

# Architecture Design Document for SNMP (Social network for medical professionals)

By THUMATI UJJIEVE (CS16BTECH11039 ), P RAMKISHAN ( CS16BTECH11029),  
SHANMUKHA RAO S (CS16BTECH11034), ABS HARSHA (CS16BTECH11002)

## 1. Overview

### 1.1 System Overview

Social Network for medical Professionals (SNMP) is intended to help a medical professional to gather intelligence through questions and communicate with other medical professionals. There are many different questioning platforms in present days but they were not customized for medical professionals and at the same time many such applications do not allow a formal or informal type of communication between them

### 1.2 System Context

The system context is defined clearly in the SRS. Basically, the registered user (Medical professional) is the main sink and source of the information. The non-registered users don't provide any information to the website and have only read access to certain sections of the website.

### 1.3 Stakeholders of SNMP

The main stakeholders for the system are the individual registered users who might use the system, non-registered users who just serve as sink and the system designer/builder who will build SNMP. The main concerns of these stakeholders are:

- **For Registered Users (medical professions):** The usability of the system and acting as a source for data. Reasonable response time is also a concern.
- **For Non-registered User:** The usability of the system, response time are concerns
- **For designer/builder:** The system is easy to modify, particularly to handle future extensions mentioned in the SRS (i.e. groups etc.) Hence, the key property for which the architecture is to be evaluated is the modifiability or extensibility of the system. Response time performance is another factor for which the system needs to be evaluated.

## 1.4 Scope of this Document

In this document, we describe two possible architectures for SNMP, compare them for various quality attributes, and then choose the most appropriate one, which is our final proposed architecture for PIMS. By discussing the two alternatives, we also provide the rationale for selecting the final architecture. For architecture, we consider only the component and connector view

## 1.5 Definitions and Acronyms

### ***Definitions:***

<i>User</i>	it means a medical professional who has been registered and can act as a source
<i>Non registered User</i>	These are a general user who need not be a medical professional and can't be a source
<i>level</i>	To measure the trust of a user we have introduced this term, it depends on points of the user which depend on the information provided by other users

### **Acronyms:**

SNMP	Social network for medical professionals
MVC	It is Model-View-Controller Architecture.

## 2. Architecture Design

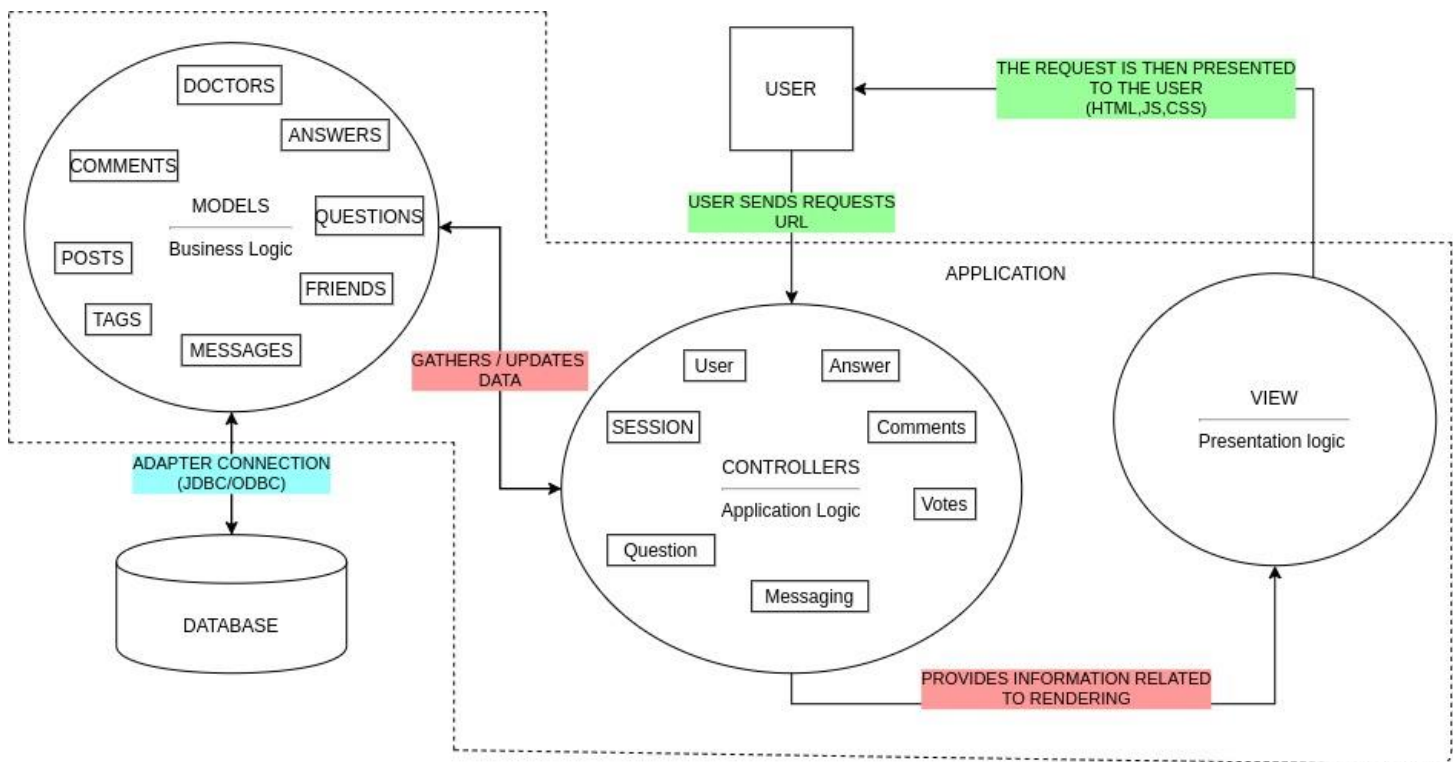
### 2.1 Architecture 1: MVC BASED ARCHITECTURE

