

A Project Report on  
**Single Sign On**

Submitted in partial fulfillment of the requirements for the award  
of the degree of

**Bachelor of Engineering**

in

**Information Technology**

by

**Nirmit Dagli(16104013)**  
**Mihir Deorukhkar(16104065)**  
**Sarvesh Sawant(17204013)**

Under the Guidance of

**Ms. Kaushiki Upadhyaya**  
**Ms. Nahid Shaikh**



**Department of Branch Name**

A.P. Shah Institute of Technology  
G.B.Road,Kasarvadavli, Thane(W), Mumbai-400615  
UNIVERSITY OF MUMBAI

**Academic Year 2019-2020**

## Approval Sheet

This Project Report entitled “*Single Sign On*” Submitted by “*Nirmit Dagli*” (16104013), “*Mihir Deorukhkar*” 16104065), “*Sarvesh Sawant*” (17204013) is approved for the partial fulfillment of the requirement for the award of the degree of *Bachelor of Engineering* in *Branch Name* from *University of Mumbai*.

(Ms. Nahid Shaikh)  
Co-Guide

(Ms. Kaushiki Upadhyaya)  
Guide

Mr. Kiran Deshpande  
Head Department of Information Technology

Place: A.P. Shah Institute of Technology, Thane  
Date:

## CERTIFICATE

This is to certify that the project entitled “**Single Sign On**” submitted by “**Nirmit Dagli**” (16104013), “**Mihir Deorukhkar**” 16104065), “**Sarvesh Sawant**” (17204013) for the partial fulfillment of the requirement for award of a degree **Bachelor of Engineering** in **Information Technology**, to the University of Mumbai, is a bonafide work carried out during academic year 2017-2018.

Ms. Nahid Shaikh  
Co-Guide

Ms. Kaushiki Upadhyay  
Guide

Mr. Kiran Deshpande  
Head Department of Information Technology

Dr. Uttam D.Kolekar  
Principal

External Examiner(s)

1.

2.

Place: A.P.Shah Institute of Technology, Thane

Date:

## Declaration

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, We have adequately cited and referenced the original sources. We also declare that We have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

---

(Nirmit Dagli, 16104013)

---

(Mihir Deorukhkar, 16104065)

---

(Sarvesh Sawant, 17204013)

Date:

## **Abstract**

Consider a college education portal needs to provide different courses and tutorials to its students. But to incorporate numerous resources and tutorials onto one education portal can be tedious and space constraint. Having multiple systems typically require multiple sign-on dialogues to access the resources. Users need to register on multiple portals to access the contents and courses and it involves the headache of remembering multiple sets of credentials. Users also have to present credentials multiple times they login to these portals/websites. With these scenarios, more the portals, the more sign-ins are required. It also requires to restrict access to unauthorized users when log-ins are authenticated. If there are redundancy of resources and inconsistent information across multiple website across the systems, users may show lack of interest. Single sign on system is the proposed method to provide access to the educational learning resources/contents. In this approach only one set of credential is required, user can access the multiple services with those same credentials once integrated into all systems. This approach provides a secure way to authenticate users and give access to all services

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Background . . . . .	1
1.2	Objectives . . . . .	2
1.3	Purpose, Scope and Applicability . . . . .	3
1.3.1	Purpose . . . . .	3
1.3.2	Scope . . . . .	3
1.3.3	Availability . . . . .	3
<b>2</b>	<b>Literature Review</b>	<b>4</b>
<b>3</b>	<b>Design</b>	<b>7</b>
3.1	Conceptual Models . . . . .	7
3.1.1	Activity Diagram . . . . .	7
3.1.2	Use Case Diagram . . . . .	8
3.1.3	Class Diagram . . . . .	9
3.2	System Architecture . . . . .	10
3.3	User Interface Design . . . . .	11
<b>4</b>	<b>Implementation</b>	<b>12</b>
4.1	Working . . . . .	12
4.2	Code Snippets . . . . .	14
<b>5</b>	<b>Testing</b>	<b>17</b>
5.1	Testing Approach . . . . .	17
5.1.1	Unit Testing . . . . .	17
5.1.2	Integrated Testing . . . . .	17
5.1.3	System Testing . . . . .	17
5.2	Test Cases . . . . .	18
<b>6</b>	<b>Conclusions and Future Scope</b>	<b>19</b>
6.1	Conclusion . . . . .	19
6.2	Future Scope . . . . .	19
	<b>Bibliography</b>	<b>20</b>
	<b>Publication</b>	<b>22</b>

# List of Figures

2.1	Project Timeline Chart . . . . .	6
3.1	Activity Diagram . . . . .	7
3.2	Use Case Diagram . . . . .	8
3.3	Use Case Diagram . . . . .	9
3.4	System Architecture . . . . .	10
3.5	User Interface Design . . . . .	11
4.1	Login Page . . . . .	12
4.2	Registration Page . . . . .	13
4.3	Home Page . . . . .	13

# List of Tables

5.1	Test cases . . . . .	18
-----	----------------------	----



# Chapter 1

## Introduction

Our single sign-on implementation mechanism is by using LDAP to serve as a personnel database, who personnel information between various systems. LDAP is short for Lightweight Directory Access Protocol. LDAP server is used to store and retrieve information, which is similar to ordinary relational database. The main differences between LDAP servers and the general relational database are as follows: LDAP using tree model rather than rational model to organize information; Mainly provides data query services, the query speed faster than the ordinary relational database; Excellent ability to copy the information makes it highly robust. LDAP tree information organization model is similar to the actual hierarchical relationships between the various departments of an organization. So, using LDAP to store personnel information made management easier.

Single sign-on (SSO) is a session and user authentication service that permits a user to use one set of login credentials (e.g., name and password) to access multiple applications. The service authenticates the end user for all the applications the user has been given rights to and eliminates further prompts when the user switches applications during the same session.

