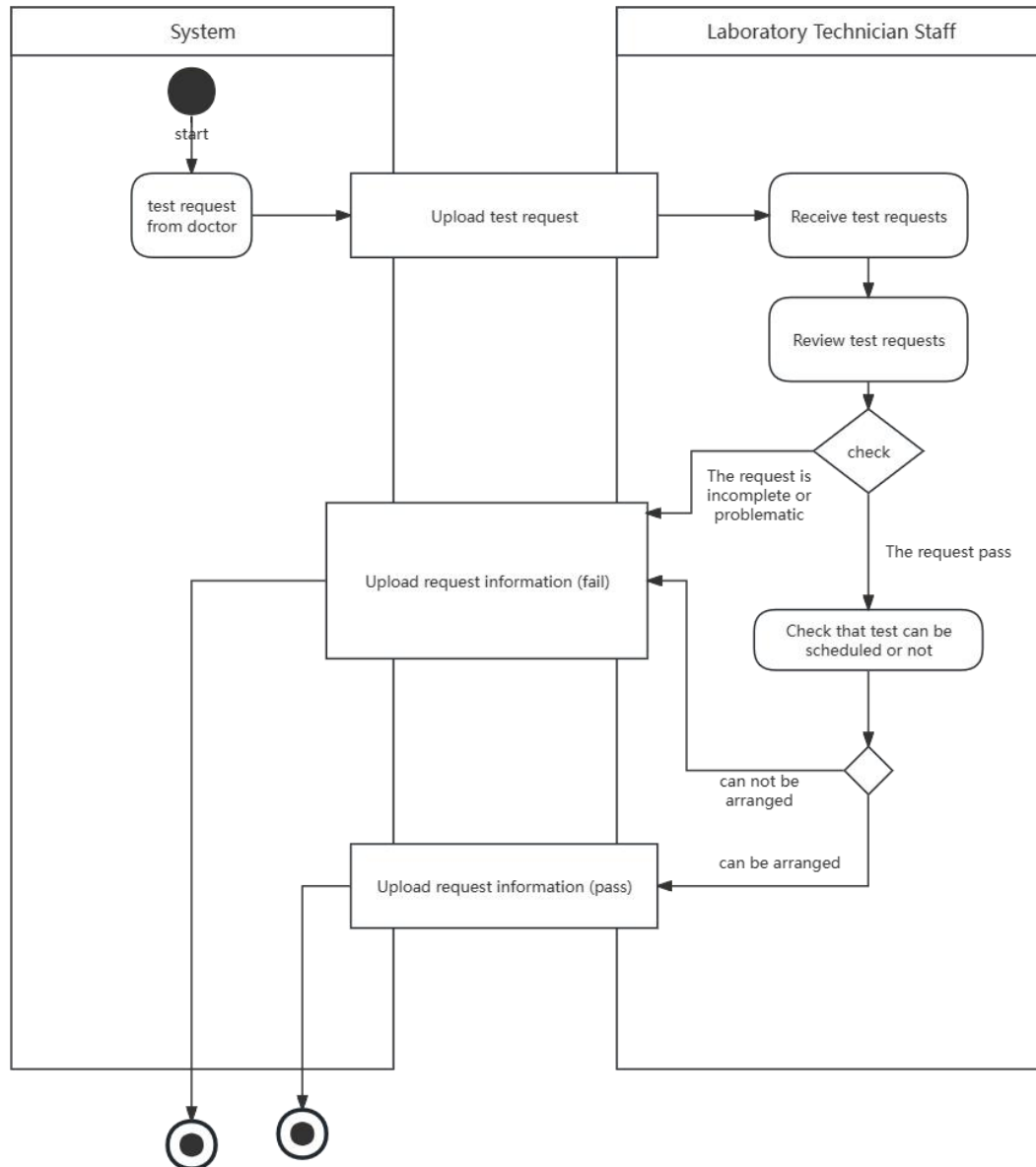
