



Review 1-First Report Winter 2017-18

MTECH-(SOFTWARE ENGINEERING)

**PROJECT BASED
COMPONENT**

SWE2004-SOFTWARE ARCHITECTURE AND DESIGN

Slot-D2

Chat Application

16MIS0257- V. SAI NIKHIL

16MIS0308- TANVI TABSHEEM.MD

22 – MARCH - 2018

FACULTY: Dr S.SREE DHARINYA

Detailed description about problem statement:

Chat application system is a Networking project which aim is connecting all the users of the system together. Particularly we are dealing with the users like Students and Faculties who can share all the contents that need to be delivered from the faculty to the student.

Teleconferencing or chatting is a method of using technology to bring people and ideas “together”. This project is an example of chat-server. It is made up of two applications one is

1)client application

Here client application is the main user who accesses this chatting application. In our project this user is Faculty and the student.

2)server application.

Here server application is the Admin’s application who can maintain and check the user’s actions through a networking concept.

To start the chatting or texting the messages both the Faculty and the Student must connect to the server where they can practice two kinds of chatting:

1) Public one

Here message is broadcasted to all the connected students.

2) Private one

Here message is broadcasted to only two users who.

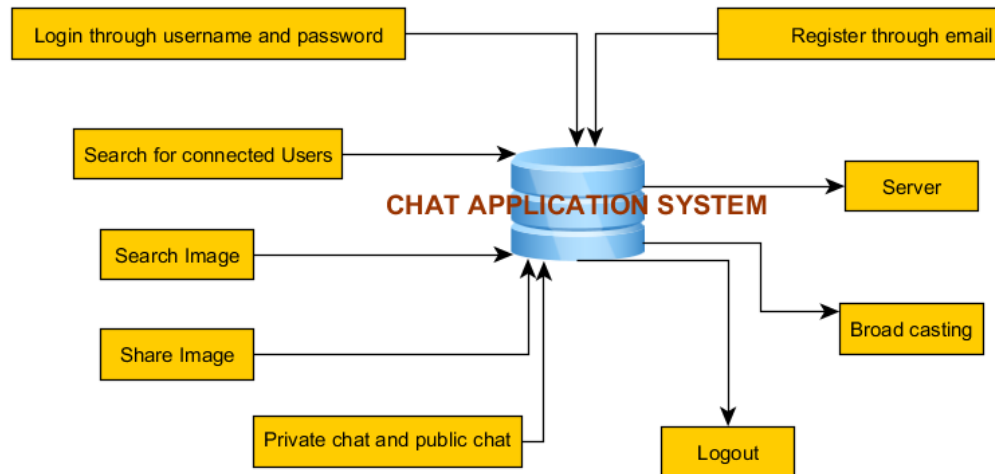
Security measures are taken in both the chatting Practices.

Chat Application System provides Register and Login to all the users that is to Faculty and Student through the **email and phone number**. Here the connected users firstly Register through the Email, then Login in order to connect to the server. After this login they search for other connected users, check their messages,

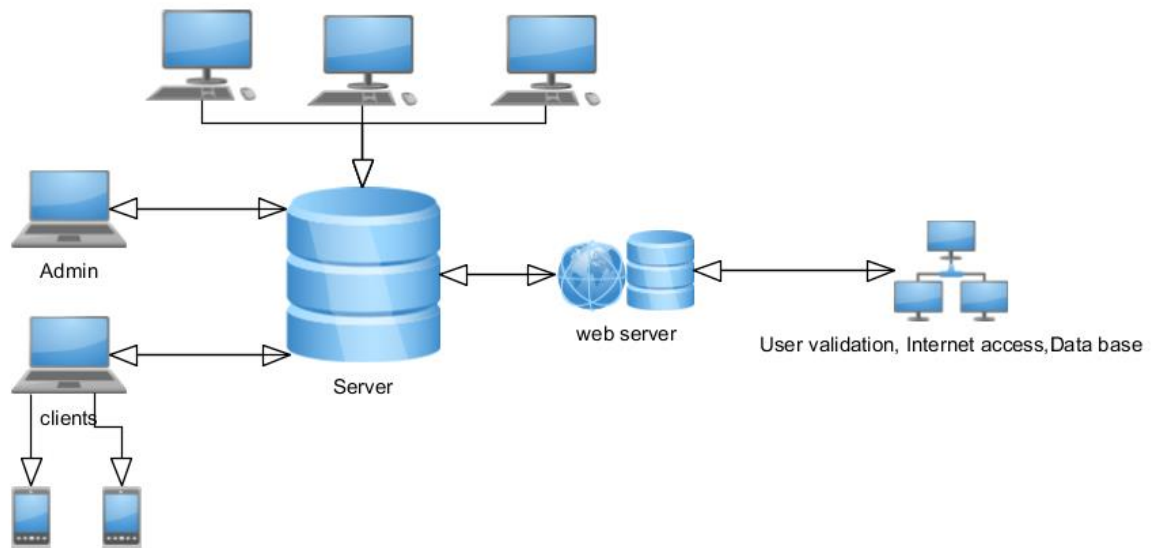
check the broad casting, chat for quires, search for other connected users through their User name or Phone Number, Reply to the broadcasted chat which are both in public and private, Share the Images through the Browse, Share the Images in the chat application.

Object Oriented Design

1.The context level diagram system:

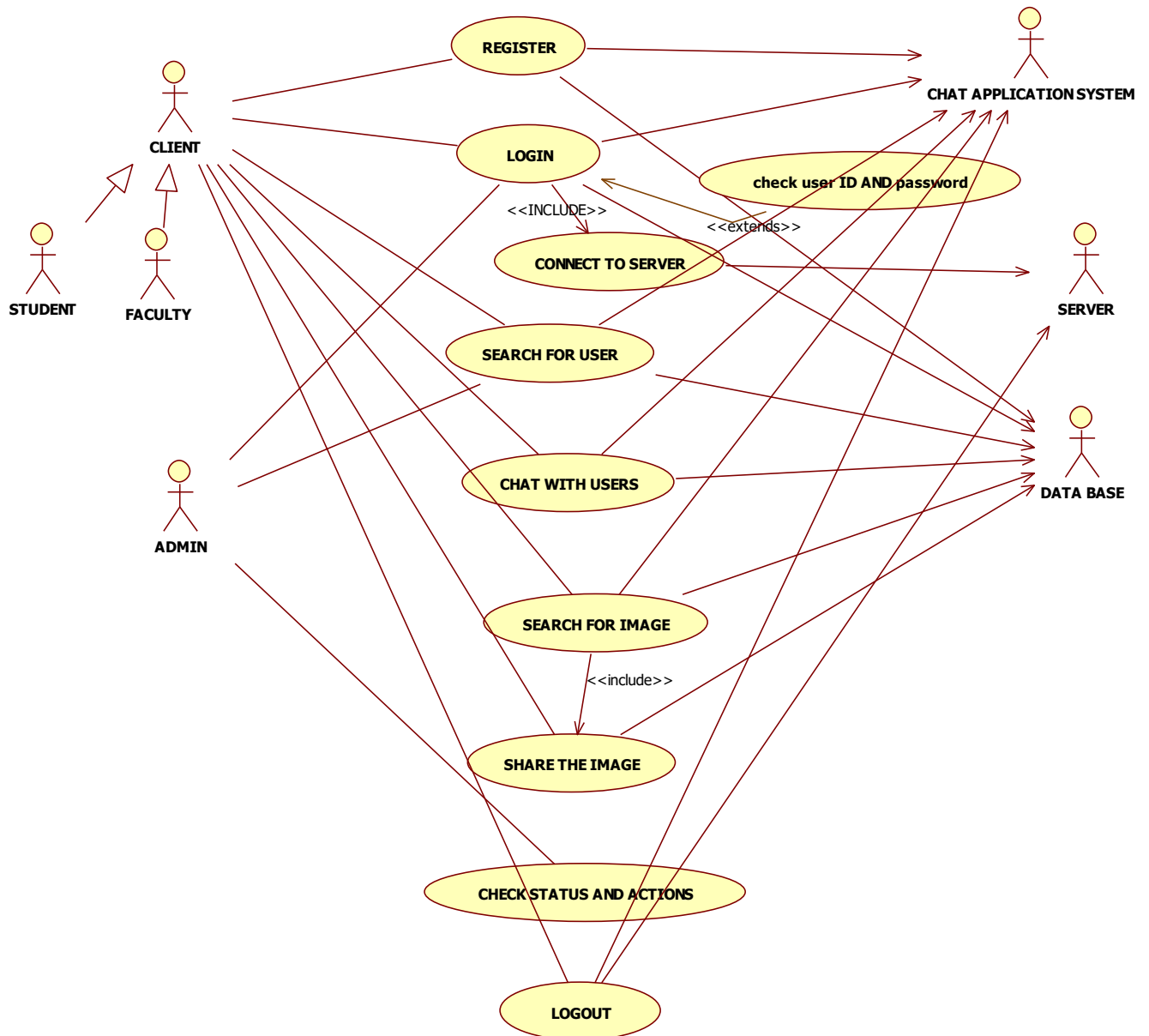


2. The system architecture



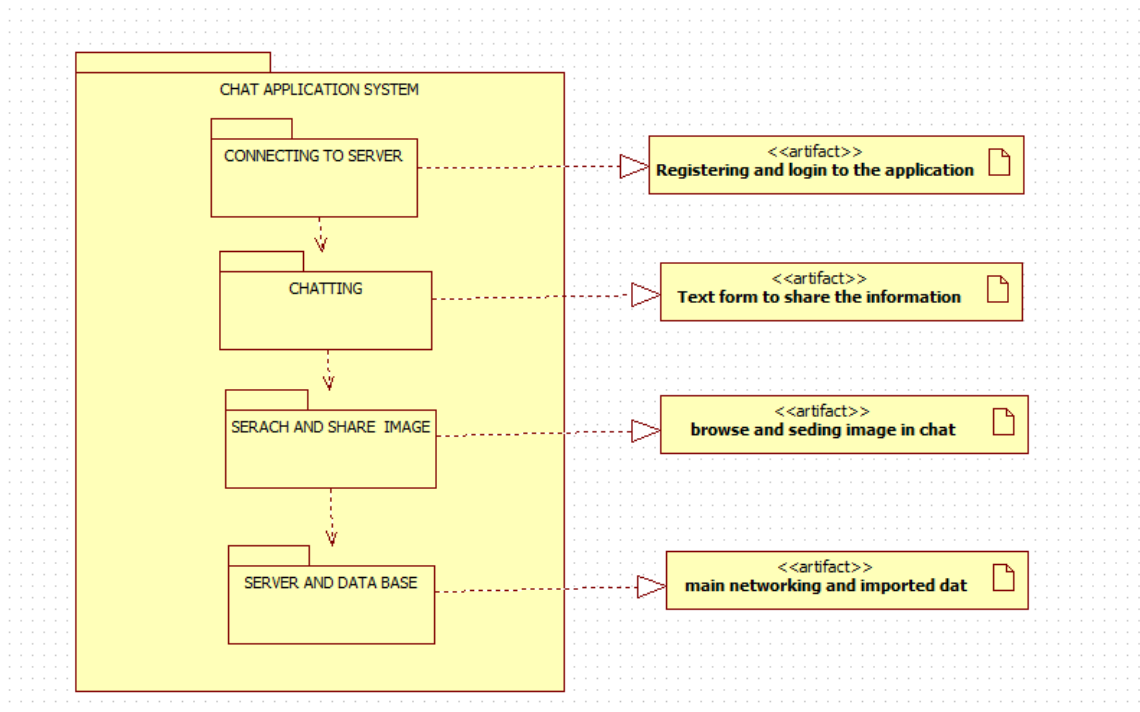
3. Identify the principal Sub-system and components

