**Open Projects**

**A PROJECT REPORT**

*submitted by*

**Harjatin Singh Baweja**

**12bce0457**

**Deepak Kumar Singh**

**12bce0534**

**Faculty: Prof.Lydia Jane G**

**Slot: B2**

**Software Engineering**

**CSE325**

**VIT**

U N I V E R S I T Y

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

Vellore - 632 014, Tamil Nadu, India

**School of Computing Science and Engineering**

April 2015

[1. Introduction 3](#_Toc384642217)

[1.1 Abstract 3](#_Toc384642218)

[1.2 Purpose 3](#_Toc384642219)

[1.3 Literature Survey 3](#_Toc384642220)

[2. Planning & Scheduling 3](#_Toc384642222)

[2.1 SDLC model 3](#_Toc384642223)

[2.2 Gnatt Chart 3](#_Toc384642224)

[2.3 PERT Chart 3](#_Toc384642225)

[2.4 Timeline Chart 3](#_Toc384642226)

[3. Software Requirements Specification 3](#_Toc384642227)

[3.1 Functional Requirements 3](#_Toc384642228)

[3.2 System Requirements 4](#_Toc384642229)

[3.3 Non-Functional Requirements 4](#_Toc384642230)

[3.4 Assumptions & Constraints 4](#_Toc384642231)

[4. Software Design Specification 4](#_Toc384642232)

[4.1 Architecture Design 4](#_Toc384642233)

[4.2 Component/Module Design 4](#_Toc384642234)

[4.3 System Models 4](#_Toc384642235)

[4.4 User Interface Design 5](#_Toc384642236)

[4.5 Alternate Designs 5](#_Toc384642237)

[5. Coding 5](#_Toc384642238)

[6. Testing 5](#_Toc384642239)

[6.1 Test Cases 5](#_Toc384642240)

[6.2 Test Execution Report 5](#_Toc384642241)

[6.3 Automated Testing Tools - A study 5](#_Toc384642242)

[7. Configuration Management 5](#_Toc384642243)

[8. Conclusion 5](#_Toc384642244)

[9. Bibliography 5](#_Toc384642245)

# Introduction

## Abstract

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.

As we know that students, especially in the engineering community come up with ideas that have the capability make several day to day tasks easier. But it is often seen that these ideas are dashed to the ground due to lack of skill set possessed by one single person to bring this idea to reality.

It is in this scenario that an individual wishes for a system that would help him find people with the required skills to help him co create. The system should not only aid this process but also act as a medium to keep all the project members updated about the developments of the project using images, videos or comments. The system should also reflect the kind of work a person has done in terms of projects in the past. This would help other people pick their team members more wisely. The user would also want to have a profile displaying users’ skill set for other people to see.

## Purpose

Currently there exists no such intra-college platform in any major universities providing the above mentioned opportunities to its students. It is often witnessed that students struggle to find enough people with required skills of other branches to create projects that span across multiple disciplines. This sought of co-creation has made MIT Media Labs a giant beast producing several state-of-the-art technologies. This system has the capability of becoming a social network for project development.

## Literature Survey

Project management software has the capacity to help plan, organize, and manage resource pools and develop resource estimates. Depending on the sophistication of the software, it can manage [estimation](http://en.wikipedia.org/wiki/Software_development_effort_estimation) and planning, [scheduling](http://en.wikipedia.org/wiki/Schedule_(workplace)), [cost control](http://en.wikipedia.org/wiki/Cost_control) and [budget management](http://en.wikipedia.org/wiki/Budget_management), [resource allocation](http://en.wikipedia.org/wiki/Resource_allocation), [collaboration software](http://en.wikipedia.org/wiki/Collaboration_software), [communication](http://en.wikipedia.org/wiki/Communication), [decision-making](http://en.wikipedia.org/wiki/Decision-making_software), quality management and [documentation](http://en.wikipedia.org/wiki/Documentation) or administration systems. Today, numerous PC & browser based project management software solutions exist and they are finding their way into almost every type of business.

