Online-Shopping Web Application

https://github.com/Tianle97/BigProject

Tianle Shu

B. Sc.(Hons) in Software Development

Final Year Project

Supervisor: Martin Hynes

Department of Computer Science and Applied Physics
Galway-Mayo Institute of Technology (GMIT)

Chapter 1

Abstract:

This final year Project is Online-Shopping Web Application. Online shopping is a new type of shopping that is popular nowadays, and it is a kind of commercial shopping activity that is popular among young people. It is an E-commerce model. Through the Internet as a platform, buyers complete the entire shopping process through online shopping, online banking and other payment methods. The advantage of online shopping is mainly because it does not require entities, and the benefits are relatively large for customers and sellers, which makes online shopping develop quite rapidly in recent years.

More and more people will choose to shop on the social media platform. With the improvement of the social shopping function, the social media platform is no longer just an advertising channel, and people can now conveniently and quickly purchase goods on the social media platform of their choice.

Social media channels such as Instagram, Twitter, Pinterest, Facebook, and YouTube have launched "buy" buttons and have significantly improved their social sales capabilities. Take Instagram as an example, launching the “shoppable post” feature and allowing companies to post product tags in posts and use product stickers in Stories.

In this project’s technologies:
About the Front-end Client Server: Python, flask-framework, jinjia2, html, JavaScript.
About the Back-end Server: Springboot, MongoDB and MySQL.

This is a project with the objective to develop a basic website where consumer is provided with a shopping cart application and to know about the technologies used to develop such as an application.

Table of Contents

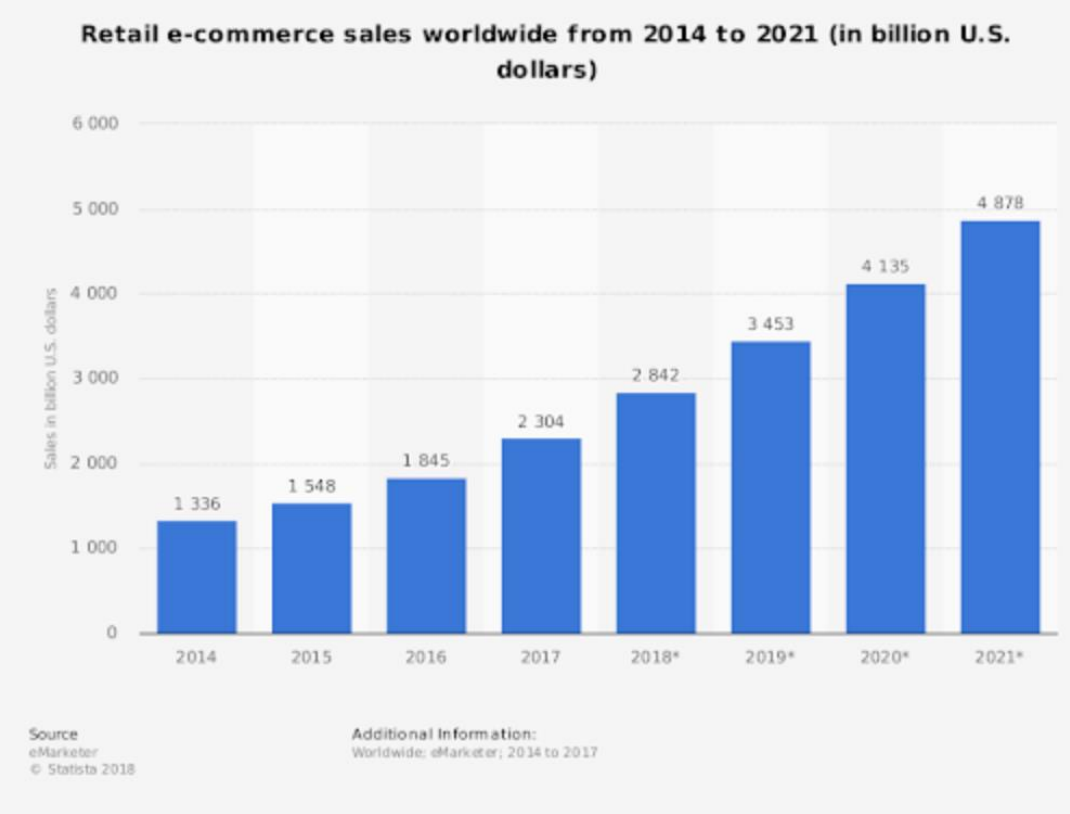
Chapter2

Introduction:

In this project, My back-end database use MySQL (storage user information) and MongoDB (storage the product information and user's order information), Server-side scripting language Springboot relational databases (MySQL and MongoDB). For my Front-end client server, I use Flask that is an API of Python and that can allow to build up web-applications. And HTML use as for the page look like and I use jinjia2 as well for every page logo tittle.

2.1 Online-Sopping Forecast[1]

The new study predicts that by 2021, global e-commerce sales will reach new highs. The global e-commerce business is expected to grow by 265%, from $1.3 trillion in 2014 to $4.9 trillion in 2021, indicating that the e-commerce market is not showing signs of decline in the future and will rise steadily.



As the boundaries between physical stores and the digital environment become more blurred, multi-channel shopping will become more common. According to statistics, 73% of customers use multiple channels in the shopping process, which proves the trend of multi-channel shopping.

For e-commerce sellers, this means understanding the way customers buy, the marketing channels they participate in, the motivation to buy, and the main drivers. In short, omni-channel shopping means gaining insight into what products people buy in a channel, where to buy, when to buy, why to buy, and how to buy.

There are many ways to shop in multiple channels. For example, people can research a product online, then buy it in the store, or choose to shop online, and pick it up in the store. The more channels your shoppers use, the more likely the average order value will increase. For example, customers who use more than four shopping channels spend an average of 9% more in the store than customers who use only one channel.

Every touch point is important, because even if you only have one puzzle missing, you can't sketch a complete story. Knowing their touch points before customers buy will give you a better idea of how to promote your product and allocate your marketing budget.

In 2019, you should integrate offline and online sales attribution into a single and coherent multi-channel shopping. You can create convenient purchase touch points for customers who conduct online research and then purchase offline. You can also use online purchases and offline self-raising strategies to allow customers to pick up their own goods at nearby stores after online shopping. This also means that your offline and online data should be kept in sync so you can make faster and smarter business decisions.

2.2 Aim and Objects

The Project can any member can register and view available products. User can login and logout their account. Add items to shopping bag and delete it before submission. Only registered member can purchase multiple products regardless

of quantity. User also can view the history of buying and comments on the product.

The objects of this project:

- To develop an easy way to use web-based interface where users can search for product view the details of the product and order it without going to market.
- The searching product can be done by product category, manufacturer as well as latest product, view it purchase it become a convenient way for customer.
- A user can view the complete specification of the product with various.
- It also facilitates the service provider to know the current stats of market and take decision which product are selling more nowadays and have to keep in store.

Chapter 3

Methodology

3.1 Software Design

In this project, My back-end database use MySQL (storage user information) and MongoDB (storage the product information and user's order information), Server-side scripting language Springboot relational databases (MySQL and MongoDB). For my Front-end client server, I use Flask that is an API of Python and that can allow to build up web-applications. And HTML use as for the page look like and I use jinjia2 as well for every page logo tittle.

3.2 Research Methodology

Firstly, as a student, we have never had the opportunity to work in a real environment, such as: methods are used as the basis for software development. In theory, we have studied various methods, as well as its processes and techniques, but without practical usage and enough resources, I can't say that I have completely developed this project. I am picking up enough reliable

technology to enhance my project development and try to get as much knowledge as possible.

At the beginning of my project development, I just used Springboot to connect to the database and the front-end page. When I happened to chance, I started to see the learning video of the Python Flask framework, and the use of jinjia2. And start learning and redesigning my front-end logic. We believe it will improve my project development and generate positive learning outcomes from first-hand experience.

3.3 Requirements of project

In order to start developing online shopping in one place, a simple enough idea must be presented to showcase the concept of online shopping, which is the design of the "Shopping-Online" project. I think if users can sign up, log in, be able to buy items and manipulate their shopping history.

