**Student Camp Management**

Chhai C.V., Nem S.T., Chan O.D., Bong C.R.., Khuon S.V.

**CONTENTS**

**Title**

**Abstract 2**

**I Introduction 2**

**II Requirement 2**

* 1. Business Modeling 2
  2. System Requirement Modeling 2
     1. Problem Requirement 2
     2. System Requirement 3

**III Analysis 3**

3.1 Purpose **3**

3.2 Scope **3**

3.3 Risk Management **3**

**IV Planning 4**

4.1 Tree Structure View 4

4.2 Work Breakdown Structure (WBS) 5

**V Design 6**

* 1. MVC Pattern 6
  2. Class Diagram 6

5.3 Data Dictionary 6

5.4 Database Design 6

5.5 ERD 6

5.6 Software Architecture 6

5.7 Interface Design

5.9 Class DAO

**Khmer Slide**

**ABSTRACT**

# *Online document is very popular for students and researcher in this time. But only small amount of document in Khmer language. So we want to help Cambodian people to find document and share their source into internet which built by Khmer. We have a small mission to achieve our aim. That mission is public a small website which refer to Khmer language documents. As a result, we made a smart, responsive, and attractive interface website called “Khmer Slide” which provide many services such as read online, like, share, add to favorite list, user account managing, keep the document in the cloud, uploading, counting document view as rating and user can comment on the document to share their own idea*.

# **I. Introduction**

Nowadays, online learning is one of the most useful source for facilitating in students’ learning. Moreover, online learning is becoming very popular as internet users and the related websites are increasing rapidly in Cambodia. However, there is limitation for Khmer e-learning services for Cambodians students.

Although casting by online materials is very difficult such, most of e-learning service in Cambodia are providing the sources as video tutorials which some Cambodian students need the sources in documents or books to read for their researching, most of Cambodian students are having foreign language limitation so they have had very hard time to understand the sources and there are very less sources in Khmer language in online website. Furthermore, the existing online system are face with more concern in the same manner with data duplication, interface not attractive, messy contents and limited of Khmer document providing. So we decided to create a new online website to guide as a sample one.

Khmer Slide is focus on accessibility of users or visitors. We considered about the absence feature of other websites which. Additionally, we have combined many Khmer sources from many website to make our website contain many documents in Khmer language. Beside this, visitors or users can search the related documents by title or category. Though, users can save the interesting documents in their save list without sharing or downloading.

As result, we have developed our news’ goal successfully which is full fill with modern functionality. We can share our concepts to the visitors or users establish an easy policy.

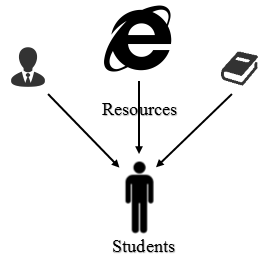
**II. REQUIREMENT**

The requirement section describes how we’re going to accomplish our software and what are needed in order to successfully archive our goal. In this section, there are to sub sections: The Business Modeling and System Requirement Modeling.

**2.1. Business Modeling**

Business Modeling covers the core aspects of business including purpose, business process, target customers, offerings, and strategies of our software according to business requirement. This section involves the understanding of context, which our software will provide. To successful, accomplish a software we have to catch up all the contexts.

Our web-based application called “Khmer Slide” will be used in Korea Software HRD Center in the purpose to help our country improving in education system and making knowledge sharing much more easily by offering several kind of study materials. Thus, the business modeling is about how all the user can daily working action such as can subscribed, can see all document categories, can create save list for documents, can create favorite collection, can comment, reply comment, like, dislike, and give rate to documents, can get notification if they got subscribes or their subscribed upload new documents and admin can controller all the process including: see dashboard, user, category, documents, add/ edit / disable user, add/ edit / disable document, add/ edit / disable category. Our web-based application targets three type of actors such as: Admin, User, anonymous.



**Figure 1. Diagram how the students find resources of knowledge:** There are many ways to support and guide them to get the resources of knowledge such as from people around them like teachers, friends etc., or using internet by researching through the search engines, and they also could find the documents in library or book stores too.