This architecture separates an application into three components:

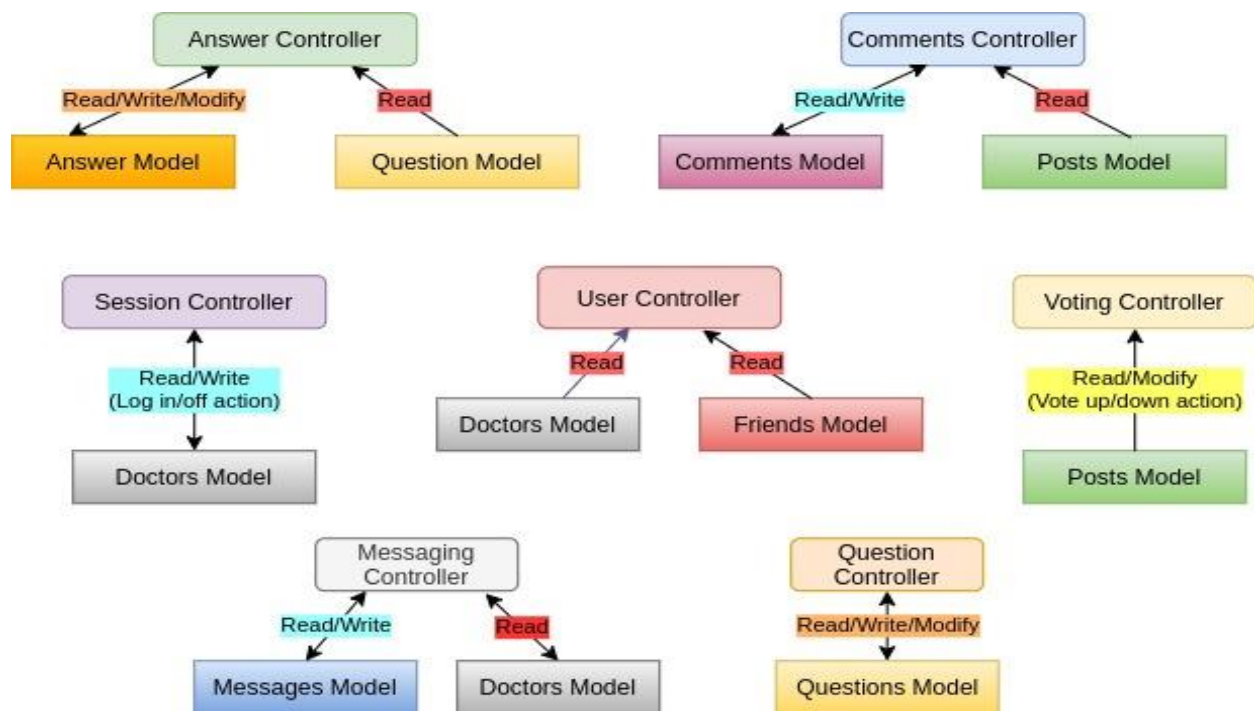
1. Models for handling data and business logic
2. Controllers for handling user interface and application
3. Views for handling GUI objects and presentation.