- Develop usecases

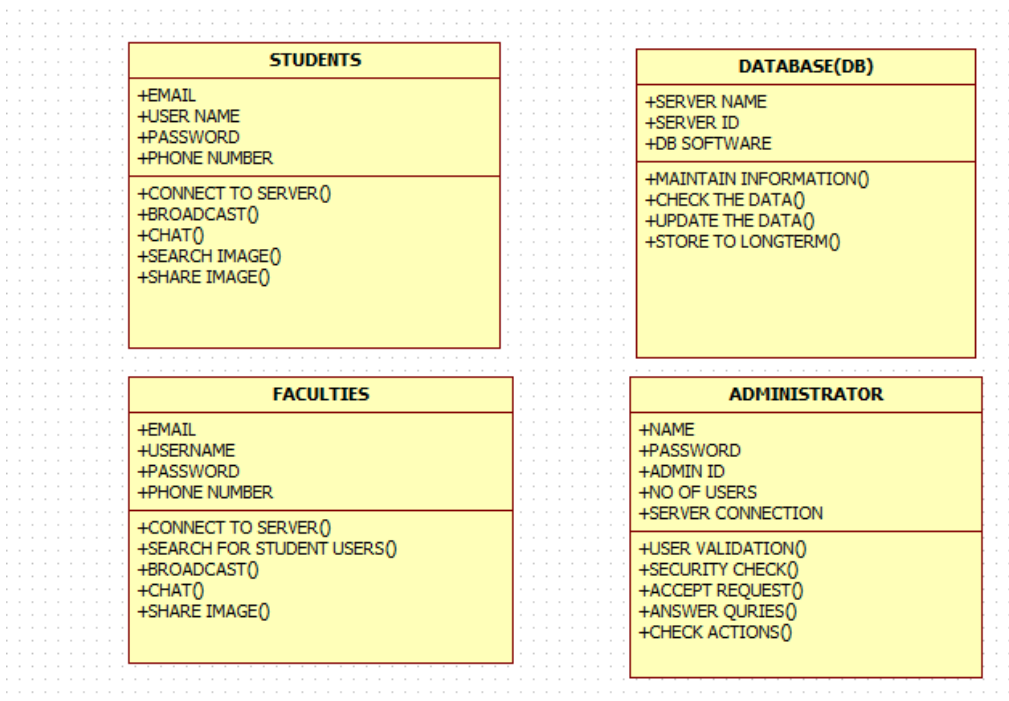


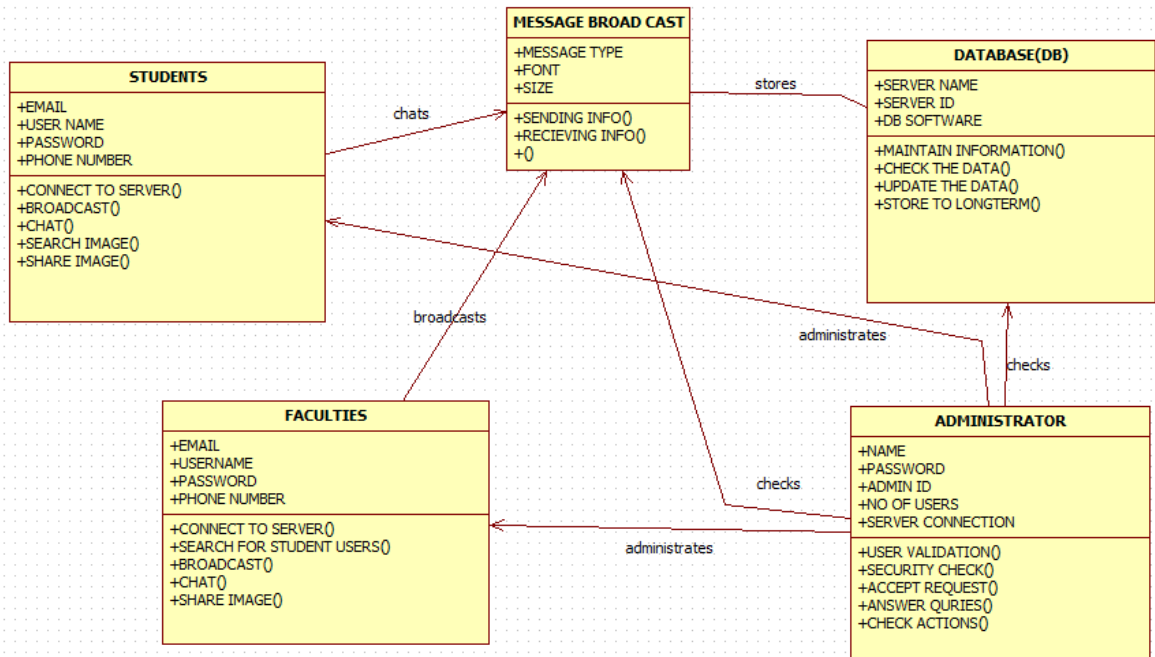
4. Developing design models

i. block diagram(architecture design)



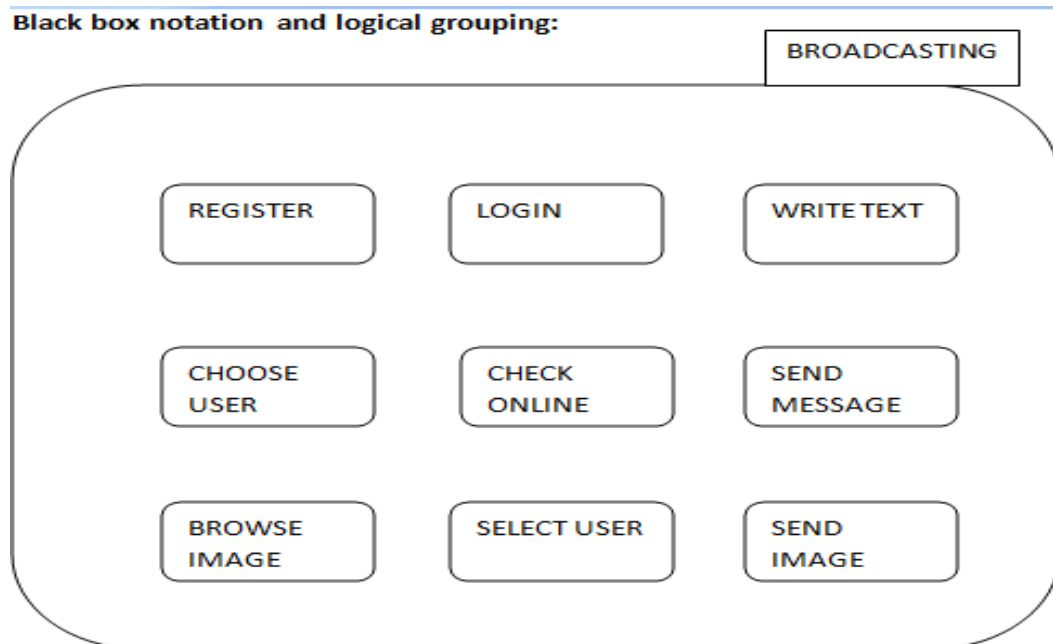
ii. Identify objects:

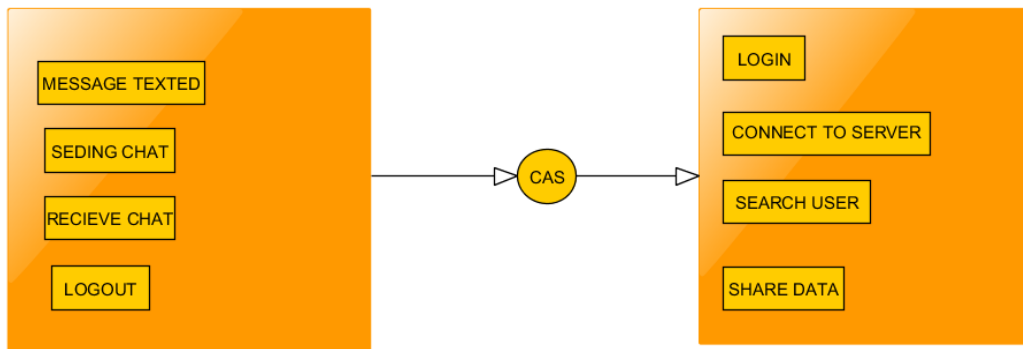




iii. Identify design model

Black box notation and logical grouping:



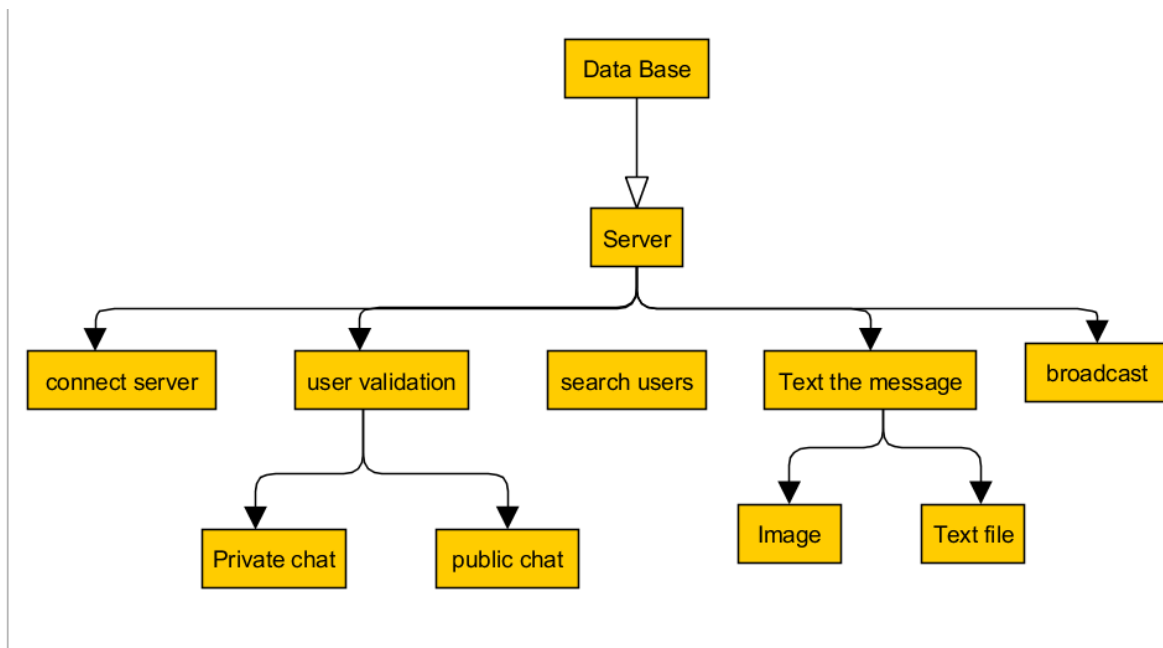


Structure diagram using JSP & JSD

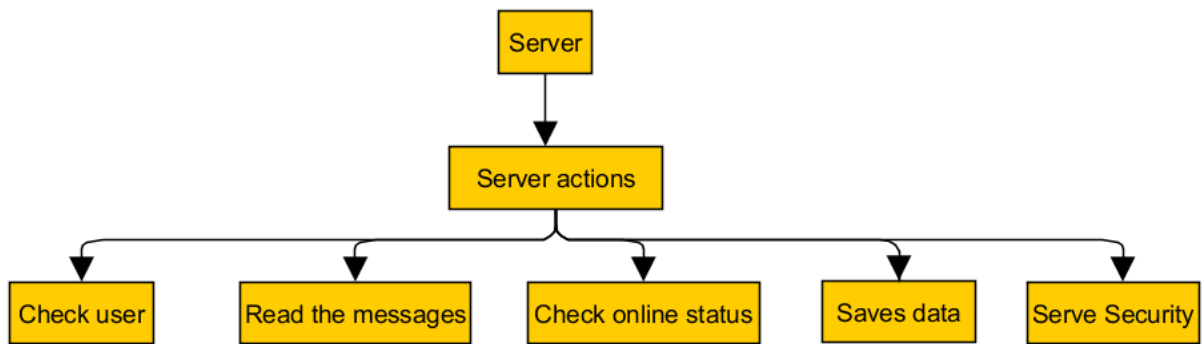
1.Broadcast process

JSP – Jackson Structured Programming

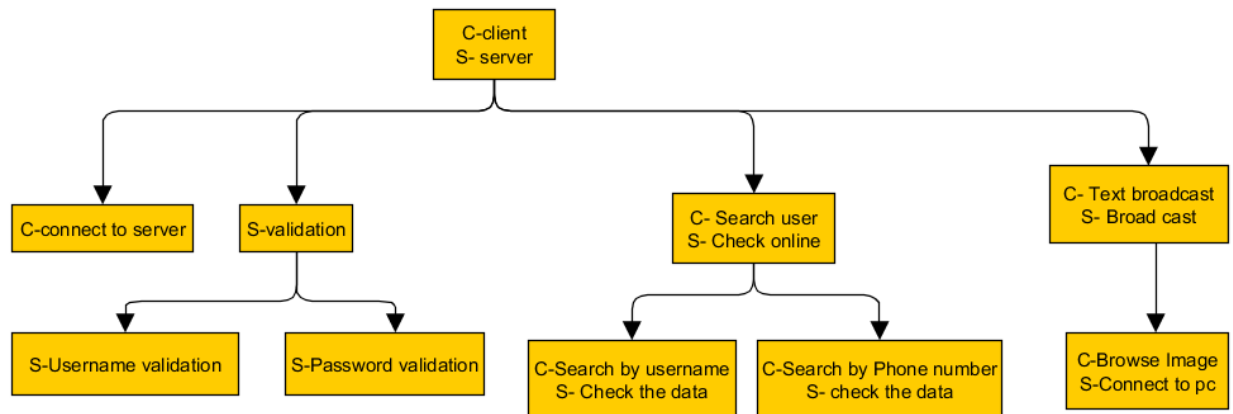
Step1: Input data Stream



OUTPUT STREAM:



Step2: Merge these to form the program Structure Diagram.



