Software Requirements Specification

for

SOE Mini Project

Version 1.0 approved

Harsh Aryan (ITM2017003)

Nayan Agarwal (IWM2017501)

Gyanendra Singh (ICM2017005)

Ayush Shukla (IIT2017106)

Omkar Khodwe (BIM2017007)

INDIAN INSTITUTE OF INFORMATION TECHNOLOGY

Table of Contents

1. Introduction

1.1 Purpose

This document is a Software Requirement Specification (SRS) for building the website for ***Real Time Complex Event Processing and Analytics for Smart Buildings****.*
Smart buildings equipped with sensors and electronic devices as a Cyber Physical Systems (CPS) offer great research perspective to explore communication, computation and controlling of physical devices by using real time Complex Event Processing (CEP) and analytics. Since, CPS like Smart-Building involves the integration of several types of equipment interoperability, maintainability, signaling, bandwidth, reliability, security, privacy, authentications, data storage, heterogeneity and cost effectiveness are the critical issues to be addressed. To handle all these challenges integration of smart buildings with complex event processing and real time data analytics are emerging as a new area of research.

1.2 Document Conventions

This document uses the following conventions:

DB	*Database*
AI	*Artificial Intelligence*
ML	*Machine Learning*
JS	*JavaScript*
API	*Application Programming Interface*
HTML	*Hyper Text Markup Language*
CSS	*Cascading Style Sheets*
NLP	*Natural Language Processing*
ER	*Entity Relationship*
UML	*Unified Modelling Language*

1.3 Intended Audience and Reading Suggestions

The project is a follow up website for the Smart Building project. The intended audience of the project include research personals, professor under which the project is being carried forward, and the people are interested in knowing about the project.

The documentation is segregated into various sections which the reader can search in the content section and subsequently follow to the desired part of the document.

1.4 Product Scope

The purpose of this smart website system is to create a convenient and easy to use environment and provide its users a complete information about the primary project containing work experience, recent works, publications, projects, contact information and other major/minor details.
It also provides a means to let the interested users get updates regarding the project on a real time basis.

1.5 References

[1] Sonali Agarwal, Rohit Raj , Rajat , BIki "Real Time Complex Event Processing and Analytics for Smart Building" , (CICT'17) .
[2] Natural Language Toolkit

2. Overall Description

2.1 Product Perspective

The project was envisioned taking in mind the importance of a fully functional website for a Project in today's world. Be it presenting your Project in a conference, or referencing it to a faculty, or a recruiter, it all comes under a single platform packaged as a website. The website helps present the project in an organised manner, taking help from links and graphics, which can't be done in case of a conference paper.

And in the tech-savvy world we live in, it is an absolute must to have a portfolio website for every Project which hopes to receive any attention and be published in a good journal.

The SRS is about a Full fledged website of the Project "Real Time Complex Event Processing and Analytics for Smart Buildings" fully equipped with links to all the references and other papers necessary to understand the project. Also, it adds a chatbot that can answer questions relevant to the website and the Project.

2.2 Product Functions

The project serves the following major functions:

1. *Provides an overall perspective about the project.*
2. *Provides an interface to contact and connect to the project members.*
3. *Provides a mechanism to directly ask the project members for doubts/queries.*

4. *Chatbot: an easy way to get the information required without having to go through the entire website.*
5. *Features description, publications and achievements related to the faculty.*

2.3 User Classes and Characteristics

The user classes for this project are:
Project Members : Project Members who are working under the project. They can add or remove contents related to modules , achievements so for and faqs related to the project. They can also interact through the audience and answer about their queries,

Audience on Internet: Any person on the internet accessing the website for getting information about the running project. Audience has very limited access. Can only view content available as public in the website.

Professor: The professor has all user accesses. He / She can edit the content on the website, add or remove publications and achievements.

2.4 Operating Environment

The product is a website designed to be used on various platforms and browsers. It is compatible on and platform-independent to various Operating Systems. It can run on Android and iOS Mobile devices as well. It is compatible to various mainstream browsers used such as Google Chrome, Mozilla Firefox, Opera Mini and the likes of it.

2.5 Design and Implementation Constraints

Every browser comes with it's own set of constraints.The software would work robustly for most of the browsers except for one that have javascript and cookies blocking enabled.The most prominent example of this is the Tor Browser which doesn't allow even canvas pictures to show on the website citing security concerns.