**2.2. System Requirement Modeling  
2.2.1. Problem of Statement**

**2.2. System Requirement Modeling**

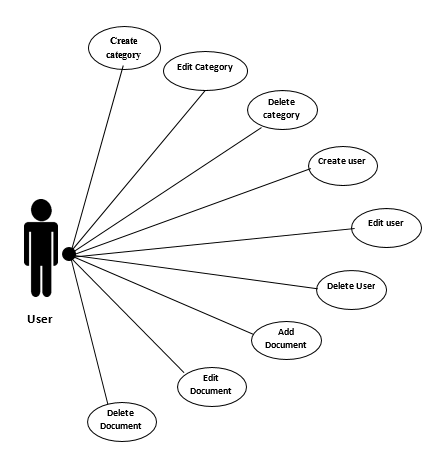
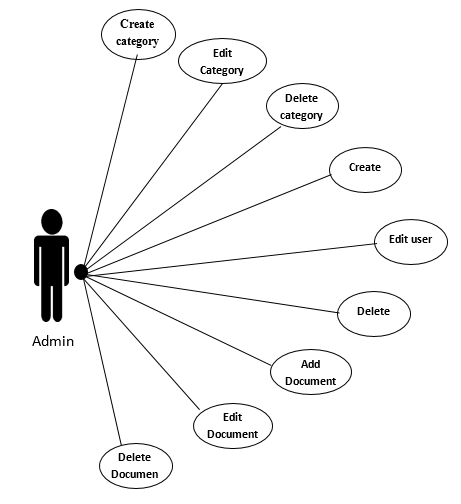
**2.2.1 Problem of Statement**

As we already described in requirement there are many ways to get documents. However, we still feel difficult because:

* We sometime have time to contact the person that has the documents that we want
* We also have to spend much time to get the documents
* We did not know who has the exact documents that we want
* We also did not know that they still have it or deleted it already
* Using Internet, we did not know the exact website address that provide the documents that we want so we have to go through many website to it.
* We also found the duplicate documents which confuse us to download again and again
* Many documents on many website are not trustworthy
* Many document on websites are in English
* Some documents share websites’ interface are not interested and content is messy
* Using Library, we have to spend much time to find the document
* The latest documents is slowly published
* We also spend much more money on it

**2.2.2 System Requirement**

System Requirement are all of the requirements at the system level that describe the functions which the system as a whole should fulfill to satisfy the stakeholder needs and requirements, and is expressed in an appreciate combination of textual statements, views and non-functional requirements.**III. ANALYSIS**



**Figure 2. Use Case Diagram**

Analysis is the process of breaking a complex topic or substance into smaller parts in order to gain a better understanding of it. In other way, it means understanding what we are dealing with in order to design a better solution. To develop a software we need to understand clearly about related components, and their inter-relationships.

**3.1. Purpose**

Khmer Slide is a web based application which design with responsive and easy to access user interface. Our web-based application contains all services about sharing document online. The main purpose of developing Khmer Slide are:

**User:**

* To help user sharing their documents on the internet
* To help them manage their documents on the internet
* To help them browse more Khmer documents on internet.
* To help them find documents faster and easier

**3.2. Scope**

Our group have some scope’s plan to develop this web application such as:

**User:**

* Attractive interface and responsive
* User can upload their documents privately or publicly
* User can view documents online
* User can like, comment, save to save list
* User can print documents

**Admin:**

* Admin can create and manage category
* Admin can manage on user
* Admin can manage on documents

