them as character data, date or time information, integers, or floating point numbers.

Account type	ID	password
1	00001	aaaa
1	00002	aaaa
……	……	……
2	147001234	abcd
2	147003456	Abcd1234

Table 10. 1 Account

In database for experiment columns represent for adjustZero, adjustWavelength, adjustBlank, lipStatus and tubeValue. Each row will have the specific value associated with different operation. Table 10.2 illustrates data model.

Table 10. 2 Database Structure

	adjustZero	adjustWavelength	adjustBlank	lipStatus	tubeValue
147001234	1	380	1	1	580
147007864	0	450	1	1	390
147004532	1	350	0	0	750
……	……	……	……	……	……

10.4 Persistent Data Storage

We design our system as an event-driven system that user can do whatever they want in the real lab. Each event will change the associated value in the database as we mentioned before. In real lab, the sequence of operations has great influence on experiment result. Identifying these different sequences and responding reasonable results are big issues when considering the execution orders. Each event may change the value in database. However, the value will be not only depended on the kind of event, but also on the sequence of the event. For instance, when you first adjusting zero then selecting the tube and first selecting the tube then adjusting zero, both methods can generate the value updating in database. But the value is totally different, because the second sequence will lead to a correct result and the first sequence will not.

10.5 Hardware Requirements

Since this is a small application developed by using Java, a computer that supports JRE can run this application. On the official website of Java, we found requirement of java are described as "*Intel and 100% compatible processors are supported. A Pentium 166MHz or faster processor with at least 64MB of physical RAM is recommended. You will also need a minimum of 98MB of free disk space*". Therefore, we take this as the Hardware Requirements for this application.

11. User Interface Design and Implementation

11.1 GUI overview

As we mentioned before, three interfaces compose our virtual device, and they are login interface, and student interface and teacher interface.

Figure 11. 1 LohinGUI

The login interface consists three parts as it is showed in the figure11-1. The first part is the blanks of the username and password as it is usually be. Account selection and two buttons are the second part and last part. We design the login interface as simple as it could be in order to reduce the user effort. In the second demo we will add background on the interface and make it more beautiful.

