

**VIETNAM NATIONAL UNIVERSITY - HO CHI MINH CITY  
INTERNATIONAL UNIVERSITY**

**SCHOOL OF COMPUTER SCIENCE AND ENGINEERING**



**BEVERAGE WEBSITE**

**ADVISOR:**

Assoc. Prof. Phạm Quốc Sơn Lâm

**TEAM MEMBER:**

Lê Thanh Phương Nam – ITITWE19025

Phạm Công Tuấn - ITITIU19060

Phạm Đinh Hoàng Việt - ITITIU19064

Đặng Đình Khang - ITITIU18055

**2022 - Ho Chi Minh City, Viet Nam**

**Contribution**

Name	Phương Nam	Công Tuấn	Hoàng Việt	Đình Khang
Percent	30%	30%	25%	15%

**GITHUB Link : <https://github.com/TomPham204/Bartender-Web>**



# TABLE OF CONTENTS

<b>TABLE OF CONTENTS</b>	<b>03</b>
<b>List of Tables</b>	<b>05</b>
<b>List of Figures</b>	<b>06</b>
<b>Chapter 1: Project Description</b>	<b>07</b>
<b>1.1. Project Overview</b>	<b>07</b>
<b>1.2. The Purpose of the Project</b>	<b>07</b>
<b>1.2.1. Abstract</b>	<b>07</b>
<b>1.2.2. Project Goals</b>	<b>07</b>
<b>1.3. Scope of Work</b>	<b>07</b>
<b>1.4. Project Organization</b>	<b>08</b>
<b>1.5. Resource Requirements</b>	<b>08</b>
<b>1.5.1. Hardware</b>	<b>08</b>
<b>1.5.2. Software</b>	<b>09</b>
<b>1.5.3. Human</b>	<b>10</b>
<b>1.6. Technology</b>	<b>12</b>
<b>1.6.1. PHP</b>	<b>12</b>
<b>1.6.2. MySQL</b>	<b>12</b>
<b>1.7. Schedule</b>	<b>14</b>
<b>1.7.1. Gantt Chart</b>	<b>14</b>
<b>1.7.2. Work Breakdown Structure</b>	<b>15</b>
<b>1.7.3. Schedule and milestone</b>	<b>16</b>
<b>1.7.4. Process Development</b>	<b>21</b>
<b>1.8. Risk Management</b>	<b>22</b>
<b>1.9. Security Aspect</b>	<b>22</b>
<b>1.10. Naming Conventions and Definitions</b>	<b>22</b>
<b>1.11. Conclusion</b>	<b>23</b>
<b>Chapter 2: Requirements</b>	<b>23</b>
<b>2.1. Product Use Cases</b>	<b>24</b>
<b>2.2. Functional &amp; Non-functional Requirements</b>	<b>30</b>

<b>2.3. Data Requirements</b>	<b>31</b>
<b>2.4. Performance Requirements</b>	<b>31</b>
<b>2.6. Security Requirements</b>	<b>32</b>
<b>Chapter 3: Design</b>	<b>33</b>
<b>3.1. System Design</b>	<b>33</b>
<b>3.2. Database Structure</b>	<b>33</b>
<b>3.3. Entity–relationship Diagram</b>	<b>34</b>
<b>3.4. Activity Diagram</b>	<b>34</b>
<b>3.5. Class Diagram</b>	<b>36</b>
<b>Chapter 4: Implementation</b>	<b>37</b>
<b>4.1. Login function</b>	<b>37</b>
<b>4.2. Register function</b>	<b>38</b>
<b>4.3. User function</b>	<b>39</b>
<b>4.4. Admin function</b>	<b>41</b>
<b>Chapter 5: Test Plans</b>	<b>42</b>
<b>5.1. Feature to be tested</b>	<b>42</b>
<b>5.1.1. In scope</b>	<b>42</b>
<b>5.1.2. Out of scope</b>	<b>43</b>
<b>5.2. Pass/Fail Criteria</b>	<b>43</b>
<b>5.3. Approach</b>	<b>44</b>
<b>5.3.1. Process of testing</b>	<b>44</b>
<b>5.3.2. Testing level</b>	<b>44</b>
<b>5.3.3. Roles and responsibilities</b>	<b>45</b>
<b>5.3.4. Types of testing</b>	<b>45</b>
<b>5.4. Test materials</b>	<b>45</b>
<b>5.5. Test cases</b>	<b>46</b>
<b>5.6. Test Plans Schedule</b>	<b>57</b>
<b>Chapter 6: Glossary</b>	<b>58</b>
<b>References</b>	<b>58</b>

## List of Tables

<b>Table 01. Project Organization</b>	<b>08</b>
<b>Table 02. Hardware resource</b>	<b>08</b>
<b>Table 03. Software resource</b>	<b>09</b>
<b>Table 04. Human Resource</b>	<b>10</b>
<b>Table 05. Schedule and milestone</b>	<b>16</b>
<b>Table 06. Risk Management</b>	<b>21</b>
<b>Table 07. Naming Convention and Definitions</b>	<b>22</b>
<b>Table 08. Use case list</b>	<b>24</b>
<b>Table 09. Use case 01</b>	<b>25</b>
<b>Table 10. Use case 22</b>	<b>26</b>
<b>Table 11. Use case 03</b>	<b>27</b>
<b>Table 12. Use case 04</b>	<b>28</b>
<b>Table 13. Use case 05</b>	<b>29</b>
<b>Table 14. Functional and Non-functional</b>	<b>30</b>
<b>Table 15. Feature - in scope</b>	<b>42</b>
<b>Table 16. Pass/Fail criteria</b>	<b>43</b>
<b>Table 17. Role and responsibilities</b>	<b>45</b>
<b>Table 18. Function testing</b>	<b>45</b>
<b>Table 19. Security Testing</b>	<b>45</b>
<b>Table 20. Test materials</b>	<b>45</b>
<b>Table 21. Test case 01</b>	<b>46</b>
<b>Table 22. Test case 02</b>	<b>47</b>
<b>Table 23. Test case 03</b>	<b>48</b>
<b>Table 24. Test case 04</b>	<b>49</b>
<b>Table 25. Test case 05</b>	<b>50</b>
<b>Table 26. Test case 06</b>	<b>51</b>
<b>Table 27. Test case 07</b>	<b>52</b>
<b>Table 28. Test case 08</b>	<b>53</b>
<b>Table 29. Test case 09</b>	<b>54</b>
<b>Table 30. Test case 10</b>	<b>55</b>
<b>Table 31. Test case 11</b>	<b>56</b>

## List of Figures

<b>Figure 01. PHP</b>	<b>12</b>
<b>Figure 02. MySQL</b>	<b>12</b>
<b>Figure 03. Javascript</b>	<b>13</b>
<b>Figure 04. Gantt Chart</b>	<b>14</b>
<b>Figure 05. Work breakdown Structure</b>	<b>15</b>
<b>Figure 06. Use Case Diagram</b>	<b>24</b>
<b>Figure 07. Data requirements</b>	<b>31</b>
<b>Figure 08. System Design</b>	<b>33</b>
<b>Figure 09. Database Structure</b>	<b>33</b>
<b>Figure 10. ERD</b>	<b>34</b>
<b>Figure 11. Activity diagram - Search for recipe</b>	<b>34</b>
<b>Figure 12. Activity diagram - New user register</b>	<b>35</b>
<b>Figure 13. Activity diagram - Login / Logout</b>	<b>35</b>
<b>Figure 14. Class Diagram</b>	<b>36</b>
<b>Figure 15. Login interface</b>	<b>37</b>
<b>Figure 16. Wrong login information</b>	<b>37</b>
<b>Figure 17. User register</b>	<b>38</b>
<b>Figure 18. Wrong email format</b>	<b>38</b>
<b>Figure 19. About us Page</b>	<b>39</b>
<b>Figure 20. Mix ingredients</b>	<b>39</b>
<b>Figure 21. Wrong ingredient amount 1</b>	<b>40</b>
<b>Figure 22. Wrong ingredient amount 2</b>	<b>40</b>
<b>Figure 23. Admin page</b>	<b>41</b>
<b>Figure 24. Test plan</b>	<b>57</b>

# **Chapter 1: Project Description**

## **1.1. Project Overview**

Due to the needs of shopping and researching some specific items, most online shopping platforms cannot guarantee the quality, variety, and specific knowledge about some special items.

In this project, the item we want to talk about is Beverages. Here, together, we create a website to support users to create and sell new and healthy drink recipes for users. Besides that, we also have some useful functions such as explaining the drink's use and many promising updates.

## **1.2. The Purpose of the Project**

### **1.2.1. Abstract**

In the 4.0 era, with the strong development of technology applied to life as well as every other field around the globe, it has brought many benefits to the economy. Moreover, even during the Covid-19 pandemic, the application of technology to sales, delivery, and home shopping is even more warmly received. However, nowadays, bars and restaurants are only based on available menus, not allowing users to create them by themselves on online platforms. At the same time, the current online buying and selling platforms are still not diverse in terms of drinking water species. Agree, there is still a sale of available drinking water but the expiry date is unknown, and the user's health and raw materials are guaranteed.

### **1.2.2. Project Goals**

With the above problems in mind, our project aims to allow users to create novel drinks from simple and easy-to-find ingredients while still ensuring the health of consumers. At the same time, this project, in addition to suggesting people drinks and selling them those drinks based on the ingredients they choose honestly, ensures quality and health for users.

## **1.3. Scope of Work**

The scope of this project is based on the version of the old "Bartender" game y8 with the beverage feature in the "Genshin Impact" game. With many promising possibilities in the future when it can be developed to a commercial level, the detailed formulation is suitable for the needs of all ages.

After this project, it will be the foundation for us to understand the needs of Web Applications to develop our programming ability and develop better applications in the future. At the same time, it improves teamwork and time management of a project.

## 1.4. Project Organization

After receiving the project, my team members held a meeting on Ms Teams to discuss the topic of the project and plan in detail what they need to prepare for the upcoming time from Technology, Language, Document, and Human,...

After agreeing, we used the ClickUp website to manage the project to remind the timeline for each member and draw a Gantt Chart of the schedule and work for each person. After dividing tasks for each member on each part, the project will begin to proceed immediately according to the timeline with the management of the team leader reminding each member and continuously updating the results through the weekly meetings and daily group chat.

Task	Name
Project Manager	Phạm Đinh Hoàng Việt
Designer	Phạm Công Tuấn
	Đặng Đình Khang
Software Engineer	Phạm Công Tuấn
	Lê Thanh Phương Nam
Tester	Phạm Đinh Hoàng Việt

*Table 01. Project Organization*

## 1.5. Resource Requirements

### 1.5.1. Hardware

Name	Item	Description	Purpose
Hoàng Việt	Laptop Asus ROG Strix G	OS: Windows 10 Pro CPU: Intel Core i5-9300H RAM: 16GB Onboard 2666MHz Storage: 512GB SSD Screen: Full HD 14" IPS	Implement Code

Đinh Khang	MacBook Pro 2020	OS: macOS CPU: Intel Core i5-8230U RAM: 8GB Onboard 2666MHz Storage: 256GB SSD Screen: 13,3" Retina	Implement Code
Công Tuấn	Laptop Dell	OS: Windows 10 Pro CPU: R5 4600U RAM: 16GB ram bus 1333 Storage: 445 GB SSD Screen: Full HD 24"	Implement Code
Phương Nam	Desktop Computer	OS: Windows 11 Enterprise CPU: Intel Core i5-9400F RAM: 8GB 2666 MHz Storage: 445 GB SSD Screen: 27"	Implement Code

*Table 02. Hardware resource*

### 1.5.2. Software

Application	Description	Purpose	Price	Total Cost	Date Needed
Visual Studio Code	An integrated development environment for OS. Debug, syntax highlights, completion of the intelligent code, fragments, code refactoring, and built-in git are just some of the available functions.	Implement code	Free	Free	DONE
MySQL	An open-source relational database management system (RDBMS)	Storing data	Free	Free	DONE
Microsoft Teams	A collaboration application	Meeting, reminding, and storing files	Free	Free	DONE

<b>PHP</b>	A web development oriented general-purpose scripting language.	Implement code	Free	Free	DONE
<b>XAMPP</b>	Easy to install Apache distribution containing PHP. The XAMPP open source package has been set up to be incredibly easy to install and use	Create localhost site, Generate code and database	Free	Free	DONE
<b>ClickUp</b>	Simplify work and get more done. Plan, track, and manage any type of work with project management	Manage Project	Free	Free	DONE
<b>Github</b>	Web for sharing and uploading code	Managing code	Free	Free	DONE
<b>Total Cost:</b>					0

*Table 03. Software resource*

### 1.5.3. Human

Name	Organization	Role	Required skills
Hoàng Việt	International University	<b>Project Manager, Tester</b>	<p><b>Technical skill:</b></p> Basic understanding of OOP, OOAD, programming languages, and database structure. Knowledge of the development process. Familiar with project management methodologies, software Risk management <p><b>Soft skills:</b></p> Leadership Communication skills Problem-solving Time management
Phương Nam	International University	<b>Back-end Developer</b>	<p><b>Technical skill:</b></p> Basic understanding of OOP, OOAD, programming languages, and database structure. Deep knowledge of Back-end Programming Languages & Data Structures and Algorithms

			<p>Familiar with databases, API, and servers Knowledge of HTML, CSS, JS</p> <p><b>Soft skills:</b> Collaboration Communication skills Problem-solving Time management</p>
Công Tuấn	International University	<b>Front-end Developer</b>	<p><b>Technical skill:</b> Basic understanding of OOP, OOAD, programming languages, and database structure. Deep knowledge of Back-end Programming Languages &amp; Data Structures and Algorithms</p> <p>Familiar with databases, API, and servers Knowledge of HTML, CSS, and JS</p> <p><b>Soft skills:</b> Collaboration Communication skills Problem-solving Time management</p>
Đình Khang	International University	<b>Analysis Design</b>	<p><b>Technical skill:</b> Basic understanding of OOP, OOAD, programming languages. Knowledge of prototyping, mockup, and wireframing.</p> <p>Proficient in using visual design software (Photoshop, AI, XD)</p> <p><b>Soft skills:</b> Collaboration, Communication skills Creativity Time management</p>

**Table 04. Human Resource**

## 1.6. Technology

### 1.6.1. PHP



*Figure 01. PHP*

PHP is a web development-oriented general-purpose scripting language. A PHP interpreter, which can be implemented as a module, a daemon, or a Common Gateway Interface (CGI) executable, is commonly used to process PHP code on a web server. The outcome of the interpreted and executed PHP code – which might be any type of data, such as produced HTML or binary image data – would make up the entirety or portion of an HTTP response on a web server. There are a variety of web template systems, online content management systems, and web frameworks that can be used to coordinate or simplify the creation of that response. PHP can also be used for a variety of programming tasks not related to the web, such as standalone graphical apps and robotic drone control.