**3.3. Risk Management**

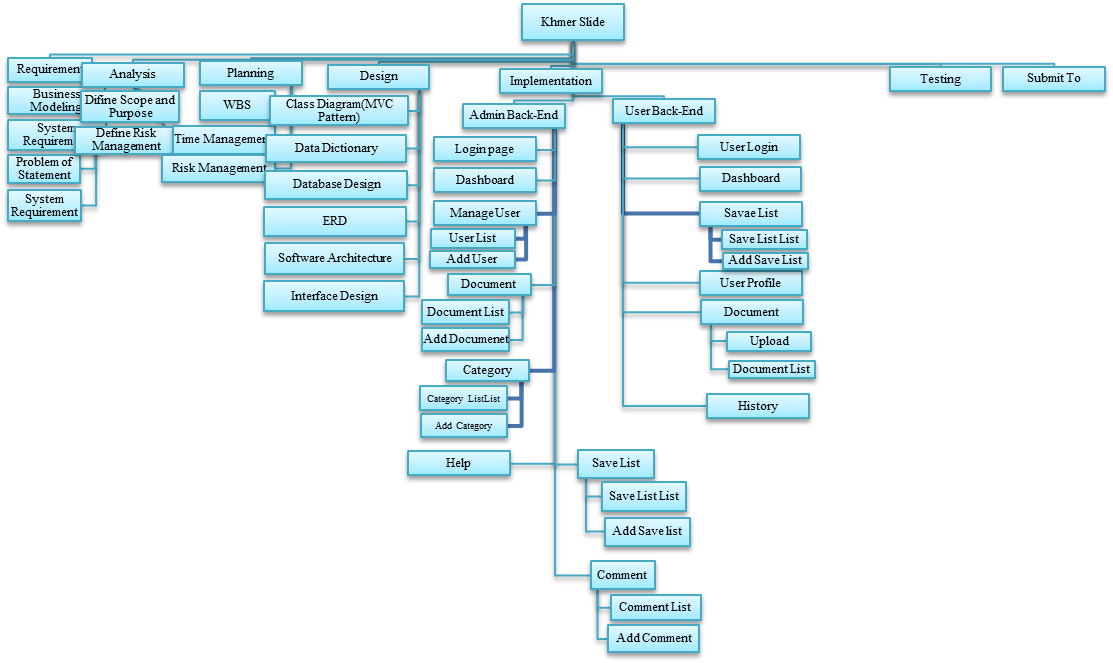
Every big or small projects always have risk, so clear risk identification and management is consider as the most important part of doing project, since clearly understand and analyst the risk will lead the project become successful.

The below table show about some risk that happen in the IT industry, and also happen in our project too:

|  |  |
| --- | --- |
| Risk | Description |
| Lack of resources | Not enough good technical people. |
| Crash project | Is a project where fast is emphasized |
| Special conditions | Law issues |
| Technical factors | No experience with the system(Hardware, Software, application) |