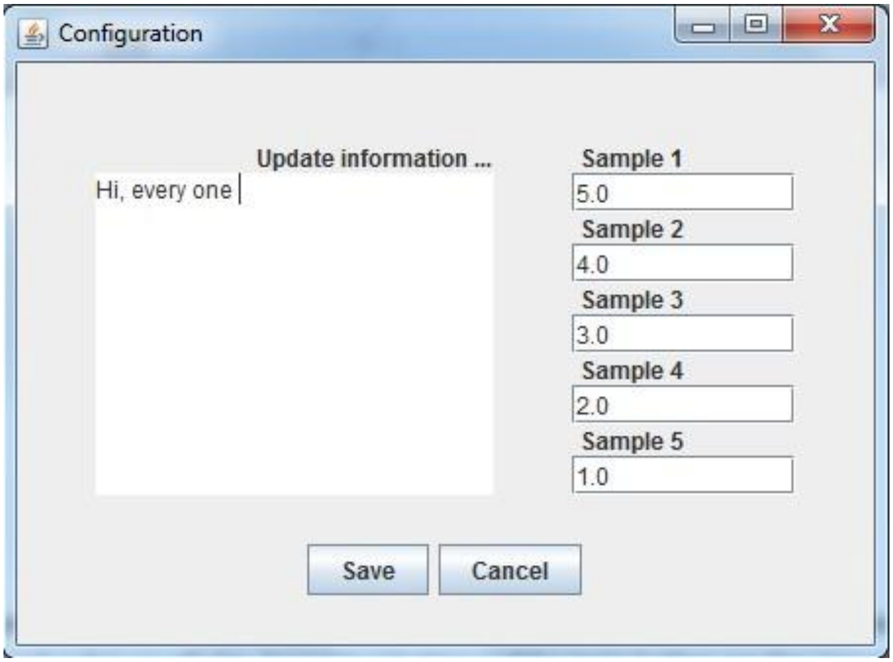


Figure 11. 2 TeacherCUI

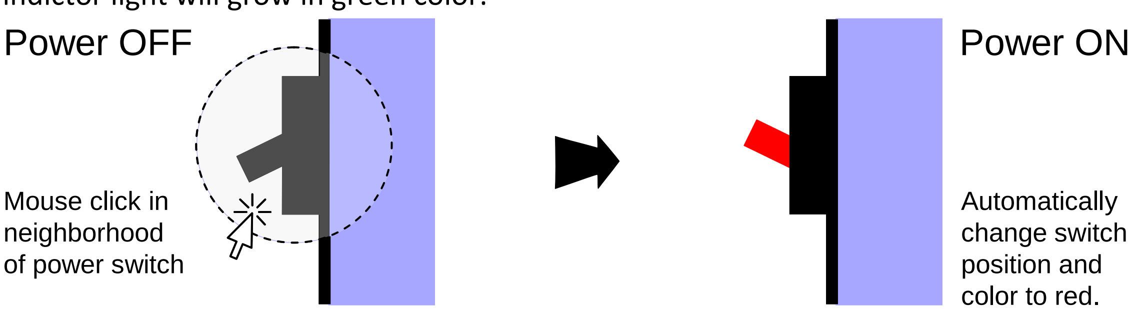
When entering the teacher interface, you will see the teacher interface. This interface is used for updating the information board and setting the test value. Compared with the teacher interface we designed before, the new one changes the distribution of blanks and make it more friendly and accessible. We change the button 'reset' with the button cancel. Because by pressing backspace you can easily delete the word you just typed in. This allows you to retype in a word instead of deleting a sentence.

Figure 11. 3 StudentGUI

Student interface design is the most difficulty and importance among the three interfaces. According to our design principle, we will design our virtual machine as students see in the real lab. Unfortunately, in the first demo, we change the meter with a screen and the tube area with a scroll pane because of the time limits. We has realize the first step that trying to realize what students do in the real lab, and in the second demo, we focus on realizing what students see in the real lab. These include the knob animation and the meter design. We haven't reduced the user effort, since we design our user effort according to what they do in the real lab.

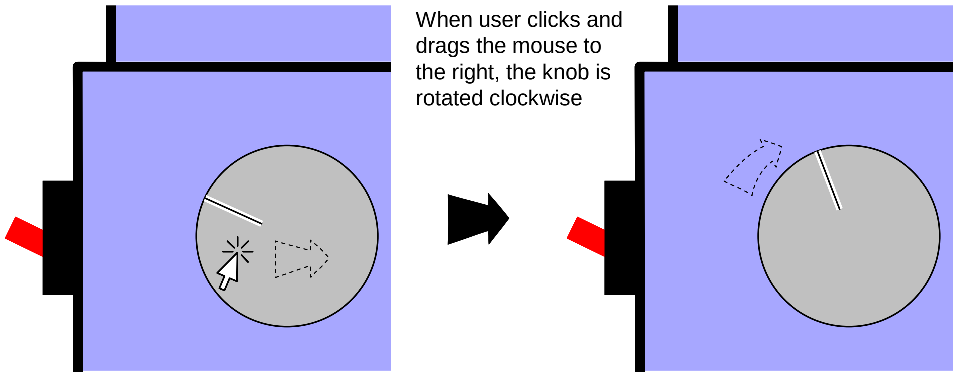
11.1.1 Power Switch

Initially, the device is turned off and the power switch, when you turn on the machine, the indictor light will grow in green color.