### 1.6.2. MySQL

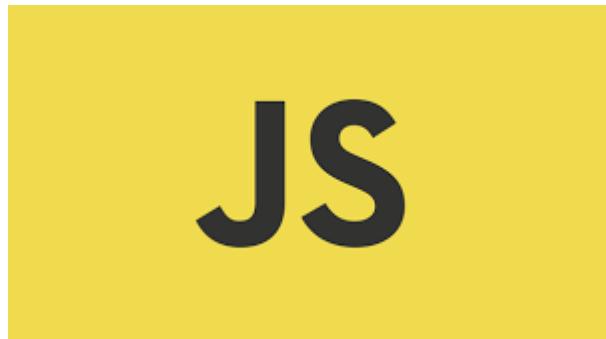


*Figure 02. MySQL*

MySQL is a relational database management system for accessing and manipulating data in a computer database. Database management systems are significant in computing not just because they specialize in processing vast volumes of data, but they can also be used as standalone utilities or as components of other

programs. MySQL Database Server is a fast, trustworthy, scalable, and user-friendly database server. It was created to manage massive databases considerably more quickly than other systems, and it has been used in high-volume commercial applications for a number of years. Furthermore, the connection, speed, and security of MySQL Server make it perfect for accessing databases via the Internet.

### 1.6.3. Javascript



*Figure 03. Javascript*

JavaScript, abbreviated as JS, is a programming language that, together with HTML and CSS, is one of the essential technologies of the World Wide Web. JavaScript is an ECMAScript-compliant high-level, frequently just-in-time compiled language. It has first-class functions, dynamic typing, and prototype-based object orientation. It's multi-paradigm, allowing you to program in event-driven, functional, or imperative styles. It contains APIs for working with text, dates, regular expressions, standard data structures, and the Document Object Model, among other things (DOM).

# 1.7. Schedule

## 1.7.1. Gantt Chart

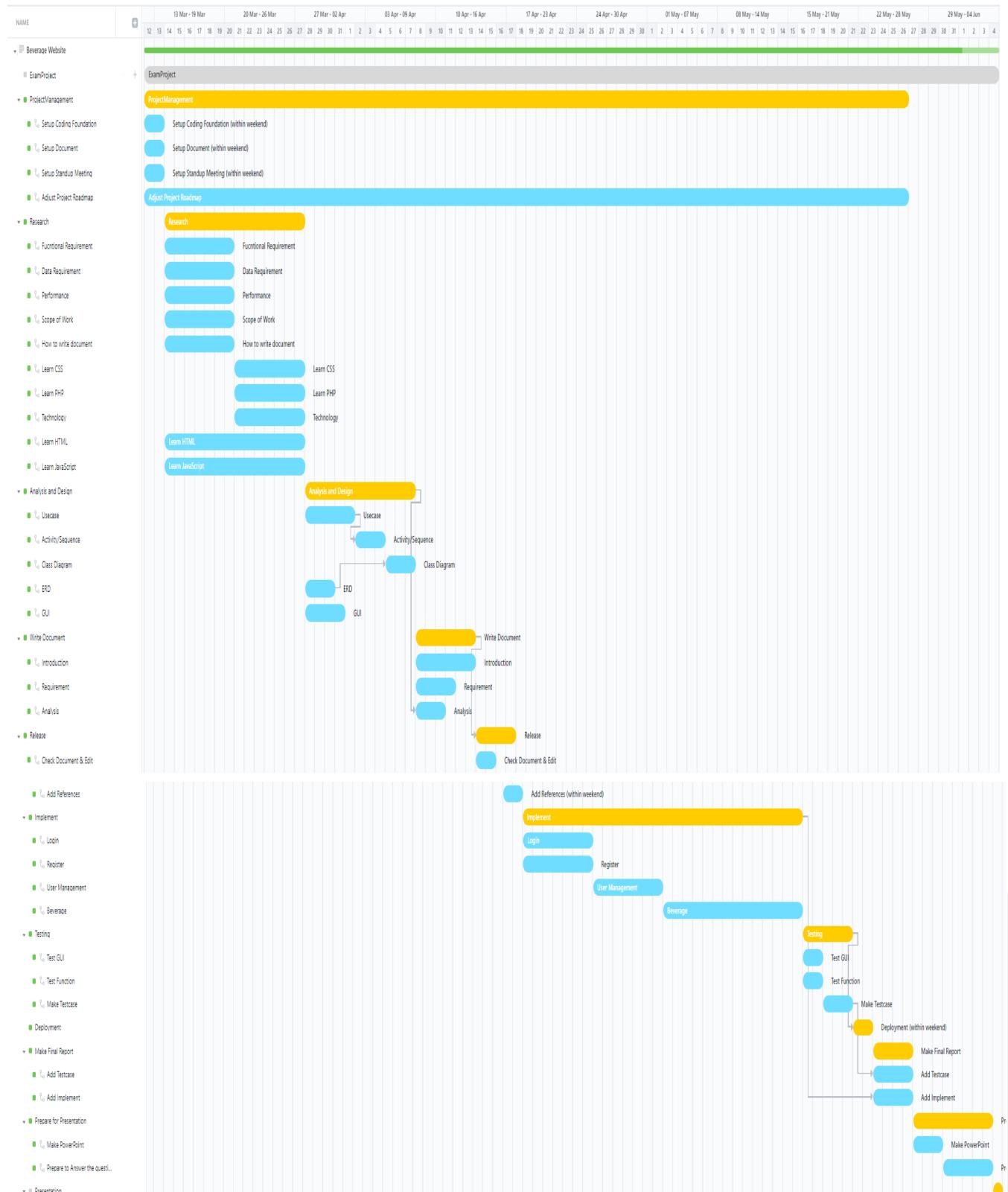
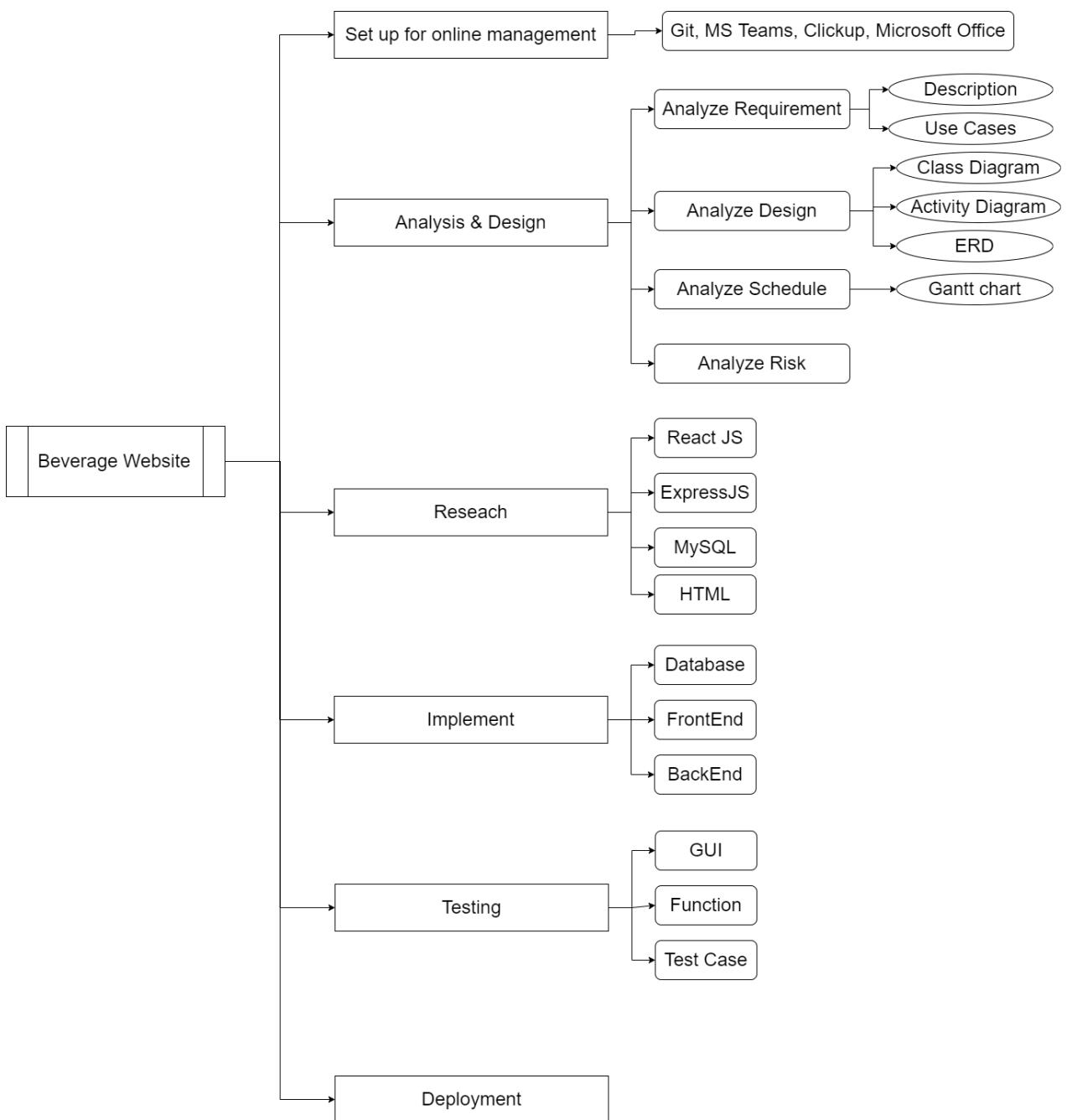


Figure 04. Gantt Chart

### 1.7.2. Work Breakdown Structure



**Figure 05. Work breakdown structure**

### 1.7.3. Schedule and milestone

#	Project / Epic / Task	Type	Assignee	Priority	Status	Start	Finish	Days
	Examine	Project				12/3/2022	4/6/2022	84 d
1	Project Management	Epic		Medium	Done	12/3/2022	26/5/2022	75 d
1.1	Setup Coding Foundation	Story	Công Tuấn	High	Done	12/3/2022	13/3/2022	1 d
1.2	Setup Document	Story	Hoàng Việt	High	Done	12/3/2022	13/3/2022	1 d
1.3	Setup Standup Meeting	Story	Phương Nam	High	Done	12/3/2022	13/3/2022	1 d
1.4	Adjust Project Roadmap	Story	Hoàng Việt	Medium	Done	12/3/2022	26/5/2022	75d
2	Research	Epic		High	Done	14/3/2022	27/3/2022	13 d
2.1	Functional Requirement	Story	Hoàng Việt	High	Done	14/3/2022	20/3/2022	6 d
2.2	Data Requirement	Story	Phương Nam	High	Done	14/3/2022	20/3/2022	6 d
2.3	Performance	Story	Đinh Khang	Medium	Done	14/3/2022	20/3/2022	6 d
2.4	Scope of Work	Story	Hoàng Việt	Medium	Done	14/3/2022	20/3/2022	6 d
2.5	How to write document	Story	Phương Nam	High	Done	14/3/2022	20/3/2022	6 d
2.6	Learn CSS	Story	Đinh Khang	Low	Done	21/3/2022	27/3/2022	6 d

