**181606: Project II**

**Project Report**

**On**

**College Commune**



**As partial fulfillment of award of**

**Bachelor of Engineering**

**In**

**Information Technology**

**Prepared by:**

**100010116004 Harsh Bhakt**

**100010116013 Vaishnavi Patel**

**100010116022 Neel Maheshwari**

**Internal Guide:**

**Prof. Urmi Agravat**

**A. D. Patel Institute of Technology**

**Gujarat Technological University**

**(December - 2013)**



**Department of Information Technology**

1. **D. Patel Institute of Technology**

**Gujarat Technological University**

**CERTIFICATE**

**This is to certify that Harsh Bhakt (100010116004), Vaishnavi Patel (100010116013) and Neel Maheshwari (100010116022) of final year Information Technology have satisfactorily completed their partial work entitled “College Commune” for the subject 170001 Project I in the first semester of academic year 2013-14 for the partial fulfillment of the award of the Bachelor of Engineering in Information Technology at Gujarat Technological University.**

**Date: 10/12/2013**

**Project Guide (Internal) Head of Department**

**Prof. Urmi Agravat Prof. Sudhir Vegad**

**Principal of Institute**

**Dr. R.K. Jain**

**Acknowledgement**

It is our pleasure to present the project report as the Information Technology Project of “College Commune” as our final year project.

We are thankful for the assistance received from various individuals from different areas in making of this project. We sincerely thank Prof. Urmi D. Agravat, Associate Faculty guide in our project for guiding us throughout the project, without which this project would not have been distant reality

We articulate deep sense of respect to Prof. Sudhir Vegad (Head, Information Technology Department) for providing us an opportunity to carry out the project, for showing trust on us. We are thankful to all staff members of Information Technology and Computer Engineering Department of our college for providing us a helping hand during the project.

Last but not the least we also express our gratitude to our institute-A.D.Patel Institute of Technology to allow us to carry out the Project Work at their institution and utilizing their provided facilities.

Name Sign

Harsh Bhakt

Vaishnavi Patel

Neel Maheshwari

**Abstract**

There are the times when the flaw of communication within class, department or college leads to problems. Few of them are students may not aware about the 'Notice' of Scheduled faculty meeting which is pinned on a general notice board of institute. Sometimes faculties may be looking for details of the student on hand in some unusual conditions but they can't have it easily, details may not be available with traditional document or it may be hard to abstract details from bulk of papers.

This project is supposed to bridge the communication flaw gap among College/ Educational Institution. It will provide the shared platform for class to share the latest issue of field. Students of class may want to share important and current issue regarding study or related area for being updated with current time, which will be fulfilled by this application.

Moreover one of the project modules will give facility to publish notice to the target users of Class/Department/Institute. Time Bound Notice will set the schedule to users account calendar and will give the reminder prior to that activity. Profile Manager will keep updated details of Student or faculty. System will offer a general platform to class members to share latest issues and respected views for the same.

**TABLE OF CONTENT**

|  |  |  |  |
| --- | --- | --- | --- |
| **Acknowledgement** | | | |
| **Abstract** | | | |
| **1. Introduction** | | | **1** |
|  | 1.1 Background in brief | | 1 |
|  | 1.2 Objectives | | 1 |
|  | 1.3 Purpose | | 1 |
|  | 1.4 Modules of the system | | 2 |
|  | 1.5 Application | | 2 |
|  | 1.6 Scope of Project | | 2 |
|  | 1.7 Expected Outcome | | 3 |
| **2. Background Study and Literature Review** | | | **5** |
|  | 2.1 Android | | 5 |
|  | 2.2 SWOT Analysis | | 6 |
|  |  | 2.2.1 Strength | 6 |
|  |  | 2.2.2 Weakness | 7 |
|  |  | 2.2.3 Opportunities | 7 |
|  |  | 2.2.4 Threats | 7 |
|  | 2.3 Why College Commune Application | | 7 |
|  | 2.4 Study of pre-existing System | | 8 |
| **3. Analysis and Design** | | | **9** |
|  | 3.1 Requirement analysis and gathering | | 9 |
|  | 3.2 System Specification | | 9 |
|  | 3.3 System Requirement | | 9 |
|  | 3.4 Functional/Non-functional Requirement | | 10 |
|  |  | 3.4.1 Functional Requirement | 10 |
|  |  | 3.4.2 Non- functional Requirements | 10 |
|  | 3.5 Feasibility Study | | 11 |
|  |  | 3.5.1 Economically Feasibility | 11 |
|  |  | 3.5.2 Testing Feasibility | 11 |
|  |  | 3.5.3 Behavioral Feasibility | 12 |
|  |  | 3.5.4 Operational Feasibility | 12 |
|  | 3.6 System Design | | 12 |
|  |  | 3.6.1 Registration Module | 13 |
|  |  | 3.6.2 Intranet Module | 18 |
|  |  | 3.6.3 Notice Board Module | 23 |
|  | 3.7 Database Design | | 27 |
| **4. Implementation and Result** | | | **29** |
|  | 4.1 Implementation | | 29 |
|  | 4.2 Testing | | 32 |
| **5. Conclusion and Future Enhancement** | | | **34** |
|  | 5.1 Objectives Achieved | | 34 |
|  | 5.2 Conclusion | | 34 |
|  | 5.3 Future Enhancement | | 35 |
| **References** | | | **36** |

**List of Figures**

**\_\_\_\_\_\_\_\_\_\_\_\_**

|  |  |  |
| --- | --- | --- |
| **S. No** | **Content** | **Page no** |
| **Fig 1** | Modules of College Commune | 4 |
| **Fig 2** | E-R Diagram |  |
| **Fig 3** | Use case Diagram: Registration Module | 13 |
| **Fig 4** | Activity Diagram of Place Request | 14 |
| **Fig 5** | Activity Diagram of Approve Request | 15 |
| **Fig 6** | Activity Diagram of Account Generation | 16 |
| **Fig 7** | Registration Module: Level 0 DFD Diagram | 17 |
| **Fig 8** | Registration Module: Level 1 DFD Diagram | 17 |
| **Fig 9** | Registration Module: Level 2 DFD Diagram | 18 |
| **Fig 10** | Intranet Module: Use Case Diagram | 19 |
| **Fig 11** | Intranet Module DFD level 0 | 20 |
| **Fig 12** | Intranet Module DFD level 1 | 21 |
| **Fig 13** | Intranet Module DFD level 2 | 22 |
| **Fig 14** | Notice Board Module: Use Case Diagram | 23 |
| **Fig 15** | Notice Board Module DFD level 0 | 24 |
| **Fig 16** | Notice Board Module DFD level 1 | 25 |
| **Fig 17** | Notice Board Module DFD level 2 | 26 |

**3. ANALYSIS AND DESIGN**

During the development of the application we would be using Spiral Model as the Software Development Life Cycle.