On the back end, SSO is helpful for logging user activities as well as monitoring user accounts. Single sign-on reduces human error, a major component of systems failure and is therefore highly desirable but difficult to implement. SSO avoids the monotonous task of confirming identity over and over again through passwords or other authentication systems

### 1.1 Background

Single sign-on (SSO) is an authentication process that allows a user to access multiple applications with one set of login credentials. SSO is a common procedure in enterprises, where a client accesses multiple resources connected to a local area network (LAN).

SSO advantages include:

- Eliminates credential reauthentication and help desk requests; thus, improving productivity

- Streamlines local and remote application and desktop workflow. Minimizes phishing
- Improves compliance through a centralized database. Provides detailed user access reporting
- Improves compliance through a centralized database. Provides detailed user access reporting.
- Improves compliance through a centralized database.
- With SSO, a user logs in once and gains access to different applications, without the need to re-enter log-in. credentials at each application. SSO authentication facilitates seamless network resource usage.
- SSO is not suited for systems requiring guaranteed access, as the loss of log-in credentials results into denial of access to all systems.
- Ideally, SSO is used with other authentication techniques, such as smart cards and one-time password

## 1.2 Objectives

To provide a service for accessing multiple platform using single credential.

By using LDAP, a single central database will maintain information of multiple accounts at the same time.

Making Authorization process more Secure.

Reducing the database chunk with one central database.

To maintain security and monitor the user activity(log)

## **1.3 Purpose, Scope and Applicability**

### **1.3.1 Purpose**

The aim of the proposed system is to develop a system with improved facilities. The proposed system can overcome all the limitation of the existing system, such as user's need to keep information about the different passwords for different web-application, it gives more security to data, ensures data privacy.

### **1.3.2 Scope**

The project has a big scope to do. Web-Application Developers and the one hosting the Web application do not need to worry about Login Information of the user also they do not need to maintain a database because it would all be managed by a centralized database used for single sign on system and also users get benefits through this project as their data is always stored in encrypted format.

### **1.3.3 Availability**

The system can be used for every kind of website.

# Chapter 2

## Literature Review

The papers referred are mentioned below:

**[1] Application of single sign-on (sso) in Digital campus Jian Hu, Qizhi Sun, Hongpin Chen College of Information Engineering, North China University of Technology, Beijing, China (IC-BNMT2010)**

The growing number of systems brings convenience for users, but also exposed a wide range of issues: Between the isolated systems, there is information redundancy and information inconsistent, so it's difficult to maintain; Too many log in points, each system has its own authentication mechanism, If a user wants to access some systems, he/she must log on several times, which brings inconvenience to the user. Users must to remember many usernames and passwords, it could result in password fatigue and lead to password disclosure, so it has security risks. Single sign-on is a good solution to these problems. The so-called single sign is that after a user logs on a system can logs on other systems which integrated into a single sign system without reauthentication. Our single sign-on implementation mechanisms: Using LDAP to establish a unified personnel database, so that personnel information between various systems was unified and information redundancy were reduced. At the same time it will provide authentication information for single sign-on. Establish single sign-on authentication service center and integrate all existing applications with the center.

**2] SSO-Key Distribution Centre Based Implementation Using Serpent Encryption Algorithm for Distributed Network Ms Durga Prasanna, Ms Roopa S - 2015 IEEE International Advance Computing Conference (IACC)**

The several single sign-on schemes have been proposed. However, most of them have security flaws, and even worse, their improvements are also insecure against possible attacks. Thus, this paper aims to give an approach into the most recent SSO schemes, identifying their flaws, issues and challenges. The second aim of this paper is to formalize the Single Sign-On (SSO) and its security model to formally resolve the issues identified. Also, an efficient and provably secure single sign-on authentication scheme without the identified drawbacks will be provided according to the formal model. It provides efficient and secure identification services with further security requirements for users in distributed systems and networks. In general, the identification services may require three factors, i.e., password, symmetric key and signature's characteristics. The authentication which is based on password is called

password-based authentication. Password-based authentication together with another factor, symmetric key, is called twofactor authentication. In which, a successful user authentication can be achieved if the user has a correct password together with a corresponding signature. The two-factor authentication consists all of these three factors, i.e., password, symmetric key and signature characteristics.

**[3] A New Identity Authentication Scheme of Single Sign On for Multi-Database. Lan Zhang, Hongyun Ning, Yunyun Du, Yan-xia Cui**

Multi-database system is a complete global logical database which is composed by multiple database servers, and it can achieve data sharing and transparent accessing. In the multidatabase system, the computer architecture, operating system, DBMS and so on, are heterogeneous, and each part has own authentication mode. The earliest theoretical research on the multi-database environment can be traced back to the seventy's in 21 st Century. Institutions in foreign countries mainly are Almaden research center database group of the United States IBM company, Stanford University, TONA, etc. Foreign major database vendors have launched commercial products which support multi-database environment according to the these forming theoretical system, including DB2, SyBase, etc [ 1 , 2 , 3] . 4Technology developed the Paronama system based on the COBAR, etc. In order to improve the limitations of the traditional multidatabase authentication model, we put forward a new identity authentication scheme of Single Sign On for multi-database, as they respectively describe the new authentication model and the authentication process. In the new scheme, we first introduce the concept of a multi-database coalition domain.

[illegible]

6

# Chapter 3

## Design

### 3.1 Conceptual Models

A conceptual model is a representation of a system, made of the composition of concepts which are used to help people know, understand, or simulate a subject the model represents. It is also a set of concepts.

