**VIETNAM NATIONAL UNIVERSITY OF HOCHIMINH CITY THE INTERNATIONAL UNIVERSITY**

**SCHOOL OF COMPUTER SCIENCE AND ENGINEERING**

Logo

Description automatically generated

**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 | Cống Tuấn | Hoàng Việt | Đình Khang |
| --- | --- | --- | --- | --- |
| **Percent** | ¼ | ¼ | ¼ | ¼ |

**GITHUB Link :** [**https://github.com/TomPham204/Bartender-Web**](https://github.com/TomPham204/Bartender-Web)

# 

# **TABLE OF CONTENTS**

[**TABLE OF CONTENTS 3**](#_heading=h.30j0zll)

[**List of Tables 5**](#_heading=h.1fob9te)

[**List of Figures 8**](#_heading=h.3znysh7)

[**Chapter 1: Project Description 10**](#_heading=h.2et92p0)

[**1.1.**](#_heading=h.tyjcwt) **Project Overview 10**

[**1.2.**](#_heading=h.3dy6vkm) **The Purpose of the Project 10**

[***1.2.1.***](#_heading=h.1t3h5sf) ***Abstract*** *10*

[***1.2.2.***](#_heading=h.4d34og8) ***Project Goals*** *11*

[**1.3.**](#_heading=h.2s8eyo1) **Scope of Work 12**

[***1.3.1.***](#_heading=h.17dp8vu) ***Included*** *12*

[***1.3.2.***](#_heading=h.3rdcrjn) ***Excluded*** *13*

[**1.4.**](#_heading=h.26in1rg) **Organization 13**

[***1.4.1.***](#_heading=h.lnxbz9) ***Organizational Boundaries and Interfaces*** *13*

[***1.4.2.***](#_heading=h.1ksv4uv) ***Project Organization*** *14*

[**1.5.**](#_heading=h.2jxsxqh) **Resource Requirements 14**

[***1.5.1.***](#_heading=h.z337ya) ***Hardware*** *14*

[***1.5.2.***](#_heading=h.1y810tw) ***Software*** *16*

[***1.5.3.***](#_heading=h.2xcytpi) ***Human*** *17*

[**1.6.**](#_heading=h.3whwml4) **Technology 19**

[***1.6.1.***](#_heading=h.2bn6wsx) ***ReactJS*** *19*

[***1.6.2.***](#_heading=h.3as4poj) ***ExpressJS*** *20*

[***1.6.3.***](#_heading=h.49x2ik5) ***MySQL*** *20*

[**1.7.**](#_heading=h.147n2zr) **Schedule 22**

[***1.7.1.***](#_heading=h.3o7alnk) ***Gantt Chart*** *22*

[***1.7.2.***](#_heading=h.ihv636) ***Work Breakdown Structure*** *23*

[***1.7.3.***](#_heading=h.1hmsyys) ***Schedule and milestone*** *24*

[***1.7.4.***](#_heading=h.41mghml) ***Process Development*** *26*

[**1.8.**](#_heading=h.2grqrue) **Risk Management 27**

[**1.9.**](#_heading=h.vx1227) **Delivery Plan 29**

[***1.9.1.***](#_heading=h.3fwokq0) ***Sprint 1 backlog (07/09 - 10/09)*** *29*

[***1.9.2.***](#_heading=h.3tbugp1) ***Sprint 2 backlog (11/09 - 02/10)*** *30*

[***1.9.3.***](#_heading=h.28h4qwu) ***Sprint 3 backlog (03/10 - 24/10)*** *32*

[***1.9.4.***](#_heading=h.nmf14n) ***Sprint 4 backlog (15/11 – 29/11)*** *33*

[***1.9.5.***](#_heading=h.37m2jsg) ***Sprint 5 backlog (30/11 - 07/12)*** *33*

[**1.10.**](#_heading=h.1v1yuxt) **Security Aspect 34**

[**1.11.**](#_heading=h.4f1mdlm) **Naming Conventions and Definitions 35**

[**1.12.**](#_heading=h.19c6y18) **Conclusion 36**

[**Chapter 2: Requirements 37**](#_heading=h.1mrcu09)

[**2.1.**](#_heading=h.46r0co2) **Product Use Cases 37**

[**2.2.**](#_heading=h.2lwamvv) **Functional & Non-functional Requirement 48**

[**2.3.**](#_heading=h.111kx3o) **Data Requirements 49**

[**2.4.**](#_heading=h.3l18frh) **Performance Requirements 51**

[**2.5.**](#_heading=h.206ipza) **Dependability Requirements 51**

[**2.6.**](#_heading=h.4k668n3) **Security Requirements 52**

[**2.7.**](#_heading=h.2zbgiuw) **Usability Requirements 53**

[**Chapter 3: Design 55**](#_heading=h.1egqt2p)

[**3.1.**](#_heading=h.3ygebqi) **System Architecture 55**

[**3.2.**](#_heading=h.2dlolyb) **Database Structure 56**

[**3.3.**](#_heading=h.sqyw64) **Entity–relationship Diagram 57**

[**3.4.**](#_heading=h.3cqmetx) **Sequence Diagram 58**

[**3.5.**](#_heading=h.1rvwp1q) **Class Diagrams 69**

[**Chapter 4: Implementation 70**](#_heading=h.4bvk7pj)

[**4.1.**](#_heading=h.2r0uhxc) **Login function 70**

[**4.2.**](#_heading=h.1664s55) **User function 72**

[***4.2.1.***](#_heading=h.3q5sasy) ***Transaction*** *72*

[***4.2.2.***](#_heading=h.25b2l0r) ***Lock account*** *76*

[***4.2.3.***](#_heading=h.kgcv8k) ***Check Balance*** *77*

[***4.2.4.***](#_heading=h.34g0dwd) ***Change Password*** *77*

[***4.2.5.***](#_heading=h.1jlao46) ***Create New Card*** *79*

[**Chapter 5: Test Plans 84**](#_heading=h.43ky6rz)