* It consists of following phases:
* Requirement analysis and gathering
* Designing
* Implementation
* Coding
* Testing
* Deployment
  + Maintenance

**3.1 Requirement analysis and gathering:**

* The basic requirement of the application is a device having **Android** as the operating system 4.0 or later version, My**SQL** as its local back-end.
* Web Services to be used by web and Mobile client.
* Intranet Read/Write Access for Intranet Client.

**3.2 System Specification:**

* Technology Used:- Android
* Language:- Java,HTML5
* Framework used: NetBeans
* Backend:-MySQL, SQLite
  1. **System Requirement:**
* **Device**:-Smart phone and/or Web browser
* **Operating System**:-Android 4.0 or later (for smartphone)

**3.4 Functional/Non-functional Requirements:**

This application gives the list of functional and non-functional requirements that are applicable to the “College Commune” system.

**3.4.1 Functional Requirement:**

Departmental Administrator (DA):

* Authentication: New Students request must be approved by DA .
* Add User: New user must enter the account details to Get In.
* Suspend User: DA can suspend any user if he feels that user is not valid.
* Add faculty user: DA will approve faculties.

Faculty:

* Change Password: faculty can change password of his/her account.
* Generate Notice: Faculty can generate notice and then put it on the notice board to be available for Target Student.

Student:

* Request for new Account: Student will request to DA for new account.
* Sign up to create account: Student will sign up for his account.
* Check out the profile.

**3.4.2 Non-functional Requirement:**

* Safety Requirement:

The database gets crashed at any certain time due to virus or operating system failure. Therefore, it is required to take the database backup.

* Security Requirement:

We are going to develop a secured database for the college. There are different categories of the user’s namely departmental admin, faculty and student. Depending upon the category of user the access right is decided. It means if the user is a DA then he can be able to modify, delete and append the data. The faculty has right to access some details of student account.

* Software Quality Attributes:

The quality of the database is maintained in such a way so that it can be very user friendly to all the user of database.

**3.5 Feasibility Study:**

**3.5.1 Economically Feasibility**

The system being developed is economic with respect to School or Collage’s point of view. As a web server required, this is available at reasonable rate.

The development framework is open source so no license fees to be paid.

The live testing is possible on rightly available Smartphones leads to cut testing costs.

The Internet Connection is provided by institution.

**3.5.2 Technical Feasibility:**

Technical feasibility tries to answer the following question to make this software feasible to develop:

* The software or tools necessary for building and running the application are easily available or not?
* The compatibility amongst software exists or not?
* Are developers aware of these technologies?

So we found answers of the above question, as given below

* The hardware require for our application was already available with us.
* The compatibility amongst software is existing as it’s developed in Net Beans and also develops in Android.
* A developer of this developing application knows the language and tries to learn the new technology.

**3.5.3 Behavioral Feasibility:**

The system working is quite easy to use and learn due to its simple but attractive interface. User requires no special training for operating the system.

**3.5.4 Operational feasibility:**

Operational feasibility measure how well the solution will work in the institute and improve communication and better user experience.

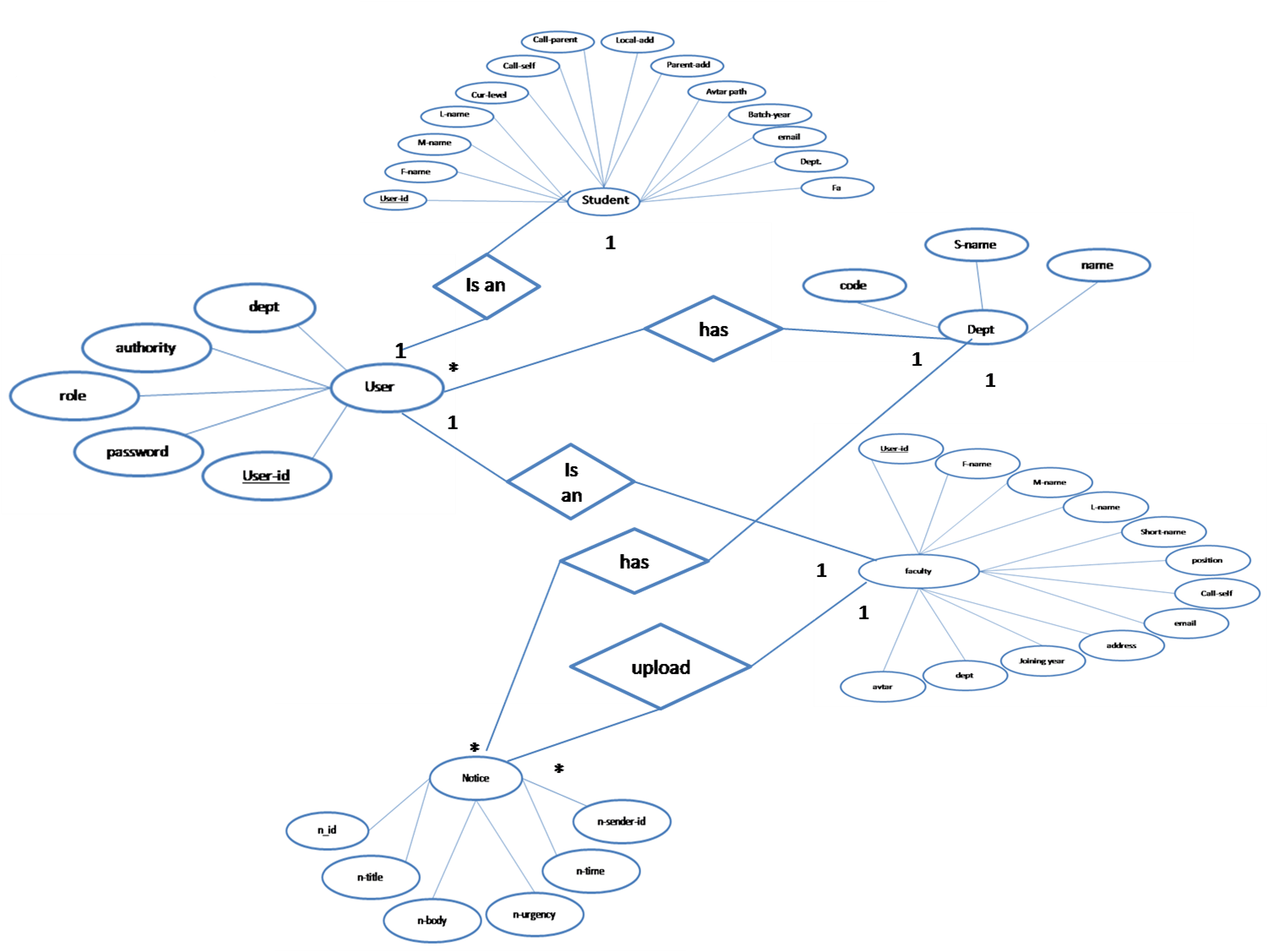
**3.6 System Design**

* The designing phase consists of constructing GUI of the application.
* It also consists of database design i.e. tables used in the application for different modules which stores information.