2.7	<i>Learn PHP</i>	<i>Story</i>	<i>Phương Nam</i>	<i>Low</i>	<i>Done</i>	<i>21/3/2022</i>	<i>27/3/2022</i>	<i>6 d</i>
2.8	<i>Technology</i>	<i>Story</i>	<i>Phương Nam</i>	<i>Low</i>	<i>Done</i>	<i>21/3/2022</i>	<i>27/3/2022</i>	<i>6 d</i>
2.9	<i>Learn HTML</i>	<i>Story</i>	<i>Công Tuấn</i>	<i>Lowest</i>	<i>Done</i>	<i>14/3/2022</i>	<i>27/3/2022</i>	<i>12 d</i>
2.10	<i>Learn Javascript</i>	<i>Story</i>	<i>Công Tuấn</i>	<i>Lowest</i>	<i>Done</i>	<i>14/3/2022</i>	<i>27/3/2022</i>	<i>12 d</i>
3	<i>Analysis and Design</i>	<i>Epic</i>		<i>High</i>	<i>Done</i>	<i>28/3/2022</i>	<i>7/4/2022</i>	<i>10 d</i>
3.1	<i>Usecase</i>	<i>Story</i>	<i>Hoàng Việt</i>	<i>High</i>	<i>Done</i>	<i>28/3/2022</i>	<i>1/4/2022</i>	<i>4 d</i>
3.2	<i>Activity/ Sequence</i>	<i>Story</i>	<i>Hoàng Việt</i>	<i>High</i>	<i>Done</i>	<i>2/4/2022</i>	<i>4/4/2022</i>	<i>2 d</i>
3.3	<i>Class Diagram</i>	<i>Story</i>	<i>Hoàng Việt</i>	<i>Medium</i>	<i>Done</i>	<i>5/4/2022</i>	<i>7/4/2022</i>	<i>2 d</i>
3.4	<i>ERD</i>	<i>Story</i>	<i>Hoàng Việt</i>	<i>Medium</i>	<i>Done</i>	<i>28/3/2022</i>	<i>30/3/2022</i>	<i>2 d</i>
3.5	<i>GUI</i>	<i>Story</i>	<i>Công Tuấn</i>	<i>Medium</i>	<i>Done</i>	<i>28/3/2022</i>	<i>31/3/2022</i>	<i>3 d</i>
4	<i>Write Document</i>	<i>Epic</i>		<i>Medium</i>	<i>Done</i>	<i>8/4/2022</i>	<i>13/4/2022</i>	<i>5d</i>
4.1	<i>Introduction</i>	<i>Story</i>	<i>Phương Nam</i>	<i>Medium</i>	<i>Done</i>	<i>8/4/2022</i>	<i>13/4/2022</i>	<i>5d</i>
4.1.1	<i>Project Overview</i>	<i>Sub task</i>	<i>Phương Nam</i>	<i>Medium</i>	<i>Done</i>	<i>8/4/2022</i>	<i>13/4/2022</i>	<i>5d</i>
4.1.2	<i>Purpose of Project</i>	<i>Sub task</i>	<i>Phương Nam</i>	<i>Medium</i>	<i>Done</i>	<i>8/4/2022</i>	<i>13/4/2022</i>	<i>5d</i>

<b>4.1.3</b>	<i>Scope of Work</i>	<i>Sub task</i>	<i>Đinh Khang</i>	Medium	<i>Done</i>	8/4/2022	<b>13/4/2022</b>	<b>5 d</b>
<b>4.1.4</b>	<i>Organization</i>	<i>Sub task</i>	<i>Phương Nam</i>	Medium	<i>Done</i>	8/4/2022	<b>13/4/2022</b>	<b>5 d</b>
<b>4.1.5</b>	<i>Resource Requirement</i>	<i>Sub task</i>	<i>Đinh Khang</i>	Medium	<i>Done</i>	8/4/2022	<b>13/4/2022</b>	<b>5 d</b>
<b>4.1.6</b>	<i>Technology</i>	<i>Sub task</i>	<i>Phương Nam</i>	Medium	<i>Done</i>	8/4/2022	<b>13/4/2022</b>	<b>5 d</b>
<b>4.1.7</b>	<i>Schedule</i>	<i>Sub task</i>	<i>Phương Nam</i>	Medium	<i>Done</i>	8/4/2022	<b>13/4/2022</b>	<b>5 d</b>
<b>4.1.8</b>	<i>Risk Management</i>	<i>Sub task</i>	<i>Phương Nam</i>	Medium	<i>Done</i>	8/4/2022	<b>13/4/2022</b>	<b>5 d</b>
<b>4.1.9</b>	<i>Security Aspect</i>	<i>Sub task</i>	<i>Công Tuấn</i>	Medium	<i>Done</i>	8/4/2022	<b>13/4/2022</b>	<b>5 d</b>
<b>4.1.10</b>	<i>Naming Conventions</i>	<i>Sub task</i>	<i>Công Tuấn</i>	Medium	<i>Done</i>	8/4/2022	<b>13/4/2022</b>	<b>5 d</b>
<b>4.1.11</b>	<i>Conclusions</i>	<i>Sub task</i>	<i>Phương Nam</i>	Medium	<i>Done</i>	8/4/2022	<b>13/4/2022</b>	<b>5 d</b>
<b>4.2</b>	<i>Requirement</i>	<i>Story</i>	<i>Hoàng Việt</i>	High	<i>Done</i>	8/4/2022	<b>11/4/2022</b>	<b>3 d</b>
<b>4.2.1</b>	<i>Product Use Cases</i>	<i>Sub task</i>	<i>Hoàng Việt</i>	Medium	<i>Done</i>	8/4/2022	<b>11/4/2022</b>	<b>3 d</b>
<b>4.2.2</b>	<i>Functional Requirements</i>	<i>Sub task</i>	<i>Công Tuấn</i>	Medium	<i>Done</i>	8/4/2022	<b>11/4/2022</b>	<b>3 d</b>
<b>4.2.3</b>	<i>Data Requirements</i>	<i>Sub task</i>	<i>Phương Nam</i>	Medium	<i>Done</i>	8/4/2022	<b>11/4/2022</b>	<b>3 d</b>
<b>4.2.4</b>	<i>Performance</i>	<i>Sub task</i>	<i>Công Tuấn</i>	Medium	<i>Done</i>	8/4/2022	<b>11/4/2022</b>	<b>3 d</b>

<b>4.2.5</b>	<i>Dependability</i>	<i>Sub task</i>	<i>Phương Nam</i>	<i>Medium</i>	<i>Done</i>	<i>8/4/2022</i>	<i>11/4/2022</i>	<i>3 d</i>
<b>4.2.6</b>	<i>Security</i>	<i>Sub task</i>	<i>Hoàng Việt</i>	<i>Medium</i>	<i>Done</i>	<i>8/4/2022</i>	<i>11/4/2022</i>	<i>3 d</i>
<b>4.3</b>	<i>Analysis</i>	<i>Story</i>	<i>Đinh Khang</i>	<i>High</i>	<i>Done</i>	<i>8/4/2022</i>	<i>10/4/2022</i>	<i>2 d</i>
<b>4.3.1</b>	<i>System Design</i>	<i>Sub task</i>	<i>Công Tuấn</i>	<i>Medium</i>	<i>Done</i>	<i>8/4/2022</i>	<i>10/4/2022</i>	<i>2 d</i>
<b>4.3.2</b>	<i>Software Architecture</i>	<i>Sub task</i>	<i>Công Tuấn</i>	<i>Medium</i>	<i>Done</i>	<i>8/4/2022</i>	<i>10/4/2022</i>	<i>2 d</i>
<b>5</b>	<i>Release</i>	<i>Epic</i>		<i>Low</i>	<i>Done</i>	<i>14/4/2022</i>	<i>17/4/2022</i>	<i>3 d</i>
<b>5.1</b>	<i>Check &amp; Edit Document</i>	<i>Story</i>	<i>Hoàng Việt</i>	<i>Low</i>	<i>Done</i>	<i>14/4/2022</i>	<i>15/4/2022</i>	<i>1 d</i>
<b>5.2</b>	<i>Add References</i>	<i>Story</i>	<i>Hoàng Việt</i>	<i>Low</i>	<i>Done</i>	<i>16/4/2022</i>	<i>17/4/2022</i>	<i>1 d</i>
<b>6</b>	<i>Implement</i>	<i>Epic</i>		<i>High</i>	<i>Done</i>	<i>18/4/2022</i>	<i>15/5/2022</i>	<i>28 d</i>
<b>6.1</b>	<i>Login</i>	<i>Story</i>	<i>Công Tuấn</i>	<i>Medium</i>	<i>Done</i>	<i>18/4/2022</i>	<i>24/4/2022</i>	<i>6 d</i>
<b>6.2</b>	<i>Register</i>	<i>Story</i>	<i>Phương Nam</i>	<i>Medium</i>	<i>Done</i>	<i>18/4/2022</i>	<i>24/4/2022</i>	<i>6 d</i>
<b>6.3</b>	<i>User Management</i>	<i>Story</i>	<i>Phương Nam</i>	<i>Medium</i>	<i>Done</i>	<i>25/4/2022</i>	<i>1/5/2022</i>	<i>6 d</i>
<b>6.4</b>	<i>Beverage</i>	<i>Story</i>	<i>Công Tuấn</i>	<i>High</i>	<i>Done</i>	<i>2/5/2022</i>	<i>15/5/2022</i>	<i>6 d</i>
<b>7</b>	<i>Testing</i>	<i>Epic</i>		<i>High</i>	<i>Done</i>	<i>16/5/2022</i>	<i>20/5/2022</i>	<i>4 d</i>
<b>7.1</b>	<i>Test GUI</i>	<i>Story</i>	<i>Hoàng Việt</i>	<i>Medium</i>	<i>Done</i>	<i>16/5/2022</i>	<i>17/5/2022</i>	<i>1 d</i>

7.2	<i>Test Function</i>	<i>Story</i>	<i>Đinh Khang</i>	<i>Medium</i>	<i>Done</i>	<i>16/5/2022</i>	<i>17/5/2022</i>	<i>1 d</i>
7.3	<i>Make Testcase</i>	<i>Sub task</i>	<i>Phương Nam</i>	<i>High</i>	<i>Done</i>	<i>18/5/2022</i>	<i>20/5/2022</i>	<i>2 d</i>
8	<i>Deployment</i>	<i>Epic</i>	<i>Phương Nam</i>	<i>Low</i>	<i>Done</i>	<i>21/5/2022</i>	<i>22/5/2022</i>	<i>1 d</i>
9	<i>Make Final Report</i>	<i>Epic</i>	<i>Hoàng Việt</i>	<i>Low</i>	<i>Done</i>	<i>23/5/2022</i>	<i>26/5/2022</i>	<i>3 d</i>
9.1	<i>Add Testcase</i>	<i>Sub task</i>	<i>Hoàng Việt</i>	<i>Low</i>	<i>Done</i>	<i>23/5/2022</i>	<i>26/5/2022</i>	<i>3 d</i>
9.2	<i>Add Implement</i>	<i>Sub task</i>	<i>Đinh Khang</i>	<i>Low</i>	<i>Done</i>	<i>23/5/2022</i>	<i>26/5/2022</i>	<i>3 d</i>
10	<i>Prepare for Presentation</i>	<i>Epic</i>		<i>Low</i>	<i>Done</i>	<i>27/5/2022</i>	<i>3/6/2022</i>	<i>7 d</i>
10.1	<i>Make PowerPoint</i>	<i>Sub task</i>	<i>Đinh Khang</i>	<i>Low</i>	<i>Done</i>	<i>27/5/2022</i>	<i>29/5/2022</i>	<i>2 d</i>
10.2	<i>Prepare to Answer the question</i>	<i>Story</i>	<i>Đinh Khang</i>	<i>Low</i>	<i>Done</i>	<i>30/5/2022</i>	<i>3/6/2022</i>	<i>4 d</i>
11	<i>Presentation</i>	<i>Epic</i>		<i>High</i>	<i>To Do</i>	<i>4/6/2022</i>	<i>4/6/2022</i>	<i>0 d</i>

*Table 05. Schedule and milestone*

#### 1.7.4. Process Development

Agile planning strategies for team communication and collaboration are required to carry out joint systems efficiently. To conclude, the project manager first devotes time to identifying the job's requirements and aligning the total collection of products. The joint venture director will call a meeting after the fundamental arrangements have been made to chat and assign errands. Throughout the life of the system, a weekly MTeams meeting will be held to help keep the system on track by completing normal tests and assessments. Each week's detail must take into account

what each individual has accomplished and what will be accomplished. Simultaneously, it will be done and tested over and again until the project manager is pleased.

## 1.8. Risk Management

This section outlines some of the potential risks that our project may face during development and/or after release, as well as some options for mitigating and avoiding the potentially harmful consequences these risks may have on the final result. There are three types of risk classifications:

#	Title	Description	Probability	Effects	Strategy
1	Member under-performance	Individual performances fall short of expectations due to a lack of knowledge and experience with the project's subject.	Low	Medium	To help overcome experience gaps, the project leader must balance the amount of junior and experienced developers in the team.
2	Unexpected Bugs	Bugs in the user interface, missing fonts, and a slew of other issues that irritate consumers but have never been discovered previously.	High	High	Update functions and refactor codes on a regular basis and use current tools.
3	Requirements changes	Users' changing needs have a direct impact on the project schedule.	High	High	As user demand changes, adjust the timing and amount of work.
5	Hardware confliction	Because certain team members' computers have the wrong version of the	Medium	High	At the start of the project, project managers will set up and inspect everything to ensure that

		application, the project may not operate on their PCs.			the team can change tools quickly.
6	Wrong Database	The system incorrectly updates data (e.g., balance, location)	High	High	Requires the developer to construct an efficient algorithm so that the system can execute with more precision
7	Late Timeline	Do not submit, and do not finish the assignment on time.	Medium	Medium	The project leader must ensure that each sub-component of the system, as well as the end result, is completed ahead of schedule.

*Table 06. Risk Management*

## 1.9. Security Aspect

Users will have password-protected access to web pages that are only to be accessed by them.

Data from transactions must be sent in an encrypted format.

## 1.10. Naming Conventions and Definitions

#	Name	Definitions
1	HTML (Hypertext Markup Language)	The programming language that was utilized to create and restructure the Website's components.
2	CSS (Cascading Style Sheets)	A language for locating and reformatting webpage (HTML) pieces that have been generated.
3	JS (JavaScript)	A scripting or programming language that allows complicated features on a website to be implemented.
4	PHP (Hypertext Preprocessor)	Popular general-purpose scripting language with a focus on web development
5	MS Teams (Microsoft Teams)	a Microsoft-developed proprietary commercial communication platform

6	GUI (Graphical User Interface)	is a type of user interface that uses graphical icons and aural indicators to allow people to interact with electronic devices, such as primary notation.
7	Subtask	The smaller task of User Story
8	User Story	An informal, natural language description of a software system's features.
10	ERD (entity relationship model)	describes a set of interconnected items of interest in a given field of knowledge.

