Survey Builder System
Software Architecture Document

Version <1.0>

Revision History

Date	Version	Description	Author
5/4/2016	1.0	Initial software architecture document	Vinujan.S
10/4/2016	1.1	Requirement change	Vinujan.S

Table of Contents

Software Architecture Document

1. Introduction

This document provides an architectural description of the Solutions Survey Builder System using a number of different architectural views to depict different aspects of the system. It is intended to capture and convey the architectural decisions that have been made and elaborates on aspects of the system that are considered to be architecturally significant.

The views include use case view, logical view, deployment view, process view and physical view. Also described MVC architecture of the system, the architectural mechanisms applying to Survey Builder System and its size, performance and security characteristics. It illustrates what can be the content of a Software Architecture Document (SAD) produced during the RUP Elaboration phase.

RUP Software Architect will typically perform height major steps in order to define a global architecture, and each time an activity is completed, a specific section of the SAD is enriched accordingly.

1.1 Purpose

The Software Architecture Document (SAD) provides a comprehensive architectural overview of the Survey Builder System. It presents a number of different architectural views to depict different aspects of the system. It is intended to capture and convey the significant architectural decisions which have been made on the system.

In order to depict the software as accurately as possible, the structure of this document is based on the "4+1" model view of architecture.
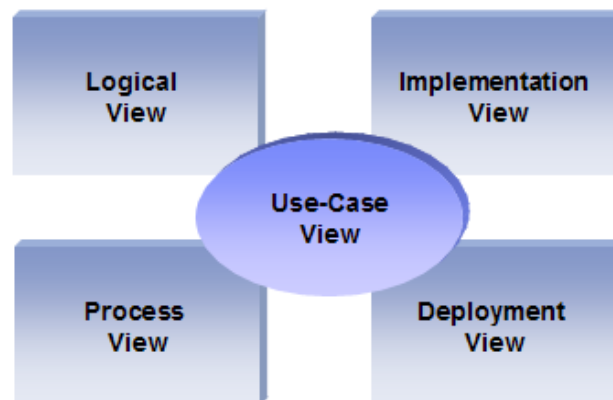
1.2 Scope

This document applies to the overall design of the system. It contains information relating to the architectural design of the application, the Structure of the Database, and of the physical servers hosting the site.

1.3 Definitions, Acronyms, and Abbreviations

MVC- Model View Controller
SBS- Survey Builder System

1.4   **References**

[1] Programmers.stackexchange.com, 'Mapping between 4+1 view model' 2014. [Online].Available: http://programmers.stackexchange.com/questions/233257/mapping-between-41-architectural-view-model-uml [Accessed 06 04 2016].

[2] Faculty.csupueblo.edu, 'SAD of online catering service' 2004. [Online].Available: faculty.csupueblo.edu/rick.huff/CIS432/SAD-OnlineCateringService.doc [Accessed 06 04 2016].C:\Users\vinujan\AppData\Roaming\Microsoft\Word\www.cse.hcmut.edu.vn\...\Document\...\Examples\SAD-OnlineCateringS

1.5   **Overview**

The rest of this document describes the following details.
Section 2: describes the use of each view
Section 3: describes the architectural constraints of the system
Section 4: describes the functional requirements with a significant impact on the architecture
Section 5: describes the most important use-case realization.
Section 6: describes design's concurrency aspects
Section 7: describes how the system will be deployed.
Section 8: describes any performance issues and constraints
Section 9: describes any aspects related to the quality of service (QoS) attributes

2.   Architectural Representation

Logical view: It describes the functionality that the system provides to end users. It can be described by using class diagram or sequence diagram.

Process view: It explains the system processes and how they communicate. It can be described by using activity diagram.

Deployment view: It concerned with the topology of software components on the physical layer, as well as the physical connection between these components. It can be described using deployment diagram.

Development view: it concerned with software management. It can be described using component diagram or package diagram.

Use case view: It shows the details of use cases of the system. It will describe the services provided by the system to users. It can be described using use case diagram.