Existing systems are:

1. Atlassian
2. Basecamp
3. Clarizen
4. Podio

### Current System

**Basecamp** is a [web-based](http://en.wikipedia.org/wiki/Web_application) [project-management](http://en.wikipedia.org/wiki/Project_management) tool developed by [Basecamp](http://en.wikipedia.org/wiki/Basecamp_(company)) and launched in 2004.[[1]](http://en.wikipedia.org/wiki/Basecamp_Classic#cite_note-launch-1) A new version was launched in 2012.[[2]](http://en.wikipedia.org/wiki/Basecamp_Classic#cite_note-launch2-2) The [Ruby on Rails](http://en.wikipedia.org/wiki/Ruby_on_Rails) [framework](http://en.wikipedia.org/wiki/Web_application_framework) was extracted from the Basecamp project.

Basecamp offers [to-do lists](http://en.wikipedia.org/wiki/Time_management#Task_list), [wiki](http://en.wikipedia.org/wiki/Wiki)-style web-based text documents, milestone management, file sharing, time tracking, and a messaging system. Basecamp Classic also offers integration with [Basecamp](http://en.wikipedia.org/wiki/37signals)'s own [Campfire](http://en.wikipedia.org/wiki/Campfire_(software)) product, and features [APIs](http://en.wikipedia.org/wiki/API) that are used by a host of web and mobile apps.

**Clarizen, Inc.** is a work management and [project management](http://en.wikipedia.org/wiki/Project_management) [software](http://en.wikipedia.org/wiki/Software) company based in [San Mateo, California](http://en.wikipedia.org/wiki/San_Mateo,_California). The company was founded in 2005 and the software was first released for public use in 2007. Clarizen's primary product is a [collaborative project execution software](http://en.wikipedia.org/wiki/Collaboration_platform).

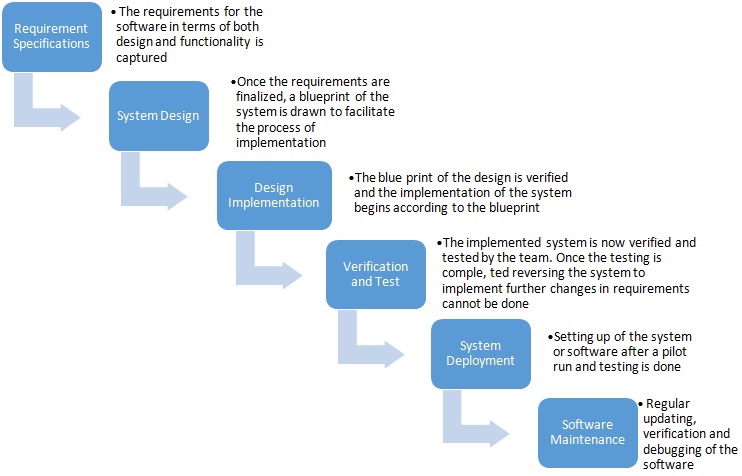
### Proposed System

# Planning & Scheduling

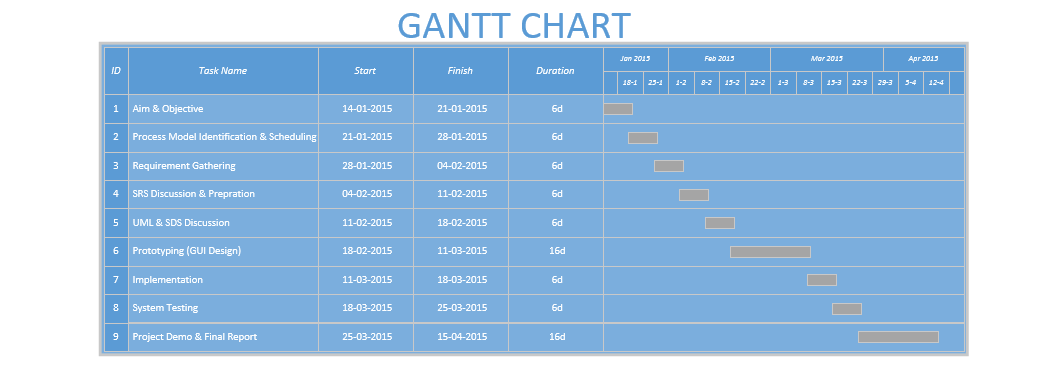
## SDLC model

All features of this Software are known prior to the Development process and are not subjected to change. Thus a model apt for static requirements is to be used – **Waterfall model**.