Requirements:

- **User Registration** - Users can register with a unique username. The newly registered user information is saved to the database (MySQL). The password must be encrypted (the front end is encrypted, Werkzeug package) before saving the password to the database.
- **Add product details** - User can upload the product what want to sale, use base64 package from python, that can make the photo change to base64 to storage in database (MongoBD).
- **Index Page** - User can view every items detail when they are not login, so index page, there are also show the photo about the item, so we need take the details from database (MongoDB) and change base64 to photo to show the Index page.
- **Buy page** - If user want to buy some item, just click the picture, and the will go to the buy page (user must logined). If user input empty or 0, it will the red words to alarm user, so must input valid value.
- **Show User order details** - User can click the “shoppingbag” link to order page to see the order details. (user must login in advance)

- **Delete User order details** - If user login in advance, go to the order page and then user can choose product which is bought and want to delete.

3.4 Software Environment

- Development Tool: Visual Studio Code and eclipse.
- Browser: Google Browser
- Database: MongoDB and MySQL

3.5 Software Development Tools Brief

- **Visual Studio code (download link:** https://code.visualstudio.com/download)

  In the front-end development, there is a very useful tool, Visual Studio Code, referred to as VS code.
  Visual Studio Code is a source code editor developed by Microsoft for Windows, Linux and macOS. It includes support for debugging, embedded Git control, syntax highlighting, intelligent code completion, snippets, and code refactoring. It is also customizable, so users can change the editor's theme, keyboard shortcuts, and preferences. It is free and open source.[2]

  Visual Studio Code supports a variety of programming languages. Unlike other editors, many of Code's features are not displayed in the menu, but are called through the Command Panel. Users can search for and enter commands in the command panel to implement specific functions such as installing plugins, setting properties, and more. Code has the features of a modern text editor such as changing character encoding, replacing line breaks, and more.

  Features:
  1. Open source, free of charge.
  2. Custom configuration.
  3. Integrate git.
  4. Smart tips are powerful.
  5. Support various file formats (html/jade/css/less/sass/xml).

6. Powerful debugging.
7. Various convenient shortcuts.
8. Powerful plugin extension.

- **Eclipse (download :** https://www.eclipse.org/downloads/ **)**

  Eclipse is an open source, Java-based, extensible development platform.

  Eclipse is Java's integrated development environment (IDE), of course, Eclipse can also be used as an integrated development environment for other development languages, such as C, C++, PHP, and Ruby.

  Eclipse comes with a standard set of plugins, including the Java Development Kit (JDK).[3]

  Feature:
  1.NLS string hover has an Open in Properties File action
  2. In Caller mode, the Call Hierarchy has an Expand With Constructors action in the context menu.
  3. When you type in the editor, the Java compare editor will update its structure.
  4. There is a new toString() generator
  5. Added an Open Implementation link for the overlay method, which can be directly opened.
  6. The editor is consistent with the execution environment
  7.Debug view now provides breadcrumb (breadcrumb) showing the active debug context
  8. The runnable JAR file output wizard can also package the required class library into a runnable JAR file to be output, or package it into a directory next to the JAR.
  9. The compiler can issue a warning if it detects unwanted code
  10. The path to the class library, variable, or container entry can be anywhere related to the project
  11. In the Jovadoc hover header and Javadoc view, links to other types and members are provided.
  12. The JUnit4 version released with the Eclipse is updated to 4.5.
  13. Javadoc view and hovers support {@inheritDoc} tags and add links to override methods

14. The comparison of the same value is now detected by the compiler and a warning is issued by default

3. Github

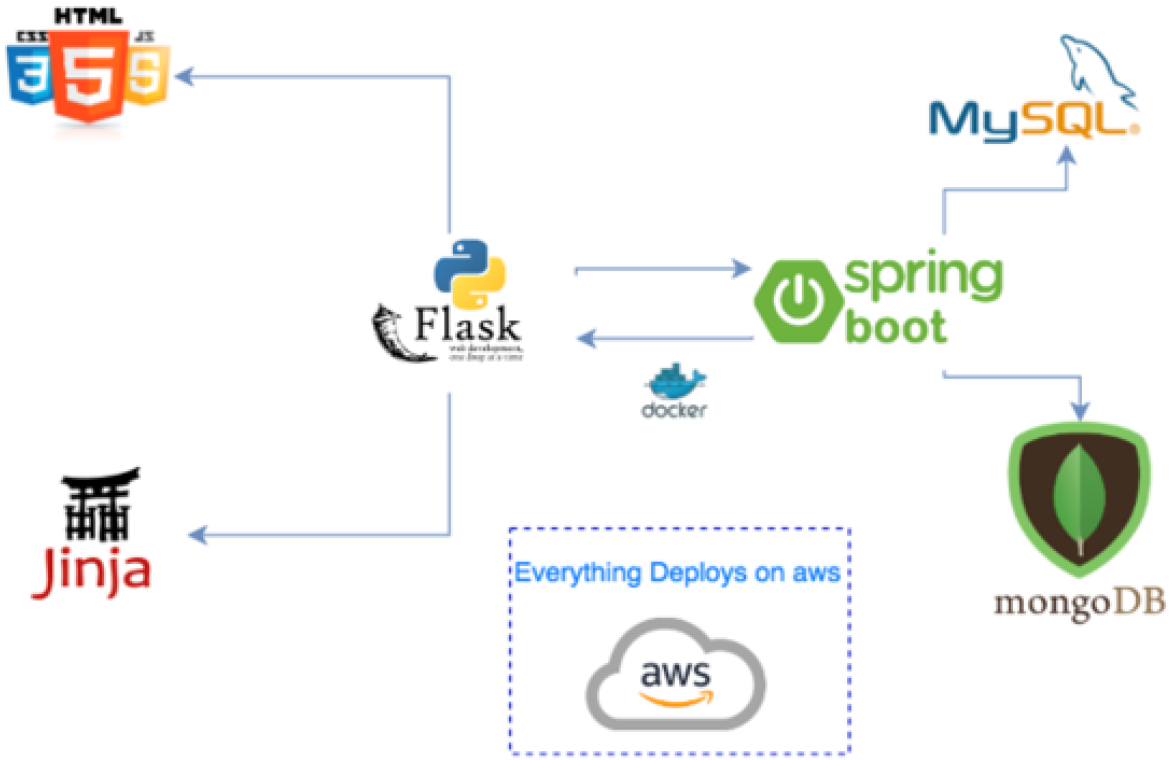
GitHub (originally called Logical Awesome LLC) is a web-based hosting service that uses git for version control. Mainly used for computer code. It provides all of Git's distributed version control and source code management (SCM) features and adds its own features. Provides access control and various collaboration features for each project, such as bug tracking, feature requests, task management, wikis, etc.

The Github API is my main code collaboration development tool for increasing the software development of the Shopping-Online project. Github repository link:https://github.com/Tianle97/BigProject

Chapter 4

Technology Review

4.1 Technology Design pattern



4.2 Back-end Server

Apache Maven is a software project management and comprehension tool. It is a tool that can now be used for building and managing any Java-based project. I hope that I have created something that will make the day-to-day work of Java developers easier and generally help with the comprehension of any Java-based project.

Maven's primary goal is to allow a developer to comprehend the complete state of a development effort in the shortest period of time. In order to attain this goal there are several areas of concern that Maven attempts to deal with making the build process easy, providing a uniform build system, providing quality project information, providing guidelines for best practices development and allowing transparent migration to new features. Based on the concept of a project object model (POM), *Maven* can manage a project's build, reporting and documentation from a central piece of information.

There is similar project build tool called ***Gradle***. *Maven* uses an XML-based configuration and *Gradle* uses the configuration of a domain-specific language Groovy. Since our course is linked to maven, so I use maven in this *Online-Shopping* project.
Takes an opinionated view of building production-ready *Spring* applications. **Spring Boot** favors convention over configuration and is designed to get you up and running as quickly as possible.

Spring Boot is Spring, it does the Spring Bean configuration that you will do without it. It uses the concept of "habits better than configuration" (there are a lot of configurations in the project, and a customary configuration built in, so you don't need to manually configure it) to get your project up and running quickly. Using Spring Boot, it's easy to create a stand-alone (running jar, embedded servlet container), quasi-production-level Spring framework-based project. With Spring Boot, you can use less or just a few Spring configurations.

Spring Boot allows you to quickly create Spring-based projects. You only need a little configuration to run this Spring project.

