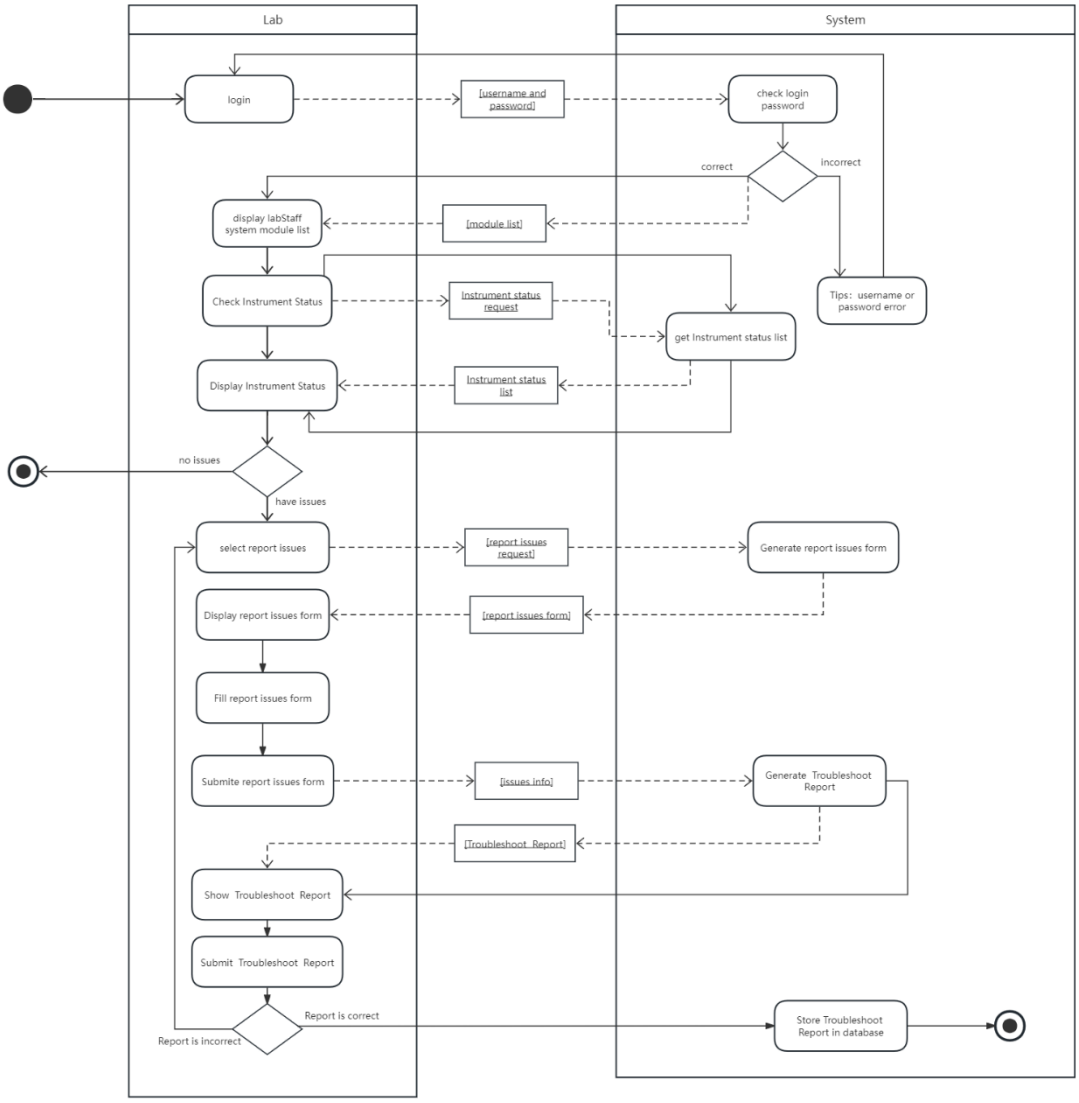
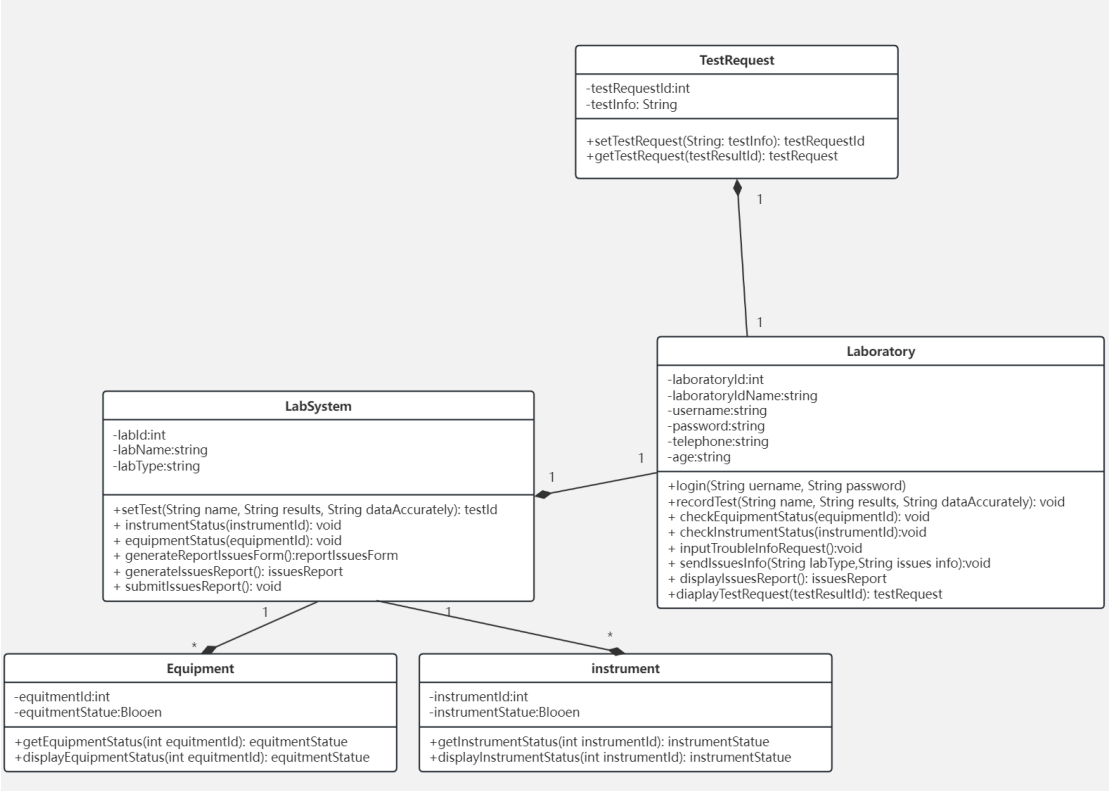
