

PES UNIVERSITY
100 feet Ring Road, BSK 3rd Stage
Bengaluru 560085



Department of Computer Science and Engineering
B. Tech. CSE - 6th Semester
Jan – May 2023

UE20CS352
Object Oriented Analysis and Design with Java

Project Report

Forensic Evidence Management System

TEAM #:

PES1UG20CS633: Bhoomika

PES1UG20CS627: ARPITA KADAGAD

PES1UG20CS615: Nayana D Raj

Class of Prof.Bhargavi Mokashi

Problem Statement(synopsis)

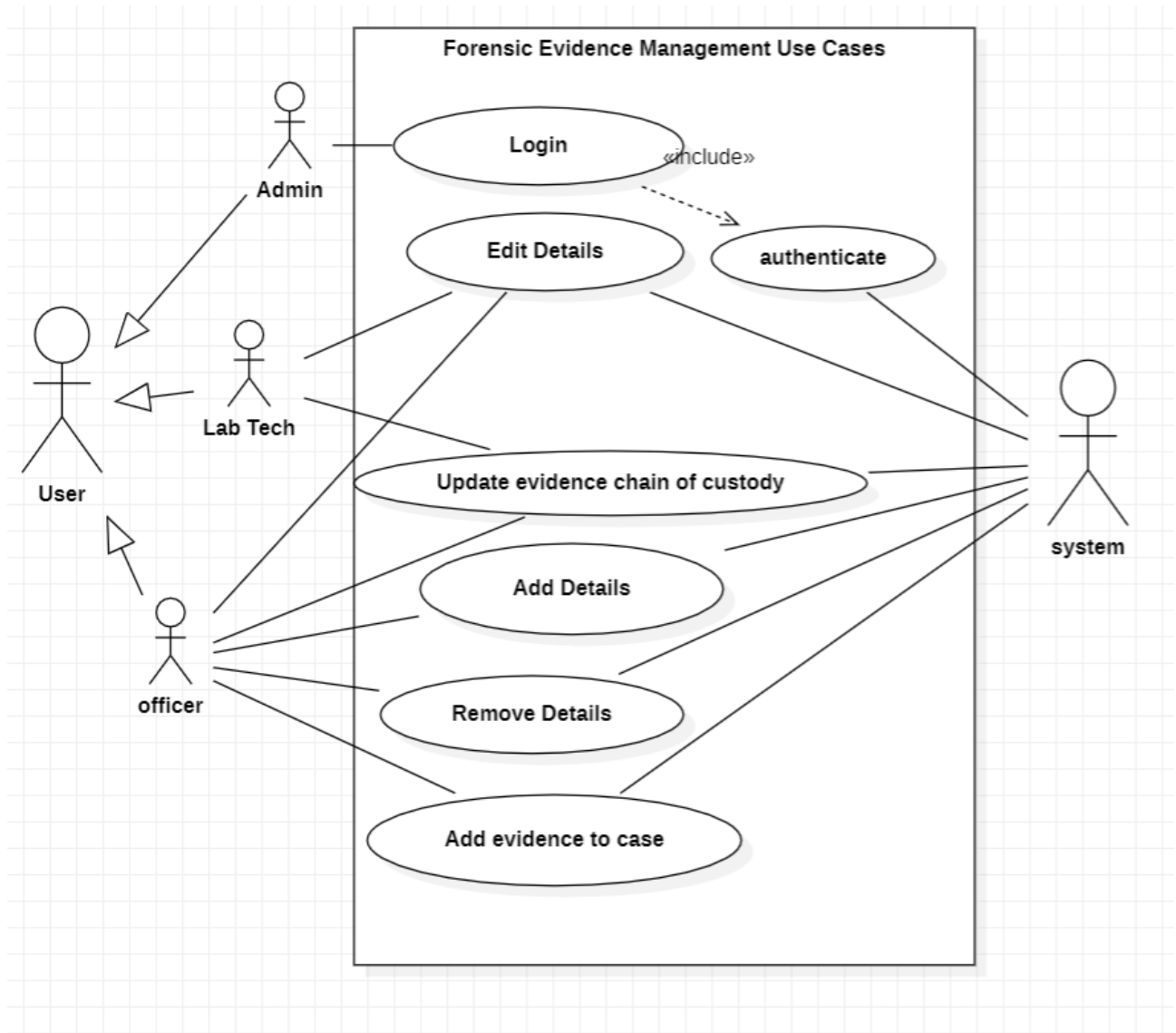
Forensic Evidence Management System is an Object-Oriented Analysis and Design project that aims to improve the management of forensic evidence in a law enforcement agency. The system will provide a centralized platform for storing, tracking and analyzing evidence data and will also enable the sharing of evidence information between different agencies.

The project will be divided into several phases, starting with requirements gathering and analysis. Once the requirements have been identified, the team will move on to the design phase, where they will create a use case diagram, class-diagram, state diagram and activity diagram.

After the design phase, the team will begin implementation, which will involve the development of the software using a suitable programming language i.e., Java and framework i.e., Spring and using spring tool suite IDE. The system will be designed to be scalable and secure, with appropriate access controls and data encryption mechanisms in place.

Overall, the Forensic Evidence Management System will provide a valuable tool for law enforcement agencies to manage and analyze forensic evidence data, leading to improved efficiency and effectiveness in solving crimes.

