The Ministry of Education of the Republic of Belarus

The institution of higher education

"The Belarusian State University

OF Informatics and Radio Electronics"

Faculty of Information Technology and Control

The department of information technology of automated systems

EXPLANATORY NOTE

to the course project

of the discipline "Computer Aided Software Systems Design"

on the topic of:

**The automated system for visitors of Cafe**

Completed by: Student of gr. 920611 Li Boyi

The Course-project manager: Senior lecturer of the ITAS Dpt. Khajynova N.V.

Minsk 2022

**contents**

[Introduction 3](#_Toc103774010)

[1 Analysis of subject area 4](#_Toc103774011)

[1.1 History of E-learning 4](#_Toc103774012)

[1.2 E-learning performance in China 6](#_Toc103774013)

[1.3 Aim and objectives of my E-learning system 7](#_Toc103774014)

[1.4 Comparison with XuetangX Online 7](#_Toc103774015)

[1.5 The required environment of this project 8](#_Toc103774016)

[2 E-learning structure and design 10](#_Toc103774017)

[2.1 Overall design 10](#_Toc103774018)

[2.2 Database design 18](#_Toc103774019)

[3 E-learning development and implementation 23](#_Toc103774020)

[3.1 Existing programming tools for web development 23](#_Toc103774021)

[3.2 Programming tools used 25](#_Toc103774022)

[3.3 SSM framework 28](#_Toc103774023)

[3.4 Project implementation description 31](#_Toc103774024)

[4 User guide 39](#_Toc103774025)

[Conclusion 43](#_Toc103774026)

[Reference 44](#_Toc103774027)

[Appendix A 45](#_Toc103774028)

**INTRODUCTION**

With the development of the Internet, the education industry promotes distance education, and realizes remote video teaching and electronic document sharing through the Internet virtual classroom, so that teachers and students can form an interaction of teaching and learning on the network; The advent of 5G era makes it more convenient to learn not only through bulky computers, but also through a mobile phone with large traffic. With the rapid promotion of 5G network, we can learn online more conveniently and directly through handheld tools such as mobile phones, and the wireless network makes people's daily interaction more effective!

With the advent of knowledge economy, our learning mode has been impacted unprecedentedly, and various new learning modes have emerged like a tide. Among all learning modes, the most powerful one is network-based learning with the development of network technology, also known as online learning. It is through the establishment of online education platform, It is a new way for students to apply the network for online learning. This online learning method is a new learning environment composed of multimedia network learning resources, online learning community and network technology platform. Compared with other learning modes, it has incomparable advantages.

One advantage of online learning is that it is not limited by time, place and space, and can realize the same interaction as in reality.

1. ANALYSIS OF SUBJECT AREA

**1.1 History of E-Learning**

With the rapid development of information technology, especially from the Internet to mobile Internet, a cross time and space way of life, work and learning has been created, which has fundamentally changed the way of knowledge acquisition. Teaching and learning can be free from the restrictions of time, space and place, and the channels of knowledge acquisition are flexible and diversified.

A Bangladeshi American named Salman Khan founded Khan college in the United States. Khan alone produced more than 2300 teaching videos on mathematics, physics, chemistry, biology, astronomy and other subjects. He used video technology to reform traditional teaching methods, provide free high-quality education to the world, and create a teacher One computer can attract tens of millions of students. 56 million primary and secondary school students around the world watch his teaching videos, and 6 million students log on to the website every month.

Khan's teaching video highlights the process of "learning" and triggers a "flipped classroom" teaching model. The traditional teaching mode is that teachers give lectures in class, assign homework, and let students go home to practice and do homework. "Flipped classroom" is a place for students to complete the learning of Khan teaching video at home, and the classroom has become a place for interaction between teachers and students and between students and students, including question answering, the application of knowledge and teamwork, so as to achieve better educational results. Some studies have shown that this personalized learning method can greatly improve students' learning efficiency. Two schools in California have tried the teaching video of "Khan College" in Grade 5 and grade 7. Students watch the video at home for self-study, while teachers answer questions in class and help students finish their homework. In 2013, the curriculum of Khan college will be adopted by more than 20 public schools in the United States. In April 2012, Time magazine rated the list of 100 people who influenced the world in 2012. Salman Khan ranked fourth. He has been recognized as a superstar in the world of teachers.

Today, more and more online education platforms have been successful, such as Coursera, edX, xuetangX, etc.

Coursera Inc. is a U.S.-based massive open online course provider founded in 2012by Stanford University computer science professors Andrew Ng and Daphne Koller.Coursera works with universities and other organizations to offer online courses, certifications, and degrees in a variety of subjects. In 2021 it was estimated that about 150 universities offered more than 4,000 courses through Coursera. Coursera website interface can be seen in Figure 1.1.

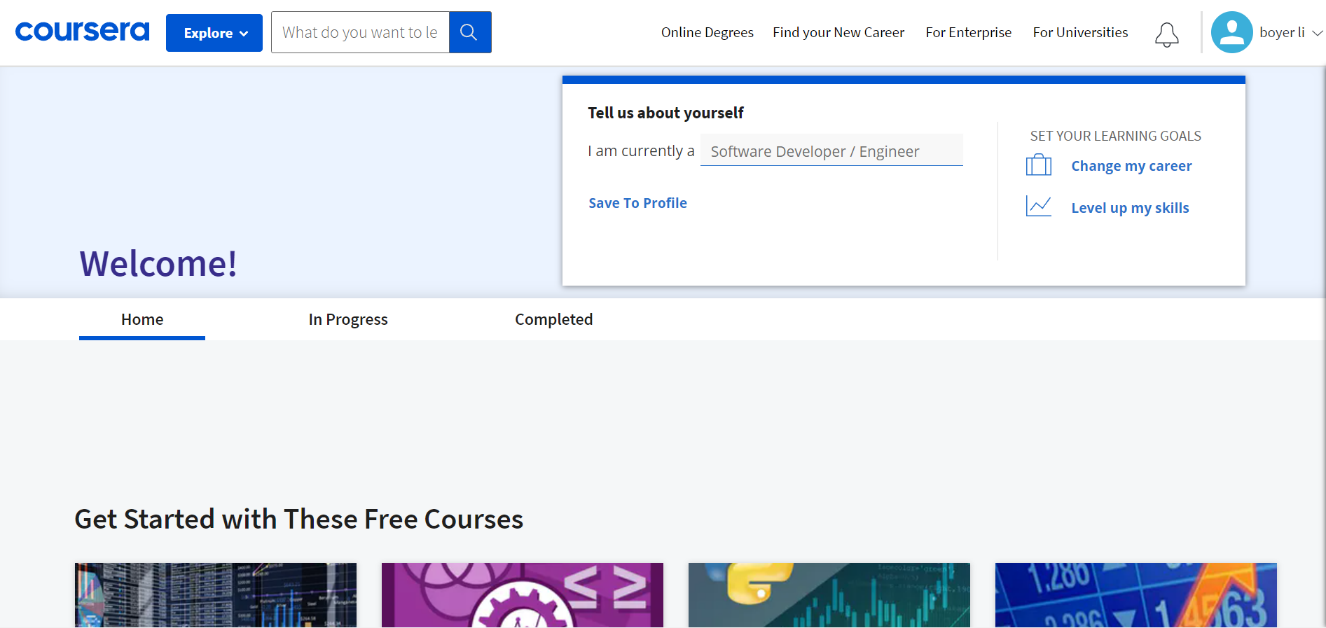


Figure 1.1 – Coursera website interface

edX is an American massive open online course (MOOC) provider created by Harvard and MIT. It hosts online university-level courses in a wide range of disciplines to a worldwide student body, including some courses at no charge. It also conducts research into learning based on how people use its platform. edX runs on the free Open edX open-source software platform. 2U is the parent company, with edX operating as its global online learning platform and primary brand for products and services. edX website interface can be seen in Figure 1.2.

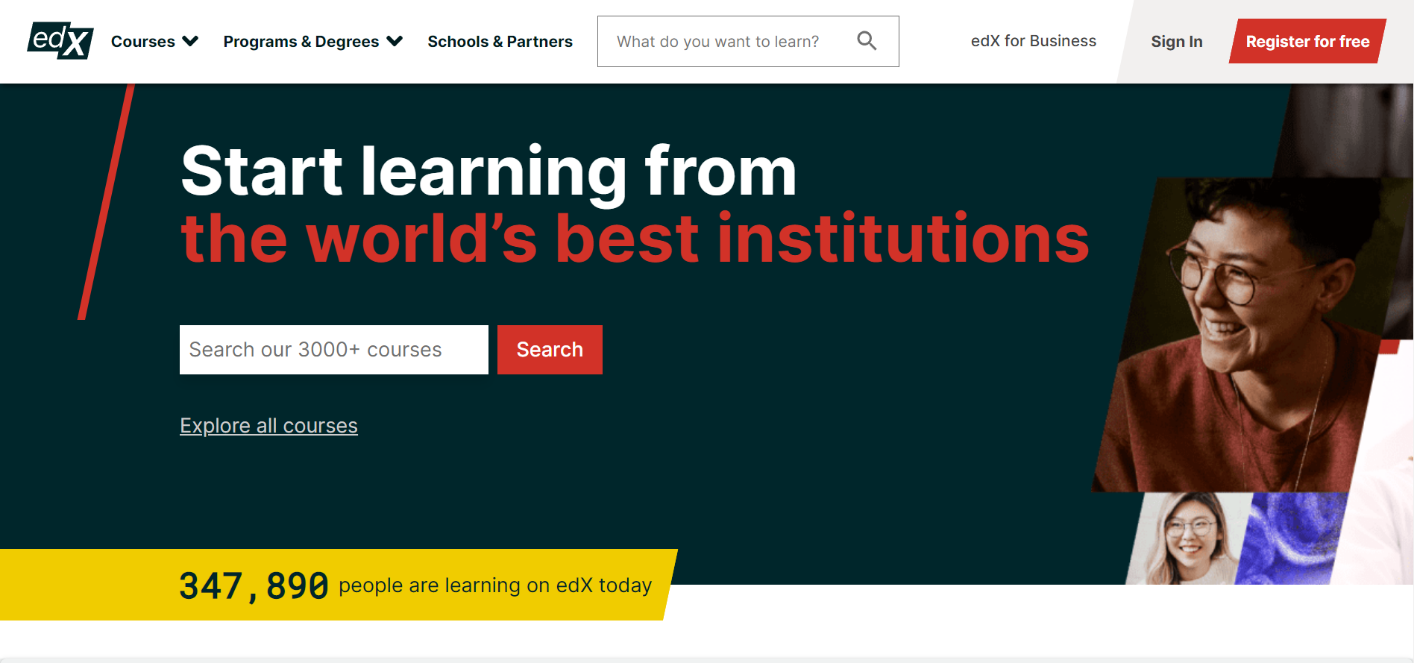


Figure 1.2 – edX website interface

XuetangX is a massive open online course (MOOC) platform. It offers online courses in multiple disciplines and also certificate and degree programs. Launched on 10 October 2013 as the first Chinese MOOC platform, XuetangX was initiated by Tsinghua University and MOE Research Center for Online Education, for the purposes of communication and the application of research in the field of online education. XuetangX website interface can be seen in Figure 1.3.



Figure 1.3 – XuetangX website interface

**1.2 E-Learning performance in China**

According to statistics, only in the epidemic stage, 63.1% of online users in China have purchased knowledge payment products. Online education is favored by many platforms. All kinds of offline educational institutions transform online and constantly develop sinking markets, which makes the online education market expand infinitely. Therefore, the development prospect of all kinds of online curriculum systems will also be infinitely possible.

The current mainstream E-Learning systems in China include XuetangX, Chao Xing, icourse163, etc. Among them, XuetangX is the most influential.

XuetangX Online operates more than 3000 high-quality courses from Tsinghua University, Peking University, Fudan University, University of Science and Technology of China, as well as MIT, Stanford University, University of California, Berkeley and other first-class universities at home and abroad, covering 13 university disciplines.

In addition, XuetangX has also developed an international version, bringing excellent educational resources to the world.

The international version of XuetangX is positioned to bring together first-class online courses at home and abroad. First, it supports Chinese and English, and later will support languages such as Russian, Spanish, French, and Japanese. The platform supports a variety of online education forms such as MOOCs, live broadcasts, certificates, and online degrees. It will focus on exploring first-class courses from Chinese universities, while taking into account the introduction of first-class courses from global and regional top universities. According to the plan, the first batch of 109 courses will be launched in the international version of XuetangX, and the introduction and launch of subsequent courses will continue to be promoted.

**1.3 Aim and Objectives of my E-Learning System**

The main purpose of the design of the E-Learning system is to provide an out-of-class learning platform for students in school, mainly for students and teachers. In the system, teachers can publish course-related assignments and upload course-related materials. Wait. And users can also learn any subject of interest through the system and complete related exercises.

And the system is online, so no implementation is required. It can be accessed from anywhere over the internet. The software overcomes existing manual processes.

**1.4 Comparison with XuetangX Online**

Compared with XuetangX, our system has the advantage of providing the function of answering quizzes and providing materials that focus on specific areas.

In addition, our system has also added a point reward system, and functions such as points redemption and gifts can be launched in the future, so as to improve users' enthusiasm for learning. The Points interface can be seen in Figure 1.4.

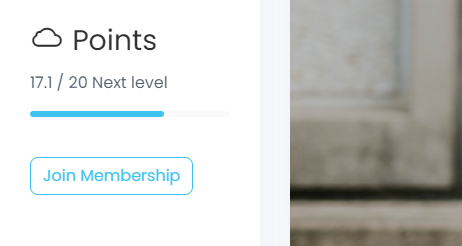


Figure 1.4 – Points interface

Users can view the points, time, and source of points they have earned on the personal points page below (Figure 1.5).