#### 3.1.1 Activity Diagram

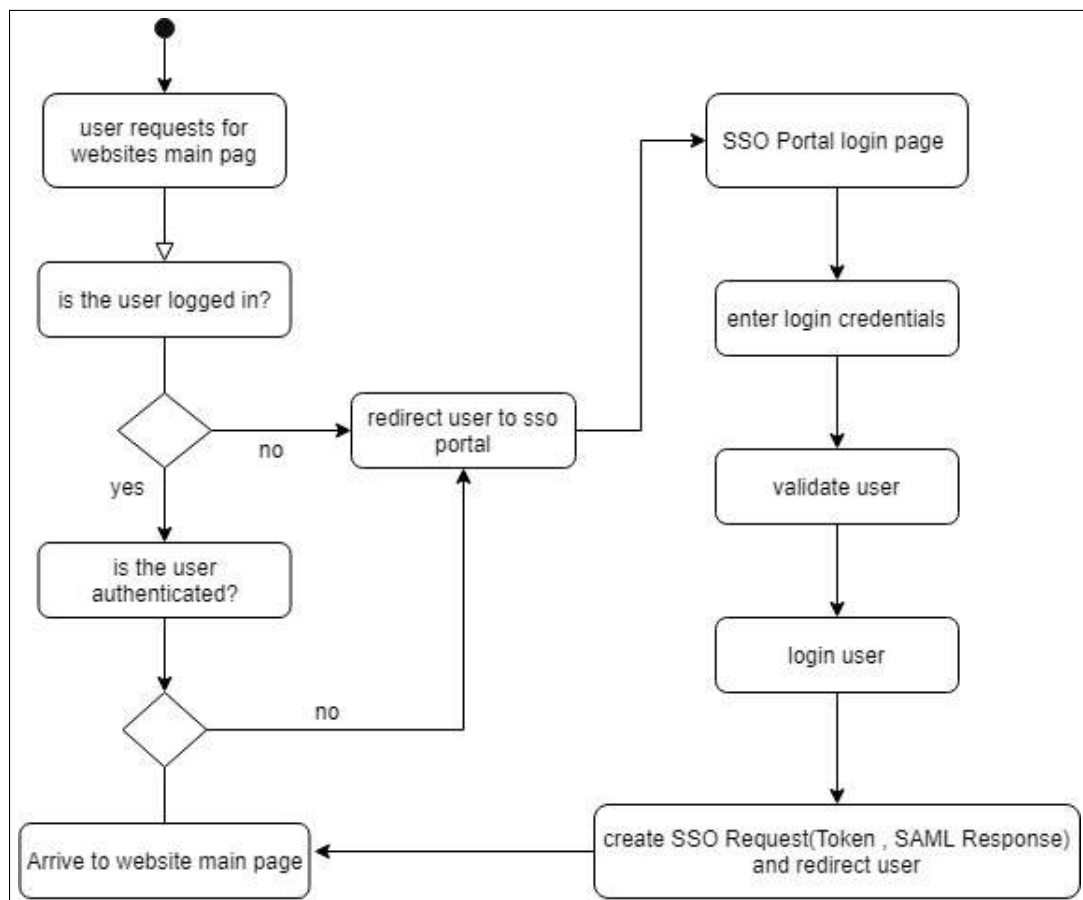


Figure 3.1: Activity Diagram

Activity diagram is another important diagram in UML to describe the dynamic aspect of the system. Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The control flow is drawn from one operation to another. Activity diagrams deal with all type of flow control by using different elements such as fork, join, etc. Activity is a particular operation of the system. Activity diagrams are not only used for visualizing the dynamic nature of a system, but they are also used to construct the executable system by using forward and reverse engineering techniques.

Activity diagram begins with user's request for a webpage. If the user is logged in then the user will be authenticated. If the user is not logged in the user will be redirected to the SSO login Portal. The user needs to enter the SSO credentials. Then the user will be validated and a token will be generated. This token will be sent to the request website and the user will be able to access the requested website.

