**Capstone Project Final Report**

**Amusement Parks Ticket Booking System**

|  |  |
| --- | --- |
| **GoBoKi Team** | |
| **Group Members** | Đào Quang Toàn - SE05372  Phùng Trí Đức - SE05234  Vũ Thị Phương Thảo - SE06143  Vương Minh Hiếu - SE05869  Nguyễn Vũ Hiếu - SE05422 |
| **Supervisor** | Lecturer: Phạm Ngọc Hà |
| **Capstone Project code** | GoBoKi |

**– HaNoi, May, 2020 –**

1

**Acknowledgement**

We wish to express our deepest gratitude to our supervisor Mr.Pham Ngoc Ha for his continuously sharing and motivating throughout the project. Under the instructions of Mr.Ha, the project team has always felt comfortable and confident. Apart from all the technologies and methodologies, Mr.Ha shared with us the philosophy of keeping the right attitude towards problems. We believe that was the most important factor that led to the success of this project.

We would also like to thank the instructors at FPT University for all the classes. We hope you will find this project as a reflection of the knowledge and experiences you have given us during this period of four years.

Finally, we truly appreciate the random fellows on the internet who has discussed it with us. Without them, it would have likely been a real struggle to solve every tiny problem.

2

**CONTENT**

**CHAPTER 1.........................................................................................................9 INTRODUCTION -** 紹介 **......................................................................................9 1. Purpose** （目的）**...................................................................................................... 9 2. Project Information** （プロジェクトの情報）**............................................................ 9 3. Project Team** （プロジェクトチーム） **................................................................... 9** 3.1. Supervisor (監督者)............................................................................................. 9 3.2. Team Members (チームの構成員) ....................................................................... 9 **4. Background (**バックグラウンド**) ............................................................................ 10 5. The current system**（存在しているシステム）**........................................................ 10** 5.1. Trippy ................................................................................................................... 10 5.2. KLOOK .................................................................................................................. 13 **6. Proposed Solution** （提案されているソリューション） **...................................... 14** 6.1. Solution Description （ソリューション説明） ............................................... 14 6.2. System Actors （システムアクター）............................................................. 15 6.2.1. Web/Front-End （ウェブ／フロントエンド）......................................... 15 6.2.2. Admin Board （管理者）............................................................................ 15 6.3. System Features （システム特徴）................................................................. 15 6.4. Intended Screen Flow （意図されている画面フロー） ................................. 16 6.4.1. GoBoKi screen flow （GoBoKi 画面フロー）............................................ 16 6.4.2. CMS screen flow （CMS 画面フロー）..................................................... 17 6.5. Technologies （技術） ..................................................................................... 18 **7. Constraints & Assumptions** （制約＆仮定） **.......................................................... 18** 7.1. Project Constraints （プロジェクト制約）..................................................... 18 7.2. Project Assumptions （プロジェクトの仮定）............................................... 18 **8. Out of scope functions** (範囲外) **............................................................................. 18 CHAPTER 2.......................................................................................................20 PROJECT PLAN -** プロジェクト計画**................................................................20 1. Overview** （概要）**.................................................................................................. 20** 1.1. Scope （範囲） ................................................................................................ 20 1.2. Objectives （目的） ......................................................................................... 20 1.3. Project Risks （プロジェクトリスク） ........................................................... 20

3

**2. Management Approach** （プロジェクト管理アプローチ）**.................................. 21** 2.1. Project Process （プロジェクトプロセス）................................................... 21 2.2. Change Management （変更管理）................................................................. 22 2.3. Quality Management （品質管理）................................................................. 22

2.3.1. Defect Prevention Strategy （欠陥防止戦略）........................................... 22 2.3.2. Review Strategy （レビュー戦略）............................................................ 22 2.4. Project Training Plan （プロジェクト研修計画）........................................... 23 **3. Estimation & Master Schedule** （見積もりとマスタースケジュール）**................ 23** 3.1. Project Size （プロジェクトサイズ） ............................................................ 23 3.1.1. Web................................................................................................................ 23 3.1.2. CMS ................................................................................................................ 24 3.2. Deliverables （成果物）................................................................................... 24 3.3. Detail Schedule （詳細なスケジュール）.......................................................... 25 3.4. Project Organization （プロジェクト組織）................................................... 26 3.4.1. Team & Structures （チームと構造）........................................................ 26 3.4.2. Roles & Responsibilities （役割と責任）.................................................... 26 3.5. Communication （プロジェクトコミュニケーション）............................... 27 3.5.1. Communication Plan （コミュニケーションプラン） ............................. 27 3.5.2. External Interface （外部インタフェース） ............................................. 27 3.6. Configuration Management （構成管理）....................................................... 28 3.6.1. Tools & Infrastructures （道具とインフラ）............................................. 28 3.6.2. Document Management （資料管理）...................................................... 28 3.6.3. Source Code Management （ソースコード管理）.................................... 28 **CHAPTER 3.......................................................................................................29 SOFTWARE REQUIREMENT SPECIFICATION -** システム要求仕様**...................29 1. Product Scope** （製品の範囲）**............................................................................... 29 2. Functional Requirements** （機能要求） **................................................................. 29** 2.1. Use case diagram （ユースケース図） .............................................................. 29 2.2. Business Rules （ビジネスルール）................................................................ 31 2.3. Use Cases （ユースケース）........................................................................... 32 2.3.1. Admin & Staff （管理者＆スタッフ） ....................................................... 34 2.3.2. User （ユーザー） .................................................................................. 132

4

2.3.3. Guest （ゲスト）..................................................................................... 161 **3. Non-Functional Requirements** （非機能要件）**....................................................... 166** 3.1. Security （セキュリティ）............................................................................ 166 3.2. Accuracy （正確さ）...................................................................................... 166 3.3. Usability （ユーザビリティ）....................................................................... 166 3.4. Performance （パフォーマンス） ................................................................... 166 3.5. Availability （可用性）................................................................................... 166 3.6. Maintainability （メンテナンス性）............................................................. 166 3.7. Performance （パフォーマンス） ................................................................ 167 3.8. Reliability （信頼性）........................................................................................ 167 **4. Entity Relationship Diagram** （テーブル **ER** 図）**............................................. 167 CHAPTER 4.....................................................................................................168 SOFTWARE DESIGN -** システム設計の説明**..................................................168 1. Purpose** （目的）**.................................................................................................. 168 2. Architecture Overview** (アーキテクチャの概要) **................................................. 168** 2.1. System architecture (システムのアーキテクチャ) ...................................... 168 2.2. System Architecture Explanation (システムのアーキテクチャの説明) .......... 168 2.2.1. React............................................................................................................. 169 2.2.2. Spring Boot ................................................................................................... 169 2.2.3. Spring Data JPA............................................................................................. 169 2.2.4. MySQL .......................................................................................................... 170 2.2.5. Amazon web services.................................................................................... 170 **3. System Design** (システムの設計) **......................................................................... 170** 3.1. Database Design (データベース設計) ........................................................... 170 3.1.1. Explanation of Database Decision （データベース決定の説明） ........... 171 3.1.2. Database Diagram (データベース図)....................................................... 171 3.1.3. Data Dictionary (データディクショナリ) ............................................... 172 **4. Detail Design** (詳細な設計)**................................................................................... 180** 4.1. Detail Class Diagram (詳細クラス図) ............................................................. 180 4.1.1. Authentication （認証）........................................................................... 181 4.1.2. Category （カテゴリー）......................................................................... 183 4.1.3. City （市）................................................................................................ 185 4.1.4. Game （ゲーム）..................................................................................... 187

5

4.1.5. Order （注文）......................................................................................... 190 4.1.6. Place （場所）.......................................................................................... 193 4.1.7. Ticket （チケット）................................................................................. 196 4.1.8. TicketType （チケットタイプ）.............................................................. 198 4.1.9. User （ユーザー）................................................................................... 200 4.1.10. VisitorType （ビジタータイプ） ........................................................... 204

4.2. Sequence Diagram (シーケンス図)................................................................ 207 4.2.1. Content Management System (コンテンツ管理システム) ..................... 207 4.2.2. Guest （ゲスト）..................................................................................... 237 4.2.3. User （ユーザー）................................................................................... 240

**CHAPTER 5.....................................................................................................244 SOFTWARE TESTING -**ソフトウェアテスト **.................................................244 1. Introduction** (紹介)**............................................................................................... 244** 1.1. Purpose (目的) .............................................................................................. 244 1.2. Scope of Testing (テスト範囲)...................................................................... 244 1.2.1. Testing phases （テスト段階）................................................................ 244 1.2.2. Testing types （テストの種類）.............................................................. 245 1.2.3. Testing range （テストの範囲）.............................................................. 245 **2. Test Plan** (テスト計画) **................................................................................... 245** 2.1. Testing Tools and Environment (テスティングツールと環境) ........................ 245 2.1.1. Testing Tools (テスティングツール) ...................................................... 245 2.1.2. Testing Environment (テスティングの環境)........................................... 246 2.2. Resources and Responsibilities (資源と責任)................................................ 247 2.3. Test Strategy (テスト戦略) .......................................................................... 247 2.3.1. Test Model (テストのモデル) ................................................................. 247 2.3.2. Test Process (テストプロセス) ............................................................... 248 2.3.3. Test Types (テストの種類)...................................................................... 248 2.3.4. Test Stages (テスト段階) ......................................................................... 250 2.3.5. Test Schedule (テスト スケジュール) ..................................................... 251 2.3.6. Test Deliverables(テスト成果物) ................................................................ 251 2.3.7. Resource & Responsibilities (リソースと責任) ........................................... 252 2.4. Features to be Tested (テストされる機能)...................................................... 253