3. External Interface Requirements

3.1 User Interfaces

3.1.1 Sublime Text Editor

Code editor interface will be the main graphical user interface where developers will interact with their website source code and its including. It will allow graphically create, edit and delete files, *running and debugging their programs and other basic feature that and average integrated development environment provided.*

3.1.2 Web Browser

A simple web browser will be required by developers as well as the users . The developers will use the browser for testing purposes. The users will used website for browsing the website. The web browser must be updated enough to load all the features of a website and then interpret it.

3.2 Hardware Interfaces

A computing device which has a web browser . The website will be loaded on the browser. The computing device must contain sufficient memory to run the browser. The website is compatible to fit to any screen size:. There are wide array of screen sizes, hardware specifications and configurations because of intense competition in mobile software and changes within each of the platforms (although these issues can be overcome with mobile device detection). The absolute minimum requirement is a 200 MHz processor, 32 MB of RAM and 32 MB of storage.

3.3 Software Interfaces

3.3.1 Operating System

We have chosen the windows operating system for its friendly nature. Also, windows use less disk space, boot faster, and better protected against attacks. Windows includes its own help section, has a vast amount of available online documentation and help, as well as books on each of the versions of Windows. It has made several advancements and changes that have made it much easier to use an operating system.

3.3.2 Machine Learning For Chatbot

It will be used to communicate with user(which is using website) . The chatbot give response to the queries of the user. Various machine learning algorithms and techniques will be used for training the model of chatbot for the user's queries.

3.4 Communication Interfaces

3.4.1 Query Sheet

User can ask queries regarding the courses, currently working projects, or any other specific query to the faculty. All of the asked queries will be stored in a google sheet file. The admin can see all the queries at any time he/she wants.

3.4.2 Communication Security

While submitting the query, the email id will be taken from the application for the communication security issues.

4. Other Nonfunctional Requirements

4.1 Performance Requirements

When any user or viewer asks a question , the question and the corresponding data related to the question is stored systematically and securely .The Proposed plan also makes sure that the database is properly normalized. These helps the admin to access the data significantly faster. There would be no problems with time to access the website as number of hits/requests to access the website would be significantly high.The chatbot would be interactive and fast as the weight for the neural networks are already optimized to make sure that there is less response time for the query asked by the user.

4.2 Safety Requirements

The system is secured as it will only provide access to authorised user. It will have proper login system which will require user id and password . At the same time the database contains the questions asked by the wider audience to the owner of the website . So if someone wants to check the questions from the database, he/she has to use the admin/faculty login to see it. The database is secured from SQL Injection techniques with the help of private keys and a SSL/TLS key encryption.This ensures that all data passed between the web server and browsers remain private and integral.

4.3 Security Requirements

It is more secure than the current websites the faculty of any institute are using for showcasing their projects which are protected only by the encryption on institute servers. But the proposed website would have dual advantage as it would not only be protected by the encryption of institute's servers but also by the SSL/TLS encryption which will stop SQL Injection attempts .The database cannot be altered by the user and only by the admin of the website.The questions can only be viewed and answered by the admin/owner of the website.The key web services security requirements are authentication, authorization, data protection, and nonrepudiation .

In the proposed website Authentication is required for accessing data through proper channels.Users or viewers of the website are only given minor privileges such as looking and reading the website as well as asking questions.

Collaborators of the project on the other hand are given more privileges than a normal user as he/she can see the progress of the project using the proposed website.Data is protected using a private key which uses SSL encryption. Nonrepudiation ensures that the message sender is the same as the creator of the message.

4.4 Software Quality Attributes

Following are the attributes which represents the quality of the software

1. *Availability: It is designed as a smart website for the project of Real Time Complex Event Processing and Analytics for Smart Buildings.*
2. *Adaptability: It will be easy to adapt with a basic user interface.*
3. *Maintainability: Low maintenance cost and can be maintained efficiently with minimum work.*
4. *Correctness: It will feed the data to the database correctly, as well as it will also ensure that the right person will get the right details.*
5. *Reusability: Code will be written in such a manner, so that it can be used further for another similar projects, also any edits to the website can be easily done.*
6. *Testability :There are multiple aspects to Testability of website namely validity check, consistency check,realism check,verifiability check and completeness check. Before launching, all these aspects of the product will be tested by the TA's and the faculty of the institute.After which the website will be deployed on an appropriate platform as per the requirements of the project.*
7. *Reliability: The privacy policy of the website has basic privacy guidelines. It is safe to use the product under the privacy policy thus the users can easily rely on the security system of the product. Also it will be free from any clashes either in terms of application or any server or database breakdown. With all of these quality attributes, software will ensure a good quality approach for the project.*

5. Software Development Life Cycle Model Used

Software development life cycle (SDLC) is a series of phases that provide a common understanding of the software building process. How the software will be realized and developed from the business understanding and requirements elicitation phase to convert these business ideas and requirements into functions and features until its usage and operation to achieve the business needs.