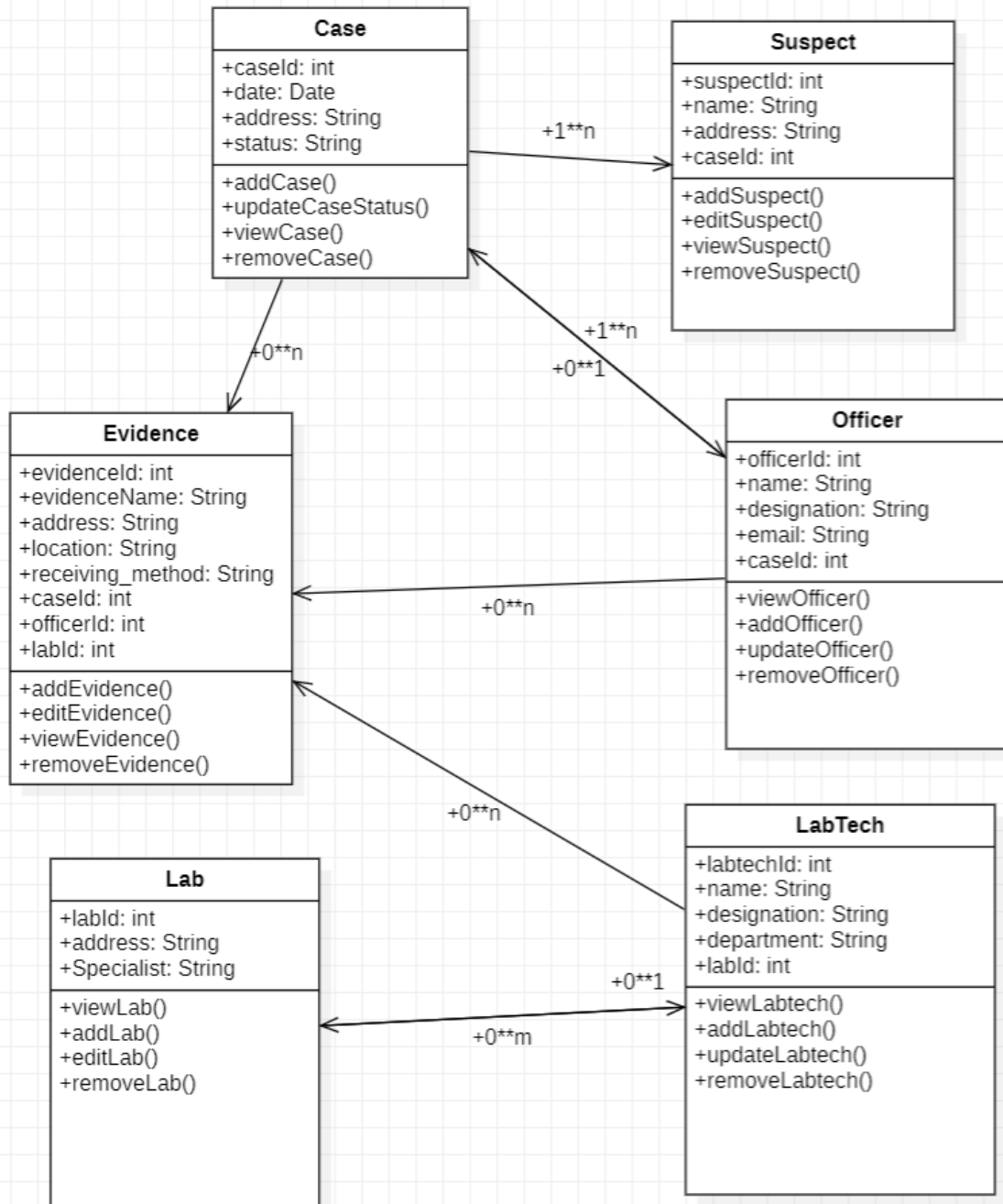
Use Case Model



Expanded Use case (Edit Details for Evidence entity)