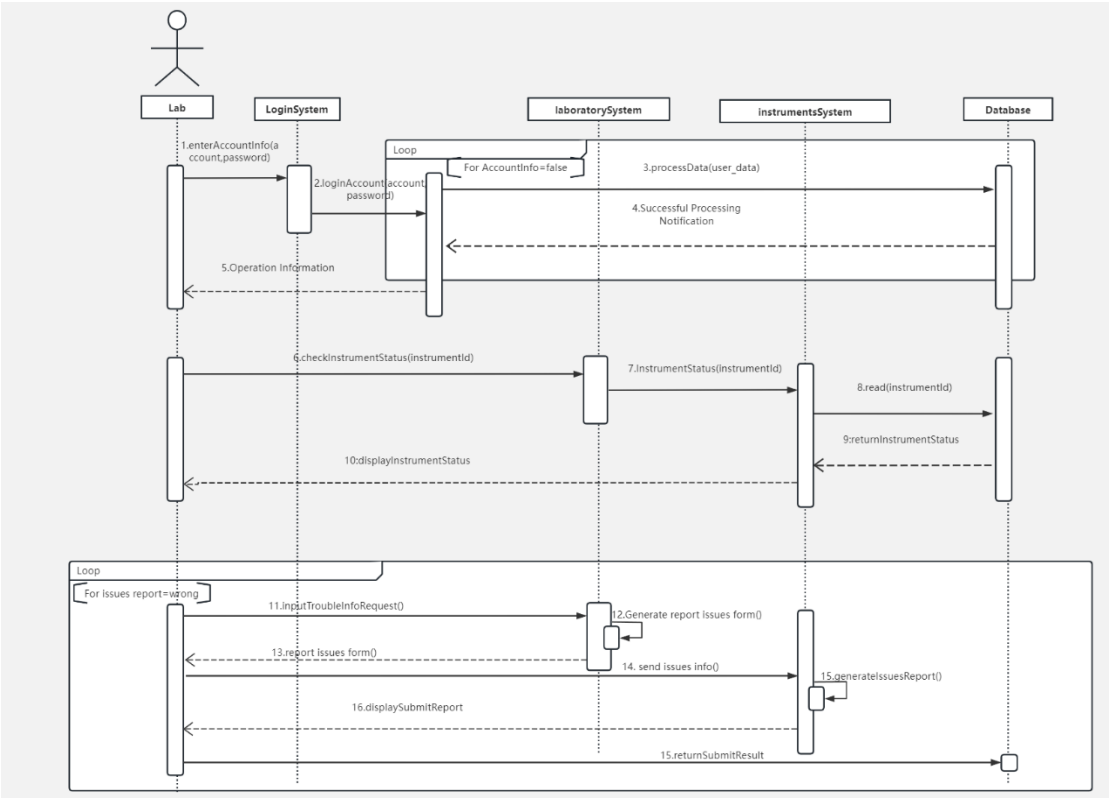
Task1.2