Benefit to use Springboot:

1. Standalone Spring application.
    a. Spring Boot can be run as a jar package. Useful for running the Spring Boot project, only need to run the java -jar xx.jar class.
2. There is no need to attach WAR files directly to Tomcat, Jetty or Undertow. Continue with the PET initiator POM to facilitate the design of Maven Configure Spring as automatically as possible
3. Provide ready-to-use features such as indicators, health checks, external settings
4. Absolutely no code generation required, no XML configuration required

Springboot brings a lot of magic to the development of Spring applications, the most important of which are the following four cores:

- Auto-configuration: Spring Boot automatically provides configuration for many of the common application features of Spring applications.
- Start-up Dependency: Tell Spring Boot what features it needs to be able to introduce the required libraries.

- Command Line Interface: This is an optional feature of Spring Boot, so you can write a complete application without writing a traditional project.
- Actuator: Lets you get a deeper set of Spring Boot applications.

For example, the directory results suggested by Spingboot are as follows:

Root Package Structure:
1, Application.java is recommended to be placed under the directory, mainly used to do some framework configuration

2, Model: directory is mainly used for entities (Entity) and data access layer (Repository)

3, the service layer is mainly the business class code

4, controller is responsible for page access control

As you can see, create a new Springboot, which basically has no code, in addition to a few empty directories, it also contains the following things.

1. Chapter1Application.java: A class with a main() method that launches the application (critical).
2. Chapter1ApplicationTests.java: An empty JUnit test class that loads a Spring application context that uses the Spring Boot dictionary configuration feature.
3. application. Properties: An empty properties file, you can add configuration properties as needed.

Spring Boot parent dependency

This configuration is the Spring Boot parent dependency. With this, the current project is the Spring Boot project. Spring-boot-starter-parent is a special starter that provides the relevant Maven default dependencies. After using it, Common package dependencies can save the version tag.
Pom.xml

```
<parent>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-parent</artifactId>
```

```
        <version>2.0.2.RELEASE</version>
        <relativePath/> <!-- lookup parent from repository -->
</parent>
```

Not everyone likes to inherit from the spring-boot-starter-parent POM. You may have your own company standard parent that you need to use, or you may prefer to explicitly declare all Maven configurations.
If you don't want to use spring-boot-starter-parent, you can still maintain dependency management by using the scope=import

Add Web Partern in pom.xml

```
<dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-web</artifactId>
 </dependency>
```

There are two modules by default in the pom.xml file:

Spring-boot-starter: core module, including automatic configuration support, logging, and YAML;

Spring-boot-starter-test: Test modules, including JUnit, Hamcrest, Mockito.

Spring Boot Maven plugin

```
<plugin>
                <groupId>com.spotify</groupId>
                <artifactId>docker-maven-plugin</artifactId>
                <version>1.0.0</version>
                <configuration>
                 <dockerDirectory>${project.basedir}/src/main/docker</dockerDirectory>
                    <imageName>${docker.image.prefix}/${project.artifactId}</imageName>
                    <dockerDirectory>${project.basedir}/src/main/docker</dockerDirectory>
                    <resources>
                        <resource>
                            <targetPath>/</targetPath>
                            <directory>${project.build.directory}</directory>
                            <include>${project.build.finalName}.jar</include>
                        </resource>
                    </resources>
                </configuration>
        </plugin>
```

The above configuration is the Spring Boot Maven plugin, and the Spring Boot Maven plugin provides a number of handy features:

Package the project into an executable super-JAR (uber-JAR), including all the dependencies of the application into the JAR file, and add a description file for the JAR, which allows you to run the application with java -jar program.
Search for the public static void main() method to mark it as a runnable class.

Development environment debugging

Hot start is already very common in normal development projects. Although the process of developing a web project usually changes the project restart, it always reports an error. However, Spring Boot has good debugging support. It can be effective in real time after modification. You need to add the following configuration:

```
<dependencies>
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-devtools</artifactId>
    <optional>true</optional>
  </dependency>
</dependencies>
<build>
  <plugins>
    <plugin>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-maven-plugin</artifactId>
      <configuration>
        <fork>true</fork>
      </configuration>
    </plugin>
  </plugins>
</build>
```

Sameple Code

```java
@RestController
public class UserController {
	@Autowired
	private UService uService;


	// Server console logger
	private static Logger logger = LoggerFactory.getLogger(UserController.class);
```

```
@PostMapping("/register")
public Resp registerPost(@RequestBody User user) {
	if(user.getUsername()== null || user.getPassword() == null || user.getAddress() == null || user.getPhone() == null)
		//uService.save(user);
		return new Resp("error");
	if (uService.findByUsername(user.getUsername()) == null && user.getUsername() != null) {
		uService.save(user);
		return new Resp("registered");
	}else {
		return new Resp("duplicate_user");
	}
```

```
@SpringBootApplication
//@SpringBootApplication(exclude={DataSourceAutoConfiguration.class,HibernateJpaAutoConfiguration.class})
public class OnlineShoppingApplication {

public static void main(String[] args) {
	SpringApplication.run(OnlineShoppingApplication.class, args);
}
```

@SpringBootApplication is the core annotation of the Sprnig Boot project, the main purpose is to enable automatic configuration. In-depth explanation will be given later when explaining the principle.
Main method This is a standard Java application's main method, the main role is to be the entry point for project startup.

The @RestController annotation is equivalent to the combination of @Controller+@ResponseBody, and the methods in the class that use this annotation are all output in json format.

4.3 Front-end Server (Python Flask)

Flask is a user network that uses Python. Microsoft is an important part of Flask. This means that Flask's purpose is to keep it simple, but at the same time it's easy. In general, Flask is not suitable for data convergence, formulas, or anything in most libraries. However, Flask supports the use of other resources to include these applications. Many offer a variety of data interfaces, design identification, download and different three sensing technologies. The owner of the device makes it very attractive for web development.

A minimal application looks like this:

```
from flask import Flask app = Flask(__name__) @app.route('/')
def hello_world():
return 'Hello World!' if __name__ == '__main__':
app.run()
```

Jinjia2 is Python's next widely used template engine. His design ideas are derived from Django's template engine and extend its syntax and a range of powerful features. One of the most notable is the addition of sandboxed execution and optional automatic escaping, which is very important for the security of most applications.
He is based on unicode and can run in versions after Python 2.4, including python3.

Feature of the Jinjia2:
1. Sandbox execution mode, each part of the template is executed under the supervision of the engine, the template will be explicitly marked in the whitelist or blacklist, so that those templates that are not trusted can also be executed.
2. Powerful automatic HTML escaping system that effectively blocks cross-site scripting attacks.
Template inheritance mechanism, this mechanism can make all templates have a similar and consistent layout, and also facilitate the developer to modify and manage the template.
3. Efficient execution efficiency, the Jinja2 engine converts the source code into Python bytecode when the template is first loaded, speeding up template execution time.
4. Optional precompilation mode.
5. The debug system incorporates the standard Python TrackBack system.
Errors during template compilation and runtime can be discovered and debugged in a timely manner.
6. The syntax is configurable and you can reconfigure Jinja2 to better adapt to LaTeX or JavaScript output.
7. Template Designer Help Manual, which guides designers through the various ways to use the Jinja2 engine.

Firstly, I imported the Flask package in python.

```
from flask import Flask
```

The second step creates an instance of the Flask class. This line of code has a parameter __name__, this parameter is used to tell flask the name of your application, the official has a sentence:

If you are using a single module, __name__ is always the correct value. If you however are using a package, it' s usually recommended to hardcode the name of your package there. This means that if it is a single application, use __name__. If it is an application package, hardcode a name for this parameter. For example:

```
app = Flask("myApp")
```

Since our application is relatively simple at the moment, __name__ is used as a parameter.

Next use the route() decorator to indicate what kind of url can be used to access our function and return the information to be displayed in the browser in the function.@app.route('/')

```
def hello_world():
 return 'HelloWorld!'
```

You can also implement different url parsing by modifying the route() decorator. For example, we change to the following way:

```
@app.route('/index')
def hello_world():
 return 'Hello World!'
```

Run the programming:

```
$ python hello.py
* Running on http://127.0.0.1:5000/
```

access /index to display *hello world*

Remember: Make sure save it as hello.py (or a similar file) and run it with the Python interpreter. Make sure your application is not called flask.py, because this will conflict with Flask itself.

4.4 Databases

For storage the data, I use 2 databases that is MongoDB and MySQL. The user details storage in MySQL and the products details and user order details storage in MongoDB. Because MongoDB is NoSQL and it is can set index of any property in the MongoDB record.