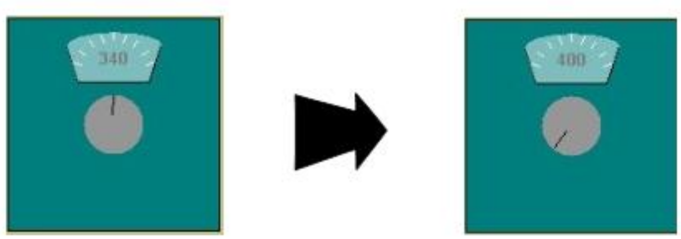
indictor turn to green

11.1.2Adjust Zero Control Dial



A dial knob can be rotated to set the desired value. If the user clicked within the dial circle area and dragged the mouse to the right, then the dial knob should be rotated in the clockwise direction. Similarly, if the user dragged the mouse to the left, then the dial knob should be rotated in the counterclockwise direction. A more sophisticated solution of detection the mouse rotational movement is left to the developer to specify and implement, if desired.

11.1.3 Wavelength Selection

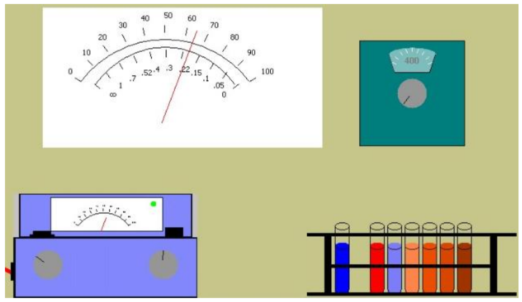


Rotate wavelength knob, select desire wavelength value

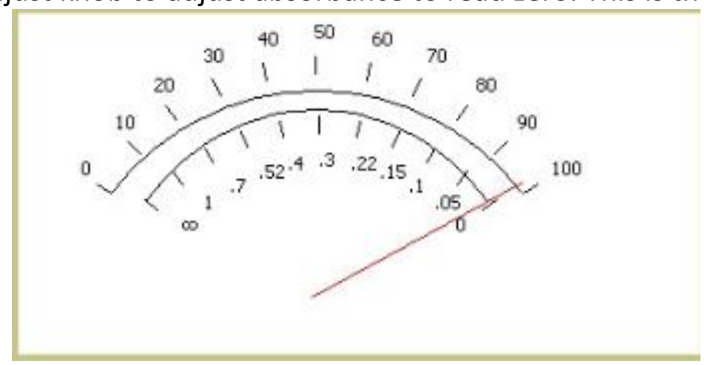
In order to use this instrument, the correct wavelength of light source must be chosen so that the light being shone is absorbed by the substance contained in a test tube when it tries to pass through the solution. Rotate the wavelength control until the desired wavelength is indicated by the wavelength scale.

11.1.4 Blank adjustment

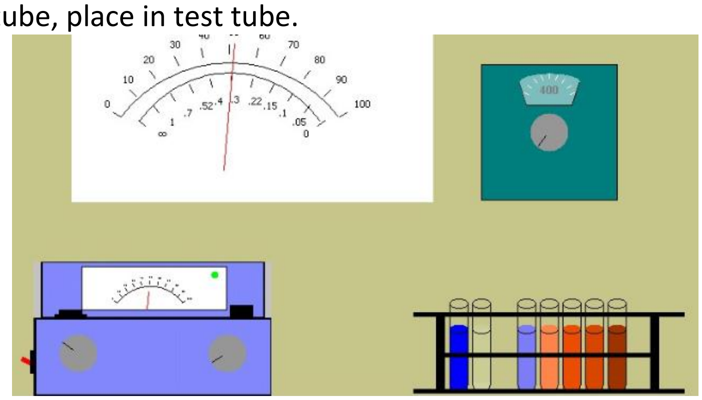
1. Fill the blank tube with the solvent used to dissolve specimen. Polish to clean, insert into the sample holder, aligning mark to front. Close holder lib.