***Table 07. Naming Convention and Definitions***

## 1.11. Conclusion

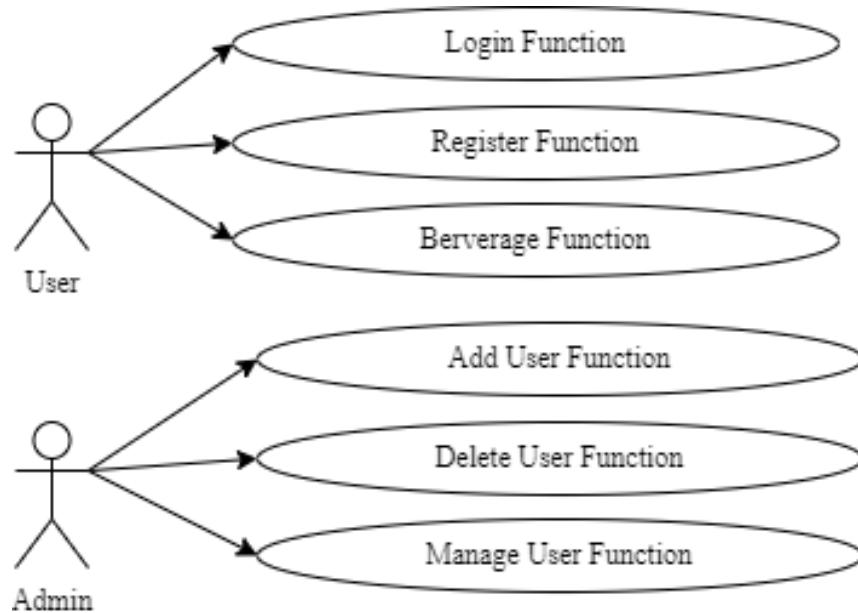
Beverage Website is a practical website serving the needs of exploring, creating, and providing new drinks for everyone. Systems are working hard to make the application useful capabilities, such as membership creation, monthly offers, and delivery policy. Later on, that makes the app a bigger reality.

This project combines some basic expertise and flair to expand executives' insight, coding prowess, logical reasoning, and commitment. Students make progress on assignments by overcoming difficulties, clearly separating obligations to individuals, handling the timing and progress of every section, and managing joint ventures with a team throughout. . The project provided an important experience that could be applied in the future. Learners found out how to develop a web application, as well as build a coherent information base suitable for the job and connect it to the programming language. Furthermore, solid program engineering planning and investigation of customer needs are like converting real-life scenarios into programming.

# Chapter 2: Requirements

## 2.1. Product Use Cases

### 2.1.1. Use Case Diagrams



*Figure 06. Use Case Diagram*

### 2.1.2. Product Use Case List

Number	Use Case Name	Use Case Description
1	User register	A visitor to the website clicks on “register”, fills in the form, and submits
2	User login	A visitor clicks on “log in”, fills in the form, and submit
3	Change password	A registered user clicks on “forget password” to change
4	Delete user	Admin bans inappropriate users (bad names, system exploiting, ...)
5	Search recipe	A user searches for the drink by choosing 3 ingredients

*Table 08. Product Use Case List*

Use Case Name:	User register
Summary:	To get saved recipes, a new user must register a username and password.
Basic Flow:	<ol style="list-style-type: none"> <li>1. The use case starts when a user indicates that he wants to register.</li> <li>2. The system requests a full name, email address, username, and password.</li> <li>3. The user enters a username and password.</li> <li>4. The system checks that the username does not duplicate any existing registered usernames.</li> <li>5. The system requests the user's full name, and email address.</li> <li>6. The user enters the information.</li> <li>7. The system determines the user's access level and stores all user information.</li> <li>8. The system starts a login session and displays a welcome message based on the user's preferences.</li> </ol>
Alternative Flows:	<p>Step 4: If the username duplicates an existing username the system displays a message and the use case goes back to step 2.</p> <p>Step 5: If the user does not enter a required field, a message is displayed and the use case repeats step 4.</p>
Extension Points:	None
Preconditions:	None
Postconditions:	The user can now obtain data and perform functions according to his registered access level.
Business Rules:	<ul style="list-style-type: none"> <li>● A registered user's location is the SBE location nearest his zip code.</li> <li>● Access levels are <ul style="list-style-type: none"> <li>○ 0: A user can access only data classification 0</li> <li>○ 1: The user can access data classification &lt;= 1</li> <li>○ 2: The user can access data classification &lt;= 2</li> </ul> </li> <li>● The default access level is 0.</li> </ul>

**Table 09. User register Use case**

Use Case Name:	User login
Summary:	To get saved recipes, a new user must register a username and password.
Basic Flow:	<ol style="list-style-type: none"> <li>1. The use case starts when a user indicates that he wants to log in.</li> <li>2. The system requests the username and password.</li> <li>3. The user enters his username and password.</li> <li>4. The system verifies the username and password against all registered users.</li> <li>5. The system starts a login session and displays a welcome message based on the user's preferences.</li> </ol>
Alternative Flows:	<p>Step 4: If the username is invalid, the use case goes back to step 2.</p> <p>Step 4: If the password is invalid the system requests that the user re-enter the password. When the user enters another password the use case continues with step 4 using the original username and new password.</p>
Extension Points:	None
Preconditions:	The user is registered.
Postconditions:	The user can now obtain data and perform functions according to his registered access level.
Business Rules:	Some data and functions are restricted to certain types of users or users with a particular access level.

**Table 10. User login Use case**

Use Case Name:	Change user's password
Summary:	A current user forgot password, and requested to change it.
Basic Flow:	<ol style="list-style-type: none"> <li>1. User clicks on forgot password button</li> <li>2. System requests user to input the email</li> <li>3. System sends verify code to the email</li> <li>4. User inputs the code and types in new password</li> </ol>
Alternative Flows:	<p>Step 2: If the email is invalid, abort.</p> <p>Step 4: If the password is invalid, prompt the user to input it again</p>
Extension Points:	None
Preconditions:	The user is registered.
Postconditions:	The user can now access the account
Business Rules:	Some data and functions are restricted to certain types of users or users with a particular access level.

**Table 11. Change user' password Use case**

Use Case Name:	Delete User
Summary:	Admin deletes a user for violating Terms of Service
Basic Flow:	<ol style="list-style-type: none"> <li>1. Admin checks an account and finds evidence that the user violated Terms of Service.</li> <li>2. Admin deletes the account from the server and blacklists the email.</li> </ol>
Alternative Flows:	None
Extension Points:	None
Preconditions:	The user is registered.
Postconditions:	The user cannot use the account anymore. User also cannot use the same email to register a new account.
Business Rules:	Some data and functions are restricted to certain types of users or users with a particular access level.

***Table 12. Delete user Use case***

Use Case Name:	Find recipe
Summary:	User find recipe based on 3 initial ingredients
Basic Flow:	<ol style="list-style-type: none"> <li>1. User clicks on Find recipe button</li> <li>2. System requires user to choose 3 ingredients</li> <li>3. System finds the recipe's name based on the ingredients chosen</li> <li>4. System displays the recipe's name</li> </ol>
Alternative Flows:	None
Extension Points:	None
Preconditions:	3 ingredients are chosen
Postconditions:	None
Business Rules:	Some data and functions are restricted to certain types of users or users with a particular access level.

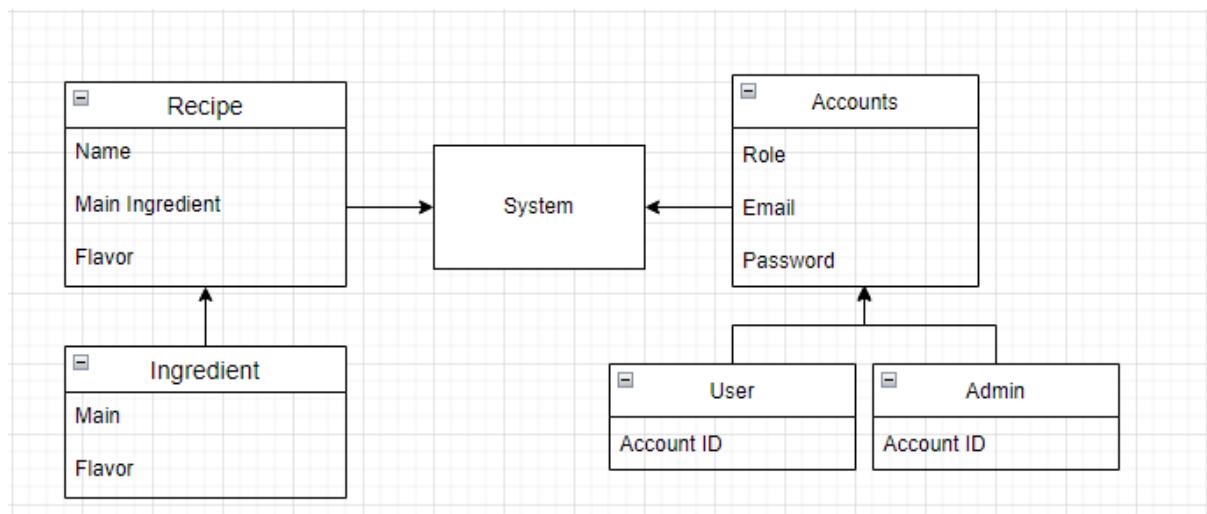
*Table 13. Find recipe Use case*

## 2.2. Functional & Non-functional Requirements

Req . ID	Requirement Name	Detailed Description	Type	Priority
1	Login	Login for users/admins if they already have account	Functional Requirement	High Priority
2	Create Account	Create account for users	Functional Requirement	High Priority
3	Logout	Logout the current account	Functional Requirement	High Priority
4	Add New User	Allow the admin to add new users	Functional Requirement	High Priority
5	Manage Users	Allow the admin to see all users' information	Functional Requirement	High priority
6	Find Users	Allow the admin to find users' information	Functional Requirement	High Priority
7	Add New Ingredient	Admin can add more ingredient manually	Functional Requirement	Medium Priority
8	Create new recipe	Admin can create new recipe manually	Functional Requirement	Medium Priority
9	Security Requirement	The information of users and admin must be encrypted	Non-functional requirement	High priority
10	Usability Requirement	The interface is simple to use. The announcement/ error message is easy to understand	Non-functional requirement	High priority
11	Extensibility Requirement	The system has the ability to update new functions that do not need to change the database structure.	Non-functional requirement	Low priority

**Table 14. Functional & Non-functional Requirements**

## 2.3. Data Requirements



*Figure 07. Data requirements*

Data is designed simply with two account roles with enough function to manage the system.

Moreover, the beverage data is categorized by name and recipe.

## 2.4. Performance Requirements

### 2.4.1. Speed and latency requirement

One important requirement of a web application is server upload time, which is called time response. The shorter the response time, the better the user experience. Therefore, the suitable maximum response time of the web application to load the whole content is 1 second. Moreover, the recommendation of downloading speed is at least 25 Mbps and the uploading speed is at least 3 Mbps.

### 2.4.2. Capacity requirement

A system must be specified clearly at first to ensure that it has enough spare capacity for dealing with each of the amount of information, data and services. Therefore, it is important to regularly check and compute the free capacity for each service.

## **2.5. Dependability Requirements**

### **2.5.1. Reliability requirement**

All contents are fictional and intended for illustrative purposes only. Users' discreteness and politeness are needed. Relating to name policy, users' liability covers responsibility for any confusing or aggressive names.

### **2.5.2. Availability requirement**

Users are required to enable Javascript on the browser during their stay on the website. A modern browser with support for html5 and css3 is needed. A stable internet connection is required for this web application.

## **2.6. Security Requirements**

### **2.6.1. Access requirements**

Login is required to save your favorite drinks to your account. Visitors can search for drinks without the need of logging in.

### **2.6.2. Privacy requirements**

We use cookies to determine who the user is, and if that user had previously logged in. Cookies also allow us to help the users stay logged in if they want to.