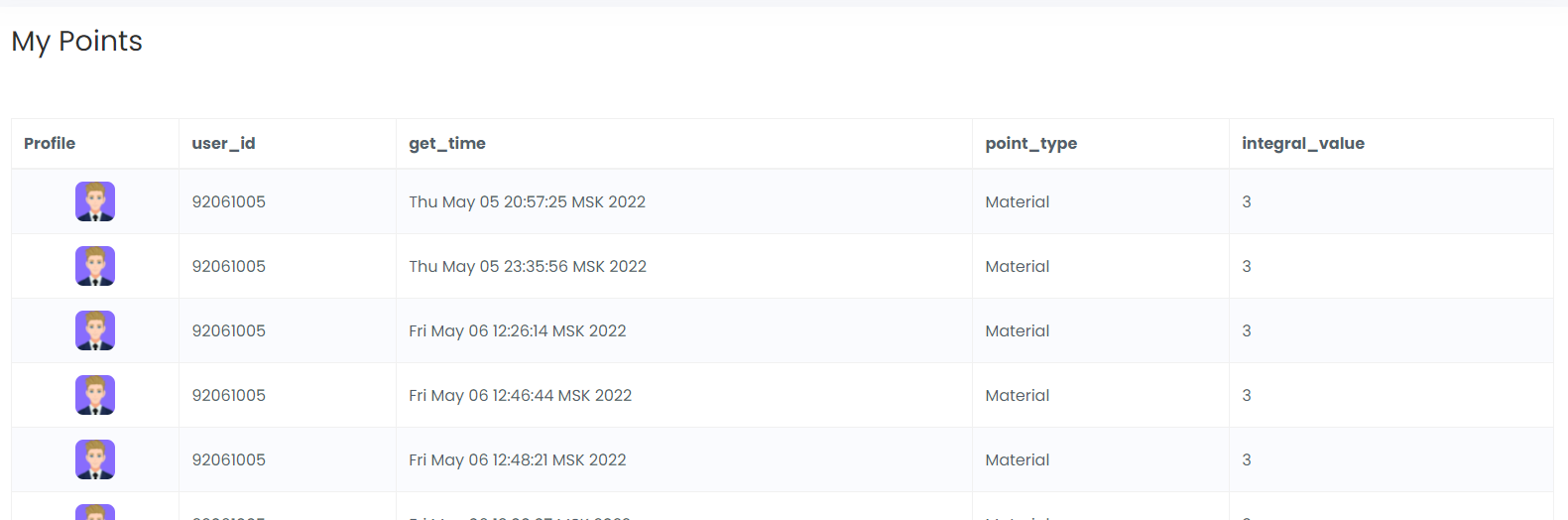


Figure 1.5 – PersonalPoints interface

**1.5 The required environment of this project**

Hardware to install E-Learning System: CPU 1 GHz, RAM 1 GB, HDD 100 MB+, VRAM 256 MB+, OS: Windows 7+, DBMS: MYSQL 5.6+, web-browser: IE 10+, Google Chrome 50+.

**Hardware Requirement:**

1. i3 Processor Based Computer or higher
2. Memory: 1 GB RAM
3. Hard Drive: 50 GB
4. Monitor
5. Internet Connection

**Software Requirement:**

1. Windows 7 or higher
2. IntelliJ Idea.
3. MYSQL 8.0
4. E-LEARNING STRUCTURE AND DESIGN

**2.1 Overall Design**

**2.1.1** Organizational Structure and Design of E-learning System (Figure 2.1)

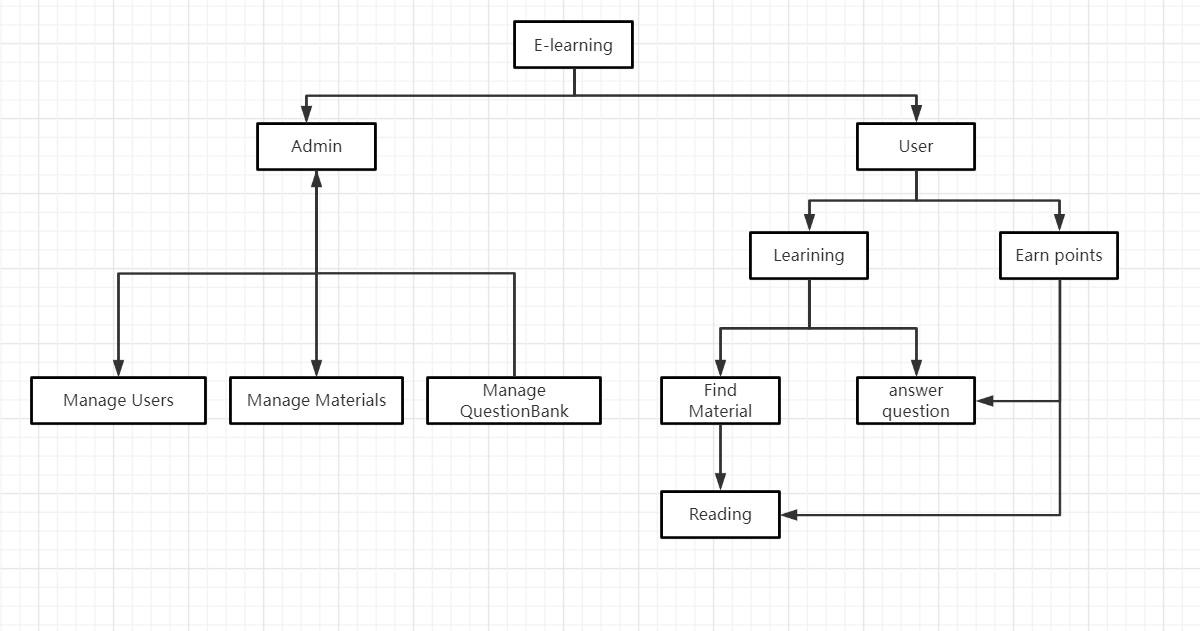


Figure 2.1 – Organizational Structure of E-learning System

This configuration shows that the online learning system is divided into two subsystems. The first is administrator system, administrators can manage information, users, question banks, etc., users can learn online can pass the answers and access information.

**2.1.2** Use Case Diagram

A use case diagram is a graphic depiction of the interactions among the elements of a system. A use case is a methodology used in system analysis to identify, clarify, and organize system requirements. In this context, the term "system" refers to something being developed or operated, such as a mail-order product sales and service Web site. Use case diagrams are employed in UML (Unified Modeling Language), a standard notation for the modeling of real-world objects and systems (figure 2.2).

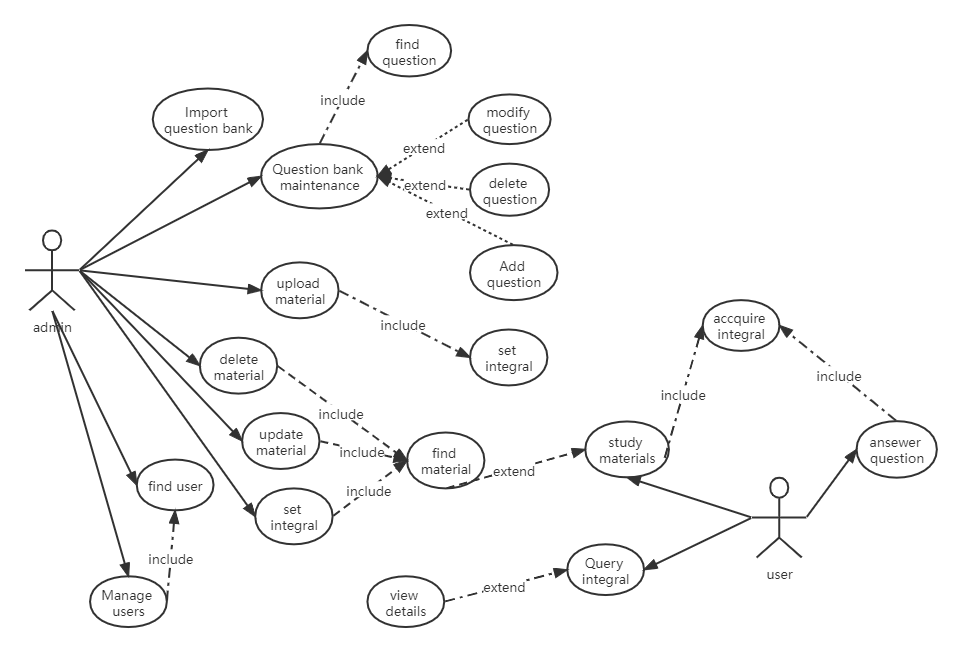


Figure 2.2 – Use Case of E-learning

The administrator can upload the question bank, maintain the question bank, upload data, delete data, update data, set questions and data points, add users, manage users, etc. Users can learn materials, obtain points, query points and answer questions.

**2.1.3** Sequence Diagram

A sequence diagram is an interaction diagram that shows how objects operate with one another and in what order. It is a construct of a message sequence chart. A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the Logical View of the system under development. Sequence diagrams are sometimes called event diagrams or event scenarios.

A sequence diagram shows, as parallel vertical lines (lifelines), different processes or objects that live simultaneously, and as horizontal arrows, the messages exchanged between them, in the order in which they occur. This allows the specification of simple runtime scenarios in a graphical manner.

The following is a sequence diagram for answering questions (Figure 2.3 to Figure 2.6):

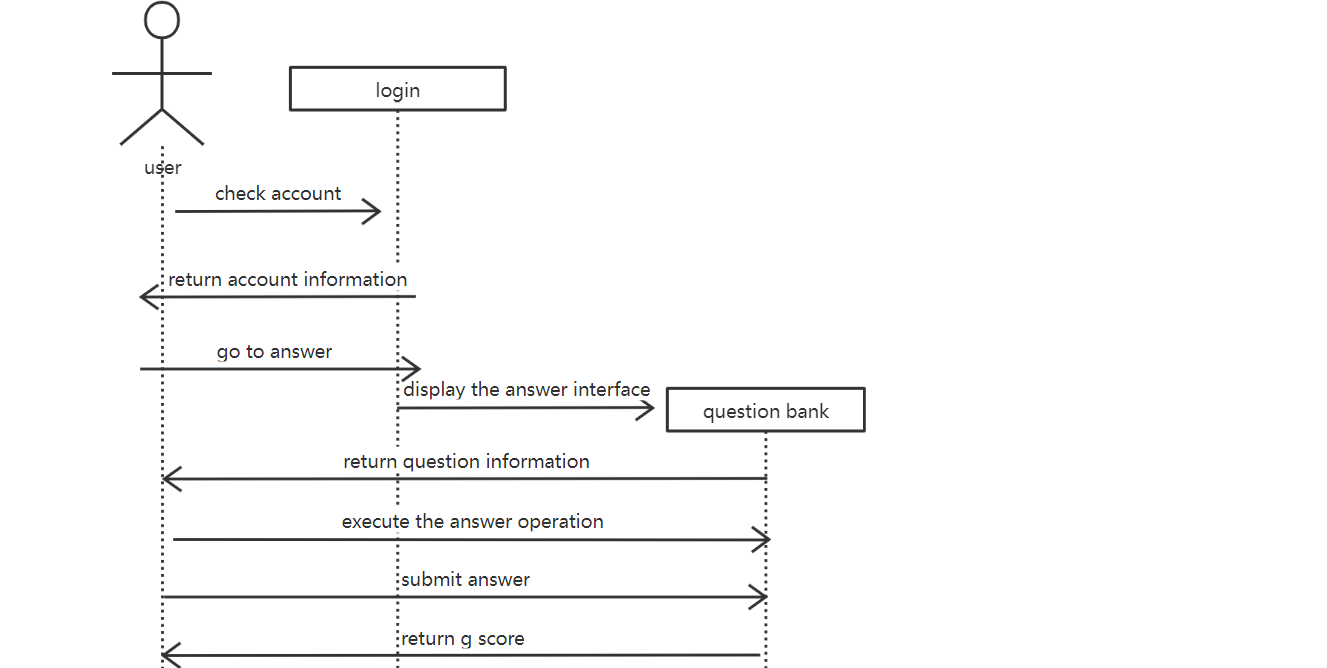


Figure 2.3 – Sequence Diagram of answering questions

The following is a sequence diagram for browsing and querying materials:

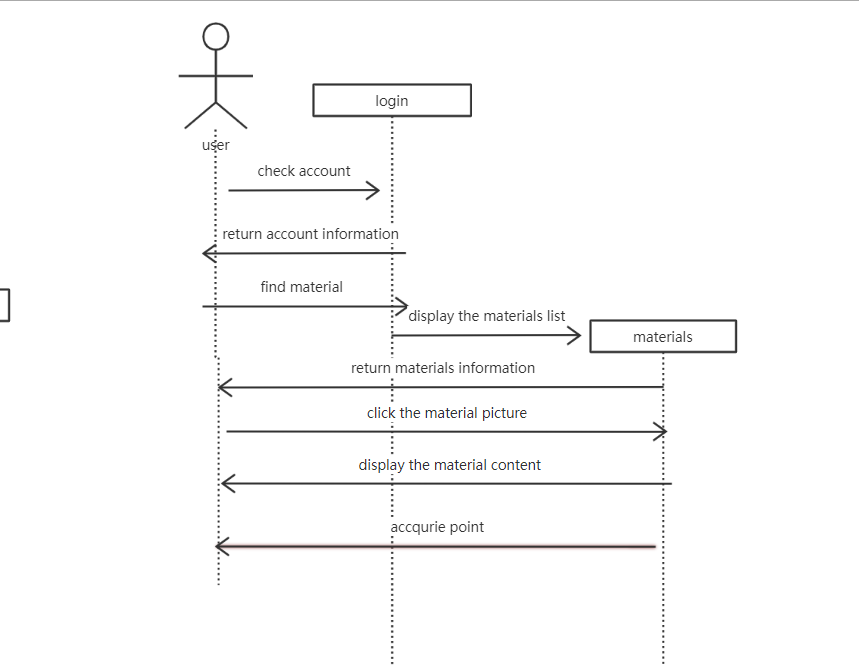


Figure 2.4 – Sequence Diagram of materials

The following is about editing the questions in the question bank:

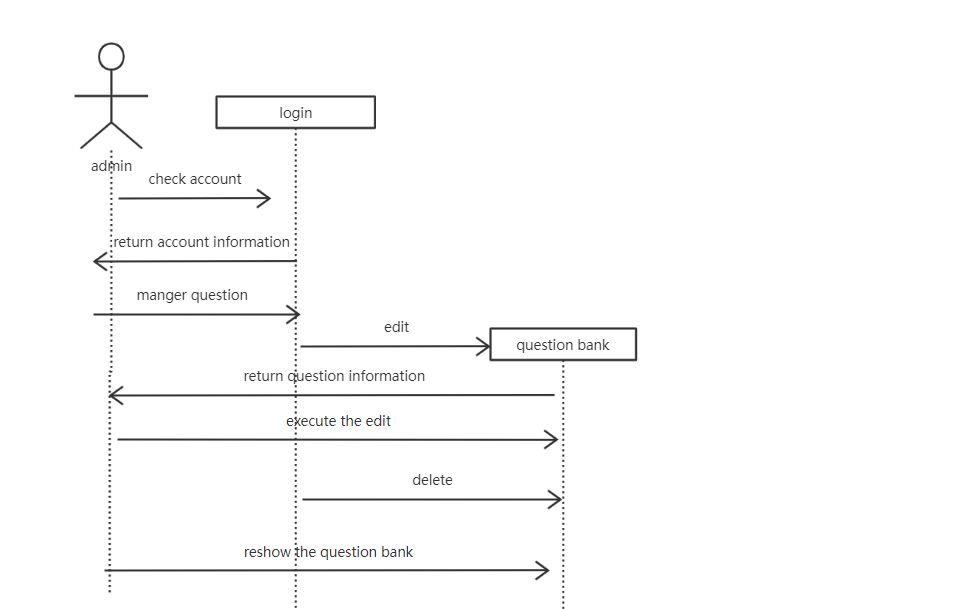


Figure 2.5 – Sequence Diagram of E-learning

The following is the sequence diagram for the administrator to modify the profile information：

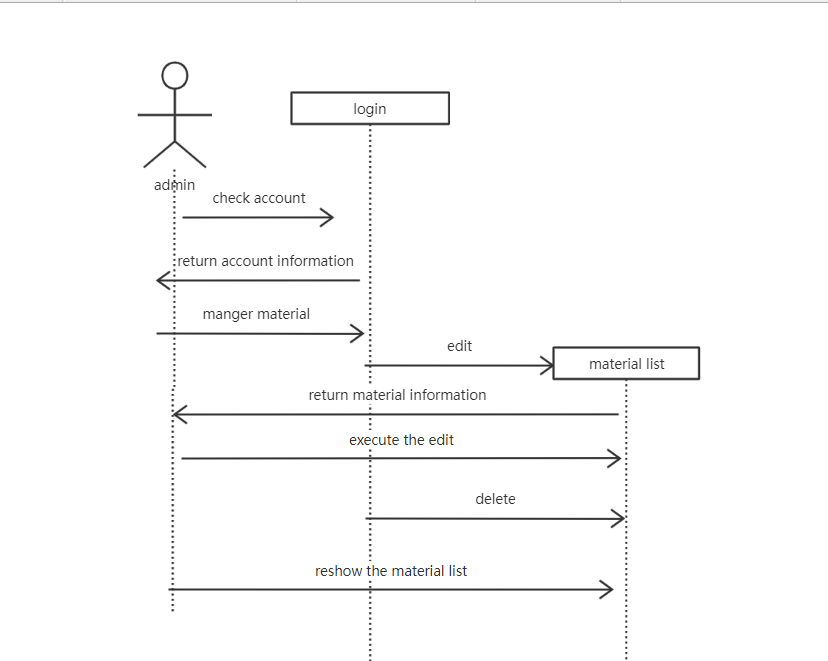


Figure 2.6 – Sequence Diagram of modify material

**2.1.4** Activity Diagram

Activity Diagrams consist of activities, states and transitions between activities and states.

Activity Diagrams describe how activities are coordinated to provide a service and the events needed to achieve some operation.

How the events in a single use case relate to one another.

How a collection of use cases coordinates to create a work for an organization. Activity diagram of user shows at Figure 2.7 and 2.8, activity diagram of admin shows at Figure 2.9 and 2.10.

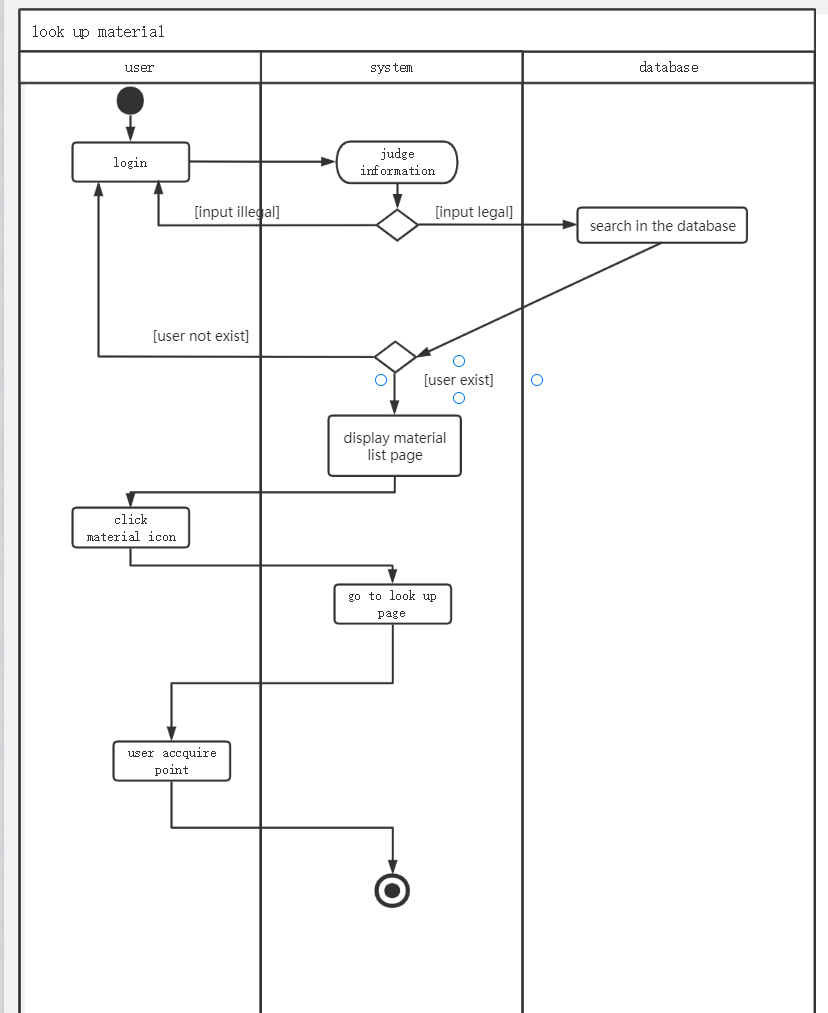


Figure 2.7 – Activity diagram of user

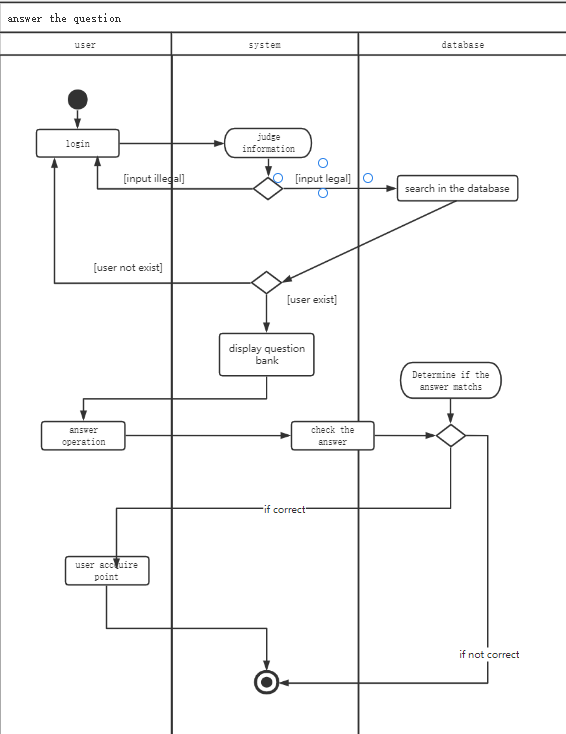


Figure 2.8 – Activity diagram of user

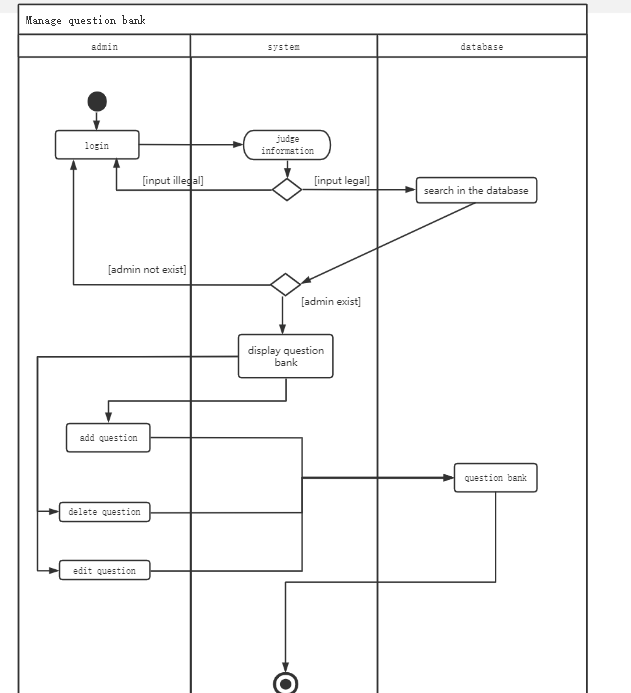


Figure 2.9 – Activity diagram of admin

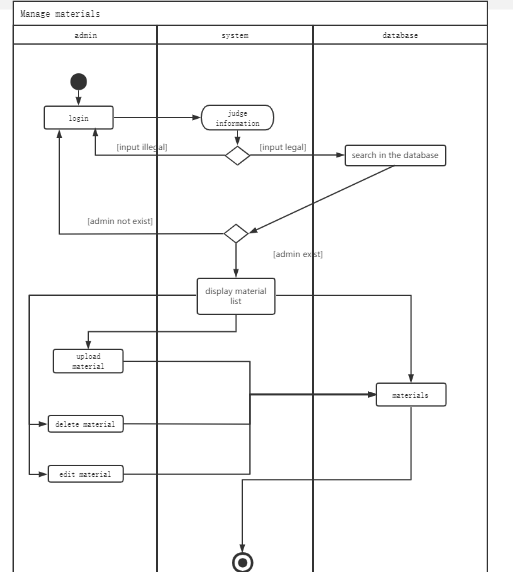


Figure 2.10 – Activity diagram of admin

**2.1.5** Class Diagram

A class represents an abstraction of entities with common characteristics. Associations represent relationships between classes.

Classes are illustrated by rectangles divided into compartments. Put the class name in the first partition (centered, bold, uppercase), list the properties in the second partition (left-aligned, not bolded, lowercase), write the operation to the third. The entity class diagram is shown in Figure 2.11, and the business class diagram is shown in Figure 2.12

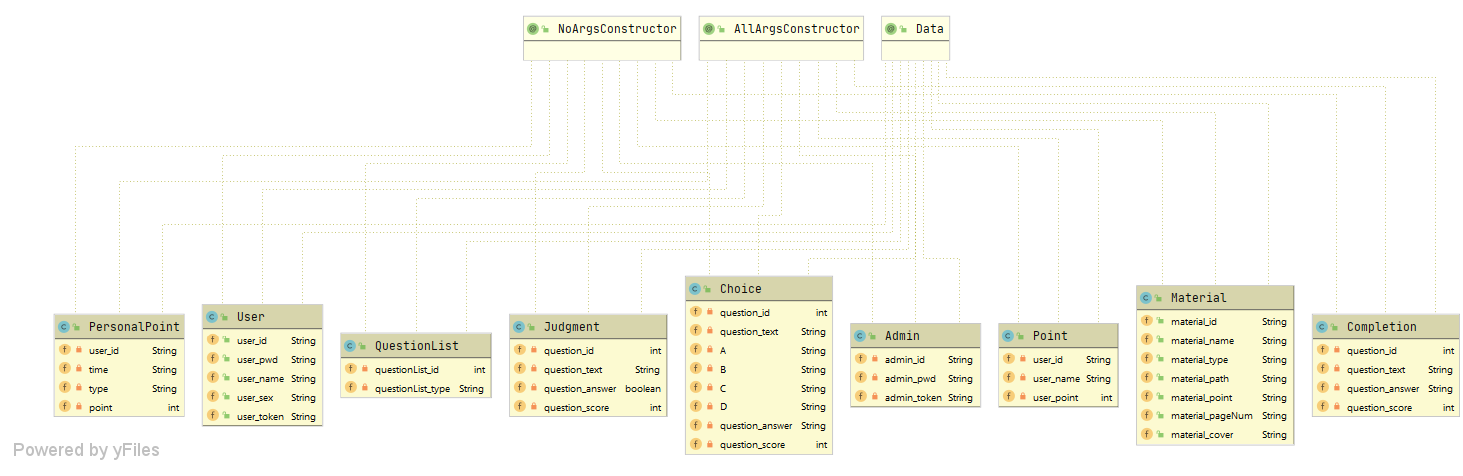


Figure 2.11 – Class Diagram of entity

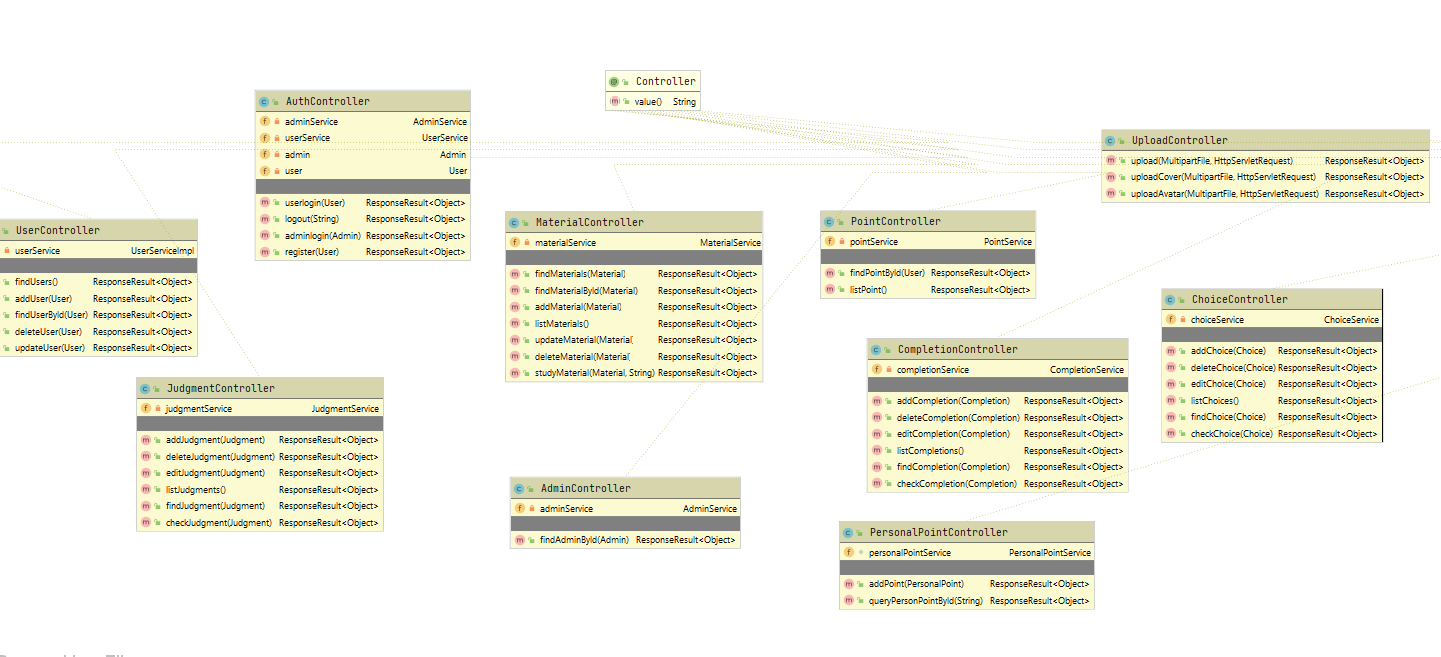


Figure 2.12 – Class Diagram of business class

**2.2 Database Design**

Database design - the process of creating a database schema and determine the necessary integrity constraints.