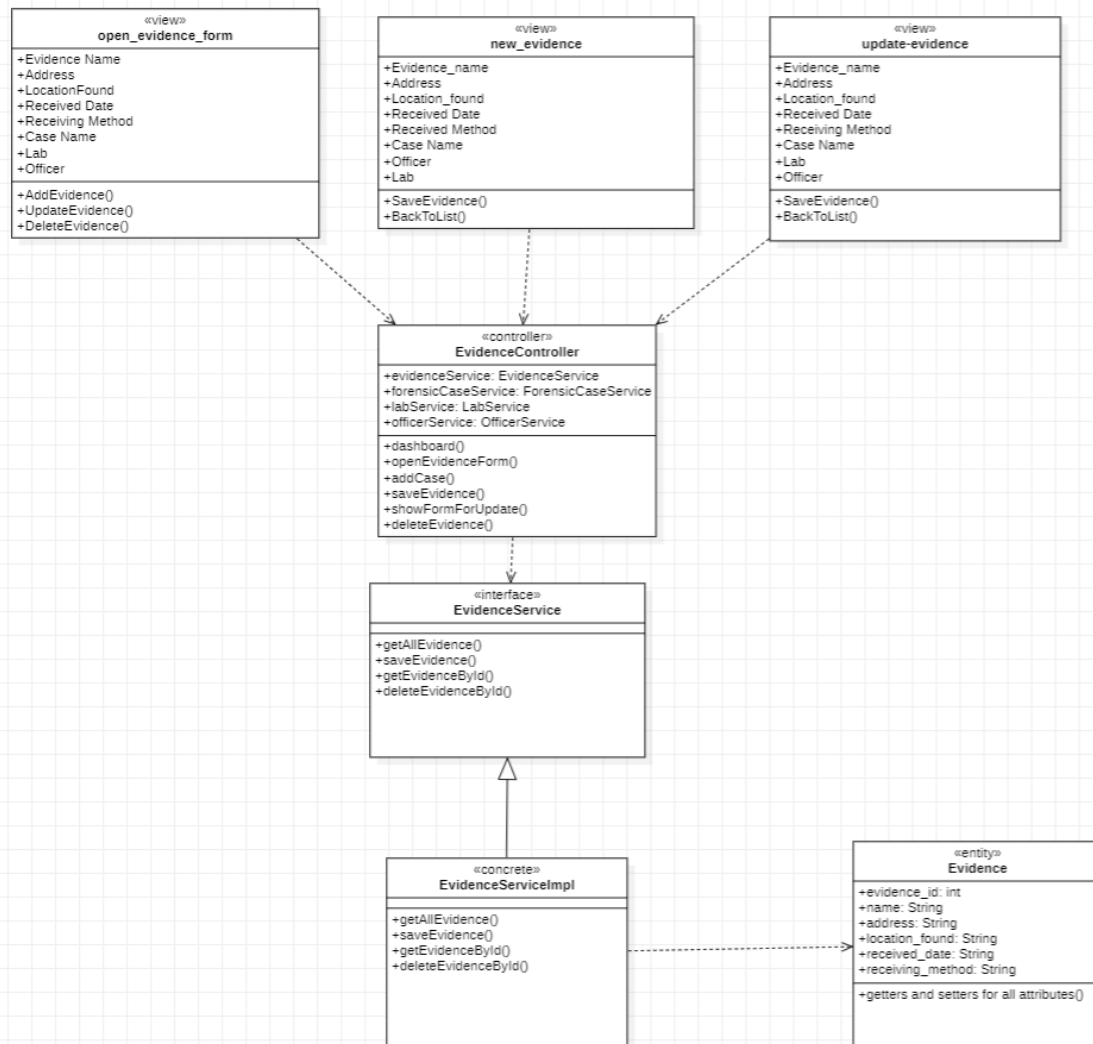


Class Model



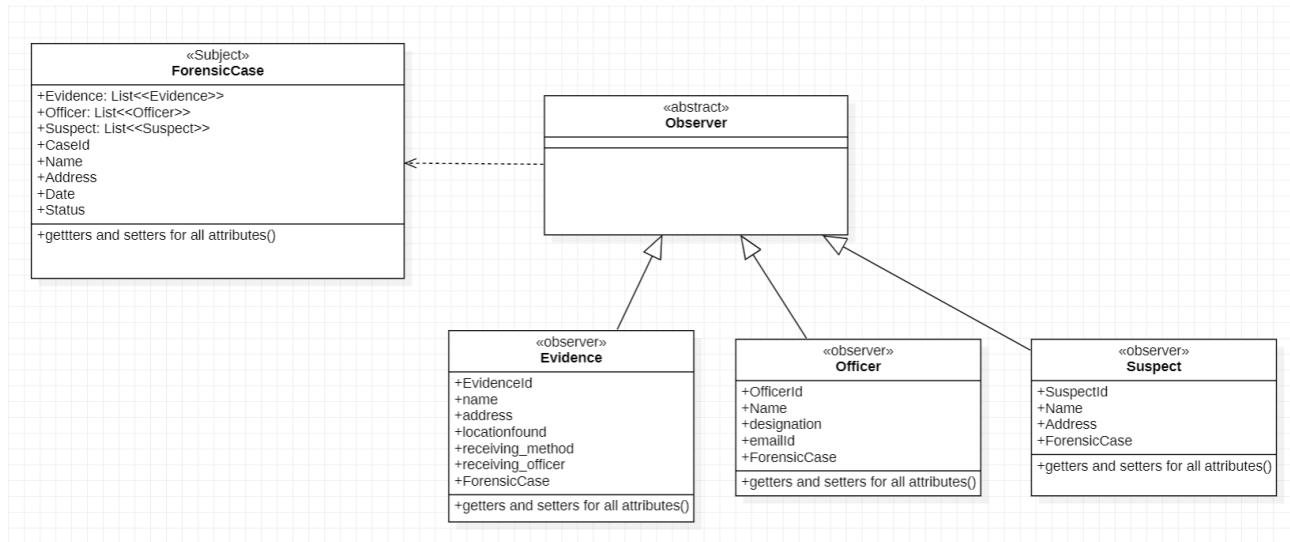
Updated Class Model

Dependency inversion principle



The same principle has been used for all entities like Forensic case, Suspect, Officer, Lab, Labtech.

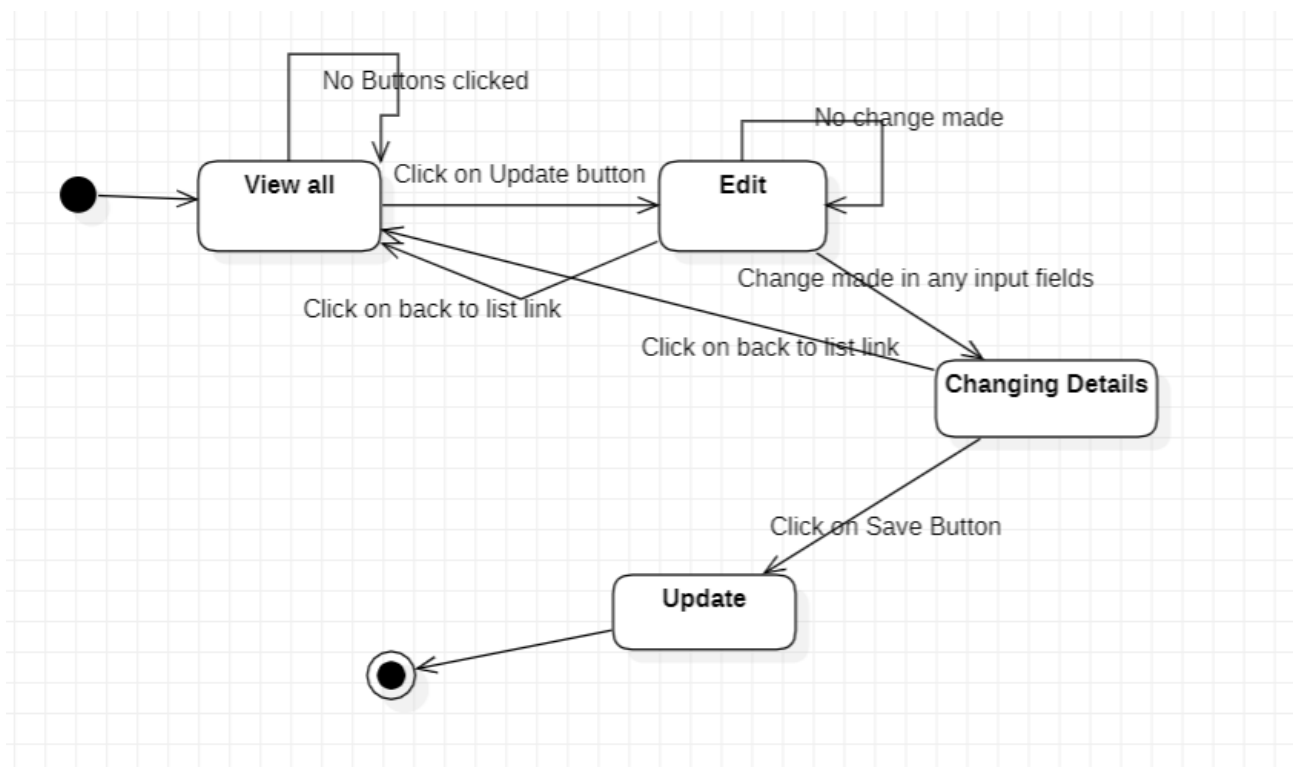
Observer Pattern



The same pattern has been used for relationship between Evidence, Labtech and Lab.