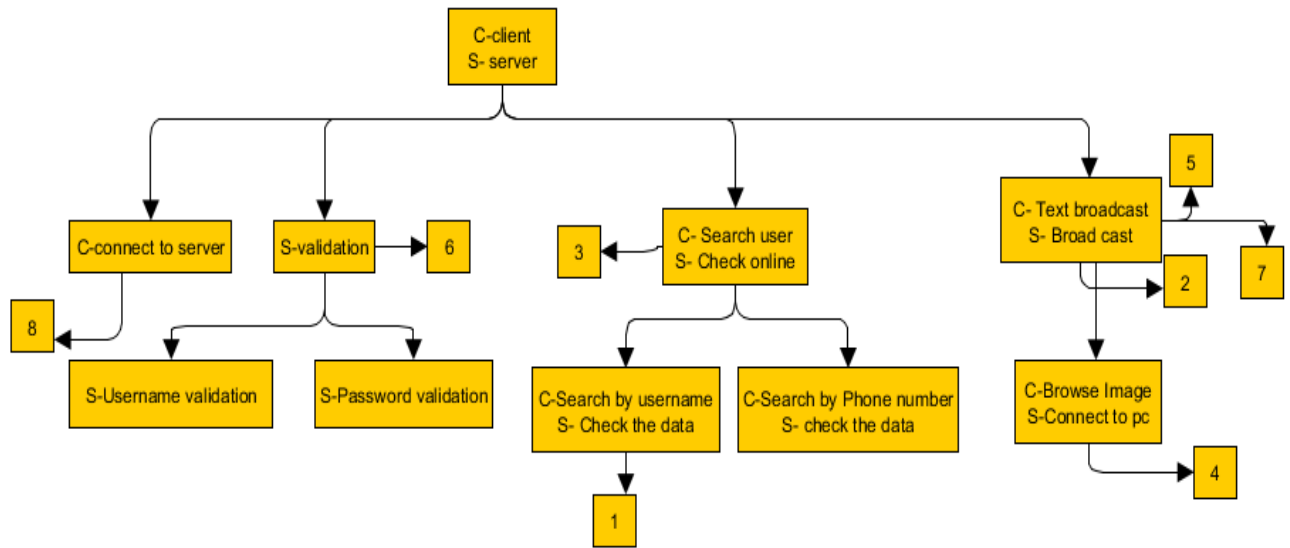
Step3: Listing the operations that need to be performed by the program, and allocating each operation to an element in the program Structure Diagram.

Inputs:

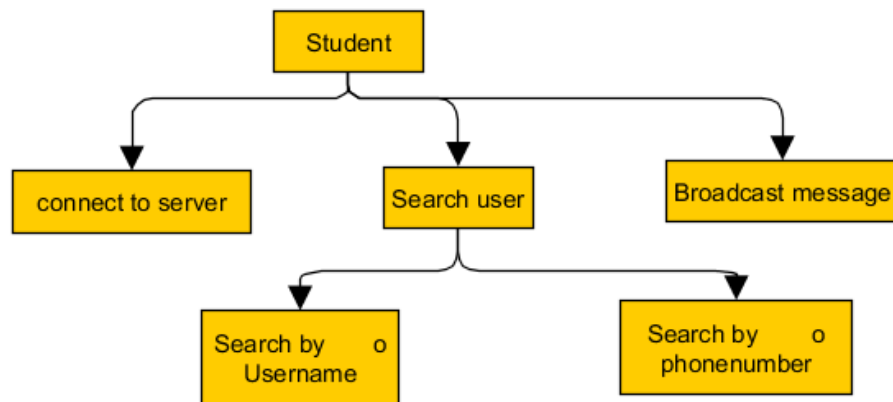
1. Check User by Username/Phone number
2. Text the message
3. View the online status
4. Browse image to share
5. Broad cast by private/public chat

Outputs:

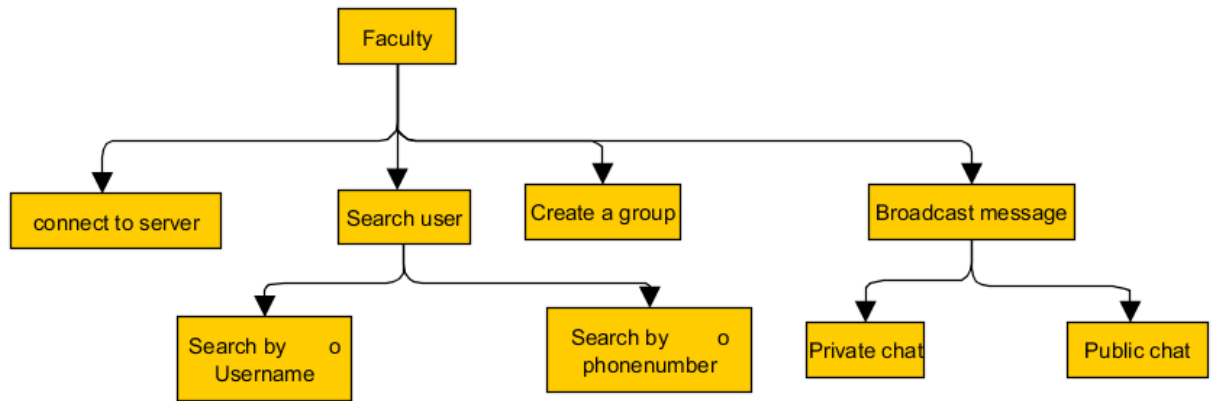
6. Register the user
7. Message gets broad casted
8. Give results for the search



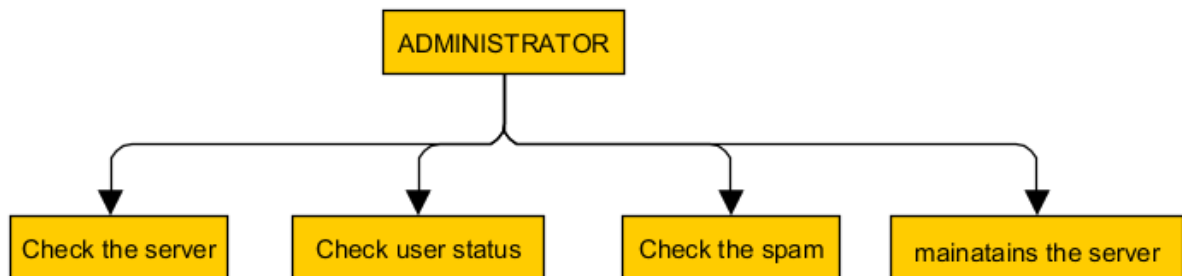
Entity: Student



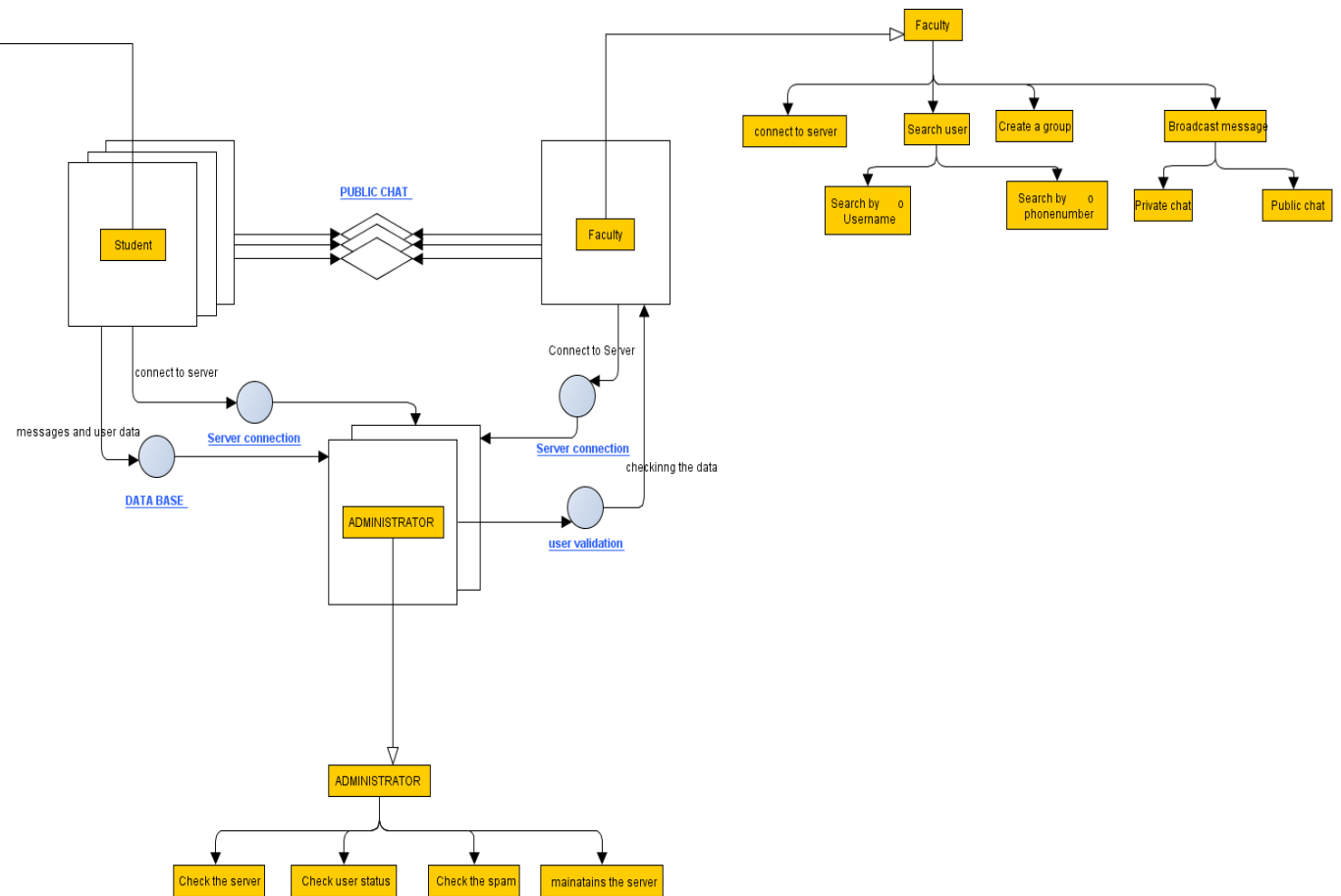
Entity: Faculty:



Entity: Administrator:

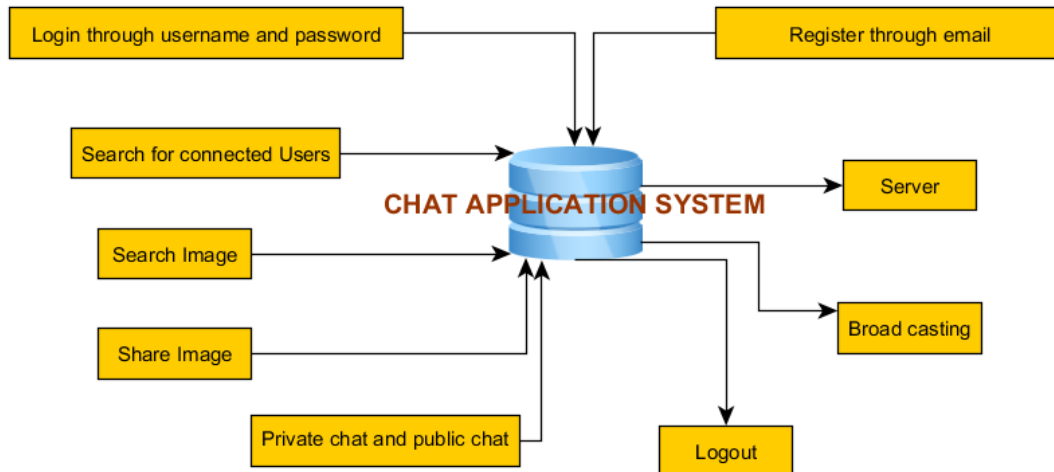


ESD+SSD= JSD

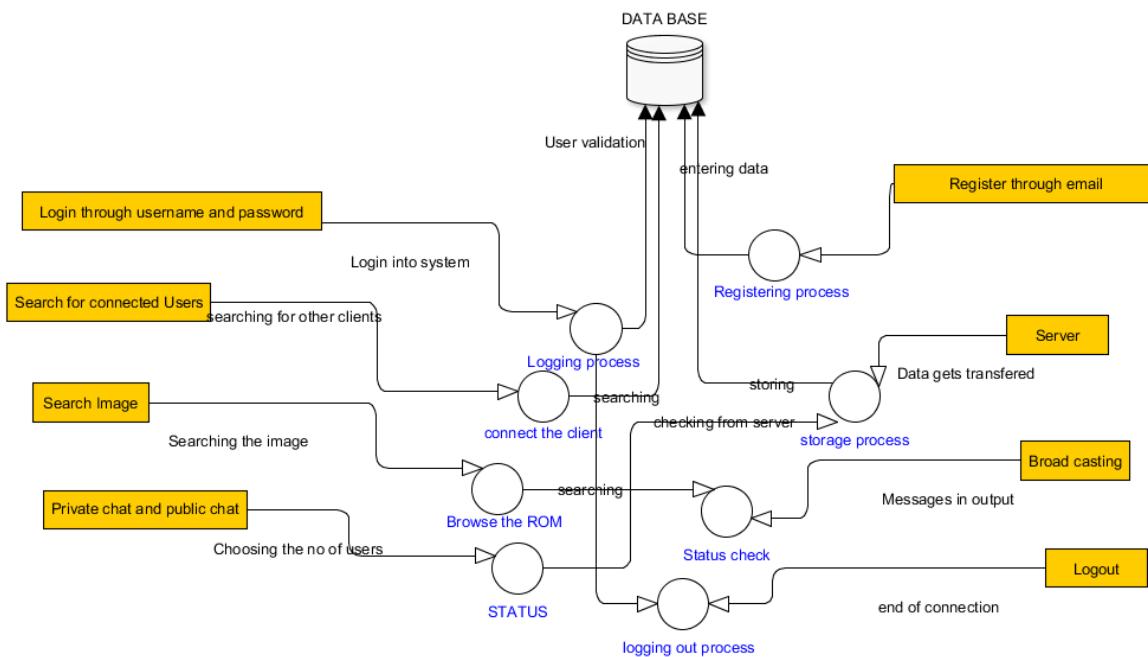


BROADCASTING ANALYSIS USING SSAD:

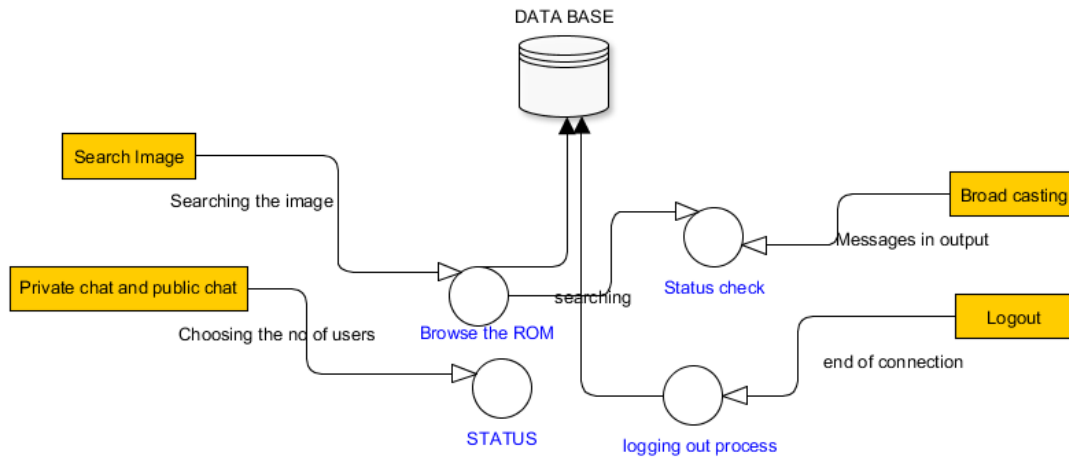
Level 0:



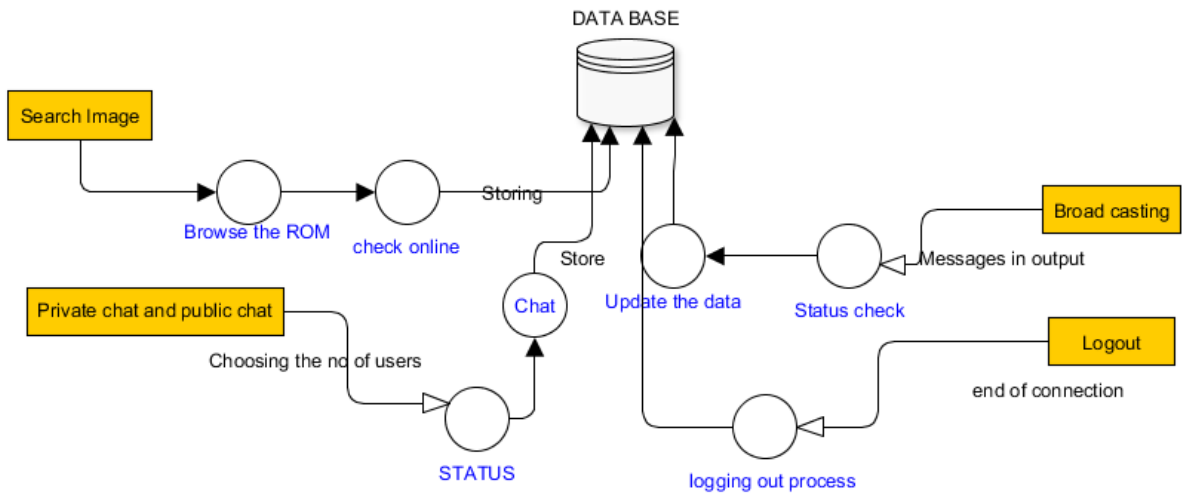
Level 1:



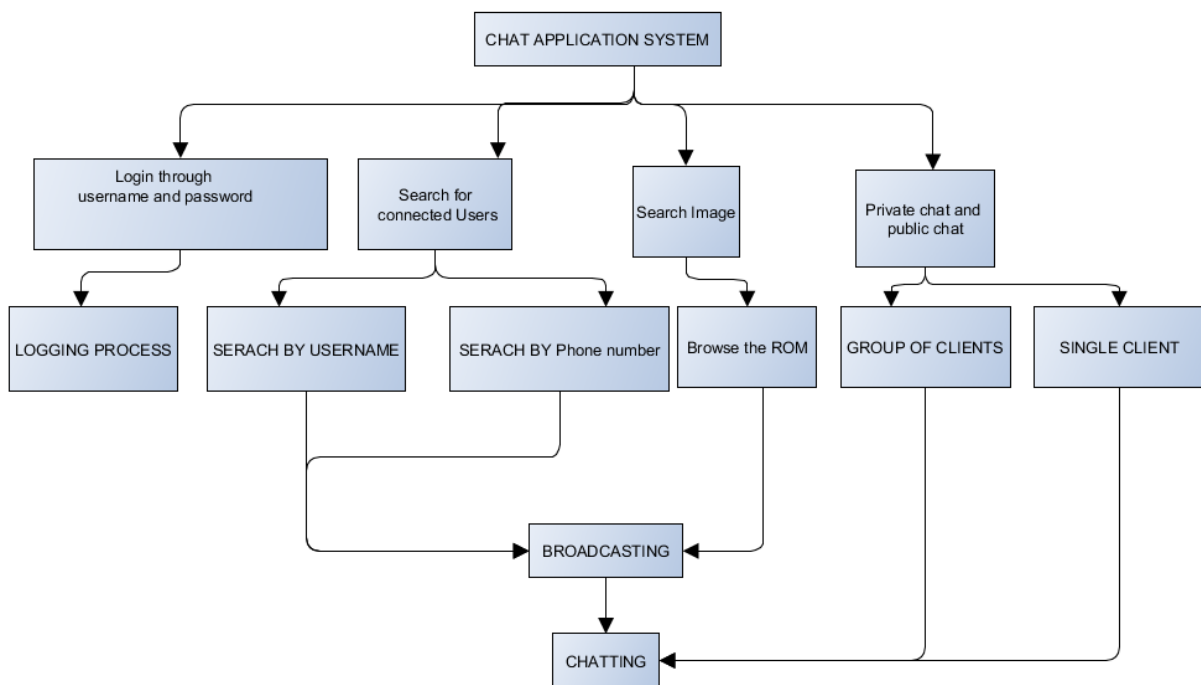
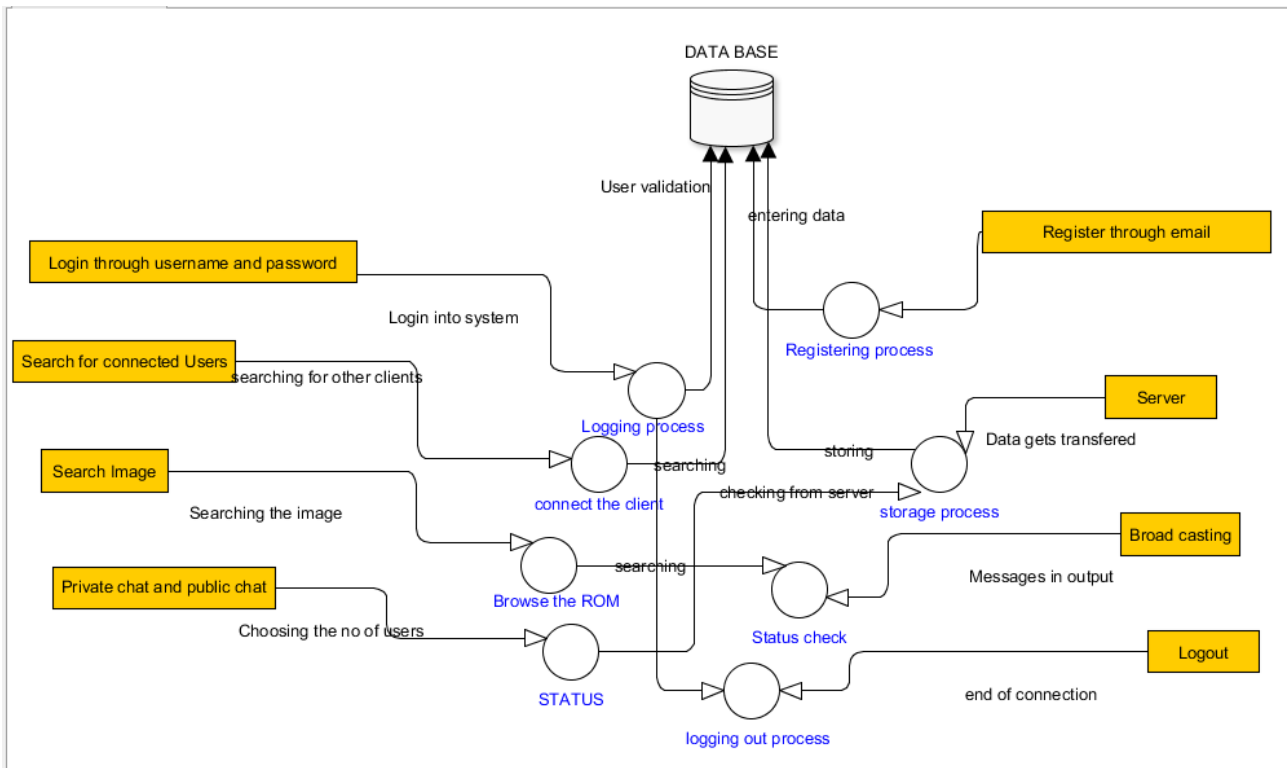
Level 2:



Level 3:



Mapping JSP with DFD





Review 2-Second Report Winter 2017-18

MTECH-(SOFTWARE ENGINEERING)

**PROJECT BASED
COMPONENT**

SWE2004-SOFTWARE ARCHITECTURE AND DESIGN

Slot-D2

Chat Application

16MIS0257- V. Sai Nikhil

16MIS0308- Tanvi

22 – MARCH - 2018

FACULTY: Dr S.SREE DHARINYA

Architectural Styles for Chat Application:

1. Repository architectural style
2. Layered architectural style
3. Client server architectural style
4. Pipe and filter architectural style
5. Process control architectural style

Repository Architectural Style

Components:

Components are the central database, the student and faculty database. The database must be maintained and independent from each other like private chat database, group chat database, and so on.

Connectors:

Relations among database which are independent components and operated by centralized database. Like faculty and student database are connected in private chat database to know who is texting and their details.

Topology:

Star Topology

Description:

In chat application we do have many databases. Few of them are student database, faculty database, private chat database, group chat database, broadcast database, and so on.

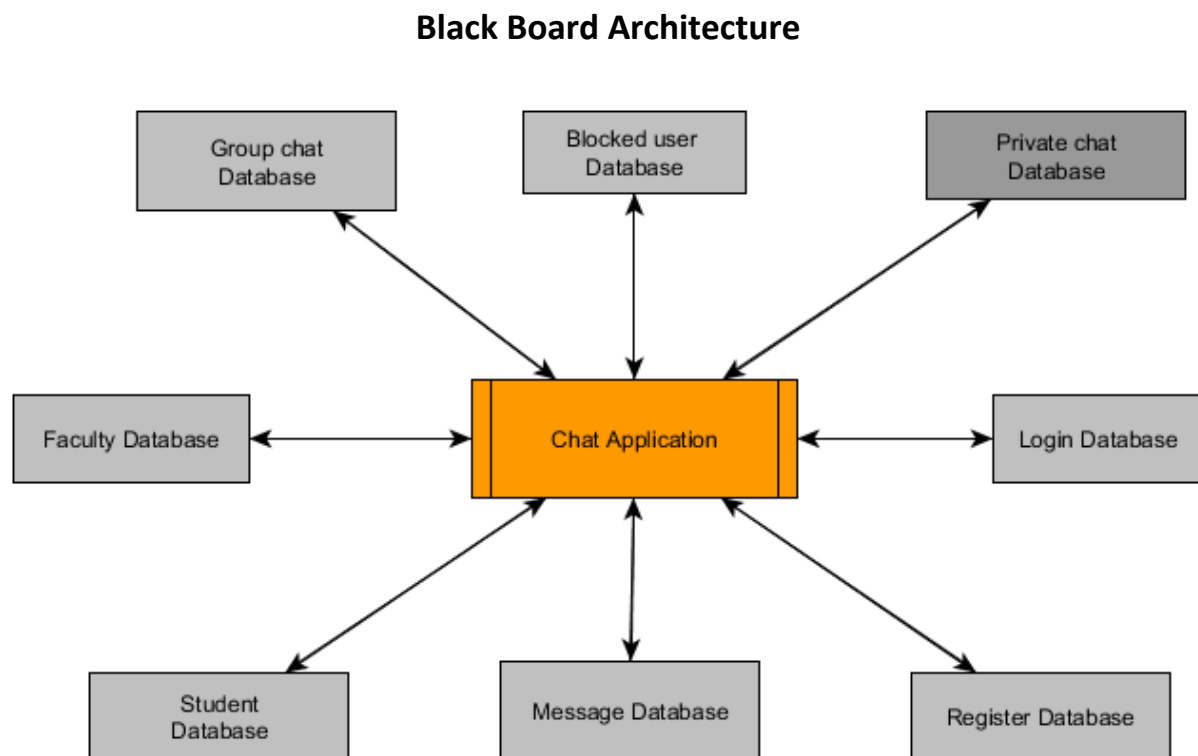
These databases are maintained by single central database to maintain and manage the chat application and maintain user.