The main objectives of the database design:

- To secure the database with all the necessary information.

- Ensuring the possibility of obtaining all the necessary data requests.

- Reducing redundancy and duplication of data.

A database in MYSQL consists of a collection of tables that contain data, and other objects, such as views, indexes, stored procedures, and triggers, defined to support activities performed with the data. The data stored in a database is usually related to a particular subject or process, such as inventory information for a manufacturing warehouse.

MYSQL can support many databases, and each database can store either interrelated data or data unrelated to that in the other databases.

For example, a server can have one database that stores personnel data and another that stores product-related data. Alternatively, one database can store current customer order data, and another; related database can store historical customer orders that are used for yearly reporting. Before you create a database, it is important to understand the parts of a database and how to design these parts to ensure that the database performs well after it is implemented.

The design theory of substantive information systems area (or, perhaps, all the real world as a whole) is accepted regarded as three views:

- Presentation of the domain in the form in which it is exist.

- How it treats people (referring to database design data).

- How it can be described by using symbols.

The external representation of the data is a set of requirements for the data from a particular function to be performed by the user. The conceptual scheme is a complete collection of all data requirements derived from user perceptions of the real world. Internal circuit is the database itself.

The main stages into the database design process information system:

**2.2.1** Conceptual Design

This type of design is the collection, analysis and editing of data requirements. To do this, make the following:

- Domain Survey, a study of its information structure.

- Modeling and integration of all ideas.

- A description of the domain objects.

-Description of the object attributes.

- Description of relationships between objects. At the end of this stage, we get a conceptual model that is invariant to the database structure. It is often represented as "entity-relationship" model. The conceptual design diagram for this system is below (Figure 2.13)

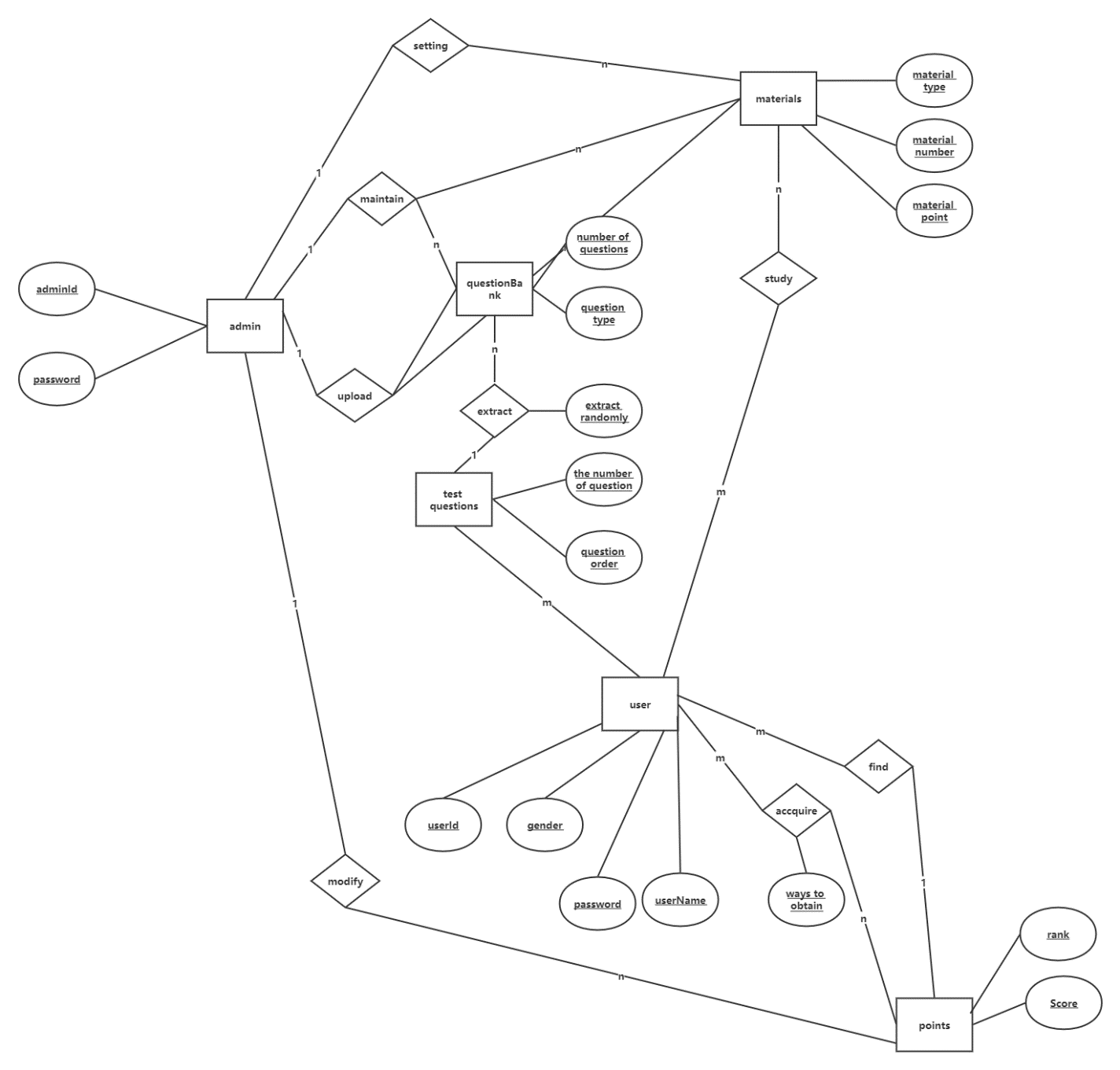


Figure 2.13 – Conceptual design diagram

**2.2.2** Data Logic Design

Data logic design is the conversion of according to the requirements in the data structure. The output is oriented database structure specification, and applied programs. At this stage model database often used to various databases and conduct a comparative analysis of the models. For relational database (the database where the data is presented in the form of tables) date logic model includes:

- Description of tables.

- Description of relationships between tables.

- A description of the attributes.

**2.2.3** Physical Design

Defining the storage features data access methods and a description of the physical structure of the database, her placing on the storage device. This description is called physical model. Physical model includes:

- Type of vehicle; - Methods of data organization.

- Ways to free memory management.

- Data compression methods, etc.

This stage is hidden by the database designer, as implemented means the database. The physical design diagram for this system is below (Figure 2.14)

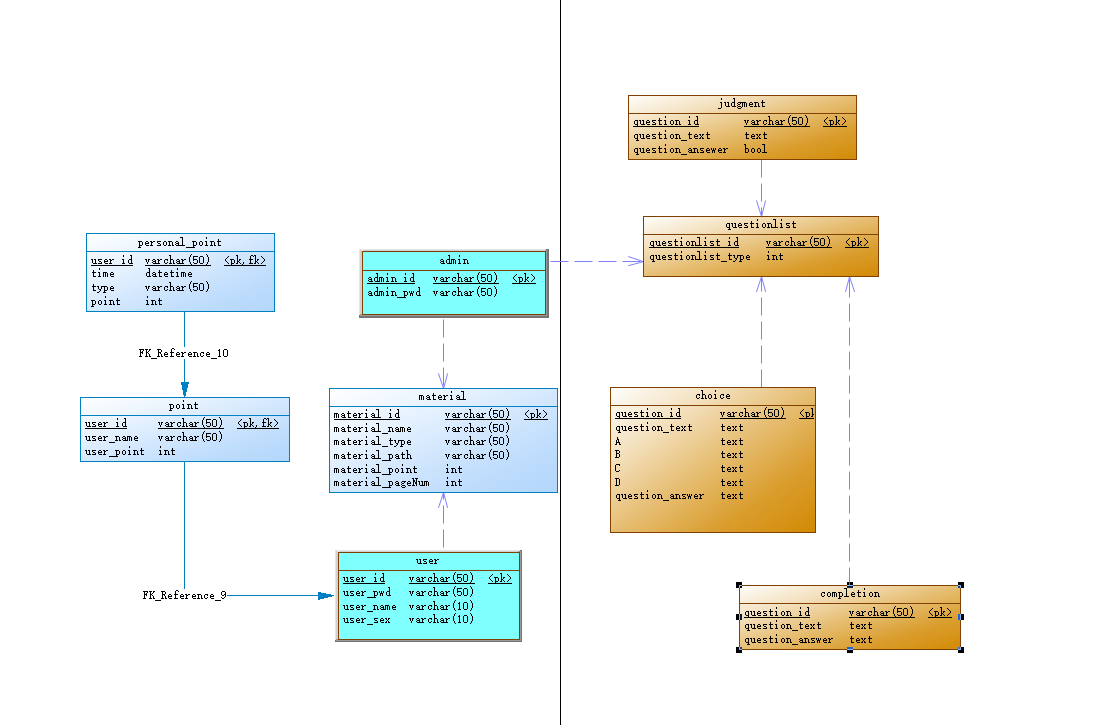


Figure 2.14 – Physical Design Diagram of E-learning

**2.2.4** Database Schema

Defining the storage features data access methods and a description of the physical structure of the database, her placing on the storage device. This description is called physical model. Physical model includes:

- Type of vehicle; - Methods of data organization.

- Ways to free memory management.

- Data compression methods, etc.

This stage is hidden by the database designer, as implemented means the database. The physical design diagram for this system is below (Figure 2.15)

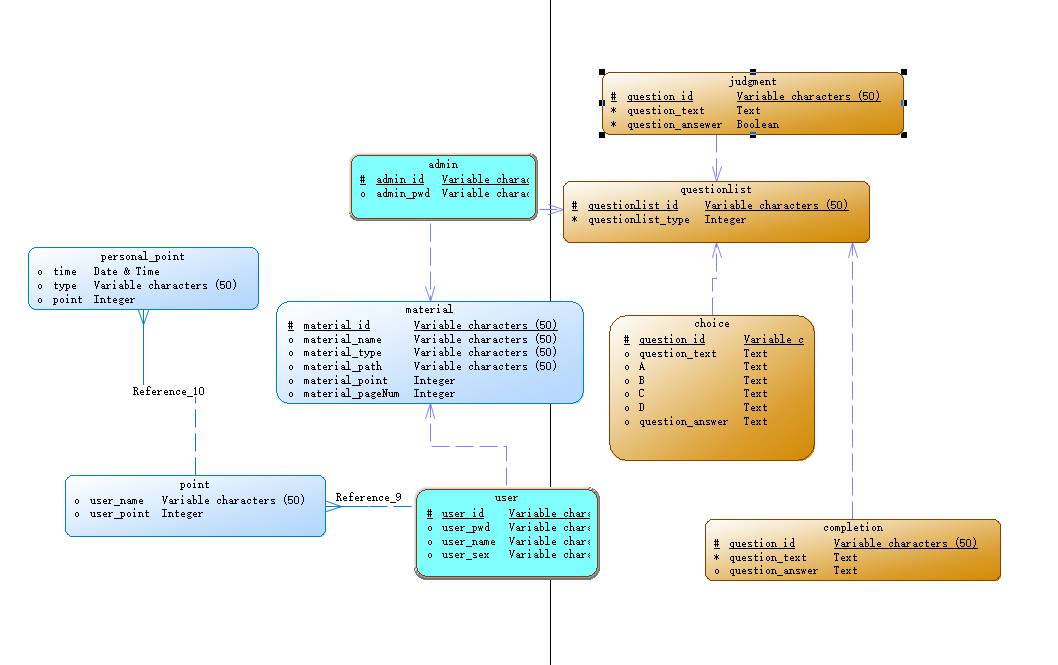


Figure 2.15 – Conceptual data model of E-learning

# **3 E-learning DEVELOPMENT AND IMPLEMENTATION**

Implementation is the stage in the project where the theoretical design is turned into a working system. The implementation phase constructs, installs and operates the new system. The most crucial stage in achieving a new successful system is that it will work efficiently and effectively.

**3.1 Existing programming tools for web development**

**3.1.1 Python**

In Python, the simplification of programming and even complex structures is an important reason for the rise of Python for Web application development. Python helps programmers reduce development time from hours to minutes. The following infographic shows us why python is used for web development.

**3.1.2 Java**

Java is especially useful for web development projects such as banking and fintech and offers better security and functionality than many other programming languages. With Java, developers need less time and less code to improve the performance and responsiveness of their websites.

Another great advantage of Java in web application development is that it is a cross-platform tool. Java can transform code into machine code compatible with native operating systems in the runtime environment, whether it's Linux, Windows, or macOS.

Java has some very powerful web application development frameworks, including Spring, Jsf, Gwt, Vaadin and Grails. Because Spring is able to build complex web applications with high performance, it is the most powerful Java framework for web application development.