# Chapter 3: Design

## 3.1. System Design

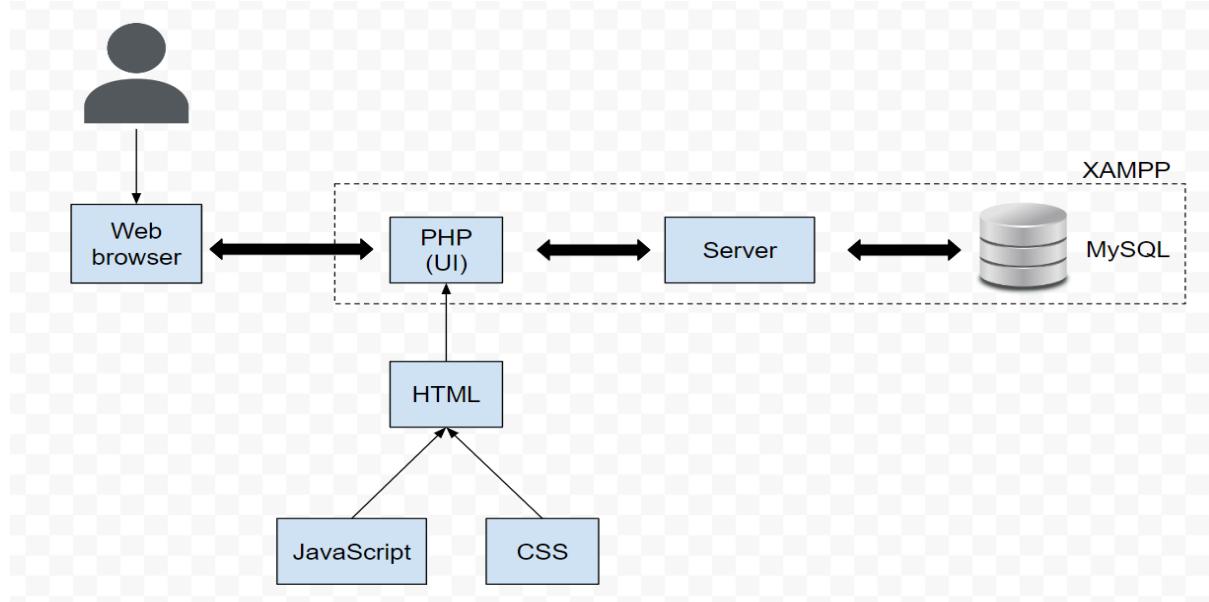


Figure 08. System design

## 3.2. Database Structure

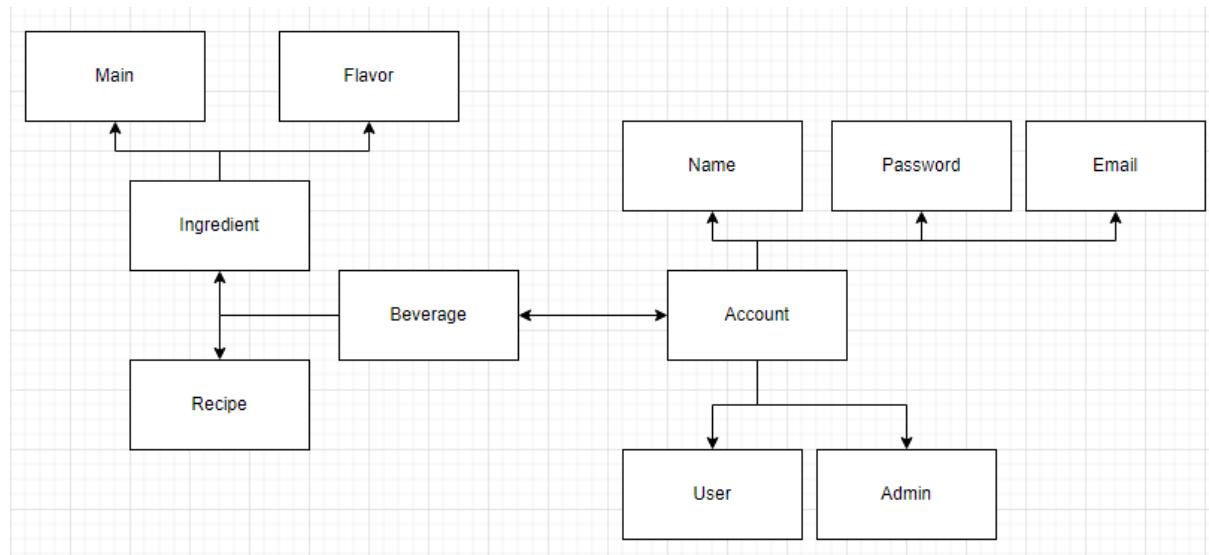
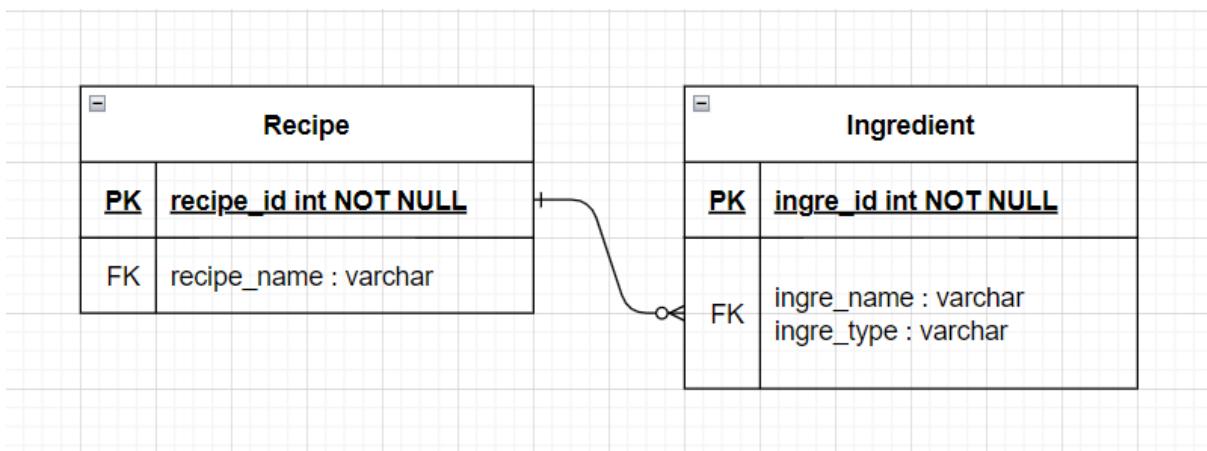


Figure 09. Database Structure

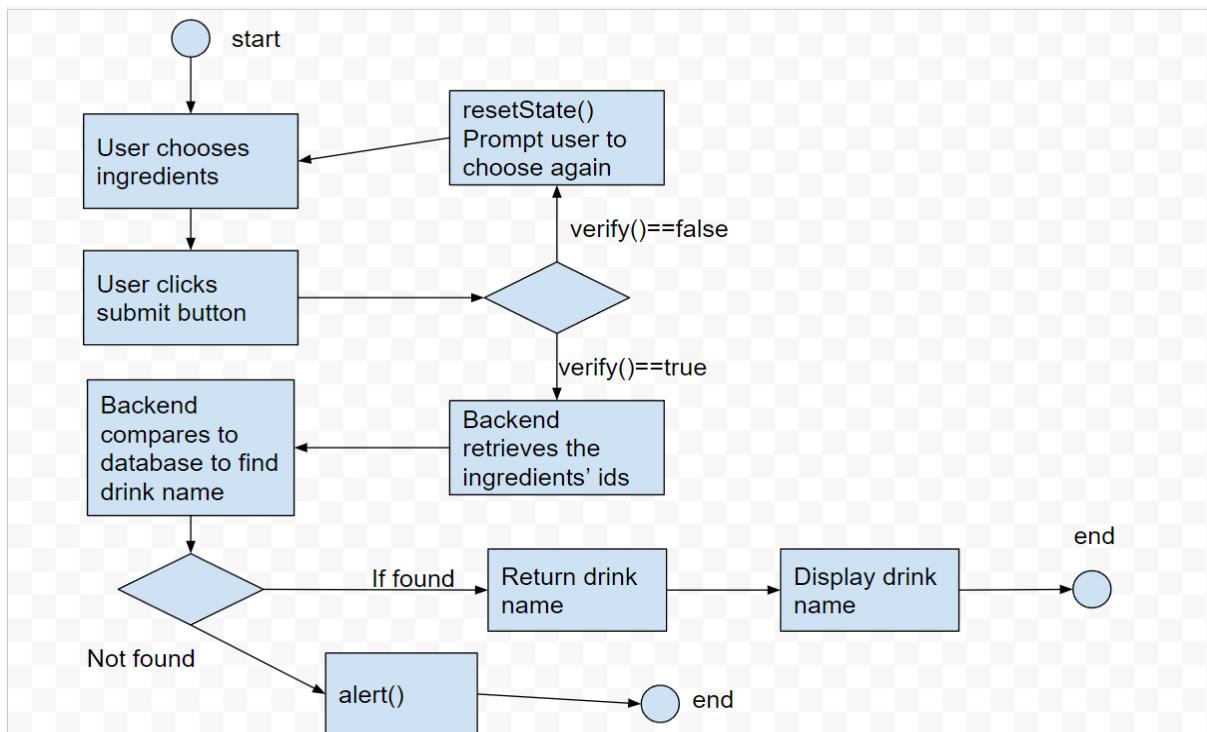
The database includes the account data (users & admin) along with the beverage data. Beverage data has some ingredients and creates recipes based on ingredient data.

### 3.3. Entity–relationship Diagram



*Figure 10. Entity relationship Diagram*

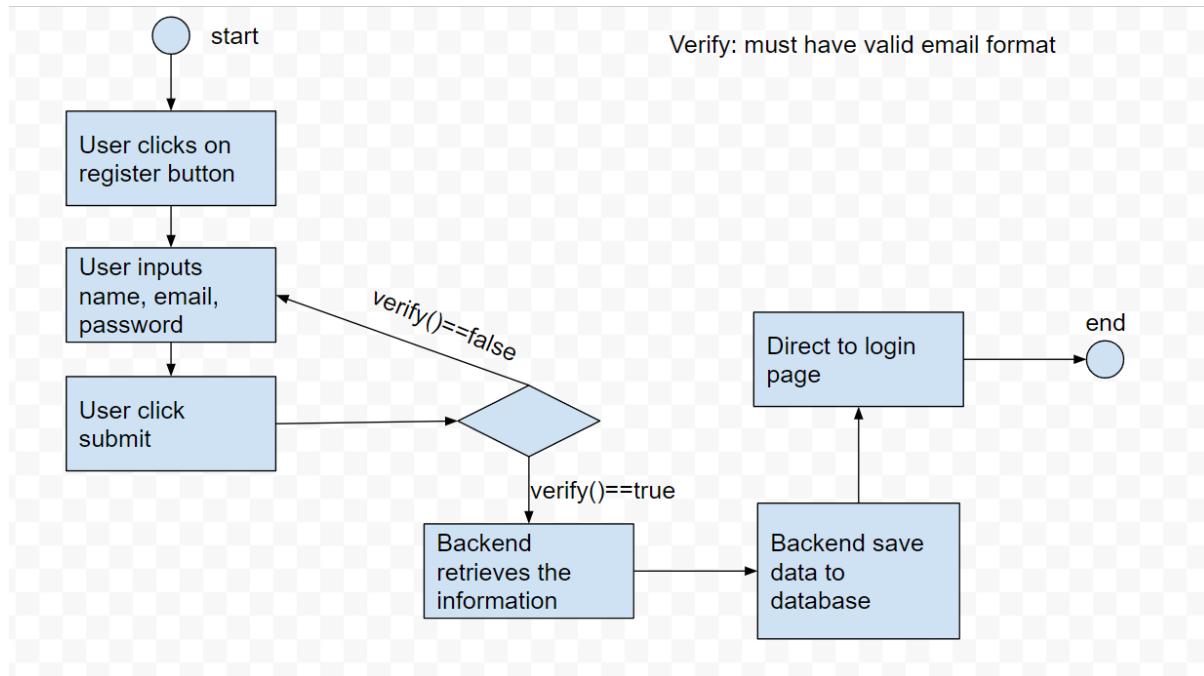
### 3.4. Activity Diagram



*Figure 11. Search for recipe Diagram*

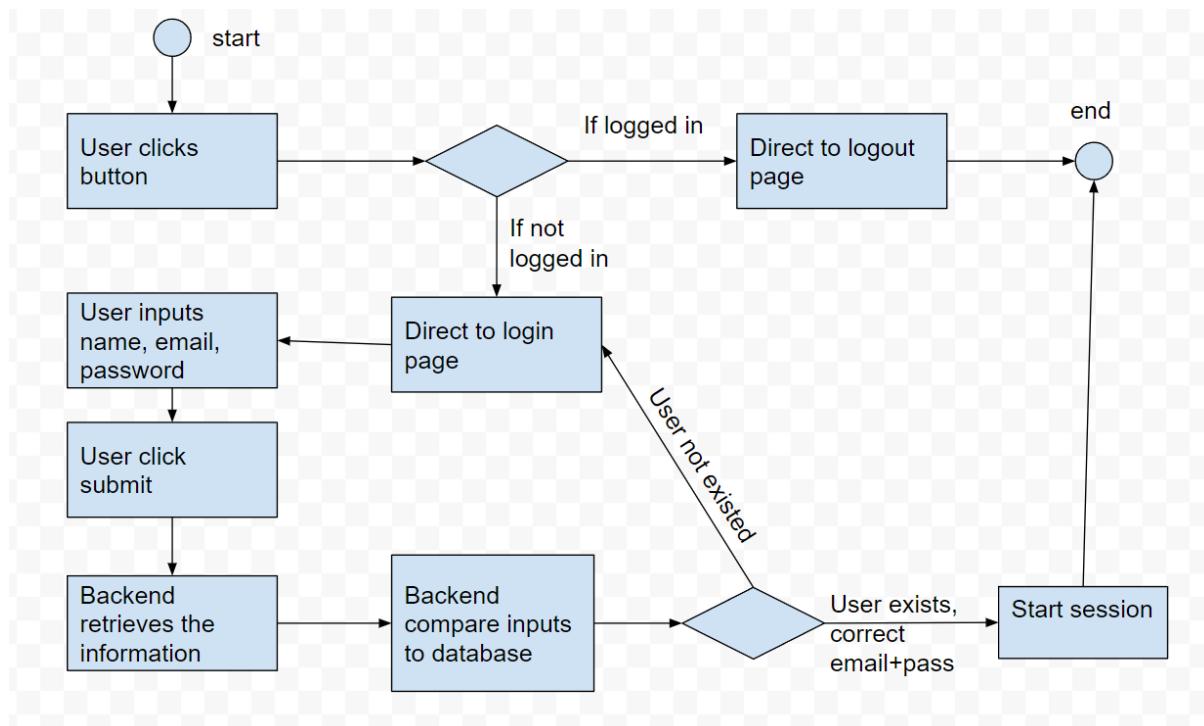
- System check verification
  - + If false return to main state
  - + If true return to backend
- Check the drink name
  - + If found, return it to client site
  - + If not found, return the warning to client