The basic way of designing the software or application is by drawing UML diagrams.

* The diagrams consists of:
* E-R Diagram
* Use Case Diagram
* Activity Diagram
* Data Flow Diagram
* Class Diagram

**E-R Diagram:**



**3.6.1 Registration Module:**

* Student and faculty send request DA to get register to the system.
* After DA’s authentication student and Faculty will get registered by providing some basic personal information. This information will be used for further use of system.
* *Like Name, Enrolment Number, Email, Contact number, Year (to avoid conflict detain student).*

**3.6.1.1 Use-Case Diagram:**

****

**Fig (2) Use case Diagram: Registration Module**

This diagram shows that to create a new account how student and faculty place request then Departmental Admin approves request and then user can create new account.

**3.6.1.2 Activity Diagram:**

****

**Fig (3) Activity Diagram of Place Request**

This diagram shows that as student enter his enrollment number and if it is valid it will place request according to that department.

****

**Fig (4) Activity Diagram of Approve Request**

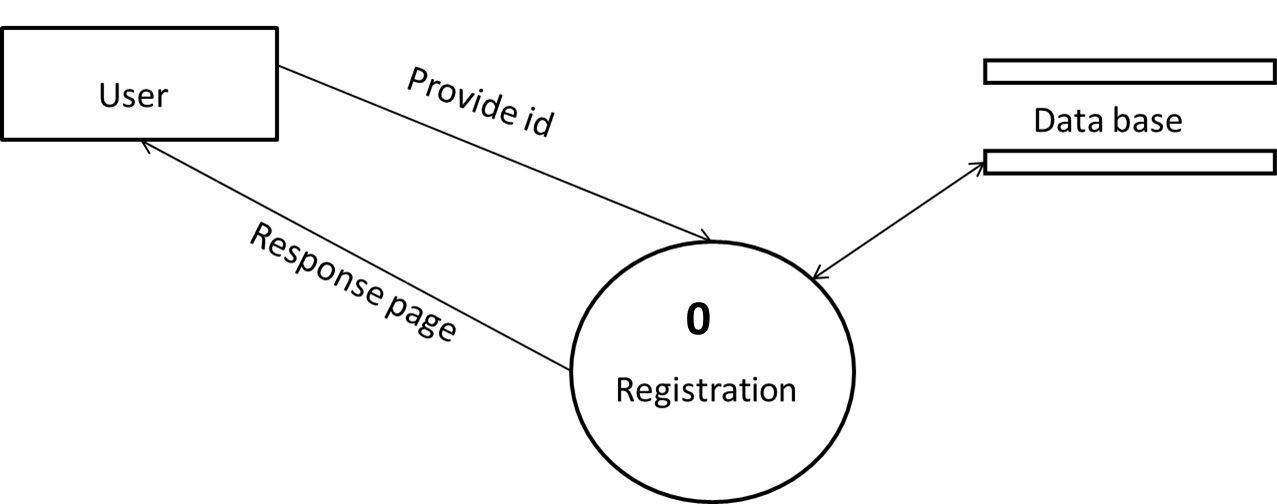
This diagram checks for the request from the department student if it is valid it will be approved by departmental admin.

****

**Fig (5) Activity Diagram of Account Generation**

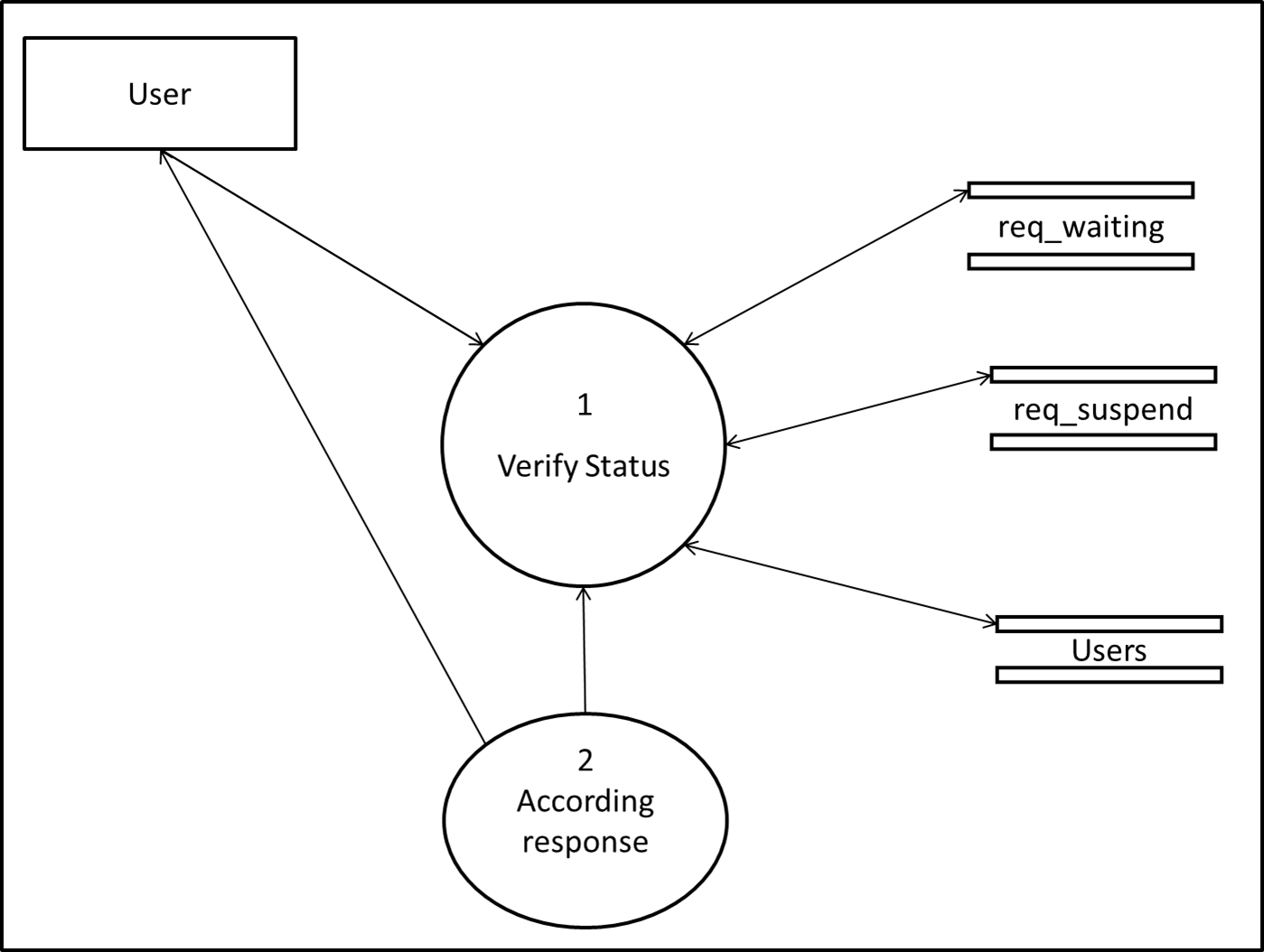
User will enter the enrollment in system. System will check the if the account request is approved or not. If approved, then System will prompt for the account details. Now system will create account based on the provided details.