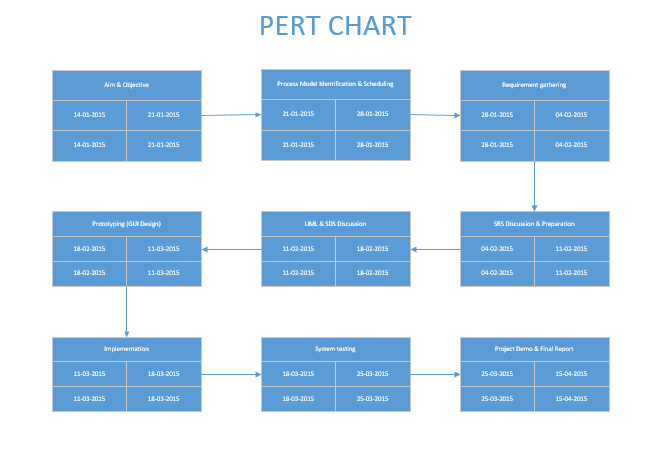
For the user interface, exploratory design pattern will be followed.



## Gantt Chart



## PERT Chart



## Timeline ChartG:\timeline.png

# Software Requirements Specification

## Functional Requirements

* + Signup

The system should allow user create a new account i.e. the a valid user account should be permitted.

* + Log in

After proper authentication, the user should be allowed to access his/her personnel information.

* + Add New Project

This is the most basic requirement of a project management system. The user should be allowed to create a new project after providing necessary details such as project title, start date, end date etc.

* + Send Project Join Request

The admin of a project should be able to send a project join request to any one registered with the system.

* + Accept Project Join Request

The receiver of Project Join request must be able to accept or reject a project join request

* + Form a permanent team

It is very time consuming to send individual request to each project member and for them to accept it, hence it is convenient for the user to form a team of people who often work together and then create team projects.

* + Create and Edit Profile

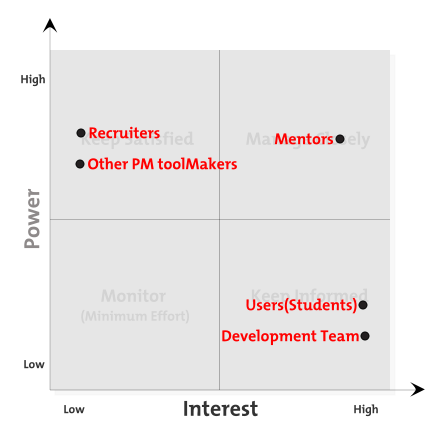
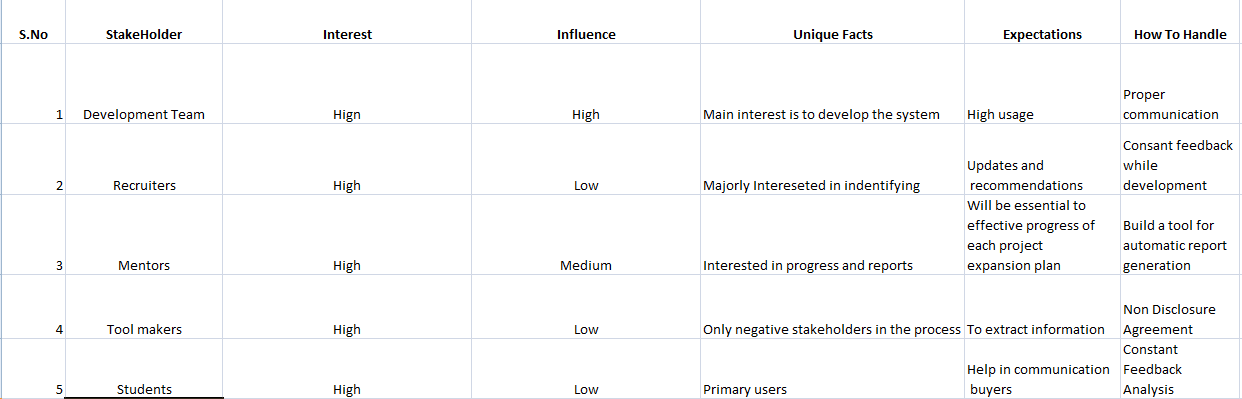
The user’s work is seen and acknowledged by his/her fellows and Project Join Requests are sent to unknown people based on their profile. It is thus imperative for a user to maintain and update their profile information.

* + Add notes and reminders

It would often be required by the user to add notes and reminders regarding project deadlines.