**3.1.3 PHP**

PHP is a universal scripting language for building entire web applications or their interactive elements. Using PHP, developers can create content management systems (CMS) and online database systems, message boards, subscription-based websites, gaming web applications, blogs with commenting capabilities, and registration systems.

The PHP part is an object-oriented language that enables code reuse. Reusable components save developers effort and reduce the time required to develop custom web applications.

PHP writes code that is compatible with Linux, Windows, Unix, and macOS, and supports more than 20 databases and most servers. With great customization potential, fast data processing, and smooth integration with different CMSs, the language is a good choice for dynamic web pages. The following two PHP frameworks make rapid web application development a reality by providing well-organized and reusable code.

**3.1.4 Ruby**

Ruby is a dynamic scripting language with advanced syntax. It was created to strike a balance between imperative and functional programming to make the code look more natural. Like Python, Ruby is a general-purpose language that is primarily used to create web applications. But the Ruby programming language has other applications, including prototyping and data analysis.

Free Ruby tools and libraries with basic features facilitate the development of web applications. Ruby's built-in apps are easy to extend. Enterprises can launch basic Ruby web applications and extend them into powerful Ruby applications that can handle thousands of user requests simultaneously.

**3.1.5 JavaScript**

JavaScript code is simple and flexible, easy to write, and easy to debug. In addition, it can be used in scripts written in other programming languages.

JavaScript is not executed on the website server, but on the user's device. This minimizes server requests and improves the user experience. JavaScript can also turn static websites into web applications by allowing developers to add menu features, animations, and hover interactions.

Most JavaScript frameworks are like a good recipe: they have full documentation and complete descriptions of features. The React JavaScript framework is supported by Google and Angular is supported by Facebook. Detailed documentation can improve the speed of Web application development.

**3.1.6 TypeScript**

TypeScript is a style or superset of JavaScript that has become an important choice for browser application front-end development. This strongly typed object-oriented language is often referred to as a set of tools or JavaScript plus additional features.

Why does TypeScript appear? JavaScript is known for being dynamic and flexible. But the flexibility of this language makes its code error-prone. The role of TypeScript is to introduce static information into JavaScript code. This static information makes the code less error-prone and easier to read, understand, and debug.

**3.1.7 Ajax**

Ajax (short for "Asynchronous JavaScript and XML")is a set of web development techniques that uses various web technologies on the client-side to create asynchronous web applications. With Ajax, web applications can send and retrieve data from a server asynchronously (in the background) without interfering with the display and behavior of the existing page. By decoupling the data interchange layer from the presentation layer, Ajax allows web pages and, by extension, web applications, to change content dynamically without the need to reload the entire page. In practice, modern implementations commonly utilize JSON instead of XML.

Ajax is not a technology, but rather a programming concept. HTML and CSS can be used in combination to mark up and style information. The webpage can be modified by JavaScript to dynamically display and allow the user to interact with the new information. The built-in XMLHttpRequest object is used to execute Ajax on webpages, allowing websites to load content onto the screen without refreshing the page. Ajax is not a new technology, nor is it a new language. Instead, it is existing technologies used in a new way.

**3.1.8 HTML/CSS**

HTML and CSS are programming languages used to build web application front and ends, and they can be supplemented by JavaScript.

HTML is the basic building block of an application. This language determines the structure and content of the web page. The main disadvantage of HTML is that it does not help developers make Web applications interactive. To enhance HTML pages with beautiful styles and ui hints, add designs and display all HTML elements, developers use CSS. See specific usages of html, CSS, and JavaScript web application development languages below.

Although HTML defines the structure of the page, CSS determines the appearance of the page. JavaScript's task is to define the behavior of a page: validation, dynamic content changes, image manipulation.

**3.2 Programming Tools Used**

After analyzing the subject area and the structure of the system, the least expensive software to consider in creating the software is as follow: MYSQL, JAVA Script, HTML, JQUERY, CSS.

**3.2.1** Hyper Text Markup Language (HTML)

First developed by Tim Berners-Lee in 1990, HTML is short for HyperText Markup Language. HTML is used to create electronic documents (called pages) that are displayed on the World Wide Web. Each page contains a series of connections to other pages called hyperlinks. Every web page you see on the Internet is written using one version of HTML code or another.

HTML code ensures the proper formatting of text and images so that your Internet browser may display them as they are intended to look. Without HTML, a browser would not know how to display text as elements or load images or other elements. HTML also provides a basic structure of the page, upon which Cascading Style Sheets are overlaid to change its appearance. One could think of HTML as the bones (structure) of a web page, and CSS as its skin (appearance).

HTML is written in the form of HTML elements consisting of tags, enclosed in angle brackets (like <html>), within the web page content. HTML tags normally come in pairs like <h1> and </h1>. The first tag in a pair is the start tag, the second tag is the end tag (they are also called opening tags and closing tags). In between these tags web designers can add text, tables, images, etc.

**3.2.2** Cascading Style Sheets (CSS)

CSS is a stylesheet language used to describe the presentation of a document written in HTML or XML (including XML dialects such as SVG or XHTML). CSS describes how elements should be rendered on screen, on paper, in speech, or on other media.

CSS is one of the core languages of the open web and is standardized across browsers according to the W3C specification. Developed in levels, CSS1 is now obsolete, CSS2.1 is a recommendation, and CSS3, now split into smaller modules, is progressing on the standardization track.

**3.2.3** JavaScript

JavaScript is a dynamic computer programming language. It is lightweight and most commonly used as a part of web pages, whose implementations allow client-side script to interact with the user and make dynamic pages. It is an interpreted programming language with object-oriented capabilities.

Client-side JavaScript is the most common form of the language. The script should be included in or referenced by an HTML document for the code to be interpreted by the browser. It means that a web page need not be a static HTML, but can include programs that interact with the user, control the browser, and dynamically create HTML content. The JavaScript client-side mechanism provides many advantages over traditional CGI server-side scripts. For example, you might use JavaScript to check if the user has entered a valid e-mail address in a form field. The JavaScript code is executed when the user submits the form, and only if all the entries are valid, they would be submitted to the Web Server. JavaScript can be used to trap user-initiated events such as button clicks, link navigation, and other actions that the user initiates explicitly or implicitly.

**3.2.4** jQuery

jQuery is a JavaScript library that allows web developers to add extra functionality to their websites. It is open source and provided for free under the MIT license. In recent years, jQuery has become the most popular JavaScript library used in web development.

To implement jQuery, a web developer simply needs to reference the jQuery JavaScript file within the HTML of a webpage. Some websites host their own local copy of jQuery, while others simply reference the library hosted by Google or the jQuery server. For example, a webpage may load the jQuery library using the following line within the <head> section of the HTML:

<script type="text/JavaScript"

src="//ajax.googleapis.com/ajax/libs/jquery/1.9.1/jquery.min.js"></script>

Once the jQuery library is loaded, a webpage can call any jQuery function supported by the library. Common examples include modifying text, processing form data, moving elements on a page, and performing animations. jQuery can also work with Ajax code and scripting languages, such as PHP and ASP to access data from a database. Since jQuery runs on the client side (rather than the web server), it can update information on a webpage in realtime, without reloading the page. A common example is "autocomplete," in which a search form automatically displays common searches as you type your query. In fact, this is how TechTerms.com provides search suggestions when you type in the search box.

**3.2.5** MySQL

MySQL is the world’s most popular open source database, enabling the cost-effective delivery of reliable, high-performance and scalable Web-based and embedded database applications. It is an integrated transaction safe, ACID-compliant database with full commit, rollback, crash recovery, and row-level locking capabilities. MySQL delivers the ease of use, scalability, and high performance, as well as a full suite of database drivers and visual tools to help developers and DBAs build and manage their business-critical MySQL applications. MySQL is developed, distributed, and supported by Oracle, and the latest information about MySQL software can be found on the MySQL Web site. The MySQL database provides the following features:

* high performance and scalability to meet the demands of exponentially growing data loads and users.
* self-healing replication clusters to improve scalability, performance and availability.
* online schema change to meet changing business requirements.
* performance schema for monitoring user- and application-level performance and resource consumption.
* SQL and NoSQL access for performing complex queries and simple, fast key value operations.
* platform independence giving you the flexibility to develop and deploy on multiple operating systems.
* big data interoperability using MySQL as the operational data store for Hadoop and Cassandra.

To achieve the highest level of MySQL scalability, security, reliability and uptime, MySQL Enterprise Edition includes the most comprehensive set of advanced features, management tools and technical support, including MySQL Enterprise Monitor, MySQL Enterprise Backup, as well as scalability, security, auditing and high availability features. MySQL Enterprise Edition also includes 24x7 Oracle Premier Support and Oracle product certifications and integrations.

**3.3 SSM Framework**

The SSM framework is the integration of Spring MVC, Spring and Mybatis frameworks, is the standard MVC pattern, dividing the entire system into four layers: View layer, Controller layer, Service layer, DAO layer, using Spring MVC for request forwarding and view management, Spring implements business object management, and Mybatis as the persistence engine for data objects.

**3.3.1** Mybatis (Persistent layer: Dao layer (mapper))

DAO layer: The DAO layer mainly does the work of the data persistence layer, and some of the tasks responsible for communicating with the database are encapsulated here.

The design of the DAO layer begins with the design of the DAO interface. Then define the implementation class for this interface in Spring's configuration file. Then you can call this interface in the module for data business processing, without caring which class the specific implementation class of this interface is, which is very clearly structured.

The data source configuration of the DAO layer, as well as the parameters regarding the database connection, are configured in Spring's configuration file.

**3.3.2 Spring:** Service layer

Service layer: The service layer is mainly responsible for the logic app design of business modules.

Design the interface first, and then design the classes that it implements.

Next, configure the association of its implementation in Spring's configuration file. In this way, we can call the service interface in the application for business processing.

The service layer of the service layer, specifically to call the interface of the defined DAO layer.

Encapsulating the business logic of the service layer is conducive to the independence and reuse of the common business logic, and the program appears to be very concise.

**3.3.3** Spring MVC: Controller layer

Controller layer: The Controller layer is responsible for the control of specific business module processes.

In this layer, the interface of the service layer is called to control the orchestration.

The configuration of the control is also carried out in the Spring configuration file, for the specific business process, there will be different controllers, our specific design process can be abstracted into the process, design can be reused sub-unit process module, so that not only the program structure becomes clear, but also greatly reduce the amount of code.

**3.3.4** View layer

View layer: The View layer and the control layer are closely integrated, and the two need to be combined to work together. The View layer is primarily responsible for the representation of the foreground java server page.

**3.3.5** SSM principles and processes (Figure 3.1)

图示

中度可信度描述已自动生成

Figure 3.1 – SSM principles and processes

**3.3.6 Main advantages of SSM Framework**

**3.3.6.1** Advantages of Spring:

Through Spring's IOC feature, the dependencies between objects are handed over to Spring control, which is convenient for decoupling and simplifies development Through Spring's AOP feature, the duplicate modules are centralized and the control of transactions, logs, and permissions is realized.

Provides integration support for other excellent open-source frameworks.

**3.3.6.2** Advantages of Spring MVC:

SpringMVC is a lightweight web framework that uses MVC design ideas to decouple the web layer and allow us to develop more concise and flexible data validation, formatting, and data binding mechanisms that are seamless with Spring.

**3.3.6.3** Advantages of Mybatis:

Database operations (sql) use xml file configuration, decoupling sql and code provide mapping tags, support the mapping of objects and relationships with database orm fields, support object relationship mapping tags, support the formation of object relationships Provide xml tags, support dynamic sql.

**3.4 Project Implementation Description**

The Project is loaded in IntelliJ IDEA 2020. We used IntelliJ IDEA for Design and coding of project. Created and maintained all databases into MYSQL, in that we create tables, write query for store data or record of project. It was decided to create software based on the following MySQL, LINQ, JAVA Script, HTML, SPRING, MVC, Framework, MYSQL, APEX, UML-DIAGRAMMS, API, JQUERY, JSON, Bootstrap.

This chapter describes the complete design of the database and presents the corresponding diagrams. Algorithmic implementation of the program is also described, and the diagram of our system is shown. This shows what actions can be performed by users with certain rights in our system. Also, in this chapter you will find a full user guide of our system. Our system is easier to use and since free technologies were used, our system will be free of charge.

**3.4.1** Tables Description

Following are the tables along with constraints used in E-learning database.

1. Admin: The administrator table contains the administrator's information, storing the administrator id, administrator password, and the token generated when the administrator logs in.

2. User: The user table contains user information, storing user id, user password, user name, user gender, and the token generated when the user logs in.

3. Choice: The multiple-choice question table contains the multiple-choice question information, stores the multiple-choice question id, multiple-choice question description, description of each option of the multiple-choice question, the answer of the multiple-choice question, and the score of the multiple-choice question. The user can get the score after selecting the correct answer when answering the question.

4. Judgment: The judgment question table contains multiple-choice question information, and stores the judgment question id, the judgment question description, the answer of the judgment question, and the score of the judgment question. The user can obtain the score after selecting the correct answer when answering the question.

5. Completion: The fill-in-the-blank question sheet contains the fill-in-the-blank question information, stores the fill-in-the-blank question id, the fill-in-the-blank question description, the answer of the fill-in-the-blank question, and the score of the fill-in-the-blank question. The user can get the score after filling in the correct answer when answering the question.

6. Material: The data table contains the data id, data name, data type, data path, data points, and data pages. Users can earn points by viewing and learning materials

7. Personal point: The personal points table is mainly used to store the number of points obtained by the user, the type of points, and the time when points are obtained.

8. Point: The points table is used to store the total points of each user, and the points table uses the personal point table to count the total points.