2. Rotate blank adjust knob to adjust absorbance to read zero. This is the front right knob.



3. Remove blank tube, place in test tube.



11.2 GUI Design

11.2.1 The use of cross-platform color

The number of colors on our system should be considered before GUI design: 8 bits (256 colors), 16 bits (thousands of colors); and 24 bits (millions of colors). There might be dithering each time our java application is run. In order to get the best results, we will optimize our graphics on platforms as many as possible.

11.2.2 The design of application graphics

As for graphic file format choosing, we chose PNG (Portable Network Graphic) format. Compared with GIF format, PNG gives a much wider range of transparency options and color depths and it is usually used in java programs.

Application graphics that we design fall into three broad categories: Icons, Button graphics and Symbols. In icons design, there are several principles: a) Icons should be designed to identify clearly the object or concepts they represent; b) Make sure that large and small icons have similar shape, color, and detail if they represent the same object; c)Specify tool tips for each icon. In button graphics design, there are two principles: a) Use tool tips to help clarify meaning of toolbar buttons; b) Clearly show the action, state, or mode that the button initiates. Symbols include any graphic that stands for a state or a concept. In our application, they appear in alert box, question box, etc. Another important factor is all the graphics should use the same design style.

Some application graphics are not shown in our first demo, which will be shown in second demo.

11.2.3 The use of graphics for product identity.

The graphics for product identity include visual identifier of our application, product logo, information box about the application, etc. We haven't done this work until now but we will design these graphics after finishing all the functions of the application.

11.3 Ease-of-use Design

As we mention before, we will design our virtual device as they do in the real lab. That's also the principle when we considering the user effort. On the login interface, we design this interface as it is common being. We make the teacher's inter face easier to understand and operate, and as a teacher, you can just type in he information on the corresponding place. We change the button 'reset' with the button 'cancel' for the reason the reset function can be realized by pressing backspace on the keyboard. In the student's interface also called device interface, we design our user effort strongly in accord with operation steps which will help students be familiar with the real operation. The only user effort we add is the button 'update', we still design this button on the screen because by clicking this button, we will ensure students will see and pay attention to the information board.

11.4 Navigation Path

1). Begin by setting the wavelength. For this, click on the wave dial to obtain a top view, rotate the wavelength control until wavelength is indicated by the wavelength scale. Click on the switch to turn it on. Bring the meter needle to infinity on the absorbance scale by adjusting the dark control.

2). Click on the sample holder to open it. Choose the test tube with the transparent solution and place it in the holder. Then close the sample holder clicking it suit. Rotate the light control until the meter indicates zero on the absorbance scale. This procedure calibrates the instrument relative to the blank.

3). Open sample holder and drop the transparent test tube back into the rack. Click on any of the other sample test tubes and drop it into the sample holder as before. Take down the readings of transmittance and absorbtion only after closing the sample holder. Repeat the above process for all the test tubes. Close the window after you are done.

12. Design of test

12.1 Use Cases will be tested

Test-case Identifier: TC-1a **Use-case Tested: UC-1 SwitchOn , main success scenario** **Complete/Fail Criteria:** Everything works	
Test Procedure	**Expected Result**
Step 1 Select student check box and login Step2 Turn on the switch	 System display student interface Everything (zero adjust knob, blank adjust knob, wavelength select knob)on virtual spectrophotometer works

Test-case Identifier: TC-1b **Use-case Tested: UC-1 SwitchOn , main fail scenario** **Complete/Fail Criteria:** Everything does not work	
Test Procedure	**Expected Result**
Step 1 Select student check box and login Step2 With the switch not be turned on	 System display student interface Everything (zero adjust knob, blank adjust knob, wavelength select knob)on virtual spectrophotometer does not work

Test-case Identifier:TC-2 **Use-case Tested:UC-2 AdjustBlank, main success scenario** **Complete/Fail Criteria:** The meter's value displayed begins to change	
Test Procedure	**Expected Result**
Step 1 Select student check box and login	System display student interface
Step2 Turn on the switch	Machine starts work
Step3 Click lid	Lid opened
Step4 Insert blank tube (0) into sample holder	Tube inserted
Step5 Rotate blank adjust knob	The meter display the corresponding selected value