### Use case: Register new user



**Figure 12. New user register Diagram**

### Use case: Login/logout



**Figure 13. Login / Logout Diagram**

### 3.5. Class Diagram

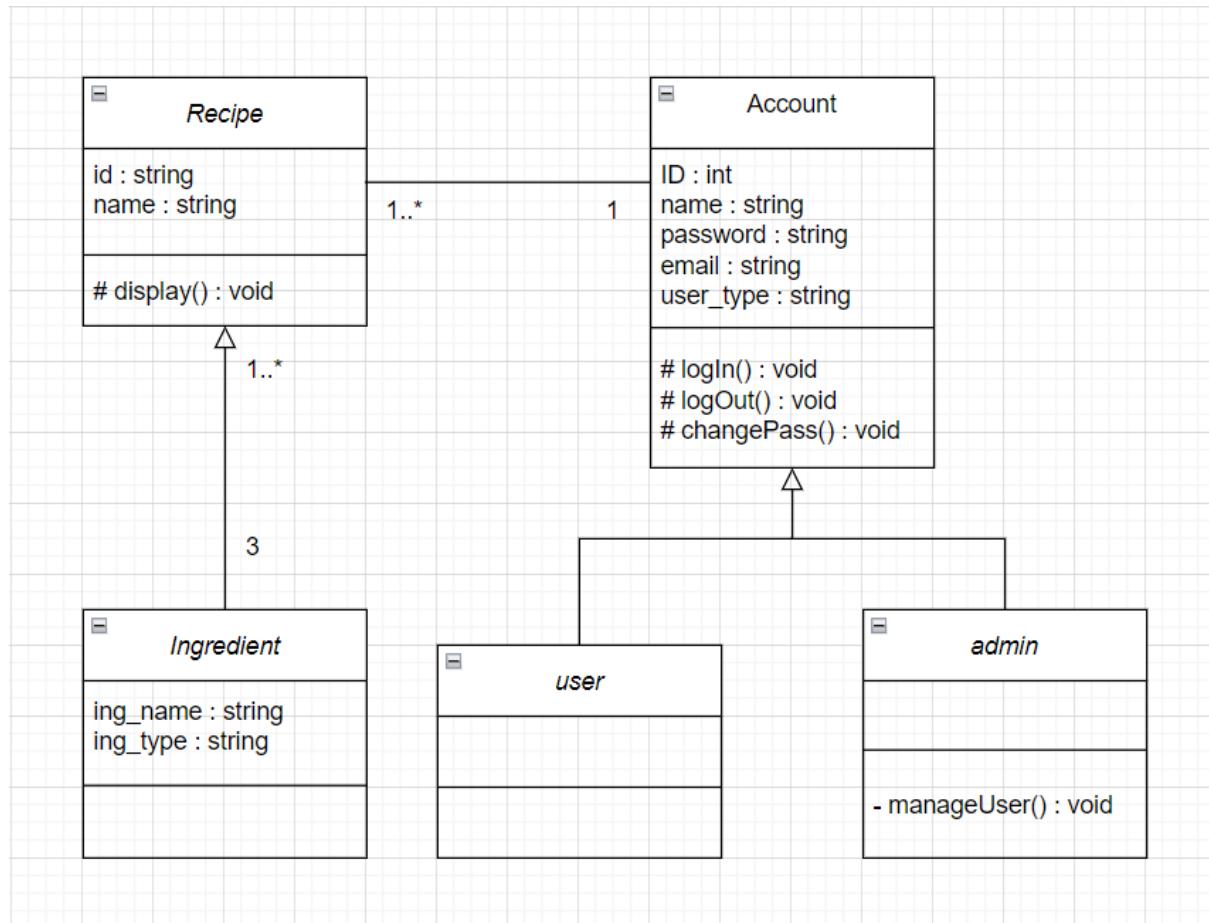
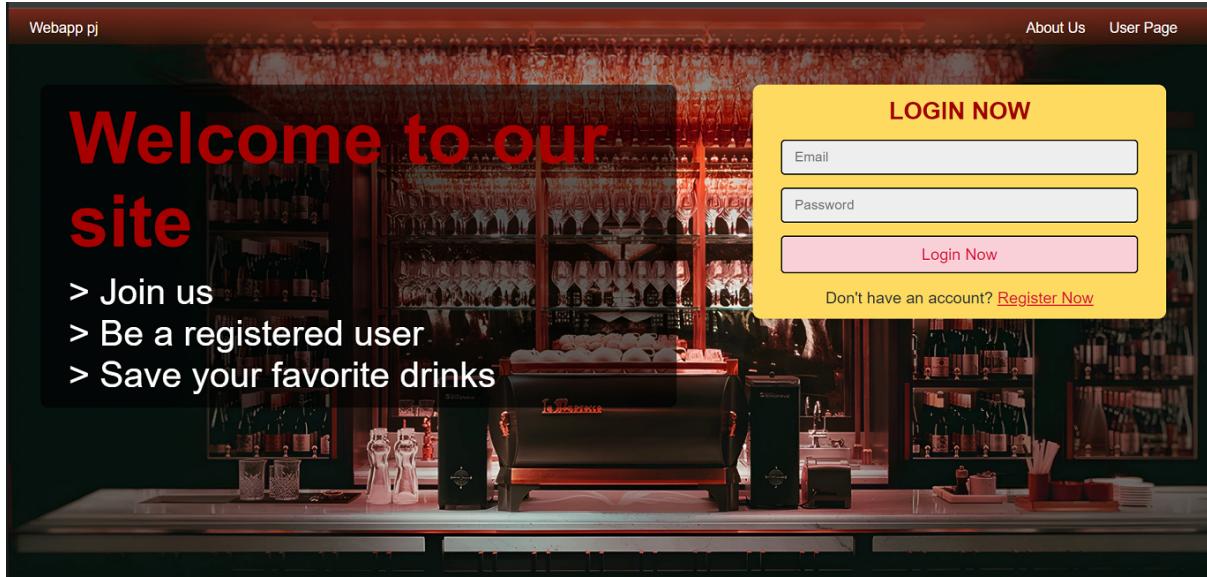


Figure 14. Class Diagram

# Chapter 4: Implementation

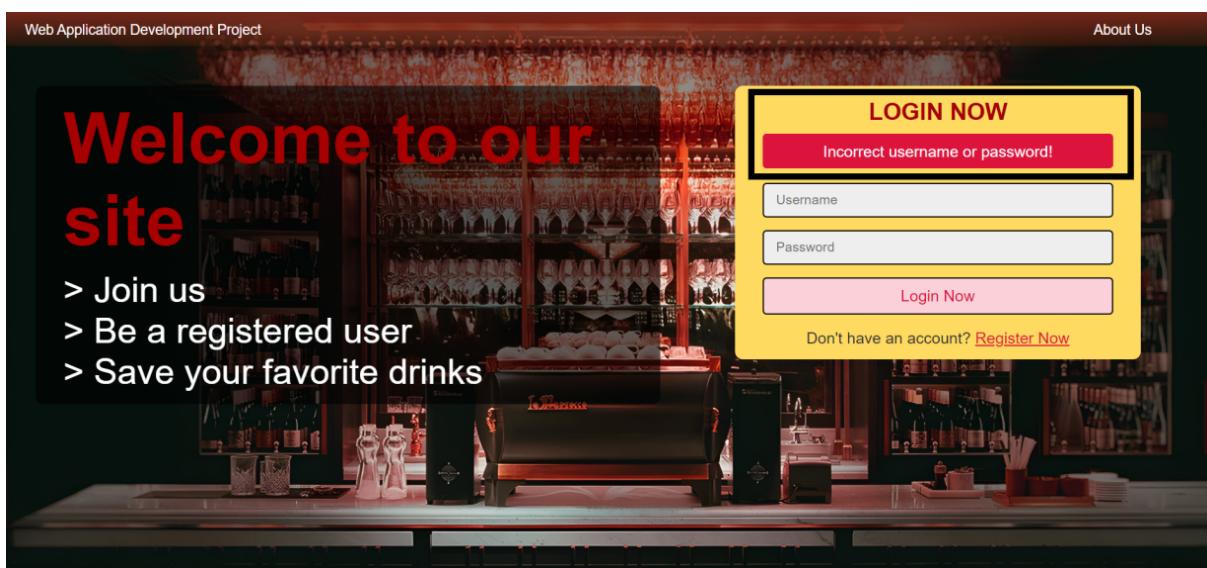
## 4.1. Login function

User needs to input email and password as previously registered.



*Figure 15. Login interface*

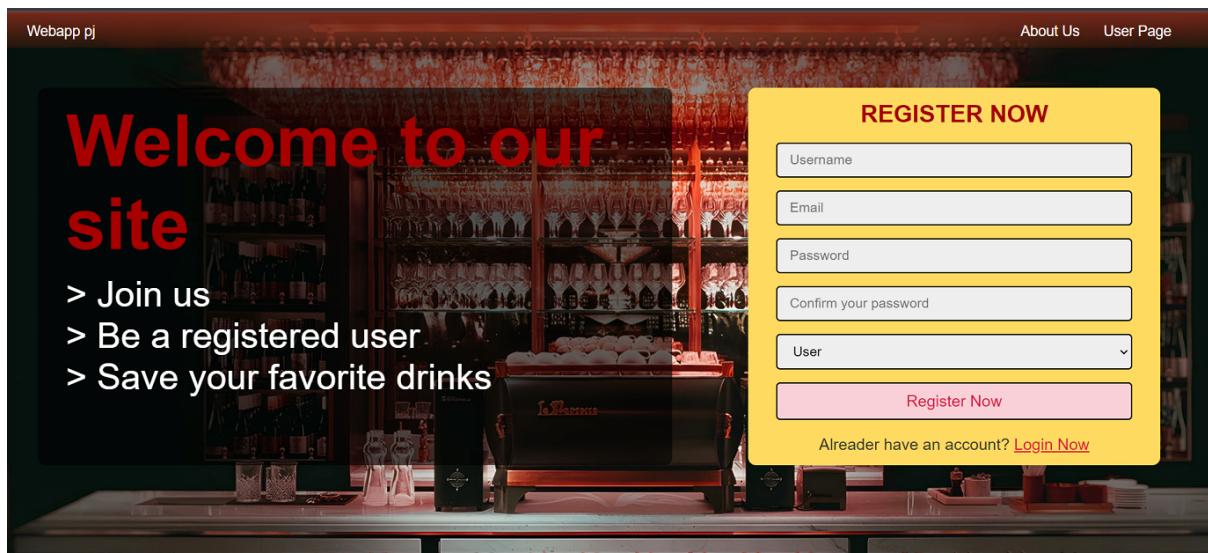
If the user input is wrong, backend will reject that attempt and prompt the user.



*Figure 16. Wrong login information*

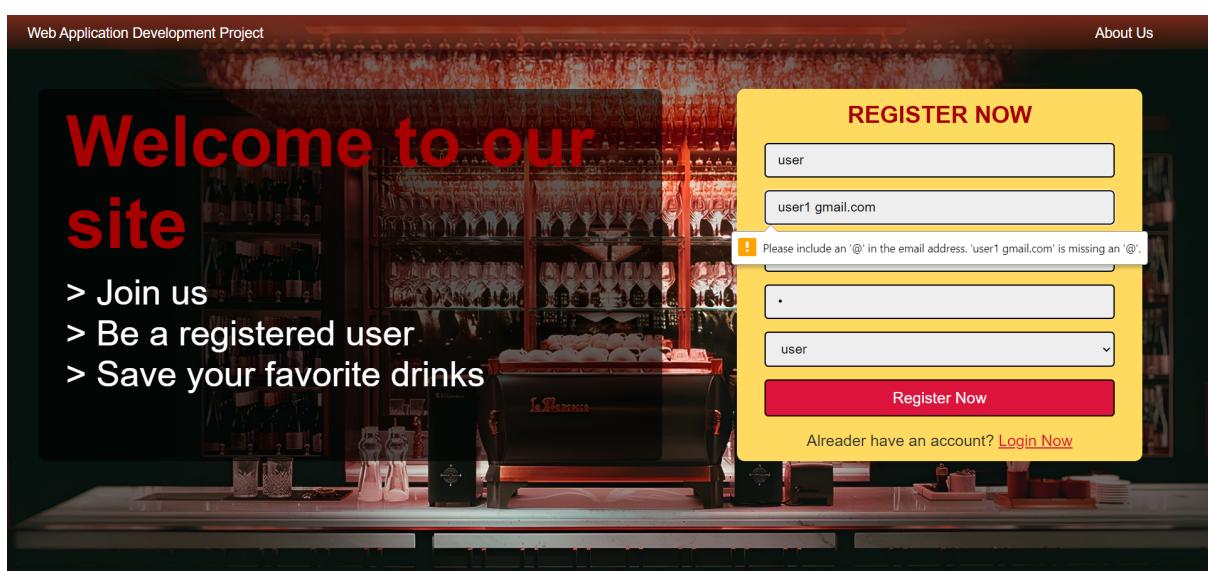
## 4.2. Register function

To register an account, a user must input a valid name, email, password and click the “Register Now” button. Backend will collect those informations and save to database



*Figure 17. Register*

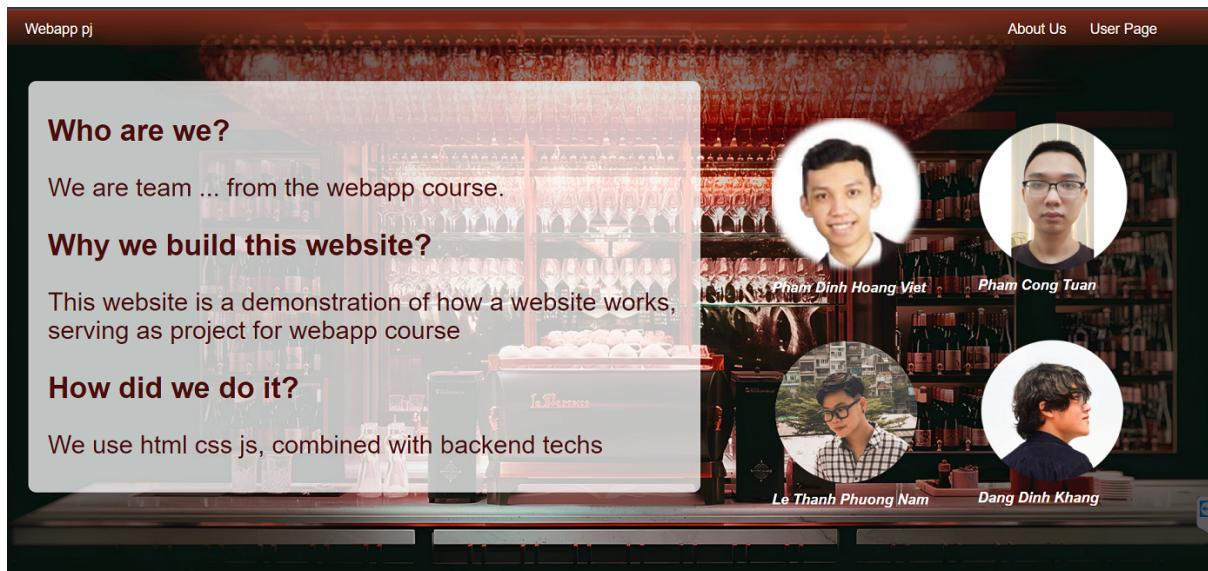
If the input is not in correct format, the browser will warn us.