In our case, we have a clear and unchanging understanding of the requirements, the stages and activities can be well defined and are frozen. Also, the interaction with the user/client is limited, and the project is a rather simply definable project without much complexity. Hence we have used the conventional Waterfall Model.

The Waterfall Model is a linear sequential flow. In which progress is seen as flowing steadily downwards (like a waterfall) through the phases of software implementation. This means that any phase in the development process begins only if the previous phase is complete. Waterfall Model is considered appropriate for projects which do not focus on changing the requirements, for example, projects initiated from a request for proposals (RFPs), the customer has a very clear documented requirements. Waterfall model is simple and easy to explain and implement, hassle-free and faster than most other models and hence we have used the Waterfall Model here.

There are six sequential phases in waterfall model.

1. Requirements : The first phase involves the specifications and preferences of input and output of the product.

2.System Design : The second phase involves defining the overall architecture.

3.Implementation : The third phase involves small program called units that are small snippets of program that are integrated into the next phase.

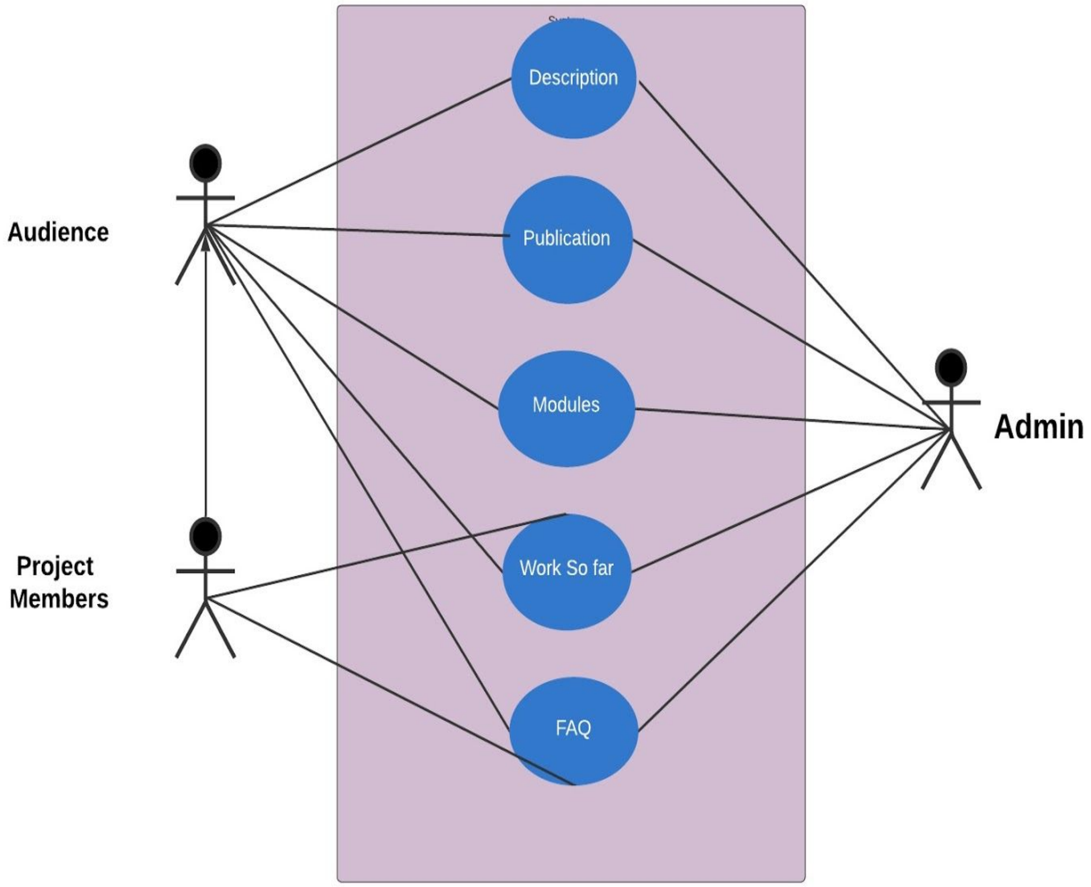
4.Integration and Testing :The fourth phase develops the units from previous phase and integrates it into the program.Furthermore, each unit is tested individually to check for various bugs and flaws.