Test-case Identifier: TC-3 **Use-case Tested:UC-3 AdjustZero, main success scenario** **Complete/Fail Criteria:** The meter's value displayed begins to change	
Test Procedure	**Expected Result**
Step 1 Select student check box and login	System display student interface
Step2 Turn on the switch	Machine starts work
Step5 Rotate zero adjust knob	Until meter display the smallest scale

Test-case Identifier:TC-4 **Use-case Tested: UC-4 SelectWavelength, main success scenario** **Complete/Fail Criteria:** Meter displays corresponding selected wavelength	
Test Procedure	**Expected Result**
Step 1 Select student check box and login	System display student interface
Step2 Turn on the switch	Machine starts work
Step3 Rotate the wavelength select knob	Meter displays corresponding selected wavelength

Test-case Identifier:TC-5 **Use-case Tested: UC-5 UpdateInfoBoard, main success scenario** **Complete/Fail Criteria:** Update the newest information **Input data:** Inforation stored in system	
Test Procedure	**Expected Result**
Step 1 Select student check box and login	System display student interface The text field will display latest information automatically

Test-case Identifier: TC-6 **Use-case Tested: UC-6 MeasureTubeValue, main success scenario** **Complete/Fail Criteria:** displayed the tested tube value	
Test Procedure	**Expected Result**
Step 1 Tested tube selected Step 2 Drug to insert it into sample holder Step 3 Close the sample lid	The test tube was selected The test tube has been inserted into sample holder, and the meter value will be changed. The meter will change to a desired value

Test-case Identifier: TC-8-a **Use-case Tested: UC-8 Login, main success scenario** **Complete/Fail Criteria: log into the system**	
Test Procedure	**Expected Result**
Step1 Type in the user name and password and select corresponding check box	Log into the system successfully.

Test-case Identifier: TC-8-b **Use-case Tested: UC-8 Login, main failure scenario** **Complete/Fail Criteria: login failed** **Input Data: invalid user name and password**	
Test Procedure	**Expected Result**
Step1 Type in the invalid user name and password and select corresponding check box Step2 Type in the valid user name and password but choice the wrong checkbox (if it is a student account	Login failed. Login failed

Test-case Identifier: TC-7 Use-case Tested: UC-7 SetSampleValue, main success scenario Complete/Fail Criteria: The value stored in system is updated by teacher	
Test Procedure	**Expected Result**
Step 1 Select teacher check box and login Step 2 Type in the sample value in corresponding field	Teacher login success. The sample value stored in system is updated.

12.2 Unit Testing test cases

In this section, 3 important objects will be tested.

StudentGUI

Input	Action	Result
Valid student username and password	Login	Login Success
Invalid student username and Password	Login	Login Fail

TeacherGUI

Input	Action	Result
Valid teacher username and password	Login	Login Success
Invalid acher username and password	Login	Login Fail

Select TestTube

Input	Action	Result
click on test tube#1	select	select value=1
click on test tube#2	select	select value=2
click on test tube#3	select	select value=3
click on test tube#4	select	select value=4
click on test tube#5	select	select value=5
click on test tube#6	select	select value=6
click on test tube#7	select	select value=7
click on test tube#8	select	select value=8

Insert TestTube

Input	Action	Result
lid = open	insert test tube	Insert Success
lid = closed	insert test tube	Insert Fail

Blank Adjust

Input	Action	Result
switch=off	rotate blank adjust knob	meter displays nothing
switch=on holder = 0 tube	rotate blank adjust knob	meter displays largest scale
switch=on holder = { 1,2,3,4,5,6,7,8 tube}	rotate blank adjust knob	system runs inaccurateresult()

Zero Adjust

Input	Action	Result
switch=on	rotate zero adjust knob	meter displays smallest scale
switch=off	rotate zero adjust knob	meter displays nothing

Wavelength Select

Input	Action	Result
switch=on	rotate wavelength knob	meter displays the selected wavelength
switch=off	rotate wavelength knob	meter displays nothing

12.3 Test Coverage

These 9 .java files listed below will be covered. They possess about 85% of our whole project.