3. Architectural Goals and Constraints

Technical platform

- This is a web application. Users can access this application via browser in smart phones and computer. Survey builder system will be deployed in J2EE application server

Performance

- If internet connection is good users can get the required service from the application.

Privacy

- Authentication credentials should be protected. Passwords of user should be encrypted. Answers given by normal users also confidential.

Security

- Users' password must be kept in a secure manner in order to avoid the problems from malwares.

The application must implement basic security behaviors:

- Authentication: Login using at least a user name and a password
- Authorization: according to their roles, users must be granted or not to perform some specific actions

For internet access, the following requirements are mandatory

- Confidentiality: sensitive data must be encrypted
- Data integrity : Data sent across the network cannot be modified by a tier
- Auditing: Every sensitive action can be logged
- Non-repudiation : gives evidence a specific action occurred
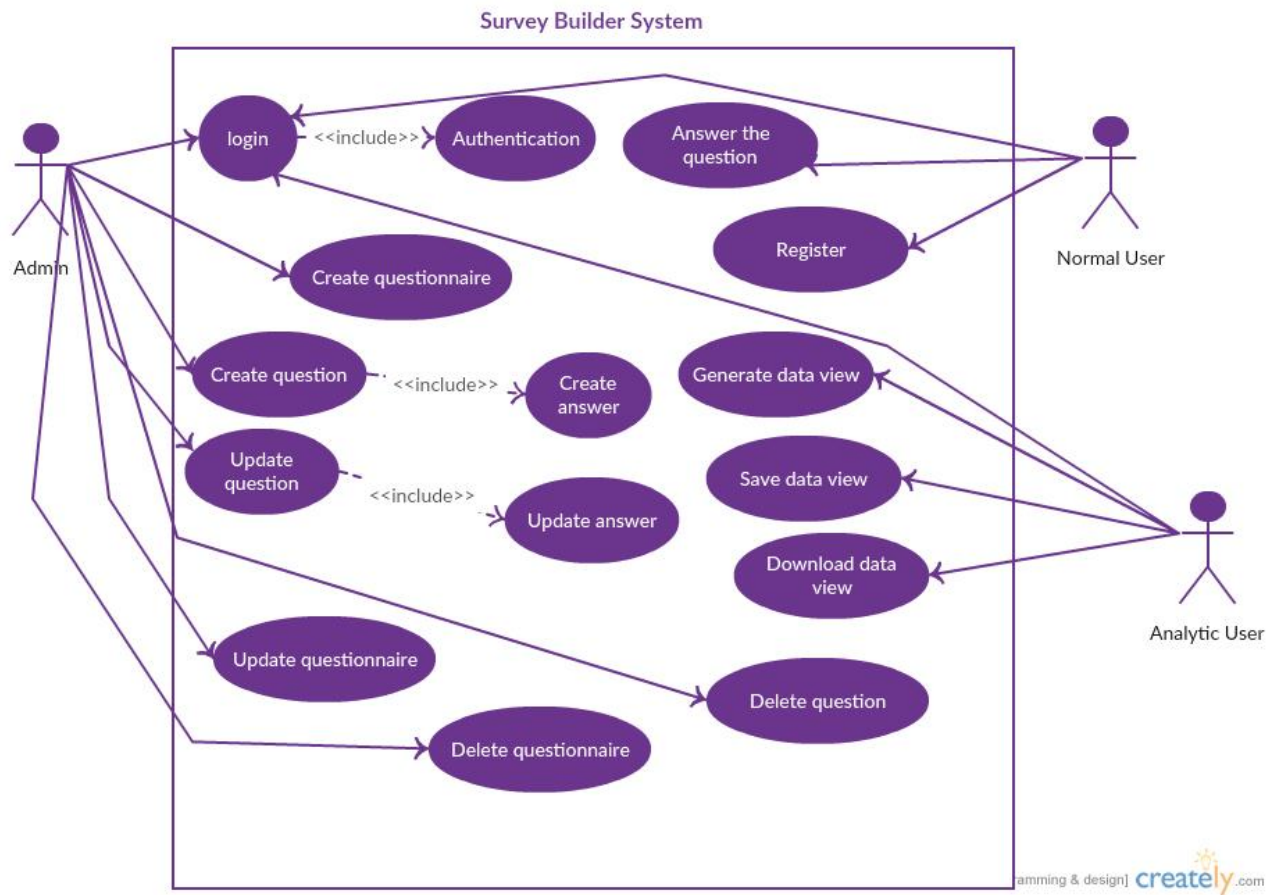
Portability