Each system or module has its own database. The central database manages the data in group chat as well as in private chat. The messages as well as the details of the student and faculty must be managed so that student can know the basic details of faculty like the name and the subject that they teach.

Faculty can also view the name as well as registration number of the student who is texting and asking the doubt.

The database is authorized by administrator, who can approve the faculty and student registration. So there will be no misuse of the chat application and their data is stored database and maintained and view by admin only.

Diagrammatic Illustration:



Advantages:

- Efficient way to share large amounts of data and messages and connect database easily.
- Sub-systems need not be concerned with how data is produced centralized management.
- e.g. backup, view secure messages, store messages, store details of faculty as well as students etc.,
- sharing model is published as the repository schema.

Disadvantages:

- Sub-systems must agree on a repository data model. Inevitably a compromised.
- Data evolution is difficult and expensive
- Security and availability issues
- Change of repository & evolution is difficult.
- If repository is not secure then the messages and other chat details are misused.

Layered Architectural Style**Components:**

Components are nothing but programs or subprograms that deals with the corresponding layers

Connectors:

Connectors are the procedure calls or the system calls. When one layer must communicate with the other layer it is done based on this procedure calls or system calls.

Topology:

Bus Topology or circle Topology

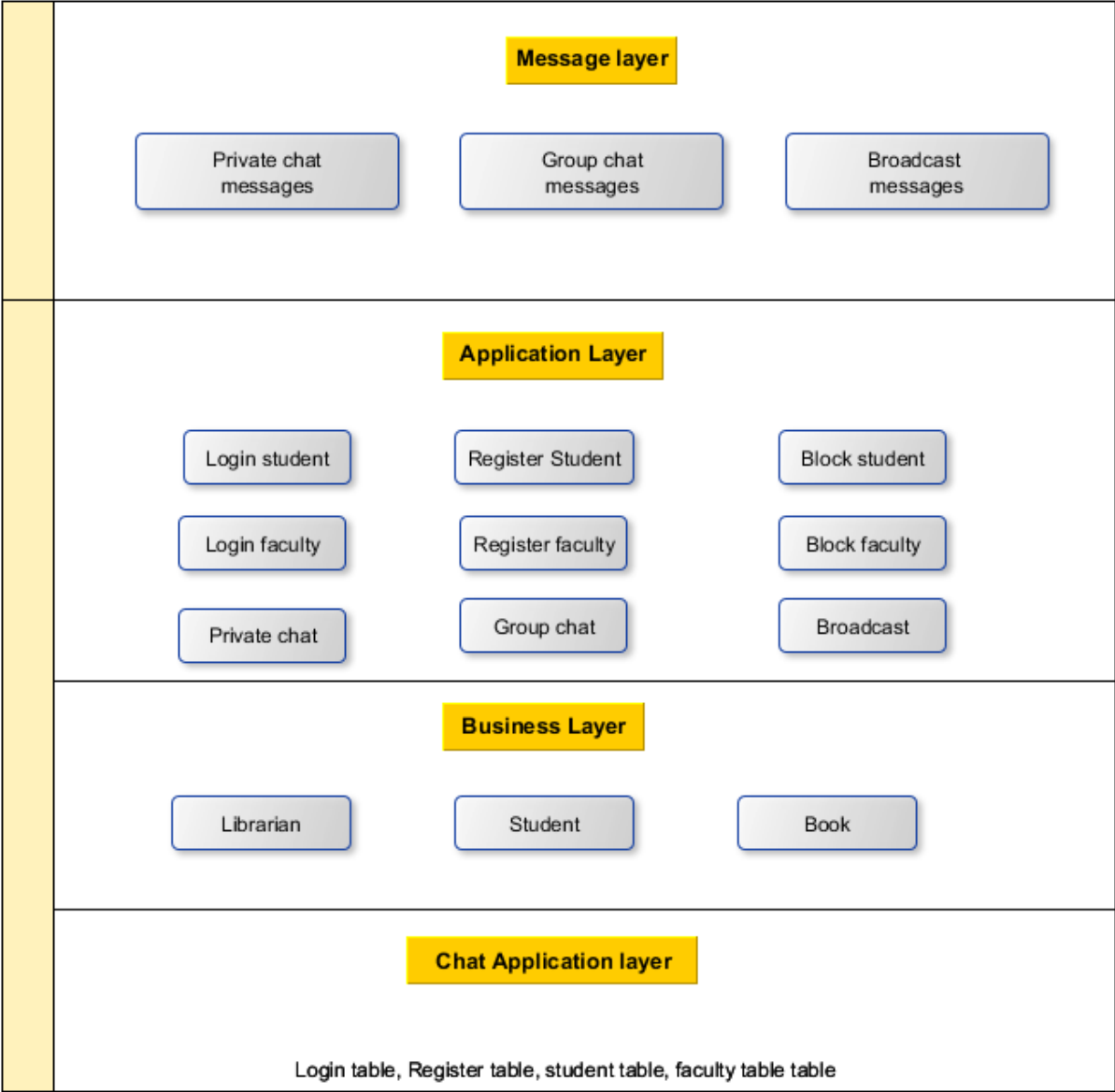
Description:

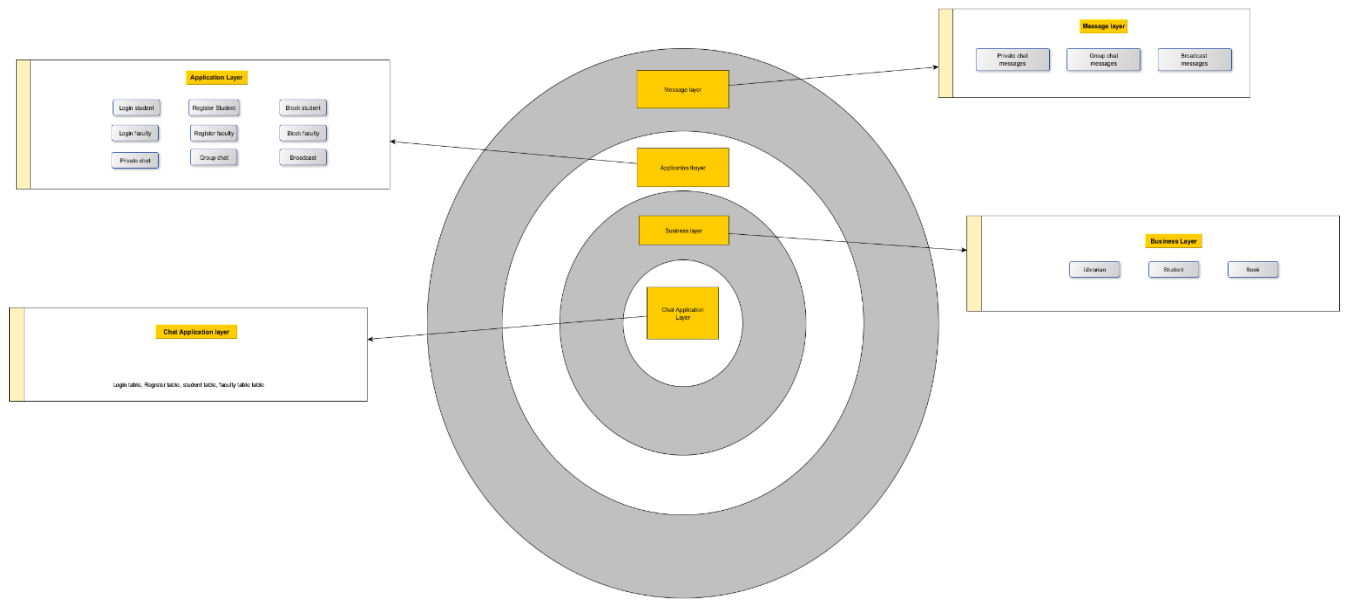
The chat application we can divide it into mainly layers. The core layer is chat application, the first layer that stays top on the core layer is the layer contain the data of student and faculty. Then comes the application layer which contain the login, register, private chat, group chat, broadcast etc., The last layer contains messages that frequently changes and from time to time. All the messages of private chat and group chat are present here.

These layers are one after the other so that the communication is done from one layer to other. The message layer is connected to private chat and group chat of the preceding layer and the data of these layers, the users of the chat are in the next layer.

The core layer is not modified. The main core things like design and other details of chat application are present in the core.

Diagrammatic Illustration:





Advantages:

- Firstly, with the help of this layered style, it supports design based on increasing levels of abstraction.
- Reuse is also another main advantage by using layered technology.
- Also, these support enhancement
- High potential for adding layers.
- Change in layers do not actually affect layers.
- Strict set of dependencies allow you to ignore outer layers.
- Can replace inner layers if interfaces remain the same.

Disadvantages:

- Performance degrades if we have too many layers (extra overhead of passing through layers and changes will pass slowly to higher layers).
- Sometimes difficulty to cleanly assign functionality to the layers.
- Can't be used because it adds more complexity even if it is simple

Client-Server Architectural Style

Components:

The components are client and the server. The one component id client which sends the request and the server accepts the request and respond.

Connectors:

The connectors are the communication between the client and server through some wireless connections like sockets or by via internet connection.

Topology:

Bus Topology or Hierarchical Topology

Description:

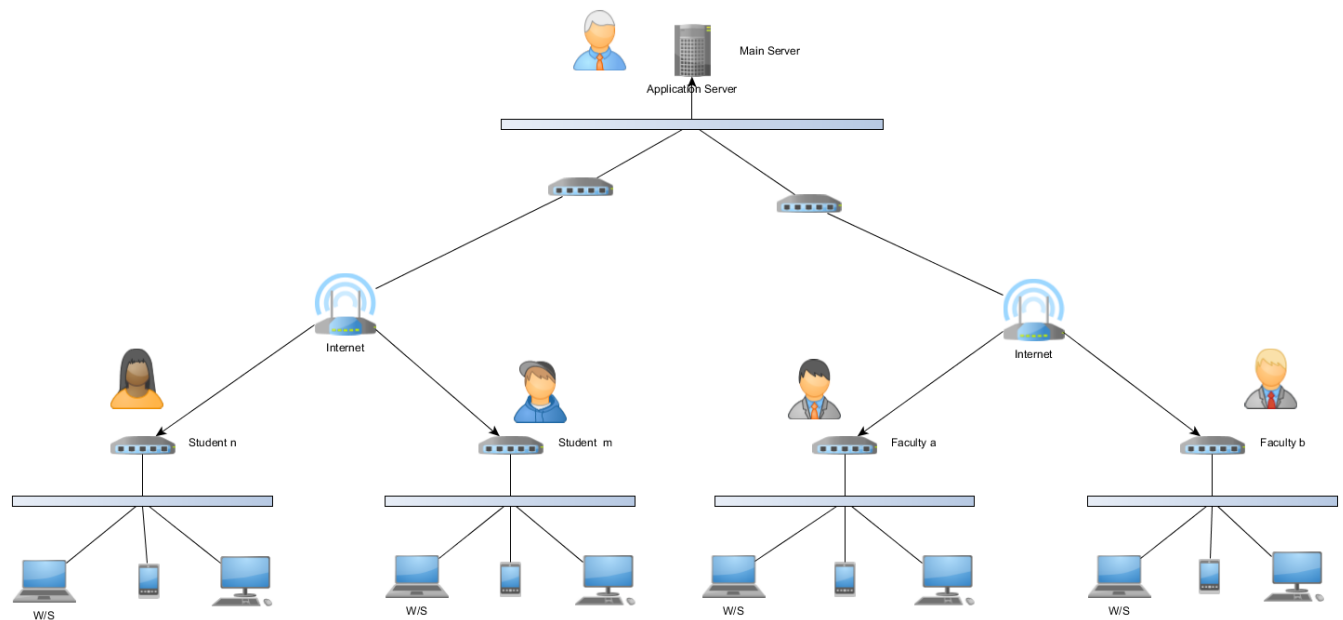
The main architecture style used in the Chat Application is client-server architecture. This is because the messages are sent, and the spontaneous response are possible through the client-server architecture.

The client i.e., the users... student and faculty must interact through server connection which accepts the response from the users through wireless connectivity like internet, intranet or by sockets and localhost.

If a student wants to chat with the faculty in private chat, the student must click on the faculty with whom he wants to chat. The request is sent to the server. The server verifies the student and the faculty whom the message is to be sent and approves the request from the client. If the any of the request is not found it throws a exception about the same.

