**Design Document**

Open Projects

Submitted To-

Prof. Lydia Jane

By-

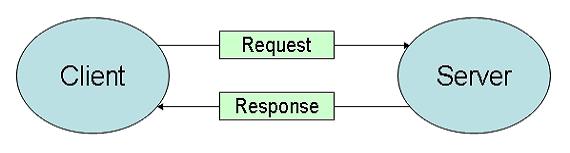
Harjatin Singh – 12BCE0457 Deepak Kumar – 12BCE0534

**Contents**

1. Purpose
2. Design Outline
   1. Client Server Model
   2. Components of the System
   3. Component Diagram
   4. Architecture Diagram
   5. Context Diagram
3. Design Issues
   1. Functional Issues
   2. Non-Functional Issues
4. Design Details
   1. Class Diagram
   2. Sequence Diagram
   3. Activity Diagram
5. **Purpose**

With an ever growing need for co creation and collaboration amongst technicians from various streams to create cross community projects, there also arises a need for a system that would enable innovators to easily find other people with the required skill set to help them take their idea forward. This is the most major problem that our system caters to.

1. **Design Outline**
   1. **Client Server Model:**



The application will be stored on servers but the user will access through client, where he can perform all the operations through a user interface, like adding a new project, managing them, adding a team, joining a project and updating profile. On the other hand, all the information (user details, project details, etc.) will be actually stored on the server.

When a user performs any operation such as adding a new project or updating his profile, he directly accesses his data on the servers through the client where the application is running. In case of adding a new project, the user generates a request through the client to add a new project, which the client executes and sends the data to the server to store it on that particular user’s directory.

**2.2 Components of the System**

* 1. Log in
  2. Add New Project
  3. Send Project Join Request
  4. Accept Project Join Request
  5. Form a permanent team
  6. Create and Edit Profile
  7. Add notes

**2.3 Component Diagram**

Create\_auth\_details

User Details

User Registration

delete

add

Update

**v**

User Profile

And timeline

Login to account

Add & Manage

Projects

And Teams

User Login

Project and Team details

User Details

Database Access System

**User Registration:-**

This component registers new users to the web application. It takes its user details such as name, last name, email id and contact number and each user creates authorization details i.e. Username and Password to access their account.

**User Login:-**

The user login component is for those who have already registered for the applications to help with the problem of project management. The login will take them into the user’s home page where they can view and manage their projects and teams. The notes and notifications can now be managed from the user’s dashboard.

**Add & Manage Projects:-**

This component is the heart of the application as this is where the purpose is really fulfilled. The Project Manager lets user to add, edit and manage projects and teams, not only manage the projects but also manage team members working on the project as per their needs. Ease of use is ensured by an intuitive User Interface.

**Database Access System:-**

The database stores all the details of the user such as specialization and bio and his projects and teams. The data is stored in the form of objects where each user is an object, each project is an object and each team is an object with their details and these are all interrelated.

The **User Registration** component interacts with the **User Login** component to let the new user login to his Home Page. The user login is for new and returning users. Through home page, **all operations** of related to project, team and notes and request are managed here. The **database** stores all the relevant details of the user, projects and teams.

**2.4 Architecture Diagram:**

The following diagram depicts the high level specification of the system.