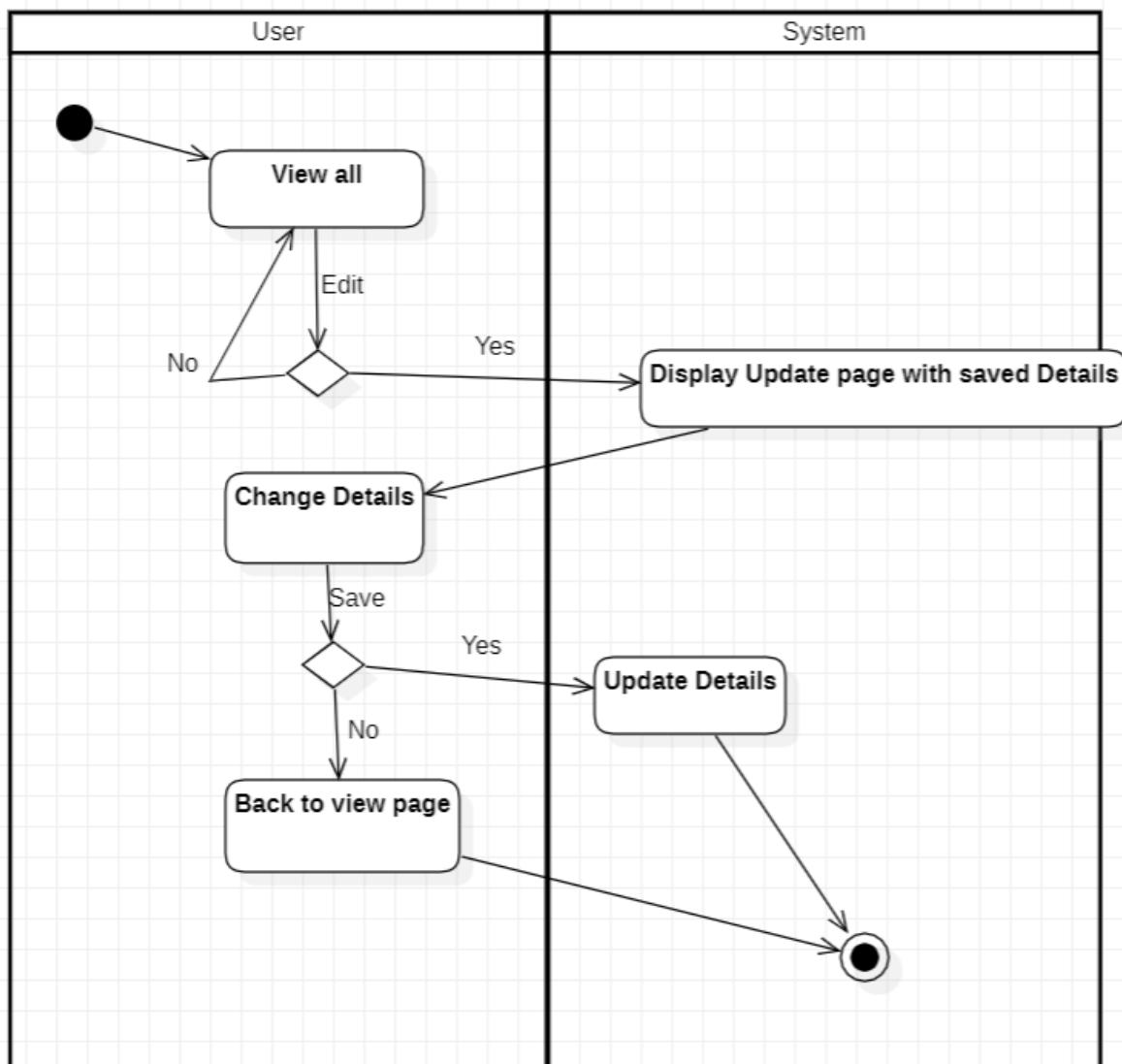
State Diagram:-

Edit Details use case



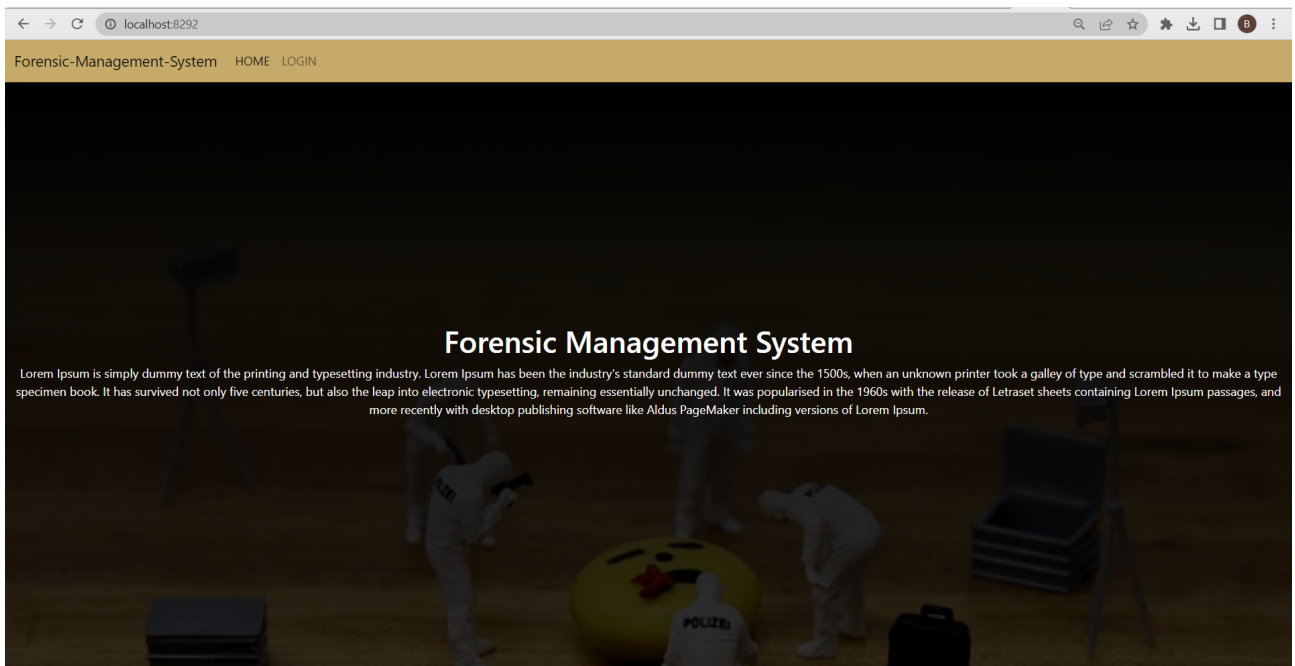
Activity Diagram:-

Edit Details use case



Output Screenshots:-

Home



Login

Forensic-Management-System

HOME

LOGIN

Login Page

Username

Password

Sign Up

Dashboard

The image is a screenshot of a web browser displaying a web application. The browser's address bar at the top shows the URL 'localhost:8292/home2'. The web application has a yellow header bar containing the text 'Forensic-Management-System' on the left, and 'HOME' and 'LOGOUT' on the right. A vertical sidebar on the left side of the page lists several menu items: 'Cases', 'Suspects', 'Officer', 'Evidence', 'Lab', and 'LabTech'. The main body of the page has a dark gray background and features the text 'Forensic Management System' centered in a large, white, sans-serif font.

CRUD Operation:-

Forensic-Management-System

HOME

LOGOUT

Cases

Suspects

Officer

Evidence

Lab

LabTech

Case List

Add Case

Case Id	Case Name	Case Address	Case Date	Case Status	Actions
---------	-----------	--------------	-----------	-------------	---------

Add

Forensic-Management-System

HOME

LOGOUT

Cases

Suspects

Officer

Evidence

Lab

LabTech

Forensic Management System

Save case

Case Name

Address

mm/dd/yyyy

Enter status

Save Case

[Back to List](#)

Update

Forensic-Management-System HOME LOGOUT

Cases
Suspects
Officer
Evidence
Lab
LabTech

Forensic Management System

Save case

Update Case

[Back to List](#)

Delete

Forensic-Management-System HOME LOGOUT