When the student or the faculty register client send request to admin and the admin approves if the details are correct or rejects or blocks if anyone misuse.

Diagrammatic Illustration:



Advantages:

- Understandability
- Small number of tiers, like layered-system properties
- Re-usability, especially regarding legacy applications
- Powerful enough server tiers can accommodate many clients
- Disreputability, components communicate over a network, generally.
- Secure as without the server permission the client cannot proceed.

Disadvantages:

- Visibility and maintainability: difficult to analyses and debug.
- Potential for server outages.
- When the load on the server is high the communication gap decreases the performance of the system.
- Legacy wrappers.
- Require sophisticate interoperability mechanisms.

Pipe and Filter Architectural Style

Components:

The components are the filters. The input data and output data is from the filters that data that is required.

Connectors:

The connectors are the pipes. These pipes connect two filters and transmit the data with the input given to one of data is transmitted to other filter through pipes.

Topology:

Ring like Topology, but ends are not connected

Description:

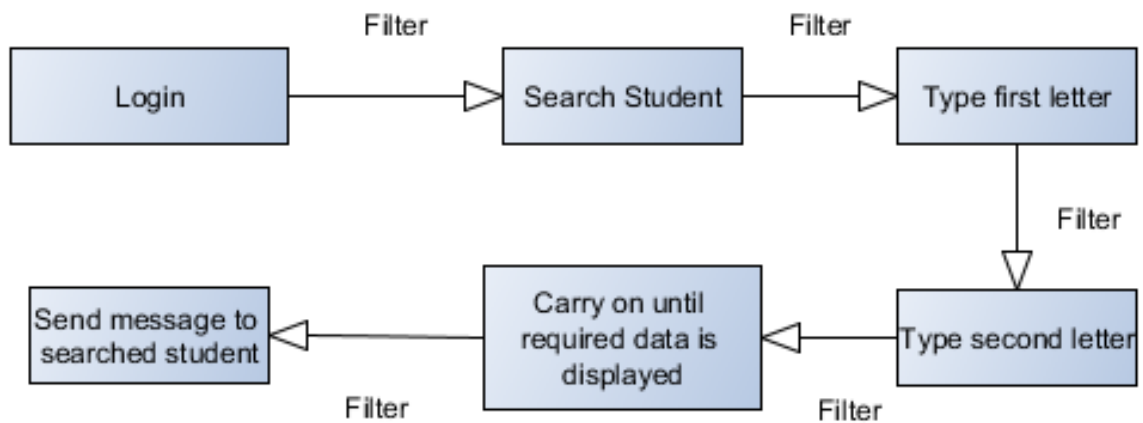
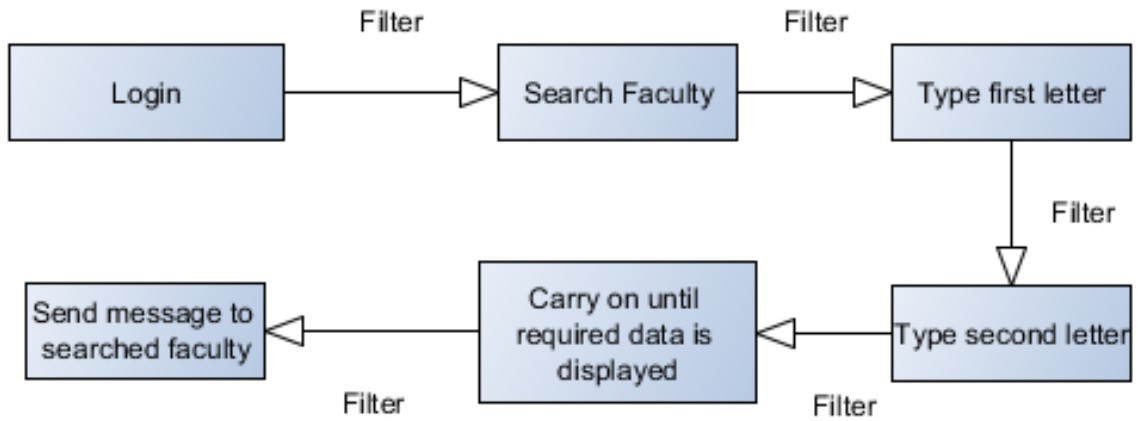
In Chat Application the module like search faculty and search student, we use pipes and filter.

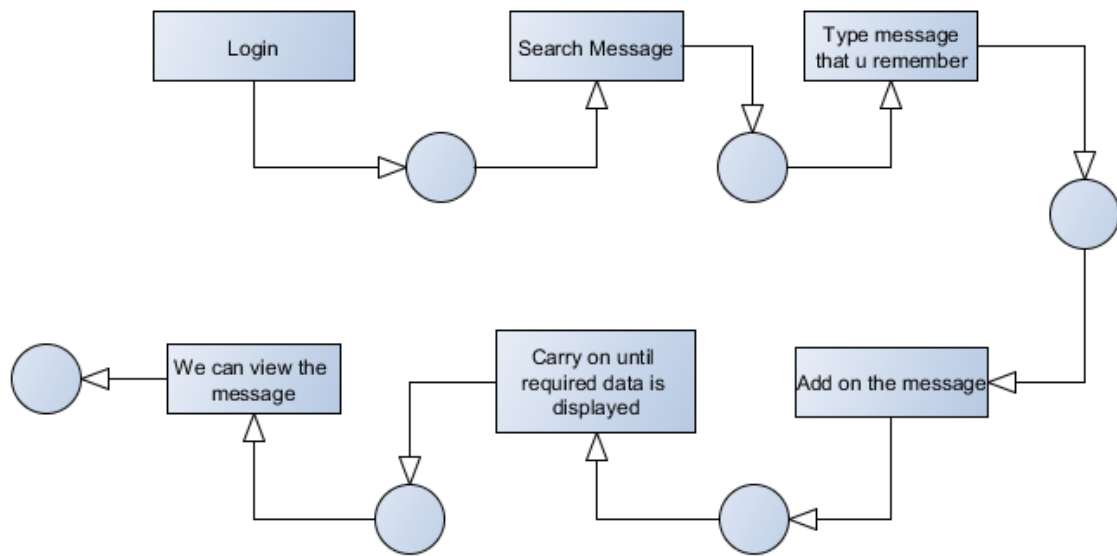
In searching the faculty, student use the search engine present. Student type the name of the faculty. When he types the character one by one the database is filtered by using the piped and filters.

At last the required data is filtered from the database using pipes and filters and resultant name is displayed. This process is used for even in student search.

If a student want to search any message in the history, pipes and filter is used and the message is typed (some what he remember) the filter and the data is sent to pipes to other filter and the resultant is displayed.

Diagrammatic Illustration:





Advantages:

- The data is filtered with each filter and the required data is identified faster.
- In all the search modules and modules that require data to be filtered we use pipes and filters.
- They allow the designer to understand the overall input/output behavior of a system as a simple composition of the behaviors of the individual filters.
- They support reuse: any two filters can be hooked together, provided they agree on the data that is being transmitted between them.
- Systems can be easily maintained and enhanced: new filters can be added to existing systems and old filters can be replaced by improved ones.
- They permit certain kinds of specialized analysis, such as throughput and deadlock analysis.
- Finally, they naturally support concurrent execution. Each filter can be implemented as a separate task and potentially executed in parallel with other filters

Disadvantages:

- In some process the data must be sequential. If the filters are changed then

the data required is not obtained.

- Pipe and filter systems often lead to a batch organization of processing
- Not good at handling interactive operations
- They may be hampered by having to maintain correspondences between two separate, but related streams
- Depending on the implementation, they may force a lowest common denominator on data transmission
- Parsing and unpausing of data
- Loss of performance, increased complexity

Process Control Architectural Style

Components:

The components in process control are the process definition and control. The process includes mechanism for manipulation some process variable. The control algorithm for deciding how to manipulate process variable.

Connectors:

The connectors like pipes that transfer the data from the process to control and to the output.

Topology:

Ring like Topology.

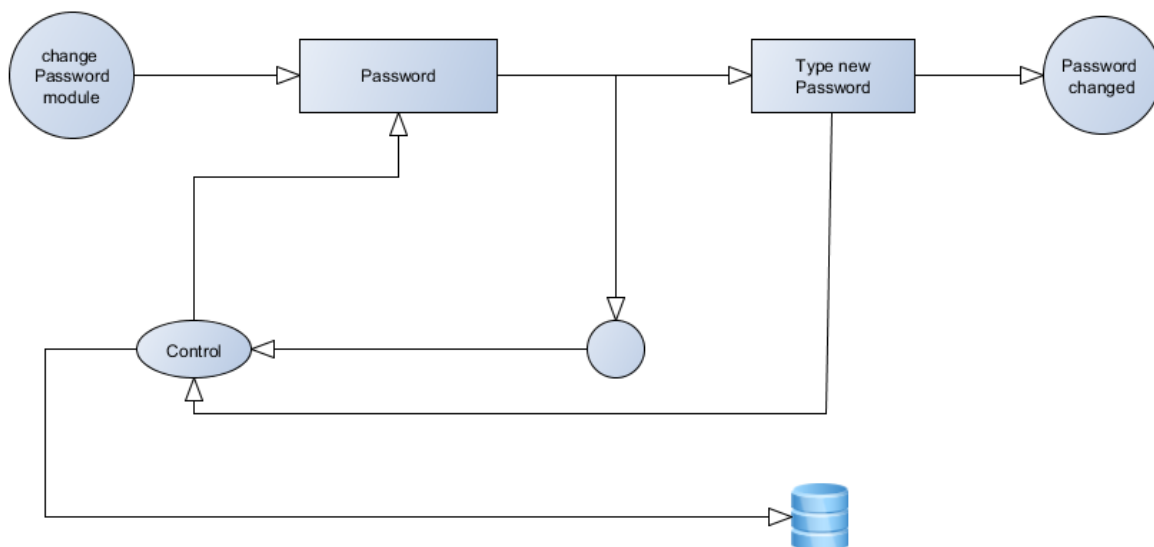
Description:

In the chat application in password change module we use process control module.

The process of changing the password, the control verifies whether the old password that we type is correct with the database and allows to change the password.

The password that was typed is updated to the database to using the same control.

Diagrammatic Illustration:



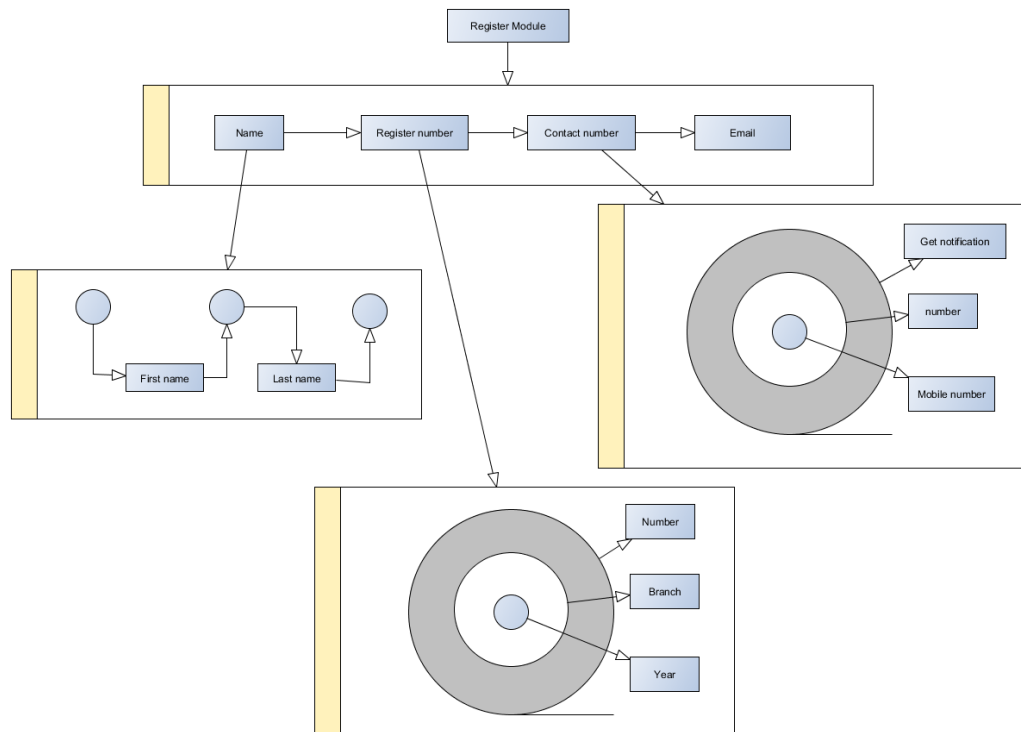
Advantages:

- The data to proceed to output is verified again using the control and proceeded to output.
- The data is double verified.
- Its secured and easily maintainable.
- Until data pass through the control, it is not moved to output.
- The control act as checkpoint.