**3.6.1.3 DFD Diagram:**

****

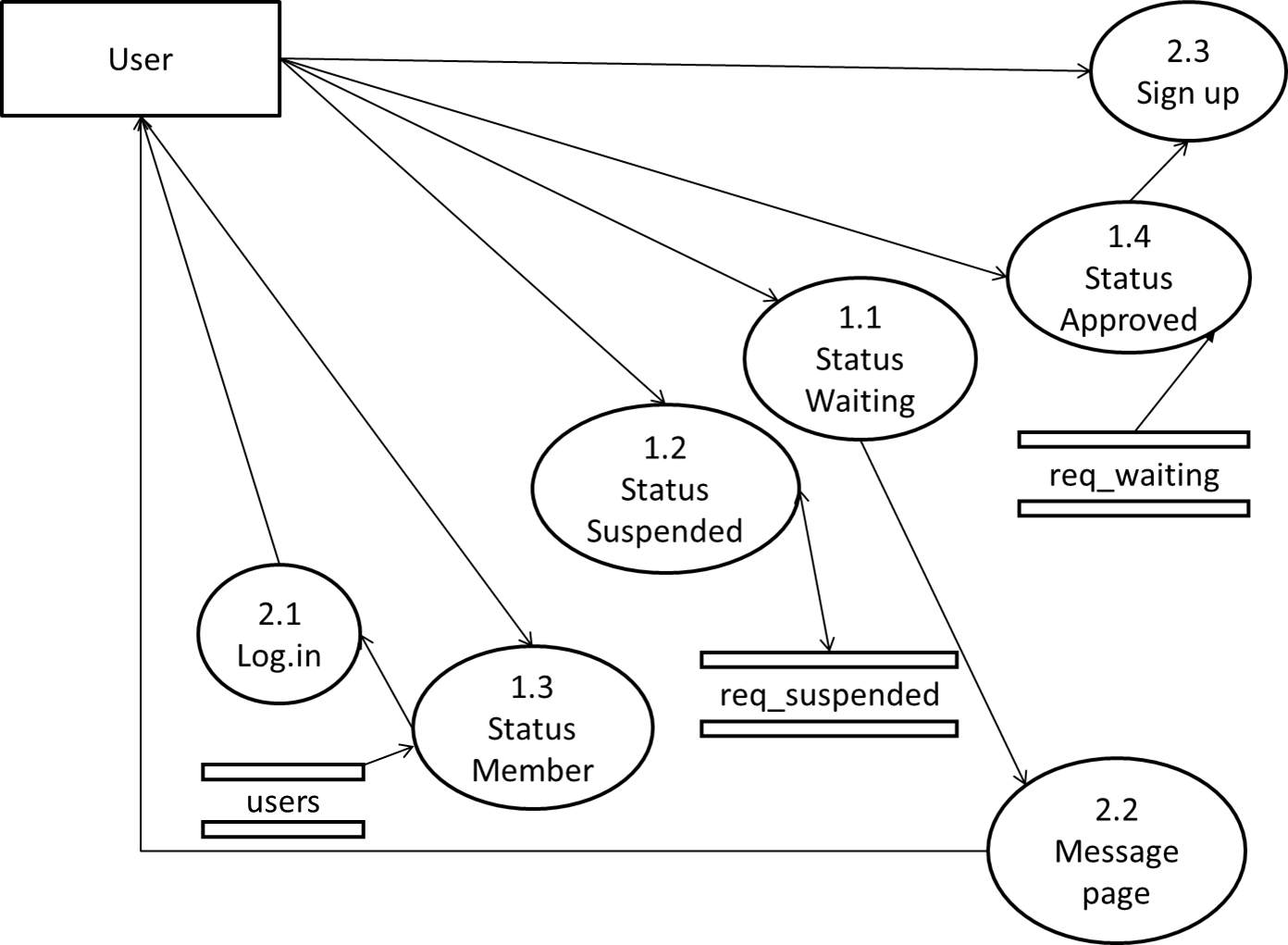
**Fig (6) Registration Module: DFD level 0**

User will give user\_id , based on that id all the further response will be given from system.

****

**Fig (7) Registration Module: DFD level 1**

As soon as user\_id will be entered , firstly status will be checked from relative databases. And according to the status, the forward services will offered.

****

**Fig (8) Registration Module: DFD level 2**

If user will new then he will be prompt to place request. If user will be member then system will ask for password to get login. If user in waiting or suspended then system will show message page.

**3.6.2 Intranet Module:**

Student and Faculty will get registered by providing some basic personal information. This information will be used for further use of system.