### 3.1.2 Use Case Diagram

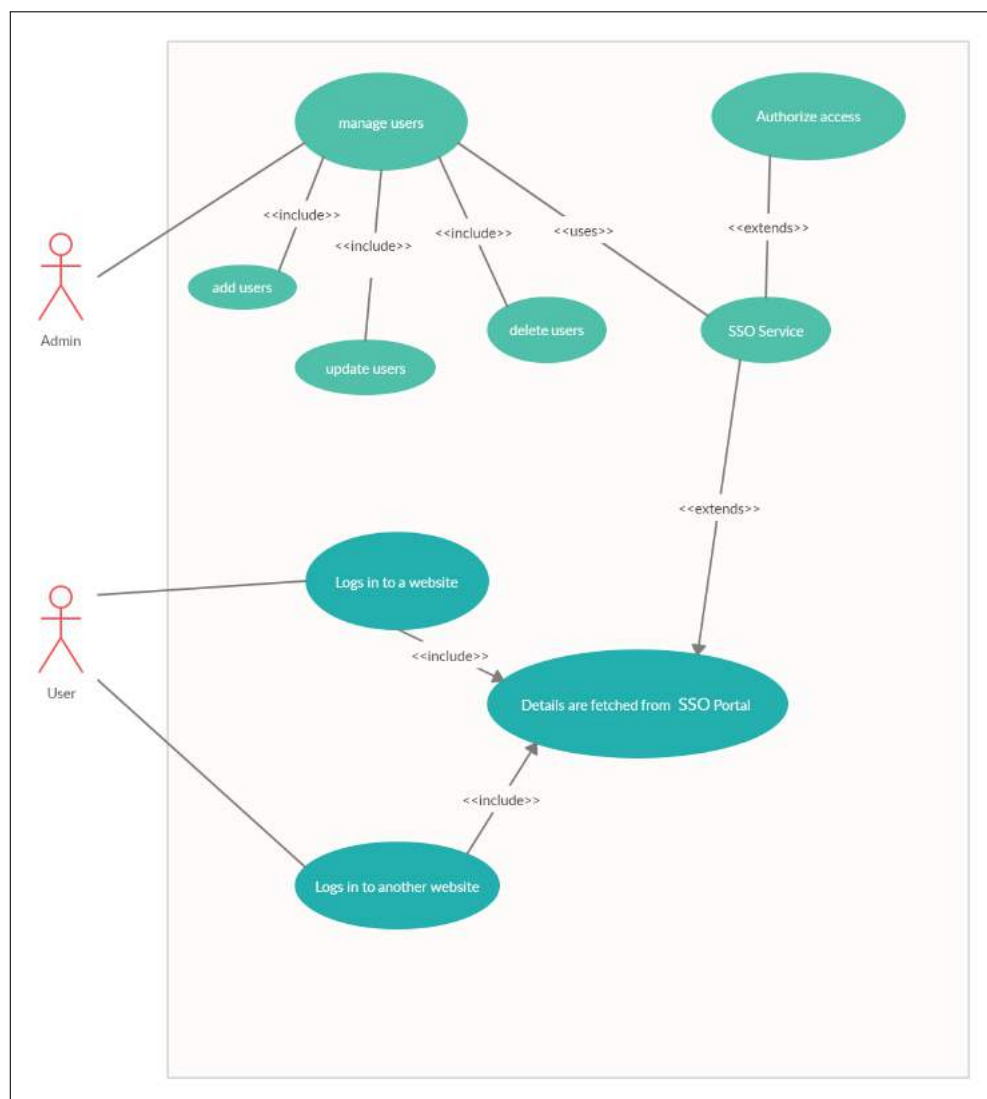


Figure 3.2: Use Case Diagram



A use case diagram at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved. A use case diagram can identify the different types of users of the system and the different use cases and will often be accompanied by other types of diagrams as well. The use cases are represented by either circles or ellipses. Use case diagrams are used to gather the requirements of a system including internal and external influences. These requirements are mostly design requirements. Hence, when a system is analyzed to gather its functionalities, use cases are prepared and actors are identified.

Our use case consists of two major entities that interact with the major modules of the system. The first one is the user who can sign up/login, login to multiple Website. The second is the administrator who has an access to access the backend database enter and manage records, handle authentication process. Also maintain various CRUD operation.

### 3.1.3 Class Diagram

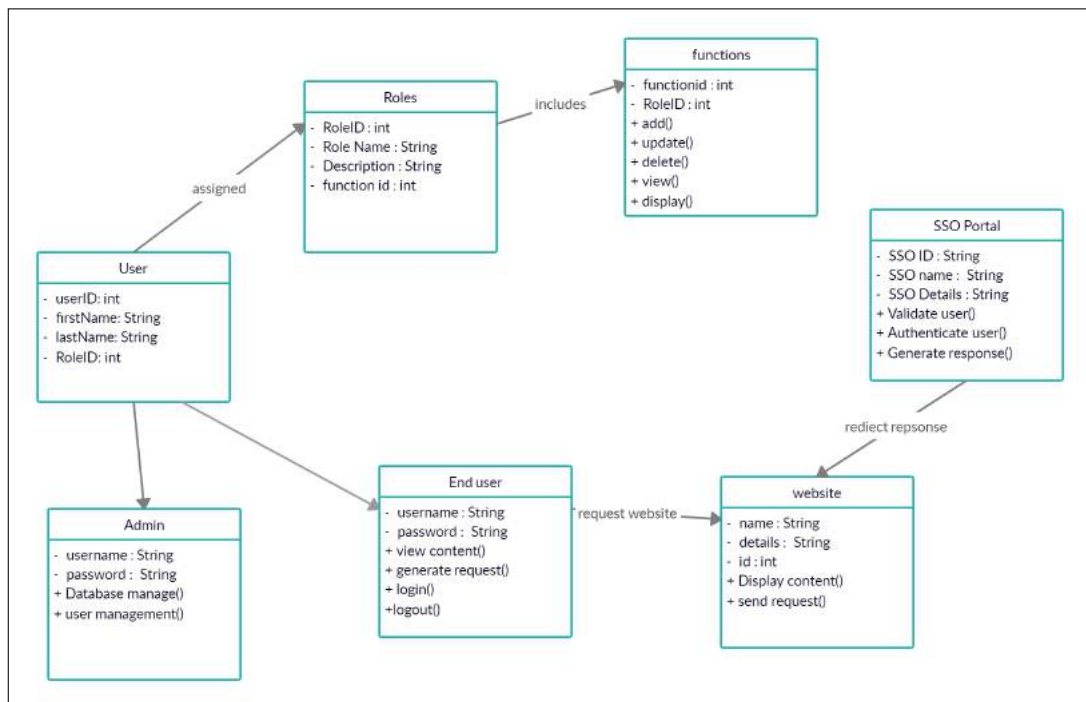


Figure 3.3: Use Case Diagram

Class diagram helps in understanding various different modules that are functioning such as user, admin, end user, SSO Portal, Website, etc. Class diagram not only provides a visual overview of the project; instead, it helps in understanding various operations (such as add(), delete() function, generate content() function, generate request() function, etc.). It also helps in understanding various attributes (such as Username, Password, SSO id, SSO name, function id, RoleID, etc.) involved in the project.

The class diagram also displays various modules of our system interacting with each other. The system is divided into two users: admin and end user. All users are classified into

different roles and each role is associated with different function. End user requests for the website and this request is sent to the SSO portal. The SSO portal validates the user and redirect the response to the website.