[**5.1.**](#_heading=h.2iq8gzs) **Feature to be tested 84**

[***5.1.1.***](#_heading=h.xvir7l) ***In scope:*** *84*

[***5.1.2.***](#_heading=h.3hv69ve) ***Out of scope:*** *85*

[**5.2.**](#_heading=h.1x0gk37) **Pass/Fail Criteria 85**

[**5.3.**](#_heading=h.4h042r0) **Approach 92**

[***5.3.1.***](#_heading=h.2w5ecyt) ***Process of testing*** *92*

[***5.3.2.***](#_heading=h.1baon6m) ***Testing level*** *92*

[***5.3.3.***](#_heading=h.3vac5uf) ***Roles and responsibilities*** *92*

[***5.3.4.***](#_heading=h.2afmg28) ***Types of testing*** *93*

[**5.4.**](#_heading=h.pkwqa1) **Suspension and resumption 94**

[**5.5.**](#_heading=h.39kk8xu) **Test materials (hardware & software requirement) 94**

[**5.6.**](#_heading=h.1opuj5n) **Test cases 95**

[**5.7.**](#_heading=h.48pi1tg) **Testing Schedule 128**

[**Chapter 6: Glossary 129**](#_heading=h.2nusc19)

[**References 129**](#_heading=h.1302m92)

# **List of Tables**

[**Table 1. Supplier** 12](#_heading=h.35nkun2)

[**Table 2. Project Organization** 12](#_heading=h.44sinio)

[**Table 3. Hardware resource** 13](#_heading=h.3j2qqm3)

[**Table 4. Software resource** 15](#_heading=h.4i7ojhp)

[**Table 5. Human Resource** 17](#_heading=h.1ci93xb)

[**Table 6. Product Backlog** 24](#_heading=h.1hmsyys)

[**Table 7. Risk Management** 26](#_heading=h.2grqrue)

[**Table 8. Sprint 1 Backlog** 27](#_heading=h.3mzq4wv)

[**Table 9. Sprint 2 Backlog** 29](#_heading=h.2250f4o)

[**Table 10. Sprint 3 Backlog** 30](#_heading=h.haapch)

[**Table 11. Sprint 4 Backlog** 31](#_heading=h.319y80a)

[**Table 12. Sprint 5 Backlog** 31](#_heading=h.3fwokq0)

[**Table 13. Naming Convention and Definitions** 33](#_heading=h.2u6wntf)

[**Table 14. Use case 1** 37](#_heading=h.1gf8i83)

[**Table 15. Use case 1.1** 37](#_heading=h.40ew0vw)

[**Table 16. Use case 2** 38](#_heading=h.2fk6b3p)

[**Table 17. Use case 3** 39](#_heading=h.upglbi)

[**Table 18. Use case 4** 40](#_heading=h.3ep43zb)

[**Table 19. Use case 5** 41](#_heading=h.1tuee74)

[**Table 20. Use case 6** 42](#_heading=h.4du1wux)

[**Table 21. Use case 7** 43](#_heading=h.2szc72q)

[**Table 22. Use case 8** 43](#_heading=h.184mhaj)

[**Table 23. Use case 8.1** 44](#_heading=h.3s49zyc)

[**Table 24. Use case 8.2** 45](#_heading=h.279ka65)

[**Table 25. Functional and Non-functional** 47](#_heading=h.meukdy)

[**Table 26. UML Class Diagram** 48](#_heading=h.36ei31r)

[**Table 27. Feature - in scope** 79](#_heading=h.1ljsd9k)

[**Table 28. Pass/Fail criteria** 85](#_heading=h.45jfvxd)

[**Table 29. Role and responsibilities** 86](#_heading=h.2koq656)

[**Table 30. Function testing** 87](#_heading=h.zu0gcz)

[**Table 31. Security Testing** 87](#_heading=h.3jtnz0s)

[**Table 32. User Interface Testing** 87](#_heading=h.1yyy98l)

[**Table 33. Suspension and resumption** 88](#_heading=h.4iylrwe)

[**Table 34. Test materials** 88](#_heading=h.2y3w247)

[**Table 35. Test case 1** 89](#_heading=h.1d96cc0)

[**Table 36. Test case 2** 90](#_heading=h.3x8tuzt)

[**Table 37. Test case 3** 91](#_heading=h.2ce457m)

[**Table 38. Test case 4** 92](#_heading=h.rjefff)

[**Table 39. Test case 5** 93](#_heading=h.3bj1y38)

[**Table 40. Test case 6** 94](#_heading=h.1qoc8b1)

[**Table 41. Test case 7** 95](#_heading=h.4anzqyu)

[**Table 42. Test case 8** 96](#_heading=h.2pta16n)

[**Table 43. Test case 9** 97](#_heading=h.14ykbeg)

[**Table 44. Test case 10** 98](#_heading=h.3oy7u29)

[**Table 45. Test case 11** 99](#_heading=h.243i4a2)

[**Table 46. Test case 12** 100](#_heading=h.j8sehv)

[**Table 47. Test case 13** 101](#_heading=h.338fx5o)

[**Table 48. Test case 14** 102](#_heading=h.1idq7dh)

[**Table 49. Test case 15** 103](#_heading=h.42ddq1a)

[**Table 50. Test case 16** 103](#_heading=h.2hio093)

[**Table 51. Test case 17** 104](#_heading=h.wnyagw)

[**Table 52. Test case 18** 105](#_heading=h.3gnlt4p)

[**Table 53. Test case 19** 106](#_heading=h.1vsw3ci)

[**Table 54. Test case 20** 107](#_heading=h.4fsjm0b)

[**Table 55. Test case 21** 108](#_heading=h.2uxtw84)

[**Table 56. Test case 22** 109](#_heading=h.1a346fx)

[**Table 57. Test case 23** 110](#_heading=h.3u2rp3q)

[**Table 58. Test case 24** 111](#_heading=h.2981zbj)

[**Table 59. Test case 25** 112](#_heading=h.odc9jc)

[**Table 60. Test case 26** 112](#_heading=h.38czs75)

[**Table 61. Test case 27** 114](#_heading=h.1nia2ey)

[**Table 62. Test case 28** 115](#_heading=h.47hxl2r)

[**Table 63. Test case 29** 116](#_heading=h.2mn7vak)