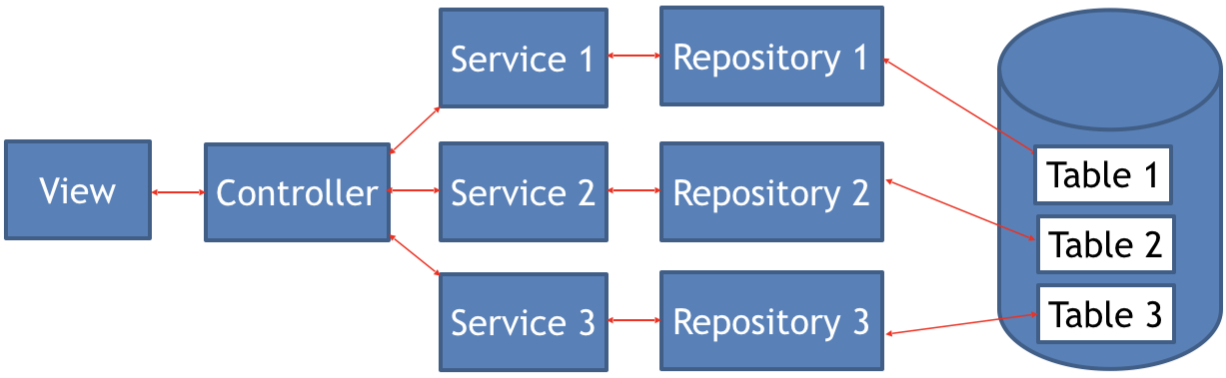
Last 2 years we study the how to use MongoDB and MySQL in third year course.
MySQL is a relational database management system developed by MySQL AB of Sweden and currently owned by Oracle Corporation. MySQL is an associative database management system that stores data in separate tables instead of putting all the data in a large repository, which increases speed and flexibility.
MongoDB is written in C++ and is an open source database system based on distributed file storage. Classified as a NoSQL database program, In the case of high load, adding more nodes can guarantee server performance.

4.4.1 MySQL

MySQL is a relational database management system developed by MySQL AB of Sweden and currently owned by Oracle. MySQL is one of the most popular relational database management systems. MySQL is the best RDBMS (Relational Database Management System) application for WEB applications.
MySQL is a relational database management system. Relational databases store data in different tables instead of putting all the data in a large repository, which increases speed and increases flexibility.
The SQL language used by MySQL is the most commonly used standardized language for accessing databases. MySQL software adopts dual authorization policy, which is divided into community version and commercial version. Due to its small size, fast speed, low total cost of ownership, especially open source, the development of small and medium-sized websites generally chooses MySQL as the website database.

MySQL connect with Spring boot diagram:

N-Tier Architecture in Spring Boot



Add *MySQL* dependency in the Maven pom file:

```
<dependency>
	<groupId>mysql</groupId>
	<artifactId>mysql-connector-java</artifactId>
</dependency>

<dependency>
	<groupId>org.springframework.boot</groupId>
	<artifactId>spring-boot-starter-jdbc</artifactId>
</dependency>
```

Edit application.yml in resource folder.

```
spring:
  datasource:
    driver-class-name: com.mysql.jdbc.Driver
    url: jdbc:mysql://127.0.0.1:3306/onlineshopping
    username: root
    password:
  jpa:
    hibernate:
      ddl-auto: update
    show-sql: true
```

Create table from eclipse to MySQL:

```
import javax.persistence.*;
```

```
@Entity
@Table(name="user_details")
public class User {
	@Id
	@GeneratedValue
	private int uid;
	private String username;
	private String password;
	private String phone;
	private String address;

	public User() {}
```

Hibernate will automatically convert the entity class into a table in the database. In the repository interface:

```
import org.springframework.data.repository.CrudRepository;
import org.springframework.stereotype.Repository;

import ie.gmit.sw.models.User;
// This will be AUTO IMPLEMENTED by Spring into a Bean called URepository
// CRUD refers Create, Read, Update, Delete
@Repository
public interface URepository extends CrudRepository<User,Long> {
```

The CrudRepository provides sophisticated CRUD functionality for the entity class being managed.

Arguments are:

- Domain Class to manage.
- Id type of the domain class.

@Repository – Indicates an interface is a repository.

The CrudRepository provides sophisticated CRUD functionality for the entity class that is being managed.

Default functionality does not even have to be specified in any interface that extends CrudRepository, e.g.:

save(S) – Updates a table with the specified entity.

findAll() – Returns all instances of a type

Delete(S) – Delete a row from a table

4.4.2 MongoDB

MongoDB is an open-source document database and leading NoSQL database. It is written in *C++*. MongoDB is a product between a relational database and a

non-relational database. It is the most feature-rich and relational database among non-relational databases.

MongoDB is designed to provide scalable, high-performance data storage solutions for web applications. *MongoDB* stores data as a document, and the data structure consists of key=>value pairs. *MongoDB* documents are similar to JSON objects. Field values can contain other documents, arrays, and document arrays.
For example:

The features of MongoDB:

1. MongoDB is a database for document storage that is simple and easy to operate.
2. You can set an index of any property in the MongoDB record (eg: FirstName="Sameer", Address="8 Gandhi Road") for faster sorting.
3. Data mirroring can be created locally or over the network, which makes MongoDB more scalable.
4. If the load increases (requires more storage space and more processing power), it can be distributed on other nodes in the computer network. This is called fragmentation.
5. Mongo supports rich query expressions. Query instructions use JSON-style tags to easily query objects and arrays embedded in a document.
6. MongoDb uses the update() command to replace a completed document (data) or some specified data field.
7. Map/reduce in Mongodb is mainly used for batch processing and aggregation of data.

8. Map and Reduce. The Map function calls emit(key, value) to iterate through all the records in the collection, passing the key and value to the Reduce function for processing.
9. Map functions and Reduce functions are written in Javascript, and MapReduce operations can be performed through db.runCommand or mapreduce commands.
10. GridFS is a built-in feature in MongoDB that can be used to store large numbers of small files.
11. MongoDB allows scripts to be executed on the server. You can write a function in Javascript, execute it directly on the server, or store the definition of the function on the server. You can call it directly next time.
12. MongoDB supports a variety of programming languages: RUBY, PYTHON, JAVA, C++, PHP, C# and many other languages.
13. MongoDB is easy to install.

Add *MongoDB* dependency in the Maven pom file:

```
<dependency>
	<groupId>org.springframework.boot</groupId>
	<artifactId>spring-boot-starter-data-mongodb</artifactId>
</dependency>
```

Add a command in application.yml in resource folder:

```
mongodb:
	uri: mongodb://127.0.0.1:27017/product
```

That means create database in MongoDB and named product.

Use annotations to create table in MongoBD:

```
import java.io.Serializable;
import java.math.BigDecimal;
import javax.persistence.*;

public class MongoProduct implements Serializable{

	/**
	 *
	 */
	private static final long serialVersionUID = 1L;
	@Id
	private String id;
	private String name;
	private BigDecimal price;
	private String type;
	private String photo;
	private int stocks;
```

It will create in mongodb and the table name is class name (MongoProduct) when initial run the program.

Similar to ***HibernateRepository***, by inheriting the ***MongoRepository*** interface, we can easily add, delete, and modify an object. To use the Repository function, first inherit the ***MongoRepository*** interface, where T is the bean class saved by the repository, and TD is the unique identifier of the bean. The type is generally ObjectId. Then inject the interface into the service and use it. There is no need to implement the method inside, and spring will automatically generate according to the defined rules.

However, MongoRepository implements only the most basic functions of adding, deleting, and changing. To add additional query methods, you can define the interface method according to the following rules. Custom query method, the format is "findBy + field name + method suffix", the parameter passed by the method is the value of the field, in addition to support paging query, by passing a Pageable object, returning the Page collection.

4.5 Docker

Docker is an open source software project that automates the deployment of applications under software containers, providing an additional layer of software abstraction and automatic management of operating system layer virtualization on the Linux operating system. Docker uses resource separation mechanisms in the Linux kernel, such as cgroups, and the Linux kernel namespace to create separate containers.

Docker Docker is further encapsulated on a container basis, from file system, network interconnection to process isolation, etc., greatly simplifying the creation and maintenance of containers. Make Docker technology lighter and faster than virtual machine technology.