**IV. PLANNING**

**Figure 5.** Tree Structure View, in this picture show about our scope planning.



**4.1. Tree Structure View**

.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Task Name** | **Start Date** | **End Date** | **Assigned To** | **Duration** | **Status** | **Description** |
| **1. Requirement** | **27-Jul-16** | **29-Jul-16** |  | **3** | **100%** | **DONE** |
| **1.1. Business Modeling** | **27-Jul-16** | **28-Jul-16** | **All Members** | **2** |  |  |
| **1.2.System Requirement Modeling** | **27-Jul-16** | **29-Jul-16** | **All Members** | **3** |  |  |
| **1.2.1. Problem of Statement** | **27-Jul-16** | **28-Jul-16** | **All Members** | **2** |  |  |
| **1.2.1. System Requirement** | **28-Jul-16** | **29-Jul-16** | **All Members** | **2** |  |  |
| **2. Analysis** | **29-Jul-16** | **30-Jul-16** |  | **2** | **100%** | **DONE** |
| **2.1. Define Scope and Purpose** | **29-Jul-16** | **30-Jul-16** | **All Members** | **2** |  |  |
| **2.2. Define Risk Management** | **29-Jul-16** | **30-Jul-16** | **All Members** | **2** |  |  |
| **3. Planning** | **30-Aug-16** | **1-Aug-16** |  | **3** | **100%** | **DONE** |
| **3.1. Work Breakdown Structure** | **30-Jul-16** | **31-Aug-16** | **All Members** | **2** |  |  |
| **3.2. Time Management** | **31-Aug-16** | **1-Aug-16** | **All Members** | **2** |  |  |
| **3.3. Risk Management** | **31-Aug-16** | **1-Aug-16** | **All Members** | **2** |  |  |
| **4. Design** | **1-Aug-16** | **5-Aug-16** |  | **5** | **100%** | **DONE** |
| **4.1. Class Diagram(MVC Pattern)** | **1-Aug-16** | **2-Aug-16** | **All Members** | **2** |  |  |
| **4.2. Data Dictionary** | **1-Aug-16** | **2-Aug-16** | **All Members** | **2** |  |  |
| **4.3. Database Design** | **1-Aug-16** | **5-Aug-16** | **All Members** | **5** |  |  |
| **4.4. ERD** | **1-Aug-16** | **5-Aug-16** | **All Members** | **5** |  |  |
| **4.5. Software Architecture** | **1-Aug-16** | **3-Aug-16** | **All Members** | **3** |  |  |
| **4.6. Interface Design** | **1-Aug-16** | **5-Aug-16** | **All Members** | **5** |  |  |
| **5. Implementation** | **5-Aug-16** | **24-Aug-16** |  | **19** | **100%** | **DONE** |
| **5.1. Admin Back-End** | **5-Aug-16** | **13-Aug-16** | **All Members** | **9** | **100%** | **DONE** |
| **5.1.1. Login page** | **5-Aug-16** | **10-Aug-16** | **Chivon** | **5** |  |  |
| **5.1.2. Dashboard** | **5-Aug-16** | **10-Aug-16** | **Sothea** | **5** |  |  |
| **5.1.3. Manage User** | **6-Aug-16** | **10-Aug-16** | **Odom** | **5** |  |  |
| **5.1.3.1. User List** | **6-Aug-16** | **10-Aug-16** | **Odom** | **5** |  |  |
| **5.1.3.2. Add User** | **6-Aug-16** | **10-Aug-16** | **Odom** | **5** |  |  |
| **5.1.4. Document** | **7-Aug-16** | **13-Aug-16** | **Sothea** | **7** |  |  |
| **5.1.4.1. Document List** | **7-Aug-16** | **13-Aug-16** | **Sothea** | **7** |  |  |
| **5.1.4.2. Add Document** | **7-Aug-16** | **13-Aug-16** | **Sothea** | **7** |  |  |
| **5.1.5. Category** | **7-Aug-16** | **13-Aug-16** | **Rith, Tey** | **7** |  |  |
| **5.1.5.1 Category List** | **7-Aug-16** | **13-Aug-16** | **Rith, Tey** | **7** |  |  |
| **5.1.5.2 Add Category** | **7-Aug-16** | **13-Aug-16** | **Rith, Tey** | **7** |  |  |
| **5.1.6. Help** | **7-Aug-16** | **8-Aug-16** | **Tey** | **2** |  |  |
| **5.1.7. Save List** | **7-Aug-16** | **13-Aug-16** | **Chivon** | **7** |  |  |
| **5.1.7.1. Save List List** | **7-Aug-16** | **13-Aug-16** | **Chivon** | **7** |  |  |
| **5.1.7.2. Add Save List** | **7-Aug-16** | **13-Aug-16** | **Chivon** | **7** |  |  |
| **5.1.8. Comment** | **10-Aug-16** | **13-Aug-16** | **Odom,Chivon,Sothea** | **4** |  |  |
| **5.1.8.1. Comment List** | **10-Aug-16** | **13-Aug-16** | **Odom,Chivon,Sothea** | **4** |  |  |
| **5.1.8.2. Add Comment** | **10-Aug-16** | **13-Aug-16** | **Odom,Chivon,Sothea** | **4** |  |  |
| **5.2. User Back-End** | **13-Aug-16** | **25-Aug-16** | **All Members** | **13** |  |  |
| **5.2.1. User Login** | **13-Aug-16** | **22-Aug-16** | **Chivon** | **10** |  |  |
| **5.2.2. Dashboard** | **13-Aug-16** | **22-Aug-16** | **Odom** | **10** | **100%** | **DONE** |
| **5.2.3. User Profile** | **13-Aug-16** | **22-Aug-16** | **Sothea** | **10** |  |  |
| **5.2.4. Document** | **13-Aug-16** | **22-Aug-16** | **Odom, Rith** | **10** |  |  |
| **5.2.4.1. Upload Document** | **13-Aug-16** | **22-Aug-16** | **Odom** | **10** |  |  |
| **5.2.4.2. Document List** | **13-Aug-16** | **22-Aug-16** | **Rith** | **10** |  |  |
| **5.2.5. Save List** | **13-Aug-16** | **22-Aug-16** | **Chivon** | **10** |  |  |
| **5.2.5.1. Save List List** | **13-Aug-16** | **22-Aug-16** | **Chivon** | **10** |  |  |
| **5.2.5.2. Add Save List** | **13-Aug-16** | **22-Aug-16** | **Chivon** | **10** |  |  |
| **5.2.6. History** | **22-Aug-16** | **25-Aug-16** | **Odom** | **4** |  |  |
| **6. Testing** | **25-Aug-16** | **25-Aug-16** | **All Members** | **1** |  |  |
| **7. Submit Assignment** | **26-Aug-16** | **26-Aug-16** | **HRD Center** |  |  |  |
| **Totally:** | | | | **30days** | | |