Lab
System
login
[username and password]
check login password
correct
incorrect
display labStaff system module list
[module list]
Check Instrument Status
Instrument status request
Tips: username or password error
get Instrument status list
Display Instrument Status
Instrument status list
no issues
have issues
select report issues
[report issues request]
Generate report issues form
Display report issues form
[report issues form]
Fill report issues form
Submite report issues form
[issues info]
Generate Troubleshoot Report
[Troubleshoot Report]
Show Troubleshoot Report
Submit Troubleshoot Report
Report is correct
Report is incorrect
Store Troubleshoot Report in database

Task1.3


TestRequest
-testRequestId:int
-testInfo: String
+setTestRequest(String: testInfo): testRequestId
+getTestRequest(testResultId): testRequest
1
1
Laboratory
-laboratoryId:int
-laboratoryIdName:string
-username:string
-password:string
-telephone:string
-age:string
+login(String uername, String password)
+recordTest(String name, String results, String dataAccurately): void
+ checkEquipmentStatus(equipmentId): void
+ checkInstrumentStatus(instrumentId):void
+ inputTroubleInfoRequest():void
+ sendIssuesInfo(String labType,String issues info):void
+ displayIssuesReport(): issuesReport
+diaplayTestRequest(testResultId): testRequest
1
1
LabSystem
-labId:int
-labName:string
-labType:string
+setTest(String name, String results, String dataAccurately): testId
+ instrumentStatus(instrumentId): void
+ equipmentStatus(equipmentId): void
+ generateReportIssuesForm():reportIssuesForm
+ generateIssuesReport(): issuesReport
+ submitIssuesReport(): void
1
1
*
*
Equipment
-equitmentId:int
-equitmentStatue:Blooen
+getEquipmentStatus(int equitmentId): equitmentStatue
+displayEquipmentStatus(int equitmentId): equitmentStatue
instrument
-instrumentId:int
-instrumentStatue:Blooen
+getInstrumentStatus(int instrumentId): instrumentStatue
+displayInstrumentStatus(int instrumentId): instrumentStatue