Unit 1 Login in test

In this unit, a Login program is used to test the login in function. In this program, user could choose to login in as student or teacher. The user need to input ID and password which are stored in the database. If the ID and password are not match with the database, the user cannot login in. When the ID, password and teacher/student are all right, it will display "Teacher login in" or "Student login in", otherwise it will display "invalid ID or password".

Unit 2 Adjust Zero

In this unit, a adjustZero program is used to test the adjust zero function. When running this program, the user could set the initial value with zero.

Unit 3 Select Wavalength

In this unit, a selectWave program is used to test selectwavelength function. User could set the wanted wavelength.

Unit 4 Adjust Blank

In this unit, a adjust blank program is used to test adjust blank function. User could set blank state with the given sample solution.

Unit 5 Open/Close Lid

User could open or close the lid by clicking the cover with mouse in this unit. And the program could return a state of the lid.

Unit 6 Change Sample Value

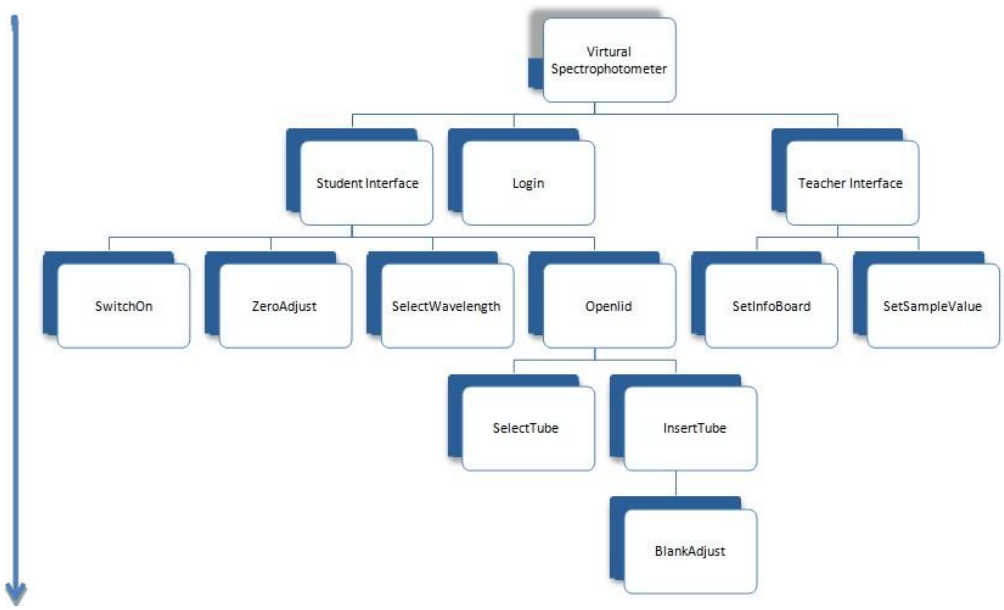
User could set the sample solution with different values. And these values will be saved by clicking ''save'' button.

Unit 7 Update Information board

User can input text in the text box. And by clicking save button, the information can be updated.

12.4 Integration Test Strategy

Our integration test will apply **top-down** integration path. As our project is a interactive system between human and system, every operation will result in different results. It is very common to a class call another class. Therefore the whole programing logic in our project is very crucial. Therefore, by implementing top-down integration test path, we could find faults easier than to separate them into small pieces.



Interaction

13. History of work

This project is a sub-project of Virtual Biology Lab, and it is not in our original plan. So we started this project a little later. It may be overdue when we finish and submit the report #1, but we will try our best to catch up all following deadlines.