[**Table 64. Test case 30** 117](#_heading=h.11si5id)

[**Table 65. Test case 31** 118](#_heading=h.3ls5o66)

[**Table 66. Test case 32** 119](#_heading=h.20xfydz)

[**Table 67. Test case 33** 120](#_heading=h.4kx3h1s)

[**Table 68. Test case 34** 121](#_heading=h.302dr9l)

[**Table 69. Test case 35** 122](#_heading=h.1f7o1he)

[**Table 70. Test case 36** 123](#_heading=h.3z7bk57)

[**Table 71. Test case 37** 124](#_heading=h.2eclud0)

[**Table 72. Test case 38** 125](#_heading=h.thw4kt)

[**Table 73. Test case 39** 126](#_heading=h.3dhjn8m)

[**Table 74. Test case 40** 127](#_heading=h.1smtxgf)

[**Table 75. Test case 41** 128](#_heading=h.4cmhg48)

[**Table 76. Test case 42** 129](#_heading=h.2rrrqc1)

**List of Figures**

[**Figure 1. ReactJS** 18](#_heading=h.qsh70q)

[**Figure 2. ExpressJS** 19](#_heading=h.1pxezwc)

[**Figure 3. MySQL Database** 19](#_heading=h.2p2csry)

[**Figure 4. Gantt Chart** 21](#_heading=h.23ckvvd)

[**Figure 5. Work breakdown structure** 22](#_heading=h.32hioqz)

[**Figure 6. Use Case Diagram** 36](#_heading=h.16x20ju)

[**Figure 7. System architecture** 54](#_heading=h.3qwpj7n)

[**Figure 8. Database Structure** 55](#_heading=h.261ztfg)

[**Figure 9. ERD** 56](#_heading=h.l7a3n9)

[**Figure 10. Sequence diagram - Login & Logout** 57](#_heading=h.356xmb2)

[**Figure 11. Sequence Diagram - Forget Password** 58](#_heading=h.1kc7wiv)

[**Figure 12. Sequence Diagram - Payment** 59](#_heading=h.44bvf6o)

[**Figure 13. Sequence Diagram - Change Password** 60](#_heading=h.2jh5peh)

[**Figure 14. Sequence diagram - Check balance** 61](#_heading=h.ymfzma)

[**Figure 15. Sequence Diagram - Create New Card** 62](#_heading=h.3im3ia3)

[**Figure 16. Sequence Diagram - Lock account** 63](#_heading=h.1xrdshw)

[**Figure 17. Sequence Diagram - Add new users** 64](#_heading=h.4hr1b5p)

[**Figure 18. Sequence Diagram - Manage user** 65](#_heading=h.2wwbldi)

[**Figure 19. Sequence Diagram - Manage Login History** 66](#_heading=h.1c1lvlb)

[**Figure 20. Sequence Diagram - Manage Transaction** 67](#_heading=h.3w19e94)

[**Figure 21. Class Diagram** 68](#_heading=h.2b6jogx)

[**Figure 22. Login interface** 69](#_heading=h.qbtyoq)

[**Figure 23. IU Bank Homepage** 70](#_heading=h.3abhhcj)

[**Figure 24. Forgot password interface** 70](#_heading=h.1pgrrkc)

[**Figure 25. Homepage with Internal transfer button** 71](#_heading=h.49gfa85)

[**Figure 26. Internal transfer interface** 72](#_heading=h.2olpkfy)

[**Figure 27. Homepage with external transfer button** 73](#_heading=h.13qzunr)

[**Figure 28. External transfer interface** 73](#_heading=h.3nqndbk)

[**Figure 29. Balance Information** 76](#_heading=h.22vxnjd)

[**Figure 30. Setting button** 76](#_heading=h.i17xr6)

[**Figure 31. Verify code interface** 77](#_heading=h.320vgez)

[**Figure 32. Change password interface** 77](#_heading=h.1h65qms)

[**Figure 33. Homepage with Create New Card button** 78](#_heading=h.415t9al)

[**Figure 34. Create new card interface** 79](#_heading=h.2gb3jie)

[**Figure 35. Admin homepage** 80](#_heading=h.vgdtq7)

[**Figure 36. Transaction History interface** 81](#_heading=h.3fg1ce0)

[**Figure 37. Login history interface** 81](#_heading=h.1ulbmlt)

[**Figure 38. Create new account interface** 82](#_heading=h.4ekz59m)

[**Figure 39. Testing schedule** 127](#_heading=h.2tq9fhf)

**Chapter 1: Project Description**

## **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.

## **The Purpose of the Project**

### **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.

### **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.

## **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.

## **Organization**

After receiving the project, my team members held a meeting on MsTeams 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.

## **Resource Requirements**

### **Project Organization**

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

***Table 1. Project Organization***

## **Resource Requirements**

### **Hardware**

| **Name** | **Item** | **Description** | **Purpose** |
| --- | --- | --- | --- |
| Hoàng Việt | Laptop Dell Inspiron 14 5402 Window | OS: Windows 10 Home  CPU: Intel Core i5-1135G7  Graphics: Intel Iris Xe  RAM: 16GB Onboard LPDDR4X 2666MHz  Storage: 1TB SSD M.2 NVMe  Screen: Full HD 14" IPS (1920 x 1080) | Implement Code |
| Đình 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 | Desktop Computer | OS: Windows 10 Pro  CPU: R5 4600U  RAM: 16GB ram bus 1333  Storage: 445 GB SSD  Screen: 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 2. Hardware resource***

### **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 |
| **PHP** | A web development oriented general-purpose scripting language. | Implement code | Free | Free | DONE |
| **XAMPP** | 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 |
| **ClickUp** | Simplify work and get more done. Plan, track, and manage any type of work with project management | Manage Project | Free | Free | DONE |
| **Github** | Web for sharing and uploading code | Managing code | Free | Free | DONE |
| **Total Cost:** | | | | | 0 |