Task1.4


Lab
LoginSystem
laboratorySystem
instrumentsSystem
Database
1.enterAccountInfo(account,password)
2.loginAccount(account,password)
Loop
For AccountInfo=false
3.processData(user_data)
4.Successful Processing Notification
5.Operation Information
6.checkInstrumentStatus(instrumentId)
7.InstrumentStatus(instrumentId)
8.read(instrumentId)
9:returnInstrumentStatus
10:displayInstrumentStatus
Loop
For issues report=wrong
11.inputTroubleInfoRequest()
12.Generate report issues form()
13.report issues form()
14. send issues info()
15.generateIssuesReport()
16.displaySubmitReport
15.returnSubmitResult

Task2.1

Component

Component	Login manager
Description	This microservice running on the could to manage laboratory login.
Type	Laboratory App
Required Interface	Login
Provided interface	Access Code

Component	Data Entry manager
Description	This microservice running on the could to manage data entry.
Type	Laboratory App
Required Interface	Lab, Access Code, Get request
Provided interface	/

Component	Instrument Troubleshooting manager
Description	This microservice running on the could to manage instrument Troubleshooting.
Type	Laboratory App
Required Interface	Equipment, instrument, Access Code
Provided interface	/

Component	Hand Test Request manager
Description	This microservice running on the could to handle the test request from the doctor.
Type	Laboratory App
Required Interface	Test request, Access code
Provided interface	Get Request

Component	Authentication manager
Description	This microservice running on the could to handle the login request from the user.
Type	Microservice
Required Interface	Check password
Provided interface	Login

Component	Laboratory Data manager
Description	This microservice running on the could to manage Laboratory Data
Type	Microservice
Required Interface	/
Provided interface	Check password, Check Access

Component	Lab
Description	This microservice running on the could to manage Lab Data

Type	Microservice
Required Interface	Check Access
Provided interface	Lab

Component	Equipment
Description	This microservice running on the could to manage equipment Data
Type	Microservice
Required Interface	Check Access
Provided interface	Equipment

Component	Instrument
Description	This microservice running on the could to manage equipment Data
Type	Microservice
Required Interface	Check Access
Provided interface	Instrument

Component	Test Result
Description	This microservice running on the could to manage test result
Type	Microservice
Required Interface	Check Access
Provided interface	Test Result

Interface

Interface name	Login
Provider	Authentication Manager
	Operation: login
	Request Parameters: username, password **Response parameter:** True if success, False if fail **Function:** Check if the user's password matches the stored password. If so, return Ture; otherwise, return False.

Interface name	Test Request
Provider	Test Request
	Operation: setTestRequest
	Request Parameters: testInfo **Response parameter:** testRequestId if success, False if fail **Function:** Store the test request into the database. Return testInfo, if success; otherwise, return False.
	Operation: getTestRequest
	Request Parameters: testResultId **Response parameter:** testRequest if success, False if fail **Function:** Get the test request store in the database use the test request id. Return testRequest, if success; otherwise, return False.

Interface name	Equipment
Provider	Equipment
	Operation: getEquipmentStatus
	Request Parameters: equipmentId **Response parameter:** equipmentId if success, False if fail **Function:** Get the equitmenpStatue store in the database use the equipment id. Return equipmentStatue, if success; otherwise, return False.
	Operation: displayEquipmentStatus
	Request Parameters: equitpmentId **Response parameter:** equipmmentStatue if success, False if fail **Function:** Get the equipment Statue store in the database use the equipmment id. Return equipmmentStatue, if success; otherwise, return False.

Interface name	instrument
Provider	instrument
	Operation: getInstrumentStatus
	Request Parameters: InstrumentId **Response parameter:** InstrumentId if success, False if fail **Function:** Get the InstrumentStatue store in the database use the Instrument id. Return InstrumentStatue, if success; otherwise, return False.
	Operation: displayInstrumentStatus
	Request Parameters: InstrumentId **Response parameter:** InstrumentStatue if success, False if fail **Function:** Get the Instrument Statue store in the database use the Instrument id. Return InstrumentStatue, if success; otherwise, return False.