*CREATE TABLE ADMIN*

*(*

*admin\_id VARCHAR(50) NOT NULL,*

*admin\_pwd VARCHAR(50),*

*PRIMARY KEY (admin\_id)*

*);*

*CREATE TABLE choice*

*(*

*question\_id VARCHAR(50) NOT NULL,*

*question\_text TEXT,*

*A TEXT,*

*B TEXT,*

*C TEXT,*

*D TEXT,*

*question\_answer TEXT,*

*PRIMARY KEY (question\_id)*

*);*

*/\*==============================================================\*/*

*CREATE TABLE COMPLETION*

*(*

*question\_id VARCHAR(50) NOT NULL,*

*question\_text TEXT NOT NULL,*

*question\_answer TEXT,*

*PRIMARY KEY (question\_id)*

*);*

*CREATE TABLE judgment*

*(*

*question\_id VARCHAR(50) NOT NULL,*

*question\_text TEXT NOT NULL,*

*question\_ansewer BOOL NOT NULL,*

*PRIMARY KEY (question\_id)*

*);*

*CREATE TABLE material*

*(*

*material\_id VARCHAR(50) NOT NULL,*

*material\_type VARCHAR(50),*

*material\_path VARCHAR(50),*

*material\_point INT,*

*material\_pageNum INT,*

*material\_cover VARCHAR(50),*

*PRIMARY KEY (material\_id)*

*);*

*CREATE TABLE USER*

*(*

*user\_id VARCHAR(50) NOT NULL,*

*user\_pwd VARCHAR(50),*

*user\_name VARCHAR(10),*

*user\_sex VARCHAR(10),*

*PRIMARY KEY (user\_id)*

*);*

*CREATE TABLE POINT*

*(*

*user\_id VARCHAR(50) NOT NULL,*

*user\_name VARCHAR(50),*

*user\_point INT,*

*PRIMARY KEY (user\_id),*

*CONSTRAINT FK\_Reference\_9 FOREIGN KEY (user\_id)*

*REFERENCES USER (user\_id)*

*);*

*CREATE TABLE personal\_point*

*(*

*user\_id VARCHAR(50) NOT NULL,*

*TIME DATETIME,*

*TYPE VARCHAR(50),*

*POINT INT,*

*PRIMARY KEY (user\_id),*

*CONSTRAINT FK\_Reference\_10 FOREIGN KEY (user\_id)*

*REFERENCES POINT (user\_id)*

*);*

**3.4.2** Main features of developed software

The structure of my code is present in the picture below (Figure 3.2 to Figure 3.6):

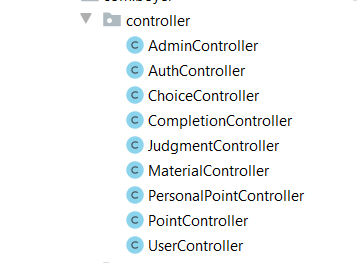


Figure 3.2– Controller

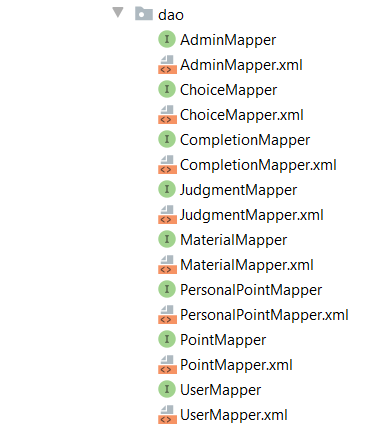


Figure 3.3 – DAO (Data Access Object)

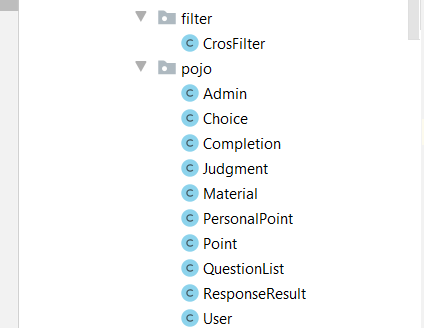


Figure 3.4 –Filter and Pojo class

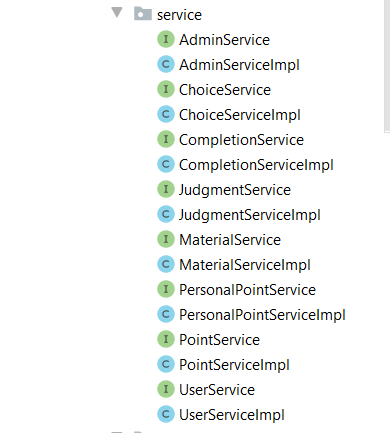


Figure 3.5 – Service layer class

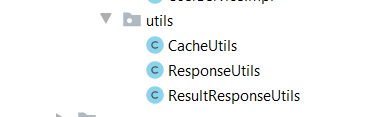


Figure 3.6 – utils class

Below it will be described some of the most important functions of the software.

*@RequestMapping(value = "/addPoint",consumes = "application/json",method = RequestMethod.PUT)  
@ResponseBody  
public ResponseResult<Object> addPoint(@RequestBody PersonalPoint personalPoint) throws IOException {  
 System.out.println(personalPoint);  
 personalPoint.setTime(new Date().toString());  
 personalPointService.addPoint(personalPoint);  
 if(!StringUtils.isEmpty(personalPoint)){  
 String token = UUID.randomUUID().toString().replace("-","");  
 CacheUtils.put(token,null);  
 return ResultResponseUtils.success(token);  
 }else{  
 return ResultResponseUtils.error("User failed to add points ");  
 }  
}*

This method is the method of obtaining points after learning materials or after the correct answer, here will be associated with the user ID, and the personal integral table is associated with the user ID, and the trigger in the database All points of each user are summarized and stored in the total table of integrals. Below is the code of the trigger in the database, the trigger implements the data in the automatic statistical personal integration table summary.

*DROP TRIGGER CountPointTrigger;*

*DELIMITER $*

*CREATE TRIGGER CountPointTrigger AFTER INSERT ON personal\_point*

*FOR EACH ROW*

*BEGIN*

*UPDATE POINT*

*SET user\_point = (SELECT SUM(POINT) FROM personal\_point WHERE user\_id = new.user\_id)*

*WHERE user\_id = new.user\_id;*

*END $;*

The following method is the point value contained in the user's learning data and then obtain the information contained in the information, and it is counted in the personal integration table.

*@RequestMapping(value = "/studyMaterial/{userId}",consumes = "application/json",method = RequestMethod.PUT)  
 @ResponseBody  
 public ResponseResult<Object> studyMaterial(@RequestBody Material material , @PathVariable("userId") String userId) throws IOException{  
 if(!StringUtils.isEmpty(material)||userId==null){  
  
// materialService.deleteMaterialById(material.getMaterial\_id());  
 String token = UUID.randomUUID().toString().replace("-","");  
 CacheUtils.put(token,null);  
 return ResultResponseUtils.success(token);  
 }else{  
 return ResultResponseUtils.error("The learning materials failed, the materials were not queried or userId");  
 }  
 }*

This method is used by the administrator to upload data, upload the data to the server, and set the information of the data, such as data pictures, data points and so on.

*@RequestMapping(value = "/addMaterial",consumes = "application/json",method = RequestMethod.PUT)  
@ResponseBody  
public ResponseResult<Object> addMaterial(@RequestBody Material material) throws IOException{  
 System.out.println("Visited Add Profile ");  
 if(!StringUtils.isEmpty(material)){  
 materialService.saveMaterial(material);  
 String token = UUID.randomUUID().toString().replace("-","");  
 CacheUtils.put(token,null);  
 System.out.println("added"+material);  
 return ResultResponseUtils.success(token);  
 }else{  
 return ResultResponseUtils.error("Failed to add data ");  
 }  
}*

# **4 USER GUIDE**

The system provides the user's login and registration interface. The system provides a login interface for users and administrators. Users can log in to their respective clients through this page. If users do not have their own accounts, they can click sign up to go to the registration interface. Only users can register, and administrators can register. Add directly in the database through the super administrator.

User registration and login pages can be seen in Figure 4.1 and Figure 4.2.

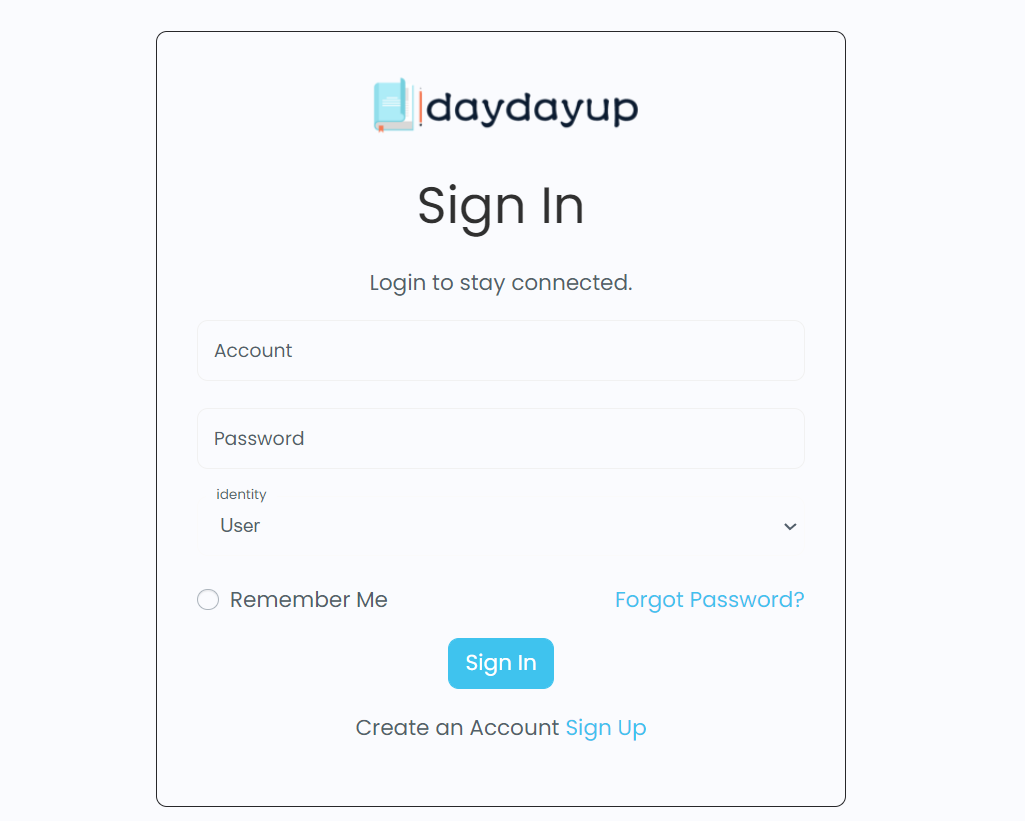


Figure 4.1 – Admin and user sign in

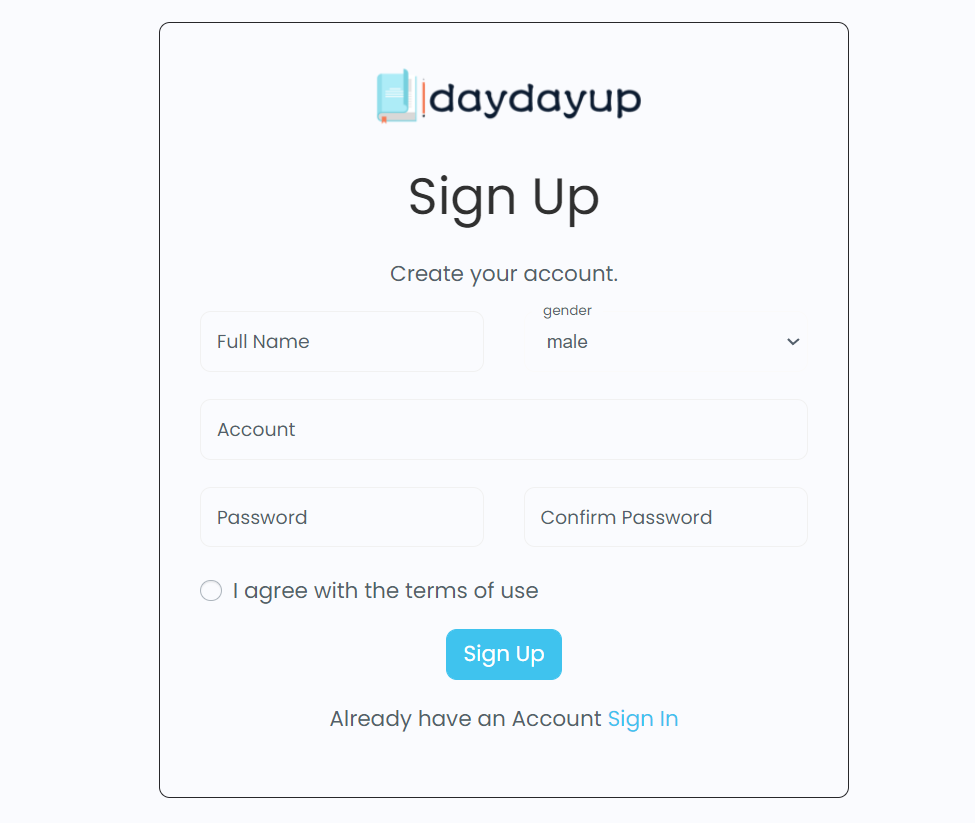


Figure 4.2 – User sign up

The system uses a graphical interface to guide users to query and read data. Users can click the icon of the data to enter the data reading interface (Figure 4.3), and open the pdf reader through the pdf loading interface (Figure 4.4).