***Table 4. Software resource***

### **Human**

| **Name** | **Organization** | **Role** | **Required skills** | **Email** |
| --- | --- | --- | --- | --- |
| Hoàng Việt | International University | **Project Manager, Tester** | **Technical skill:**  Basic understanding of OOP, OOAD, programming languages, and database structure.  Knowledge of the development process.  Familiar with project management methodologies, software  Risk management  **Soft skills:**  Leadership Communication skills  Problem-solving  Time management | ITITIU19064@student.hcmiu.edu.vn |
| Phương Nam | International University | **Front-end Developer, Back-end Developer** | **Technical skill:**  Basic understanding of OOP, OOAD, programming languages, and database structure.  Deep knowledge of Back-end Programming Languages & Data Structures and Algorithms  Familiar with databases, API, and servers  Knowledge of HTML, CSS, JS  **Soft skills:**  Collaboration Communication skills  Problem-solving  Time management | ITITWE19025@student.hcmiu.edu.vn |
| Công Tuấn | International University | **Front-end Developer, Back-end Developer** | **Technical skill:**  Basic understanding of OOP, OOAD, programming languages, and database structure.  Deep knowledge of Back-end Programming Languages & Data Structures and Algorithms  Familiar with databases, API, and servers  Knowledge of HTML, CSS, and JS  **Soft skills:**  Collaboration Communication skills  Problem-solving  Time management | ITITIU19060@student.hcmiu.edu.vn |
| Đình Khang | International University | **Analysis Design** | **Technical skill:**  Basic understanding of OOP, OOAD, programming languages.  Knowledge of prototyping, mockup, and wireframing.  Proficient in using visual design software (Photoshop, AI, Xd)  **Soft skills:**  Collaboration, Communication skills  Creativity  Time management | ITITIU18055@student.hcmiu.edu.vn |

***Table 5. Human Resource***

## **Technology**

### **PHP**

### 

***Figure 1. 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.

### **MySQL**

### 

***Figure 2. MySQL Database***

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.

**1.7.3. Javascript**

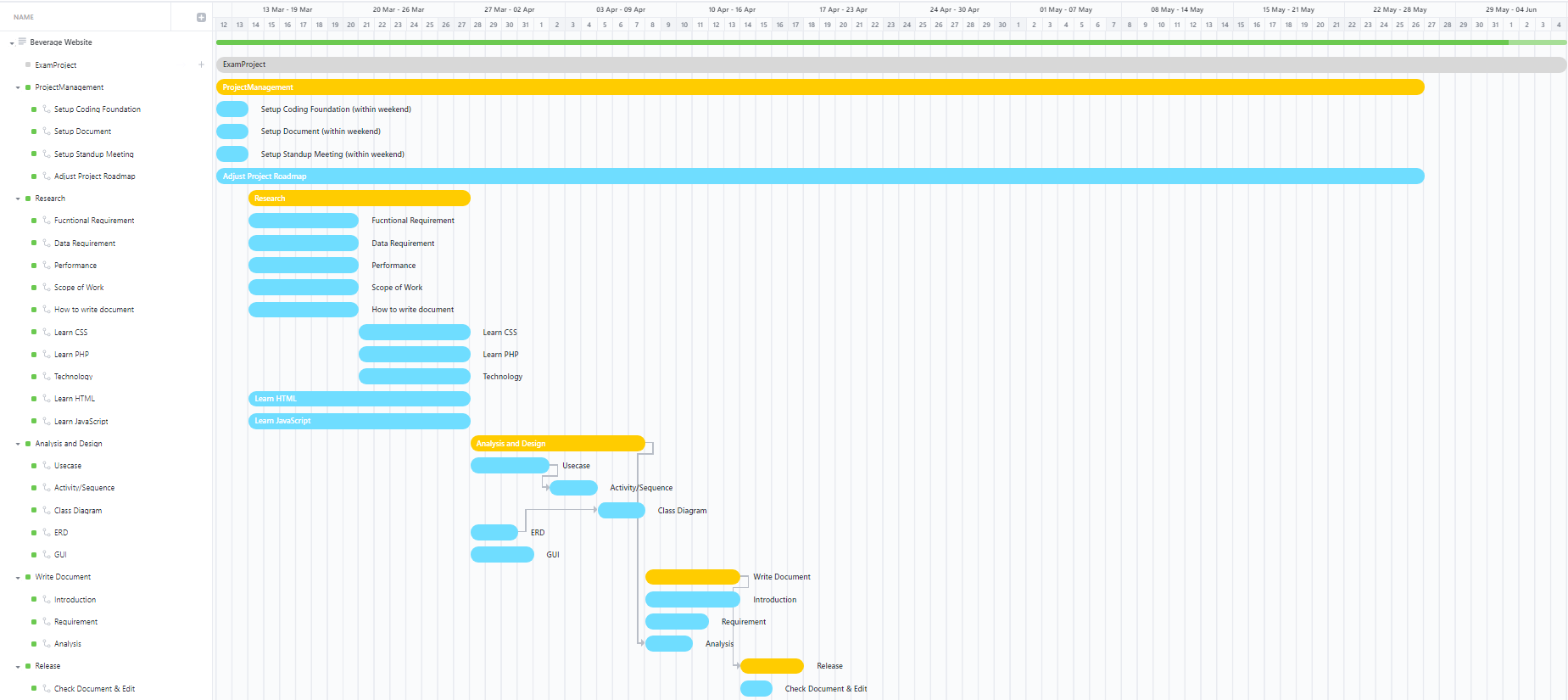


***Figure 3. Javascript***

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).

