VIT®

UNIVERSITY

(Estd. u/s 3 of UGC Act 1956)

ITE1007- Object oriented analysis and design

Title: Near Field Communication

Name:Garlapati Saiteja

Reg. no: 17BIT0217

Signature:

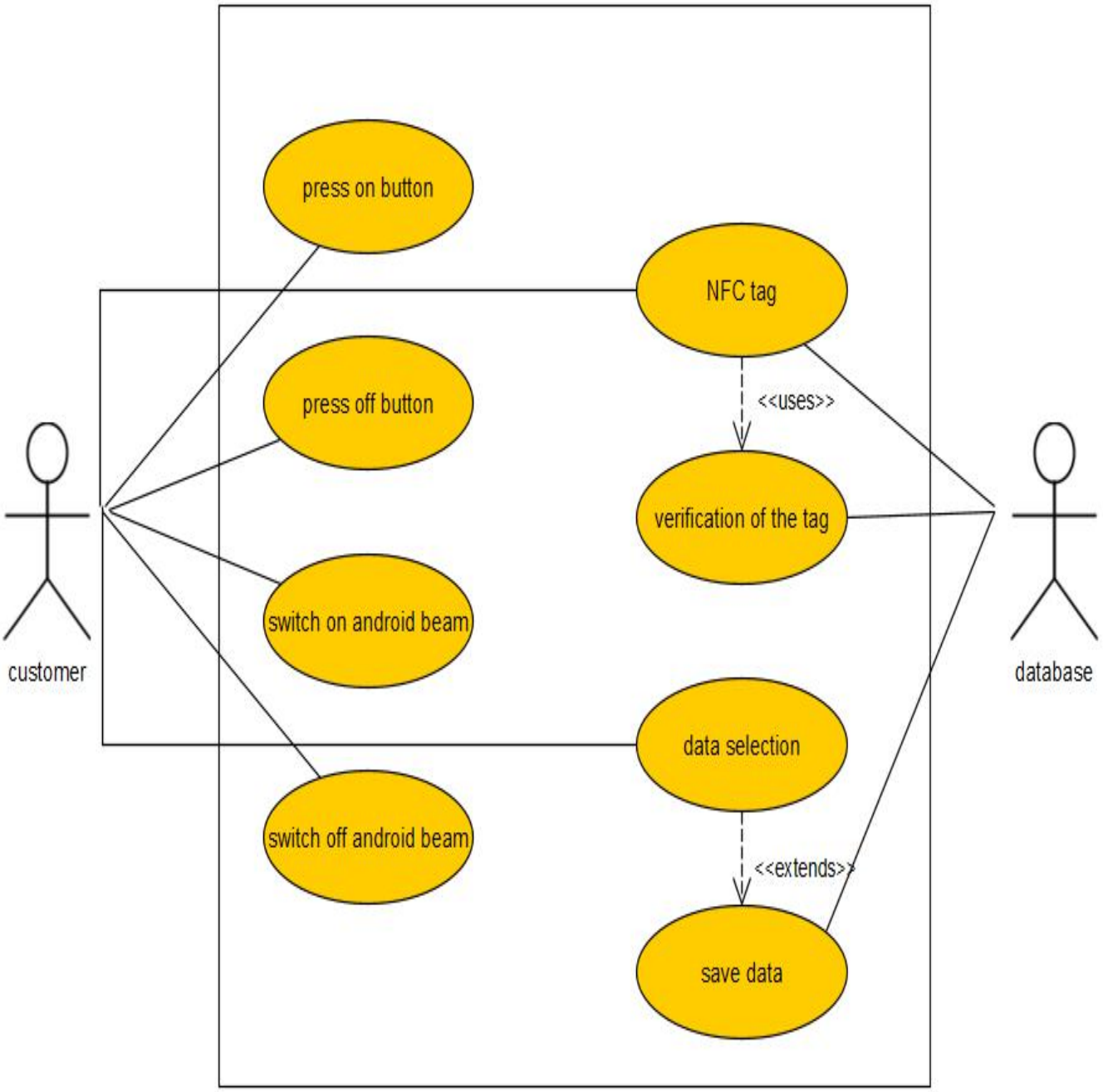
Problem Definition

Near Field Communication (NFC) is a special category or a case of Radio Frequency Identification Technology. It is a wireless technology which provides communication between two mobile phones which contain a NFC tag, using short range radio waves. Like bluetooth it works just in short range and information exchange happens at low speed. The two gadgets can speak with one another utilizing NFC innovation when they contact one another or conveyed near one another. NFC also has an advantage over bluetooth that it provides bi-directional communication between devices, i.e., both devices can send and receive data simultaneously. Several security issues are as well attached with NFC which is a big concern. Security attacks like eavesdropping, data corruption and modification, interference attacks and theft, are the most dangerous for the customer who is using their smart phones for payment purpose.