For example:

1. The browser (client) sends an http request for a page to the controller
2. The controller receives the data it need from the model
3. The controller gives the retrieved data to the view.
4. The view is rendered and sent back directly to the client.



The Controllers and Models are subject to change as per requirements.



### Description of Each Component:

Component	Component Type	Description
Controllers	Processing (Interface Module, Processes Application Logic)	A controller can thus be thought of as a middleman between models and views. It makes the model data available to the view so it can display that data to the user, and it saves or updates user data to the model.
View	Processing ( Processes presentation logic of the application)	It contains code for displaying HTML, CSS etc. files only. It receives responses from the controller. It processes the information received from controller and then puts it in a template
Model	Processing ( Processes the business logic of the application )	The models are like Ruby classes having attributes. They are like object representation of the relations/table stored in the database. They talk to the database, store and validate data, perform the business logic and otherwise do the heavy lifting
Data Repository	Database	This module is the database that stores the information in relational models. SQL language is used to interact with the database.

Connector	Connector Type	Description
Read	Database Access (Indirect through models)	This connector represents that the model the controllers are accessing is only in Read only Mode. This connection is between some controllers and models where the controller can only send read requests for that particular model
Read/Write/Modify	Database Access (Indirect access through models)	This connector represents that the controllers want to access the model in read, write or modify mode. So the controller has the right to send requests to modify the information stored in the database as well.
URL	HTTP (requests made to the Controller)	This connector is between the User and the Controller of the application. The user interacts with the application only through Http requests.
JDBC/ODBC	Database adapter	The models in the application accesses the data from the database through the adapters provided by them. Data migration is simple since the business logic resides in the Model component.
Control Connector	To perform application logic	This connector is between the controller and the model and view component of the application. This connection acts as a bridge between the model and the view component of the application.