### Stakeholders

* Project Development and Design Team
* Users
* Mentors
* Other Project Management Tool Makers
* Recruiters



### Modules

### User stories/requirements

1. Register:   
   As a new user I would like to register on your web application. So that I can create new projects and find team members.
2. Login:   
   As an existing user, I would like to log in to my account to access my customised feed.
3. Add New Project:   
   As a current user of the application I would like to add a new project for which I will be the admin.
4. Send Join Project Request:   
   As a logged in user in your application I would like send a project join request to a suitable user for further development.
5. Accept Join Request:

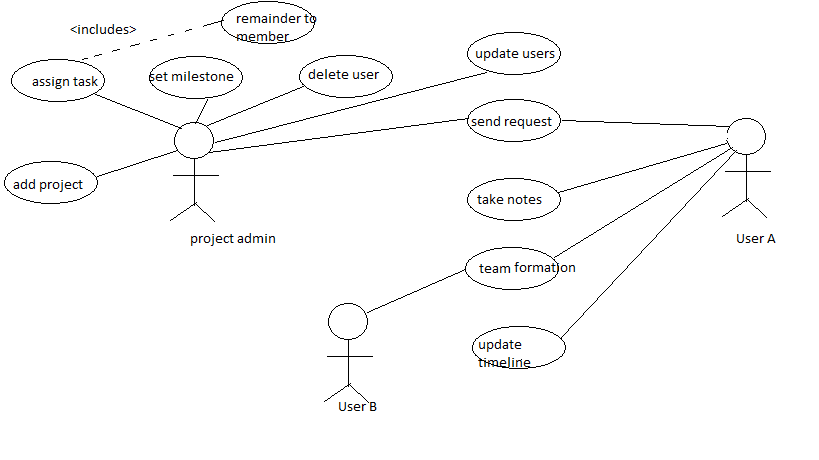
As a user of the application I would like to accept requests from the project admin to join the project.

1. Create Team:

As a user of the application, I would want to create a team of members who usually work together so that I do not have to send requests every time I create a new project.

1. Add Notes:   
   As a user of the application I would like make notes and reminders regarding various projects and related deadlines.
2. Project Timeline:  
   As a user of the application I want to update project timeline so that I can keep rest of the members updated with what I have done.
3. Manage Profile:   
   As a registered user of the application I would like create and manage my profile so that people can know my skill set and approach me to work on related projects.

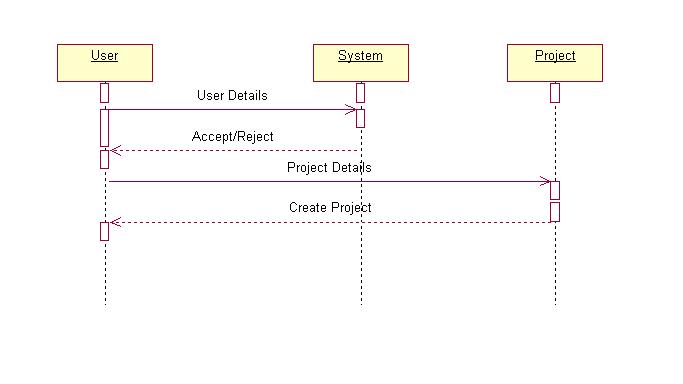
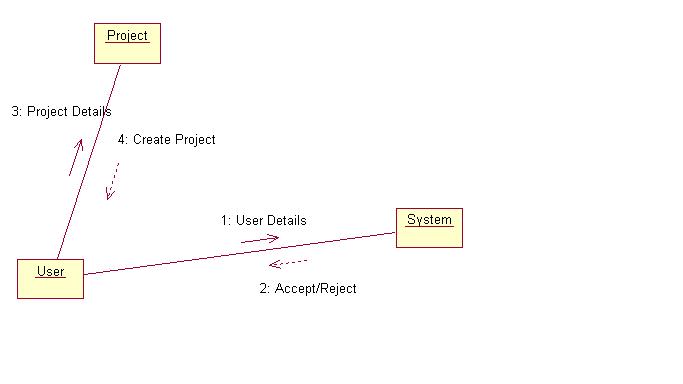
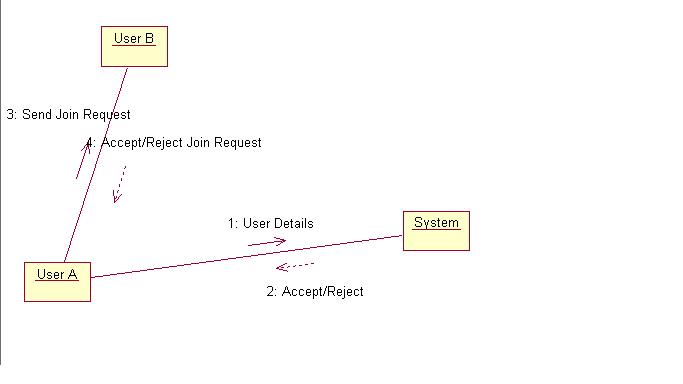
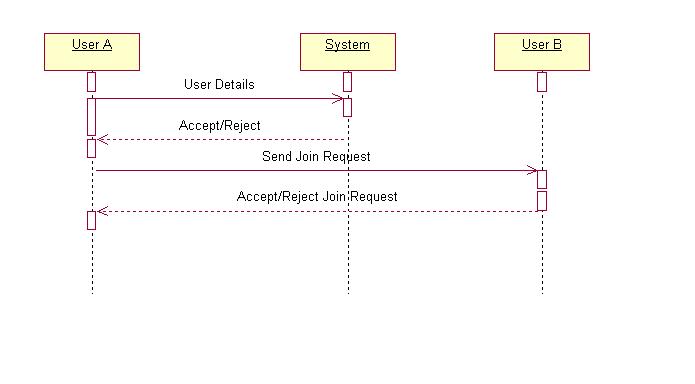
#### Use Case Diagram