5.Deployment :The fifth phase involves the deployment after all the units in the previous phase are combined.The deployment platform is chosen as per appropriate requirements.

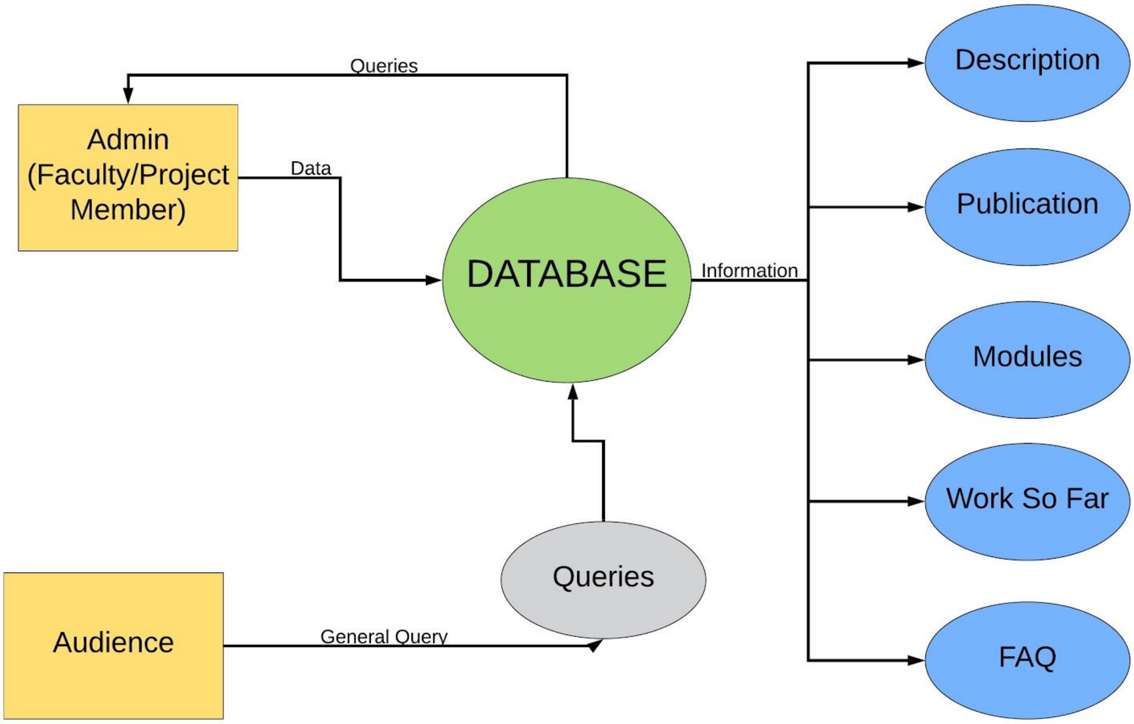
6.Maintainence :The sixth phase involves the care to be taken after installation i.e deployment.It involves making modifications and changes to improve the performance of the software.