- As this is a web application. Every devices can access the application via a browser.

Persistence

- Data persistence will be addressed using database.

4. Use-Case View

Survey Builder System

login <<include>> Authentication

Answer the question

Admin

Create questionnaire

Register

Normal User

Create question <<include>> Create answer

Generate data view

Update question <<include>> Update answer

Save data view

Download data view

Update questionnaire

Analytic User

Delete question

Delete questionnaire

4.1   Use-Case Realizations

In the above use case we can clearly see that main functional requirements of the application are

- Creation of question and answer by admin user
- Answer the question by normal user
- Generating data view by analytic user

These are the use cases for managing questionnaires, questions and answers

- Update and deleting of question and answer
- Update and deleting of questionnaire

Analytic users' helper use cases

- Save and download generated data view

4.1.1   Creation of question and answer

For creating of question and answer admin user interface will be filled with set of question and answer retrieved from database. Admin user can drag and drop question type and answer type to create question and answer. Then admin can edit question answer template to add the content to the component. Adding component will be saved to the database. Content may be multimedia type. Admin can add photo, voice and video via camera and mic. These contents will be stored to database. And also admin can add any number of sub questions.

4.1.2   Answer the question

When normal user navigate to application. User will be prompt with questionnaire according to his identity. Question will change according to the time period and location. Normal user can also add multimedia file as answer (all question will have comment about question area) in real time or from device storage.

4.1.3   Generation of data view

After analytic user logged in to application, user interface with graph, report and datasheet generation configuration will we shown. By giving suitable input configuration analytic user can generate data view.