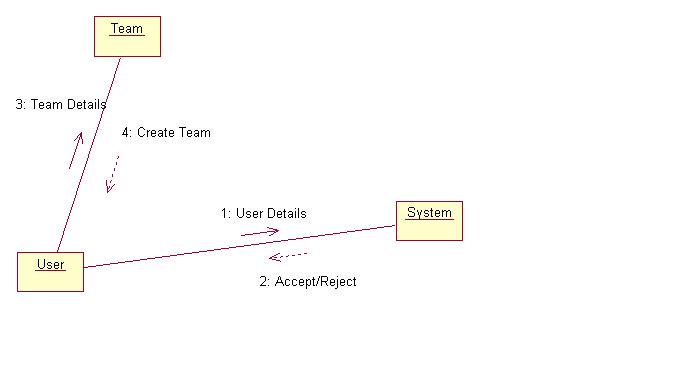
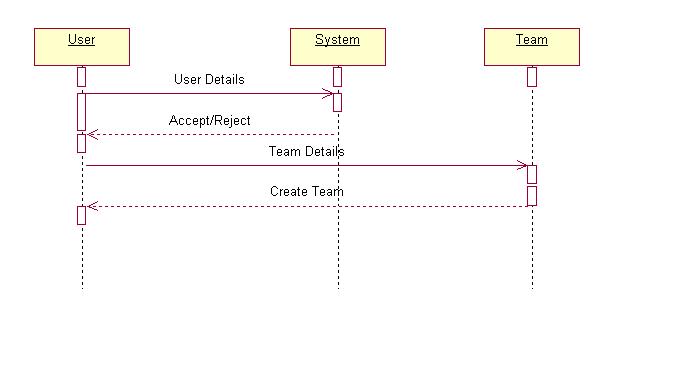


#### Sequence Diagram

A **Sequence diagram** is an interaction diagram that shows how processes operate with one another and what is their order. It is a construct of a Message Sequence Chart. A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the Logical View of the system under development. Sequence diagrams are sometimes called **event diagrams** or **event scenarios**.

Messages, written with horizontal arrows with the message name written above them, display interaction. Solid arrow heads represent synchronous calls, open arrow heads represent asynchronous messages, and dashed lines represent reply messages. If a caller sends a synchronous message, it must wait until the message is done, such as invoking a subroutine. If a caller sends an asynchronous message, it can continue processing and doesn’t have to wait for a response.