*Like Name, Enrolment Number, Email, Contact number, Year (to avoid conflict detain student).*

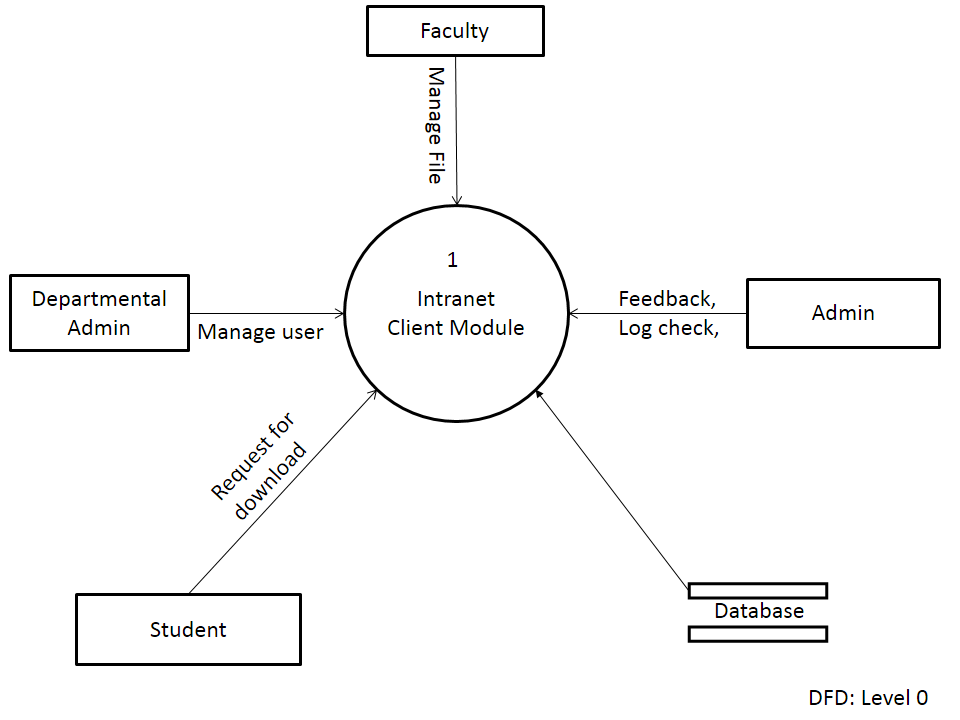
**3.6.2.1 Use Case Diagram:**



**Fig (9) Intranet Module: Use Case Diagram**

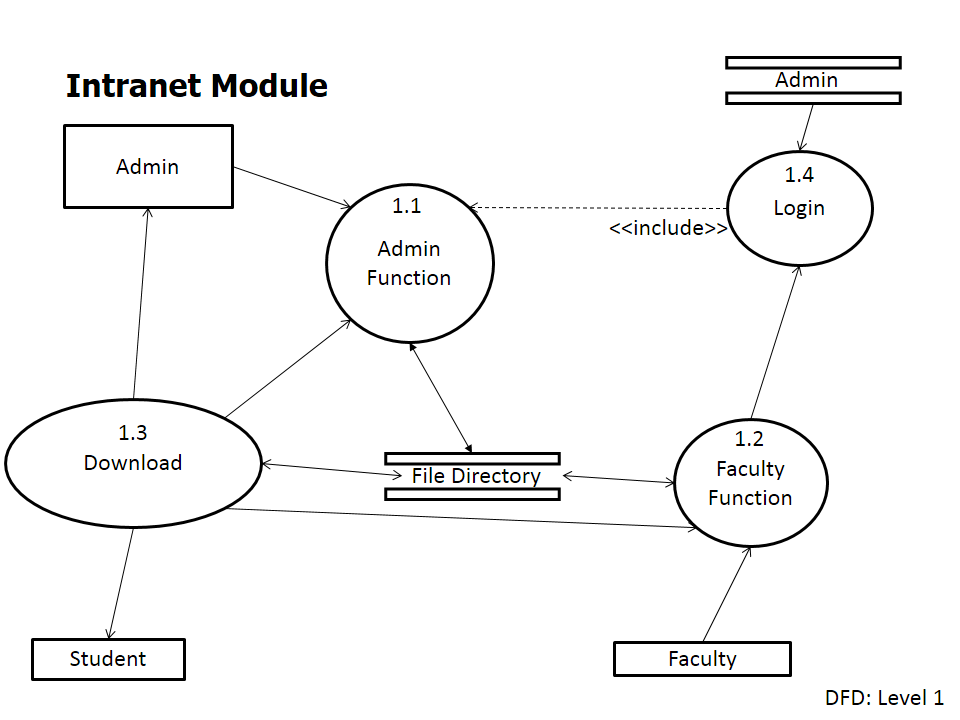
This figure describes the user and their activity with the system according to their role as Departmental admin, Student and Faculty. Each user has their own activity as per role given.

**3.6.2.2 DFD Diagram:**

****

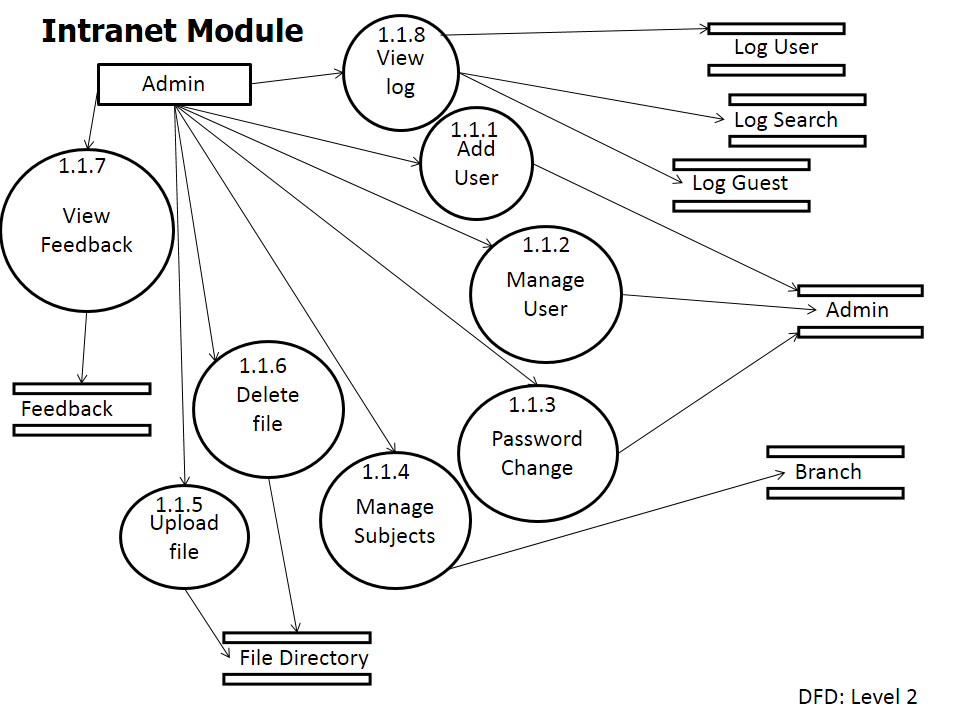
**Fig (10) Intranet Module DFD Level 0**

This figure describes the 0 level DFD diagram of Intranet Module. It shows that student and faculty requests for downloading documents and DA give feedback from database by allotted privileges and authentication.