**4.2. Work Breakdown Structure (WBS)**

**Figure4:** Work Breakdown Structure, show about our scope and time that our group have defined.

**V. DESIGN**

**5.1. MVC Pattern**

Model

View

Controller

Dispatcher

Routes

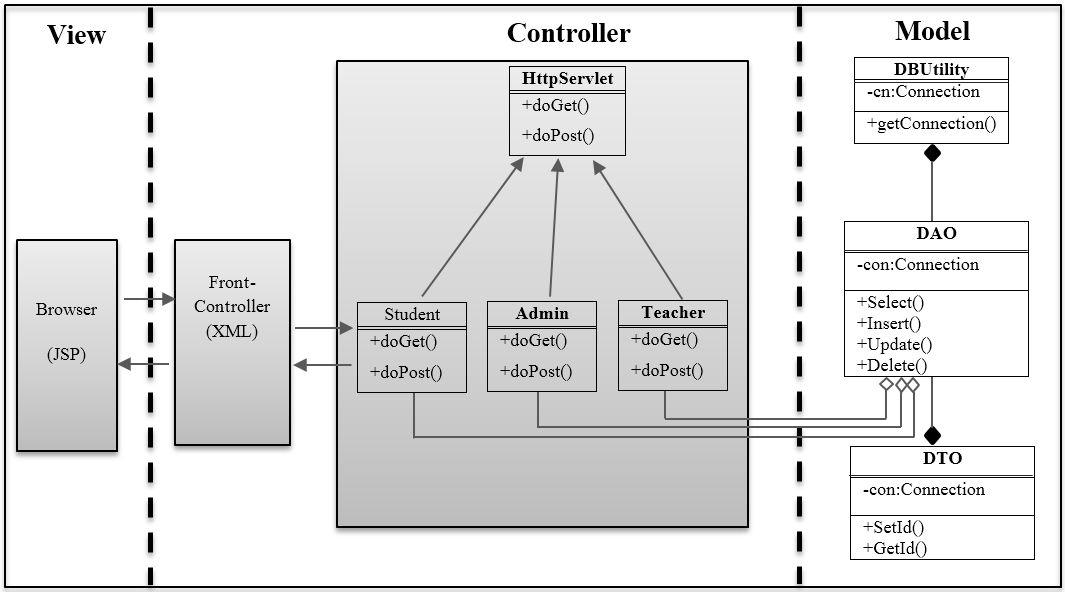
PostgreSQL

Web Server

Browser

**Figure5:** MVC Pattern

**5.2. Class Diagram**



**Figure6:** Class Diagram

* 1. **Data Dictionary**

Table1: ksl\_save\_list

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Col. Name | Type | Not Null | Pri-Key | Refer. | Default |
| sl\_id | SERIAL | Yes | Yes |  |  |
| sl\_name | TEXT |  |  |  |  |
| saved\_date | TEXT |  |  |  |  |
| status | INTEGER |  |  |  |  |
| user\_id | INTEGER |  |  | ksl\_user(user\_id) |  |
| doc\_id | INTEGER |  |  | ksl\_document(doc\_id) |  |
| description | TEXT |  |  |  |  |

Table 2: ksl\_category

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Col. Name | Type | Not Null | Pri-Key | Refer. | Default |
| cat\_id | SERIAL | Yes | Yes |  |  |
| parent\_id | INTEGER |  |  | ksl\_category(cat\_id) |  |
| cat\_name | TEXT |  |  |  |  |
| created\_date | TEXT |  |  |  |  |
| status | INTEGER |  |  |  |  |
| user\_id | INTEGER |  |  | ksl\_user(user\_id) |  |
| description | VARCHAR |  |  |  |  |
| Icon | VARCHAR |  |  |  |  |