## 3.2 System Architecture

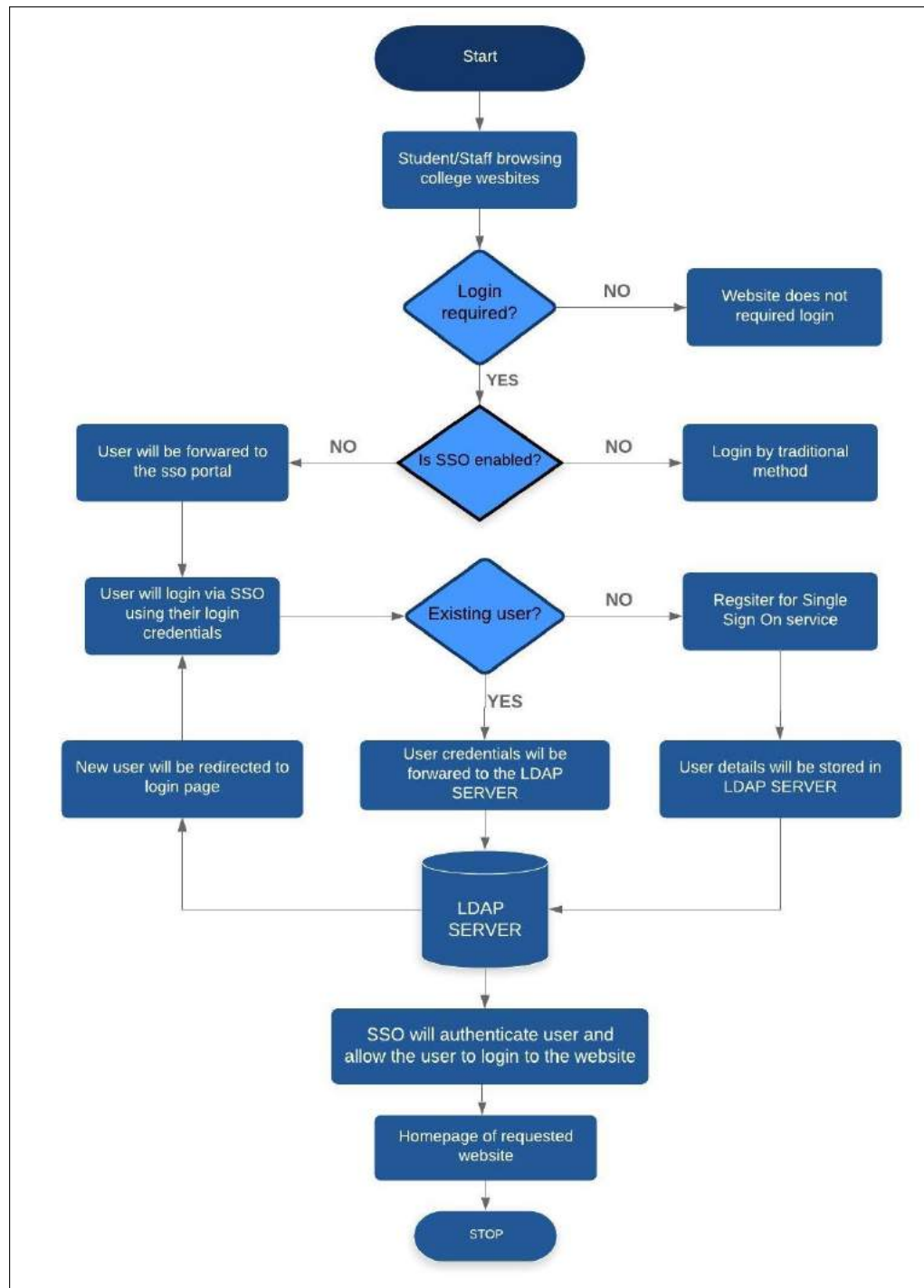


Figure 3.4: System Architecture

### 3.3 User Interface Design

User interface design or UI design generally refers to the visual layout of the elements that a user might interact with in a website, or technological product. This could be the control buttons of a radio, or the visual layout of a webpage. User interface designs must not only be attractive to potential users, but must also be functional and created with users in mind. We have used the following design for our user interface.



Figure 3.5: User Interface Design

# Chapter 4

## Implementation

### 4.1 Working

1) User will login to the school website/portal if he/she has an account. If user doesn't have an account he must sign in through the Registration page.