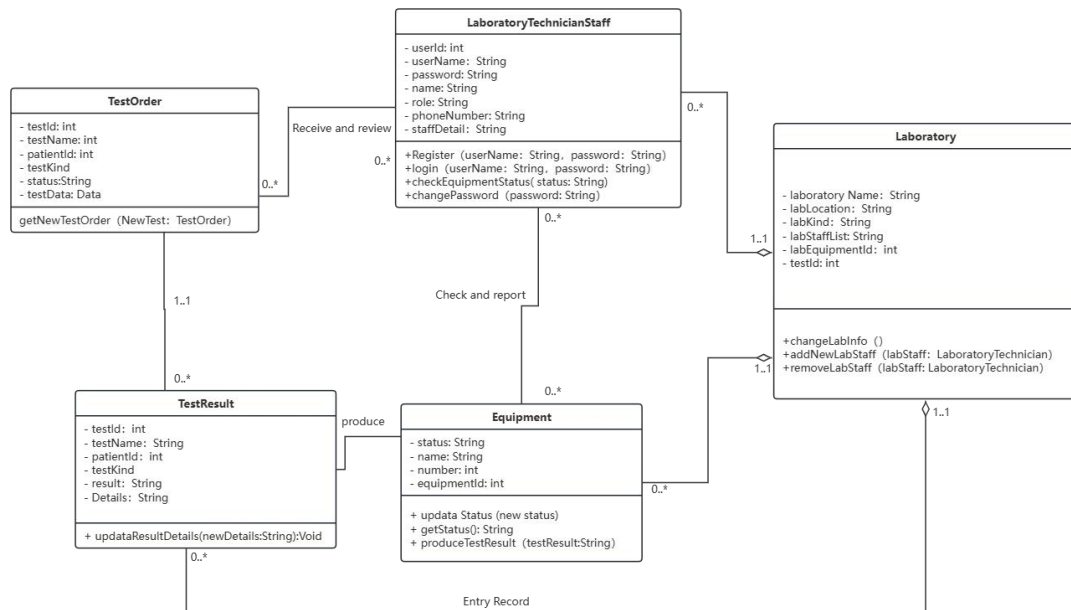