The image below compares the differences between Docker and traditional virtualization. The traditional virtual machine technology is to virtualize a set of hardware, run a complete operating system on it, and then run the required application process on the system; and the application process in the container runs directly on the host kernel, and the container does not have its own The

kernel, and there is no hardware virtualization. Therefore, containers are lighter than traditional virtual machines.

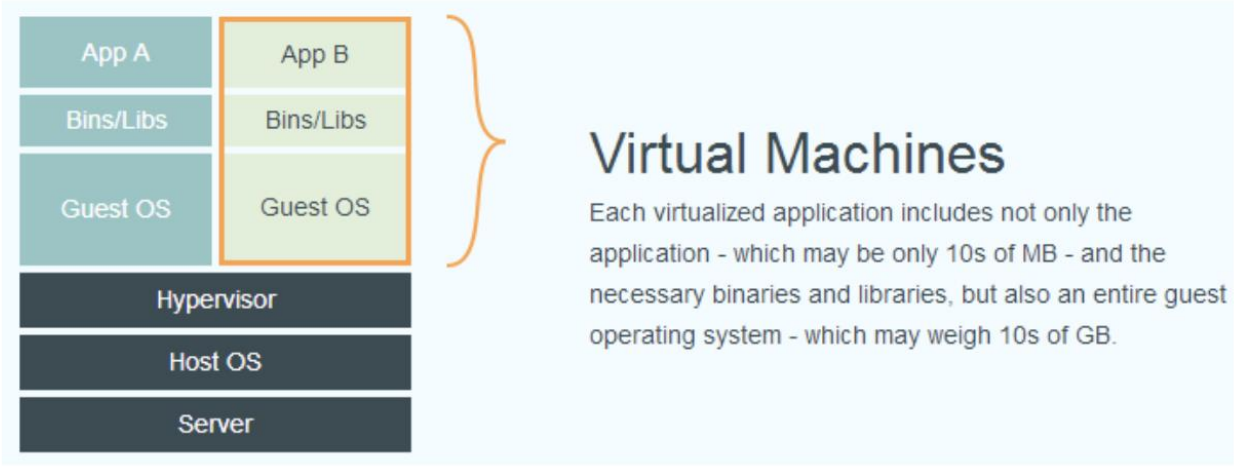


图 1.4.1.1 - 传统虚拟化

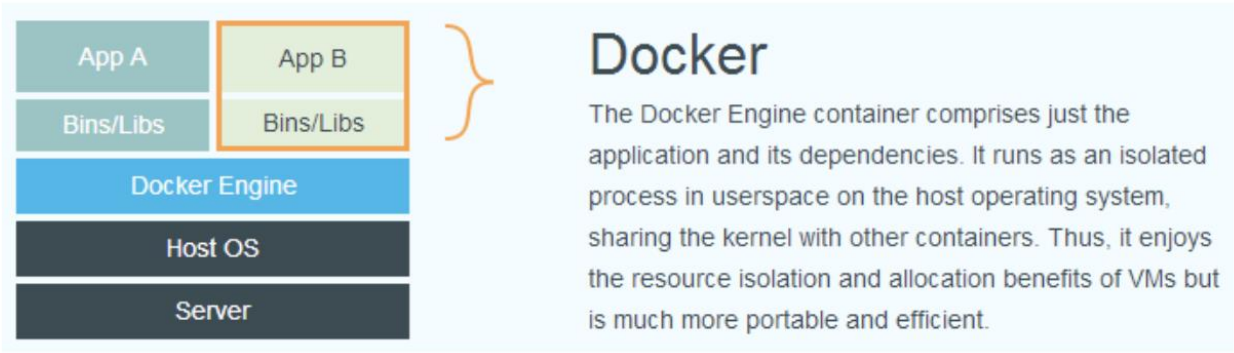


图 1.4.1.2 - Docker

Docker application scenario:

- Automated packaging and publishing of web applications.
- Automated testing and continuous integration, release.
- Deploy and tune your database or other background applications in a service-oriented environment.
- Compile or extend your existing OpenShift or Cloud Foundry platform to build your own PaaS environment.

The advantages of Docker:

1. Simplify the program:

Docker allows developers to package their applications and dependencies into a portable container and then publish them to any popular Linux machine for virtualization. Docker has changed the way virtualization is done so that developers can directly put their own results into Docker for management. Convenience and speed is already the biggest advantage of Docker. In the past, it took several days or even weeks to complete the task. In the Docker container, it only takes a few seconds to complete.

2. Avoid choosing phobias:
If you have a choice of phobia, you are still a senior patient. Docker helps you pack your tangles! For example, the Docker image; the Docker image contains the runtime environment and configuration, so Docker can simplify the deployment of multiple application instances. For example, web applications, background applications, database applications, big data applications such as Hadoop clusters, message queues, etc. can be packaged into a single image deployment.

3. Save money:
On the one hand, the advent of the cloud computing era, so that developers do not have to configure high-cost hardware in pursuit of results, Docker has changed the mindset of high performance and high price. The combination of Docker and the cloud allows cloud space to be more fully utilized. Not only solved the problem of hardware management, but also changed the way of virtualization.

Docker deployment problems

1. Publish or expose port

Normal Docker deploying process:

- create a Dockerfile
- add Docker dependency and plugin
- build Docker image
- **run Docker image**

Generally, command *server.port = port* in *application.yml* specifies a specific port for a spring project when it running. but if the project deploying with Docker, docker configuration is important as well. In other words, Docker will ignore the port set of spring project.

There are two options for setting port: `-p` & `-expose`
Docker run command:

```
$ docker run [OPTIONS] IMAGE [COMMAND] [ARG...]
```

- This binds port 8080 of the container to port 80 on localhost of the host machine.

  ```
  $ docker run  p 80:8080
  ```

- This exposes port 80 of the container without publishing the port to the host system's interfaces.

  ```
  $ docker run --expose 80
  ```

  *There is another way exposing a specific port: add **EXPOSE** command to Docker file*

  ```
  ~ ~
  EXPOSE 80
  ~ ~
  ```

1. Install Dock

1. For Desktops:

- Mac
- Windows

2. For Cloud Providers:

   AWS

   - Update the installed packages and package cache on your instance. $ sudo yum update -y
   - Install the most recent Docker Community Edition package. $ sudo yum install -y docker
   - Start the Docker service. $ sudo service docker start

3. For Servers:

   Ubuntu

- Update the apt package index.

  $ sudo apt-get update

- Install the latest version of Docker CE, or go to the next step to install a specific version. Any existing installation of Docker is replaced.

$ sudo apt-get install docker-ce

- On production systems, you should install a specific version of Docker CE instead of always using the latest. This output is truncated. List the available versions.

$ apt-cache madison docker-ce

Choose a specific version to install. To install a specific version, append the version string to the package name and separate them by an equals sign (=):

$ sudo apt-get install docker-ce=<VERSION>

- Verify that Docker CE is installed correctly by running.

$ sudo docker --version

2. Building Docker image with Maven

Containerize an existing project: Docker has a simple Dockerfile file format that it uses to specify the "layers" of an image. So let's create **src/main/docker** folder and create a Dockerfile in Spring Boot project:

```
FROM frolvlad/alpine-oraclejdk8:slim
VOLUME /tmp
ADD Online-Shopping-0.0.1-SNAPSHOT.jar app.jar
RUN sh -c 'touch /app.jar'
EXPOSE 8093
ENV JAVA_OPTS=""

ENTRYPOINT [ "sh", "-c", "java $JAVA_OPTS -Djava.security.egd=file:/dev/./urandom -
jar /app.jar"
```

demo-0.0.1-SNAPSHOT.jar can be founnd in the **target** folder, change the file name with your .jar file. you can also use IDE (Eclipse/Intellij etc.) to build a .jar file.