5. Logical View
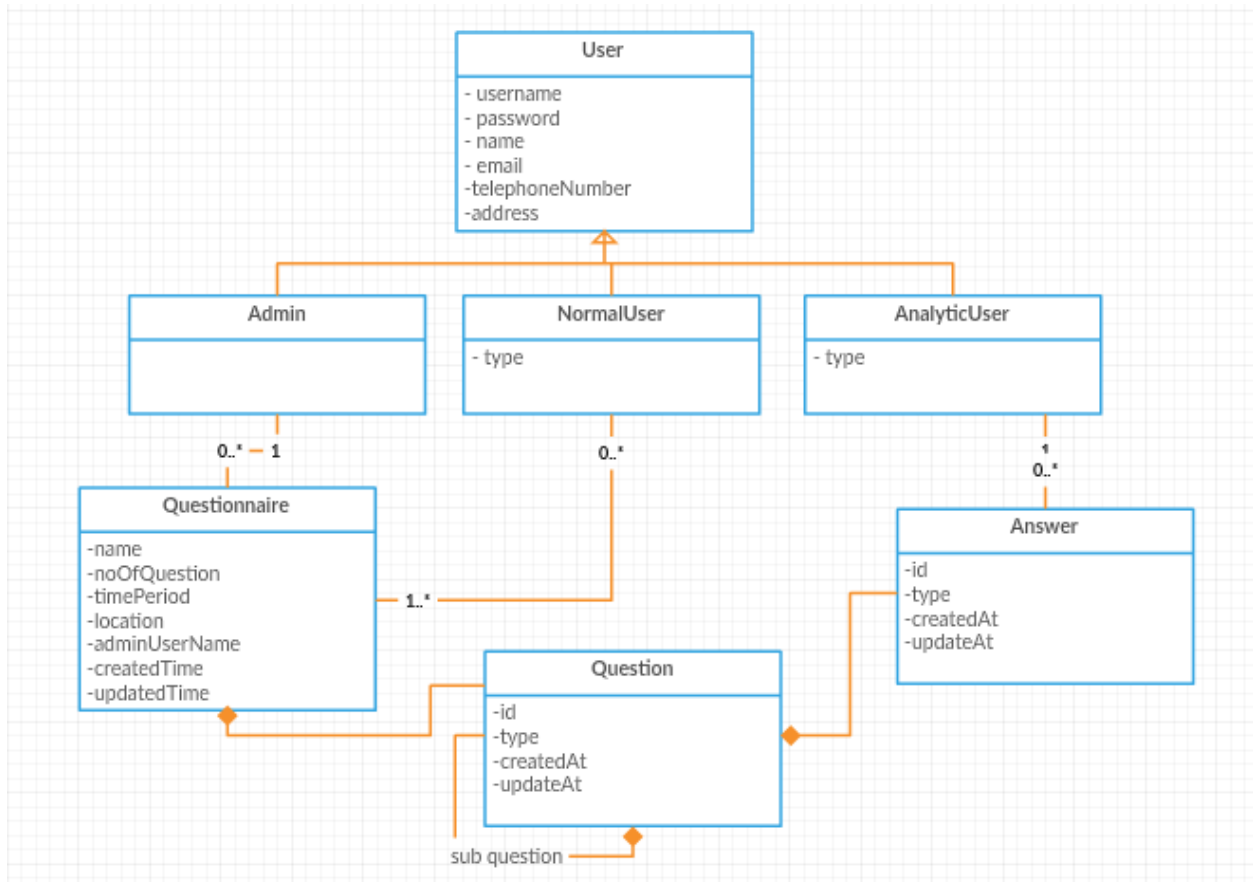
5.1   Overview

The SBS is divided into layers based on the N-tier architecture. The layering model of the SBS application is based on a responsibility layering strategy that associates each layer with a particular responsibility.
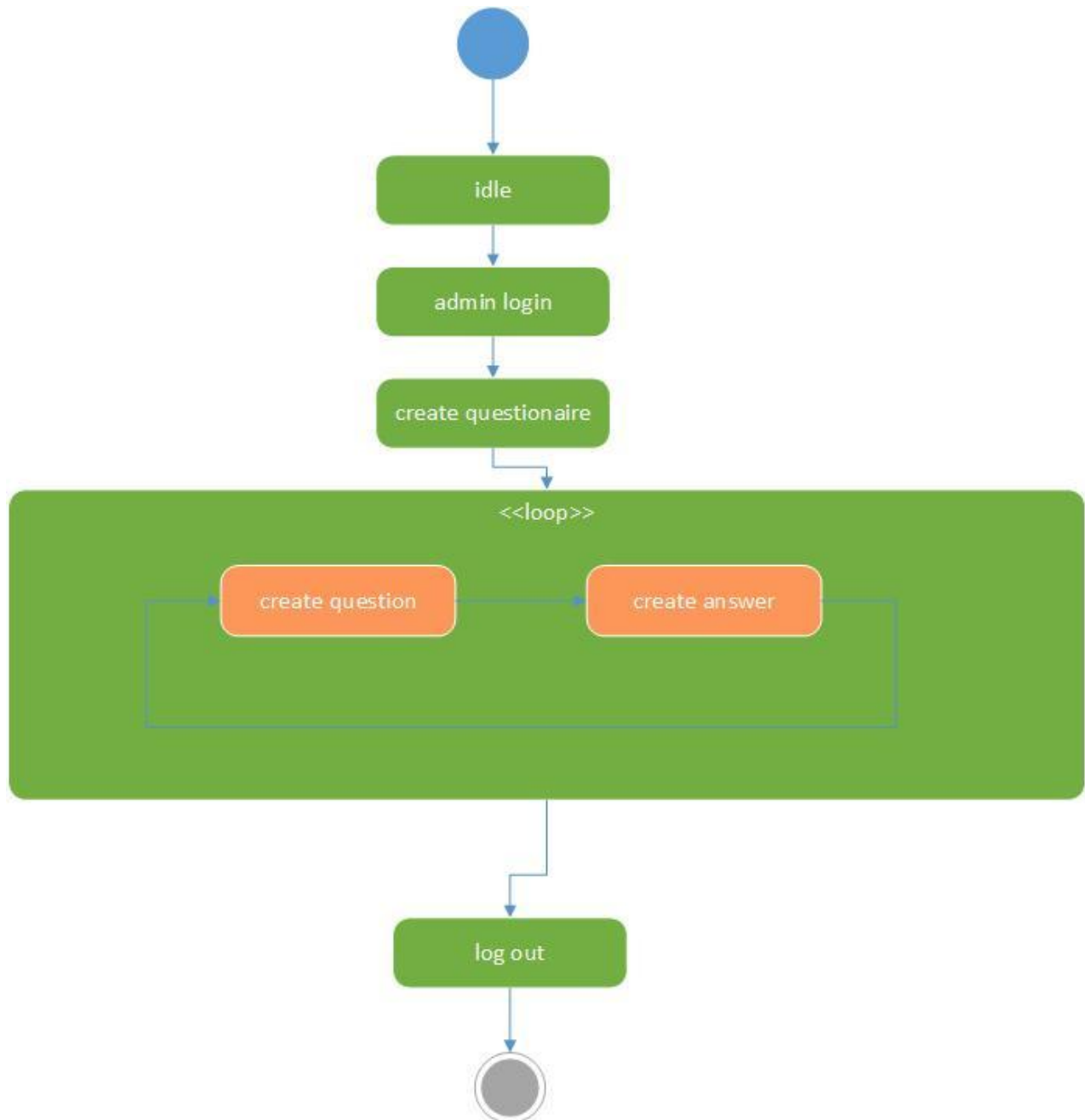