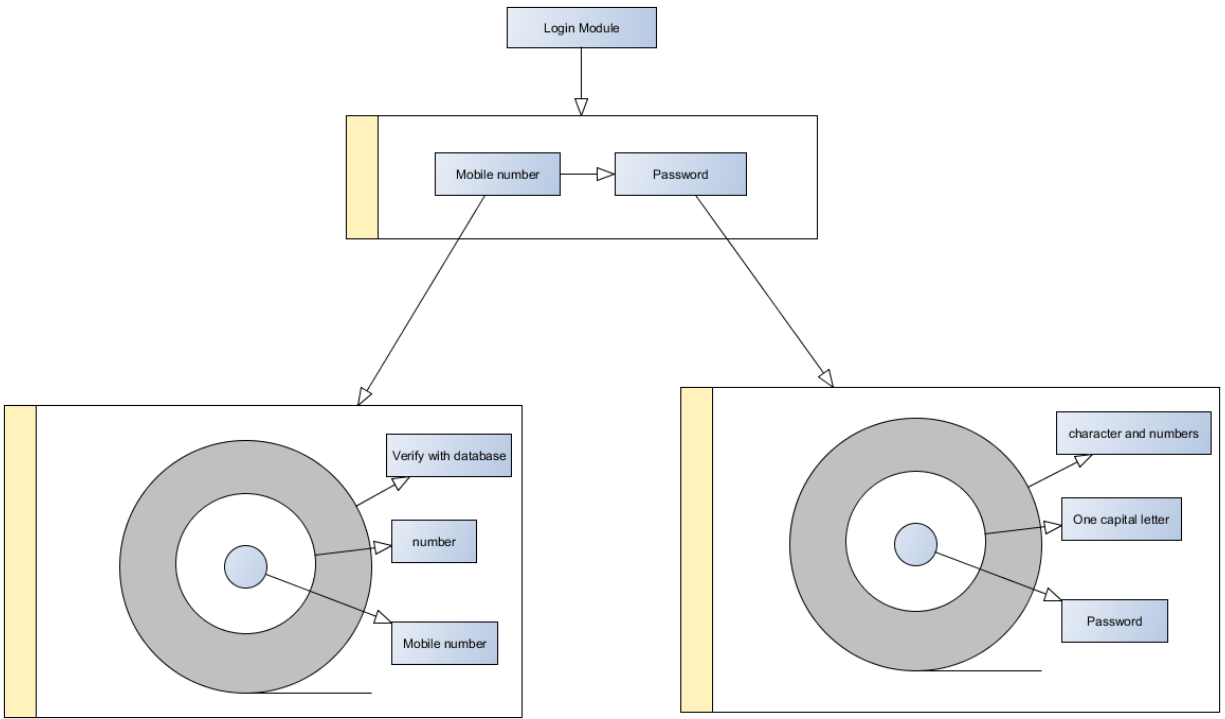
Disadvantages:

- The output data is little time taking as it must go through series of control to come to output.

Register Module:



Login Module:





Review 3-Final Report Winter 2017-18

MTECH-(SOFTWARE ENGINEERING)

**PROJECT BASED
COMPONENT**

SWE2004-SOFTWARE ARCHITECTURE AND DESIGN

Slot-D2

Chat Application

16MIS0257- V. Sai Nikhil

16MIS0308- Tanvi

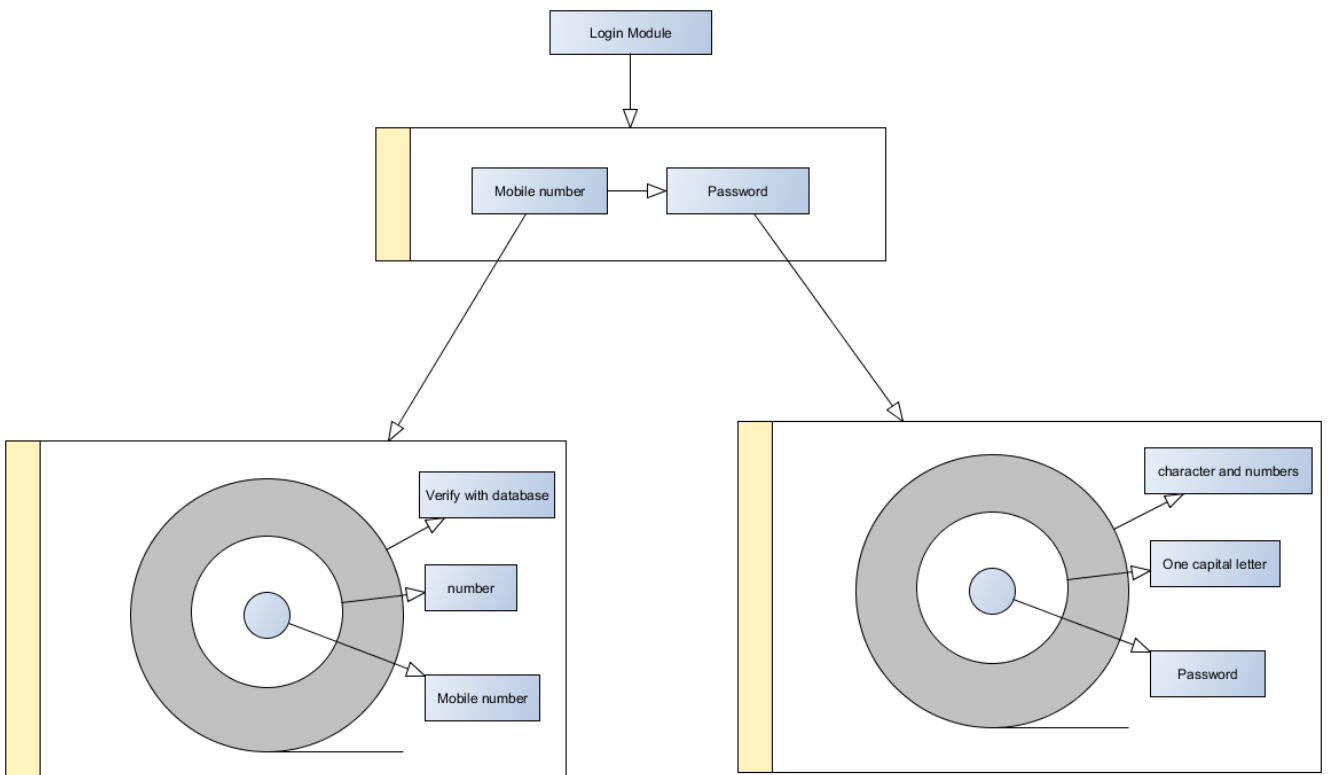
22 – MARCH - 2018

FACULTY: Dr S. SREE DHARINYA

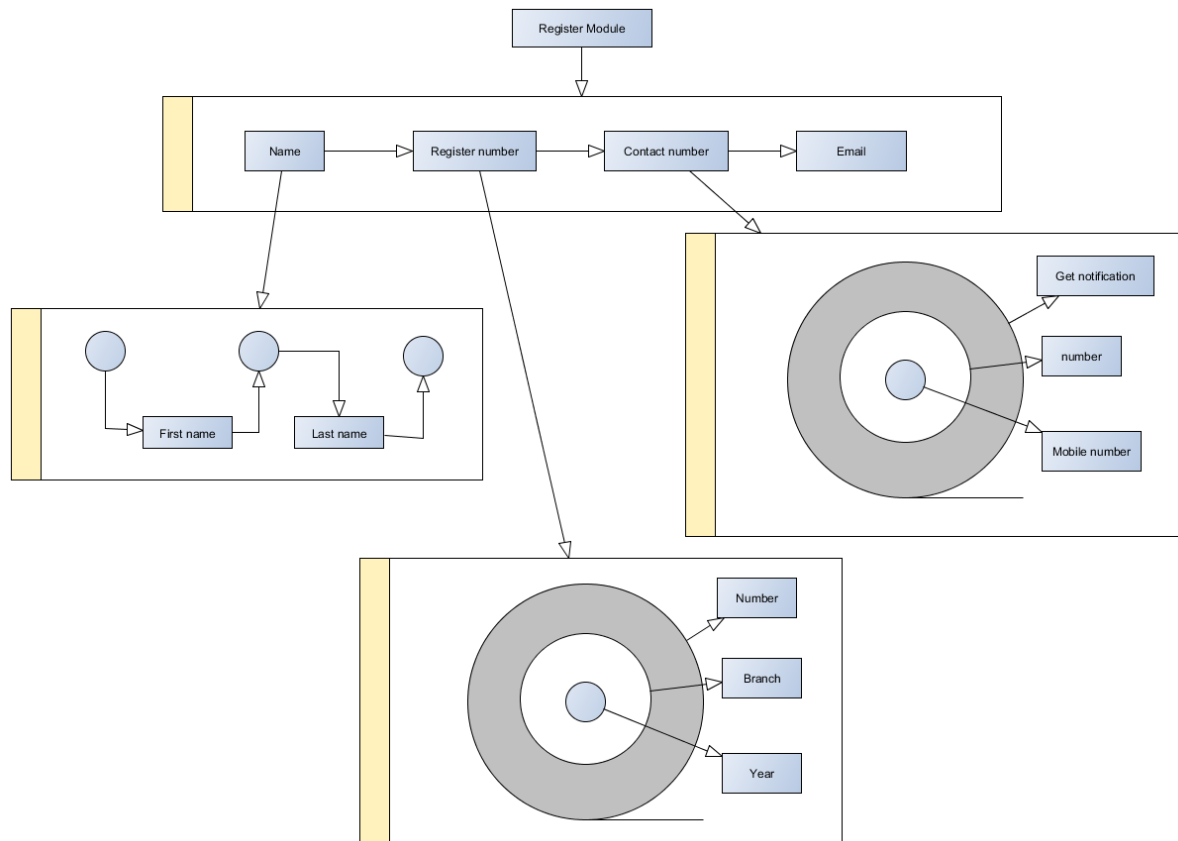
Modules in Chat Application:

1. Login module
2. Register module
3. Student module
4. Search student/faculty module
5. Search message module
6. Client Server module
7. Change password module
8. Group chat module
9. Block user module

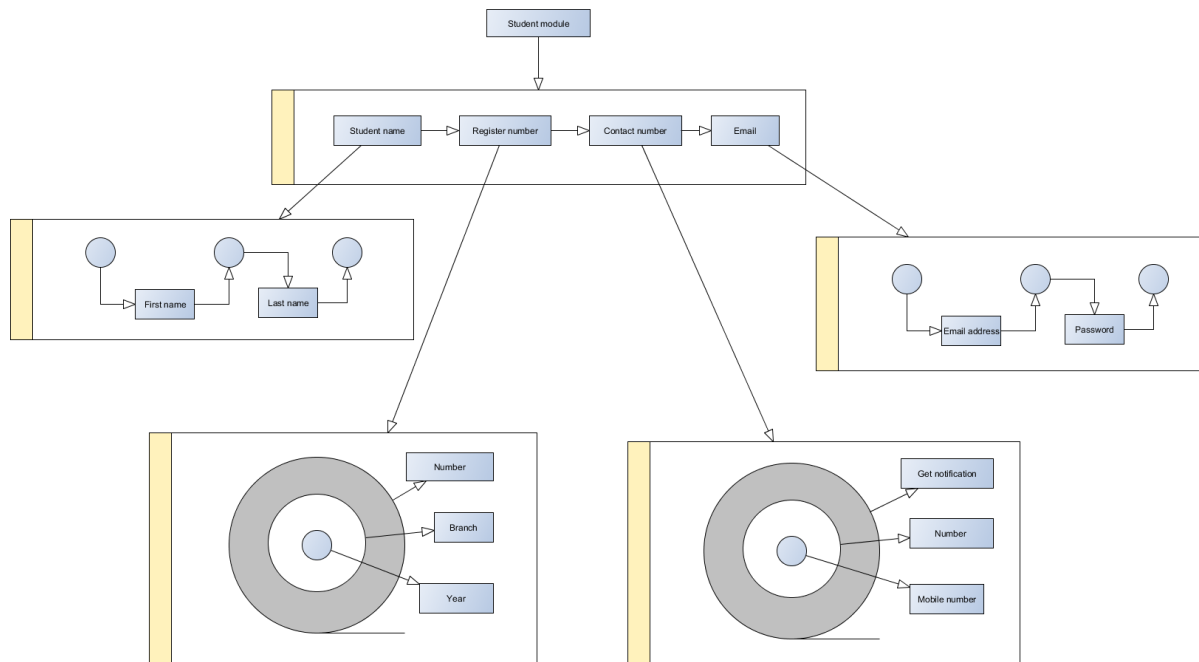
1.Login Module:



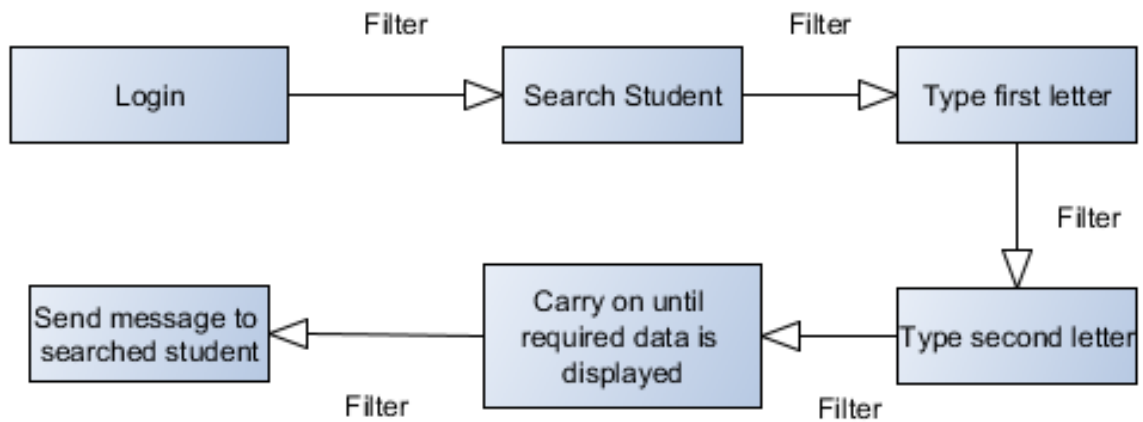
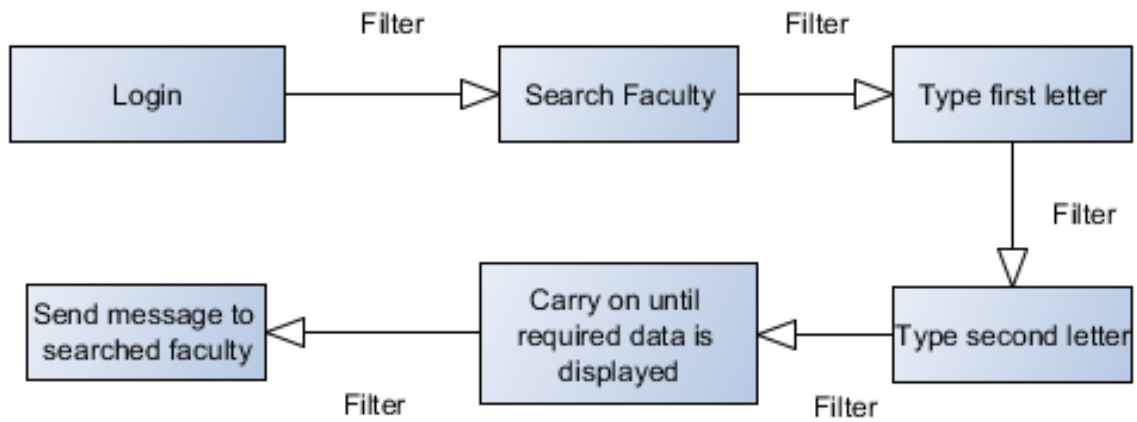
2.Register Module:



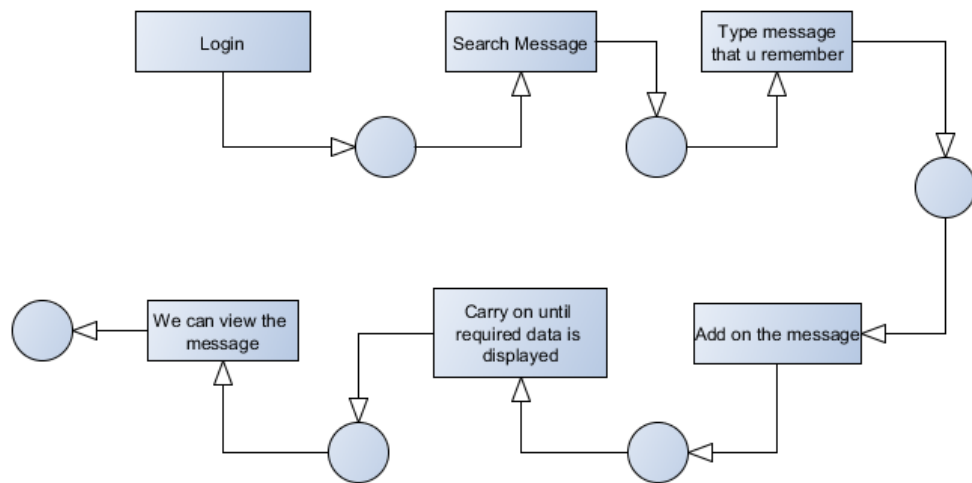
3.Student Module:



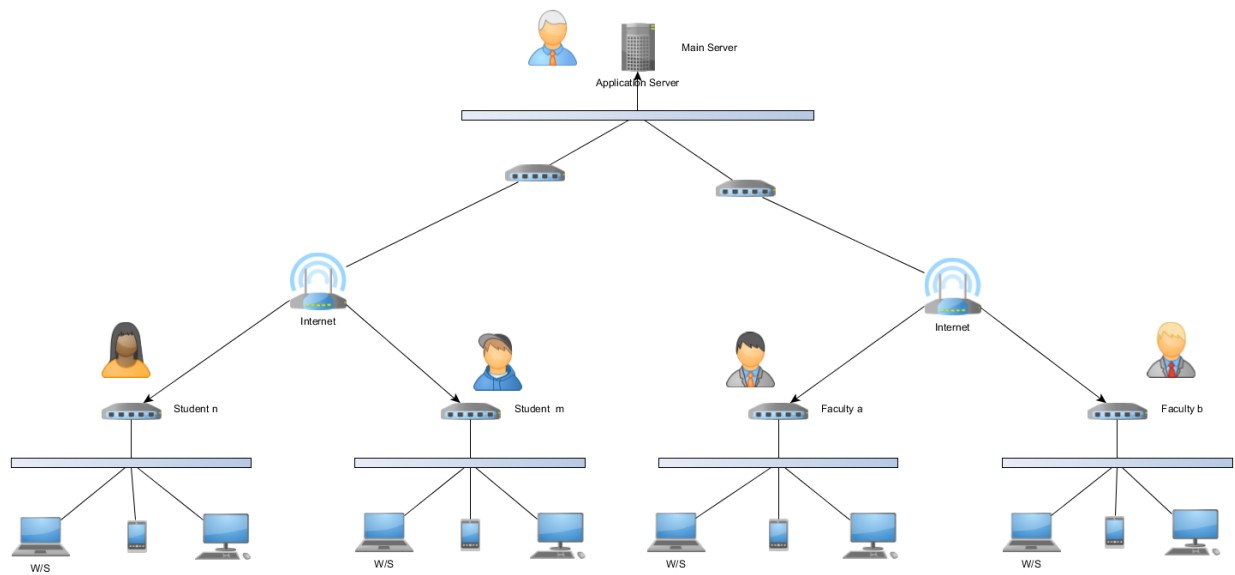
4. Search Student/Faculty Module:



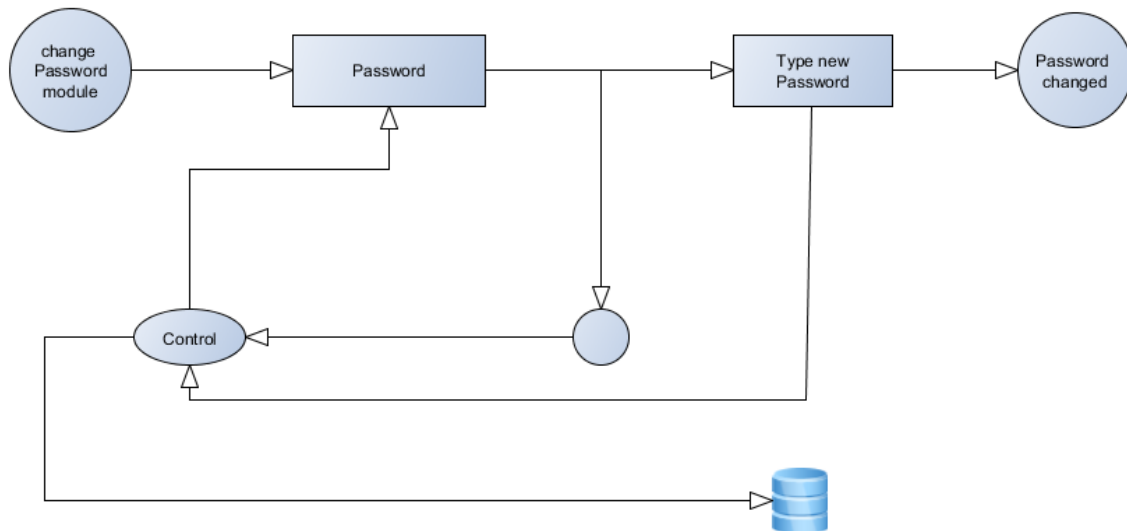
5. Search Message Module:



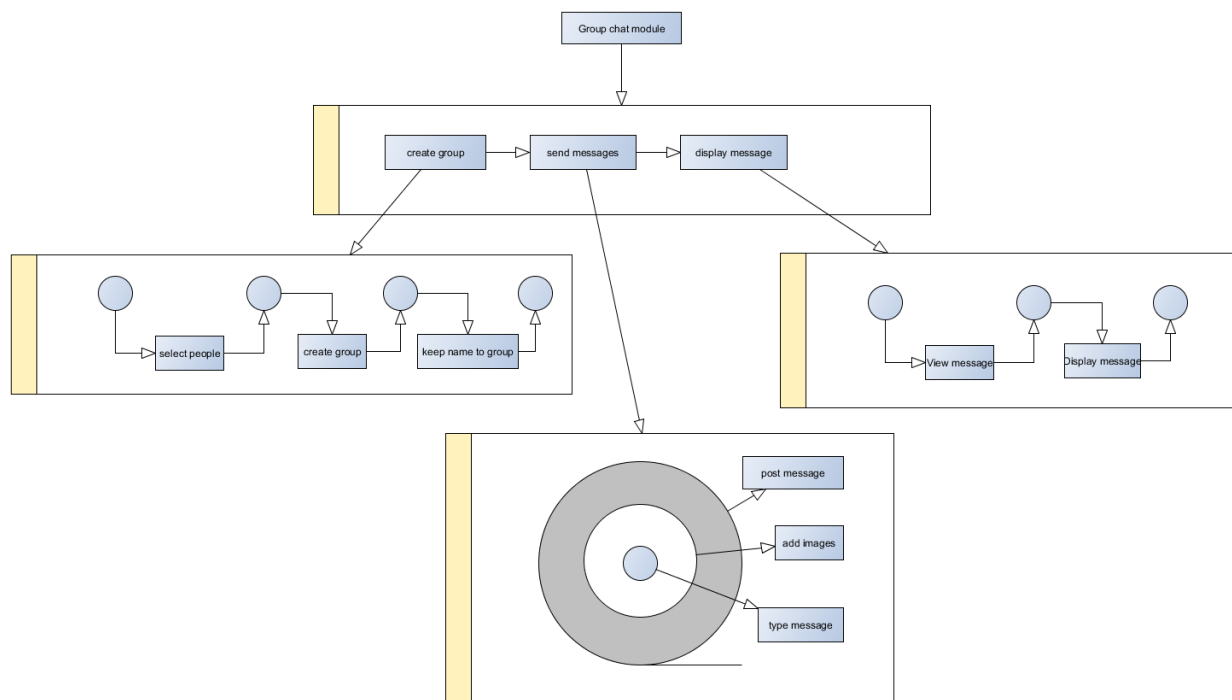
6.Client Server Module:



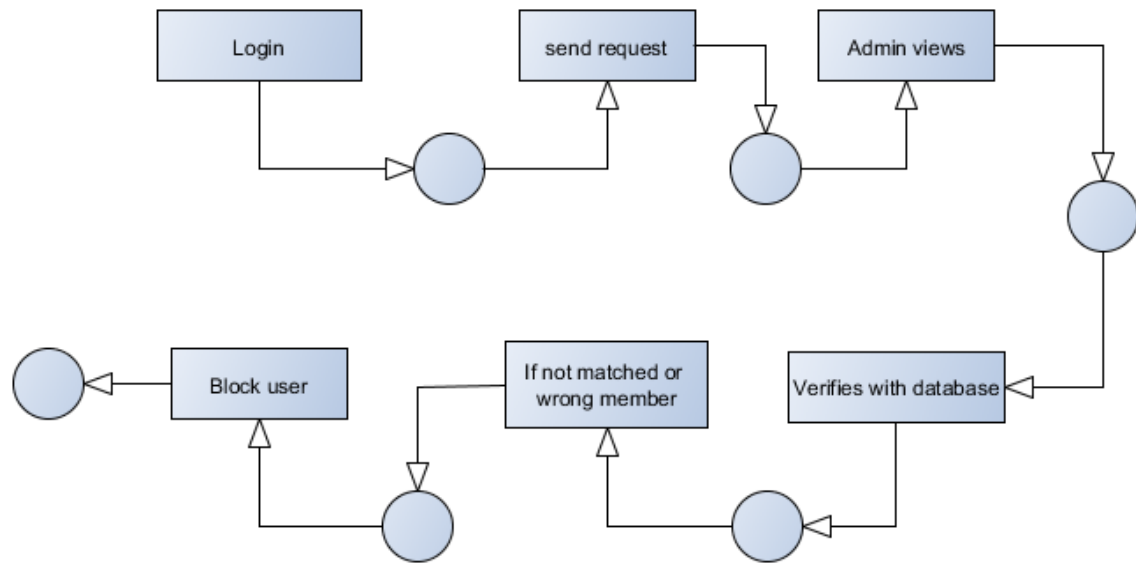
7.Change Password Module:



8.Group Chat Module:



9.Block User Module:



Mapping of Architectural styles with modules:

