MY SHOPPU

Capstone Design

Youness Ayyada

Supervised by:
Pr. Omar Iraqui Houssaini

Spring 2020
Capstone Report

**Student Statement:**

I Youness Ayyada, assert that I have applied ethics to the design process and in the selection of the final proposed design. Moreover, I have held the safety of the public to be paramount and have addressed this in the presented design wherever applicable.

Youness Ayyada

Approved by the Supervisor(s)

Pr. Omar Iraqui Houssaini
Acknowledgements

I would like to seize this opportunity to acknowledge my supervisor Pr. Iraqi Houssaini, Omar for his help throughout the design process and for sharing his experience with the technologies I used to build my application. I would also like to thank all my professors, who taught me the skills and tools of this capstone, for allowing me to make it this far. I offer my gratitude to AUI for providing an adequate environment for my studies. Finally, a special thanks to my family who regardless of any hardship, continued to encourage me and support me.
# Table Of Contents

Abstract ........................................................................................................................................... 7

I. Introduction ................................................................................................................................... 8

II. STEEPLE ANALYSIS .................................................................................................................. 9

  1. Social Impact .......................................................................................................................... 9
  2. Technological Impact ............................................................................................................. 9
  3. Environmental Impact .......................................................................................................... 9
  4. Economic Impact ................................................................................................................. 9
  5. Political Impact ................................................................................................................... 9
  6. Legal Impact ........................................................................................................................ 9
  7. Ethical Impact ........................................................................................................................ 9

III. Feasibility Study ....................................................................................................................... 10

IV. Requirement specifications ...................................................................................................... 11

  1. Functional requirements ....................................................................................................... 11
    a. User .................................................................................................................................. 11
    b. Manager .......................................................................................................................... 11
  2. Use Case Diagram ................................................................................................................. 12
  3. Non-Functional Requirement ............................................................................................... 12
    a. Security ............................................................................................................................ 12
    b. Extensibility ..................................................................................................................... 13
    c. Performance ...................................................................................................................... 13
    d. Scalability ........................................................................................................................ 13

V. Design .......................................................................................................................................... 13

  1. Class Diagram ....................................................................................................................... 13
  2. Technology Enablers ............................................................................................................. 15
    a. Languages and frameworks ............................................................................................. 15
    b. Angular ............................................................................................................................. 16
    c. Spring Boot ....................................................................................................................... 16
    d. Storage technologies ......................................................................................................... 17
3. Software Architecture ................................................................. 19
4. Sequence Diagram ........................................................................ 20

VI. Implementation ........................................................................... 21
1. User side ..................................................................................... 21
   a. Authentication ........................................................................ 21
   b. Search .................................................................................... 25
   c. Order ....................................................................................... 26
   d. Review .................................................................................... 27
2. Manager side ................................................................................ 27
   a. Authentication ........................................................................ 27
   b. Manage Orders ....................................................................... 27
   c. Manage Products .................................................................... 28
3. Cashing Feature ........................................................................... 28

VII. Conclusion ................................................................................ 29
VIII. References ................................................................................ 30
# Table of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Use case diagram</td>
<td>12</td>
</tr>
<tr>
<td>2</td>
<td>Class diagram</td>
<td>14</td>
</tr>
<tr>
<td>3</td>
<td>Docker Toolbox menu</td>
<td>17</td>
</tr>
<tr>
<td>4</td>
<td>Docker containers</td>
<td>18</td>
</tr>
<tr>
<td>5</td>
<td>Back-end Code Structure</td>
<td>18</td>
</tr>
<tr>
<td>6</td>
<td>System Architecture</td>
<td>19</td>
</tr>
<tr>
<td>7</td>
<td>Sequence Diagram</td>
<td>20</td>
</tr>
<tr>
<td>8</td>
<td>MyShoppu Main Page</td>
<td>21</td>
</tr>
<tr>
<td>9</td>
<td>User Sign In Form</td>
<td>22</td>
</tr>
<tr>
<td>10</td>
<td>JWT Generating Code</td>
<td>23</td>
</tr>
<tr>
<td>11</td>
<td>User Sign Up Form</td>
<td>24</td>
</tr>
<tr>
<td>12</td>
<td>Bcrypt Encoder</td>
<td>24</td>
</tr>
<tr>
<td>13</td>
<td>Search by Products Categories</td>
<td>25</td>
</tr>
<tr>
<td>14</td>
<td>Search by Shop Names</td>
<td>26</td>
</tr>
<tr>
<td>15</td>
<td>Shopping Cart</td>
<td>27</td>
</tr>
<tr>
<td>16</td>
<td>Shopping Cart Service and Caching</td>
<td>28</td>
</tr>
</tbody>
</table>
Abstract
The purpose of this capstone is to implement a web application to both help small businesses sell their product online and provide a one uniform platform where a user can buy different products of different companies. Shop managers, along with other functionalities, can manage orders and update their product page. Other users can search products either through the application’s main platform or through the businesses’ shop page.

This report illustrates the conception part of the application including, the requirement specifications, STEEPLE analysis, design and architecture along with an explanation of the technologies used and results.
I. **Introduction**

My Shoppu is a web application platform designed to facilitate businesses to sell their products online while having an individual that is partially independent from the main platform. This should be achieved by providing subdomains for each user and keep each shop’s information well separated. The shop’s’ manager can keep track of the users who bought or ordered their products as well as view their reviews and process orders. This will allow shops to maintain their reputation growing through online sales and grow economically using a personal website, without hiring any technical expert.

The application will also help users by providing a general platform grouping all different shops and products to save their time in their research. It will be achieved by implementing a search functionality with options to filter and sort results according to shops and product categories.

My Shoppu will be accessible through most popular web browsers and compatible with computers, tablets and smartphones.
II. STEEPLE ANALYSIS

1. Social Impact
The aim of this project is to help small businesses to have their own online shop page to sell their products or services, while providing to buyers a platform which groups different shops. This will allow users have a large choice of products, while being able to give feedback for the interest of shops to evolve.

2. Technological Impact
This application is implemented thanks to recent and advanced web technologies and tools. These are open source enabling an easy access and enough learning material to better use them. By using them, a friendly and attractive application is possible to implement.

3. Environmental Impact
Using the application will reduce the paperwork generated to keep transaction records or advertising the shop. In addition, this will allow users to search through a list of shops online instead of moving to different locations.

4. Economic Impact
This application will allow shops to expend their range of clients through online transactions and its virtual availability. Moreover, it will be self-maintained, thanks to the revenue which comes from some special functionalities which shop managers can purchase. As a starting point, the application will be free for both managers and users. A small percentage is however deducted from the sales made by shops for each item. Other optional features can also be added later to managers like visualizing sales per time periods and user preferences, which will not be free. The revenues generated will be essential for the application to be maintained.

5. Political Impact
There is not any political impact made by the application.

6. Legal Impact
The application will be using only free and open source technologies. Therefore, there will be no illegal tool usage to implement this application.

7. Ethical Impact
All vulnerabilities in treating user’s information shall be avoided, sensitive data shall be encrypted then stored. Transactions shall also be well secured. These preventions will ensure the ethical use of the application.

III. Feasibility Study

The feasibility study represents the different reasons this Capstone project can succeed by evaluating three categories, technology, economy, and time. The project consists of a full stack web application, made for businesses or individuals willing to sell their products online and other types users which would by these products.

As the project is heavily technological, I will discuss this as a first aspect. Technologically speaking, since the project is a web full stack application, it consists of a front-end interface and back-end server. For the interface part, I used Angular as framework along HTML for placing contents, CSS for styling, and Typescript. The back end was developed using another framework called Spring based on Java language. It will be used in order to handle the authentication, privacy, and database. These frameworks were chosen based on their reliability and popularity over the last years in the workplace. This will ensure having enough online sources and easy access to learn more about these tools. On the other hand, being previously exposed to both technologies in my last courses will allow a better refinement and diversity in terms of their features use.