6. Analysis Model

6.1 Use Case Diagram



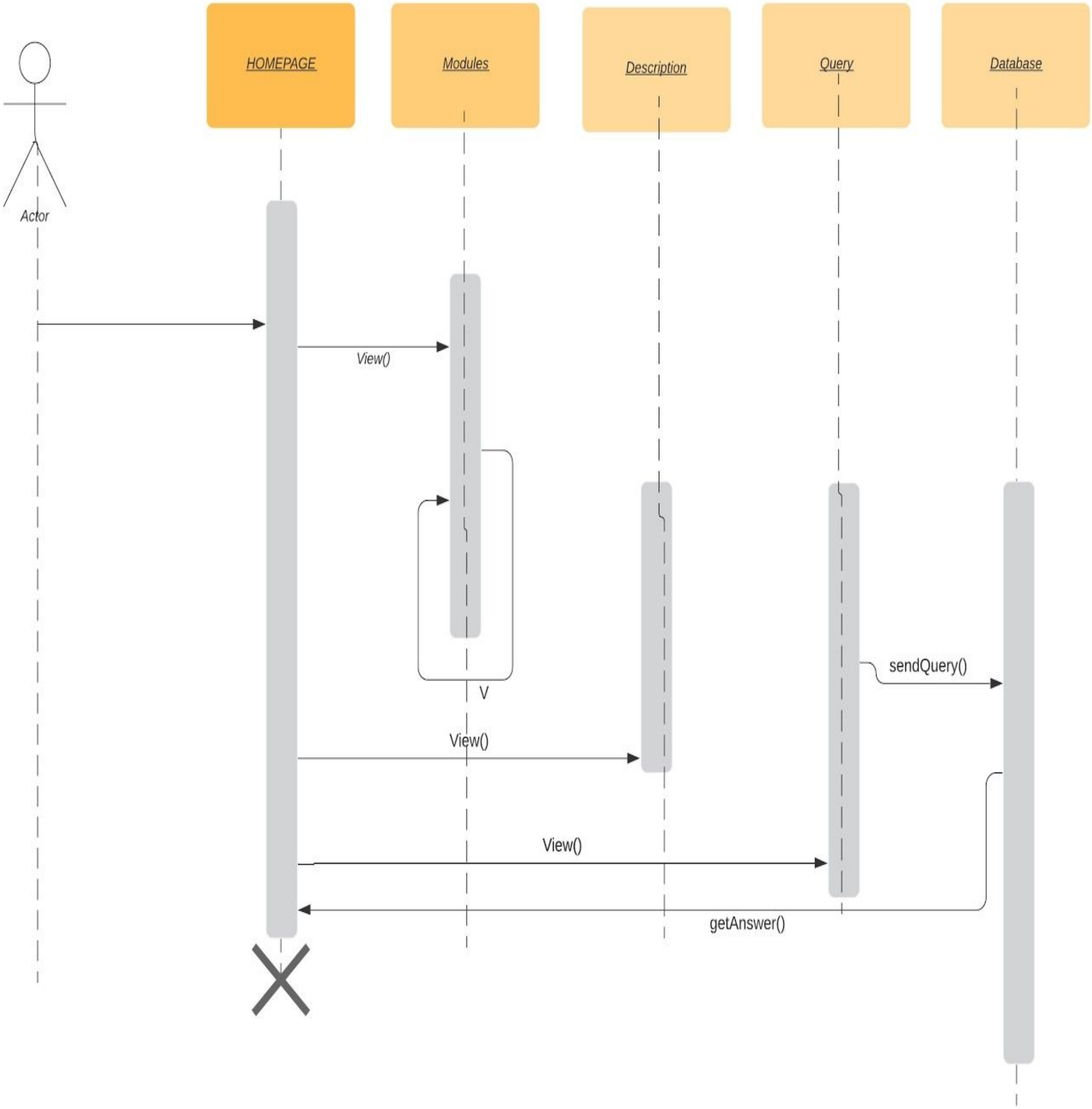
6.2 Data Flow Diagram



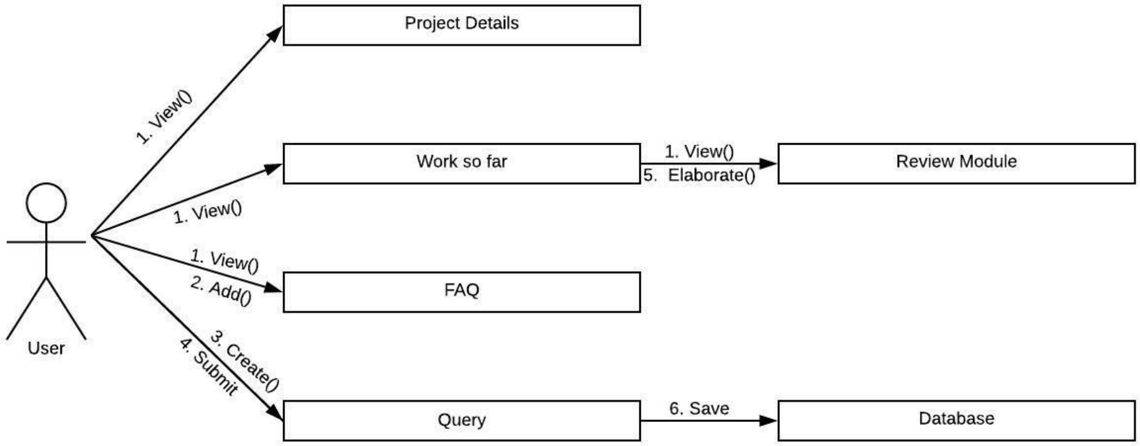
6.3 Sequence Diagram

A sequence diagram simply depicts interaction between objects in a sequential order i.e. the order in which these interactions take place. We can also use the terms event diagrams or event scenarios to refer to a sequence diagram. Sequence diagrams describe how and in what order the objects in a system function.

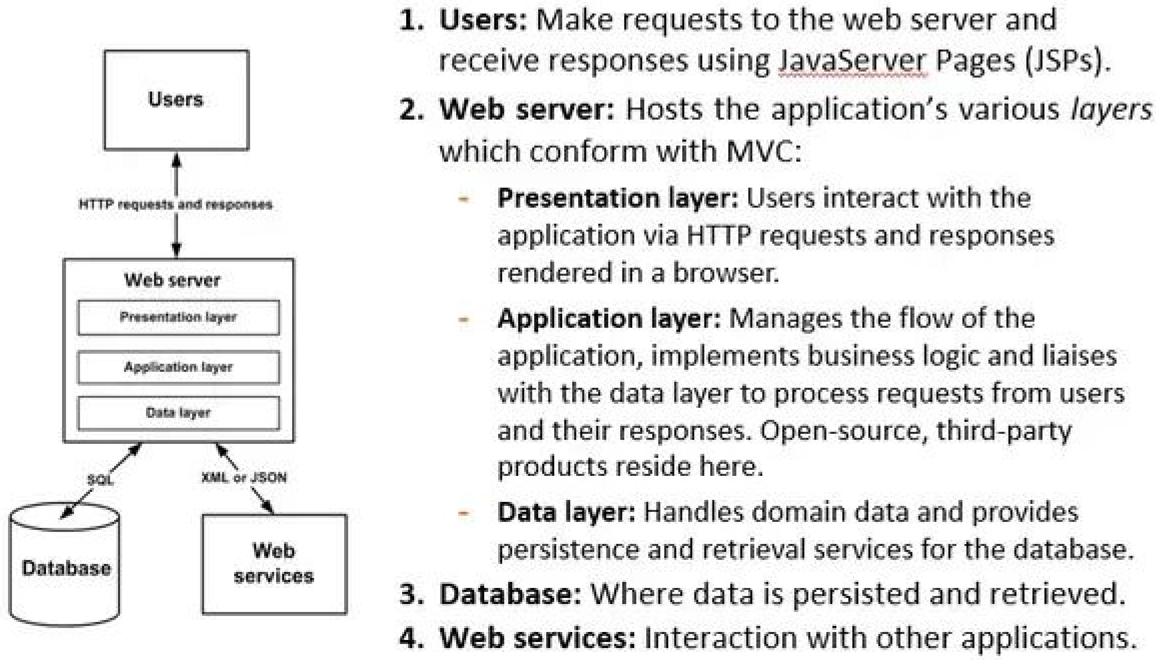
Given below the sequence diagram of this project


HOMEPAGE
Modules
Description
Query
Database
Actor
View()
V
sendQuery()
View()
View()
getAnswer()

6.4 Collaboration Diagram



6.5 System Architecture Diagram



1. **Users:** Make requests to the web server and receive responses using JavaServer Pages (JSPs).
2. **Web server:** Hosts the application's various *layers* which conform with MVC:
   - **Presentation layer:** Users interact with the application via HTTP requests and responses rendered in a browser.
   - **Application layer:** Manages the flow of the application, implements business logic and liaises with the data layer to process requests from users and their responses. Open-source, third-party products reside here.
   - **Data layer:** Handles domain data and provides persistence and retrieval services for the database.
3. **Database:** Where data is persisted and retrieved.
4. **Web services:** Interaction with other applications.

7. Project Timeline

Task	**Timeline**
Requirement Elicitation , making model	September 20

Adding Front End	October 16
Adding Forms Back End	October 23
Modeling, training and adding chatbot	October 27
Project Submission	October 30