This strategy has been chosen because it isolates various system responsibilities from one another, so that it improves both system development and maintenance.
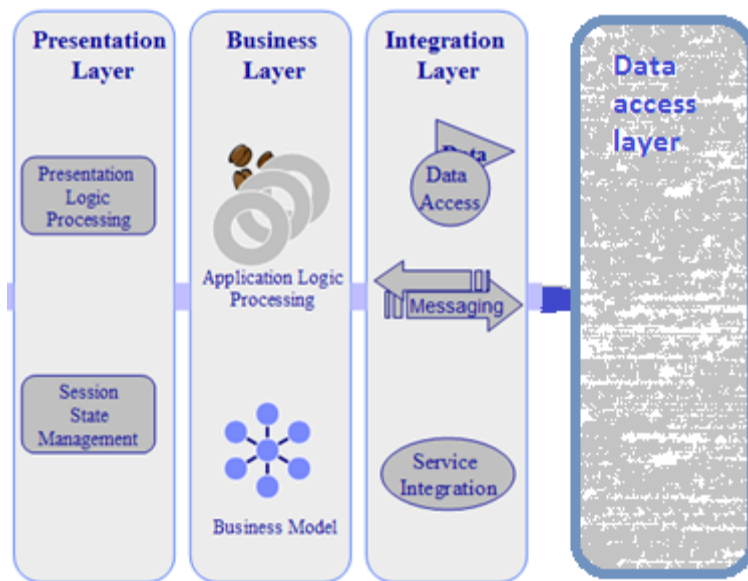
6.   Process View

Below figure showing how admin will create questionnaire and add question into that