Cases
Suspects
Officer
Evidence
Lab
LabTech

Case List

Add Case

Case Id	Case Name	Case Address	Case Date	Case Status	Actions
2	case_1	addr1	2023-04-18	progress	<div>UpdateDelete</div>

Database

```
mysql> use forensic_management
```

```
Database changed
```

```
mysql> show tables;
```

```
+-----+
| Tables_in_forensic_management |
+-----+
| evidence                      |
| forensic_case                 |
| lab                           |
| labtech                      |
| officer                      |
| suspect                      |
+-----+
6 rows in set (0.01 sec)
```

Tables Updated

Forensic_case

```
mysql> select * from forensic_case;
```

```
+-----+-----+-----+-----+-----+
| case_id | address | date                | name  | status |
+-----+-----+-----+-----+-----+
|        2 | addr1   | 2023-04-17 18:30:00.000000 | case_1 | progress |
+-----+-----+-----+-----+-----+
1 row in set (0.01 sec)
```

Suspect

```
mysql> select * from suspect;
```

```
+-----+-----+-----+-----+
| suspect_id | address          | name          | case_id |
+-----+-----+-----+-----+
|          1 | suspect_addr_1  | suspect_1    |        2 |
+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

Lab

```
mysql> select * from lab;
```

lab_id	address	department
2	lab_addr_1	evidence_analysis

1 row in set (0.01 sec)

Labtech

```
mysql> select * from labtech;
```

labtech_id	department	designation	name	lab_id
1	evidence_analysis	junior_tech	labtech_1	2

1 row in set (0.00 sec)

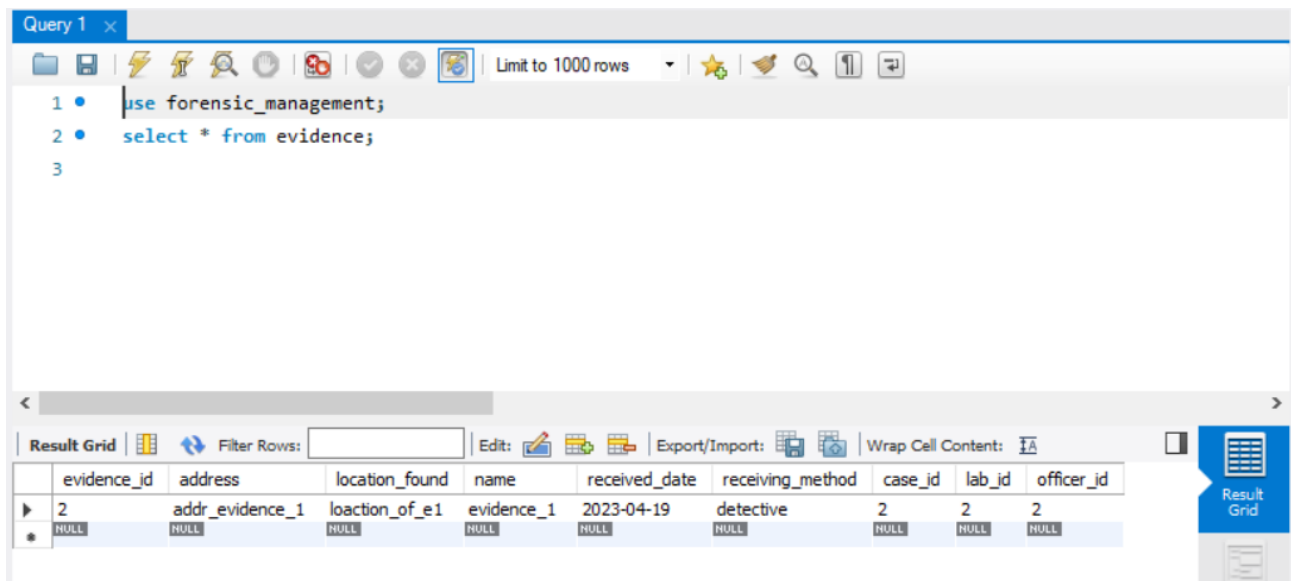
Officer