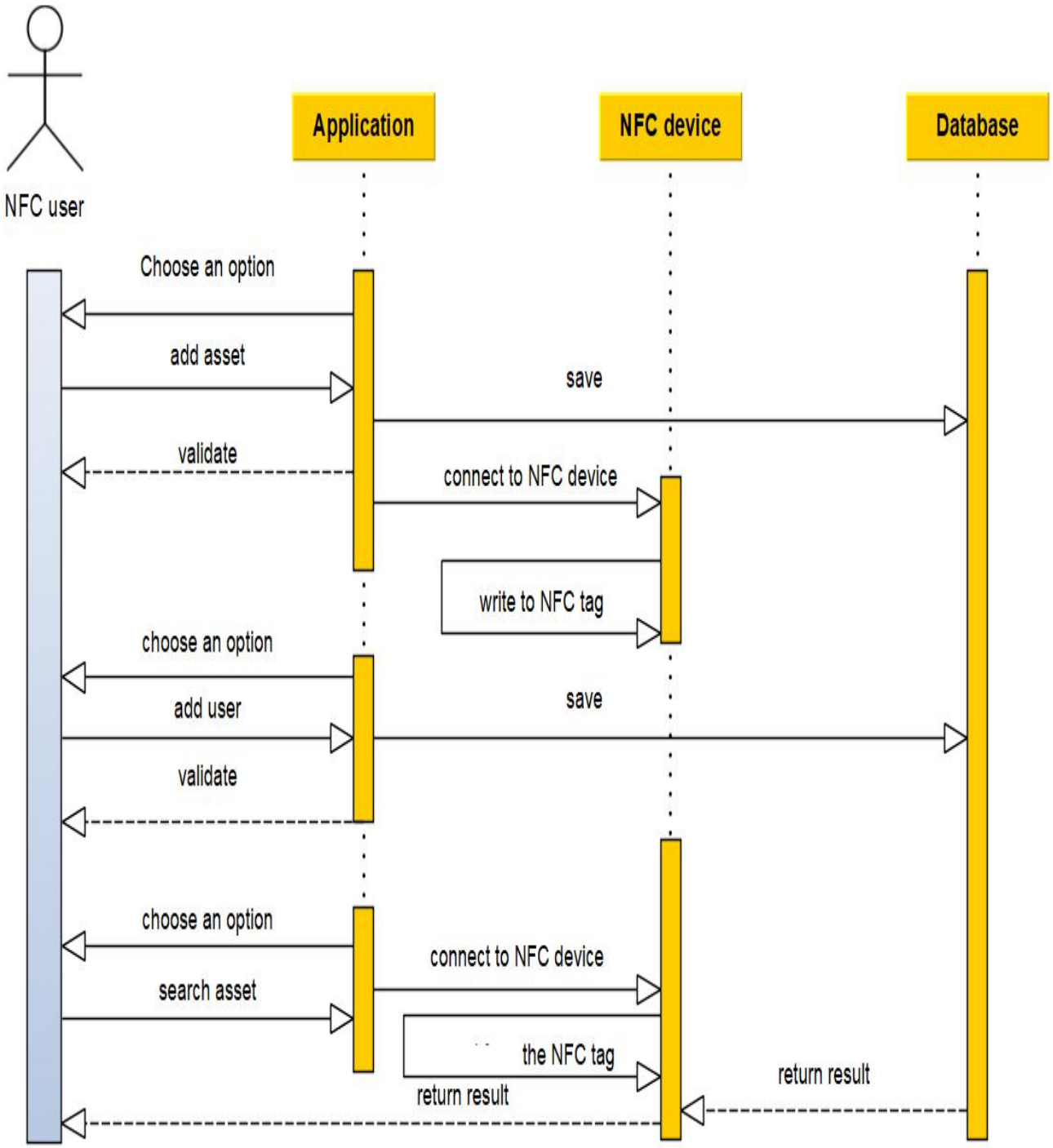
Software design tool used for the project