7.   Deployment View
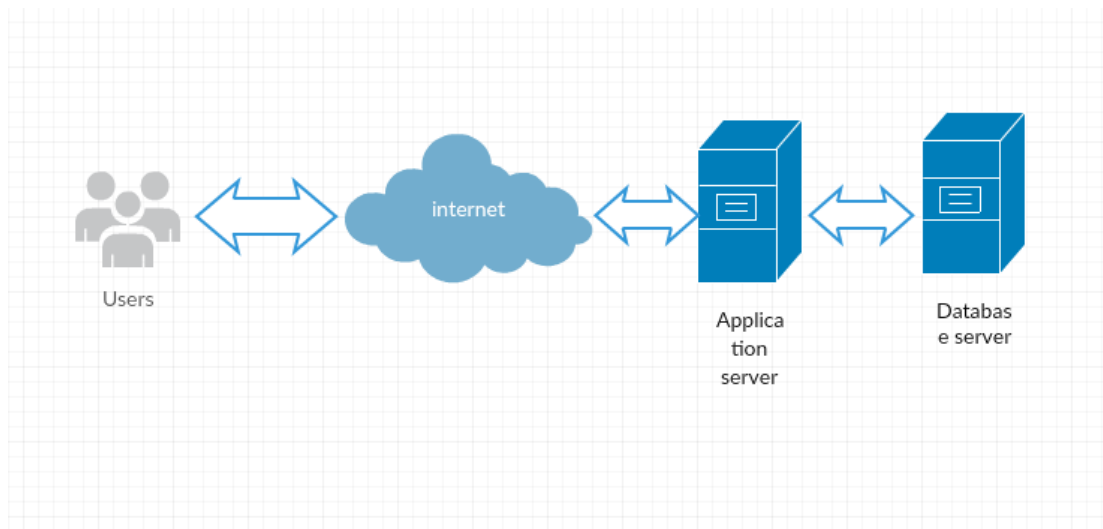
Given above is the high level picture of the architecture of the architecture of the application. Users access the application server through the internet. Web server does the necessary task according the requests.

8. Implementation View

The survey builder system is divided into layers based on the MVC architecture. In this model the application is divided into 3 separate components called model, view and controller. They are interconnected. Controller controls overall application. View get the support from model and controller in order to output to the user. Model is used to save the temporary data of the application.