This part is the schedule of our future work (Table 6-1). We are dedicated to finishing this work step by step. Therefore, our plan can be separate into 5 parts.

1. Preparation Oct.15 –Oct.18
   In this part, we will do the following work:
   i. Revising our Report #1
   ii. Studying previous work
   Because our project is based on a previous "just-started" work, so we need to check what they have done. We can utilize some their work or improve their design.
   iii. Scientific Research
   This virtual machine is designed to be used in education. Therefore, the final result must be in accordance with the truth. That means the result of this virtual machine must be as same as the result that students get on a physical machine even operated incorrectly.
   iv. Learning Java
   We need to keep learning and discuss problems during learning process.
2. Interface Implement Oct. 19-Oct. 26(First Demo)
   At the end of this part, we need to show our first demo of our program and finish part1 of Report #2. Therefore we need the following work:
   i. Drawing picture
   We will use PS to draw every component of the spectrophotometer.
   ii. Interface Design
   We will design where every component is located on the screen. Dials can be rotated by user and needle can move.
   iii. Finishing part1 of Report #2
3. Database Design Oct. 27- Nov. 6 (report #2)
   We will finish Report #2 during this period, and design the structure of database.
4. Full-Feature Completion & Debugging Nov. 7- Dec. 7 (second Demo)
   We will finish all features showed above. For instance, showing scientific result and making operations more reasonable.
5. Finishing project Dec. 8- Dec. 12(final report)
   We will collect all documents and submit final report.

13.1 Problems and Progress Report

Our project has three parts, LoginGUI, TeacherGUI, and StudentGUI(main part). Our group is divided into two subgroups, one is responsible for coding and the other is responsible for debugging and testing. The first two parts has been finished. Users, teacher and student, can log in their system, respectively, by inputting correct ID, password and account type, which is saved in the AccountStorage file. Teacher can change the information board and save it in a file named infoboard.txt. Student can see what teacher input after they enter their GUI. Also teacher can change the values of samples in the tubes. We are now concentrated on implementing the main part of our project, student GUI. By now, our Demo1 has been finished. We meet some problems, and we are trying our best to fix it. Since our project is based on a previous unfinished work, we have a part of source code of this program. However, we find that some source file is missing. For some classes, we only have .class file but not .java file, which makes us hard to continue to work. We are now facing dilemma that using plug-in to decompile the original file and re-write it, or design a new framework by ourselves.

13.2 Plan of Work

The next step of our work is to complete StudentGUI. Firstly, we are planning to divide our group into two directions, one is to study on the previous works and figure out solution of the problems mentioned above, and the other one is to do some research in order to make our Virtual Device works as a real one. Secondly, we will keep doing programming and testing work. We will try our best to implement all functions of Demo 2 before Dec. 7. We will collect all documents related to this project and finish the third report by Dec. 12 (as figure 5-1 shows).

Task Name	Duration	Start	Finish
Source code analysis	5 days	Mon 11/12/12	Fri 11/16/12
Scientific research	4 days	Wed 11/14/12	Mon 11/19/12
Programming	15 days	Wed 11/14/12	Tue 12/4/12
testing	11 days	Sun 11/25/12	Fri 12/7/12
Demo 2 deadline	0 days	Fri 12/7/12	Fri 12/7/12
finishing Demo 2	3 days	Wed 12/5/12	Fri 12/7/12
Report #3 Deadline	0 days	Wed 12/12/12	Wed 12/12/12
finishing Report #3	4 days	Sat 12/8/12	Wed 12/12/12
finishing Project	2 days	Thu 12/13/12	Fri 12/14/12