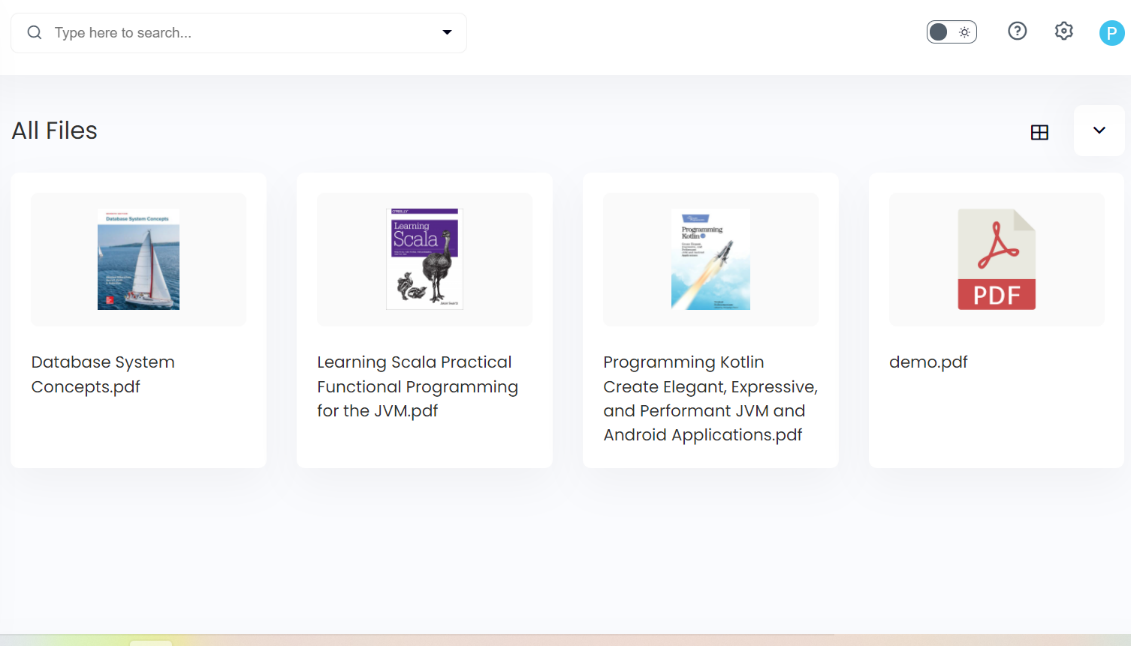


Figure 4.3 – Materials

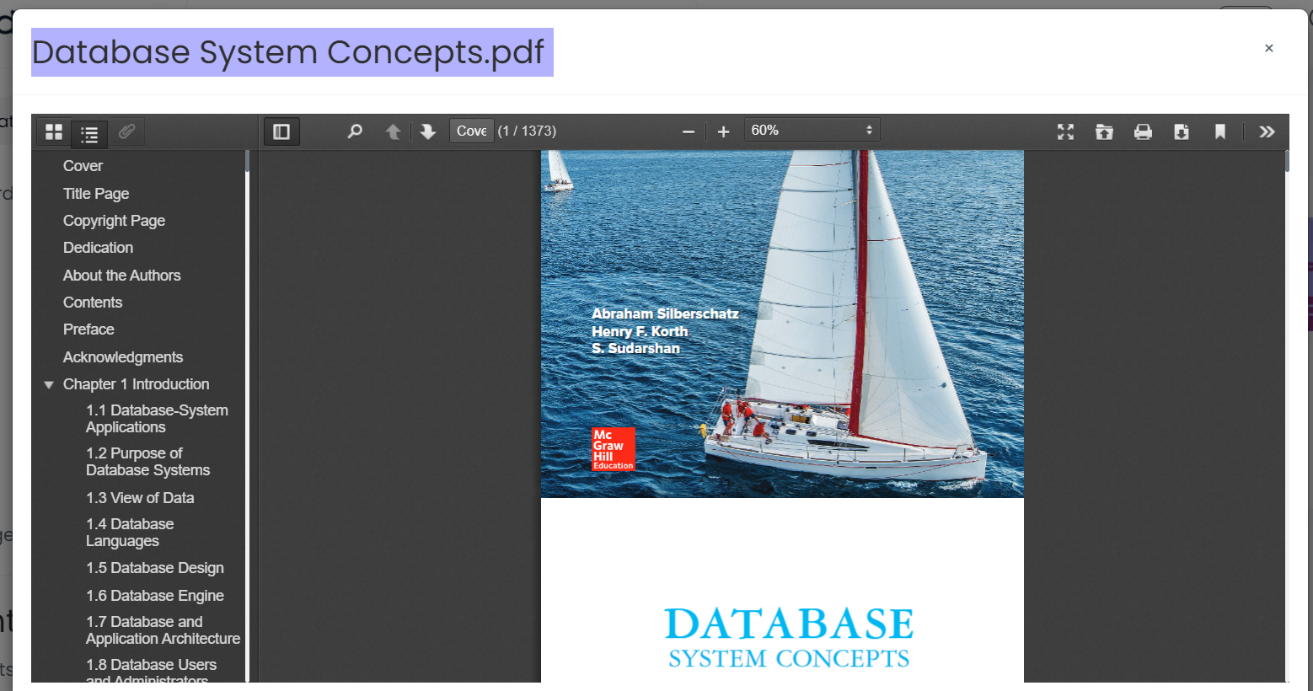


Figure 4.4 – The page of reading material

The system provides the administrator with an interface for managing users. The interface displays a user list. The administrator can choose to edit, delete, and add administrators through the list (Figure 4.5).

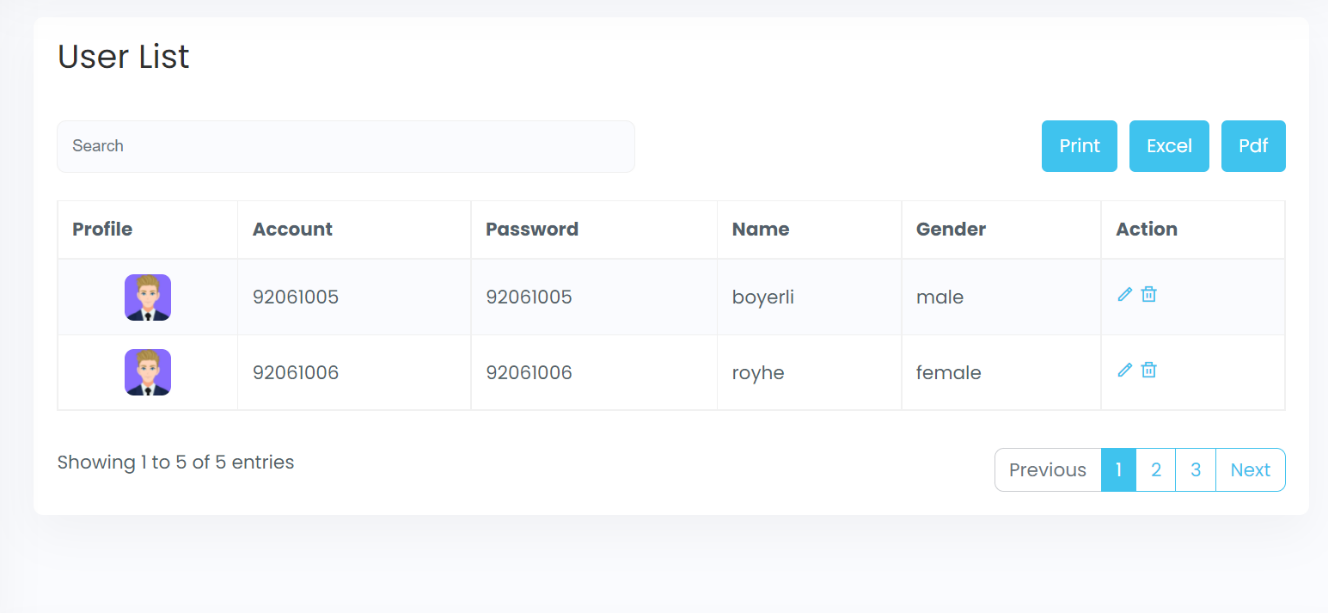


Figure 4.5 – The page of user list

The administrator can add data to the system. The adding interface (Figure 4.6) is as follows. The administrator selects the data cover and data file to be uploaded to the server, and fills in the corresponding information.

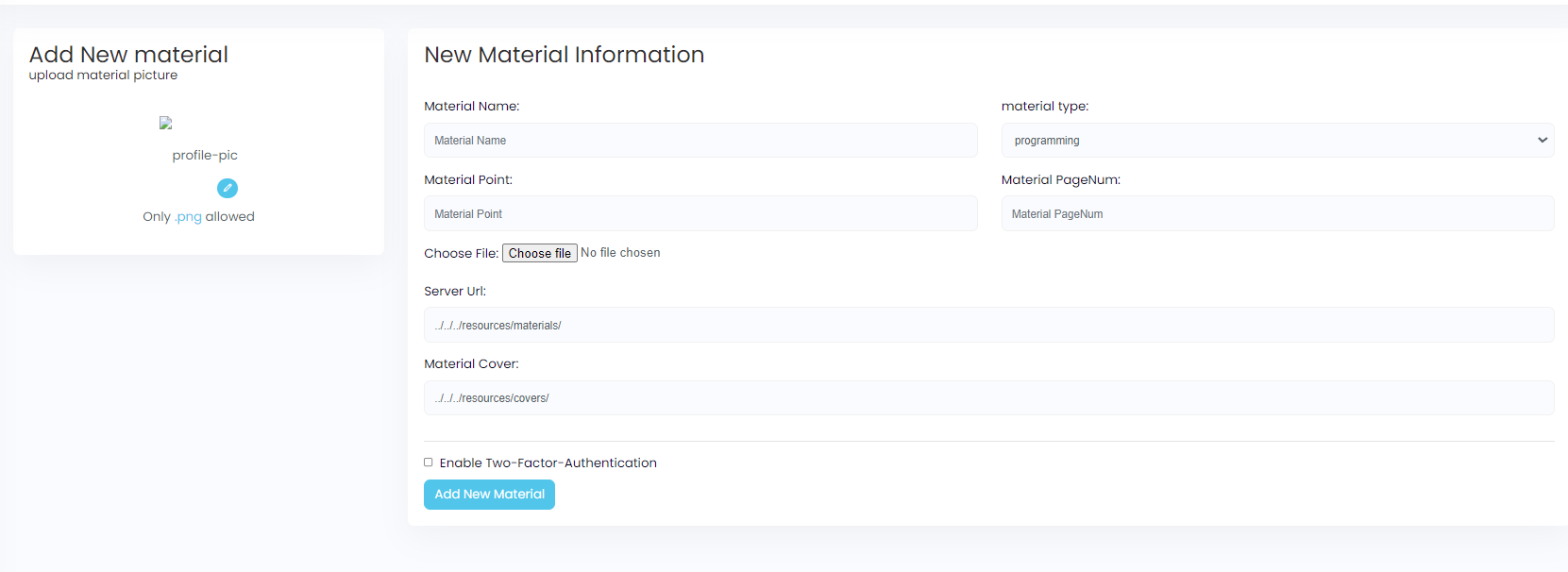


Figure 4.6 – The page of add material

This is the page which display the information about added material (Figure 4.7).

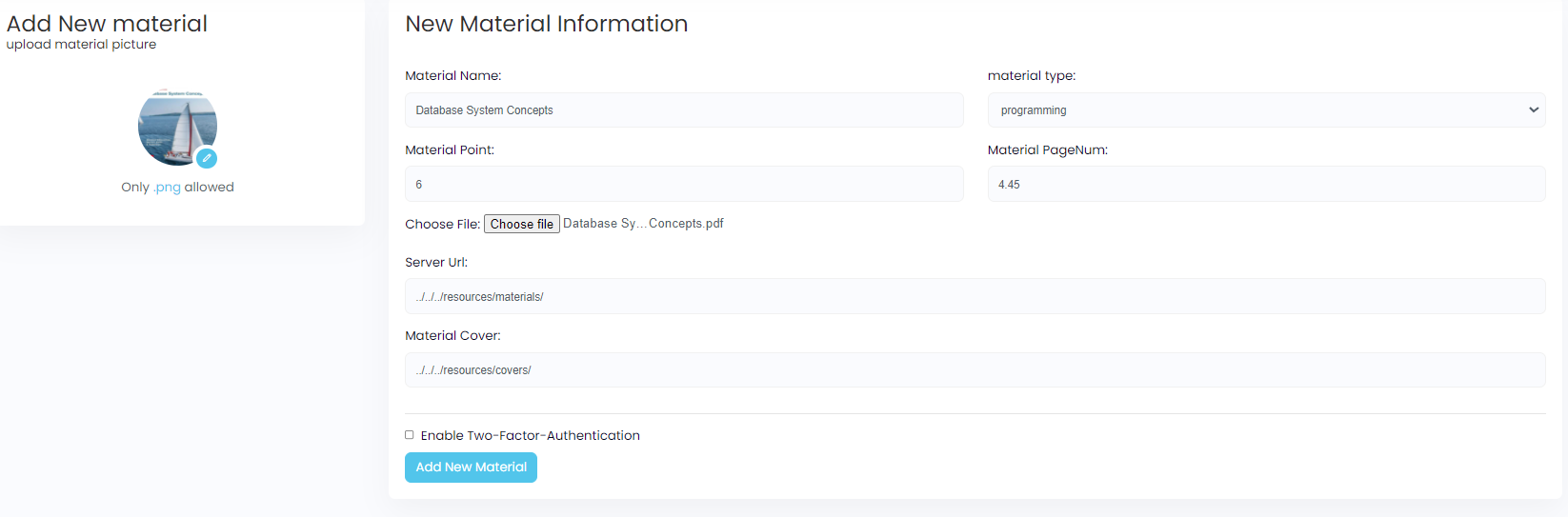


Figure 4.7 – The page of added material

**CONCLUSION**

This online learning system has basically realized all functions and passed the preliminary test. The system includes two ports: administrator and user. Users can choose user login or administrator login by accessing the system. If users do not have an account, they can register first. The account needs to be added by a super administrator in the database background. Users can realize online learning, including material learning and answering exercises, and add a point reward mechanism to improve users' interest in learning, and users can view their own points details and points rankings. The page is designed with day and night modes to meet the different needs of users. The administrator can log in to the administrator port to manage users, materials and question banks, and select material files and cover sheets to upload.

The overall design of the project adopts the Spring+SpringMVC+MyBatis model, which makes the overall design of the project more reasonable and more distinct.

In the process of project implementation, we encountered many problems:

(1) The most difficult to solve is the front-end and back-end cross-domain problem. Since the front-end and back-end separation technology is considered in the design, there will be different sources of policies when sending requests from the front-end and back-end. The problem, in order to solve this problem, I added a cross-domain filter to process the request header of the cross-domain request. Under the action of this filter, requests can be cross-domain. But there is still another problem, that is, if two Ajax get requests are used in the same interface, the cross-domain Ajax problem will occur again, but if one of the get is changed to other types of requests, the cross-domain problem will disappear.

(2) The second difficulty is the conflict between form submission and ajax submission. If a page has form submission and ajax submission at the same time, the two requests will conflict and may override the ajax request, so you can set the attribute onsubmit="return false" in the form tag to prevent the submission of the form

**REFERENCE**

[1] [Crazy God says Java] Mybatis latest complete tutorial IDEA version is easy to understand. https://www.bilibili.com/video/BV1NE411Q7Nx?spm\_id\_from=333.1007.top\_right\_bar\_window\_custom\_collection.content.click.

2. [Crazy God Says Java] The latest integrated teaching IDEA version of the SSM framework.

https://www.bilibili.com/video/BV1RE41127rv?spm\_id\_from=333.1007.top\_right\_bar\_window\_custom\_collection.content.click.

