Health Care System

Problem Statement

To create a user-friendly java application for a Health Care System.

Description

The system consists of various diagnostic centers, with their own tests which the customers can opt for. The system should be accessible to two types of users, namely admin and customer. Admin has control over adding, removing and updating diagnostic centers and the tests under them.

Customer makes an appointment after selecting a proper date-time as well as the test and diagnostic center. The appointment will be confirmed only after the admin approves it.

Scope

User can select only one test at a time. Thus, an appointment can be only for 1 test at a time inside a center. Users have to make an appointment for a test under any diagnostic center and that appointment has to be approved by an admin.

Out of Scope

An appointment will be in the database till it is manually deleted by the database admins.

User methods:

+Add Center (Admin)

Admin can add new centers. Center name has to be specified and the centerId will be generated automatically.

+Remove Center (Admin)

A center can be removed from the system by the admin. The center object will be removed from the centerList and then destroyed.

+Add Test(Admin)

Admin first selects a diagnostic center and then adds a new test in it.

+Remove Test(Admin)

Admin first selects a diagnostic center and then gets a list of all tests available inside the selected diagnostic center. Any test can be removed from the list and saved.

+Approve Appointment(

Admin first get list of all centers, then gets the appointmentList from that center. Admin can then approve or reject the appointments.

+makeAppointment(user: User, center: DiagnosticCenter, test: Test ,datetime: Datetime):BigInteger

User selects a test in a diagnostic center and books a datetime for the same. This appointment is added in the AppointmentList of the Diagnostic center the test is booked in.

register(): String

User id is generated automatically. By default user role is set to customer.

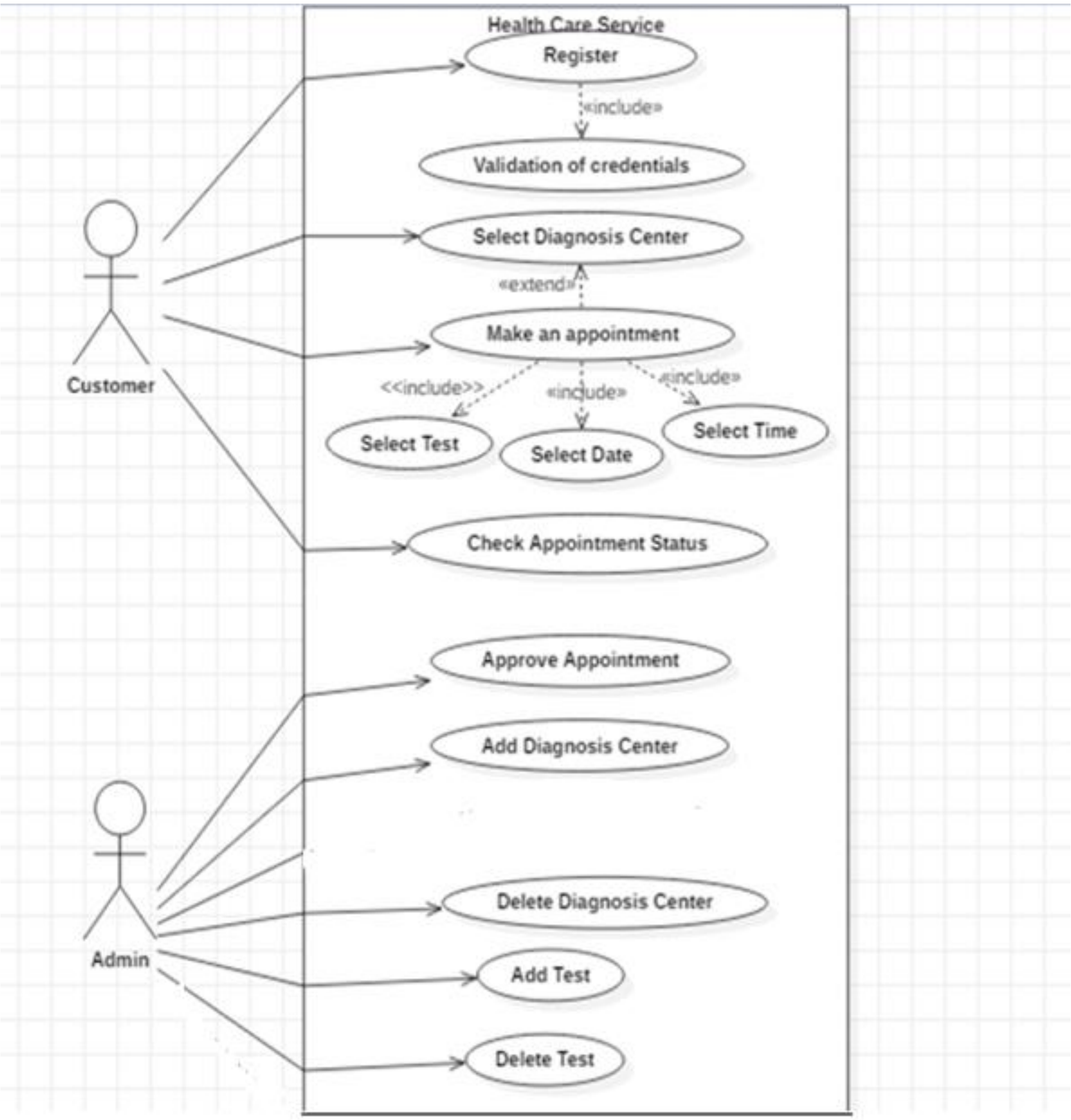
Assumptions and validation:

1. Register - (customer):

    a. Userid: Automatically generated.
    b. Password: Password must contain one number,one alphabet,one special character and size should be at least 8 characters and not more than 14 characters.
    c. Phone: Phone number should be 10 digits.
    d. Name: Name should not start with special characters and should not be blank. First letter should be capital.
    e. For registration, default role is user.
2. Make an Appointment -
    a. User gets a list of diagnostic centers, and he selects one of them.
    b. After selecting a center, user can select the test he wants from the list of available te sts in that center.
    c. After submission of the form, the new appointment will be added in the "Appointment List", so that the admin can approve it.
    d. One appointment can be for only one particular test.A new Test can be created by an admin which can contain multiple test, like a superset of two of more tests, however that super test will be independent of the others and will be treated as one.
3. Add Center:
    a. Center ID will be generated automatically.
    b. A newly added Diagnostic Center will have 3 default tests present in it, namely blood group, blood sugar and blood pressure. Admin can remove or update them later if needed.
4. Add Test:
    a. TestID will be generated automatically.
5. Remove Center:
    a. All the test objects under the center will be destroyed.
6. Approve Appointment:

a. Admin retrieves a list of all centers and selects one. A list of all appointments corresponding to the center he selects is shown to him, which he can approve/disapprove.

Use case Diagram:



Classes: For this project, we are creating 4 Classes.
1. User
2. Diagnostic Center

3. Test
4. Appointment

Class Description and roles

A. DTO Layer

1. **Diagnostic Center:** *For each diagnostic center the admin will create.*
Each center has a unique centerId and a list of available tests under it. Any new appointment made for the tests under a particular diagnostic center will be added in the “appointmentList”. This appointment list will be viewed by the admin and reviewed.

Attributes:
-centerName: String
-centerId: String
-listOfTests: List<Test>
-appointmentList: List<Appointment>

2. **Test:** *For each test present under a diagnostic center.*
Each test will have a unique id across all diagnostic centers. Tests will be saved in the listOfTests under the corresponding DiagnosticCenter.

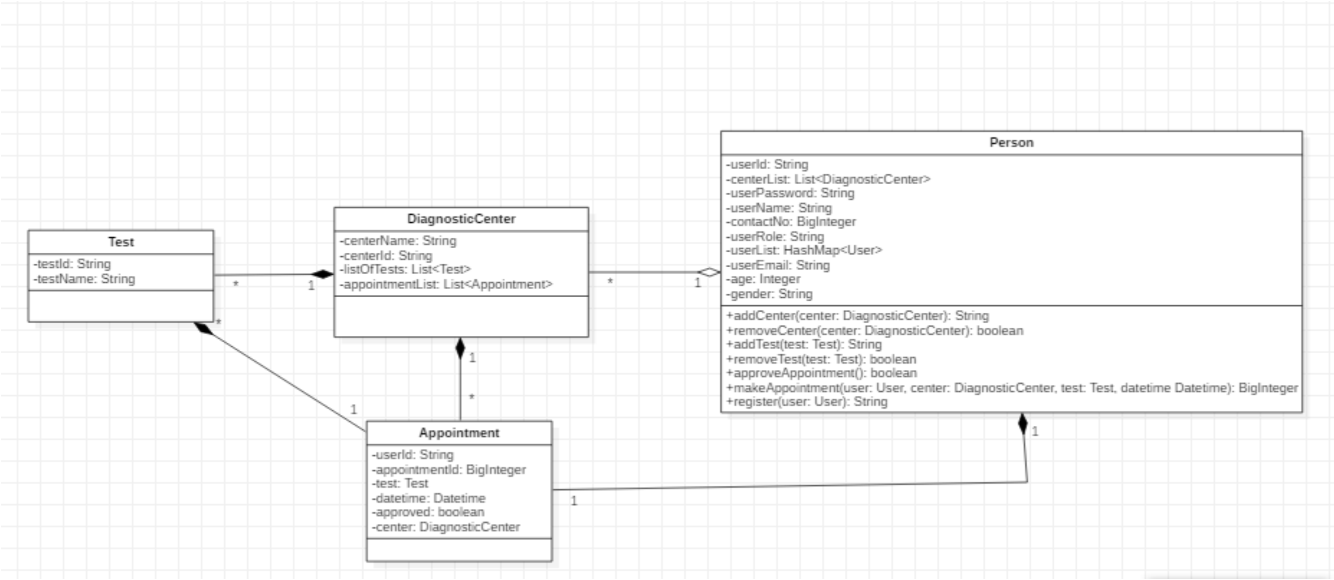
Attributes:
-testId String
-testName String

3. **Appointment:** Saves the details of the appointment the customer makes. Each appointment will have information of the user, the Test it is for, the DiagnosticCenter it is placed in and the date-time. Each appointment has to be approved by the admin.
-user User;
-appointmentId: BigInteger
-test: Test
-datetime: DateTime
-approved: boolean