*Figure 18. Wrong email format*

## 4.3. User function

This is our “About us” page, showing our team members and reason to build this website.



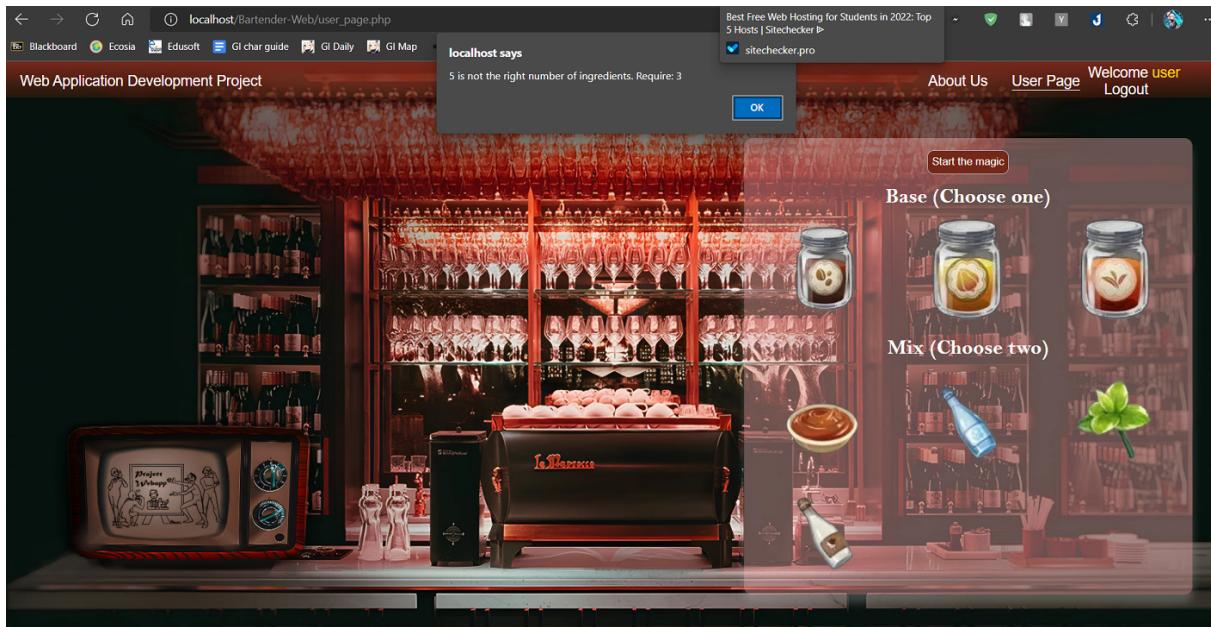
*Figure 19. About us Page*

This is the main page. Users can click on ingredients to make a drink. When a user clicks on an ingredient (black), there's a “click effect” and the image will become smaller. The result will be displayed on the left panel (red), along with the drink's name, recipe and shake animation.

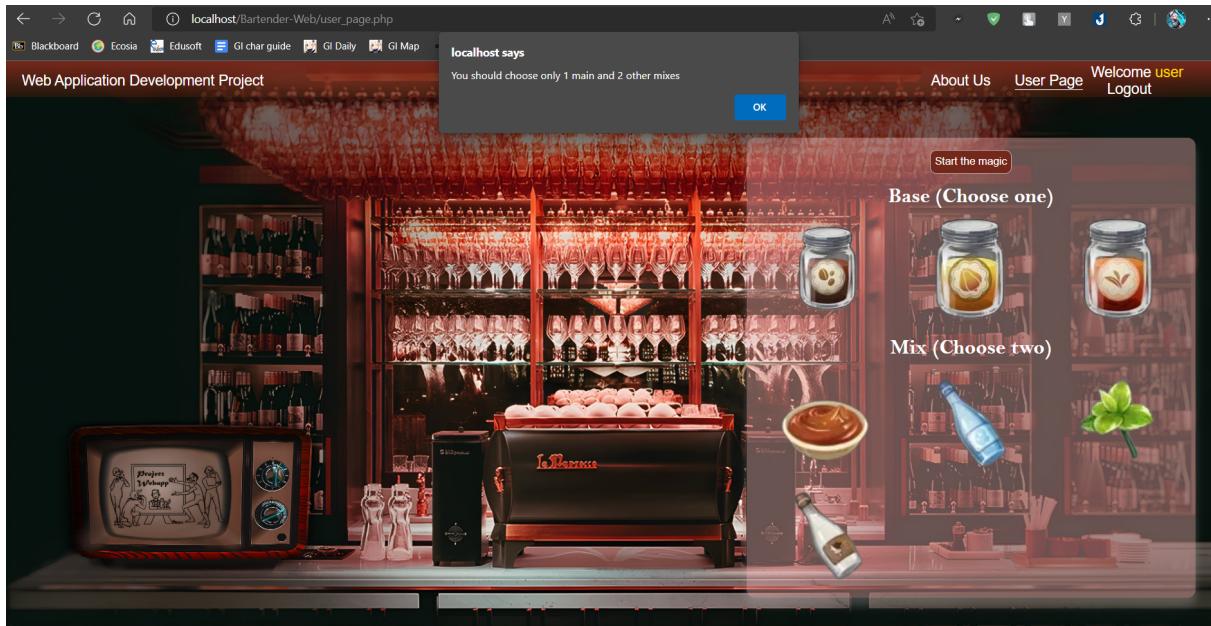


*Figure 20. Mix ingredients*

If the user chooses the wrong amount of ingredients, the chosen state will be resetted and the site will prompt the use.



*Figure 21. Wrong ingredient amount 1*



*Figure 22. Wrong ingredient amount 2*

## 4.4. Admin function

Admin can manage users.

The screenshot shows a web application interface for managing users. At the top, there's a navigation bar with 'Web Application Development Project' on the left and 'Welcome admin Logout' on the right. Below the navigation is a table listing users:

Name	Email	Password	Role	Type	Action
admin	admin@gmail.com	admin	admin		<a href="#">Edit</a> <a href="#">Delete</a>
user	user@gmail.com	1	user		<a href="#">Edit</a> <a href="#">Delete</a>
user10	user10@yahoo.com	zx	user		<a href="#">Edit</a> <a href="#">Delete</a>

Below the table is a yellow modal window titled 'ADD/UPDATE USER'. It contains four input fields: 'Enter your name', 'Enter your email', 'Enter your password', and a dropdown menu currently set to 'user'. At the bottom of the modal is a blue 'Add' button. The background of the page features a blurred image of a well-stocked bar.

Figure 23. Admin page

# Chapter 5: Test Plans

## 5.1. Feature to be tested

### 5.1.1. In scope

All of the features of the Beverage website that are included in the software requirements must be tested.

Module Name	Roles Involved	Description
Add User/Admin	Admin	An admin can only have one account with a single email address. An admin can create several user/admin accounts.
Login	User/Admin	A user and admin can only have one account with a single email address. They can use their username and password to access the Website.
Register	User	Users can create their account based on typing their username, email, user_type, password, and confirmed password.
Edit User	Admin	An admin can edit the information of several user/admin accounts.
Delete User	Admin	An admin can delete several user/admin accounts out of the database.
Beverage	User	A user can beverage many types of drinks based on specific materials.

*Table 15. Feature - in scope*

### 5.1.2. Out of scope

Legal concern: We are not legal consultants, and the severity of a fault is not determined by legal requirements, procedures, or standards.

Browser extensions, ad blockers, and virus scanners can cause issues such as blocking certain content or even executing apps.

The setup challenges have not progressed to the point where commerce, searching for more thorough recipes, or even the ability to safeguard accounts (OTP notifications), update, and forget passwords are all possible.

## 5.2. Pass/Fail Criteria

The criteria has been marked as passed because it meets the criteria. It is still regarded as a pass if it discovers some minor deviations from the intended outcome.

If the Test Case cannot execute due to an environment element, and the output does not match the expected output, the case is marked as failed.

Test case	Cases	Pass	Fail
Login	Enter valid username and password	Make sure to log into the correct account.	A user logs in to the incorrect account. The information is valid but cannot change to the user page.
	Enter invalid username and password	Notify the user if the account cannot be located or if the password is incorrect.	The user is unable to log in, however, the website is missing an alert message.
Edit User	Edit users' information	Change correct account information of each user Notify successful message	Not alert successful message. Cannot display user's information. Display the incorrect user's information. Change the information of another user.
Add users	Fill in the incorrect form.	Alert users input in correct form	Users enter incorrect information, the system does not notify them of the error and continues.

	Add an existing account.	Alert that user is existed	
	Fill in the correct information account.	Data stored in database	
<b>Delete User</b>	Delete users' information	1. Delete correct account information of each user	Cannot display user's information. Delete the information of another user.
<b>Register</b>	Fill in the incorrect form.	Alert users input in correct form	Users enter incorrect information, the system does not notify them of the error and continues.
	Add an existing account.	Alert users that user is existed	
	Fill in the correct information account.	Data stored in database	
<b>Beverage</b>	Choose correct materials	Display mixed drinks	Not display any drinks
	Choose incorrect materials	Alert users choose correct materials	Not alert user chosen message.

**Table 16. Pass/Fail criteria**

## 5.3. Approach

### 5.3.1. Process of testing

- Planning and Control
- Analysis and Design
- Implementation
- Evaluating exit criteria and Reporting
- Test Closure activities

### 5.3.2. Testing level

- Unit testing : A method of determining whether or not individual components of software are fulfilling their intended functions.
- Integration testing: Involves having testers test the data flow from one module to another.
- System testing: Identify the functional and non-functional requirements for testing.

- Acceptance testing: Determines whether or not a specification's requirements are met in the delivery.
- API testing: Involves putting all of the software's APIs to the test.

### 5.3.3. Roles and responsibilities

#	Role	Name	Responsibilities
01	Test Manager	Hoang Viet	Prepares test strategy
02	Software Tester	Cong Tuan	Execute various test scenarios.
03	Database Tester	Dinh Khang	Test database environment and assets are managed and maintained.
04	Developer Tester	Phuong Nam	Implement the Test Case.

*Table 17. Role and responsibilities*

### 5.3.4. Types of testing

Function testing

Test Objective	Application Navigation, Data Entry, Processing
Technique	Taking each use case, function, and scenario into action Using both valid and invalid data When invalid data is used, the warning messages are displayed.
Criteria	Plan Tests have been passed

*Table 18. Function testing*

Security Testing

Test Objective	Data Access Verification
Technique	Identify and list each user type, as well as the functions and data to which each type of user has access.
Criteria	For each known user type, the function/data is available The function runs as expected

*Table 19. Security testing*

## 5.4. Test materials

Hardware	Computers: 3 Computers
Software	MySQL and Visual Studio Code

*Table 20. Test materials*

## 5.5. Test cases

<b>Test Scenario</b>	Check Login Functionality	<b>Test Case Name</b>	Enter a valid username and password
<b>Designed by</b>	Hoang Viet	<b>Design Date</b>	April 18
<b>Executed by</b>	Phuong Nam	<b>Execution Date</b>	April 18
<b>Pre-Conditions</b>	1. User accounts already registered 2. User account has been authorized 3. The user's device is already connected to the internet when logging in	<b>Post-Condition (s)</b>	User successfully logged into the application
<b>Test Data</b>	<b>Username:</b> userhere <b>Password:</b> passis123		
<b>Test Step</b>	<b>Expected Result</b>	<b>Comments</b>	<b>Pass/Fail</b>
1. User access beverage website (login page).  2. User chooses the login method with the account.  3. User enters valid username, password and selects login command	1. Display login page  2. The system authenticates the login information successfully and allows the user to access the application		PASS

*Table 21. Test case 01*

<b>Test Scenario</b>	Check Login Functionality	<b>Test Case Name</b>	Enter an invalid username or password
<b>Designed by</b>	Hoang Viet	<b>Design Date</b>	April 18
<b>Executed by</b>	Phuong Nam	<b>Execution Date</b>	April 19
<b>Pre-Conditions</b>	1. User accounts did not registered 2. User account has been authorized 3. The user's device is already connected to the internet when logging in	<b>Post-Condition (s)</b>	User successfully logged into the application
<b>Test Data</b>	<b>Username:</b> userdayne <b>Password:</b> xxxxxx		
<b>Test Step</b>	<b>Expected Result</b>	<b>Comments</b>	<b>Pass/Fail</b>
1. User accesses Beverage website (login page). 2. User chooses the login method with the account. 3. User enters invalid username, password and selects login command	1. Display login page 2. The system validates that the login failed and displays a message " <i>incorrect email or password!</i> "		PASS