Table 3: ksl\_view\_history

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Col. Name | Type | Not Null | Pri-Key | Refer. | Default |
| vh\_id | SERIAL | Yes | Yes |  |  |
| add\_date | TEXT |  |  |  |  |
| user\_id | INTEGER |  |  | ksl\_user(user\_id) |  |
| doc\_id | INTEGER |  |  | ksl\_document(doc\_id) |  |
| description | TEXT |  |  |  |  |
| status | INTEGER |  |  |  |  |

Table4: ksl\_comment

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Col. Name | Type | Not Null | Pri-Key | Refer. | Default | |
| cmt\_id | SERIAL | Yes | Yes |  |  |
| cmt\_text | VARCHAR |  |  |  |  |
| cmt\_date | TEXT |  |  |  |  |
| status | INTEGER |  |  |  |  |
| user\_id | INTEGER |  |  | ksl\_user(user\_id) |  |
| doc\_id | INTEGER |  |  | ksl\_document(doc\_id) |  |
| description | TEXT |  |  |  |  |

Table5: ksl\_user\_type

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Col. Name | Type | Not Null | Pri-Key | Refer. | Default |
| role\_id | SERIAL | Yes | Yes |  |  |
| role\_name | TEXT |  |  |  |  |
| status | INTEGER |  |  |  |  |
| description | TEXT |  |  |  |  |

Table6: ksl\_user

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Col. Name | Type | Not Null | Pri-Key | Refer. | Default |
| user\_id | SERIAL | Yes | Yes |  |  |
| user\_name | TEXT |  |  |  |  |
| gender | TEXT |  |  |  |  |
| email | TEXT |  |  |  |  |
| password | TEXT |  |  |  |  |
| registered\_date | TEXT |  |  |  |  |
| photo | TEXT |  |  |  |  |
| description | TEXT |  |  |  |  |
| status | INTEGER |  |  |  |  |
| role\_id | INTEGER |  |  | ksl\_user\_type(role\_id) |  |

Table7: ksl\_doc\_type

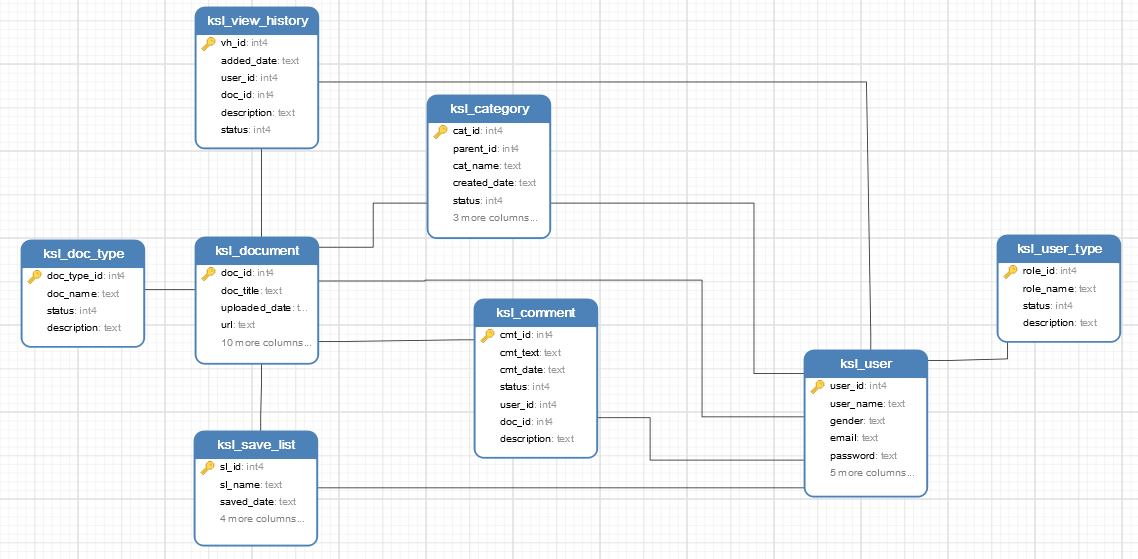
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Col. Name | Type | Not Null | Pri-Key | Refer. | Default |
| doc\_type\_id | SERIAL | Yes | Yes |  |  |
| doc\_name | TEXT |  |  |  |  |
| status | INTEGER |  |  |  |  |
| description | TEXT |  |  |  |  |