6

**3. Test Approach** (テストアプローチ)**........................................................................ 253** 3.1. Unit Testing (単体テスト) ................................................................................ 253 3.2. Integration Testing (結合テスト)...................................................................... 253 3.3. System Testing (システムテスト).................................................................... 254 3.4. Performance Testing (性能テスト)................................................................ 254

**4. Test Report** (テスト報告)**........................................................................................ 254** 4.1. Unit Test (単体テスト) ..................................................................................... 254 4.2. Integration Test (結合テスト) .......................................................................... 255 4.3. System Test (システムテスト)......................................................................... 256

4.3.1. Client (クライアント) ................................................................................ 256 4.3.2. CMS (コンテンツ管理システム)................................................................ 259 4.4. Performance Test (性能 の テスト)................................................................ 261 **CHAPTER 6.....................................................................................................263 USER MANUAL -** ユーザーマニュアル**.........................................................263 1. Installation Guide** （インストールガイド）**........................................................ 263** 1.1. Required components （ 必要なコンポーネント） ..................................... 263

1.2. Create front-end builds and deploy with source code （フロントエンドビルド を作成し、ソースコードでデバッグします）..................................................... 264 1.3. Create a back-end build and deploy with source code （バックエンドビルドを作 成し、ソースコードを使用してデプロイします） ............................................. 266

**2. Client Screen flow** (クライアント画面フロー) **.................................................... 268** 2.1. Home page (ホームページ) ........................................................................... 268 2.2. Login page (ログインページ) ........................................................................ 269 2.3. Reset password page (パスワードリセットページ) .................................... 270

2.3.1. Enter new password page (新しいパスワード入力ページ).................... 270 2.4. Register page (登録ページ) ........................................................................... 271 2.5. Search result page (検索結果ページ) ............................................................ 272 2.6. Location detail page (場所詳細ページ) ......................................................... 273

2.6.1. Select ticket (チケットを選択します)..................................................... 273 2.7. Payment page (支払いページ)....................................................................... 274 **3. Client Screen flow** (クライアント画面フロー)**.................................................... 277** 3.1. User profile page (ユーザープロフィールページ) .......................................... 277

7

3.2. Edit profile page (プロフィール編集ページ) ................................................ 278 3.3. My order page (私の注文ページ) .................................................................. 278 3.4. Order detail page (注文詳細ページ).............................................................. 279 3.5. Change password page (パスワード変更ページ) ......................................... 280

**4. CMS page flow** （CMS ページフロー ）**.............................................................. 280** 4.1. Login page （ログインページ）....................................................................... 280 4.2. User manager page （ユーザーマネージャーページ）............................... 281

4.2.1. Add new user page （新しいユーザ追加ページ）.................................. 283 4.2.2. Update user page （ユーザーアップデートページ）............................ 284 4.3. Place Manager page （場所マネージャーページ）..................................... 284 4.3.1. Add new place page （新しい場所追加ページ ）...................................... 286 4.3.2. Update place page （場所アップデートページ)..................................... 287 4.4. Category manager page (カテゴリーマネージャーページ)......................... 287 4.4.1. Add new category page （新しいカテゴリー追加ページ） .................. 289 4.4.2. Update Category page （カテゴリーアップデートページ）................. 290 4.5. Game Manager page （ゲームマネージャーページ）................................. 290 4.5.1. Add new game page（新しいゲーム追加ページ）.................................... 292 4.5.2. Update game page （ゲームアップデートページ）.............................. 293 4.6. City Manager page （都市マネージャーページ）........................................ 293 4.6.1. Add new city page （新しい都市追加ページ）....................................... 294 4.6.2. Update city page （都市アップデートページ）..................................... 295 4.7. Ticket Manager page （チケットマネージャーページ）............................. 296 4.7.1. Add new ticket type page (新しいチケットタイプ追加ページ) ............ 298 4.7.2. Update ticket type page （チケットタイプアップデートページ) ......... 299 4.7.3. Add new visitor type page（新しい訪問者タイプ追加ページ） ............... 300 4.7.4. Update visitor type page （訪問者タイプアップデートページ）............. 300 4.8. Order Manager page (注文マネージャーページ)......................................... 301 4.8.1. Order detail page (注文詳細ページ)........................................................ 303 4.9. Report statistic page (レポート統計ページ)................................................. 303

8

**CHAPTER 1**

**INTRODUCTION -** 紹介

**1. Purpose** （目的）

This chapter provides an overview of the project, including background information, review of existing systems, their problems and proposal for ideas of improvement. **2. Project Information** （プロジェクトの情報）

∙ Project name: Amusement Parks Ticket Booking System.

∙ Project code: GoBoKi.

∙ Project group name: GoBoKi Team.

∙ Product type: Website application.

∙ Timeline: May 20th 2020 – 2020.