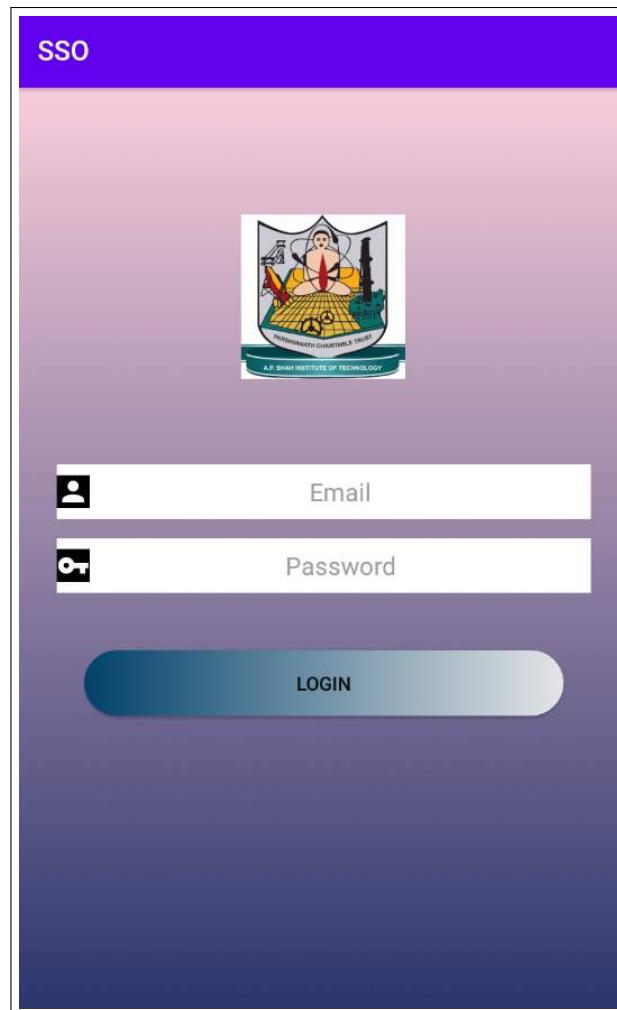
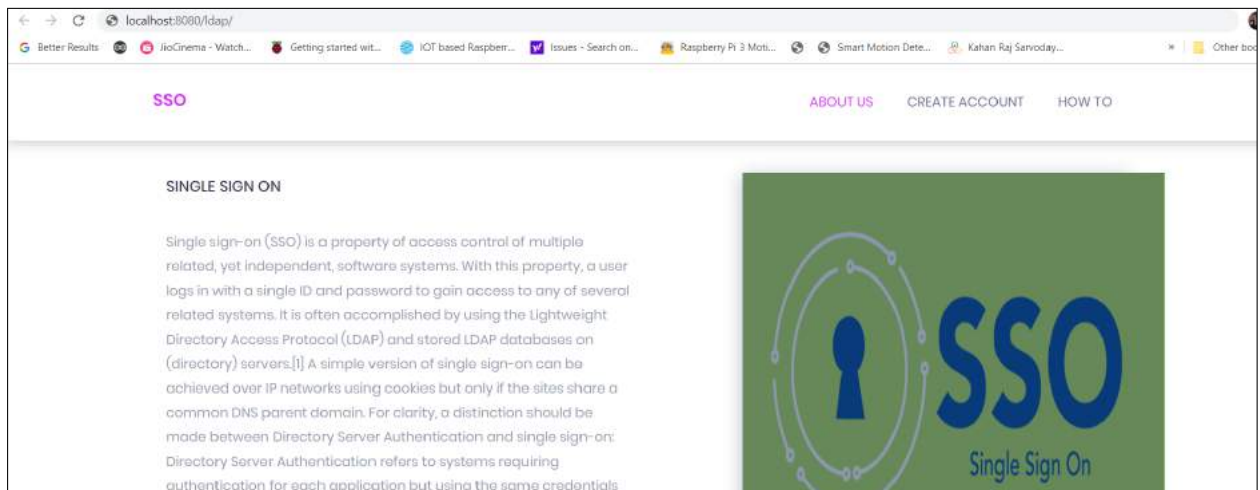
The image shows a mobile application login screen. At the top, there is a purple header bar with the text "SSO" in white. Below the header, the background is a gradient of purple and blue. In the center, there is a school crest featuring a book, a lamp, and a gear, with the text "A.P. BHAI INSTITUTE OF TECHNOLOGY" below it. Below the crest, there are two input fields: the first is labeled "Email" with a person icon on the left, and the second is labeled "Password" with a key icon on the left. Below these fields is a large, rounded, blue button with the text "LOGIN" in white.

Figure 4.1: Login Page

**SIGN UP**

**Figure 4.2: Registration Page**

2))Once the users login,the SSO verifies the credentials and then it setting up a cookie on the browser storing the username (that could be coded with a private key). he/she will be forwarded to the website.On website there will be Affiliated/Education Portals/website links.



**Figure 4.3: Home Page**

3) When user click some other institute link the SSO system will accept the parameter and validate it, It searches the cookie and reads the username on the value (using the key for decode the string).

4) If validation success then SSO will redirect user to the specific website's dashboard without login. SSO system will generate a hash signature and will pass to the server to validate before sending them directly to dashboard.

5) If authentication fails, SSO will redirect user back to the website.

6) User have to provide login credentials that he created on college portal and the SSO system will authenticate it from the database and provide access to user of the website.

## 4.2 Code Snippets

### 1. Login Form :

```
try
{
    URL urlForGetRequest = new URL("http://localhost:8080/WebApplication1/webresources/generic?u="+x+"&p="
    String readLine = null;
    HttpURLConnection conection = (HttpURLConnection) urlForGetRequest.openConnection();
    conection.setRequestMethod("GET");

    int responseCode = conection.getResponseCode();
    if (responseCode == HttpURLConnection.HTTP_UNAUTHORIZED)
    {
        out.println("<script type=\"text/javascript\">");
        out.println("alert('User or password incorrect');");
        out.println("</script>");
        RequestDispatcher rd=request.getRequestDispatcher("/Login.html");
        rd.include(request,response);
    }
    else if(responseCode == HttpURLConnection.HTTP_OK)
    {
        String u=conection.getHeaderField("Set-Cookie");
        String[] k=u.split("=");
        String[] kl=k[1].split(";");

        Cookie c=new Cookie(k[0],kl[0]);
        c.setPath("/");
        response.addCookie(c);
        String site="http://localhost:8080/Test1/index.jsp";
        response.sendRedirect(site);
    }
}
```

### 2. LDAP Server Code :

```
{
    Hashtable<String, String> ldapEnv = new Hashtable<>();
    ldapEnv.put( Context.INITIAL_CONTEXT_FACTORY, "com.sun.jndi.ldap.LdapCtxFactory");
    ldapEnv.put(Context.PROVIDER_URL, "ldap://localhost:10389");
    ldapEnv.put(Context.SECURITY_AUTHENTICATION, "simple");
    ldapEnv.put(Context.SECURITY_PRINCIPAL, "uid=admin,ou= system");
    ldapEnv.put(Context.SECURITY_CREDENTIALS, "secret");
    DirContext context = new InitialDirContext(ldapEnv);
    Attributes attributes =new BasicAttributes();
    Attribute attribute =new BasicAttribute("objectClass");
    attribute.add("inetOrgPerson");
    attributes.put(attribute);
    Attribute cn =new BasicAttribute("cn");
    Attribute mail=new BasicAttribute("mail");
    mail.add(request.getParameter("email"));
    cn.add(request.getParameter("fname"));
    Attribute sn=new BasicAttribute("sn");
    sn.add(request.getParameter("lname"));
    Attribute userPassword=new BasicAttribute("userPassword");
    userPassword.add(request.getParameter("pass"));
    attributes.put(userPassword);
    attributes.put(mail);
    attributes.put(sn);
    attributes.put(cn);
    context.createSubcontext("mail="+request.getParameter("email")+" ,ou=Employees,o=Nirmitt",attributes);
    System.out.println(" success");
}
catch(Exception e)
{
    out.println(e);
}
```