3. [Crazy God Says Java] The IDEA version of the latest SpringMVC tutorial is easy to understand. https://www.bilibili.com/video/BV1aE41167Tu?spm\_id\_from=333.1007.top\_right\_bar\_window\_custom\_collection.content.click.

4.Detailed explanation of @ResponseBody. https://blog.csdn.net/originations/article/details/89492884.

**Appendix A**

Server-side code

**1. Filters used to solve cross-domain problems**

*package com.boyer.filter;  
import javax.servlet.\*;  
import javax.servlet.annotation.WebFilter;  
import javax.servlet.http.HttpServletResponse;  
import java.io.IOException;  
  
@WebFilter(urlPatterns = {"/\*"})  
public class CrosFilter implements Filter {  
 @Override  
 public void init(FilterConfig filterConfig) throws ServletException {  
  
 }  
  
 @Override  
 public void doFilter(ServletRequest request, ServletResponse response, FilterChain filterChain) throws IOException, ServletException {  
// System.out.println("CorsFilter doFilter");  
 HttpServletResponse resp = (HttpServletResponse) response;  
 resp.setHeader("Access-Control-Allow-Origin","\*");  
 //GET POST PUT DELETE  
 resp.setHeader("Access-Control-Allow-Methods","\*");  
 resp.setHeader("Access-Control-Max-Age","3600");  
 resp.setHeader("Access-Control-Allow-Headers","\*");  
 resp.setHeader("Access-Control-Allow-Credentials","false");  
 filterChain.doFilter(request,response);  
 }  
   
}*

**2. Controller for materials management**

*package com.boyer.controller;  
  
import com.boyer.pojo.Material;  
import com.boyer.pojo.ResponseResult;  
import com.boyer.service.MaterialService;  
import com.boyer.utils.CacheUtils;  
import com.boyer.utils.ResultResponseUtils;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.beans.factory.annotation.Qualifier;  
import org.springframework.stereotype.Controller;  
import org.springframework.util.StringUtils;  
import org.springframework.web.bind.annotation.\*;  
  
import java.io.IOException;  
import java.util.List;  
import java.util.UUID;  
  
@Controller  
public class MaterialController{  
  
 @Autowired  
 @Qualifier("MaterialServiceImpl")  
 private MaterialService materialService;  
  
  
 @RequestMapping(value = "/findMaterials",consumes = "application/json",produces = "application/json",method = RequestMethod.GET)  
 @ResponseBody  
 public ResponseResult<Object> findMaterials(@RequestBody Material material) throws IOException{  
 System.out.println("list");  
 List<Material> list = materialService.queryMaterialsByType(material.getMaterial\_type());  
 System.out.println(material.getMaterial\_type());  
 if(!list.isEmpty()){  
 String token = UUID.randomUUID().toString().replace("-","");  
 CacheUtils.put(token,list);  
 System.out.println(list);  
 return ResultResponseUtils.success(token,list);  
 }else{  
 return ResultResponseUtils.error("");  
 }  
 }  
  
 @RequestMapping(value = "/addMaterial",consumes = "application/json",method = RequestMethod.PUT)  
 @ResponseBody  
 public ResponseResult<Object> addMaterial(@RequestBody Material material) throws IOException{  
 System.out.println("");  
 if(!StringUtils.isEmpty(material)){  
 materialService.saveMaterial(material);  
 String token = UUID.randomUUID().toString().replace("-","");  
 CacheUtils.put(token,null);  
 System.out.println(""+material);  
 return ResultResponseUtils.success(token);  
 }else{  
 return ResultResponseUtils.error("");  
 }  
 }  
  
  
 @RequestMapping(value = "/listMaterials",produces = "application/json",method = RequestMethod.GET)  
 @ResponseBody  
 public ResponseResult<Object> listMaterials() throws IOException{  
 System.out.println("list");  
 List<Material> list = materialService.queryMaterials();  
 if(!list.isEmpty()){  
 String token = UUID.randomUUID().toString().replace("-","");  
 CacheUtils.put(token,list);  
 System.out.println(list);  
 return ResultResponseUtils.success(token,list);  
 }else{  
 return ResultResponseUtils.error("");  
 }  
  
 }  
  
 @RequestMapping(value = "/updateMaterial",consumes = "application/json",method = RequestMethod.PUT)  
 @ResponseBody  
 public ResponseResult<Object> updateMaterial(@RequestBody Material material) throws IOException{  
 System.out.println("");  
 System.out.println(material);  
 Material material1 = materialService.queryMaterialById(material.getMaterial\_id());  
 if(!StringUtils.isEmpty(material1)){  
 materialService.updateMaterial(material);  
 String token = UUID.randomUUID().toString().replace("-","");  
 CacheUtils.put(token,null);  
 return ResultResponseUtils.success(token);  
 }else{  
 return ResultResponseUtils.error("");  
 }  
  
 }  
  
 @RequestMapping(value = "/deleteMaterial",consumes = "application/json",method = RequestMethod.DELETE)  
 @ResponseBody  
 public ResponseResult<Object> deleteMaterial(@RequestBody Material material) throws IOException{  
 System.out.println("");  
 if(!StringUtils.isEmpty(material)){  
 materialService.deleteMaterialById(material.getMaterial\_id());  
 String token = UUID.randomUUID().toString().replace("-","");  
 CacheUtils.put(token,null);  
 return ResultResponseUtils.success(token);  
 }else{  
 return ResultResponseUtils.error("");  
 }  
 }  
  
 @RequestMapping(value = "/studyMaterial/{userId}",consumes = "application/json",method = RequestMethod.PUT)  
 @ResponseBody  
 public ResponseResult<Object> studyMaterial(@RequestBody Material material , @PathVariable("userId") String userId) throws IOException{  
 System.out.println("");  
 if(!StringUtils.isEmpty(material)||userId==null){  
  
// materialService.deleteMaterialById(material.getMaterial\_id());  
 String token = UUID.randomUUID().toString().replace("-","");  
 CacheUtils.put(token,null);  
 return ResultResponseUtils.success(token);  
 }else{  
 return ResultResponseUtils.error("");  
 }  
 }  
}*

**3. Points management controller**

*package com.boyer.controller;  
  
import com.boyer.pojo.PersonalPoint;  
import com.boyer.pojo.ResponseResult;  
import com.boyer.pojo.User;  
import com.boyer.service.PersonalPointService;  
import com.boyer.utils.CacheUtils;  
import com.boyer.utils.ResultResponseUtils;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.beans.factory.annotation.Qualifier;  
import org.springframework.stereotype.Controller;  
import org.springframework.util.StringUtils;  
import org.springframework.web.bind.annotation.RequestBody;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RequestMethod;  
import org.springframework.web.bind.annotation.ResponseBody;  
  
import java.io.IOException;  
import java.util.Date;  
import java.util.List;  
import java.util.UUID;  
  
@Controller  
public class PersonalPointController {  
 @Autowired  
 @Qualifier("PersonalPointServiceImpl")  
 PersonalPointService personalPointService;  
  
 @RequestMapping(value = "/addPoint",consumes = "application/json",method = RequestMethod.PUT)  
 @ResponseBody  
 public ResponseResult<Object> addPoint(@RequestBody PersonalPoint personalPoint) throws IOException {  
 System.out.println("");  
 System.out.println(personalPoint);  
 personalPoint.setTime(new Date().toString());  
 personalPointService.addPoint(personalPoint);  
 if(!StringUtils.isEmpty(personalPoint)){  
 String token = UUID.randomUUID().toString().replace("-","");  
 CacheUtils.put(token,null);  
 return ResultResponseUtils.success(token);  
 }else{  
 return ResultResponseUtils.error("");  
 }  
 }  
  
 @RequestMapping(value = "/queryPersonPointById",consumes = "application/json",produces = "application/json",method = RequestMethod.GET)  
 @ResponseBody  
 public ResponseResult<Object> queryPersonPointById(@RequestBody User user) throws IOException {  
 System.out.println("");  
 System.out.println(user.getUser\_id());  
 if(!StringUtils.isEmpty(user)){  
 List<PersonalPoint> list = personalPointService.queryPersonPointById(user.getUser\_id());  
 String token = UUID.randomUUID().toString().replace("-","");  
 CacheUtils.put(token,list);  
 return ResultResponseUtils.success(token,list);  
 }else{  
 return ResultResponseUtils.error("");  
 }  
 }  
}*

**4. The controller for question management, including question creation and verification of question answers.**

*package com.boyer.controller;  
  
import com.boyer.pojo.Choice;  
import com.boyer.pojo.ResponseResult;  
import com.boyer.service.ChoiceService;  
import com.boyer.utils.CacheUtils;  
import com.boyer.utils.ResultResponseUtils;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.beans.factory.annotation.Qualifier;  
import org.springframework.stereotype.Controller;  
import org.springframework.util.StringUtils;  
import org.springframework.web.bind.annotation.RequestBody;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RequestMethod;  
import org.springframework.web.bind.annotation.ResponseBody;  
  
import java.io.IOException;  
import java.util.List;  
import java.util.UUID;  
  
@Controller  
public class ChoiceController {  
  
 @Autowired  
 @Qualifier("ChoiceServiceImpl")  
 private ChoiceService choiceService;  
  
 @RequestMapping(value = "/addChoice",consumes = "application/json",method = RequestMethod.PUT)  
 @ResponseBody  
 public ResponseResult<Object> addChoice(@RequestBody Choice choice) throws IOException {  
 System.out.println("");  
 if(!StringUtils.isEmpty(choice)){  
 choiceService.saveChoice(choice);  
 String token = UUID.randomUUID().toString().replace("-","");  
 CacheUtils.put(token,null);  
 System.out.println(""+choice);  
 return ResultResponseUtils.success(token);  
 }else{  
 return ResultResponseUtils.error("");  
 }  
 }  
  
 @RequestMapping(value = "/deleteChoice",consumes = "application/json",method = RequestMethod.DELETE)  
 @ResponseBody  
 public ResponseResult<Object> deleteChoice(@RequestBody Choice choice) throws IOException {  
 System.out.println("");  
 if(!StringUtils.isEmpty(choice)){  
 choiceService.deleteChoiceById(choice.getQuestion\_id());  
 String token = UUID.randomUUID().toString().replace("-","");  
 CacheUtils.put(token,null);  
 System.out.println(""+choice.getQuestion\_id());  
 return ResultResponseUtils.success(token);  
 }else{  
 return ResultResponseUtils.error("");  
 }  
 }  
  
 @RequestMapping(value = "/editChoice",consumes = "application/json",method = RequestMethod.PUT)  
 @ResponseBody  
 public ResponseResult<Object> editChoice(@RequestBody Choice choice) throws IOException {  
 System.out.println("");  
 Choice choice1 = choiceService.queryChoiceById(choice.getQuestion\_id());  
 if(!StringUtils.isEmpty(choice1)){  
 choiceService.updateChoice(choice);  
 String token = UUID.randomUUID().toString().replace("-","");  
 CacheUtils.put(token,null);  
 System.out.println(""+choice.getQuestion\_id());  
 return ResultResponseUtils.success(token);  
 }else{  
 return ResultResponseUtils.error("");  
 }  
 }  
  
 @RequestMapping(value = "/listChoices",produces = "application/json",method = RequestMethod.GET)  
 @ResponseBody  
 public ResponseResult<Object> listChoices() throws IOException{  
 System.out.println("");  
 List<Choice> list = choiceService.queryChoices();  
 if(!list.isEmpty()){  
 String token = UUID.randomUUID().toString().replace("-","");  
 CacheUtils.put(token,list);  
 System.out.println(list);  
 return ResultResponseUtils.success(token,list);  
 }else{  
 return ResultResponseUtils.error("");  
 }  
  
 }  
  
 @RequestMapping(value = "/findChoice",consumes = "application/json",produces = "application/json",method = RequestMethod.GET)  
 @ResponseBody  
 public ResponseResult<Object> findChoice(@RequestBody Choice choice) throws IOException{  
 System.out.println("");  
 System.out.println(choice.getQuestion\_id());  
 Choice choice1 = choiceService.queryChoiceById(choice.getQuestion\_id());  
 if(!StringUtils.isEmpty(choice1)){  
 String token = UUID.randomUUID().toString().replace("-","");  
 CacheUtils.put(token,choice1);  
 System.out.println(choice1);  
 return ResultResponseUtils.success(token,choice1);  
 }else{  
 return ResultResponseUtils.error("");  
 }  
 }  
  
 @RequestMapping(value = "/checkChoice",consumes = "application/json",produces = "application/json",method = RequestMethod.POST)  
 @ResponseBody  
 public ResponseResult<Object> checkChoice(@RequestBody Choice choice) throws IOException{  
 System.out.println("");  
 Choice choice2 = choiceService.queryChoiceById(choice.getQuestion\_id());  
 if(!StringUtils.isEmpty(choice2)){  
 Boolean result;  
 if(choice2.getQuestion\_answer().equals(choice.getQuestion\_answer()))  
 result = true;  
 else  
 result = false;  
 String token = UUID.randomUUID().toString().replace("-","");  
 CacheUtils.put(token,result);  
 System.out.println(choice.getQuestion\_id()+""+choice.getQuestion\_answer()+""+result+" ");  
 return ResultResponseUtils.success(token,result);  
 }else{  
 return ResultResponseUtils.error("");  
 }  
 }  
  
}*

**5. Tool class used to encapsulate sending data to the backend**

*package com.boyer.utils;  
import com.boyer.pojo.ResponseResult;  
import java.io.IOException;  
public class ResultResponseUtils {  
 public static ResponseResult<Object> success(Integer code, String message, Object data) throws IOException{  
 return new ResponseResult<Object>(code,message,data);  
 }  
  
 public static ResponseResult<Object> success(String message, Object data) throws IOException{  
 return success(0,message,data);  
 }  
  
 public static ResponseResult<Object> success(Object data) throws IOException{  
 return success("",data);  
 }  
  
 public static ResponseResult<Object> error(Integer code, String message) throws IOException{  
 return new ResponseResult<Object>(code,message,null);  
 }  
  
 public static ResponseResult<Object> error(String message) throws IOException{  
 return error(-1,message);  
 }  
  
 public static ResponseResult<Object> error() throws IOException{  
 return error("");  
 }  
}*