## **Schedule**

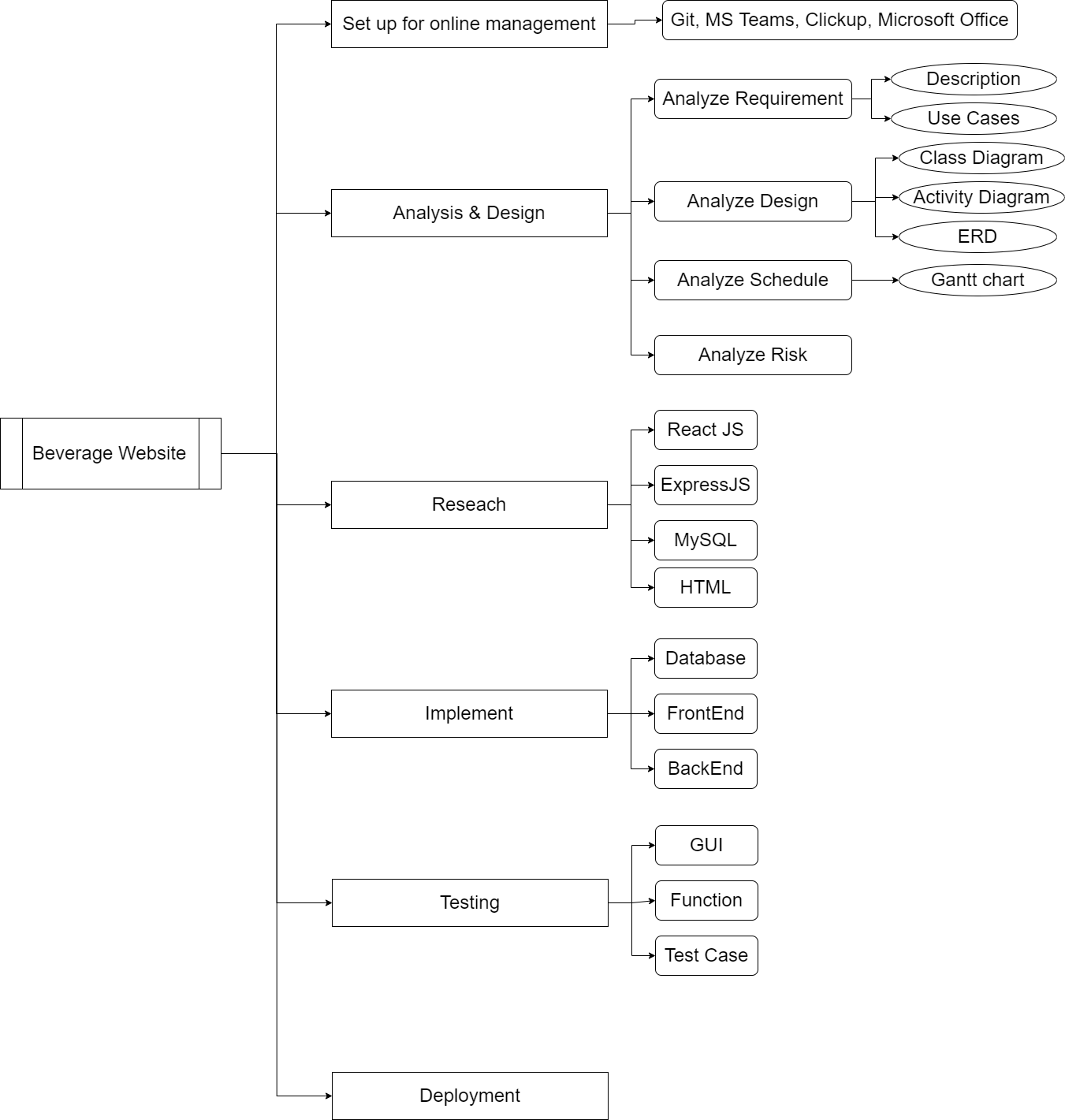
### **Gantt Chart**



### 

***Figure 4. Gantt Chart***

### **Work Breakdown Structure**

******

***Figure 5. Work breakdown structure***

### **Schedule and milestone**

| ***#*** | ***Project / Epic / Task*** | ***Type*** | ***Assignee*** | ***Priority*** | ***Status*** | ***Start*** | ***Finish*** | ***Days*** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | ***ExamProject*** | ***Project*** |  |  |  | ***12/3/2022*** | ***4/6/2022*** | ***84d*** |
| ***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*** | ***Đình Khang*** | ***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*** | ***Đình 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*** | ***Đình Khang*** | ***Low*** | ***Done*** | ***21/3/2022*** | ***27/3/2022*** | ***6 d*** |
| ***2.7*** | ***Learn PHP*** | ***Story*** | ***Phương Nam*** | ***Low*** | ***Done*** | ***21/3/2022*** | ***27/3/2022*** | ***6 d*** |
| ***2.8*** | ***Technology*** | ***Story*** | ***Phương Nam*** | ***Low*** | ***Done*** | ***21/3/2022*** | ***27/3/2022*** | ***6 d*** |
| ***2.9*** | ***Learn HTML*** | ***Story*** | ***Công Tuấn*** | ***Lowest*** | ***Done*** | ***14/3/2022*** | ***27/3/2022*** | ***12 d*** |
| ***2.10*** | ***Learn Javascript*** | ***Story*** | ***Công Tuấn*** | ***Lowest*** | ***Done*** | ***14/3/2022*** | ***27/3/2022*** | ***12 d*** |
| ***3*** | ***Analysis and Design*** | ***Epic*** |  | ***High*** | ***Done*** | ***28/3/2022*** | ***7/4/2022*** | ***10 d*** |
| ***3.1*** | ***Usecase*** | ***Story*** | ***Hoàng Việt*** | ***High*** | ***Done*** | ***28/3/2022*** | ***1/4/2022*** | ***4 d*** |
| ***3.2*** | ***Activity/ Sequence*** | ***Story*** | ***Hoàng Việt*** | ***High*** | ***Done*** | ***2/4/2022*** | ***4/4/2022*** | ***2 d*** |
| ***3.3*** | ***Class Diagram*** | ***Story*** | ***Đình Khang*** | ***Medium*** | ***Done*** | ***5/4/2022*** | ***7/4/2022*** | ***2 d*** |
| ***3.4*** | ***ERD*** | ***Story*** | ***Đình Khang*** | ***Medium*** | ***Done*** | ***28/3/2022*** | ***30/3/2022*** | ***2 d*** |
| ***3.5*** | ***GUI*** | ***Story*** | ***Công Tuấn*** | ***Medium*** | ***Done*** | ***28/3/2022*** | ***31/3/2022*** | ***3 d*** |
| ***4*** | ***Write Document*** | ***Epic*** |  | ***Medium*** | ***Done*** | ***8/4/2022*** | ***13/4/2022*** | ***5d*** |
| ***4.1*** | ***Introduction*** | ***Story*** | ***Phương Nam*** | ***Medium*** | ***Done*** | ***8/4/2022*** | ***13/4/2022*** | ***5d*** |
| ***4.1.1*** | ***Project Overview*** | ***Subtask*** |  | ***Medium*** | ***Done*** | ***8/4/2022*** | ***13/4/2022*** | ***5d*** |
| ***4.1.2*** | ***Purpose of Project*** | ***Subtask*** |  | ***Medium*** | ***Done*** | ***8/4/2022*** | ***13/4/2022*** | ***5d*** |
| ***4.1.3*** | ***Scope of Work*** | ***Subtask*** |  | ***Medium*** | ***Done*** | ***8/4/2022*** | ***13/4/2022*** | ***5 d*** |
| ***4.1.4*** | ***Organization*** | ***Subtask*** |  | ***Medium*** | ***Done*** | ***8/4/2022*** | ***13/4/2022*** | ***5 d*** |
| ***4.1.5*** | ***Resource Requirement*** | ***Subtask*** |  | ***Medium*** | ***Done*** | ***8/4/2022*** | ***13/4/2022*** | ***5 d*** |
| ***4.1.6*** | ***Technology*** | ***Subtask*** |  | ***Medium*** | ***Done*** | ***8/4/2022*** | ***13/4/2022*** | ***5 d*** |
| ***4.1.7*** | ***Schedule*** | ***Subtask*** |  | ***Medium*** | ***Done*** | ***8/4/2022*** | ***13/4/2022*** | ***5 d*** |
| ***4.1.8*** | ***Risk Management*** | ***Subtask*** |  | ***Medium*** | ***Done*** | ***8/4/2022*** | ***13/4/2022*** | ***5 d*** |
| ***4.1.9*** | ***Security Aspect*** | ***Subtask*** |  | ***Medium*** | ***Done*** | ***8/4/2022*** | ***13/4/2022*** | ***5 d*** |
| ***4.1.10*** | ***Naming Conventions*** | ***Subtask*** |  | ***Medium*** | ***Done*** | ***8/4/2022*** | ***13/4/2022*** | ***5 d*** |
| ***4.1.11*** | ***Conclusions*** | ***Subtask*** |  | ***Medium*** | ***Done*** | ***8/4/2022*** | ***13/4/2022*** | ***5 d*** |
| ***4.2*** | ***Requirement*** | ***Story*** | ***Hoàng Việt*** | ***High*** | ***Done*** | ***8/4/2022*** | ***11/4/2022*** | ***3 d*** |