Second, we will evaluate the time frame of the project to ensure its completion within the allowed time period. The next steps after this document are the specifications and requirements, design, and finally the implementation phase. For both the requirements and design phase, thanks to many classes I took, I will be able to finish them and refine them within the time period allocated previously, respectively one and two weeks. Whereas, the last and most important longest phase will need further research and learning, however, as stated in the technology feasibility, this process will be less complex. Therefore, four weeks will be enough for this phase and additional time will be used to enhance and add more features, beside one week for testing.

Finally, the economic feasibility will help us determine the cost of the project completion and get an idea about its revenue. The technologies used are available for free additionally to the learning
materials. The application will initially have two types of users, one which can register for free and allowed to purchase available products, whereas other users which will be selling products need to pay per sold item.

IV. Requirement specifications

1. Functional requirements

   a. User

   - User shall be able to sign up or sign in.
   - Logged user shall be able to add, save, modify, or delete his personal information.
   - User shall be able to search products by name.
   - User shall be able to sort products by categories or shop name.
   - User shall be able to view the provider’s information about a specific product.
   - User shall be able to grade and post a review about a provider’s products.
   - User shall be able to view other users’ grades and review on a specific product.
   - User shall be able to add, modify, or delete products.
   - A logged user shall be able to order his selected products.

   b. Manager

   - A manager shall be able to sign in or sign in.
   - A manager shall be able to create a shop.
   - A manager shall be able to add, modify, delete a product or its information.
   - A manager shall be able to proceed orders.
2. Use Case Diagram

![Use Case Diagram](image)

*Figure 1 Use case diagram*

The use case diagram illustrates the possible types of users that can register in the application. Each of the users is restricted to certain functionalities. For example, regular users as well as manager can sign up. A manager can manage his shop, its product, information, and orders. Whereas, users can access their orders, search through the list of available shops, check their details and product list.

3. Non-Functional Requirement

   a. Security

      i. Confidentiality

         user information shall be kept confidential

      ii. Integrity

         User operations made by the user shall be well preserved and integrated

      iii. Availability

         The application shall always be available
All single points of failures must be avoided

b. Extensibility

New features can be easily integrated to the system

c. Performance

The application shall not exceed a response time of 2 seconds

d. Scalability

The increasing number of users should not affect performance

V. Design

1. Class Diagram
The class diagram represents the necessary entities according to the requirement phase and shows their relationships [7]. It contains ten entities whose relationships fall into two types:

- Bi-directional association: A sign is displayed on both sides of the relationship expressing that both entities are aware of each other. The relationship on the diagram between User and UserType illustrates a bi-directional one.
- Composition aggregation: This relationship can be expressed on one end of the relationship and means that one instance of the
object is required for the creation of the second. The symbol on User on its link with Reviews describes that an instance of review cannot be created without a user.

The abstract entity is linked to every entity since it contains common elements that are necessary to each entity and acts as a master class. Hence, each entity is given an ID, a Boolean which triggers the visibility of the instance, date of last time modified, the user that made last modification, date of creation and user who created it.

2. Technology Enablers
   a. Languages and frameworks

The technology enablers part was essential to decide the set of technologies and tools used for the implementation part of the project. Different criteria had been questioned in order to make such decision, the familiarity, availability, code organization, etc. They will also ensure a better quality of the application’s result.

Hypertext markup language (HTML) will be used in this application as an essential part for displaying and organizing the contents on the web [1]. A cascading style sheet (CSS) is also necessary to further customize the user interface [2]. It can change colors, font, formatting, and other aspects of the text generated by HTML. Combined with these two tools I used Typescript language which adds another layer to our web output. It is considered as JavaScript’s superscript, allowing us to have better control over our front-end code which enables different action triggering and makes the page more dynamic and reactive. I used Angular in order to help me encapsulate these languages into a decent looking web application view.