```
mysql> select * from officer;
```

officer_id	designation	email	name	case_id
2	Senior	abc@gmail.com	officer_1	2

1 row in set (0.00 sec)

Evidence



Design Pattern

Observer Pattern

For case-id initially case_name is case-9 after updating case_name to case-10 in case table it will be updated in evidence table also

Before updation case table:-

Forensic-Management-System

HOME

LOGOUT

Cases

Suspects

Officer

Evidence

Lab

LabTech

Case List

Add Case

Case Id	Case Name	Case Address	Case Date	Case Status	Actions
3	Case-9	Bangalore	2023-04-12	In progress	<div>UpdateDelete</div>
4	case-2	Mumbai	2023-04-06	cancelled	<div>UpdateDelete</div>
5	case-3	Udupi	2023-04-20	Closed	<div>UpdateDelete</div>
6	case-4	Kerala	2023-04-15	Initial	<div>UpdateDelete</div>

Before updation evidence table:-

Forensic-Management-System HOME LOGOUT									
Cases	Evidence List								
Suspects	Add Evidence								
Officer	Evidence Name	Address	Location Found	Received Date	Receiving Method	Case Name	Lab	Officer	Actions
Evidence	knife	JP Nagar	Table	2023-04-05	Detective	case-2	Analysis	KAYLING	Update Delete
Lab	mobile	girinagar	rack	2023-04-13	Lost_item	Case-9	evidence_management	Shyamu	Update Delete
LabTech	dairy_book	Jayanagar_home	Wardrobe	2023-04-13	Search	case-3	Analysis	Chaman	Update Delete

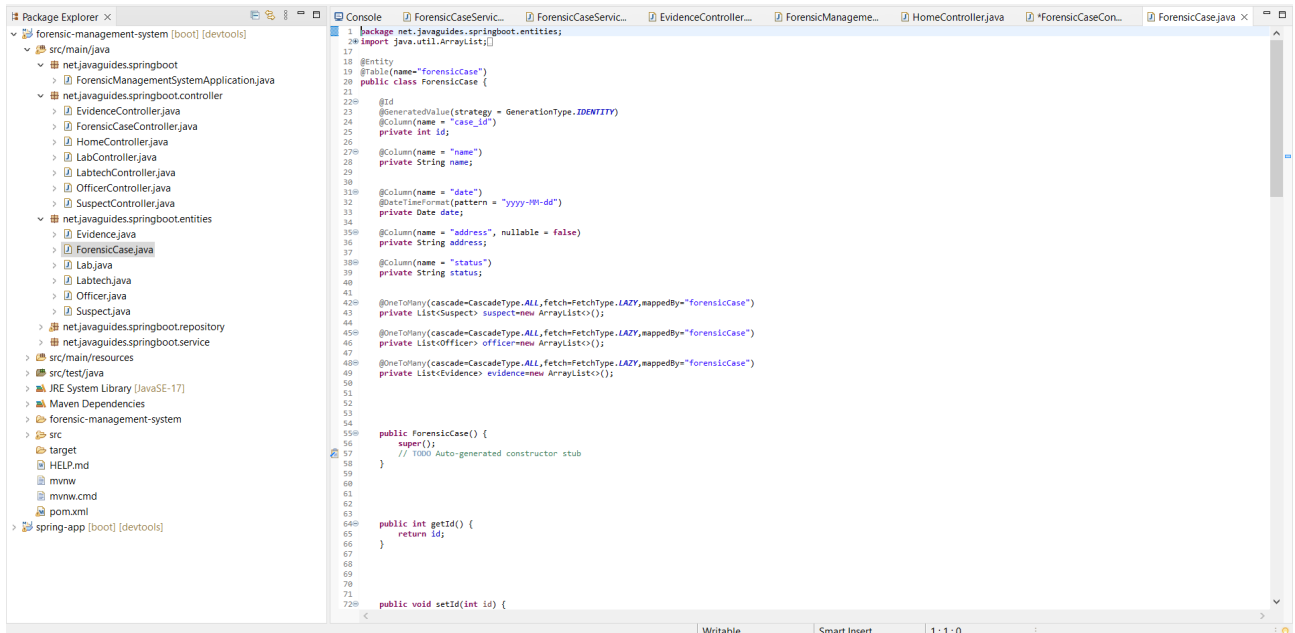
After updation in case table:-

Forensic-Management-System HOME LOGOUT						
Cases	Case List					
Suspects	Add Case					
Officer	Case Id	Case Name	Case Address	Case Date	Case Status	Actions
Evidence	3	Case-10	Bangalore	2023-04-12	In progress	Update Delete
Lab	4	case-2	Mumbai	2023-04-06	cancelled	Update Delete
LabTech	5	case-3	Udupi	2023-04-20	Closed	Update Delete
	6	case-4	Kerala	2023-04-15	Initial	Update Delete

After updation in evidence table:-

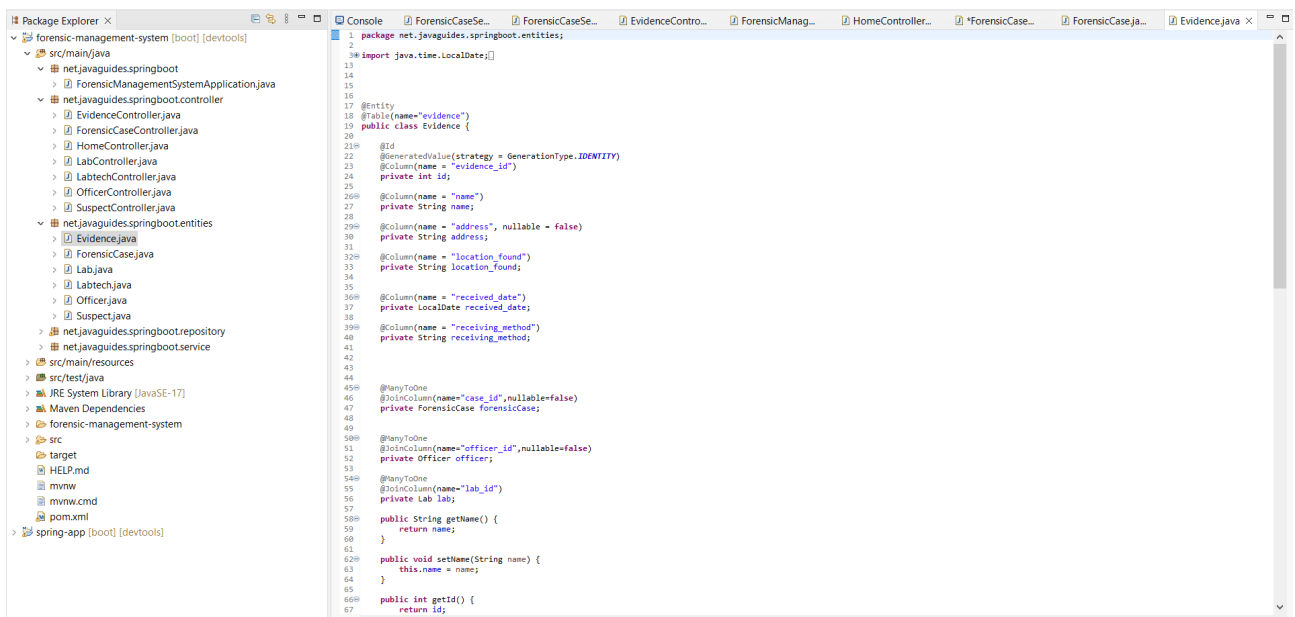
Forensic-Management-System HOME LOGOUT									
Cases	Evidence List								
Suspects	Add Evidence								
Officer	Evidence Name	Address	Location Found	Received Date	Receiving Method	Case Name	Lab	Officer	Actions
Evidence	knife	JP Nagar	Table	2023-04-05	Detective	case-2	Analysis	KAYLING	Update Delete
Lab	mobile	girinagar	rack	2023-04-13	Lost_item	Case-10	evidence_management	Shyamu	Update Delete
LabTech	dairy_book	Jayanagar_home	Wardrobe	2023-04-13	Search	case-3	Analysis	Chaman	Update Delete

Code for observer pattern:-



The screenshot shows an IDE with the Package Explorer on the left and the code editor on the right. The Package Explorer shows the project structure for 'forensic-management-system'. The code editor displays the 'ForensicCaseService' entity, which is a JPA entity with the following attributes:

```
1 package net.javaguides.springboot.entities;
2 import java.util.ArrayList;
3
4 @Entity
5 @Table(name="ForensicCase")
6 public class ForensicCase {
7
8     @Id
9     @GeneratedValue(strategy = GenerationType.IDENTITY)
10    @Column(name = "case_id")
11    private int id;
12
13    @Column(name = "name")
14    private String name;
15
16    @Column(name = "date")
17    @DateTimeFormat(pattern = "yyyy-MM-dd")
18    private Date date;
19
20    @Column(name = "address", nullable = false)
21    private String address;
22
23    @Column(name = "status")
24    private String status;
25
26    @OneToMany(cascade=CascadeType.ALL, fetch=FetchType.LAZY, mappedBy="ForensicCase")
27    private List<Suspect> suspect=new ArrayList<>();
28
29    @OneToMany(cascade=CascadeType.ALL, fetch=FetchType.LAZY, mappedBy="ForensicCase")
30    private List<Officer> officer=new ArrayList<>();
31
32    @OneToMany(cascade=CascadeType.ALL, fetch=FetchType.LAZY, mappedBy="ForensicCase")
33    private List<Evidence> evidence=new ArrayList<>();
34
35    public ForensicCase() {
36        super();
37        // TODO Auto-generated constructor stub
38    }
39
40    public int getId() {
41        return id;
42    }
43
44    public void setId(int id) {
```



The screenshot shows an IDE with the Package Explorer on the left and the code editor on the right. The Package Explorer shows the project structure for 'forensic-management-system'. The code editor displays the 'Evidence' entity, which is a JPA entity with the following attributes:

```
1 package net.javaguides.springboot.entities;
2 import java.time.LocalDate;
3
4 @Entity
5 @Table(name="evidence")
6 public class Evidence {
7
8     @Id
9     @GeneratedValue(strategy = GenerationType.IDENTITY)
10    @Column(name = "evidence_id")
11    private int id;
12
13    @Column(name = "name")
14    private String name;
15
16    @Column(name = "address", nullable = false)
17    private String address;
18
19    @Column(name = "location_found")
20    private String location_found;
21
22    @Column(name = "received_date")
23    private LocalDate received_date;
24
25    @Column(name = "receiving_method")
26    private String receiving_method;
27
28    @ManyToOne
29    @JoinColumn(name="case_id",nullable=false)
30    private ForensicCase forensicCase;
31
32    @ManyToOne
33    @JoinColumn(name="officer_id",nullable=false)
34    private Officer officer;
35
36    @ManyToOne
37    @JoinColumn(name="lab_id")
38    private Lab lab;
39
40    public String getName() {
41        return name;
42    }
43
44    public void setName(String name) {
45        this.name = name;
46    }
47
48    public int getId() {
49        return id;
50    }
```

Dependency Inversion principle:-

The Dependency Inversion Principle (DIP) states that high level modules should not depend on low level modules; both should depend on abstractions.

ForensicCaseController and ForensicCaseServiceImpl are dependent on ForensicCaseService Interface.

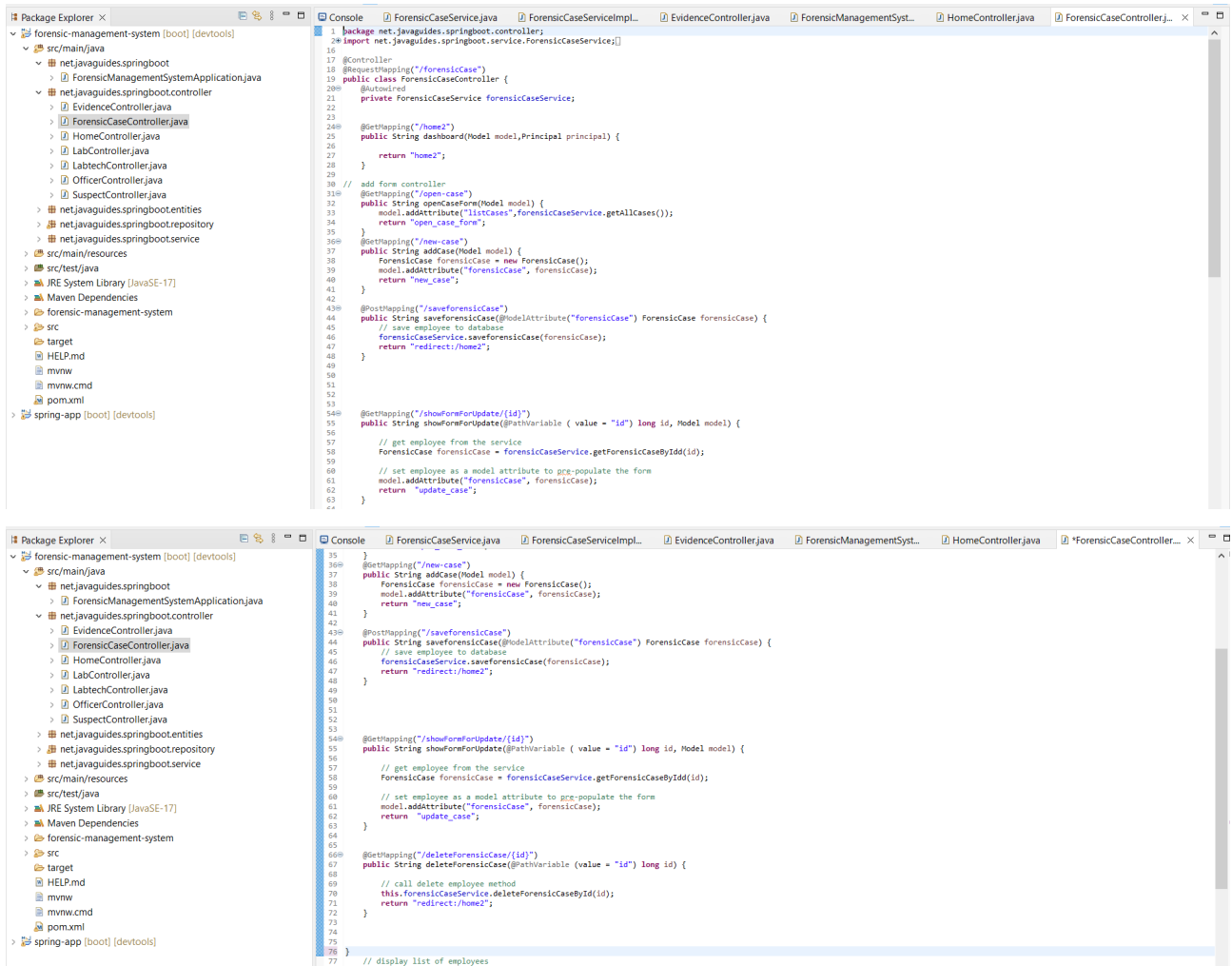
ForensicCaseService.java