**3. Project Team** （プロジェクトチーム）

**3.1. Supervisor (**監督者**)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Full Name** | **Email** | **Phone Number** | **Title** |
| Phạm Ngọc Hà | HaPN10@fe.edu.vn | 0988623000 | Lecturer |

*Table 3.1-1: Supervisor’s information*

**3.2. Team Members (**チームの構成員**)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Full Name** | **Student**  **Code** | **Email** | **Mobile** | **Role** |
| Đào Quang Toàn | SE05372 | toandqse05372@fpt.edu.v n | 0888662369 | Leader |
| Phùng Trí Đức | SE05234 | ducptse05234@fpt.edu.vn | 0919436297 | Member |
| Vũ Thị Phương Thảo | SE06143 | thaovtpse06143@fpt.edu. vn | 0961393725 | Member |

9

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Vương Minh Hiếu | SE05869 | hieuvmse05869@fpt.edu. vn | 0382726111 | Member |
| Nguyễn Vũ Hiếu | SE05422 | hieunvse05422@fpt.edu.v n | 0983450322 | Member |

*Table 3.2-1: Team member’s information*

**4. Background (**バックグラウンド**)**

Online travel is an indispensable trend when everyone can easily design tours such as buying air tickets, booking reservations or self-scheduling trips ... However, it is only at work. Booking travel tickets and hotel rooms. Due to the size and potential development of each resort, currently only large tourist areas such as Ba Na Hills or Vin-wondersystem invest and develop online booking. Therefore, ticket sales in tourist areas still take place in traditional ticket sales channels.

Digital and information technology platforms developed with just one smartphone, tourists can learn, book tours, make reservations and actively buy tickets, pay easily via e-wallets or bank accounts.

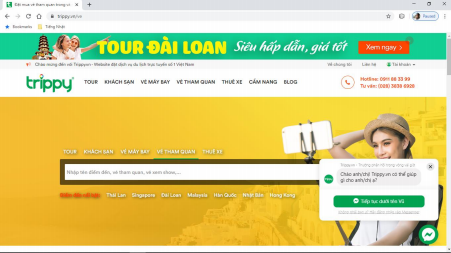
About 80% of the market share of online travel deals made on foreign sites such as Agoda.com, Booking.com, Klook.com... the rate of bookings on domestic sites is still low. **5. The current system**（存在しているシステム）

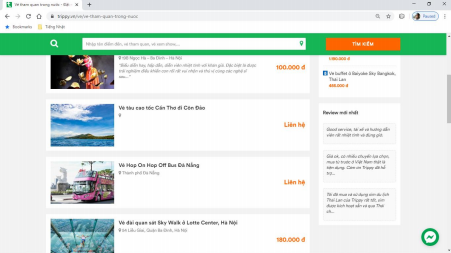
While there are websites that allow users to book ticket online but they all have certain problems. Below, we will go over a few ticket booking systems and their problems.

**5.1. Trippy**

Trippy is a travel website formed in 2014, which is a website specializing in the country and abroad selling tours. After more than 3 years of development, Trippy.vn has expanded other travel services to meet the increasing and diverse needs of Vietnamese tourists: hotels, book flights, amusement park tickets, airport shuttle service, and tour car rental service. This system is well implemented and has a severe investment in both web technology and features.

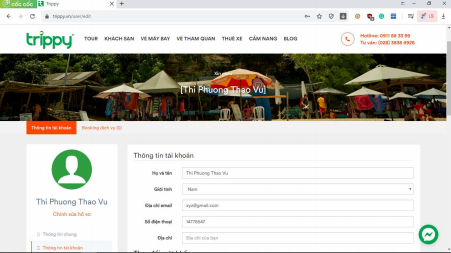
10

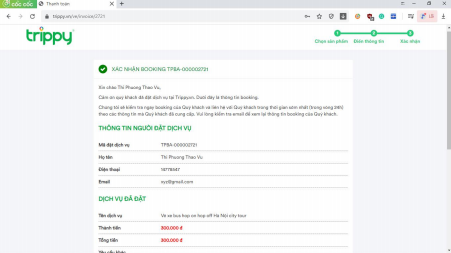
 *Figure 5.1-1: Trippy home page*

But they have a few problems that make tourists feel inconvenient when booking an amusement ticket. First of all, the system displays amusement park tickets but shows no prices on the screen. This repeatedly makes users feel uncomfortable when using websites. *Figure 5.1-2: All list amusement park page*

Secondly, we don't need to verify an account but still can book ticket with fake information.