Internet

Projects saved by user

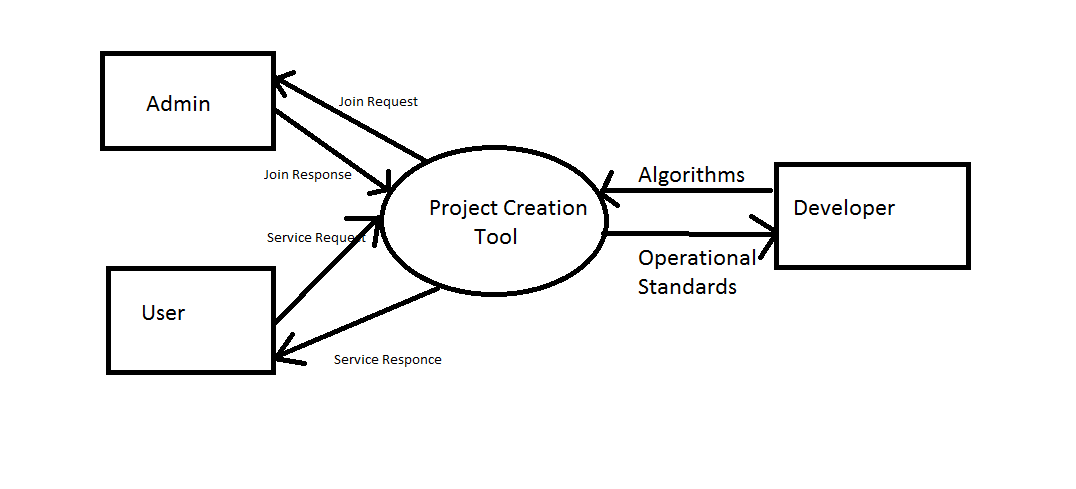
Projects saved by team

Notes saved by a user

Bio created by a user

**The diagram shows how over the internet clients via registering on the application can save their required purpose.**

**2.5 Context Diagram:**

****

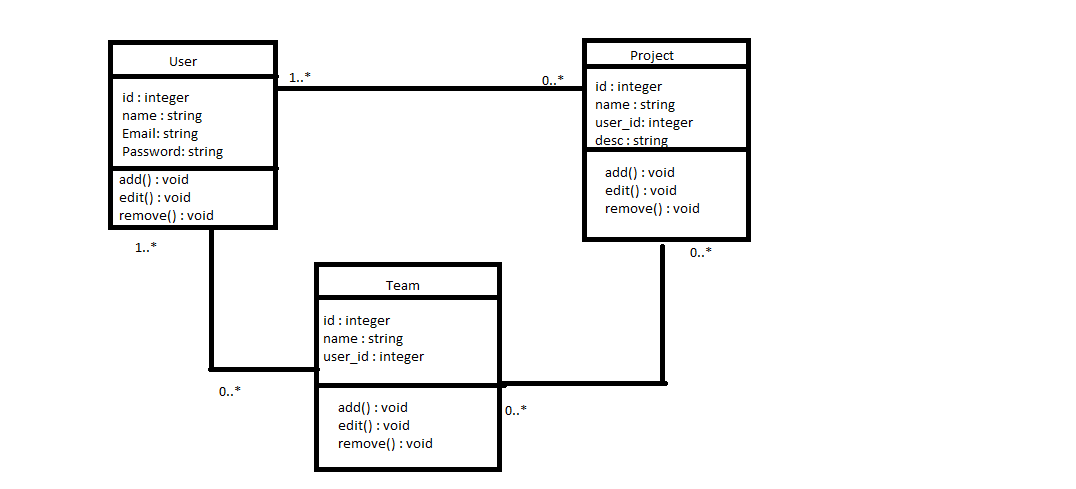
1. **Design Issues:**

**3.1 Functional Issues:**

* The application won’t be able to have the iFrame feature we thought of as sites like google, facebook, gmail etc. don’t allow it.
* **Non Functional issues :**
* Privacy:  
  The users personnel details shall not be shared with any advertisement or funding agency without users’ consent.
* Security:   
  The application will be hosted on safe and secure cloud servers. All the registration information of the users will be safe and secure and will not be shared with any third party. All personnel information will be encrypted using state of the art techniques.
* Failure Management:  
  In case of any disaster or unexpected failure the user’s data will not be compromised because it will be stored on reliable servers. Periodic backup of user data will ensure ease of recovery.
* Price:   
  The application is free to use. No charges of any kind to use the application will be collected.
* Response time:   
  The response time of the application will depend on the bandwidth of the internet at which the user is accessing the application. However the minimum time to load the application from our server side will be not more than 1.5 sec.
* Usability:   
  It will be extremely easy to use the application. It would take about 3 minutes to get to know all the features of application and how to use it. The new user registration will take 30 seconds.
* Scalability:   
  The application will be designed keeping in mind the population using the internet and so there will be real time monitoring of number of users registering for it and the extra window will be implemented. Also since the application will be hosted on cloud based servers, they can be unscaled anytime according to requirement.

1. **Design Details**

**4.1 Class Diagram**

****

**Class User:-**

User class maintains details of the user, its username, password and a user has the following functions:-

1. Add a Project
2. Add a team
3. Add a note
4. Edit bio and add specializations

The following functions have to be taken care of on the request of the user. A user can have zero or more projects or zero or more teams.

**Class Project:-**

A project is an independent entity which is related to one or more users and zero or more teams. The admin/creator of the project has the right to create/edit project name and description. Admin can also add project members.

**Class Team:-**

A team is an independent entity which is related to one or more users and zero or more projects. The admin/creator of the project has the right to create/edit team name and description. Admin can also add team members. A team basically functions by removing redundancy of adding frequently collaborating users again and again. Adding a team project after adding team members is sufficient to add the project to all members of the team.

**4.2 Sequence Diagram**

**Registration:-**

Web Interface

Enter Details

Create Username & Password

Confirmation

**Login:-**

Authenticate user

Enter username & password

Web Interface

Home Page

**Adding a Project:-**

Project

details

Add new Project

Web Interface

Login

Project added by user

**Adding a Team:-**

Team

details

Add new Team

Web Interface

Login

Team added by user

**Adding a Note:-**

Project

Details of note

Add new Note

Web Interface

Login

Note added by user

**Deleting a Note:-**

Delete Note

Select Note

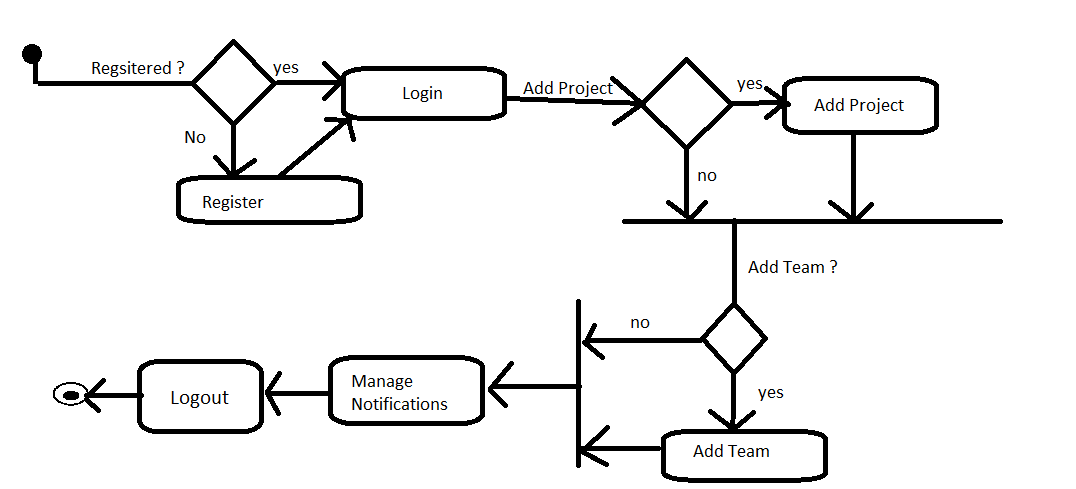
Web Interface

Login

Note

deleted

**4.3 Activity Diagram**

****