Model-View-Controller architecture is used for the design of the application architecture, since it can be easily handled in Android. MVC supports flexibility by separation of responsibilities. Following diagram describes the overview of the MVC architecture of the application.
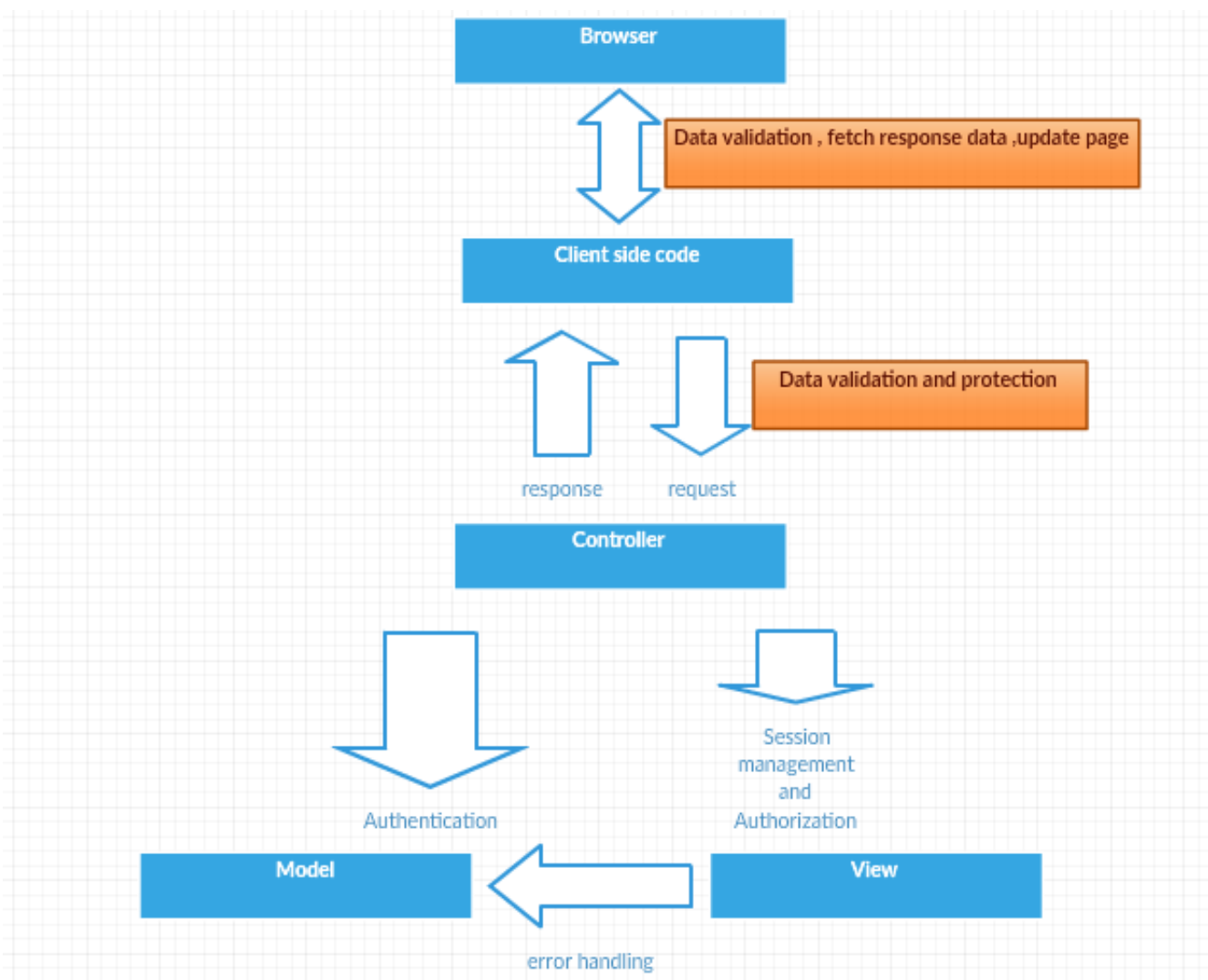
8.1   Overview
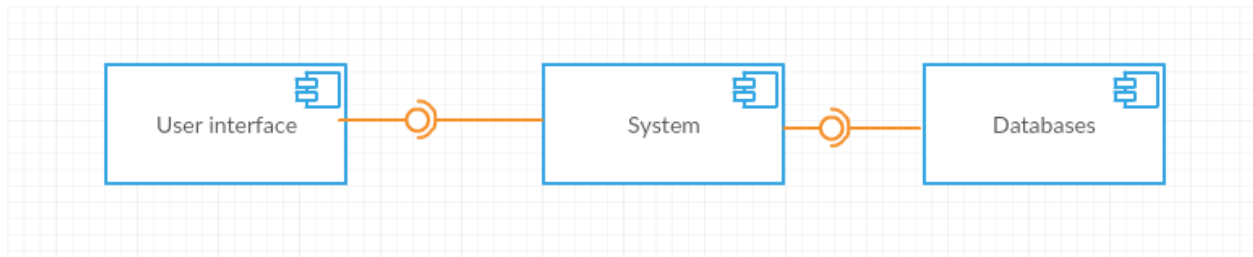
9.   Size and Performance

10.1.1 Size

Currently, the application is expected to handle 1000 users at a time. But in the future, the capacity is expected to increase.

10.2 Performance

Response time for creation of question by admin and answering by normal users will be low, because it is only have database access. But response time for viewing data by analytic user will have response time greater than other users, because it has database access and some processing

10.   Quality

As far as the blood bank management application is concerned, the following quality goals have been identified:

Portability:
As this is a web application. Every devices can access the application via a browser.

Reliability:
All users can use this application without any financial agreement. So it is reliable.

Security:
User's password must be kept secure. In order to maintain this use Authentication and authorization mechanisms