#### Use Case Description

|  |  |
| --- | --- |
| Use Case ID | 1 |
| **Use Case Name** | Add Project |
| **Summary** | The System will add to the database, the following parameters   * Name Of the Project * Description Of the Project * Admin of the Project * Team members |
| **Pre-Conditions** | The User wanting to add project must have registered with the application and must be logged in. |
| **Success End Conditions** | An separate entity named project with the provided details is created |
| **Failed End Conditions** | Please register/login to complete the action. |
| **Primary, Secondary Actors** | Project admin, added members |
| **Trigger** | This Use Case is initiated based on the user’s request. |
| **Description** | The User logins into this application and creates a new project as soon as he has selected the desired project name, description and team members. |
|  |  |

|  |  |
| --- | --- |
| Use Case ID | 2 |
| **Use Case Name** | Add Team |
| **Summary** | The System will add to the database, the following parameters   * Name Of the Team * Description Of the Team * Admin of the Team * Team members |
| **Pre-Conditions** | The User wanting to add team must have registered with the application and must be logged in. |
| **Success End Conditions** | An separate entity named team with the provided details is created |
| **Failed End Conditions** | Please register/login to complete the action. |
| **Primary, Secondary Actors** | Team admin, added members |
| **Trigger** | This Use Case is initiated based on the user’s request. |
| **Description** | The User logins into this application and creates a new team as soon as he has selected the desired team name, description and team members. |
|  |  |

|  |  |
| --- | --- |
| Use Case ID | 3 |
| **Use Case Name** | Add Note |
| **Summary** | The System will add to the database, the following parameters   * Name Of the User * Description Of the note * Project associated |
| **Pre-Conditions** | The User wanting to add a note must have logged in and have an associated project. |
| **Success End Conditions** | An separate entity named note with the provided details is created |
| **Failed End Conditions** | Please register/login to complete the action. |
| **Primary, Secondary Actors** | User creating the note |
| **Trigger** | This Use Case is initiated based on the user’s request. |
| **Description** | The User logins into this application and creates a new note and associates it with the desired project. |
|  |  |

|  |  |
| --- | --- |
| Use Case ID | 4 |
| **Use Case Name** | Update Profile |
| **Summary** | The System will add to the database, the following parameters   * User’s Projects * User’s Skill set * User’s Description |
| **Pre-Conditions** | The User wanting to update profile must have registered with the application and must be logged in. |
| **Success End Conditions** | Users profile information is updated. |
| **Failed End Conditions** | Please register/login to complete the action. |
| **Primary, Secondary Actors** | Logged in user |
| **Trigger** | This Use Case is initiated based on the user’s request. |
| **Description** | The User logins into this application and updates profile as soon as he has selected the projects, description and skills. |
|  |  |

## System Requirements

## Hardware requirements:

## For high-traffic sites may be dependent on the following issues: number of transactions per second; number of hits per second; number of queries per second; number of queries done by RDBMS per second; number of pages served per second involving all of the above parameters. Some other factors that need to be considered when setting up a high traffic site include clustering i.e. use of backup servers which automatically takes over operations in case of failure of primary ones. Low-traffic sites can be easily served from a single machine depending on the needs of the business. Pentium II/III based Intel server running Linux can serve hundreds of unique customers each day.

## Software requirements:

## Several software are available free on the Internet that can be used to build e-commerce exchanges. Some examples are Apache Web Server, Apache-Jserv Servlet Engine, Linux Operating System, mySQL database, postgresql etc. Many of these open source software may not be adequate for high-traffic sites.

## Non-Functional Requirements

* 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.

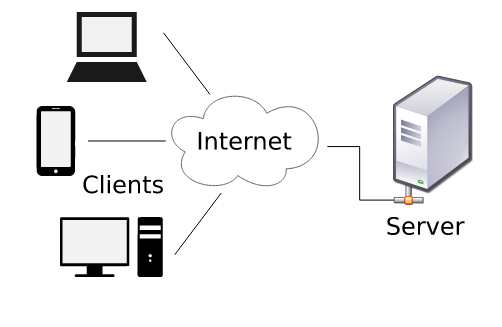
# 

# Software Design Specification

## Architecture Design

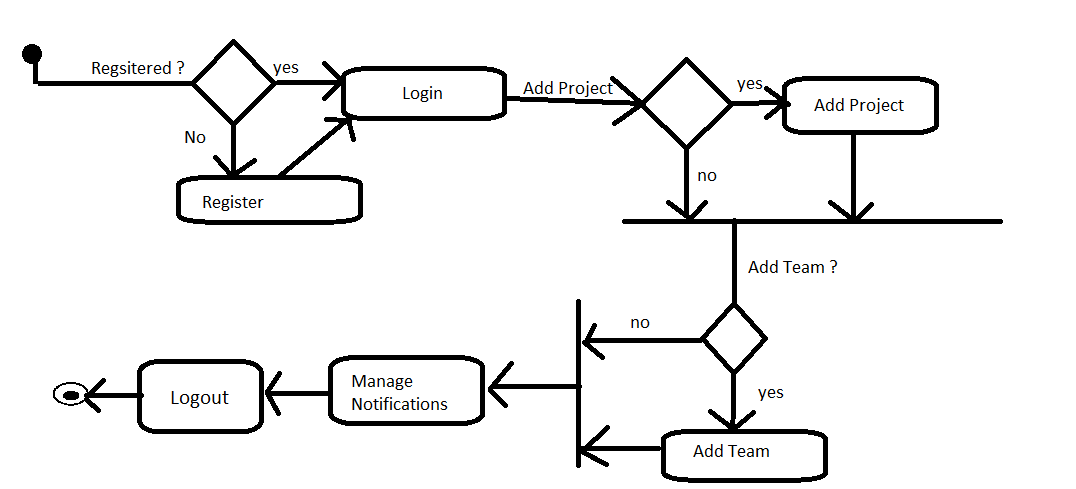
The **client–server model** of computing is a [distributed computing](http://en.wikipedia.org/wiki/Distributed_computing) structure that partitions tasks or workloads between the providers of a resource or service, called [servers](http://en.wikipedia.org/wiki/Server_(computing)), and service requesters, called [clients](http://en.wikipedia.org/wiki/Client_(computing)).[[1]](http://en.wikipedia.org/wiki/Client%E2%80%93server_model#cite_note-1) Often clients and servers communicate over a [computer network](http://en.wikipedia.org/wiki/Computer_network) on separate hardware, but both client and server may reside in the same system. A server host runs one or more server programs which share their resources with clients. A client does not share any of its resources, but requests a server's content or service function. Clients therefore initiate communication sessions with servers which await incoming requests.

Examples of computer applications that use the client–server model are [Email](http://en.wikipedia.org/wiki/Email), [network printing](http://en.wikipedia.org/wiki/Network_printing), and the [World Wide Web](http://en.wikipedia.org/wiki/World_Wide_Web).

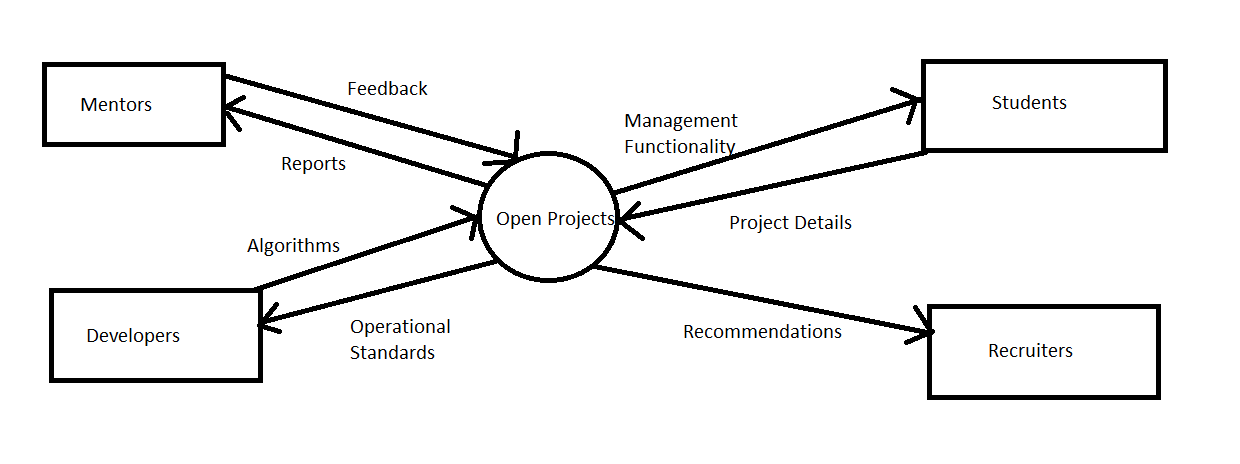


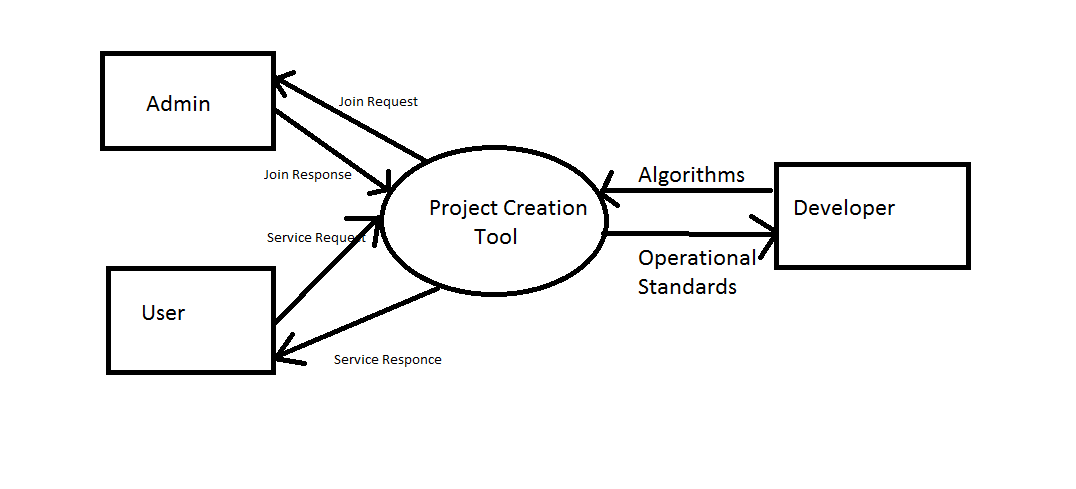
## Component/Module Design

## Activity Model



### Context Model

A **System Context Diagram** (SCD) in [software engineering](http://en.wikipedia.org/wiki/Software_engineering) and [systems engineering](http://en.wikipedia.org/wiki/Systems_engineering) is a [diagram](http://en.wikipedia.org/wiki/Diagram) that defines the boundary between the [system](http://en.wikipedia.org/wiki/System), or part of a system, and its environment, showing the entities that interact with it.[[2]](http://en.wikipedia.org/wiki/System_context_diagram#cite_note-2) This diagram is a high level view of a [system](http://en.wikipedia.org/wiki/System). It is similar to a [block diagram](http://en.wikipedia.org/wiki/Block_diagram).



### Behavioral/Process Model

DFDs

### Data Model

#### Database Description

#### ER Diagram

### Object Model

#### 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.

## User Interface Design

* snapshots of GUI, tools used etc.,

## Alternate Designs

* Other alternate design methods/solutions available for the current scenario. Why were the other designs not chosen and justification on the chosen solution
* Ex: client side checking Vs Server side checking and etc.,

# Coding

* sample code

# Testing

## Test Cases

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case id | Purpose | Steps | Expected Result | Actual Result | Test Case Status  Pass/Fail |
|  | <To test what> |  |  |  |  |
|  |  |  |  |  |  |

## Traceability Matrix

## Test Execution Report



##### Testing Tools

* [TET (Test Environment Toolkit)](http://tetworks.opengroup.org/Products/tet.htm)
  + The goal behind creating the Test Environment Toolkit (TET) was to produce a test driver that accommodated the then current and anticipated future testing needs of the test development community. To achieve this goal, input from a wide sample of the community was used for the specification and development of TET’s functionality and interfaces.
* [TETware](http://tetworks.opengroup.org/)
  + The TETware is the Test Execution Management Systems which allows you to do the test administration, sequencing of test, reporting of the test result in the standard format (IEEE Std 1003.3 1991) and this tools is supports both UNIX as well as 32-bit Microsoft Windows operating systems, so portability of this is with test cases you developed. The TETware tools allow testers to work on a single, standard, test harness, which helps you to deliver software projects on time. This is easily available for download on ftp download.
* [Test Manager](http://www.durablesoftwareltd.com/)

The Test Manager is an automated software testing tool is used in day to days testing activities. The Java programming language is used to develop this tool. Such Test Management tools are used to facilitate regular Software Development activities, automate & mange the testing activities. Currently Test Manager 2.1.0 is ready for download.

* [Apache](http://sourceforge.net/projects/rth/) JMeter
  + It is an [Apache](http://en.wikipedia.org/wiki/Apache_Software_Foundation) project that can be used as a [load testing](http://en.wikipedia.org/wiki/Load_testing) tool for analyzing and measuring the performance of a variety of services, with a focus on [web applications](http://en.wikipedia.org/wiki/Web_application)

.

# Configuration Management

* how was version control done
* any specific tools used for the purpose

# Conclusion

# Bibliography