### 3. API used for cookie based authentication :

```
Hashtable<String, String> environment = new Hashtable<String, String>();
environment.put(javax.naming.Context.INITIAL_CONTEXT_FACTORY, "com.sun.jndi.ldap.LdapCtxFactory");
environment.put(javax.naming.Context.PROVIDER_URL, "ldap://localhost:10389");
environment.put(javax.naming.Context.SECURITY_AUTHENTICATION, "simple");
environment.put(javax.naming.Context.SECURITY_PRINCIPAL, "mail="+u+",ou=Employees,o=Nirmit");
environment.put(javax.naming.Context.SECURITY_CREDENTIALS, p);

try
{
    DirContext authContext = new InitialDirContext(environment);
    rs=Response.status(Response.Status.ACCEPTED).build();
    ResponseBuilder rb;
    NewCookie c=new NewCookie("sso","true");

    rs=Response.ok().cookie(new NewCookie(c)).build();

    // user is authenticated
}
catch (AuthenticationException ex)
{
    rs=Response.status(Response.Status.UNAUTHORIZED).build();
}
catch (NamingException ex)
{
    ex.printStackTrace();
}
```

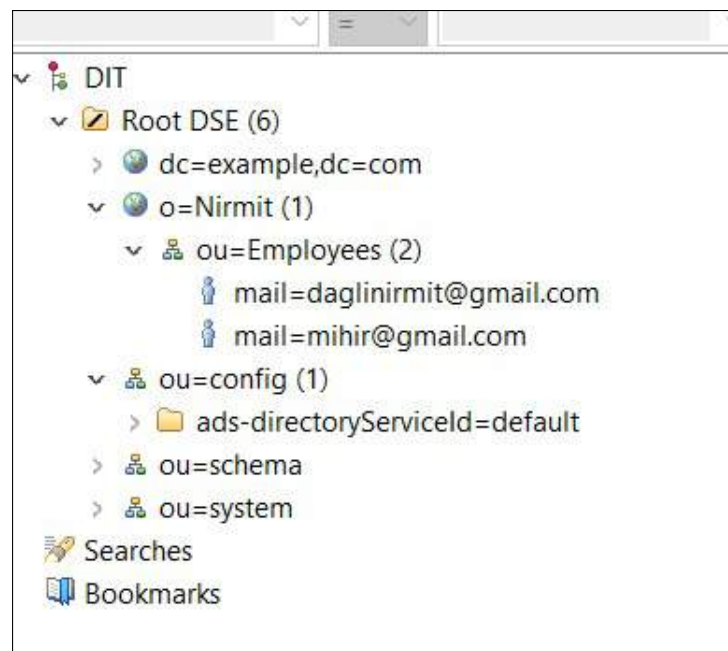
### 4. Android App Snippet :

```
public void signin()
{
    int status=-1;
    OkHttpClient client = new OkHttpClient();
    String x= "daglinirmit@gmail.com"; //un.getText().toString();
    String y= "123456"; //pd.getText().toString();
    String url="http://192.168.1.29:8080/WebApplication1/webresources/generic?u="+x+"&p="+y;
    Request request=new Request.Builder().url(url).build();
    client.newCall(request).enqueue(new Callback() {
        @Override
        public void onFailure(@NotNull Call call, @NotNull IOException e) {
            Log.d( tag: "Error", msg: "fail");
        }

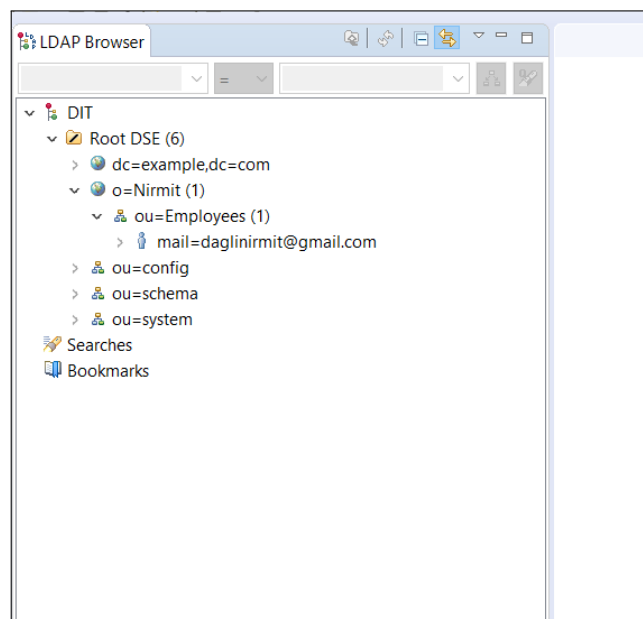
        @Override
        public void onResponse(@NotNull Call call, @NotNull Response response) throws IOException
        {
            try {

                int responsecode = response.code();
                if (responsecode == HttpURLConnection.HTTP_OK) {
                    SharedPreferences sharedPreferences = getSharedPreferences( name: "pref", MODE_PRIVATE);
                    SharedPreferences.Editor editor = sharedPreferences.edit();
                    editor.putString("sso", "true");
                    editor.commit();
                    Intent i = new Intent( packageContext: Login.this, MainActivity.class);
                    startActivity(i);
                    finish();
                }
            }
        }
    });
}
```

5. LDAP server where user details gets stored :



5. Schema of the entries in LDAP Server :





# Chapter 5

## Testing

### 5.1 Testing Approach

For the testing purpose we opted to go for the functional testing methods. It incorporates all test types designed to guarantee each part of a piece of software behaves as expected by using uses cases provided by the design team or business .

**The testing method approaches that we adopted for this system are:**

- Unit Testing
- Integrated Testing
- System Testing

#### 5.1.1 Unit Testing

Unit testing is the first level of testing and is often performed by the developers themselves. It is the process of ensuring individual components of a piece of software at the code level are functional and work as they were designed to. Developers in a test-driven environment will typically write and run the tests prior to the software or feature being passed over to the test team. Unit testing also makes debugging easier because finding issues earlier means they take less time to fix than if they were discovered later in the testing process. Therefore, opting for the unit testing method in our project played a crucial role in assessing each module of the application separately. This testing method best suited our project as we had various modules at the start which were to be tested and verified. It made the testing process easier by helping us discover the minute errors in each module and therefore we could rectify them efficiently.

#### 5.1.2 Integrated Testing

After each unit is thoroughly tested, it is integrated with other units to create modules or components that are designed to perform specific tasks or activities. Integration testing is a level of software testing where individual units are combined and tested as a group. The purpose of this level of testing is to expose faults in the interaction between integrated units. Integration testing was a necessity to check whether each individual module/unit was working well in synchronisation with one another. There were multiple problems while integrating the various modules which were only discovered with the help of integration testing methodology. For example we found out a cookie can be shared between two websites so that user does not need to login again once he accesses another website.

#### 5.1.3 System Testing

System testing is a black box testing method used to evaluate the completed and integrated system, as a whole, to ensure it meets specified requirements. The functionality of the software is tested from end-to-end and is typically conducted by a separate testing team than the development team before the product is

pushed into production.

Finally, the entire system was tested as a whole using the System testing Methodology. Here, the functional requirements of our applications that include Logging in, Sign Up, Administrative function, API's , Database records where thoroughly checked to see if they were working properly . The applications was given to other professionals (faculty) who checked and verified the proper working of all the modules and the system as a whole.

## 5.2 Test Cases

Test ID	Testcase	Description	Expected Result	Actual Result
1	Registration User	Validate the details and store the details and show the message	Pass / Fail	Pass
2	Register Client	Validate the details and store the details and provide API key to the clients	Pass / Fail	Pass
3	Invalid Registration Details For Users	Show the message	Pass / Fail	Pass
4	Invalid Registration Details for Client	Show the message	Pass / Fail	Pass
5	User Login	Provide access to underlying features	Pass / Fail	Pass
6	Invalid Username & Password	No access to underlying features and error message	Pass / Fail	Pass
7	Forget Password	Enter Registered Email id password must be reset and send to users Emailid Pass	Pass / Fail	Fail
8	Change Password	User must be Logged in to change password	Pass / Fail	Fail
9	Update Profile Details	User Details must be updated successfully	Pass / Fail	Fail
10	Change Password	If user does not enter correct current password, he must not be able to change his password	Pass / Fail	Fail
11	Admin Login	If user id and password is correct provide access to underlying features	Pass/Fail	Fail

Table 5.1: Test cases

# Chapter 6

## Conclusions and Future Scope

### 6.1 Conclusion

After completing the project, the lessons learned is we must design a good user interface for the system. The Single Sign On System is developed with the aim of reducing the burden of the users of remembering different set of credentials for different websites, web application, mobile apps etc and also reduces the burden of login to applications again and again that is if they login into one website they are automatically logged into all other website. .

### 6.2 Future Scope

By increasing the users of the applications and websites that should often requires access to remote resource, different authentication techniques are used when users want to access the systems. Therefore, SSO technology has been introduced as a special form of authentication mechanisms. This technology is meant to facilitate the job for users in a way that with one credentials they could be able to access to several software resources on different servers.

# Bibliography

- [1] APPLICATION OF SINGLE SIGN-ON (SSO) IN DIGITAL CAMPUS Jian Hu, Qizhi Sun, Hongping Chen
- [2] An Automated Enterprise IT Management System Based on LDAP
- [3] OAuth-SSO: A Framework to Secure the OAuth-based SSO Service for Packaged Web Applications
- [4] Moodle Server, <https://cloudcone.com/docs/article/how-to-install-moodle-on-ubuntu-18-04/>,
- [5] LDAP Server, <https://computingforgeeks.com/install-and-configure-openldap-pldap-admin-on-ubuntu-18-04-lts/>,

## Acknowledgement

We have great pleasure in presenting the report on **Single Sign On**. We take this opportunity to express our sincere thanks towards our guide **Ms. Kaushiki Upadhyaya** & Co-Guide **Ms. Nahid Shaikh** Department of IT, APSIT thane for providing the technical guidelines and suggestions regarding line of work. We would like to express our gratitude towards his constant encouragement, support and guidance through the development of project.

We thank **Mr. Kiran B. Deshpande** Head of Department,IT, APSIT for his encouragement during progress meeting and providing guidelines to write this report.

We thank **Ms.Anagha Aher** & **Ms.Apeksha Mohite**, BE project co-ordinator, Department of IT, APSIT for being encouraging throughout the course and for guidance.

We also thank the entire staff of APSIT for their invaluable help rendered during the course of this work. We wish to express our deep gratitude towards all our colleagues of APSIT for their encouragement.

**Nirmit Dagli:**  
**16104013:**

**Mihir Deorukhkar:**  
**16104065:**

**Sarvesh Sawant:**  
**17204013:**

# Publication

Paper entitled “**Implementation of Single Sign on (SSO) for College websites** ” is presented at “**International Research Journal of Engineering and Technology (IRJET)** ” by “**Nirmit Dagli, Mihir Deorukhkar & Sarvesh Sawant**”.