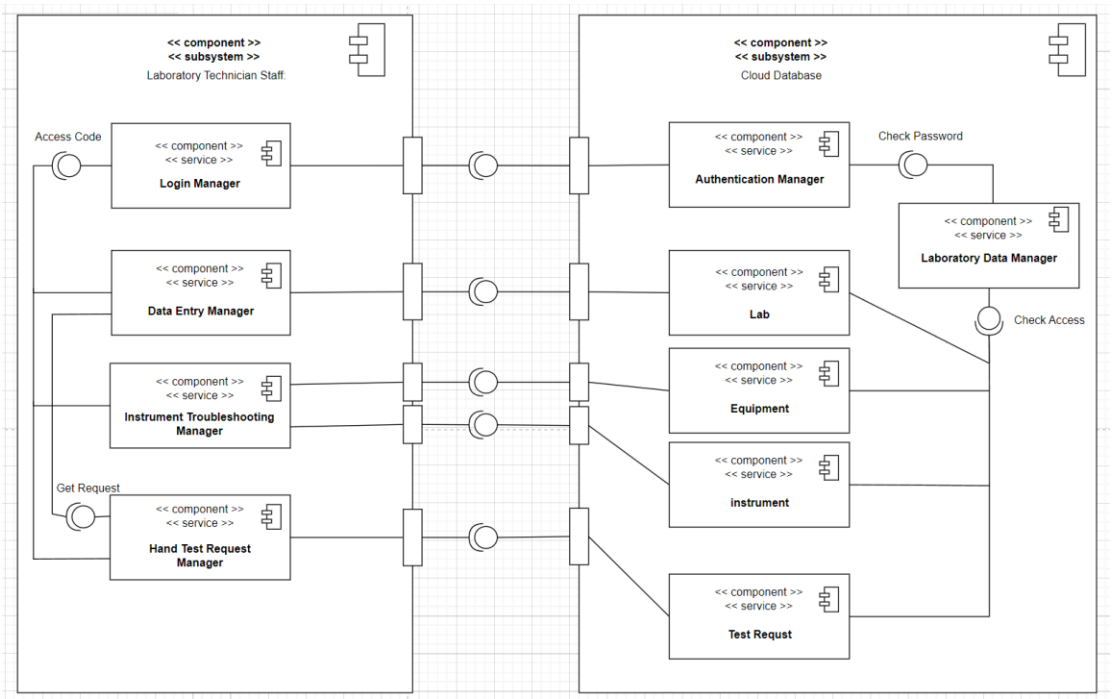
Interface name	Lab
Provider	Lab
	Operation: setTest
	Request Parameters: name, results, dataAccurately **Response parameter:** True if success, False if fail **Function:** Record the test the database. Return true, if success; otherwise, return False.
	Operation: instrumentStatus
	Request Parameters: instrumentId **Response parameter:** True if success, False if fail **Function:** Send the get instrument statue request to instrument database. Return true if sucess; otherwise, return False.
	Operation: equipmentStatus
	Request Parameters: equipmentId **Response parameter:** True if success, False if fail **Function:** Send the get equipment statue request to equipment database.

	Return true if sucess; otherwise, return False.
	Operation: generateReportIssuesForm
	Request Parameters: / **Response parameter:** reportIssuesForm if success, False if fail **Function:** Generate report Issues Form for user to fill in information. Return reportIssuesForm, if success; otherwise, return False.
	Operation: generateIssuesReport
	Request Parameters: / **Response parameter:** IssuesReport if success, False if fail **Function:** Generate Issues report for user to see what they update. Return issuesReport, if success; otherwise, return False.
	Operation: submitIssuesReport
	Request Parameters: / **Response parameter:** True if success, False if fail **Function:** Set the issues that user input into the database. Return true, if success; otherwise, return False.

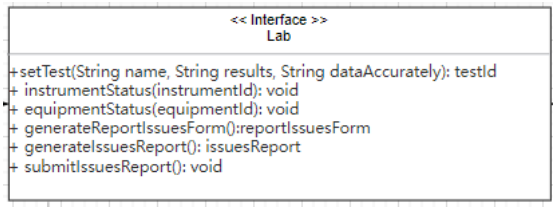
Database

Table: Authentication Manager	Fields: uername; password
Table: Lab	Fields: laboratoryId; laboratoryIdName; username; password; telephone; age
Table: Equipment	Fields: equitmentId; equitmentStatue
Table: Instrument	Fields: instrumentId; instrumentStatue
Table: Test Requst	Fields: testRequestId; testInfo

Task2.2


<< component >>
<< subsystem >>
Laboratory Technician Staff:
Access Code
<< component >>
<< service >>
Login Manager
<< component >>
<< service >>
Data Entry Manager
<< component >>
<< service >>
Instrument Troubleshooting Manager
Get Request
<< component >>
<< service >>
Hand Test Request Manager
<< component >>
<< subsystem >>
Cloud Database
<< component >>
<< service >>
Authentication Manager
Check Password
<< component >>
<< service >>
Laboratory Data Manager
<< component >>
<< service >>
Lab
Check Access
<< component >>
<< service >>
Equipment
<< component >>
<< service >>
instrument
<< component >>
<< service >>
Test Requst

Task3.1



Test Case	Method and Parameters	Expected output
Set test successfully	setTest(Blood, fresh, accurately)	Test id
Set test failed	setTest(Blood09, fres++, accura-)	Error Message
Get instrument status successfully	instrumentStatus(000001)	Instrument status
Get instrument status failed	instrumentStatus(ouhy*1)	Error Message
Get equipment status successfully	equipmentStatus(000001)	equipment status
Get equipment status failed	equipmentStatus(ouhy*1)	Error Message
Generate Report Issues Form successfully	generateReportIssuesForm()	Report issues form
Generate Report Issues Form failed	generateReportIssuesForm()	Error Message
Generate Issues Report successfully	generateIssuesReport()	Issues report
Generate Issues Report failed	generateIssuesReport()	Error Message
Submit Issues Report successfully	submitIssuesReport()	Submit success alert
Submit Issues Report failed	submitIssuesReport()	Error Message

Task3.2

Use Case: Laboratory login and check instrument status

Scenario: Successful login and check instrument status

1. Laboratory enter username and password	
2. Laboratory login	3. Check laboratory information in database, get module list in lab database
4. Display lab system module list	
5. Select check Instrument Status	6. Get Instrument Status list in database
7. Display Instrument Status list	
8. No issues find	

Test data:

Search laboratory information: username, password

Stored data:

- On mobile devise:
  laboratory information(not empty)
  module list(not empty)
  Instrument Status list(not empty)

Output:

Module list

Instrument Status list

Test Process

1. Enter username and password and select login.

a. Except output: display lab system module list

b. Check: (a)if output match the expect; (b)if detail of the list correct and complete

2. Select check Instrument Status

a. Except output: display Instrument Status list

b. Check: (a)if output match the expect; (b)if detail of the list correct and complete

Use Case: Laboratory login and check instrument status

Scenario: Successful login after failed login and check instrument status

1. Laboratory enter username and password	
2. Laboratory login(failed)	3. Check laboratory information in database, get module list in lab database
4. Laboratory enter username and password	
5. Laboratory login	6. Check laboratory information in database, get module list in lab database
7. Display lab system module list	
8. Select check Instrument Status	9. Get Instrument Status list in database
10. Display Instrument Status list	
11. No issues find	

Test data:

Search laboratory information: username, password(wrong)

Search laboratory information: username, password(right)

Stored data:

- On mobile devise:
  laboratory information(not empty)
  module list(not empty)
  Instrument Status list(not empty)

Output:

Error message

Module list

Instrument Status list

Test Process

1. Enter wrong username and password and select login.

a. Except output: error message

b. Check: if output match the expect;

2. Enter right username and password and select login.

a. Except output: display lab system module list

b. Check: (a)if output match the expect; (b)if detail of the list correct and complete

3. Select check Instrument Status

a. Except output: display Instrument Status list

b. Check: (a)if output match the expect; (b)if detail of the list correct and complete

Use Case: Laboratory login and submit instrument Troubleshooting

Scenario: Successful login and submit instrument Troubleshooting

1. Laboratory enter username and password	
2. Laboratory login	3. Check laboratory information in database, get module list in lab database
4. Display lab system module list	
5. Select check Instrument Status	6. Get Instrument Status list in database
7. Display Instrument Status list	
8. Find issues	
9. Select report issues	10. Generate report issues form
11. Display report issues form	
12. Fill report issues form	
13. Submit report issues form	14. Generate Troubleshoot Report
15. Show Troubleshoot Report	
16. Submit Troubleshoot Report	17. Store Troubleshoot Report in database

Test data:

Search laboratory information: username, password

Instrument issues: issues type, issues info

Stored data:

- On mobile devise:
  laboratory information(not empty)
  module list(not empty)
  Instrument Status list(not empty)
  Generate right Troubleshoot Report

Output:
Module list
Instrument Status list
Report issues form
Troubleshoot Report

Test Process

1. Enter username and password and select login.
a. Except output: display lab system module list
b. Check: (a)if output match the expect; (b)if detail of the list correct and complete
2. Select check Instrument Status
a. Except output: display Instrument Status list
b. Check: (a)if output match the expect; (b)if detail of the list correct and complete
3. Select report issues
a. Except output: display report issues form
b. Check: (a)if output match the expect; (b)if detail of the report issues form correctly and complete
4. Troubleshoot Report
a. Except output: display troubleshoot report
b. Check: (a)if output match the expect; (b)if detail of the report form correctly and complete

Use Case: Laboratory login and submit instrument Troubleshooting
Scenario: Successful login and submit instrument Troubleshooting after modify report's problem(user input the wrong information and find it when check the report)

1. Laboratory enter username and password	
2. Laboratory login	3. Check laboratory information in database, get module list in lab database
4. Display lab system module list	
5. Select check Instrument Status	6. Get Instrument Status list in database
7. Display Instrument Status list	
8. Find issues	
9. Select report issues	10. Generate report issues form
11. Display report issues form	
12. Fill report issues form	
13. Submit report issues form	14. Generate Troubleshoot Report
15. Show Troubleshoot Report	
16. Find report is incorrect	

17. Back to step 9 until the report is correct	
18. Submit Troubleshoot Report	19. Store Troubleshoot Report in database

Test data:

Search laboratory information: username, password

Instrument issues: issues type, issues info(wrong), issues info(right)

Stored data:

- On mobile devise:
  laboratory information(not empty)
  module list(not empty)
  Instrument Status list(not empty)
  Generate right Troubleshoot Report

Output:

Module list

Instrument Status list

Report issues form

Troubleshoot Report

Test Process

1. Enter username and password and select login.

a. Except output: display lab system module list

b. Check: (a)if output match the expect; (b)if detail of the list correct and complete

2. Select check Instrument Status

a. Except output: display Instrument Status list

b. Check: (a)if output match the expect; (b)if detail of the list correct and complete

3. Select report issues

a. Except output: display report issues form

b. Check: (a)if output match the expect; (b)if detail of the report issues form correctly and complete

4. Troubleshoot Report(wrong)

a. Except output: display troubleshoot report(with wrong information)

b. Check: (a)if output match the expect; (b)if detail of the report form correctly and complete(match the wrong input information)

5. Troubleshoot Report(right)

c. Except output: display troubleshoot report(with right information)

d. Check: (a)if output match the expect; (b)if detail of the report form correctly and complete(match the right input information)