| ***4.2.1*** | ***Product Use Cases*** | ***Subtask*** |  | ***Medium*** | ***Done*** | ***8/4/2022*** | ***11/4/2022*** | ***3 d*** |
| ***4.2.2*** | ***Functional Requirements*** | ***Subtask*** |  | ***Medium*** | ***Done*** | ***8/4/2022*** | ***11/4/2022*** | ***3 d*** |
| ***4.2.3*** | ***Data Requirements*** | ***Subtask*** |  | ***Medium*** | ***Done*** | ***8/4/2022*** | ***11/4/2022*** | ***3 d*** |
| ***4.2.4*** | ***Performance*** | ***Subtask*** |  | ***Medium*** | ***Done*** | ***8/4/2022*** | ***11/4/2022*** | ***3 d*** |
| ***4.2.5*** | ***Dependability*** | ***Subtask*** |  | ***Medium*** | ***Done*** | ***8/4/2022*** | ***11/4/2022*** | ***3 d*** |
| ***4.2.6*** | ***Security*** | ***Subtask*** |  | ***Medium*** | ***Done*** | ***8/4/2022*** | ***11/4/2022*** | ***3 d*** |
| ***4.2.7*** | ***Usability*** | ***Subtask*** |  | ***Medium*** | ***Done*** | ***8/4/2022*** | ***11/4/2022*** | ***3 d*** |
| ***4.3*** | ***Analysis*** | ***Story*** | ***Đình Khang*** | ***High*** | ***Done*** | ***8/4/2022*** | ***10/4/2022*** | ***2 d*** |
| ***4.3.1*** | ***System Design*** | ***Subtask*** |  | ***Medium*** | ***Done*** | ***8/4/2022*** | ***10/4/2022*** | ***2 d*** |
| ***4.3.2*** | ***Software Architecture*** | ***Subtask*** |  | ***Medium*** | ***Done*** | ***8/4/2022*** | ***10/4/2022*** | ***2 d*** |
| ***5*** | ***Release*** | ***Epic*** |  | ***Low*** | ***Done*** | ***14/4/2022*** | ***17/4/2022*** | ***3 d*** |
| ***5.1*** | ***Check Document & Edit*** | ***Story*** | ***Hoàng Việt*** | ***Low*** | ***Done*** | ***14/4/2022*** | ***15/4/2022*** | ***1 d*** |
| ***5.2*** | ***Add References*** | ***Story*** | ***Hoàng Việt*** | ***Low*** | ***Done*** | ***16/4/2022*** | ***17/4/2022*** | ***1 d*** |
| ***6*** | ***Implement*** | ***Epic*** |  | ***High*** | ***Done*** | ***18/4/2022*** | ***15/5/2022*** | ***28 d*** |
| ***6.1*** | ***Login*** | ***Story*** | ***Công Tuấn*** | ***Medium*** | ***Done*** | ***18/4/2022*** | ***24/4/2022*** | ***6 d*** |
| ***6.2*** | ***Register*** | ***Story*** | ***Phương Nam*** | ***Medium*** | ***Done*** | ***18/4/2022*** | ***24/4/2022*** | ***6 d*** |
| ***6.3*** | ***User Management*** | ***Story*** | ***Phương Nam*** | ***Medium*** | ***Done*** | ***25/4/2022*** | ***1/5/2022*** | ***6 d*** |
| ***6.4*** | ***Barverage*** | ***Story*** | ***Công Tuấn*** | ***High*** | ***Done*** | ***2/5/2022*** | ***15/5/2022*** | ***6 d*** |
| ***7*** | ***Testing*** | ***Epic*** |  | ***High*** | ***Done*** | ***16/5/2022*** | ***20/5/2022*** | ***4 d*** |
| ***7.1*** | ***Test GUI*** | ***Story*** | ***Hoàng Việt*** | ***Medium*** | ***Done*** | ***16/5/2022*** | ***17/5/2022*** | ***1 d*** |
| ***7.2*** | ***Test Function*** | ***Story*** | ***Đình Khang*** | ***Medium*** | ***Done*** | ***16/5/2022*** | ***17/5/2022*** | ***1 d*** |
| ***7.3*** | ***Make Testcase*** | ***Subtask*** | ***Phương Nam*** | ***High*** | ***Done*** | ***18/5/2022*** | ***20/5/2022*** | ***2 d*** |
| ***8*** | ***Deployment*** | ***Epic*** | ***Phương Nam*** | ***Low*** | ***Done*** | ***21/5/2022*** | ***22/5/2022*** | ***1 d*** |
| ***9*** | ***Make Final Report*** | ***Epic*** |  | ***Low*** | ***Done*** | ***23/5/2022*** | ***26/5/2022*** | ***3 d*** |
| ***9.1*** | ***Add Testcase*** | ***Subtask*** | ***Hoàng Việt*** | ***Low*** | ***Done*** | ***23/5/2022*** | ***26/5/2022*** | ***3 d*** |
| ***9.2*** | ***Add Implement*** | ***Subtask*** | ***Đình Khang*** | ***Low*** | ***Done*** | ***23/5/2022*** | ***26/5/2022*** | ***3 d*** |
| ***10*** | ***Prepare for Presentation*** | ***Epic*** |  | ***Low*** | ***Done*** | ***27/5/2022*** | ***3/6/2022*** | ***7 d*** |
| ***10.1*** | ***Make PowerPoint*** | ***Subtask*** | ***Đình Khang*** | ***Low*** | ***Done*** | ***27/5/2022*** | ***29/5/2022*** | ***2 d*** |
| ***10.2*** | ***Prepare to Answer the question*** | ***Story*** | ***Đình Khang*** | ***Low*** | ***Done*** | ***30/5/2022*** | ***3/6/2022*** | ***4 d*** |
| ***11*** | ***Presentation*** | ***Epic*** |  | ***High*** | ***To Do*** | ***4/6/2022*** | ***4/6/2022*** | ***0 d*** |