Table8: ksl\_document

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Col. Name | Type | Not Null | Pri-Key | Refer. | Default |
| doc\_id | SERIAL | Yes | Yes |  |  |
| doc\_title | TEXT |  |  |  |  |
| Uploaded\_date | TEXT |  |  |  |  |
| url | TEXT |  |  |  |  |
| Liked | INTEGER |  |  |  |  |
| Shared | INTEGER |  |  |  |  |
| Viewed | INTEGER |  |  |  |  |
| Description | TEXT |  |  |  |  |
| Status | INTEGER |  |  |  |  |
| Doc\_type\_id | INTEGER |  |  | ksl\_doc\_type(doc\_type\_id) |  |
| User\_id | INTEGER |  |  | ksl\_user(user\_id) |  |
| Cat\_id | INTEGER |  |  | ksl\_category(cat\_id) |  |
| Thumbnail | TEXT |  |  |  |  |
| source | TEXT |  |  |  |  |

**5.4. ERD Diagram**

**Figure 7.** Relational Schema

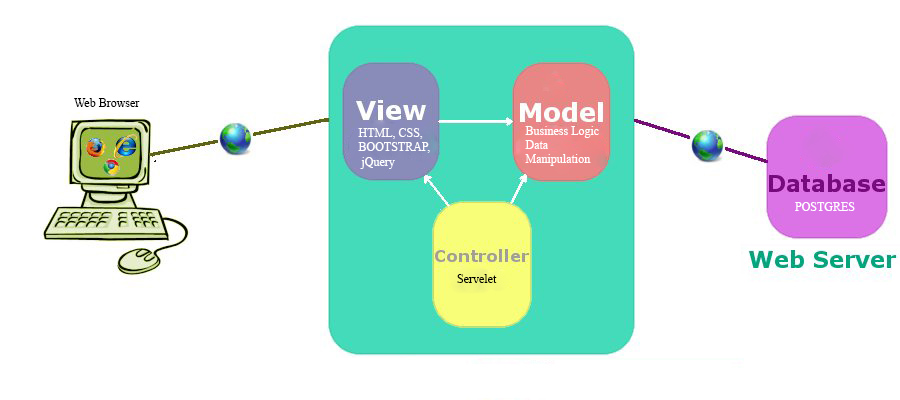


**5.5. ERD Diagram**

****

**Figure 8.** ERD Diagram

**5.6. Software Architecture**

****

**Figure11:** Software Architecture

Our application is based on MVC (Model, View, and Controller) design pattern. The model-view-controller pattern proposes three main components or objects to be used in software development:

A Model, which represents the underlying, logical structure of data in a software application and the high-level class associated with it. This object model does not contain any information about the user interface.

A View, which is a collection of classes representing the elements in the user interface (all of the things the user can see and respond to on the screen, such as buttons, display boxes, and so forth)

A Controller, which represents the classes connecting the model and the view, and is used to communicate between classes in the model and view.

As in the above figure, descript the flow of MVC model architecture.

* Web Browser sends request for JSP pages
* JSP pages access to Java classes to invoke the business logic and data manipulation
* Java classes interacts with data source(database) and retrieve or manipulation data
* The Controller response back to the browser

**SELECTED TECHNOLOGY:**

**Server**

* **Spring Boot**: use in MVC pattern as Controller which to communicate with Model and View. It is used to request to and respond from server.

**Client**

* **JSP:** use in MVC pattern as View, which we can use Java code embed with HTML code.
* **HTML:** is a markup language that we can use it to make a simple static website.
* **CSS:** we use CSS for decorating of our web application such as setting color of text and background, align text, and format text, and so on.
* **Bootstrap:** it is used to make our web application more interesting because it provides many styles.
* **JavaScript and jQuery:** they are scripting languages, which we can use to embed with HTML, and CSS to create dynamic websites run on client side.

**Database:**

* + **PostgresSQL**: we use postgresSQL as DBMS to store, retrieve, and manipulate data.

**5.7. Interface Design**

**5.8. Sources Code**

**5.9. Class DAO**