- yEd graph editor

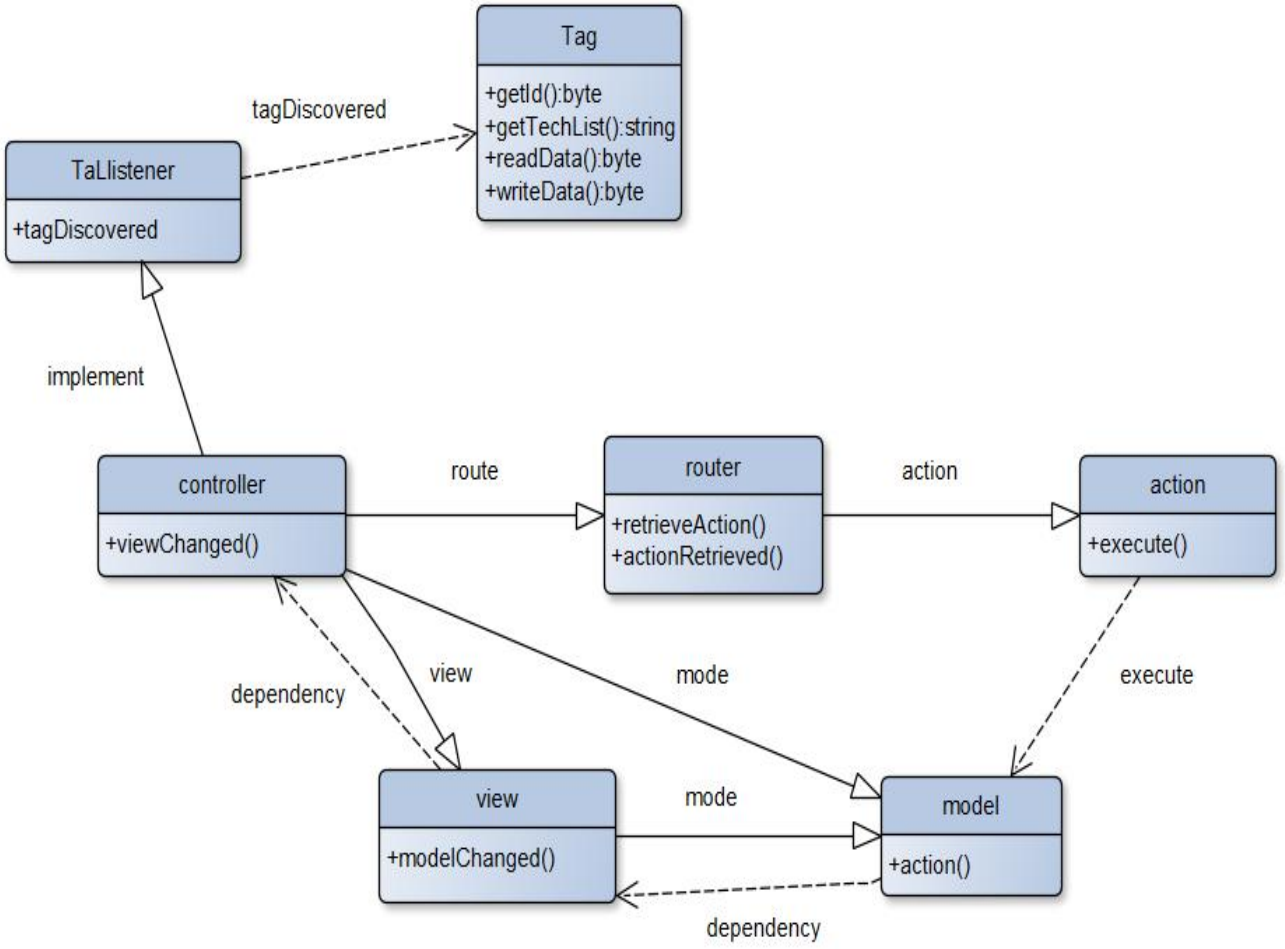
USE CASE DIAGRAM



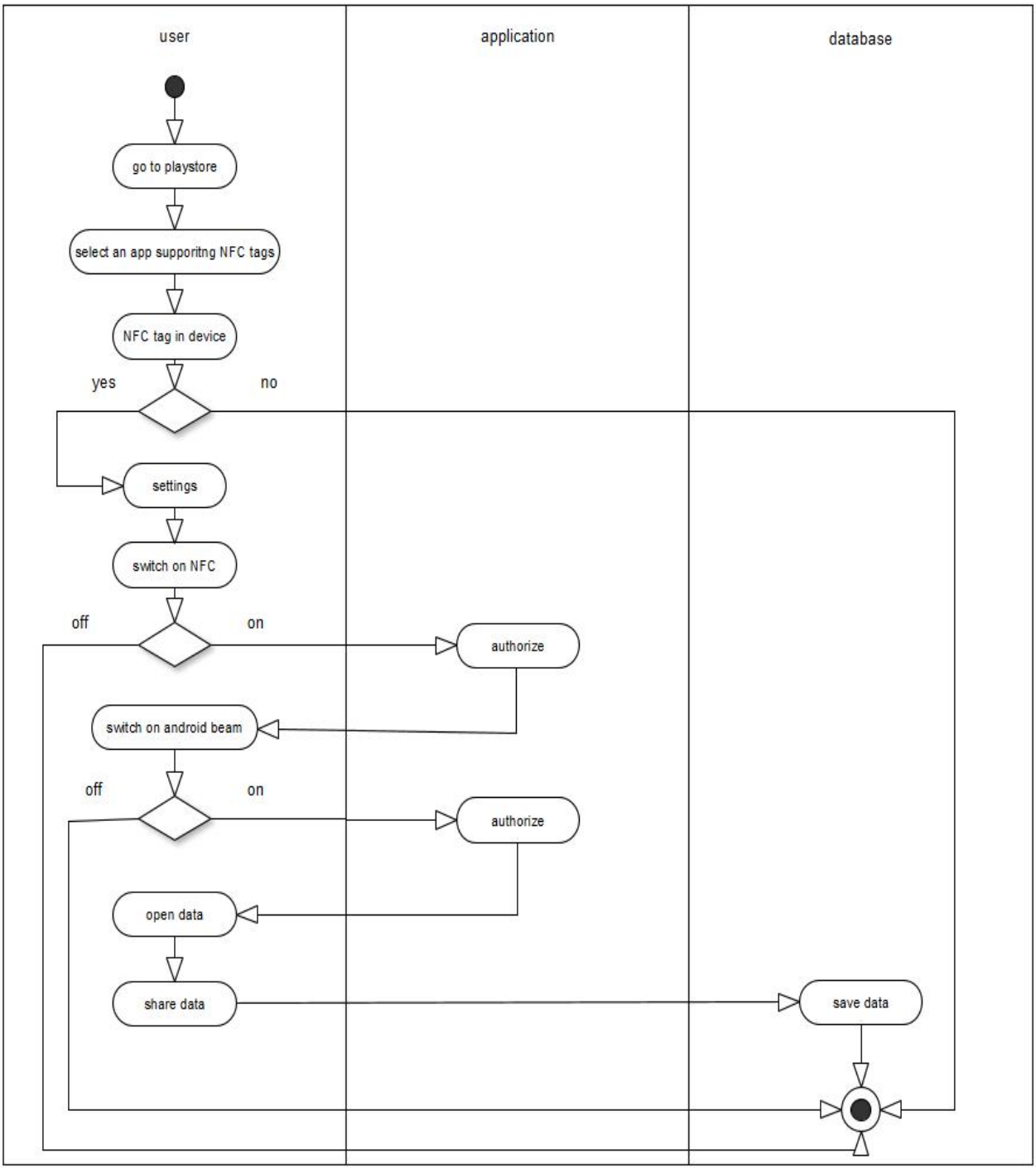
SEQUENCE DIAGRAM



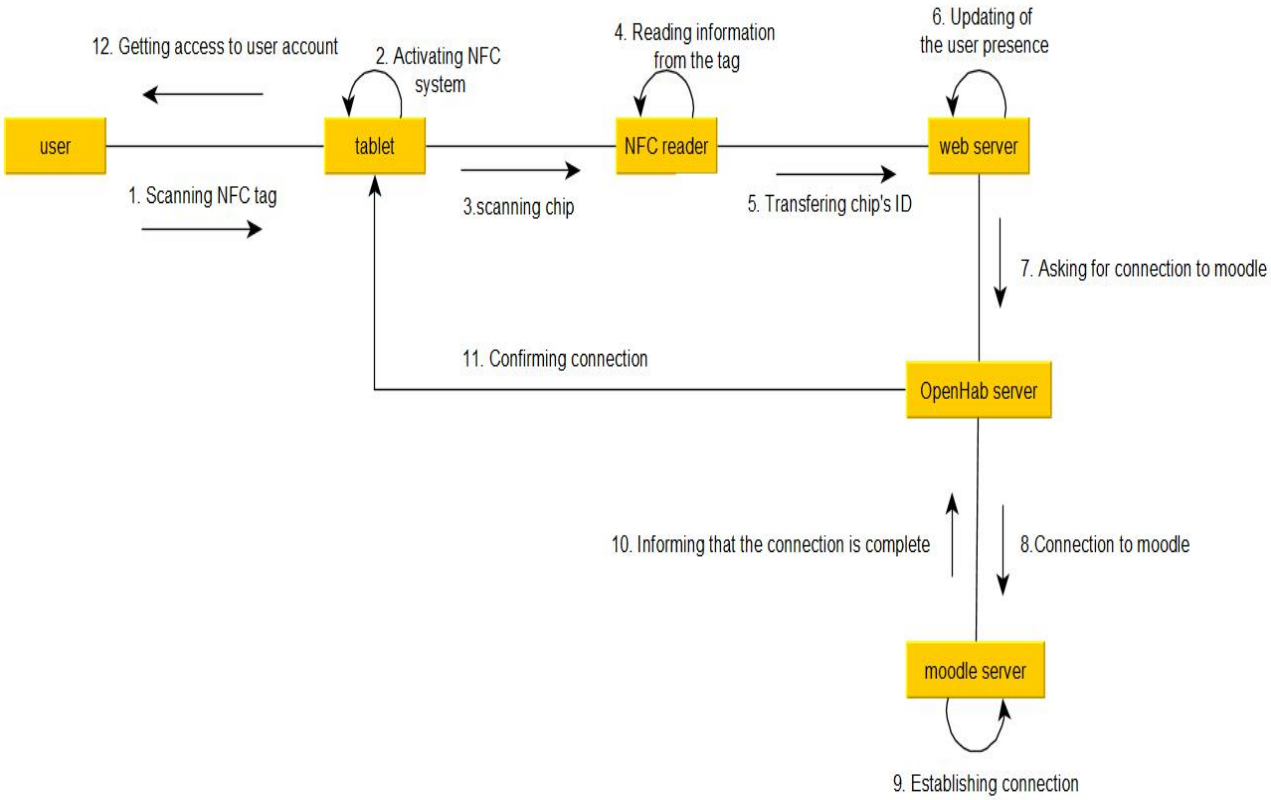
CLASS DIAGRAM



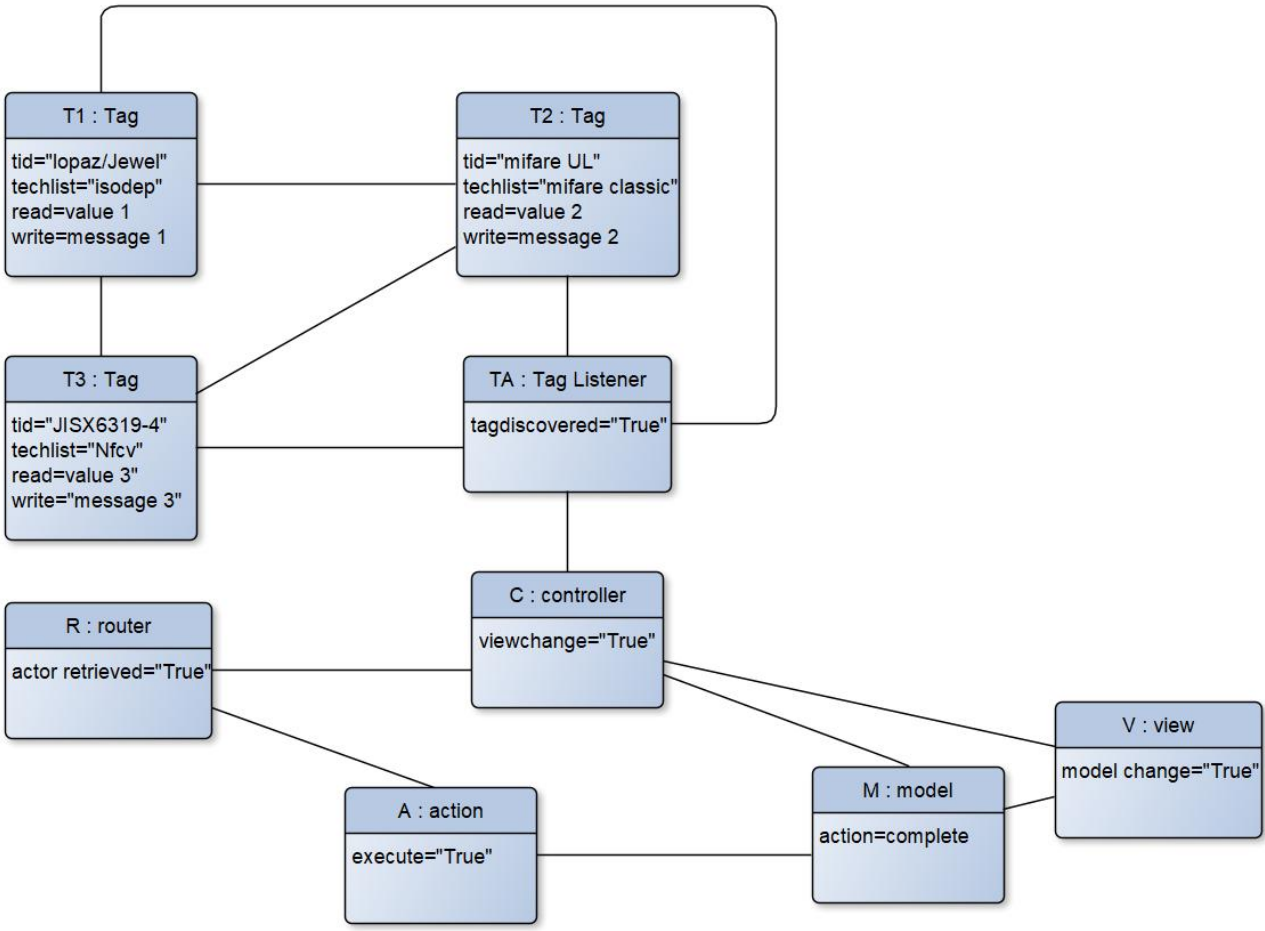
ACTIVITY DIAGRAM



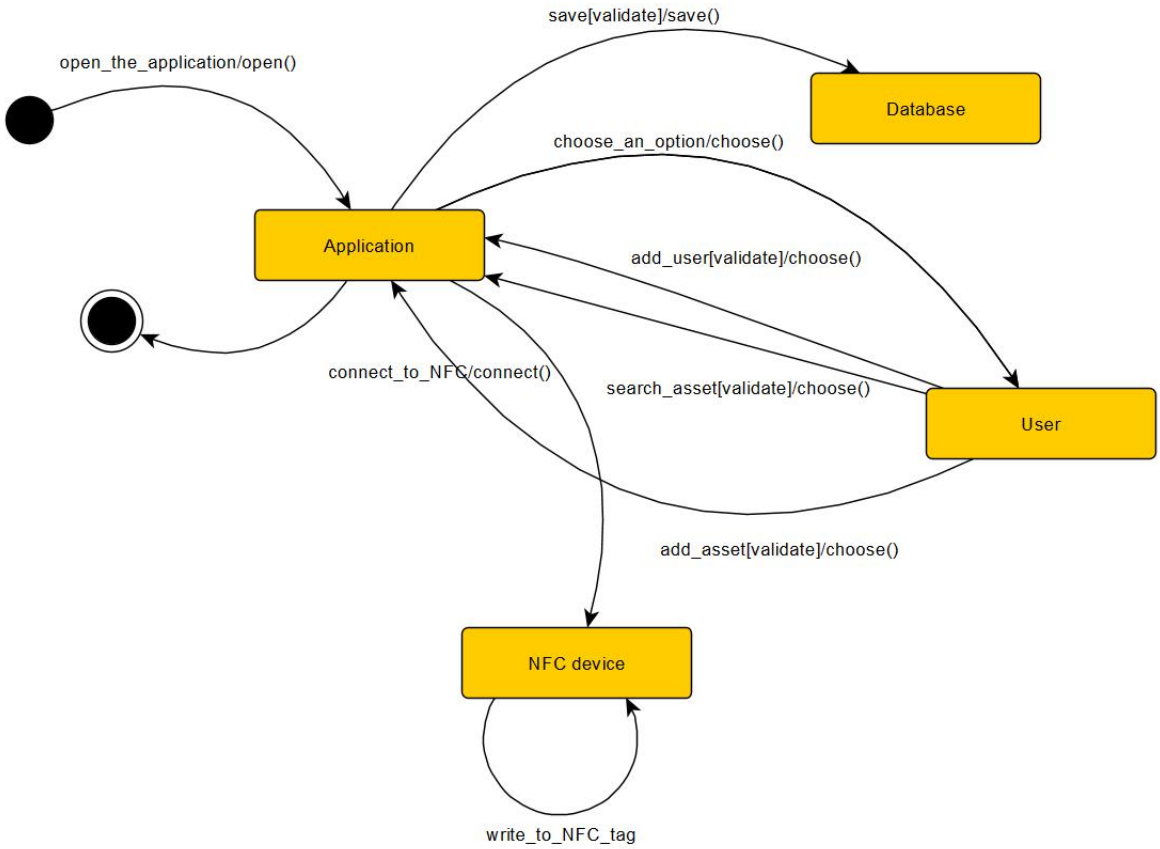
Collaboration Diagram



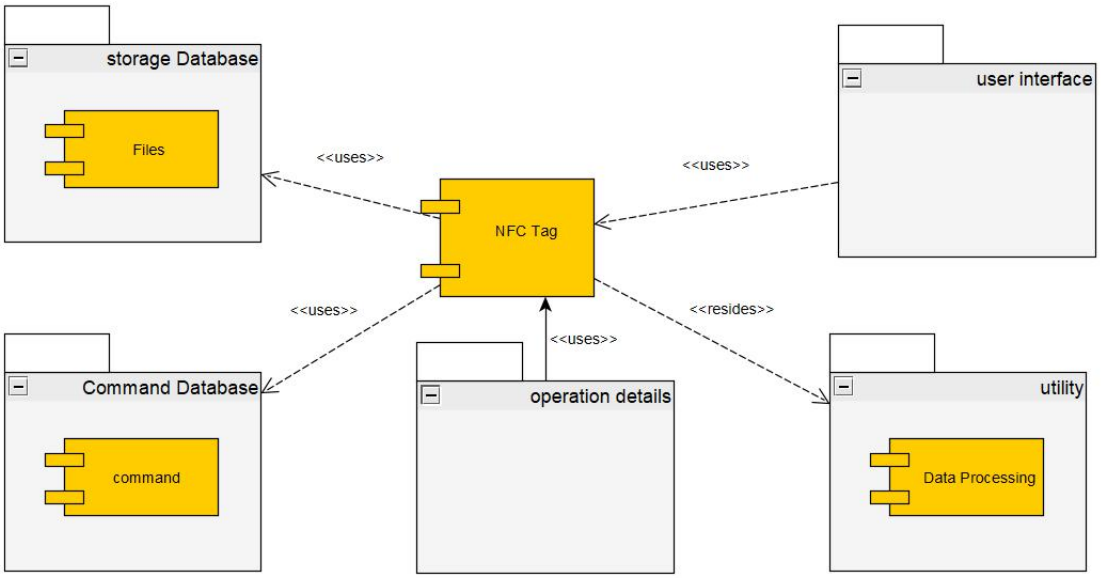
Object Diagram



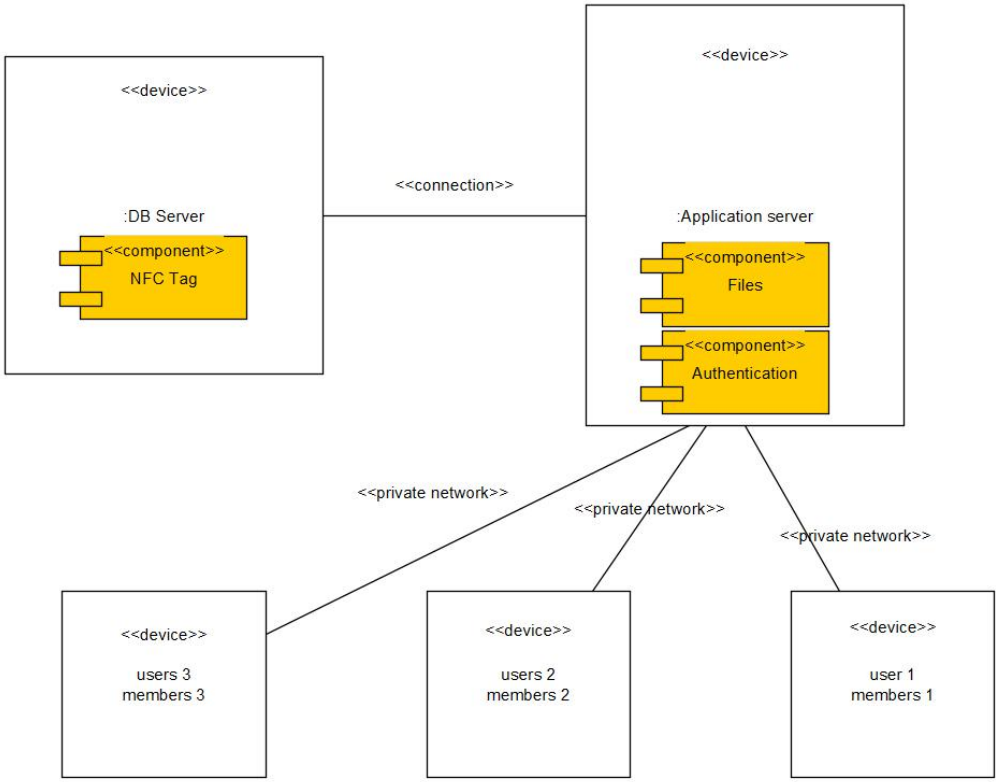
State transition diagram