***Table 6.* Schedule and milestone**

### **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 MSteams 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.

## **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 refractor 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 application, the project may not operate on their PCs. | Medium | High | At the start of the project, project managers will set up and inspect everything to ensure that 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 7. Risk Management***

## **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.

## **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 | Sub Task | 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 13. Naming Convention and Definitions***

## **Conclusion**

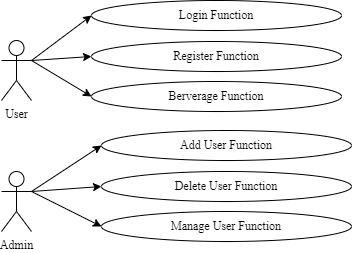
Beverage Website is a practical website serving the needs of exploring, creating, and providing new drinks for everyone. Systems are working hard to make the application useful capabilities, such as membership creation, monthly offers, and delivery policy. Later on, that makes the app a bigger reality.

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 6. 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 |

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

| Use Case Name: | User Login |
| --- | --- |
| Summary: | To get saved recipes, a new user must register a username and password. |
| Basic Flow: | 1. The use case starts when a user indicates that he wants to log in. 2. The system requests the username and password. 3. The user enters his username and password. 4. The system verifies the username and password against all registered users. 5. The system starts a login session and displays a welcome message based on the user's preferences. |
| Alternative Flows: | Step 4: If the username is invalid, the use case goes back to step 2.  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. |
| 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. |

| Use Case Name: | User Change Password |
| --- | --- |
| Summary: | A current user forgot password, and requested to change it. |
| Basic Flow: | 1. User clicks on forgot password button 2. System requests user to input the email 3. System sends verify code to the email 4. User inputs the code and types in new password |
| Alternative Flows: | Step 2: If the email is invalid, abort.  Step 4: If the password is invalid, prompt the user to input it again |
| 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. |

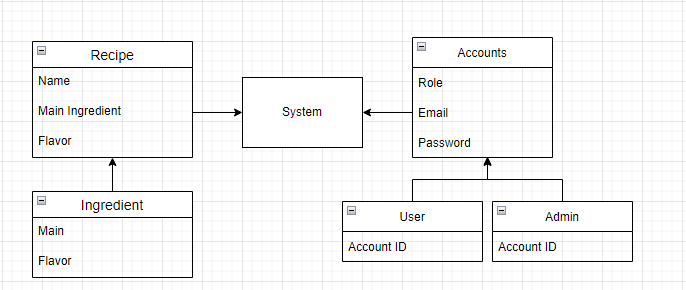
| Use Case Name: | Delete User |
| --- | --- |
| Summary: | Admin deletes a user for violating Terms of Service |
| Basic Flow: | 1. Admin checks an account and finds evidence that the user violated Terms of Service. 2. Admin deletes the account from the server and blacklists the email. |
| Alternative 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. |

| Use Case Name: | User finds recipe |
| --- | --- |
| Summary: | User find recipe based on 3 initial ingredients |
| Basic Flow: | 1. User clicks on Find recipe button 2. System requires user to choose 3 ingredients 3. System finds the recipe’s name based on the ingredients chosen 4. System displays the recipe’s name |
| 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. |