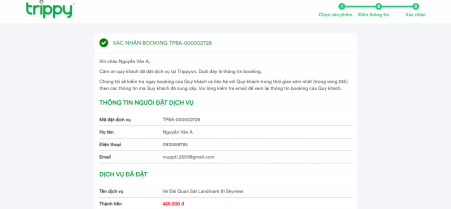
11

*Figure 5.1-3: View user information*

*Figure 5.1-4: View user booking*

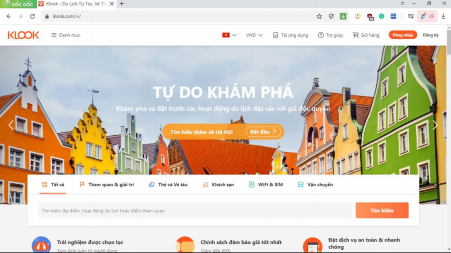
Thirdly, after successful booking at Trippy, you will receive a confirmation email and ticket redemption appointment voucher with all the details.

12

*Figure 5.1-5: Order ticket*

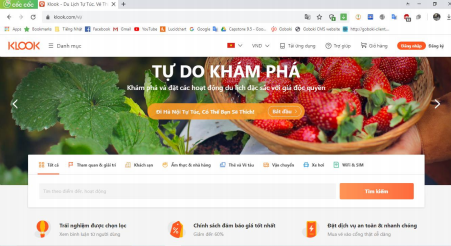
**5.2. KLOOK**

KLOOK a professional travel website. It not only allows users to search for information easily but also the layout is very scientific and beautiful. Users who want to travel anywhere can find their amusement parks ticket that they need quickly and the ticket is arranged relatively and easy to see with coherent information.

 *Figure 5.2-1: KLOOK home page*

However, the variety of content in KLOOK leaves users feeling a lack of in-depth content. KLOOK has taken full advantage of its technology platform and built too much content on a website, which has advantages and disadvantages. For example, users want to book hotel tickets, but ads for other content keep appearing ...

13

*Figure 5.2-2: KLOOK home page*

Additionally, after successfully booking your ticket at KLOOK, you will receive a confirmation email and a redemption appointment voucher with your complete information. At the amusement park, users need to redeem the park’s ticket according to the voucher provided by KLOOK.

**6. Proposed Solution** （提案されているソリューション） **6.1. Solution Description** （ソリューション説明）

Our product is an entertainment service that focuses on amusement park, so our system will have the following characteristics:

∙ Intuitive interactions:

Basic functionality will be easy to recognize and use, the UI will provide an elegant look that makes the user feel more comfortable when interacts.

∙ Availability:

Our service will be available on the web platform which will help users get to access more easily.

∙ Personalized experience:

Each user can search and book amusement parks based on their interests and accommodation.

∙ Open and Convenient:

Login and payment method are abundant.

14

**6.2. System Actors** （システムアクター）

**6.2.1. Web/Front-End** （ウェブ／フロントエンド）

|  |  |  |
| --- | --- | --- |
| **#** | **Actor** | **Actor Description** |
| 1 | Guest | People who don’t have an account on the service yet. They are just able to view and search for amusement parks only. They will have to register a new account to be able to use the features which the service provides. |
| 2 | User | People who possess an account on the website. They will be able to sign-in and use all of the features available. |

*Table 6.2.1-1: Web/Front-End’s actors*

**6.2.2. Admin Board** （管理者）

|  |  |  |
| --- | --- | --- |
| **#** | **Actor** | **Actor Description** |
| 1 | Admin | People who are assigned with a system-generated account. Their jobs are to manage service account systems. |
| 2 | Staff | People who are assigned with a system-generated account. Their jobs are to manage user accounts and order ticket in the system. |

*Table 6.2.2-1: Admin board’s actors*

**6.3. System Features** （システム特徴）

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Roles** | **Features** | **Feature Description** |
| 1 | Admin | Admin web login | Sign-in and Log-out from Admin web |
| 2 | Admin | Information of  amusement parks  management | Add, delete and edit information of amusement parks |
| 3 | Admin | Customer Information Management | Add, change status and edit  information of customer |

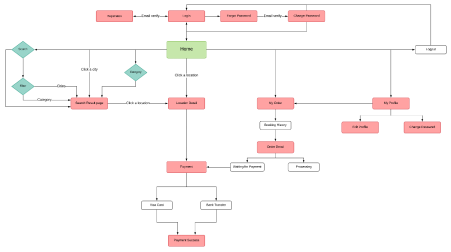
15

|  |  |  |  |
| --- | --- | --- | --- |
| 4 | Admin | Transaction information management | Explore transaction information |
| 5 | Admin | Ticket types management | Add, delete and edit ticket type |
| 6 | Admin | Visitor types management | Add, delete and edit visitor type |
| 7 | Admin | Revenue statistics | Explore revenue statistics of  amusement parks |
| 8 | Guest | Sign-up a new account |  |
| 9 | User | Sign-in/Log-out |  |
| 10 | User, Guest | Explore amusement parks | Search amusement parks by name |
| 11 | User | Tickets Booking | Order ticket |
| 12 | User | Payment |  |

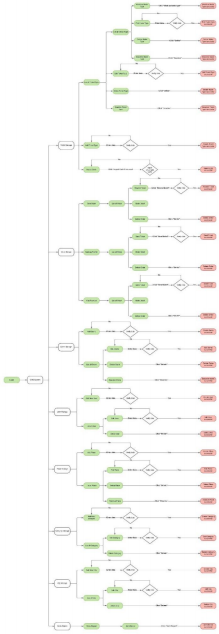
*Table 6.3-1: System features*

**6.4. Intended Screen Flow** （意図されている画面フロー）

**6.4.1. GoBoKi screen flow** （GoBoKi 画面フロー）

*Figure 6.4.1-1: GoBoKi screen flow*

16

**6.4.2. CMS screen flow** （CMS 画面フロー） *Figure 6.4.2-1: CMS screen flow*

17

**6.5. Technologies** （技術）

Programming Languages:

∙ Java 8.