```
1 package net.javaguides.springboot.service;
2
3
4 import java.util.List;
5
6
7
8
9
10
11 public interface ForensicCaseService {
12     List<ForensicCase> getAllCases();
13     void saveForensicCase(ForensicCase forensicCase);
14     ForensicCase getForensicCaseById(long id);
15     void deleteForensicCaseById(long id);
16     Page<ForensicCase> findPaginated(int pageNo, int pageSize, String sortField, String sortDirection);
17
18 }
19 }
```

ForensicCaseServiceImpl.java

```
1 package net.javaguides.springboot.service;
2
3 import java.util.List;
4
5
6
7
8
9
10
11
12
13
14
15
16
17 public class ForensicCaseServiceImpl implements ForensicCaseService {
18
19     @Autowired
20     private CaseRepository caseRepository;
21
22     @Override
23     public List<ForensicCase> getAllCases() {
24         return caseRepository.findAll();
25     }
26
27     @Override
28     public void saveForensicCase(ForensicCase forensicCase) {
29         this.caseRepository.save(forensicCase);
30     }
31
32     @Override
33     public ForensicCase getForensicCaseById(long id) {
34         Optional<ForensicCase> optional = caseRepository.findById(id);
35         ForensicCase forensicCase = null;
36         if (optional.isPresent()) {
37             forensicCase = optional.get();
38         } else {
39             throw new RuntimeException("ForensicCase not found for id :: " + id);
40         }
41         return forensicCase;
42     }
43
44     @Override
45     public void deleteForensicCaseById(long id) {
46         this.caseRepository.deleteById(id);
47     }
48
49     @Override
50     public Page<ForensicCase> findPaginated(int pageNo, int pageSize, String sortField, String sortDirection) {
51         Sort sort = sortDirection.equalsIgnoreCase(Sort.Direction.ASC.name()) ? Sort.by(sortField).ascending() :
52             Sort.by(sortField).descending();
53
54         Pageable pageable = PageRequest.of(pageNo - 1, pageSize, sort);
55         return this.caseRepository.findAll(pageable);
56     }
57
58 }
```

ForensicCaseController.java



```
1 package net.javaguides.springboot.controller;
2 import net.javaguides.springboot.service.ForensicCaseService;
3
4 @Controller
5 @RequestMapping("/forensicCase")
6 public class ForensicCaseController {
7     @Autowired
8     private ForensicCaseService forensicCaseService;
9
10    // Home method
11    @GetMapping("/home2")
12    public String dashboard(Model model, Principal principal) {
13        return "home2";
14    }
15
16    // add form controller
17    @GetMapping("/open-case")
18    public String openCaseForm(Model model) {
19        model.addAttribute("listCases", forensicCaseService.getAllCases());
20        return "open_case_form";
21    }
22
23    // add case
24    @PostMapping("/new-case")
25    public String addCase(Model model) {
26        ForensicCase forensicCase = new ForensicCase();
27        model.addAttribute("forensicCase", forensicCase);
28        return "new_case";
29    }
30
31    // save forensic case
32    @PostMapping("/saveforensicCase")
33    public String saveForensicCase(@ModelAttribute("forensicCase") ForensicCase forensicCase) {
34        // save employee to database
35        forensicCaseService.saveForensicCase(forensicCase);
36        return "redirect:/home2";
37    }
38
39    // show form for update
40    @GetMapping("/showFormForUpdate/{id}")
41    public String showFormForUpdate(@PathVariable ( value = "id" ) long id, Model model) {
42        // get employee from the service
43        ForensicCase forensicCase = forensicCaseService.getForensicCaseById(id);
44
45        // set employee as a model attribute to pre-populate the form
46        model.addAttribute("forensicCase", forensicCase);
47        return "update_case";
48    }
49
50    // delete forensic case
51    @GetMapping("/deleteForensicCase/{id}")
52    public String deleteForensicCase(@PathVariable ( value = "id" ) long id) {
53        // call delete employee method
54        this.forensicCaseService.deleteForensicCaseById(id);
55        return "redirect:/home2";
56    }
57
58    // display list of employees
59}
```