*Table 22. Test case 02*

<b>Test Scenario</b>	Check Add New User/Admin Functionality	<b>Test Case Name</b>	Add a valid user account
<b>Designed by</b>	Hoang Viet	<b>Design Date</b>	April 18
<b>Executed by</b>	Phuong Nam	<b>Execution Date</b>	April 19
<b>Pre-Conditions</b>	1. The admin's device is already connected to the internet. 2. The admin has logged into the system.	<b>Post-Condition (s)</b>	1. The admin click "Save" button
<b>Test Data</b>	Username: Nam Email: abcxyz@gmail.com Password: passla123 User_type: user		
<b>Test Step</b>	<b>Expected Result</b>	<b>Comments</b>	<b>Pass/Fail</b>
1. Admin accesses Beverage website. 2. Admin enters Username, Email, Password, User_type 4. Admin selects "Save" button	1. Display admin page 2. The system verifies information and creates an account. 3. The system will send successful message "Address Updated"		PASS

*Table 23. Test case 03*

<b>Test Scenario</b>	Check Register a New User Functionality	<b>Test Case Name</b>	Register a valid user account
<b>Designed by</b>	Hoang Viet	<b>Design Date</b>	April 18
<b>Executed by</b>	Phuong Nam	<b>Execution Date</b>	April 19
<b>Pre-Conditions</b>	1. The user's device is already connected to the internet. 2. The admin has logged into the system.	<b>Post-Condition (s)</b>	
<b>Test Data</b>	<b>Username:</b> Namb <b>Email:</b> abc@gmail.com <b>Password:</b> passla123 <b>User type:</b> user		
<b>Test Step</b>	<b>Expected Result</b>	<b>Comments</b>	<b>Pass/Fail</b>
1. User accesses Beverage website (login page).  2. User choose “Register Now” command  2. User enters Username, Email, Password, User_type  4. User selects “Register Now” button	1. Display login page.  2. Display register page.  3. The system verifies information and creates an account.  4. The system will display a login page.		PASS

**Table 24. Test case 04**

<b>Test Scenario</b>	Check Register a New User Functionality	<b>Test Case Name</b>	Register a existing user account
<b>Designed by</b>	Hoang Viet	<b>Design Date</b>	April 19
<b>Executed by</b>	Phuong Nam	<b>Execution Date</b>	April 20
<b>Pre-Conditions</b>	1. The user's device is already connected to the internet. 2. The admin has logged into the system.	<b>Post-Condition (s)</b>	1. User selects "Register Now" button
<b>Test Data</b>	<b>Username:</b> Namb <b>Email:</b> abc@gmail.com <b>Password:</b> passla123 <b>User type:</b> user		
<b>Test Step</b>	<b>Expected Result</b>	<b>Comments</b>	<b>Pass/Fail</b>
1. User accesses Beverage website (login page).  2. User choose "Register Now" command  3. User enters Username, Email, Password, User_type  4. User selects "Register Now" button	1. Display login page.  2. Display register page.  3. The system verifies information.  3. The system checks invalid information and displays a message "user already exists!"		PASS

*Table 25. Test case 05*

<b>Test Scenario</b>	Check Add New User/Admin Functionality	<b>Test Case Name</b>	Add a existing user account
<b>Designed by</b>	Hoang Viet	<b>Design Date</b>	April 19
<b>Executed by</b>	Phuong Nam	<b>Execution Date</b>	April 20
<b>Pre-Conditions</b>	1. The admin's device is already connected to the internet. 2. The admin has logged into the system.	<b>Post-Condition(s)</b>	The admin click "Save" button
<b>Test Data</b>	<b>Username:</b> Nam <b>Email:</b> abcxzy@gmail.com <b>Password:</b> passla123 <b>User_type:</b> user		
<b>Test Step</b>	<b>Expected Result</b>	<b>Comments</b>	<b>Pass/Fail</b>
1. Admin accesses Beverage website. 2. Admin enters Username, Email, Password, User_type 3. Admin selects "Save" button	1. Display admin page 2. The system verifies information and creates an account. 3. The system checks invalid information and displays a message "User already exists!"		PASS

*Table 26. Test case 06*

<b>Test Scenario</b>	Check Edit an Account Information Functionality	<b>Test Case Name</b>	Edit an valid Account Information
<b>Designed by</b>	Hoang Viet	<b>Design Date</b>	April 20
<b>Executed by</b>	Phuong Nam	<b>Execution Date</b>	April 20
<b>Pre-Conditions</b>	1. Accounts already registered 2. The admin's device is already connected to the internet. 3. The admin has logged into the system.	<b>Post-Condition(s)</b>	1. The admin click "Update" button
<b>Test Data</b>	<b>Username:</b> Nam <b>Email:</b> abcxyz@gmail.com <b>Password:</b> passla123 <b>User_type:</b> user		
<b>Test Step</b>	<b>Expected Result</b>	<b>Comments</b>	<b>Pass/Fail</b>
1. Admin accesses Beverage website. 2. Admin clicks the "edit" command. 3. Admin enters Username, Email, Password, User_type 4. Admin selects "Update" button	1. Display admin page 2. Display account information 3. The system verifies information. 3. The system will send successful message "User Updated"		PASS

**Table 27. Test case 07**

<b>Test Scenario</b>	Check Edit an Account Information Functionality	<b>Test Case Name</b>	Edit a invalid Account Information
<b>Designed by</b>	Hoang Viet	<b>Design Date</b>	April 20
<b>Executed by</b>	Phuong Nam	<b>Execution Date</b>	April 21
<b>Pre-Conditions</b>	1. Accounts already registered 2. The admin's device is already connected to the internet. 3. The admin has logged into the system.	<b>Post-Condition (s)</b>	1. The admin click "Update" button
<b>Test Data</b>	<b>Username:</b> admin <b>Email:</b> admin@gmail.com <b>Password:</b> passla123 <b>User_type:</b> user		
<b>Test Step</b>	<b>Expected Result</b>	<b>Comments</b>	<b>Pass/Fail</b>
1. Admin accesses Beverage website. 2. Admin clicks the "edit" command. 3. Admin enters Username, Email, Password, User_type 4. Admin selects "Update" button	1. Display admin page 2. Display account information 3. The system verifies information. 4. The system will send an unsuccessful message "User already exists!"		PASS

*Table 28. Test case 08*

<b>Test Scenario</b>	Check Delete an Account Functionality	<b>Test Case Name</b>	Delete an Account Information
<b>Designed by</b>	Hoang Viet	<b>Design Date</b>	April 20
<b>Executed by</b>	Phuong Nam	<b>Execution Date</b>	April 21
<b>Pre-Conditions</b>	1. Accounts already registered 2. The admin's device is already connected to the internet. 3. The admin has logged into the system.	<b>Post-Condition (s)</b>	1. The admin click "Delete" command
<b>Test Data</b>			
<b>Test Step</b>	<b>Expected Result</b>	<b>Comments</b>	<b>Pass/Fail</b>
1. Admin accesses Beverage website.  2. Admin clicks the "delete" command.  3. Admin checks Username, Email, Password, User_type  4. Admin selects "Delete" button	1. Display admin page  2. The system deletes that account out of the database.  4. The system will send successful message "User Deleted"		PASS

*Table 29. Test case 09*

<b>Test Scenario</b>	Check Beverage Functionality	<b>Test Case Name</b>	Beverage a valid drinks
<b>Designed by</b>	Hoang Viet	<b>Design Date</b>	April 22
<b>Executed by</b>	Phuong Nam	<b>Execution Date</b>	April 23
<b>Pre-Conditions</b>	1. User logged into the system.	<b>Post-Condition (s)</b>	1. User clicks “Start the magic” button
<b>Test Data</b>			
<b>Test Step</b>	<b>Expected Result</b>	<b>Comments</b>	<b>Pass/Fail</b>
1. User accesses Beverage website. 2. User clicks “Start” button 3. User choose 1 main material and 2 flavors materials 4. User clicks “Start the magic” button	1. Display user page 2. Display beverage function. 3. The system store the chosen data materials 4. The system display the mixed drinks		PASS

*Table 30. Test case 10*

<b>Test Scenario</b>	Check Beverage Functionality	<b>Test Case Name</b>	Beverage a invalid drinks
<b>Designed by</b>	Hoang Viet	<b>Design Date</b>	April 22
<b>Executed by</b>	Phuong Nam	<b>Execution Date</b>	April 23
<b>Pre-Conditions</b>	1. User logged into the system.	<b>Post-Condition(s)</b>	1. User clicks “Start the magic” button
<b>Test Data</b>			
<b>Test Step</b>	<b>Expected Result</b>	<b>Comments</b>	<b>Pass/Fail</b>
1. User accesses Beverage website.	1. Display user page 2. Display Beverage function.		
2. User clicks “Start” button	3. The system store the chosen data materials		
3a. User choose 2 main material and 1 flavors materials	4a. The system alerts a message “You should choose only 1 main and 2 other mixes” and have to choose “OK” to try again.		PASS
3b. User choose 1 main material and 3 flavors materials	4b. The system alerts a message “4 is not the right number of ingredients. Require: 3” and have to choose “OK” to Beverage again.		
4. User clicks “Start the magic” button			

**Table 31. Test case 11**

## 5.6. Test Plans Schedule

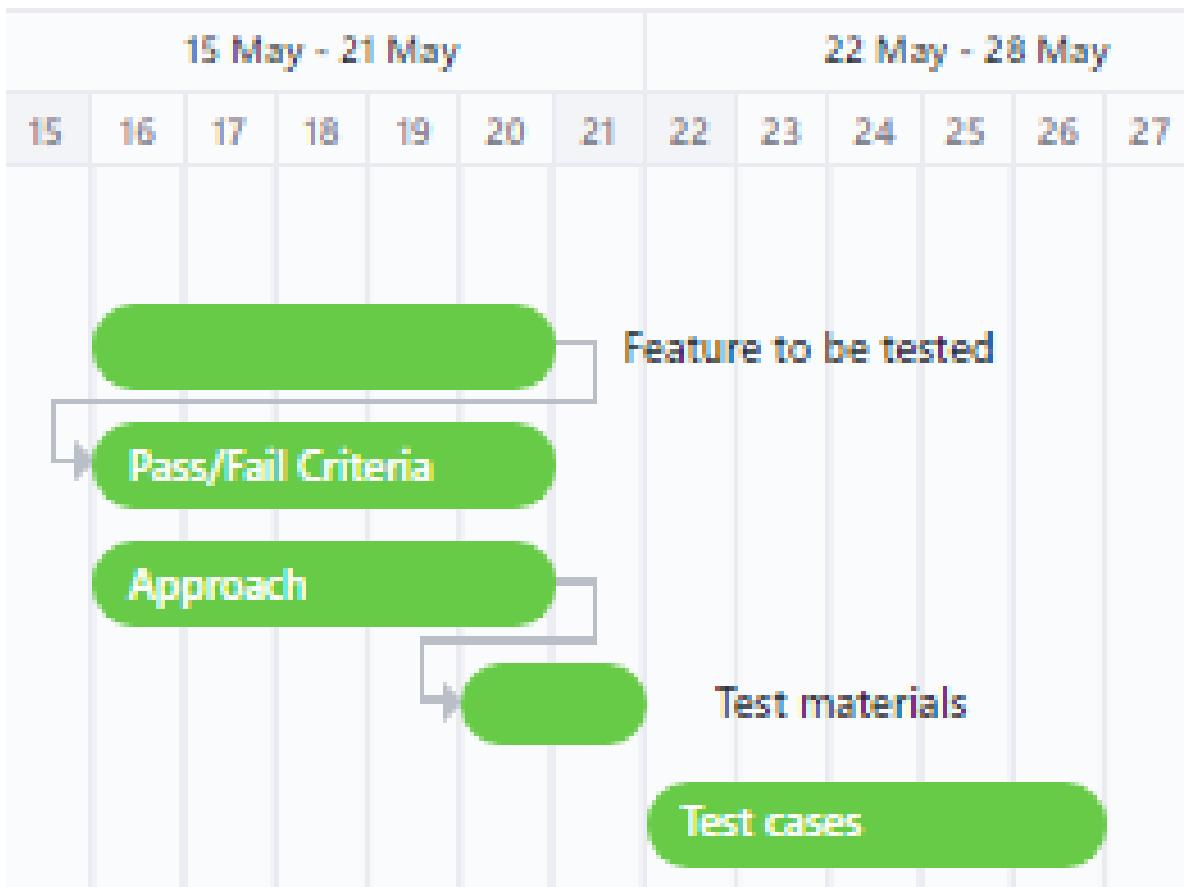


Figure 24. Test plan

# Chapter 6: Glossary

## References

<https://github.com/TomPham204/Bartender-Web>  
<https://en.wikipedia.org/wiki/SQL>  
<https://en.wikipedia.org/wiki/JavaScript>  
[https://en.wikipedia.org/wiki/Microsoft\\_Teams](https://en.wikipedia.org/wiki/Microsoft_Teams)  
<https://en.wikipedia.org/wiki/PHP>  
[https://en.wikipedia.org/wiki/Entity%2C%20relationship\\_model](https://en.wikipedia.org/wiki/Entity%2C%20relationship_model)  
[https://en.wikipedia.org/wiki/Graphical\\_user\\_interface](https://en.wikipedia.org/wiki/Graphical_user_interface)  
<https://www.php.net>  
[http://web.cse.ohio-state.edu/~bair.41/616/Project/Example\\_Document/Req\\_Doc\\_Example.html#Func](http://web.cse.ohio-state.edu/~bair.41/616/Project/Example_Document/Req_Doc_Example.html#Func)  
<https://app.clickup.com/14289914/v/g/dm2zu-187>  
<https://www.youtube.com/watch?v=8MgpE2DTTKA>  
<https://www.youtube.com/watch?v=lvYnfMOUQJY>