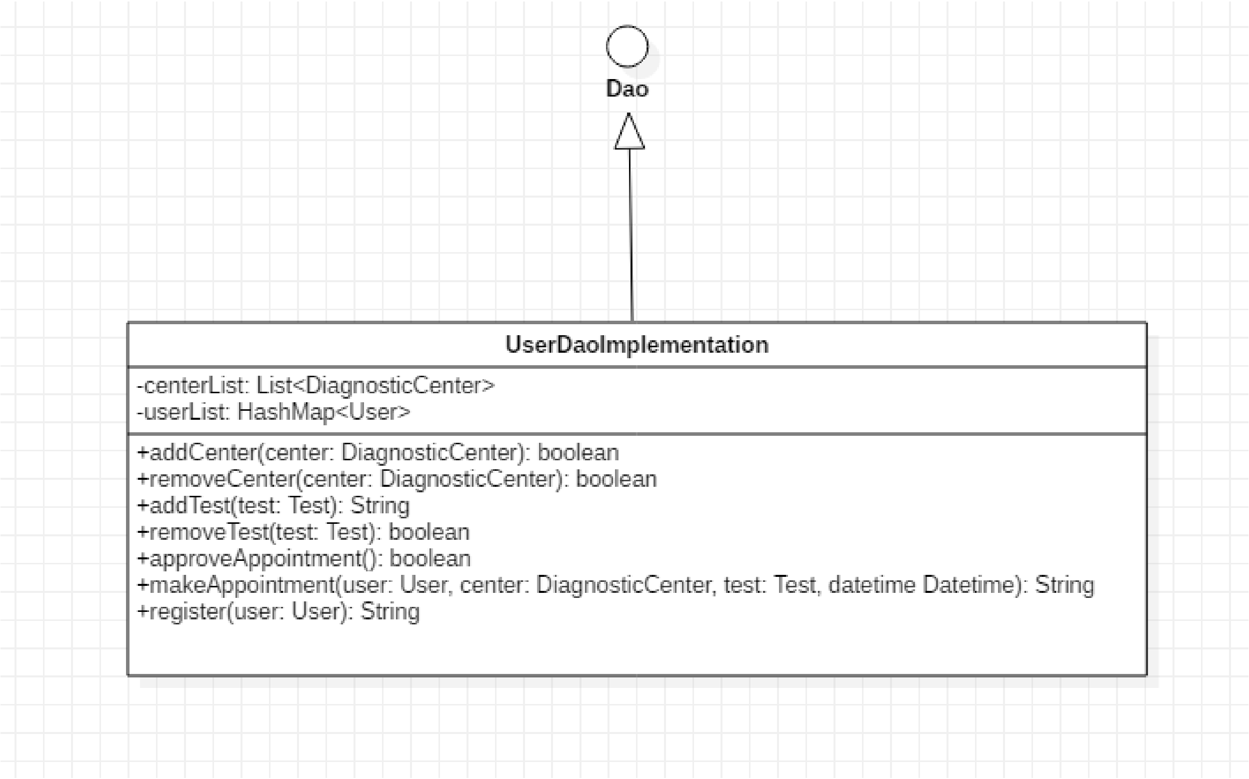
4. **User:** Depending on the type of user, a user role specific menu will be shown.
-userId: String
-centerList: List<DiagnosticCenter>

-userPassword: String
-userName: String
-contactNo: BigInteger
-userRole: String
-emailId: String

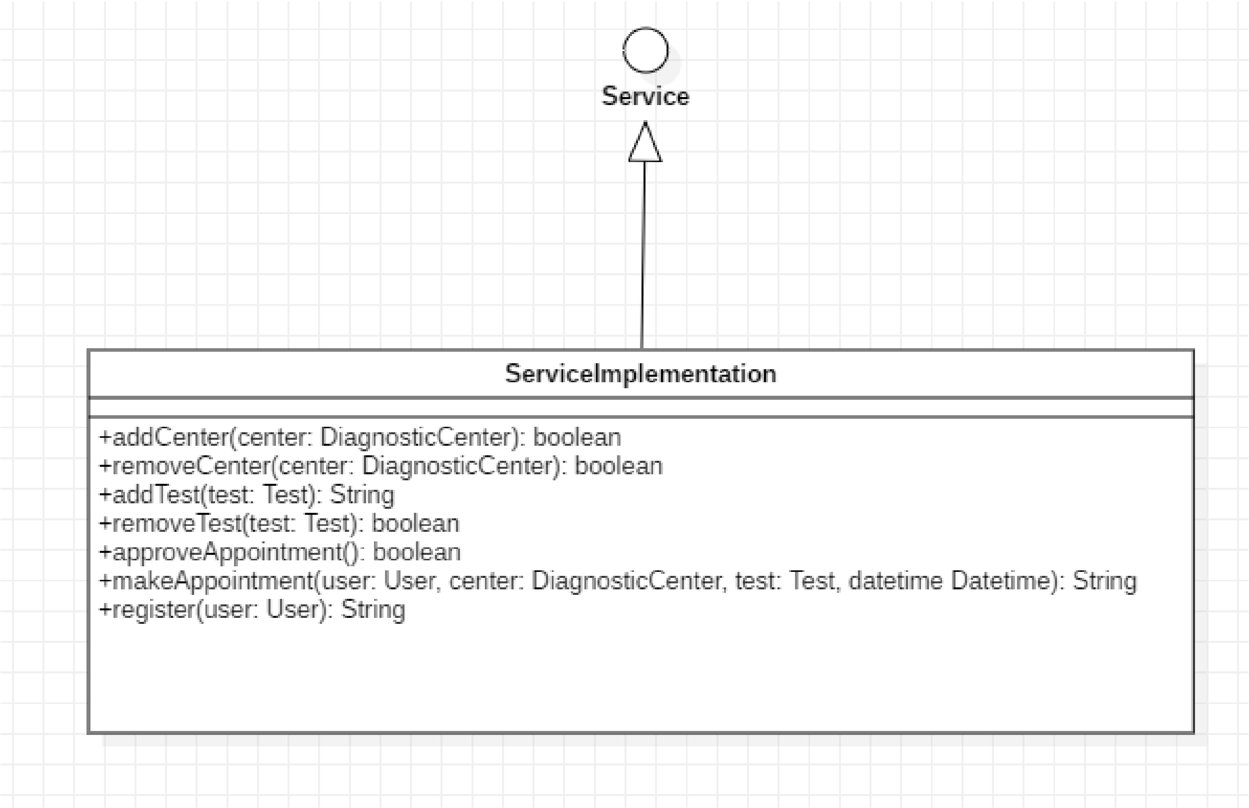
Class diagram for DTO layer:


Test
-testId: String
-testName: String
DiagnosticCenter
-centerName: String
-centerId: String
-listOfTests: List<Test>
-appointmentList: List<Appointment>
Person
-userId: String
-centerList: List<DiagnosticCenter>
-userPassword: String
-userName: String
-contactNo: BigInteger
-userRole: String
-userList: HashMap<User>
-userEmail: String
-age: Integer
-gender: String
+addCenter(center: DiagnosticCenter): String
+removeCenter(center: DiagnosticCenter): boolean
+addTest(test: Test): String
+removeTest(test: Test): boolean
+approveAppointment(): boolean
+makeAppointment(user: User, center: DiagnosticCenter, test: Test, datetime Datetime): BigInteger
+register(user: User): String
Appointment
-userId: String
-appointmentId: BigInteger
-test: Test
-datetime: Datetime
-approved: boolean
-center: DiagnosticCenter
*
1
*
1
*
1
1
*
1
1

B. DAO layer


Dao
UserDaoImplementation
-centerList: List<DiagnosticCenter>
-userList: HashMap<User>
+addCenter(center: DiagnosticCenter): boolean
+removeCenter(center: DiagnosticCenter): boolean
+addTest(test: Test): String
+removeTest(test: Test): boolean
+approveAppointment(): boolean
+makeAppointment(user: User, center: DiagnosticCenter, test: Test, datetime Datetime): String
+register(user: User): String

C. Service Layer:

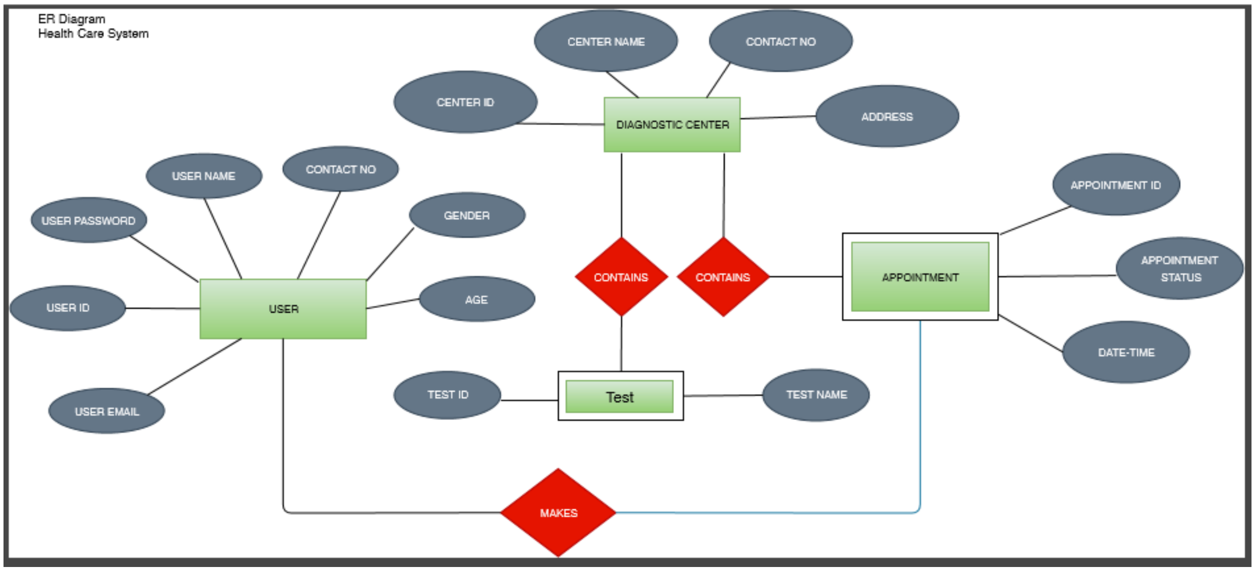

Service
ServiceImplementation
+addCenter(center: DiagnosticCenter): boolean
+removeCenter(center: DiagnosticCenter): boolean
+addTest(test: Test): String
+removeTest(test: Test): boolean
+approveAppointment(): boolean
+makeAppointment(user: User, center: DiagnosticCenter, test: Test, datetime Datetime): String
+register(user: User): String

Relationships:

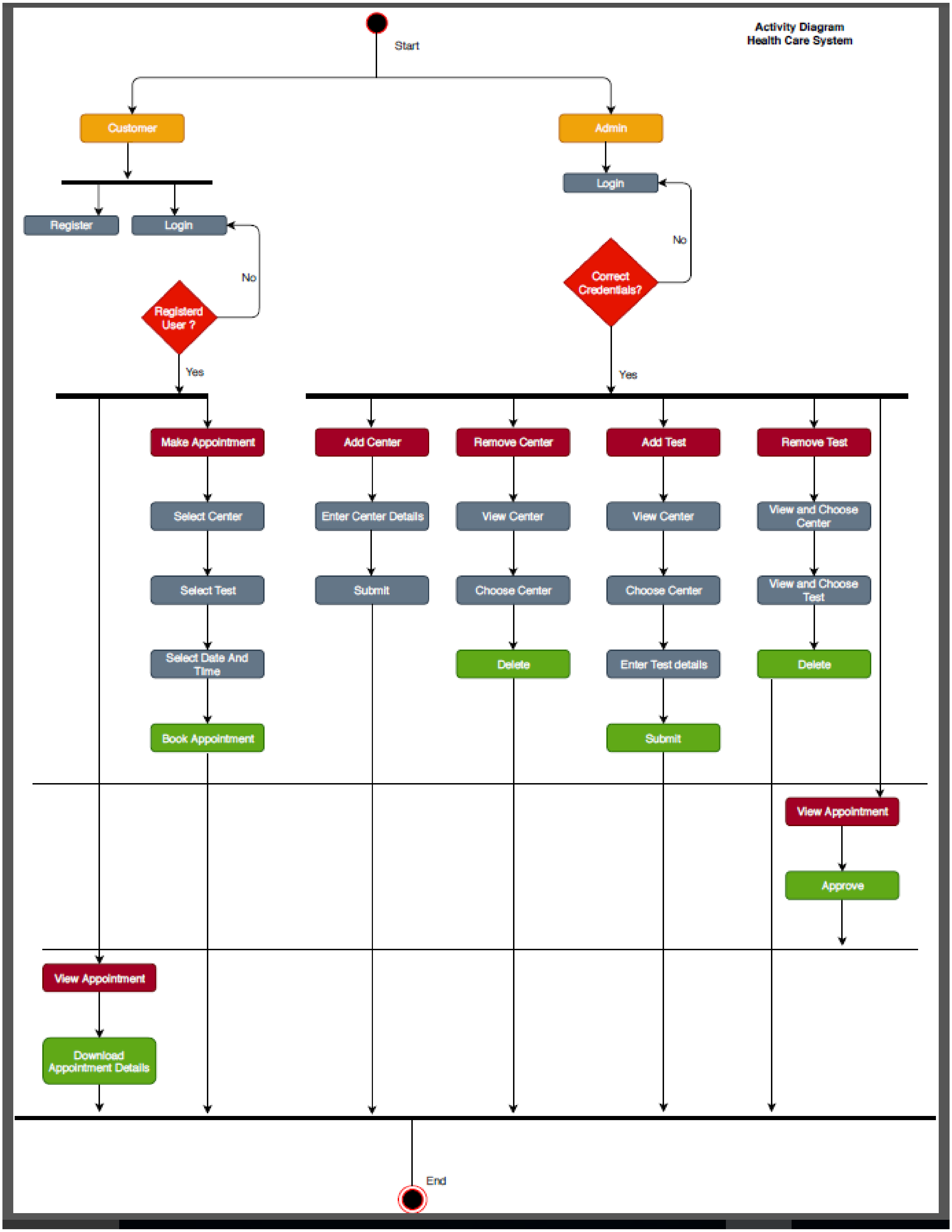
1. HealthCareSystem and DiagnosticCenter have **1 to many relationship.**
2. DiagnosticCenter and Test have **1 to many relationship.**
3. Appointment and DiagnosticCenter have **many to one relationship.**
4. Appointment and User have **one to one relationship**.

Diagrams:

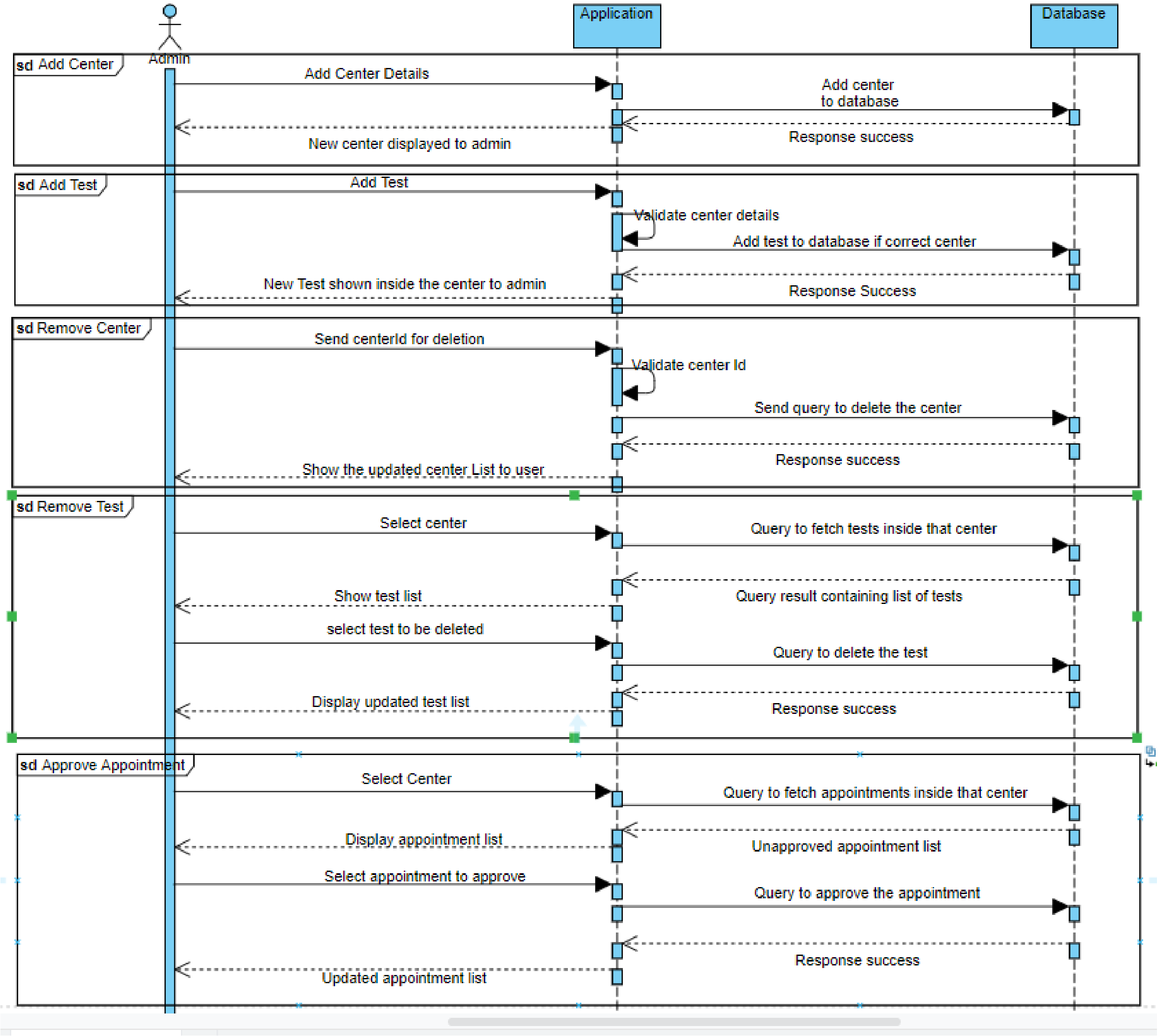
ER Diagram:



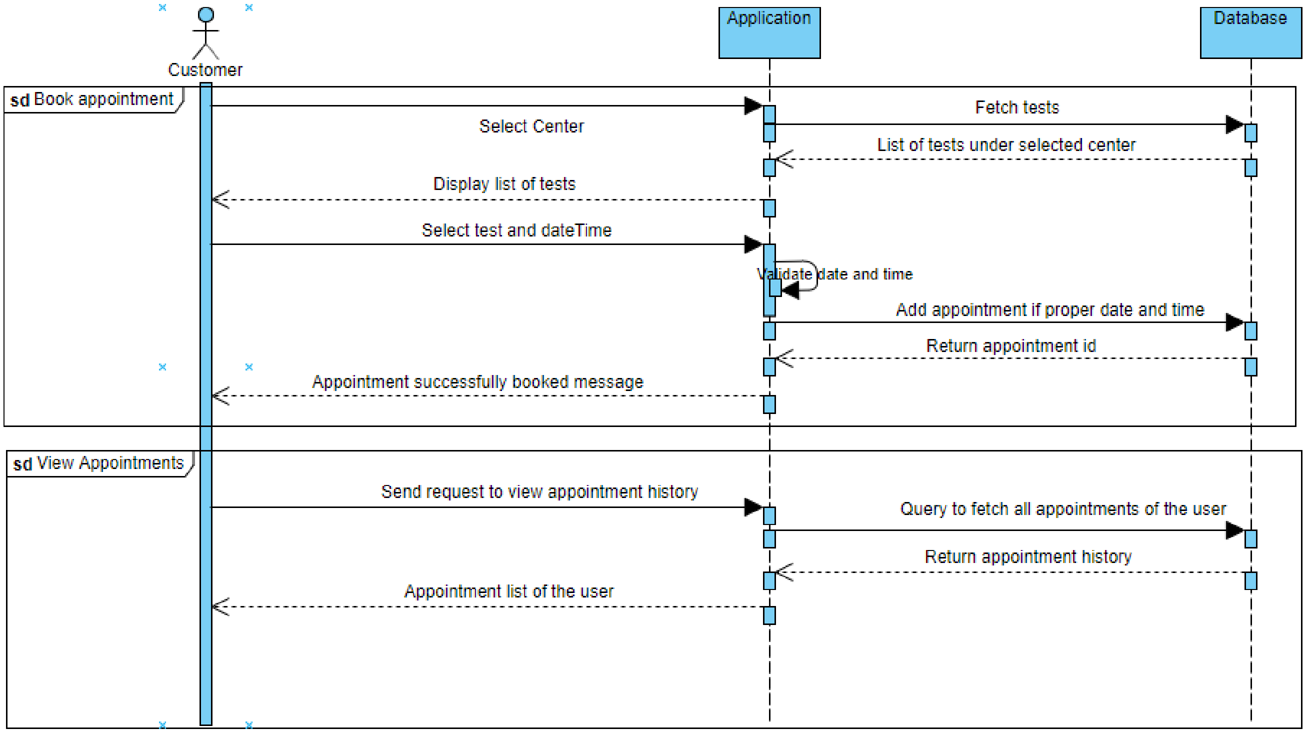
Activity Diagram :