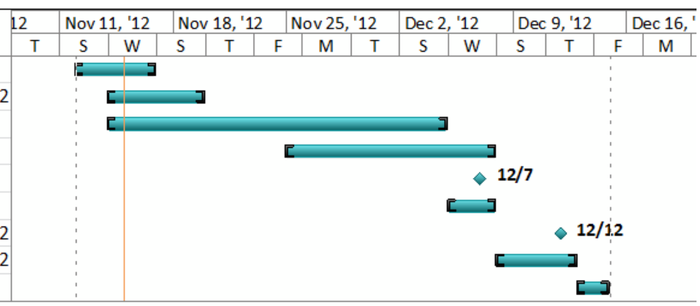


Figure 6-1
Plan of work

Reference

[1] Marsic, Ivan. *Software Engineering*, Sep. 10, 2012. URL:http://www.ece.rutgers.edu/~marsic/books/SE/book-SE_marsic.pdf

[2] Spectrophotometry. URL: http://en.wikipedia.org/wiki/Spectrophotometry

[3] Spectrophotometry. URL:http://www.chm.davidson.edu/vce/spectrophotometry/Spectrophotometry.html

[4] Biology Lab Spectrophotometry. URL: http://iws.collin.edu/dcardenas/labs/labs/spectrophotometry.pdf

[5] David B. Fankhauser. Spectrophotometer Use. URL: http://biology.clc.uc.edu/fankhauser/Labs/Microbiology/Growth_Curve/Spectrophotometer.htm

[6] Principles of Spectrophotometer. URL: http://www.ruf.rice.edu/~bioslabs/methods/protein/spectrophotometer.html

[7] Spectrophotometer tutorial.URL: http://www.youtube.com/watch?v=20TpmeaTJho

[8] Alistair Cockburn. *Writing effective use case*.URL: http://www2.dis.ulpgc.es/~jsanchez/MDS/EffectiveUseCases.pdf

[9] *Software Engineering-concepts and implementation*.URL: http://www.msuniv.ac.in/soft.pdf

[10] Defining Non-Functional Requirements.URL: http://www.bredemeyer.com/pdf_files/NonFunctReq.PDF

[11] Domain modeling.URL: http://www.aptprocess.com/whitepapers/DomainModelling.pdf

[12] Making good software.URL: http://www.makinggoodsoftware.com/2010/05/17/how-to-create-a- good-domain-model-top-10-advices/

[13] Rajaram Subramanian and Ivan Marsic, *ViBE: Virtual Biology Experiments*

[14] UML Sequence Diagram Tutorial. http://www.sequencediagrameditor.com/uml/sequence- diagram.htm

[15] UML Sequence Diagrams: Guidelines. http://msdn.microsoft.com/en-us/library/dd409389.aspx

[16] Russ Miles and Kim Hamilton *Learning UML 2.0*

[17] Java look and feel design guidelinesURL: http://www.oracle.com/technetwork/java/index-136139.html

[18] User interface design URL: http://en.wikipedia.org/wiki/User_interface_design

[19]Java Testing and DesignURL:
http://pttdownloads.s3.amazonaws.com/JavaTestAndDesign_Cohe n.pdf
[20]Usability URL: http://en.wikipedia.org/wiki/Usability

Summary of Changes

1.1 Detail the Customer Statements of Requirement.
3.4 On Screen Appearance Requirement, We illustrate the strategy to allow students to make mistake.
4.3 Replaced UC-6,7,8,9,10,11 by MeasureTubeValue.
4.4 Modify USE CASE DIAGRAM.
4.7 Update System Sequence Diagram.
5 Add part Effort Estimation using User Case Points.
6.1 Preliminary design. Based on the second demo, we update the interface design.
6.2 User Effort Estimation. We optimize the user efforts, update the user effort as we do in the second demo.
7.1 Modify domain model, modify concepts, attributes.
7.1.2 Add to line in association table.
8.1 Update all the interaction diagrams.
9 remove unused Class Diagram
11.1 GUI overview. Add the animation specification of power switch, adjust zero control dial, wavelength selection and Blank adjustment.
11.4 Based on the optimization of user effort, we add navigation path to help students use it.