Add a new plugin to **pom.xml** :

```
<properties>
            <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
            <project.reporting.outputEncoding>UTF-8</project.reporting.outputEncoding>
            <java.version>1.8</java.version>
            <docker.image.prefix>onlineShopping</docker.image.prefix>
```

```
</properties>

<plugin>
            <groupId>com.spotify</groupId>
            <artifactId>docker-maven-plugin</artifactId>
            <version>1.0.0</version>
            <configuration>
            <dockerDirectory>${project.basedir}/src/main/docker</dockerDirectory>
            <imageName>${docker.image.prefix}/${project.artifactId}</imageName>
            <dockerDirectory>${project.basedir}/src/main/docker</dockerDirectory>
             <resources>
                <resource>
                     <targetPath>/</targetPath>
                     <directory>${project.build.directory}</directory>
                     <include>${project.build.finalName}.jar</include>
                </resource>
            </resources>
        </configuration>
</plugin>
```

Create Docker image

- cmd to the prohect folder
- run $ mvn package -Dmaven.test.skip=true docker:build

***Note:** Make sure Maven is installed. [Maven install tutorial]

Run Docker image

docker run --name=ProjectName -p 8080:8080 -t ImageName

4.6 Amazon Web Services (AWS)

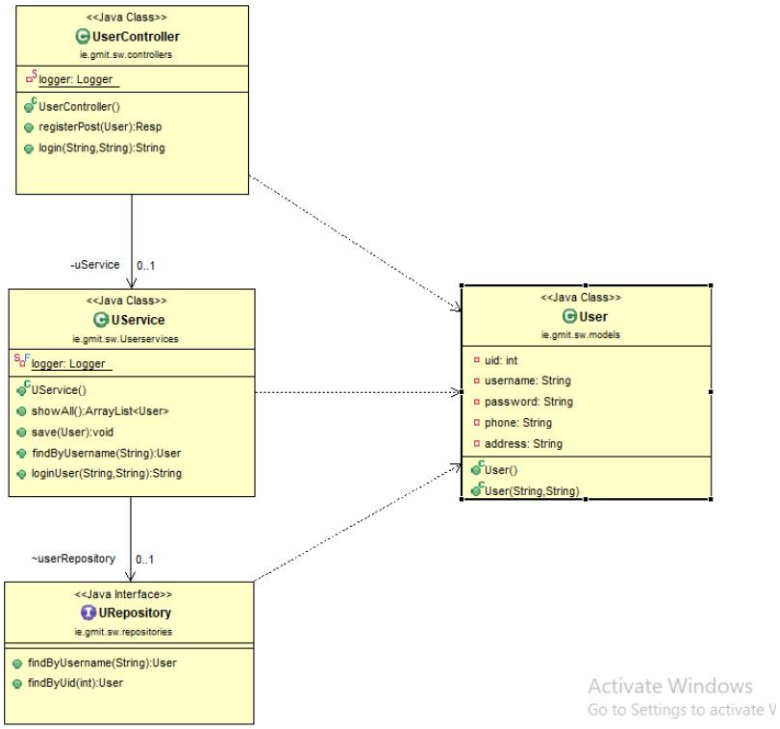
Amazon Web Services (English: Amazon Web Services, abbreviated AWS), a cloud computing platform created by Amazon, provides many remote web services. Both Amazon EC2 and Amazon S3 are architected on this platform. It was first publicly operated in July 2002, providing other websites and client-side services.

Whether you're running a photo sharing app with millions of mobile users or supporting the critical operations of your business, the cloud services platform gives you quick access to flexible, low-cost IT resources. With cloud computing, you don't have to invest heavily in hardware and spend a lot of time maintaining and managing the hardware.

Chapter 5

System Design

User Information Storage Design (login and register)



This is user System Back-end Design for user login and register, then if register successful it will storage in MySQL.

```
public String loginUser(String username, String password) {

	// Check if user exist in DB. Delegate this task to SqlDAO repository.
	// Hibernate provides a method findBy___(param) to perform querying.
	  if (userRepository.findByUsername(username) == null) {
		Logger.info("+++++++ Wrong Username! ++++++");
		return "no_user";
	  }
	  else {
		//If can search the same name in MySQL.
		//Let the 'passwor' which equals to
userRepository.findByUsername(username).getPassword()
		password = userRepository.findByUsername(username).getPassword();
		//return password compare in the front server
		return password;
	  }
    }
```

This is Login function in the Springboot, firstly will search the username get from front-end server in MySQL. If in MySQL not have that user name, then return the string ‘no_user’ to front-end. If match the username in database, will return password to front-end, it will compare the password which is user input from web-page.(Front-end login function is below)

```python
@app.route('/login',methods=['POST'])
def login_get():
    global user
    form = request.form
    username = form.get('username')
    password = form.get('password')
    print("pas: ",password)
    re = ServerLogin(username,password)
    if not username:
        flash("please input username !")
    elif not password:
        flash("please input password !")
    # Compare the password if result == true, it will go to route "logined"
    elif(check_password_hash(re,password) == True):
        session['username'] = username
        user = username
        return redirect("logined")
    else:
        flash("username/password is wrong!")
    return render_template("login.html")
```

About the register function, only have unique username in the database, if user input a username same with database, it will back a string “duplicate_user” to front-end.

```
@PostMapping("/register")
            public Resp registerPost(@RequestBody User user) {
                  if(user.getUsername()== null || user.getPassword() == null ||
user.getAddress() == null || user.getPhone() == null)
                        //uService.save(user);
                        return new Resp("error");
                  if (uService.findByUsername(user.getUsername()) == null &&
user.getUsername() != null) {
                        uService.save(user);
                        return new Resp("registered");
                  }else {
                        return new Resp("duplicate_user");
                  }
            }
```

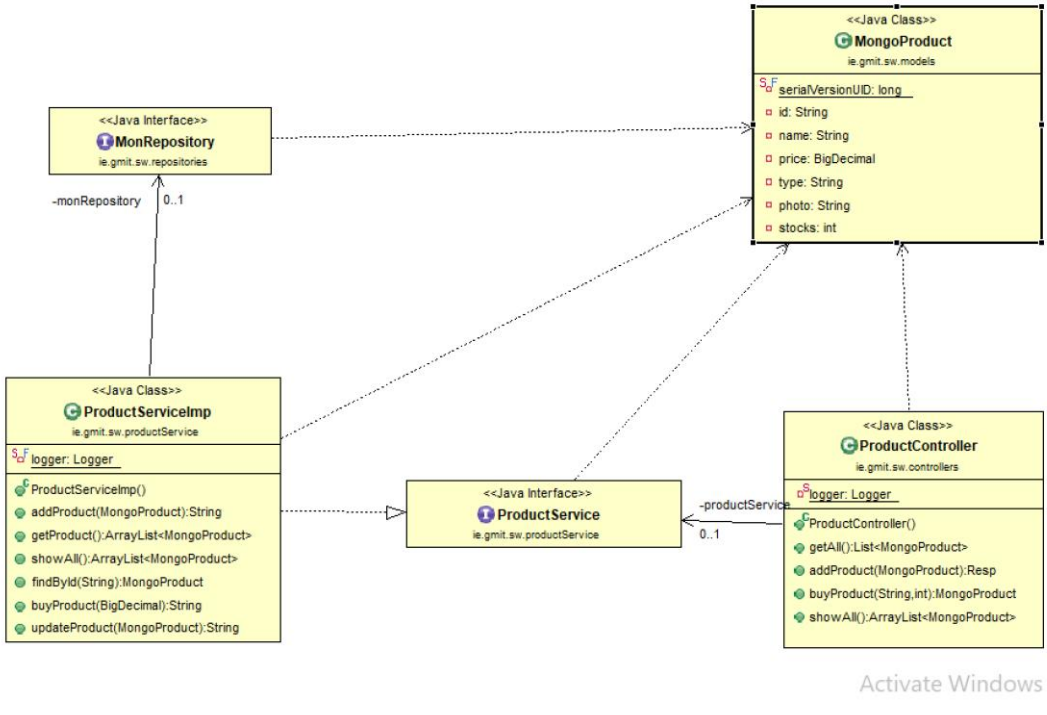
Below is the front-end server register function

```python
@app.route('/register')
def register_init():
    return render_template("register.html")

def ServerRegister(user,psd,add,ph):
    data = {'username': user, 'password': psd,'address':add,'phone':ph}
    url = 'http://127.0.0.1:8080/register'
    r = requests.post(url, json = data)
    Parsed_json = json.loads(r.text)
    return Parsed_json['msg']
```

```
@app.route('/user', methods=['POST'])
def register():
    form = request.form
    username = form.get('username')
    password = generate_password_hash(form.get('password'))
    print("pas: ", password)
    address = form.get('address')
    phone = form.get('phone')
    re = ServerRegister(username, password,address,phone)
    if not username:
        flash("please input username !")
        return render_template("register.html")
    elif not password:
        flash("please input password !")
        return render_template("register.html")
    elif(re == 'duplicate_user'):
        flash('ERROR: user exist!')
        return render_template("register.html")
    elif(re == 'registered'):
        flash('register success!')
        return render_template("login.html")
```

Product Information Storage Design

Add product function is like register function but except storage the photo storage to MongoDB. This is a **difficult point** in my project.