## Architecture 2: The Repository Model

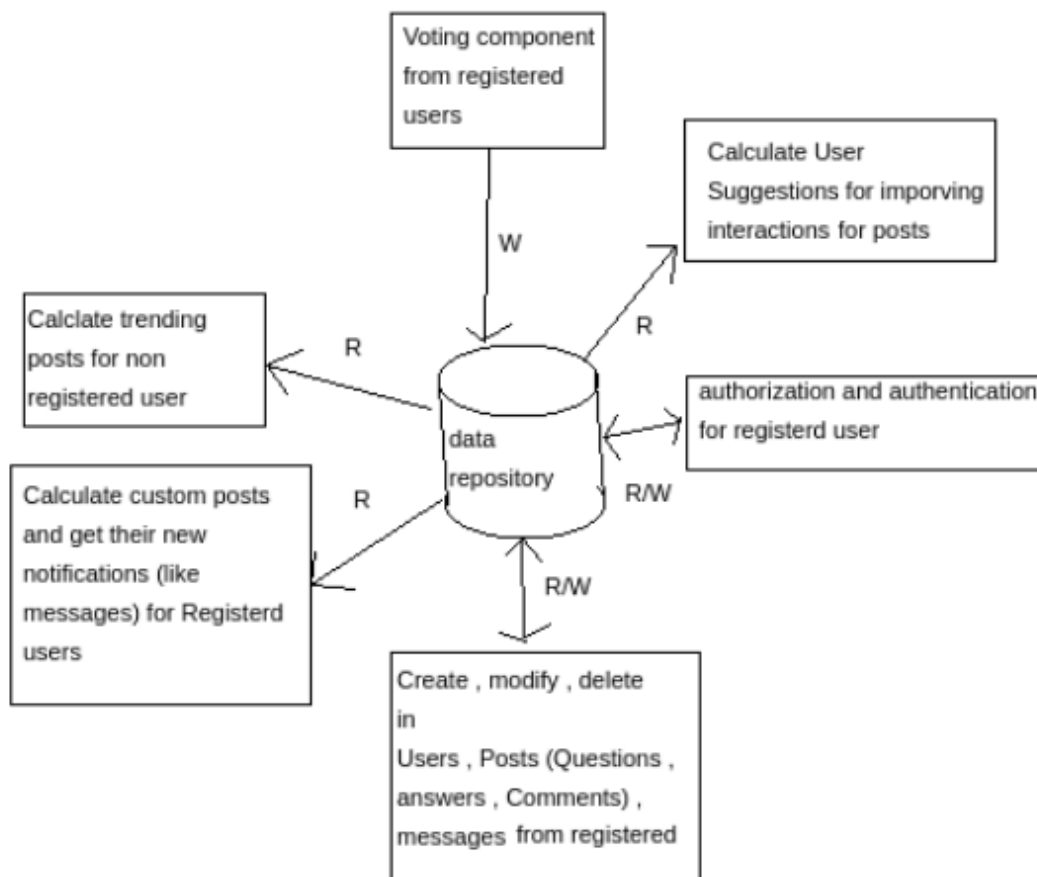
This architecture consists of a data repository, which contains information about

- Users , Questions , Answers ,Comments ,Messages

And there are separate modules for performing various tasks such as calculating ratings of a user and rating of questions, answers etc..

#	Component	Component type	Description
1	Data repository	Database	This module is the database containing information about the Users , Questions , Answers , Messages and Comments
2	Create , Modify , Delete	Processing (Database modification)	Modifies the information related to the data in databases.
3	Calculate trending posts for non-registered Users	Processing (computation)	This module calculates top most accessed questions/answers using various norms from information in the data repository
4	Calculate Custom posts and get notifications for registered users	Processing (Database access and computation)	This module calculates the personalized things for registered users based on data for which they acted as source
5	Voting component from registered Users	Processing (database modification)	This acts as a source for ratings that affect connector 3 and 4.
6	Calculate User Suggestions for improving interactions for posts	Processing (data base access and computation)	When a question/post is posted it gathers information from data repository from various users and tries to select users who might be interested in answering/viewing it
7	Authorization and authentication for registered user	Processing (database access )	To manage sessions, authorization, and for other permissions.

8	Master controller	Processing (server module)	This is module where actual user interacts from other end , when a request comes based on request it invokes other modules and sends response
---	-------------------	-------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------



Information about Connectors:

#	Connector	Connector Type	Description
1	Control connector	Module invocation or function calls	This connector is between the master controller and the other modules (except data repository)
2	R/W connector	Database access and modification	For certain modules like 2, 7 it's used. it represents reading relevant data and modifying of data repository
3	R only connector	Database access	Some modules only use data repository for reading data. in those cases R connector is used
4	W only connector	Database modification	Its present between data repository and voting module , when a valid user up/down votes it modifies data in data repository

The diagram below show this architecture

## 2.3 Comparing the architectures:

Criteria	Architecture1	Architecture2
Change in data repository	Easy, Since it's Model component contains all the logic	Not easy
Adding more functionalities	Very Easy add more controllers	easy
Large scale	Could upgrade or tune each component to support large scale	Difficult as it might form bottle neck in master controller
Providing additional securities	Easy	Difficult
Code reuse	More	Less



### **3. Final Architecture of SNMP**

The architecture 1, MVC architecture is better for upgradability, database migration and code reuse. It's also easier to provide additional security. The reusability is also very high since a model may be reused by multiple views. Ease of maintenance is more in MVC architecture since the components have low dependency on each other.

Since scalability is a big issue in 2<sup>nd</sup> architecture. So the final architecture chosen for the web application is Model-View-Controller Architecture.