****

**Fig (11) Intranet Module DFD level 1**

This figure shows process of intranet module in which first login is shown. Faculty has their functions like uploading or editing documents. Students can download the document.

****

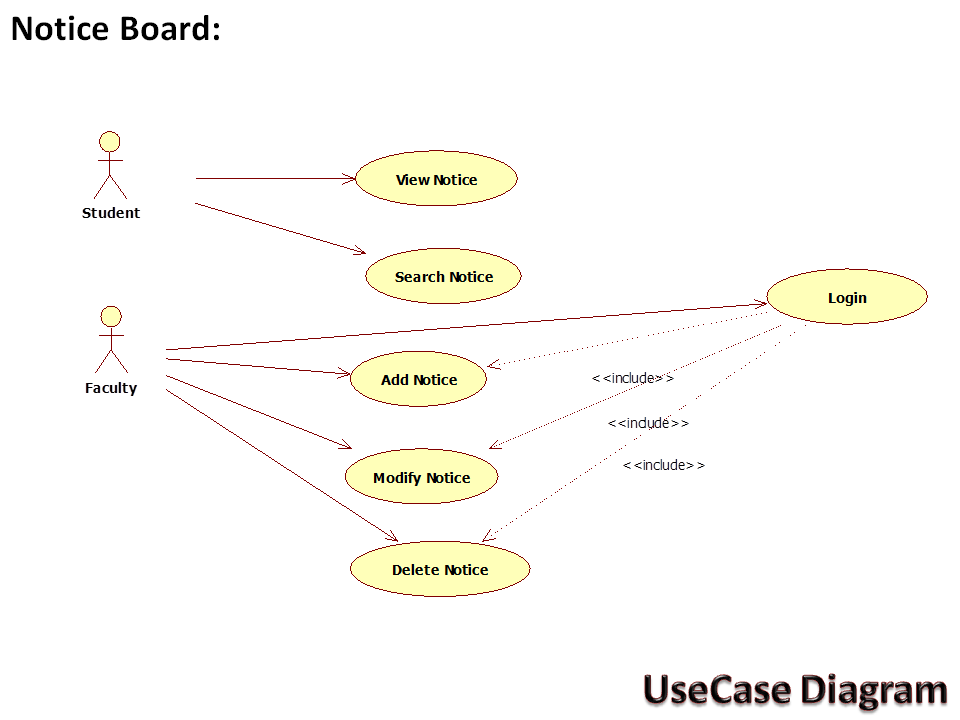
**Fig (12) Intranet Module DFD level 2**

This diagram shows function of admin in detail like add, manage, delete user; delete or upload file; manage log and give feedback, etc.

**3.6.3 Notice Board Module:**

* Any important NOTICE, announced by faculty or Student will be placed on Notice Board and the concern members will receive it.
* “E.g.,” *FA wants to inform all students about to receive the mark sheet then the particular class will be notified for that.*

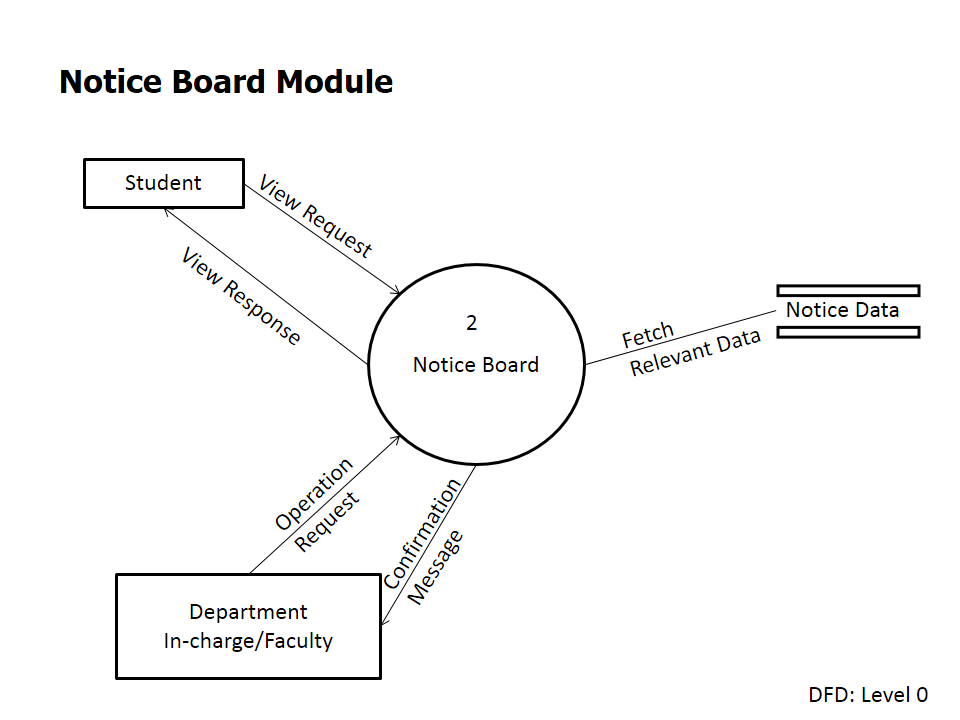
**3.6.3.1 Use Case Diagram**



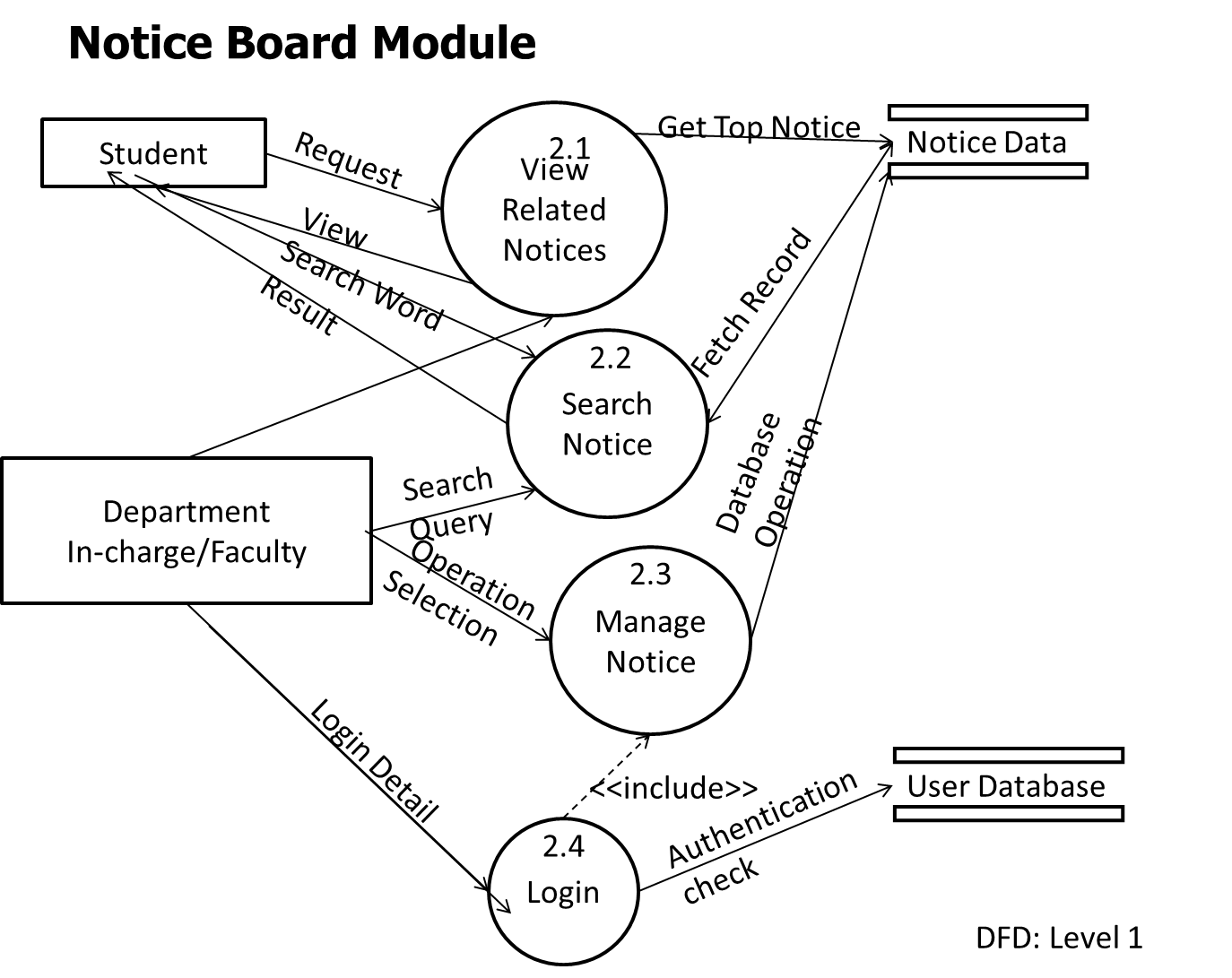
**Fig (13) Notice Board Module: Use Case Diagram**

This diagram shows the activity of student and faculty. Faculty can put the notice on notice board and student can view that notice.

**DFD Diagram:**

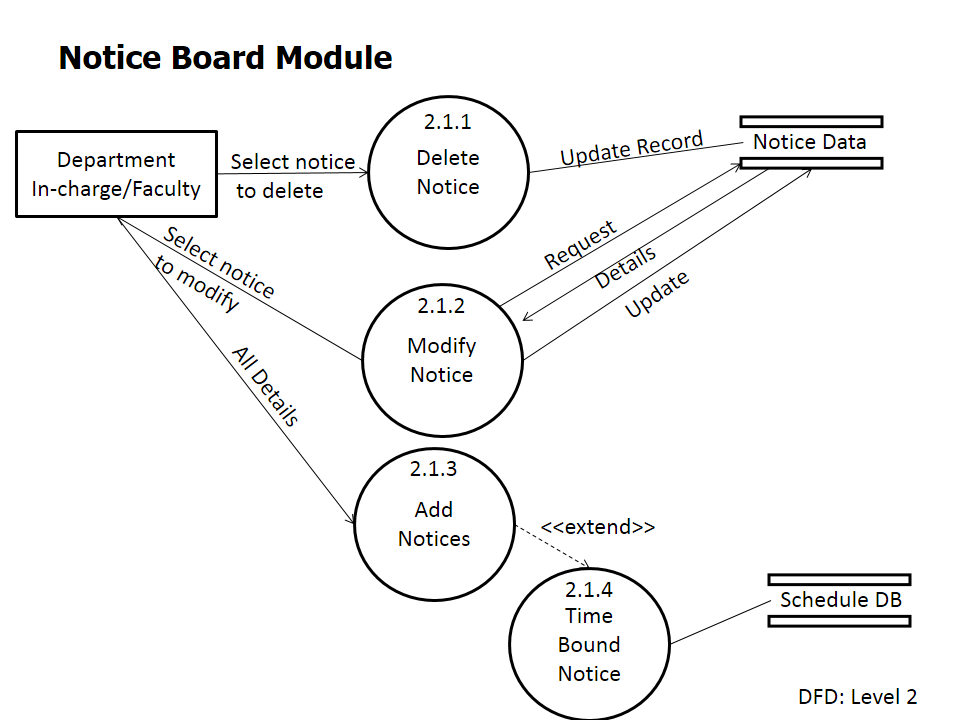
****

**Fig (14) Notice Board Module: DFD level 0**

This DFD diagram shows how this module will accept, access and provide notice to/by the external body say users.****

**Fig (15) Notice Board Module: DFD level 1**

This diagram shows how student and faculty will search, view and manages the notices. According to time of notice generation it will be stacked into the database.

****

**Fig (16) Notice Board Module: DFD level 2**

This diagram shows the modifications that can be apply on notification if required by DA. Extension to the notice some time bound notice can also be provided.

**3.7 Database Design:**

1. **dept\_table :**

|  |  |
| --- | --- |
| **Field** | **Data type** |
| Code | VARCHAR(45) |
| s\_name | VARCHAR(7) |
| name | VARCHAR(45) |

1. **faculty table:**

|  |  |
| --- | --- |
| **Field** | **Data type** |
| user\_id | VARCHAR(45) |
| f\_name | VARCHAR(45) |
| m\_name | VARCHAR(45) |
| l\_name | VARCHAR(45) |
| short\_name | VARCHAR(4) |
| Position | VARCHAR(20) |
| cell\_self | VARCHAR(12) |
| Email | VARCHAR(45) |
| Address | VARCHAR(45) |
| joining\_year | INTEGER |
| Dept | VARCHAR(45) |
| avatar\_path | VARCHAR(200) |

**(3) faculty\_association:**

|  |  |
| --- | --- |
| **Field** | **Datatype** |
| faculty\_id | VARCHAR(45) |
| branch | VARCHAR(45) |
| year | VARCHAR(45) |

**(4) req\_suspended table:**

|  |  |
| --- | --- |
| **Field** | **Data type** |
| user\_id | VARCHAR(45) |
| f\_name | VARCHAR(45) |
| l\_name | VARCHAR(45) |
| dept | VARCHAR(45) |

**(5) req\_waiting table:**

|  |  |
| --- | --- |
| **Field** | **Data type** |
| user\_id | VARCHAR(45) |
| f\_name | VARCHAR(45) |
| l\_name | VARCHAR(45) |
| dept | VARCHAR(45) |
| approved | VARCHAR(5) |
| role | VARCHAR(15) |

**(6) student table:**

|  |  |
| --- | --- |
| **Field** | **Data type** |
| user\_id | VARCHAR(45) |
| f\_name | VARCHAR(45) |
| m\_name | VARCHAR(45) |
| l\_name | VARCHAR(45) |
| current\_level | INTEGER(11) |
| cell\_self | VARCHAR(12) |
| cell\_parent | VARCHAR(12) |
| local\_add | VARCHAR(100) |
| per\_add | VARCHAR(100) |
| avatar\_path | VARCHAR(200) |
| batch\_year | INTEGER(11) |
| email | VARCHAR(45) |
| dept | VARCHAR(45) |
| fa | VARCHAR(45) |

**(7) users table:**

|  |  |
| --- | --- |
| **Field** | **Datatype** |
| user\_id | VARCHAR(45) |
| password | VARCHAR(45) |
| role | VARCHAR(45) |
| authority | VARCHAR(45) |
| Dept | VARCHAR(45) |

**(8) notice table:**

|  |  |
| --- | --- |
| **Field** | **Datatype** |
| n\_id | INTEGER(45) |
| n\_title | VARCHAR(100) |
| n\_body | TEXT |
| n\_sender\_id | VARCHAR(45) |
| n\_urgency | INTEGER(1) |
| n\_time | DATETIME |

**(9) notice\_asso table:**

|  |  |
| --- | --- |
| Field | Datatype |
| na\_id | INTEGER(11) |
| na\_branch | VARCHAR(45) |
| na\_level | VARCHAR(45) |
| na\_enroll | VARCHAR(45) |

**4. IMPLEMENTATION AND RESULTS**

**4.1 Implementation**

* Up to the stage, the work done over the Web application using java language.
* Registration module in which a student can generate account with security.
* Totally MVC (3 tire) architecture is followed.
* View: Developed java server pages (jsp) for dynamic view to end user.
* Controller: Servlet written to control the flow of activity.
* Model: The pojo classes with the related database level methods.

**4.1.1 Implementation of Registration module (screen shots):**

4.1.1.1 Home page (When user launches application)

**HOME PAGE(JM1\_STARTUP.JSP)**

4.1.1.2 New user (to place the account request)

**NEW USER (S1\_USER\_CHECK)**

4.1.1.3 Request Confirmation (Acknowledgement of the request)

**S1\_VERIFY\_NEW**

4.1.1.4 Log in as Admin (by departmental admin id and password)

**S1\_USER\_CHECK**

4.1.1.5 Approve request (approve the pending request of Students)

4.1.1.6 Approve acknowledge (Shows latest approved acknowledge)

**S1\_LOGIN\_CHECK**

**APPROVED PAGE**

4.1.1.7 Sign up page (Approved account offered the page)

4.1.1.8 Account Generation acknowledge (when signs up

4.1.1.9 User Profile page (Details of Student)

**4.1.2 Implementation of Notice Board Module (screen shots):**

4.1.2.1 Add notice (faculty and DA can add notice)

4.1.2.2 Student View Notice (student can view only class notice )

4.1.2.3 Faculty View Notice (Faculty and DA can view class as well as departmental notice)

**4.2 Testing**

* Testing is a process of executing a program with the intent of finding an error. A successful test is one that uncovers a yet undiscovered error
* Testing demonstrates that system functions appear to be working according to specification, performance requirement appear to have been met. In addition to that the data collection is conducted, which provides a good indication of the system reliability and some indication of the system quality as a whole.

Testing can be described as follows:

**1) Unit Testing**

* Each and every class is checked individually for errors.
* Created the Testing classes for testing the Model level class and their functionalities.

**2) Integration Testing**

* While integrating all modules together some usual problems like database connectivity problem were encountered.
* Thus were corrected.
* We had followed this procedure of integration testing during whole project development cycle.

**3) Validation Testing**

* The newly created system was validated against all the requirement specifications that were presented during the analysis phase.
* Validation testing provides the final assurance that the system met all functional, behavioural and performance requirements.

**5. Innovation**

We have used GIT(German Imaging Technology) through github. It provides version control system.

**6. Future Enhancements and Conclusion**

**6.1 Objectives achieved:**

* To learn java technology.

We learnt OOP concept with respect to java.

Come to know about the various java libraries and provided methods.

* Study of web application development techniques.

Studied and developing the web application using java technology

* Register user to the system in secure way.

Implemented the registration way in which fake account probability and therefore security threats come to lower level.

* To learn Application development that follows MVC architecture.

The learning actual implementation of M-V-C is going parallel with implementation. The basic requirement and the gained ease of development is found really smart.

**6.2 Conclusion:**

The main objective of the project is to develop a system is to establish an active medium for sharing of information and scheduling among COLLEGE that give ease of communication.

The above objective in literal statement looks very good and useful, but it just turned out be an eventful beginning exploration of different module that dictate, decide and develop through the mobile application using android and web application using java language in total easy way of communication between faculty and student.

Up to this progress we come across so many experience and by means of them, we learn a lot that a book is not able to speak. Many alternative strategies comes in mind while developing but choosing best out of them is a skill.

**6.3 Future Enhancement:**

For this system many enhancement could be done which can help the users in prodigious way, they are as noted below:

* In ‘College Commune’ developing a web application of Registration and Noticed module. We are planning to develop a mobile application of all listed modules.
* As we are going to develop a mobile application on android supported phone we used HTML5, for that we learn how to make android application. This will result in newer application like
* Student Profile Management
* Class Canvas Module
* There should be a developed marks generation as per latest internal marking system which reduce time and managerial paper work which enhance productivity in less effort**.**
* We are building the POJO classes with the according methods in model view. Such classes will be more useful when the new module will be implemented. The testing classes will be expanded as new methods will be introduced.

**REFERENCES**

[1] Head First Servlet and JSP

[2] Complete reference JAVA, 7th edition.

[3] BEGIN ANDROID JOURNEY IN HOURS

[4] www.stackoverflow.com

[5] [www.w3schools.com](http://www.w3schools.com)

[6] www.github.com

[7] various google groups