Difficult Point storage the product's picture

Add product function is like register function but except storage the photo storage to MongoDB. This is a **difficult point** in my project.
I use javascript to make the image convert to base 64
Below is the code to add a picture display on the page and then convert to base64 to back to back-end through the route for storage in MongoDB.

```
<link rel="stylesheet" href="{{ url_for("static", filename="css/addProduct.css") }}"
type="text/css"/>
    <script type="text/javascript" >
        function selectFile(){
            var files = document.getElementById('pic').files;
            console.log(files[0]);
            if(files.length == 0){
                return;
            }
            var file = files[0];
            //把上传的图片显示出来
            //display the picture when user choose a picture for upload
            var reader = new FileReader();
            // 将文件以 Data URL 形式进行读入页面
            console.log(reader);
            reader.readAsBinaryString(file);
            reader.onload = function(f){
                var result = document.getElementById("img");
                var src = "data:" + file.type + ";base64," + window.btoa(this.result);
                result.innerHTML = '<img src ="'+src+'" width="150" height="150" />';
                // <img id = 'img/>
                // document.getElementById("img").src = src;
            }
            console.log('file',file);
        };
    </script>
```

```
@app.route('/add', methods=['POST'])
def addProduct():
    form = request.form
    name = form.get('name')
    price = form.get('price')
    t = form.get('type')
    st = form.get('stocks')
    image_file = request.files['photo']
    # imgdata = base64.b64decode(image.split(",")[1])
    imgdata_b = str(base64.b64encode(image_file.read()))
    # Because show the picture will have little bit of different with the imgdata_b so
need delete some chatacters
    imgdata = ''+imgdata_b[2:-1]
```

```
    re = ServerAddProduct(name, price, t,imgdata,st)
    if not name:
        flash("please input name !")
        return render_template("addProduct.html")
    elif not price:
        flash("please input price !")
        return render_template("addProduct.html")
    elif not t:
        flash("please input type !")
        return render_template("addProduct.html")
    elif not st:
        flash("please input stocks !")
        return render_template("addProduct.html")
    elif not image_file:
        flash("please input photo !")
        return render_template("addProduct.html")
    elif(re == 'seccess'):
        return redirect('index')
```

New knowledge Base64

Base64 is often used to represent, transfer, and store some binary data, including MIME emails and some complex XML data, where text data is normally processed.

Base64 is a representation of binary data based on 64 printable characters. Since {\displaystyle 2^{6}=64} {\displaystyle 2^{6}=64}, each 6 bits is a unit corresponding to a printable character. Three bytes have 24 bits, corresponding to 4 Base64 units, ie 3 bytes can be represented by 4 printable characters. It can be used as a transmission encoding for email. The printable characters in Base64 include the letters A-Z, a-z, and numbers 0-9, which have a total of 62 characters, and the two printable symbols differ in different systems. Some other encoding methods such as uuencode, and later versions of BinHex use different 64 character sets to represent 6 binary digits, but are not called Base64.

For example:

Image change to base 64

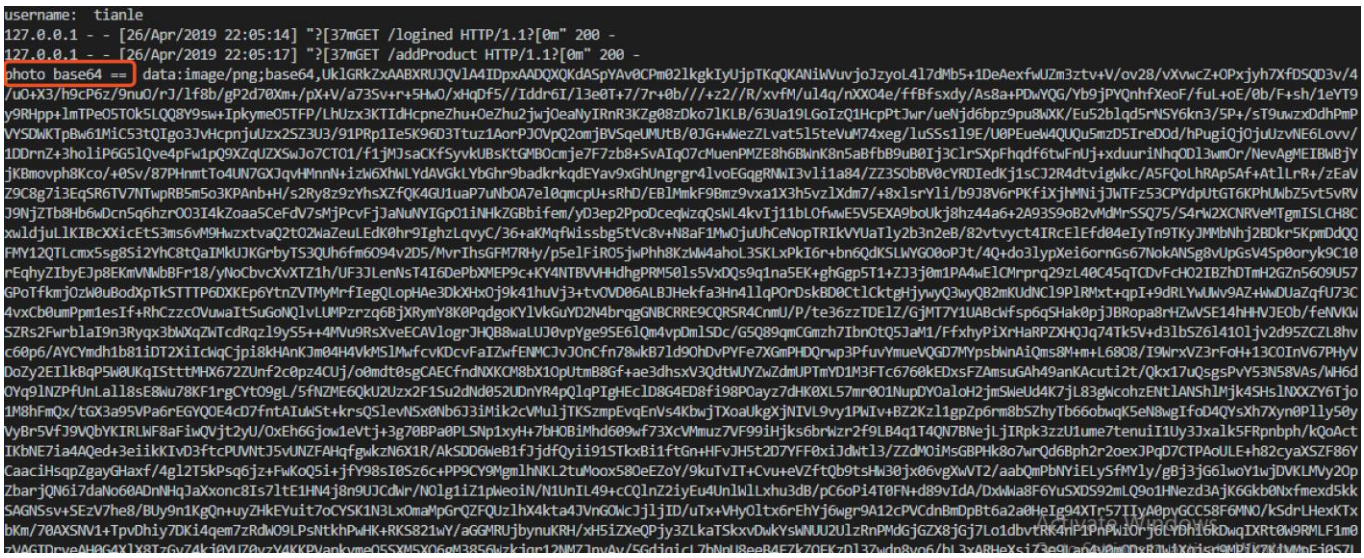
But firstly you need install base64 package from python.

```
#image 转 base64
import base64
with open("C:\\Users\\wonai\\Desktop\\1.jpg","rb") as f:#转为二进制格式
    base64_data = base64.b64encode(f.read())#使用 base64 进行加密
    print(base64_data)
    file=open('1.txt','wt')#写成文本格式
    file.write(base64_data)
    file.close()
```

Base64 change to image

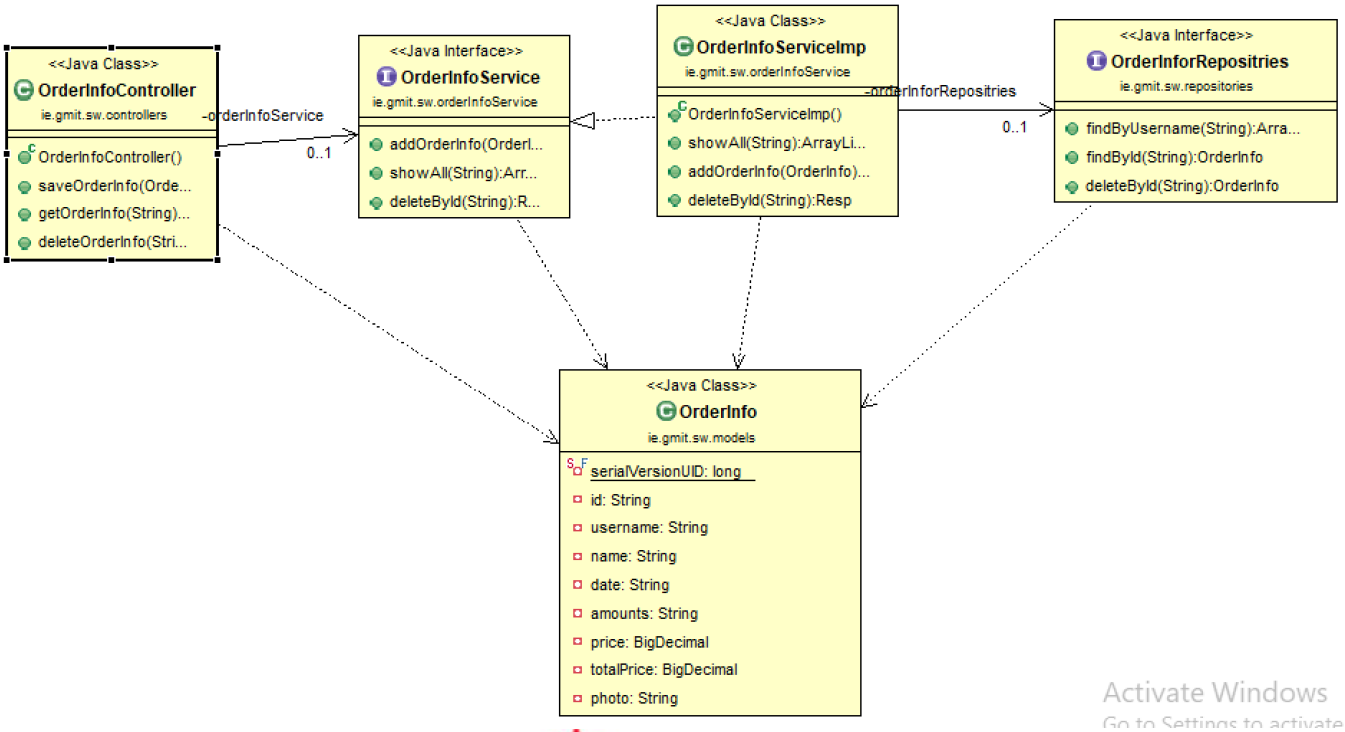
```
import os,base64

with open("C:\\Users\\wonai\\Desktop\\1.txt","r") as f:
    imgdata = base64.b64decode(f.read())
    file = open('1.jpg','wb')
    file.write(imgdata)
    file.close()
```