Activity Diagram
Health Care System
Start
Customer
Admin
Register
Login
Login
No
Registerd User ?
Yes
No
Correct Credentials?
Yes
Make Appointment
Select Center
Select Test
Select Date And Time
Book Appointment
Add Center
Enter Center Details
Submit
Remove Center
View Center
Choose Center
Delete
Add Test
View Center
Choose Center
Enter Test details
Submit
Remove Test
View and Choose Center
View and Choose Test
Delete
View Appointment
Approve
View Appointment
Download Appointment Details
End

Sequence Diagram: ADMIN

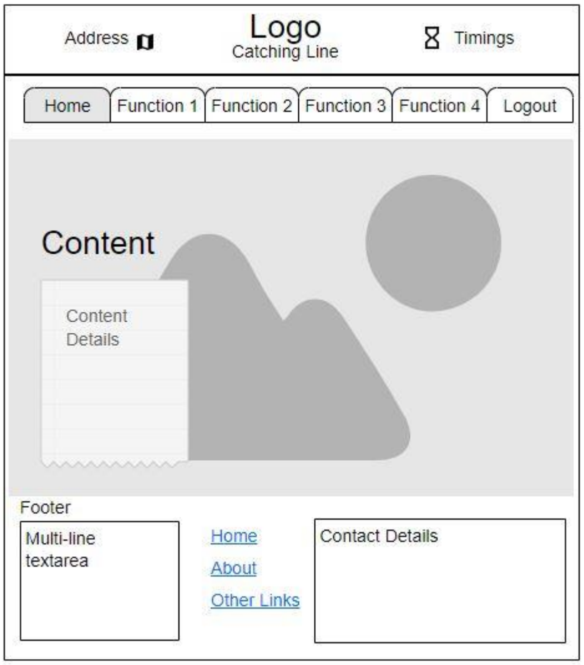

Admin
Application
Database
sd Add Center
Add Center Details
Add center
to database
Response success
New center displayed to admin
sd Add Test
Add Test
Validate center details
Add test to database if correct center
New Test shown inside the center to admin
Response Success
sd Remove Center
Send centerId for deletion
Validate center Id
Send query to delete the center
Response success
Show the updated center List to user
sd Remove Test
Select center
Query to fetch tests inside that center
Show test list
Query result containing list of tests
select test to be deleted
Query to delete the test
Display updated test list
Response success
sd Approve Appointment
Select Center
Query to fetch appointments inside that center
Display appointment list
Unapproved appointment list
Select appointment to approve
Query to approve the appointment
Response success
Updated appointment list