For the backend I will be using Spring Boot, a JAVA language framework as it is one of the most used technology for enterprise applications.
b. Angular

Angular is an open-source framework used to build single page apps [8]. It was created by a Google team with the help of individuals from AngularJS user's community. The application is considered as a new one although it is based on AngularJS, an older Google frontend framework. Angular has released nine versions over the last three years. The single app feature releases the browser from reloading an entire webpage by keeping the static elements and changing only specific components of the page. This technique additionally affects users in a positive way by avoiding the several page transitions and reload. The framework comes along with multiple libraries and annotations that facilitate coding in way to reduce considerably repeated and have a better code structure.

Angular is composed of modules and components. Modules are the base of the application and groups sets of related code which take forms as components. At least one module is necessary for the application to work, considered the root module and responsible for the first view displayed by the application. Angular modules are defined with an annotation @NgModule in the beginning of the file.

An angular component defines the classes of the application. A minimum of one component is needed for each application and considered as the root component, commonly called appcomponent. It is composed mainly of three files, one HTML, CSS, and Typescript file.

c. Spring Boot

Spring Boot is an open source back-end JAVA based framework, used to create micro Services and a stand-alone ready to run applications [9]. It was created by Pivotal Team. It is easy to scale, deploy and comes with a standard configuration that can be customized. For example, the Java file called DemoApplication on the figure contains the main function and is annotated with @SpringBootApplication, which includes an auto-configuration.
d. Storage technologies

I used Docker tools software for my application, which is an installer allowing to quickly create a Docker environment [10]. Docker is a virtual machine software which allows users to create separate containers with an available list of free software services namely, Redis and PostgreSQL. Redis is an open-source database mainly used for caching since it is an in-memory data structure store [11]. It was used to enable caching of frequently accessed data allowing a smoother flow of http request, better responsiveness, and less load on the server.
Figure 4 Docker containers

Figure 5 Back-end Code Structure
The figure shows the backend code structure of Spring Boot. Following the MVC architecture it separates between the business logic and other functionalities like security. The controller is responsible for the request received from and sent to the view. The model part is represented by the data file containing the business logic, entities and direct transfer objects (DTO). Services handle complicated functions or user requests which are injected into controllers using dependency injection. The security file contains the JWT functions needed to verify users and generate for new sessions.

3. Software Architecture

The software architecture diagram describes the different parts of the application, and how they communicate to each other. Myshoppu follows a model-view-controller (MVC) architecture. This type of architecture brings number of advantages mainly a clear separation of functional and non-functional requirements. The view is generated on the browser using Angular, which communicates with the backend. The controller takes care of the requests sent by the frontend through a Rest over http manner. The requests are then redirected to the corresponding service in case of complicated functionality or directly to the corresponding repository.

*Figure 6 System Architecture*
4. Sequence Diagram

The sequence diagram describes the sequence of instructions a user can make using the application [6]. It represents also the possible error actions and their results. The figure above shows the sequence actions for a user to proceed an order. First, user shall search shops from the list displayed. Then, after selecting the desired shop a list of their products is displayed. A get request is sent to the server to retrieve corresponding products and send them back. The user can then add any product on the list. When the list of desired products has been added, the user can
initiate the order. The server will either respond with a successful order notification if no error have been detected or an unsuccessful one in case the user is not logged in and send the log in page.