## **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 |

## **2.3.** **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.

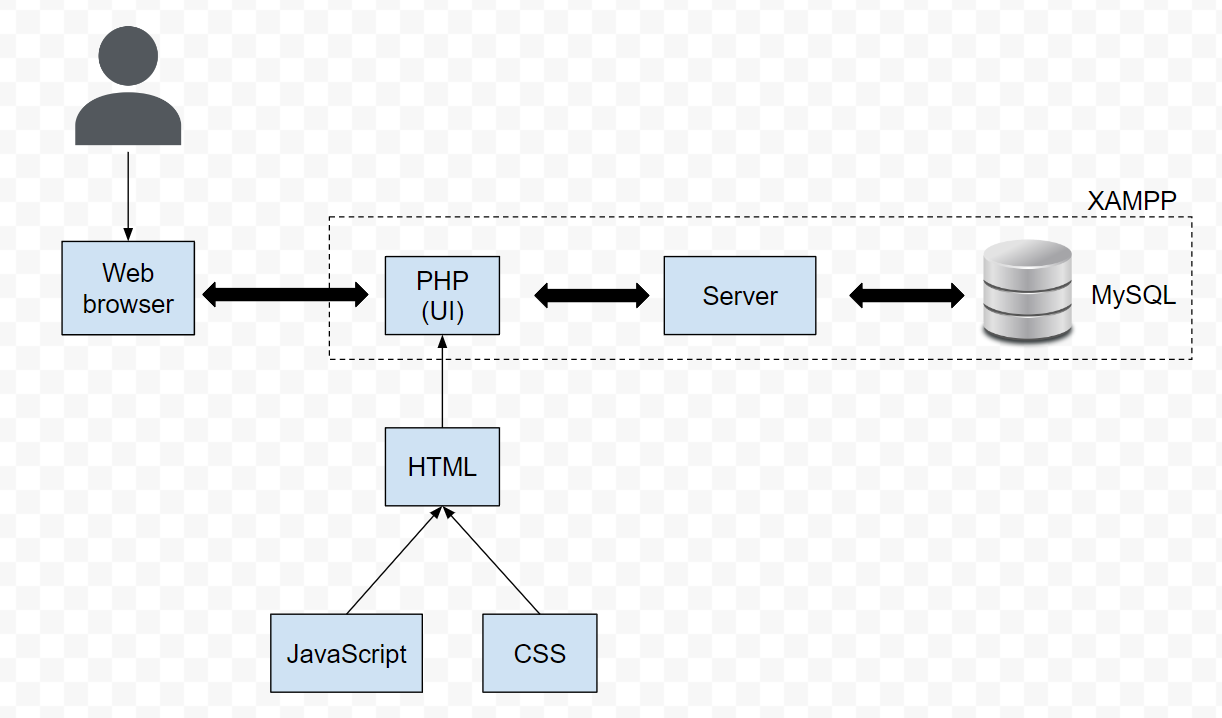
**2.6.2. Privacy requirements**

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. All users

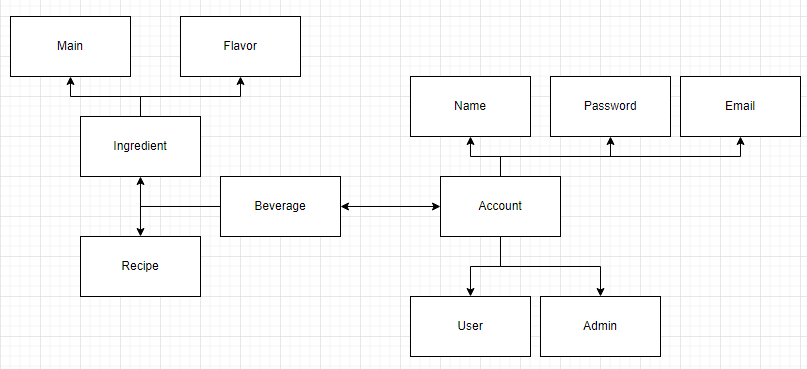
## **2.7.** **Usability Requirements**

**Chapter 3: Design**

## **3.1.** **System Design**



## **3.2.** **Database Structure**



## **3.3.** **Entity–relationship Diagram**

## **3.4.** **Sequence Diagram/Activity Diagram**

## **3.5.** **Class Diagram**

**Chapter 4: Implementation**

## **4.1.** **Login function**

## **4.2.** **Register function**

## **4.3.** **User function**

## **4.4.** **Admin function**

**Chapter 5: Test Plans**

## **5.1.** **Feature to be tested**

### **5.1.1. In scope:**

All of the features of the Berverage 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. |
| Berverage | User | A user can beverage many types of drinks based on specific materials. |

***Table 27. 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 the 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**

## **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 29. 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 incorrect data  When invalid data is used, the warning messages are displayed. |
| Criteria | Plan Tests have been passed |

***Table 30. 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 31. Security Testing***

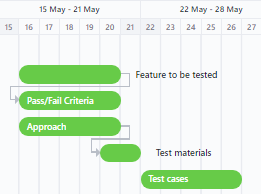
## **5.4. Test materials**

| Hardware | Computers: 3 Computers |
| --- | --- |
| Software | MySQL  Visual Studio Code |

***Table 34. Test materials***

## **5.5.** **Test cases**

## **5.6.** **Test Plans Schedule**



**Chapter 6: Glossary**

## **References**

[*https://github.com/TomPham204/Bartender-Web*](https://github.com/TomPham204/Bartender-Web)

[*https://en.wikipedia.org/wiki/SQL*](https://en.wikipedia.org/wiki/SQL)

[*https://en.wikipedia.org/wiki/JavaScript*](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/PHP)

[*https://en.wikipedia.org/wiki/Entity%E2%80%93relationship\_model*](https://en.wikipedia.org/wiki/Entity%E2%80%93relationship_model)

[*https://en.wikipedia.org/wiki/Graphical\_user\_interface*](https://en.wikipedia.org/wiki/Graphical_user_interface)

[*https://www.php.net*](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?fbclid=IwAR2_R6FK7HJWy4OrlKZvUatTheQGqp5C0Gn21AnNB_naUYd_IQhWbmMLgNo#Func)

[*https://app.clickup.com/14289914/v/g/dm2zu-187*](https://app.clickup.com/14289914/v/g/dm2zu-187)

[*https://www.youtube.com/watch?v=8MgpE2DTTKA*](https://www.youtube.com/watch?v=8MgpE2DTTKA)

[*https://www.youtube.com/watch?v=lvYnfMOUOJY*](https://www.youtube.com/watch?v=lvYnfMOUOJY)