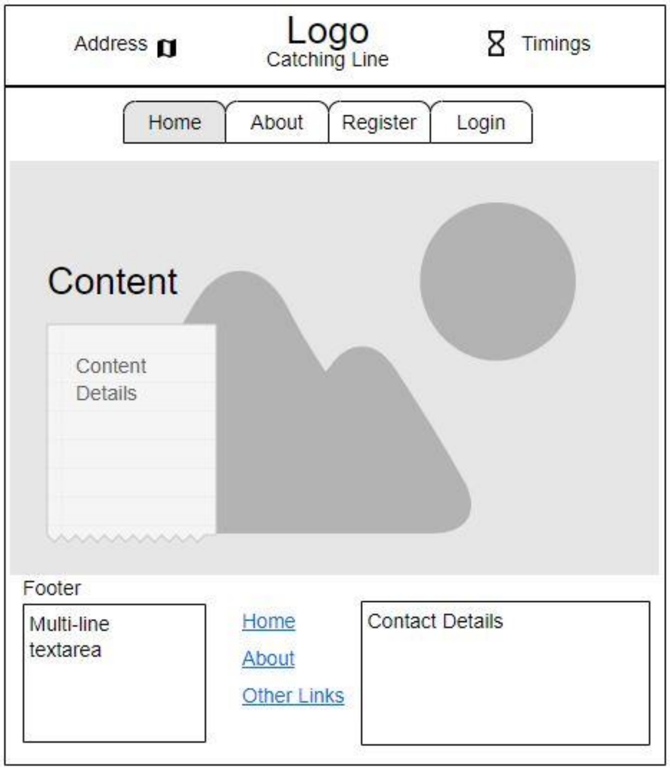
Sequence Diagram: Customer

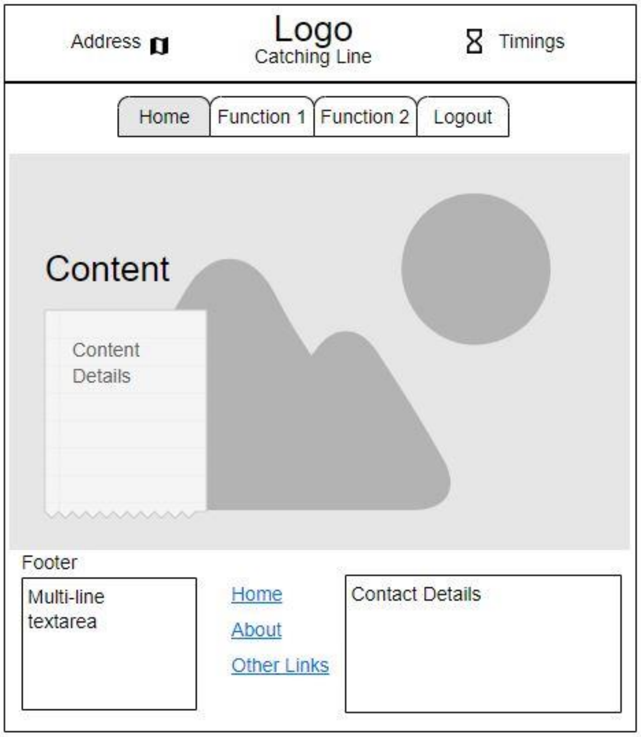


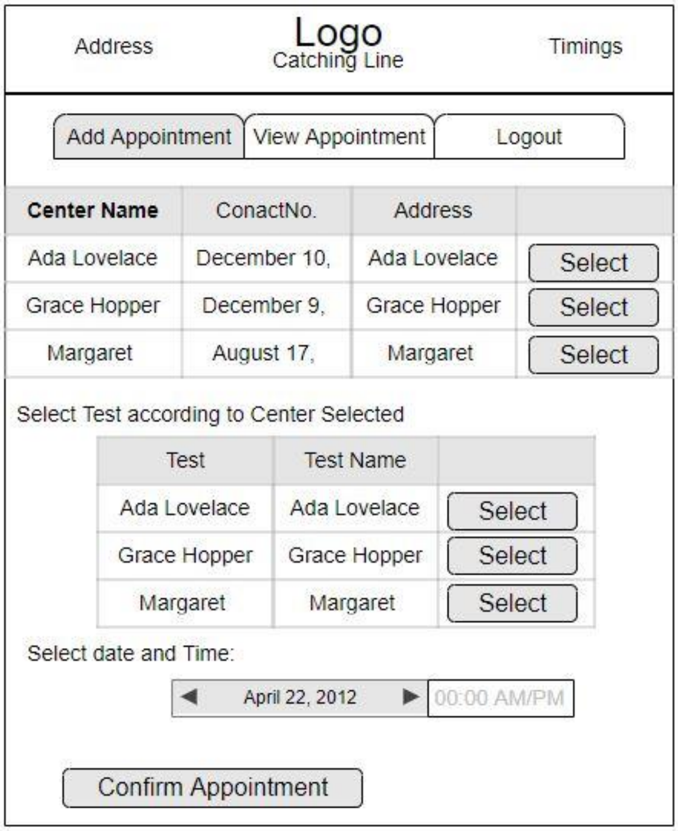
Customer
Application
Database
sd Book appointment
Select Center
Fetch tests
List of tests under selected center
Display list of tests
Select test and dateTime
Validate date and time
Add appointment if proper date and time
Return appointment id
Appointment successfully booked message
sd View Appointments
Send request to view appointment history
Query to fetch all appointments of the user
Return appointment history
Appointment list of the user

Wire Frames : Home Pages




Address
Logo
Catching Line
Timings
Home
About
Register
Login
Content
Content Details
Footer
Multi-line textarea
Home
About
Other Links
Contact Details


Address
Logo
Catching Line
Timings
Home
Function 1
Function 2
Logout
Content
Content Details
Footer
Multi-line textarea
Home
About
Other Links
Contact Details



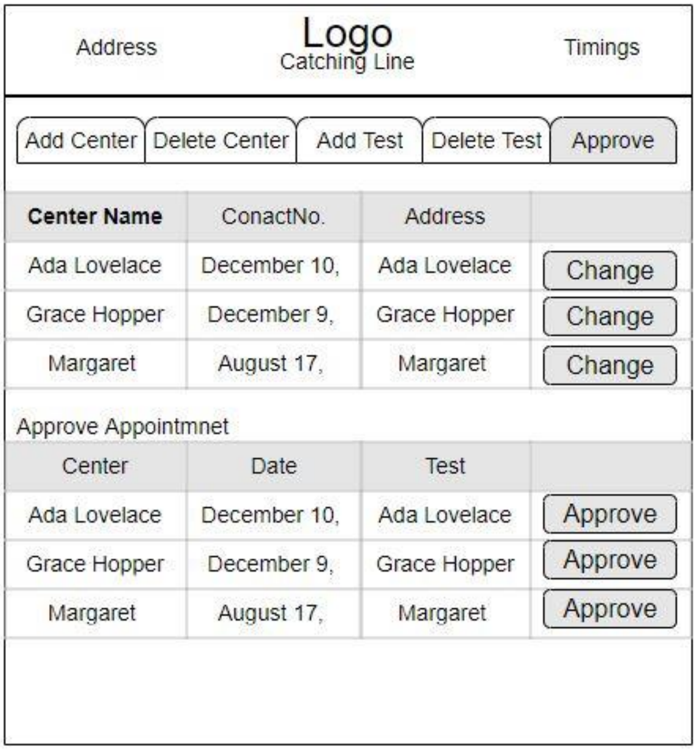
Center Name	ConactNo.	Address	
Ada Lovelace	December 10,	Ada Lovelace	Select
Grace Hopper	December 9,	Grace Hopper	Select
Margaret	August 17,	Margaret	Select

Select Test according to Center Selected

Test	Test Name	
Ada Lovelace	Ada Lovelace	Select
Grace Hopper	Grace Hopper	Select
Margaret	Margaret	Select

Select date and Time:

◀ April 22, 2012 ▶ 00:00 AM/PM



Center Name	ConactNo.	Address	
Ada Lovelace	December 10,	Ada Lovelace	Change
Grace Hopper	December 9,	Grace Hopper	Change
Margaret	August 17,	Margaret	Change

Approve Appointmnet

Center	Date	Test	
Ada Lovelace	December 10,	Ada Lovelace	Approve
Grace Hopper	December 9,	Grace Hopper	Approve
Margaret	August 17,	Margaret	Approve