VI. Implementation

Figure 8 MyShoppu Main Page

1. User side
   a. Authentication
Nonregistered users can perform some actions but have to create an account in order to submit an order. For keeping track of the different users and their status I used JSON web tokens (JWT) which is compact code of JSON files containing user session information, the algorithm used to encrypt the token and the server’s digital signature. [5]. The token is assembled with the three distinct parts namely, a header, a payload, and a signature, using a dot “.” and represented as follow: “header.payload.signature”. The header contains the algorithm used to encrypt our token and the type of token. Payload has a personalized list of information generally user’s relevant information as the ID, name, expiration date of the token. Finally, the third part of the token which is a verification which checks if someone modified the token before sending it to the server. It is created using the base64 encoded header, a base64 encoded payload and a secret key, all combined using a comma and then using the algorithm used in the header encodes the signature. The JWT is first generated when a user first access the application and is sent back to
the user. After that, for every request sent by the user, the token must be included in order to verify the identity of the user or give authorization to the user.

```java
public static String generateToken(Authentication authentication) {
    User user = (User) authentication.getPrincipal();
    List<String> authorities = user.getAuthorities().stream()
        .map(GrantedAuthority::getAuthority)
        .collect(Collectors.toList());
    long now = System.currentTimeMillis();
    long expirationTime = now + 1000 * 60 * 60 * 24;

    return Jwts.builder().setSubject(user.getUsername()).setHeaderParam("typ", "JWT")
        .setIssuedAt(new Date(now))
        .setExpiration(new Date(expirationTime))
        .claim("roles", authorities)
        .signWith(Keys.hmacShaKeyFor(SIGNING_KEY.getBytes()),
        SignatureAlgorithm.HS512)
        .compact();
}
```

*Figure 10*  JWT Generating Code
The registration part requires from the user their username, email address, and password. The password is first hashed using a bcrypt algorithm and then stored in the database in order to reinforce security using a library as shown in the next figure.

```
@Bean
public PasswordEncoder passwordEncoder() {
    return new BCryptPasswordEncoder();
}
```

*Figure 12 Bcrypt Encoder*
b. Search

Users can search for products by typing the corresponding name in the search field or access to the list of categories by clicking on the category button (figure 13). They can also search for products through the list of shops by clicking on the shops button (figure 14).

Figure 13 Search by Products Categories
c. **Order**

The user then can select products and add the products desired to their shopping cart. The price is dynamically updated in the shopping cart view. Users can adjust the quantity from the shopping cart at any time. When the user is satisfied with the order list, they can proceed to order if they are logged in. The order is then sent to the shop’s manager.
d. Review

By clicking on a product, a review section is displayed under the product information. Users can there rate the product and leave a comment. Other users’ reviews are also displayed.

2. Manager side

a. Authentication

A different registration is provided for users who seek to have a shop. To register they will need, additionally to the information required from regular users, they need to give a shop name. Similarly, to a regular user, JWT are used to keep track of managers’ session.

b. Manage Orders

Managers can view the list of orders and their details by selecting orders on the menu.
c. Manage Products

By selecting the product tab, managers can add new products to their list. They need to provide its name, price, a description, and select a category from the list. The user then can see his product on the list displayed and remove or disable if necessary.

3. Cashing Feature

![Shopping Cart Service and Caching](image)

Thanks to our backend framework it is possible to easily access the current user information from our Redis cache and simply by using an annotation. @CachePut is used here to enable updating the user shopping cart as soon as the function is called.
VII. Conclusion

This capstone was a fruitful learning experience despite the different challenges encountered. I used amazing tools to build this project, which are also famous in the working filed. Working on a Spring Boot framework made me change my opinion about JAVA positively and made me believe that it will be more popular in the near future thanks to its incredible features and flexibility.

The main challenges were during the implementation phase which was also related to a change of environment. Although some of the technologies used were previously encountered, I have yet not mastered completely their features. Angular, for example, was still a black box to me before starting coding, but now I can better understand the logic behind its code and when to use it. Another challenge was due to the pandemic situation which forced students to leave the university and its generous and productive environment. I had some technical limitations beside the pressure of confinement, however, it allowed me to improve my adapting skills, which will help me a lot in the future.

The application is yet to be finished but given the circumstances and time limit, it was only possible to finish a ready back end and partial front-end server. For my future work on this project, I would add more features related to shops information such as locations which will require the use of Google map web service. Another objective I would have liked to achieve was to add notifications for new products and the ability to follow specific shops. Making the application progressive would also enhance it considerably allowing an off-line access to MyShoppu.

I am glad I had such a unique experience and I truly believe that it has greatly helped me create a path to my long-term goals.
VIII. References