Get the base64 picture, but data too big , so just shortcut a part of the picture's base64



Order Information Storage Design

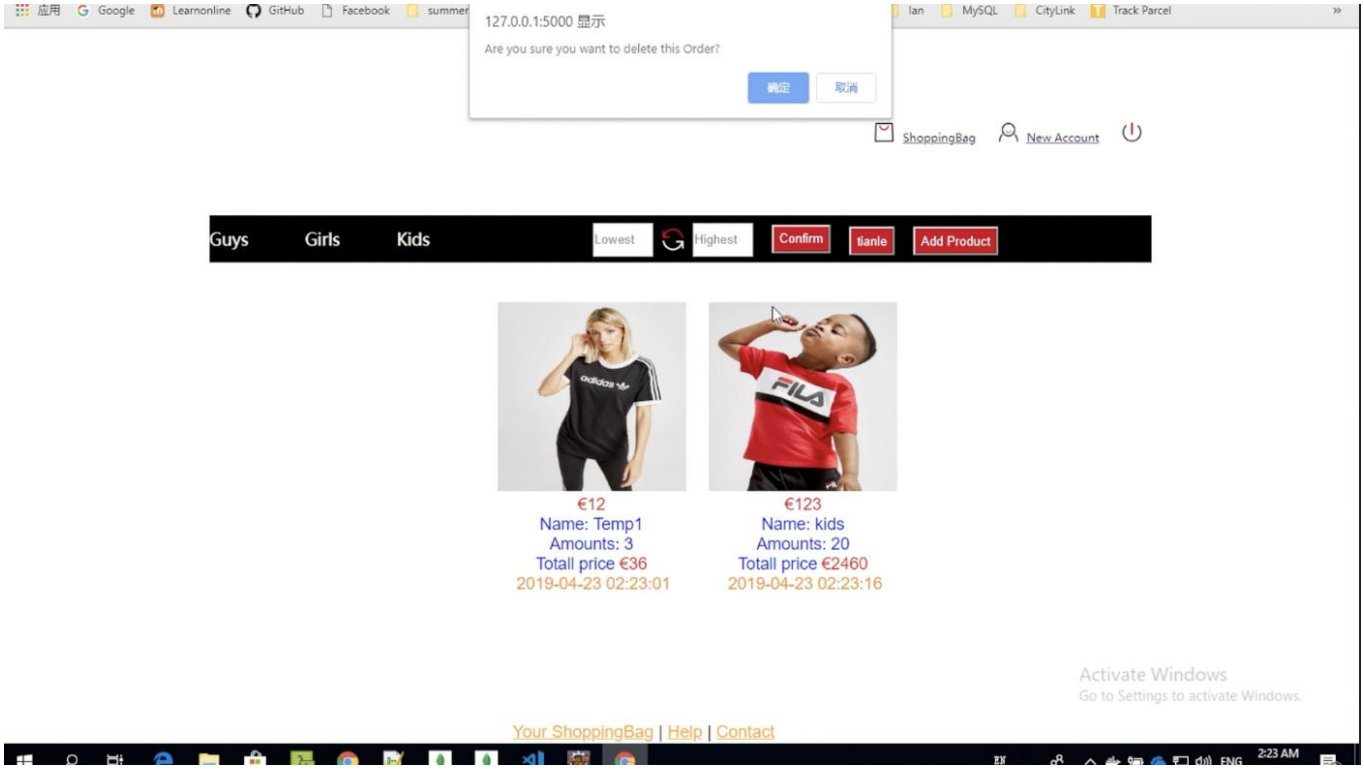
This is for Product design, if user login successful, then user can ads the product what they want to sale, and the product information will storage in MongoDB, the storage function like user register function, except the image storage, I explained on the above. The relationship in each class diagram below.
Get the username from MySQL and get the product information from mongodb, get time from python package. And finally, storage in the MongoDB



Order information will storage in MongoDB and user can delete what the user want to delete shopping history.

```
@app.route('/orderCreate')
def goOrderPage():
    global user
    if(user == ''):
        return redirect('login')
    username = session['username']
    url = 'http://127.0.0.1:8080/getOrderInfo?username='+username
    r = requests.get(url)
    Parsed_json = json.loads(r.text)
    return render_template("orderCreate.html", username = session['username'], items =
Parsed_json)

def delete(id):
    url = 'http://127.0.0.1:8080/deleteOrderInfo?id='+id
    r = requests.post(url)
    print('888888',r.text)
    return json.loads(r.text)

@app.route('/deleteOrderInfo',methods = ['POST'])
def deleteOrderInfo():
    global orderId
    orderId = request.args.get('id')
    print("order id: ",orderId)
    re = delete(orderId)
    if (re == 'delete succssful'):
        print("@@")
        return redirect('orderCreate')
```

There are 2 function one is for user create the order information that means save user's buy product information, another one is delete function, that is user go to there order page can choose order to delete.



Chapter 6

System Evaluation

6.1 Milestones

M1. Research technologies what will suit for this project.

M2. Develop the login and register functions (connect with MySQL).

M3. The images and other information’s are storage in the MongoDB and can be read.

M4. Make the web pages more better.

M5. Test and Run the application.

M6. Submission and Dissertation.

6.2 Future Development and Optimization

- Firstly, I will continue this project in the future, because I am very interested to the online-shopping and in the future, I will fix some problem about the project.

- Seller, customer and Admin will be appeared in the future, Seller have the power to sale something, user only have buy function, and the last one Admin will fix some problem to manage this shopping System.

- Seller can add more pictures to show the picture and also can upload small video so that user can more clear to know the product.

- Add reset password and forgot password function that can add a email address, through the send a link to email to change the password.

- In the era of mobile intelligence, mobile phones have become the most important online tools for people, and online shopping for mobile phones has become an indispensable part of modern people's daily lives. When the mobile phone mall consumption becomes a fashion, we need to understand some of the concept knowledge and basic functions of the mobile online store. So, I will
- add a mobile view for people convenient to buy the items.

Thank for your attention,
Tianle Shu

Chapter 7

Bibliography/References:

[1] Online-Sopping Forecast: https://www.cifnews.com/article/40276
[2] Visual Studio Code: https://zh.wikipedia.org/zh-hans/Visual_Studio_Code
[3] Eclipse: https://baike.baidu.com/item/Eclipse/61703

[4] Database: https://moon-walker.iteye.com/blog/2389231
[5] Flask Tutorial: https://pythonspot.com/flask-web-app-with-python/
[6] Flask Tutorial: https://zhuanlan.zhihu.com/p/34143669/
[7] MySQL: http://www.runoob.com/mysql/mysql-tutorial.html
[8] MongoDB: http://www.runoob.com/mongodb/mongodb-intro.html
[9] MongoDB: https://winterlei27.iteye.com/blog/2405225/
[10] Docker: https://www.docker.com/
[11] Docker: https://zh.wikipedia.org/wiki/Docker/
[12] Docker container: https://yeasy.gitbooks.io/docker_practice/introduction/what.html
[13] Base64 : https://en.m.wikipedia.org/wiki/Base64
[14] Solution for Base64 problem: https://stackoverflow.com/questions/8499633/how-to-display-base64-images-in-html
[15] Base64 change to photo in javascript : https://stackoverflow.com/questions/6150289/how-to-convert-image-into-base64-string-using-javascript