## Task1 :

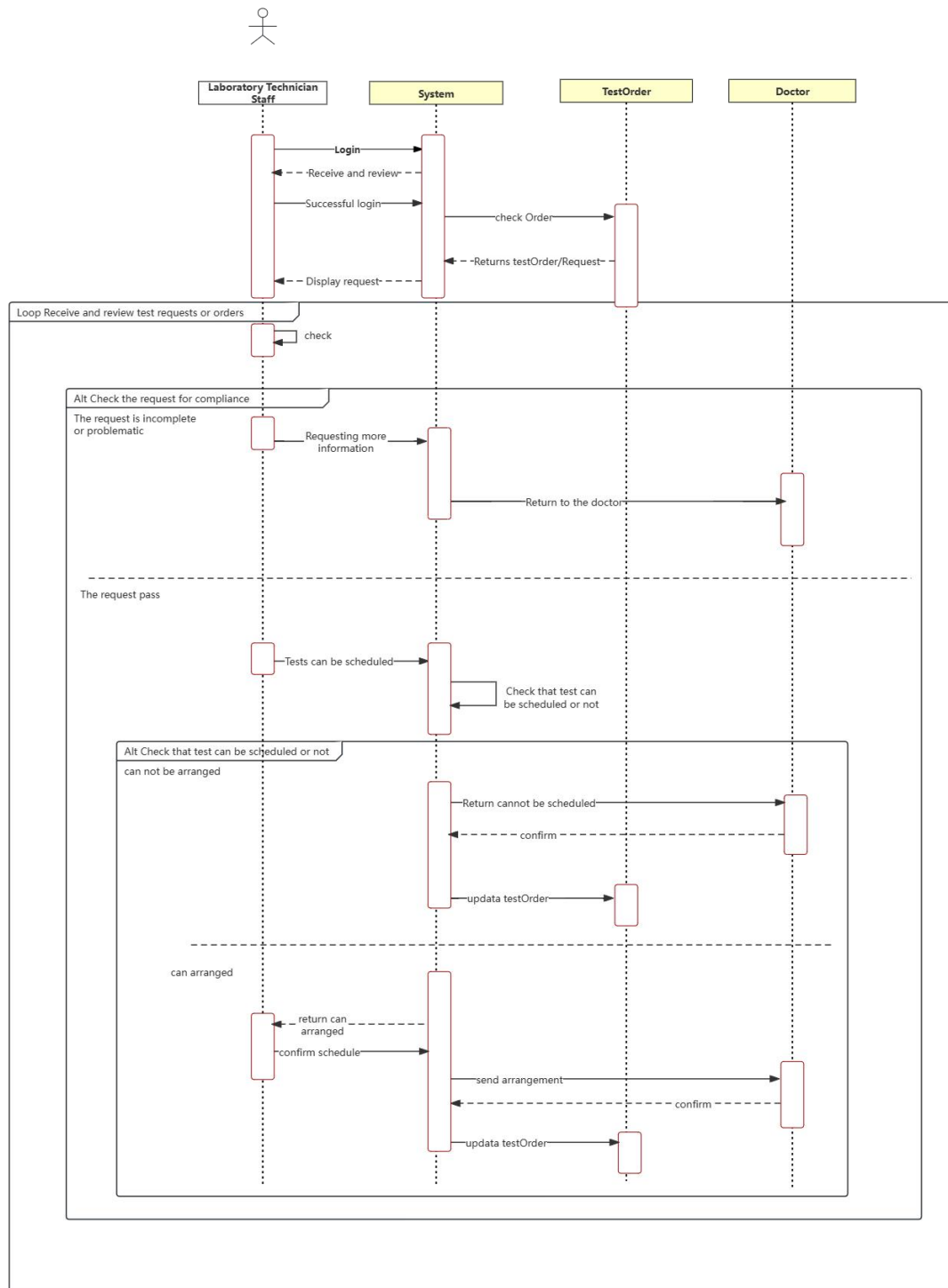
### A .Activity Diagram:



## B. Class Diagram

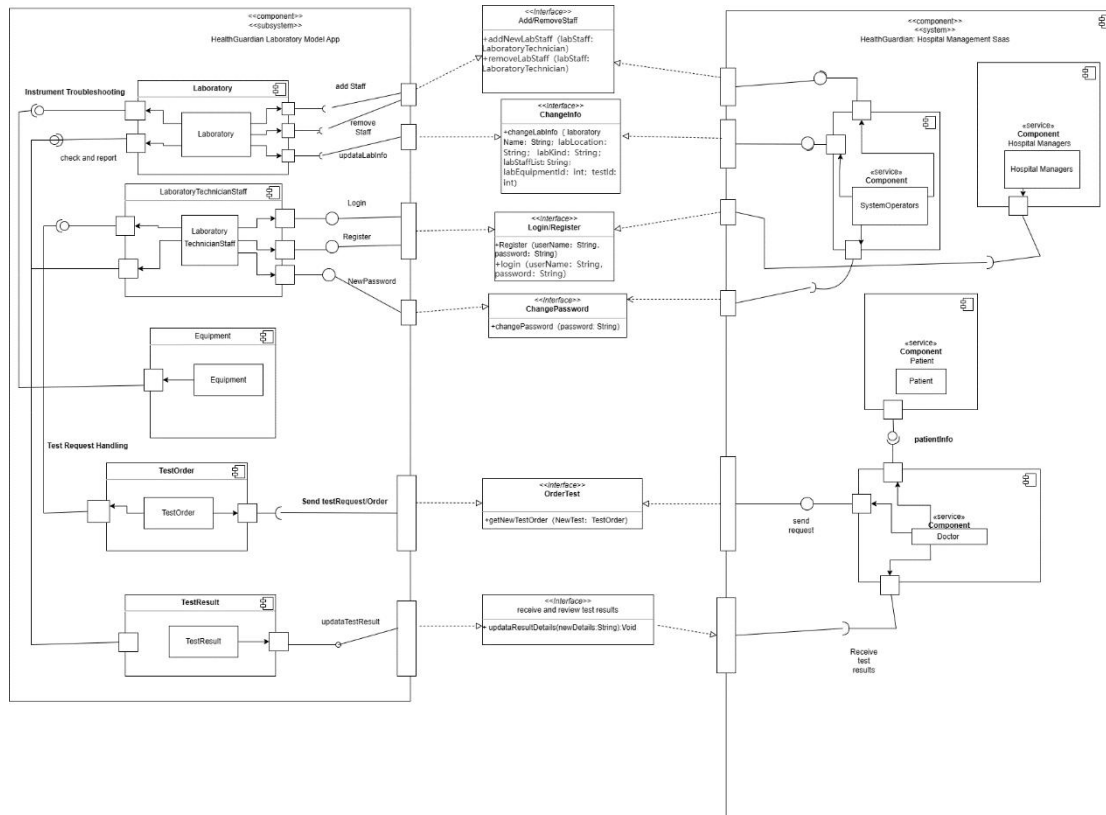


## C. Behaviour Digram



## 2.Task2: Software Architectural Design

### 2.1Architecture of the subsystem



## 2.2 Component:

	Laboratory Manager
Description	Change Lab Info, add New Lab Staff, remove Lab Staff: Store lab Name 、 lab Location 、 lab Kind 、 lab Staff List 、 lab Equipment Id、 test Id
Type	HealthGuardian Laboratory Model App
Required Interfaces:	check and report , add Staff , remove Staff , updata Lab Info
Provided Interfaces:	<b>Instrument Troubleshooting</b>

Component	Laboratory Technician Staff Manager
Description	Register, login, check Equipment Status, change Password,
Type	HealthGuardian Laboratory Model App
Required Interfaces:	<b>Test Request Handling</b> , check and report
Provided Interfaces:	Login, Register, NewPassword

Component	Equipment Manager
Description	Storing information about laboratory equipment and instruments and updating problems with laboratory equipment and instruments
Type	HealthGuardian Laboratory Model App
Required Interfaces:	None
Provided Interfaces:	<b>Instrument Troubleshooting</b>

Component	Test Order Manager
Description	Record and update testOrder and send test Order List to Laboratory Technician Staff
Type	Service
Required Interfaces:	<b>Test Request Handling , Send testRequest/Order</b>
Provided Interfaces:	None

Component	Test Result Manager
Description	to Record test name, results and data accurately into the laboratory information system.
Type	HealthGuardian Laboratory Model App
Required Interfaces:	None
Provided Interfaces:	check and report , updata Test Result

Component	System Operators Manager
Description	System operators can create, modify,

	delete user accounts for hospital staff (hospital managers, doctors, receptionists, nurses, pharmacists and laboratory technician staff) and reset their passwords.
Type	HealthGuardian: Hospital Management Saas
Required Interfaces:	NewPassword
Provided Interfaces:	add Staff, remove Staff

Component	HospitalManagers Manager
Description	Access to user management features to control access permissions and roles for hospital staff.
Type	HealthGuardian: Hospital Management Saas
Required Interfaces:	Login, Register
Provided Interfaces:	None

Component	Doctor Managment
Description	Provide test Order and receive test Result
Type	HealthGuardian: Hospital Management Saas
Required Interfaces:	Receive test results, <b>PatientInfo</b>
Provided Interfaces:	send request

Component	Patient Manager
Description	Provide business for doctors so that doctors can generate test requirements.
Type	HealthGuardian: Hospital Management Saas
Required Interfaces:	None
Provided Interfaces:	<b>PatientInfo</b>

## 2.3 Interface:

Interface Name	Add/RemoveStaff
Provider	SystemOperators Managment
	Operation: addNewLabStaff
	Request Parameters: labStaff: LaboratoryTechnician Response Parameters: None or success/error indicator Function: Adds a new laboratory technician to the system.
	Operation: removeLabStaff
	Request Parameters: labStaff: LaboratoryTechnician Response Parameters: None or success/error indicator Function: Removes an existing laboratory technician from the system.

Interface Name	<b>ChangelInfo</b>
Provider	SystemOperatorsManagment
	<b>Operation: changeLabInfo</b>
	Request Parameters: laboratoryName: String; labLocation: String; labKind: String; labStaffList: String; labEquipmentId: int; testId: int Response Parameters: None or success/error indicator Function: Updates laboratory information in the system.

Interface Name	<b>Login/Register</b>
Provider	Laboratory TechnicianStaff Management
	Operation: Register
	Request Parameters: username: String, password: String Response Parameters: None or success/error indicator Function: Registers a new user in the system with login credentials.
	Operation: Login
	Request Parameters: username: String, password: String Response Parameters: None or success/error indicator Function: Authenticates a user for access to the system.

Interface Name	<b>ChangePassword</b>
Provider	Laboratory TechnicianStaff Management
	Operation: ChangePassword
	Request Parameters: password: String Response Parameters: None or success/error indicator Function: Changes the password for the current user.

Interface Name	<b>OrderTest</b>
Provider	Doctor Management
	Operation: <code>getNewTestOrder</code> (
	Request Parameters: <code>newTest</code> : <code>TestOrder</code> Response Parameters: None or details of the test order Function: Retrieves a new test order from the system.

Interface Name	Receive and review test results
Provider	TestResult Management
	Operation: <code>updateResultDetails</code>
	Request Parameters: <code>newDetails</code> : <code>String</code> Response Parameters: None or success/error indicator Function: Updates the details of a test result in the system.



### 3.Test

#### 1. Unit Test Plan for Login/Register Interface

##### Test Cases for 'Register' Method

Test Case	Method and Parameters	Expected Output
Successful Registration	`Register("newUser", "Bobbyds ")`	'true'
Register with Existing Username	`Register("existingUser", "Bobbyds ")`	'false'
Register with Invalid Username	'Register(" ", "Bobbyds ")'	'false'
Register with Invalid Password	'Register("newUser", " ")'	'false'

##### Test Cases for login'Method

Test Cases	Method and Parameters	Expected Output
Valid Login	`login("authorizedUser", "Bobbyds ")`	'true'
Invalid Password	'login("authorizedUser", "WrongPassword")`	'false'
Non-Existent User	'login("nonExistentUser", "Bobbyds ")`	'false '
SQL Injection Attack	'login("';DROP TABLE users; --", "password")`	'false`

## 2. System Test Plan for Use Case: User Authentication and Authorization

Scenario: Secure Login for Laboratory Technician

Test Progress for Secure Login:

Actor (Laboratory Technician)	System (Laboratory Information System)
Navigates to the login page and enters credentials.	Prompts for username and password.
Submits the login request with valid credentials.	Validates credentials against the user database.
Awaits confirmation of successful login.	Grants access and directs the user to the appropriate dashboard based on role.

Test Data:

Input: userName = "authorizedUser" , 'password = "Bobbyds"

Expected Output: Access granted to laboratory technician dashboard.

Test Progress for Registration:

Actor (New User)	System (Laboratory Information System)
Navigates to the registration page and enters new user details.	Prompts for new username and password.
Submits the registration request.	Validates the new credentials and ensures the username is unique.
Awaits confirmation of successful .	Creates a new user account and confirms registration to the user.

Test Data:

Input: userName = "newTechnician" , password = "Bobbyds"

Expected Output: Confirmation of successful registration

Additional Scenarios:

**Role-Based Access Control:** After login, attempt to access patient data outside the laboratory technician's role permissions.

Expected Output: Access to sensitive data should be denied.

**Session Management:** After a period of inactivity, the system should automatically logout the technician.

Expected Output: Session timeout with a prompt to log in again.

These test plans ensure that the system supports the high-priority use case of secure user authentication and authorization, specifically tailored to laboratory technician staff. The unit tests validate the functionality of registration and login methods, while the system tests confirm the end-to-end process of user access control, from secure login to role-based permissions and session management.