Component diagram



Deployment diagram



Code:

Action:

```
import java.util.*;
/**
 *
 */public class action {

    /**
     * Default constructor
     */
    public action() {
    }

    /**
     *
     */
    public void execute() {
        // TODO implement here    }

}
```

Router:

```
import java.util.*;
/**
 *
 */public class router extends action {

    /**
     * Default constructor
     */
    public router() {
    }

    /**
     *
     */
    public void retrieveAction() {
        // TODO implement here    }

    /**
```

```
     *
     */
    public void actionRetrieve() {
        // TODO implement here    }

}
```

Tag:

```
import java.util.*;
/**
 *
 */public class Tag {

    /**
     * Default constructor
     */
    public Tag() {
    }

    /**
     * @return
     */
    public byte getId() {
        // TODO implement here        return 0;
    }

    /**
     * @return
     */
    public String getTechlist() {
        // TODO implement here        return "";
    }

    /**
     * @return
     */
    public byte readData() {
        // TODO implement here        return 0;
    }

    /**
     * @return
```

```
     */
    public byte writeData() {
        // TODO implement here        return 0;
    }

}
```

Controller:

```
import java.util.*;
/**
 *
 */public class Controller extends TagListener {

    /**
     * Default constructor
     */
    public Controller() {
    }

    /**
     *
     */
    public void viewchanged() {
        // TODO implement here    }

}
```

Model:

```
import java.util.*;
/**
 *
 */public class model extends Controller {

    /**
     * Default constructor
     */
    public model() {
    }

    /**
```

```
     *
     */
    public void action() {
        // TODO implement here    }

}
```

Tag Listener:

```
import java.util.*;
/**
 *
 */public class TagListener {

    /**
     * Default constructor
     */
    public TagListener() {
    }

    /**
     *
     */
    public void tagDiscovered() {
        // TODO implement here    }

}
```

View:

```
import java.util.*;
/**
 *
 */public class view extends model {

    /**
     * Default constructor
     */
    public view() {
    }

    /**
```

```
     *
     */
    public void modelChanged() {
        // TODO implement here    }

}
```