∙ JavaScript ES6.

∙ JSX (JavaScript XML).

Frameworks and Libraries:

∙ Spring Boot v2.

∙ React JS v16.

∙ Redux v4.

Database:

∙ MySQL Database.

Platforms:

∙ Web Browser.

Technologies:

∙ Amazon Elastic Beanstalk.

∙ Amazon RDS.

∙ Amazon EFS or S3.

**7. Constraints & Assumptions** （制約＆仮定） **7.1. Project Constraints** （プロジェクト制約）

∙ Deadline will be fulfilled on time.

∙ Team members are capable of doing their work and report accordingly. ∙ There are no changes in HR during the course of the project.

**7.2. Project Assumptions** （プロジェクトの仮定）

∙ Time: 15 weeks until final product and documents.

∙ HR: 5 members of the project team.

∙ Deadline for each phase is specified.

**8. Out of scope functions** (範囲外)

∙ Currently, our website version works only on PC browsers. Not supported on devices such as phones, tablets ...

∙ No support registration, login with social network accounts.

18

∙ The system does not automatically send tickets to users, requiring staff support. ∙ The system has not processed the situation if the user spam orders.

∙ Currently, the system only has a visa method to be verified. Also, the bank transfer method needs to be re-verified.

19

**CHAPTER 2**

**PROJECT PLAN -** プロジェクト計画

**1. Overview** （概要）

**1.1. Scope** （範囲）

∙ This project will produce a website booking amusement ticket in VietNam. The amusement parks in the system are 100% in the territory of Vietnam, the current target that users all of Vietnamese.

∙ With digital and information technology platforms developed, tourists can learn, book amusements ticket, make reservations and actively buy tickets, pay easily for visa card or bank accounts line.

∙ Due to the time and the capability of our team, the app version will not be provided during this time.

∙ The stages associated with the ticket provider, commercial connection with amusement parks will not be conducted, all of which will be done in the form of re simulation according to the actual process.

∙ The management of this project will include analysis, design, implementation and testing. This project will be carried out for a fixed duration, maintenance and further features enhancement will not be included in the working flow.

**1.2. Objectives** （目的）

∙ This project will be carried out in 15 weeks starting from 18/05/2020, working days will have at least 8 hours attempt and also include the weekend.

∙ Each milestone will produce a deliverable with review and details update for the previous deliverable.

∙ Documents will follow the standard template and easy to interpret.

∙ The final product will be deployable to Amazon Web Service(AWS) before the end of week 12 with all specified functionality.

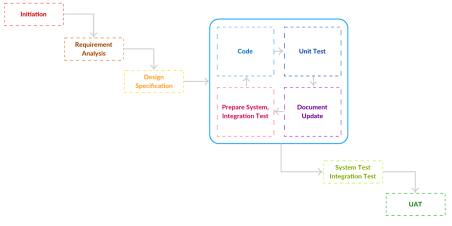
**1.3. Project Risks** （プロジェクトリスク）

20

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Risk Description** | **Impact** | **Possibility** | **Response Plans** |
| 1 | Some team members lack experience in technology used in the project | High | High | Establish 4 week training before the design phase |
| 2 | School schedule conflicts with project schedule | Low | Low | Meeting result will be informed to that member in person by team lead |
| 3 | Technical difficulty | Medium | Medium | Provide extra time for research and implementation |
| 4 | Some requirements may not have been analyzed carefully | High | High | Apply incremental update to SRS |
| 5 | Estimation for task required effort may be incorrect | Medium | High | Require extra time and resource |
| 6 | Project team Member misunderstands  requirement | High | Low | Members need to redefine their mission before each stage. |

*Table 1.3-1: Project risks*

**2. Management Approach** （プロジェクト管理アプローチ） **2.1. Project Process** （プロジェクトプロセス）

*Figure 2.1-1: Project process*

21

**2.2. Change Management** （変更管理）

|  |  |
| --- | --- |
| Where is the change request logged? | RequirementChangeManagement.xls |
| Who logs the change requests? | PM |
| Who reviews the change requests? | PM, BA |
| Who approves the change requests? | PM |

*Table 2.2-1: Change Management*

**2.3. Quality Management** （品質管理）

**2.3.1. Defect Prevention Strategy** （欠陥防止戦略）

|  |  |  |
| --- | --- | --- |
| **Item**  **(Process/Product)** | **Strategy** | **Expected Benefits** |
| Requirement Ambiguity | A quick review and analyze to agree on the flow of use-case | Clear use-case and reduce code implementation |
| Wrong Design | Ask instructor clearly before design | Save time |
| Weak coding skill | Training from week 2 | Reduce bug and time to fix |

*Table 2.3.1-1: Defect prevention strategy*

**2.3.2. Review Strategy** （レビュー戦略）

|  |  |  |
| --- | --- | --- |
| **Review Item** | **Reviewer** | **Review Type** |
| Project Introduction | Instructor | One-side review |
| Project Management Plan | Instructor | One-side review |
| Software Requirements Specification | BA, Instructor | One-side review  Group review |
| Software Design Document | BA, Instructor | One-side review  Group review |
| Code Package | Tech Lead | Group review |

22

|  |  |  |
| --- | --- | --- |
| Test Case | BA, Instructor | Group review |

*Table 2.3.2-1: Review strategy*

**2.4. Project Training Plan** （プロジェクト研修計画）

|  |  |  |  |
| --- | --- | --- | --- |
| **Training Area** | **Participants** | **When, Duration** | **Waiver Criteria** |
| Java Spring Boot | Nguyễn Vũ Hiếu | Start of week 1 : 3 weeks duration | Mandatory |
| ReactJS | Vương Minh Hiếu | Start of week 1: 3 weeks | Mandatory |
| Git, Github | Đào Quang Toàn  Vương Minh Hiếu  Nguyễn Vũ Hiếu | Start of week 1: 2 weeks | Mandatory |
| Document | Vũ Thị Phương Thảo | Start of week 1: 2 weeks | Mandatory |
| Selenium Test | Phùng Trí Đức | Start of week 1: 3 weeks | Mandatory |

*Table 2.4-1: Project training plan*

**3. Estimation & Master Schedule**

（見積もりとマスタースケジュール）

**3.1. Project Size** （プロジェクトサイズ）

**3.1.1. Web**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Feature** | **Screen/Service** | **Size** |
| 1 | Authentication | Sign in with Email | Medium |
| 2 | Authentication | Logout | Medium |
| 3 | Account Management | Edit account details | Medium |
| 4 | Account Management | Change password | Simple |
| 5 | Account Management | Edit settings | Simple |
| 6 | Browse | Search amusement parks | Medium |
| 7 | Browse | Search location | Medium |
| 8 | Browse | Home | Medium |
| 9 | Browse | Order ticket | Medium |
| 10 | User Management | My order | Hard |

23

|  |  |  |  |
| --- | --- | --- | --- |
| 11 | User Management | Profile | Simple |
| 12 | User Management | Edit profile | Simple |
| 13 | User Management | Payment | Simple |
| 14 | User Management | Transaction statistics | Medium |
| 15 | Parks Management | Amusement park’s page | Simple |

*Table 3.1.1-1: Project size - web*

**3.1.2. CMS**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Feature** | **Screen/Service** | **Size** |
| 1 | CMS | CMS Login | Simple |
| 2 | CMS | CMS Logout | Simple |
| 3 | CMS | Add/Remove User | Medium |
| 4 | CMS | Add/Remove/Deactivate Place | Medium |
| 5 | CMS | Add/Remove Category | Medium |
| 6 | CMS | Add/Remove/Deactivate Game | Medium |
| 7 | CMS | Add/Remove City | Medium |
| 8 | CMS | Add/Remove/Deactivate Ticket type | Medium |
| 9 | CMS | Add/Remove/Deactivate Visitor type | Medium |
| 10 | CMS | Order | Medium |
| 11 | CMS | Sale report | Medium |

*Table 3.1.2-1: Project size - cms*

**3.2. Deliverables** （成果物）

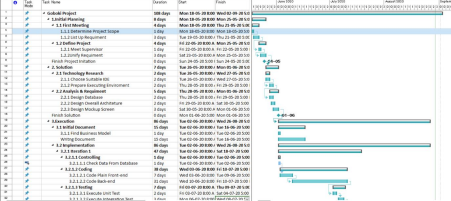
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Deliverable** | **Effort** | **Due Date** | **Notes** |
| 1 | Project Plan | 28 | 18/05 | Project Management Plan |
| 2 | Analysis & Requirement | 56 | 01/06 | Architecture Design, Detailed Design, Database |
| 3 | Code Package1 | 56 | 18/07 | Code & Unit test, System test cases |

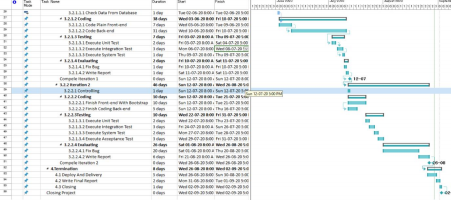
24

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 4 | Code Package2 | 56 | 26/08 | Code & Unit test, System test cases |
| 5 | UAT Package | 28 | 26/08 | Codes, System test reports |
| 6 | Final Package | 56 | 02/09/2020 | Final Codes & documents, User manual |

*Table 3.2-1: Deliverables*

**3.3. Detail Schedule** （詳細なスケジュール）

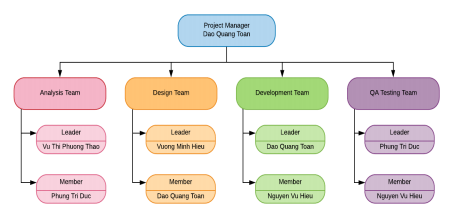
*Figure 3.3-1: Detail schedule*

*Figure 3.3-1: Detail schedule*

25

**3.4. Project Organization** （プロジェクト組織）

**3.4.1. Team & Structures** （チームと構造）

*Figure 3.4.1-1: Team & Structures*

**3.4.2. Roles & Responsibilities** （役割と責任）

|  |  |
| --- | --- |
| **Role** | **Responsibility** |
| Project Manager | Dao Quang Toan |
| Analysis Leader | Vu Thi Phuong Thao |
| Analysis Member | Phung Tri Duc |
| Design Leader | Vuong Minh Hieu |
| Design Member | Dao Quang Toan |
| Technical Leader | Dao Quang Toan |
| Technical member | Nguyen Vu Hieu |
| Test Leader | Phung Tri Duc |
| Test Member | Nguyen Vu Hieu |
| Test Member | Vu Thi Phuong Thao |
| Test Member | Dao Quang Toan |

*Table 3.4.2-1: Roles & Responsibilities*

26

**3.5. Communication** （プロジェクトコミュニケーション） **3.5.1. Communication Plan** （コミュニケーションプラン）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Communication Item** | **Who/ Target** | **Purpose** | **When,**  **Frequency** | **Type,**  **Method(s)** |
| Daily meeting | Project team | Update on progress and assign tasks to members | Daily | Google meet, Offline,  Messenger |
| Weekly meeting | Project team and  Supervisor | Update for project progress and Supervisor review | Weekly | Offline, Skype |

*Table 3.5.1-1: Communication Plan*

**3.5.2. External Interface** （外部インタフェース）

**3.5.2.1. FU Contacts**

|  |  |  |  |
| --- | --- | --- | --- |
| **Function** | **Contact Person (name, position)** | **Contact address**  **(email, telephone)** | **Responsibility** |
| Supervisor | Pham Ngoc Ha | HaPN10@fe.edu.vn 0988623000 | - Provide document template - Give instructions to project team  - Review deliverables - Supervise project status |
| Supporter | Nguyen Thi Mai Phuong | Phuongntm5@fe.ed u.vn | - Provide document template - Receive report project  - Answer questions about the project |

*Table 3.5.2.1-1: Fu contarts*

**3.5.2.2. Customer Contracts**

|  |  |  |  |
| --- | --- | --- | --- |
| **Function** | **Contact Person (name, position)** | **Contact address**  **(email, telephone)** | **Responsibility** |
| Supervisor | Pham Ngoc Ha | HaPN10@fe.edu.vn 0988623000 | - Provide document template - Give instructions to project team  - Review deliverables - Supervise project status |

27

|  |  |  |  |
| --- | --- | --- | --- |
| Supporter | Nguyen Thi Mai Phuong | Phuongntm5@fe.ed u.vn | - Provide document template - Receive report project  - Answer questions about the project |

*Table 3.5.2.2-1: Customer contracts*

**3.6. Configuration Management** （構成管理）

**3.6.1. Tools & Infrastructures** （道具とインフラ）

|  |  |
| --- | --- |
| **Programming languages** | **Java, ReactJS** |
| Framework | Java Spring Boot |
| API | API |
| DBMS | My SQL |
| IDEs/Editors | Visual Studio Code, IntelliJ |
| UML tools | Visual Paradigm, Astah, Lucidchart |
| Version Control | GitHub |
| Deployment server | Amazon Web Service |
| Testing tool | Selenium, Apache JMeter, Postman |

*Table 3.6.1-1: Tools & Infrastructures*

**3.6.2. Document Management** （資料管理）

● All documents will be managed in a Google Drive with the following structure: ○ “content” folder: Contain project content

■ “document”: project reports, required and additional documents. ● Each document report will be named in the format:

Report#\_”Report Name”\_SWP493\_G2

■ “resource”: image, video, resources for the project.

○ “reference” folder: Contain reference resources

● Updates, changes will be recorded each week.

**3.6.3. Source Code Management** （ソースコード管理）

∙ Team members use GitHub for source code and version control.

∙ Team members are only allowed to commit code to the local repository. ∙ Code must be fully tested and run well on the local side before sending pull request to master repository for finished implementations.

∙ Main branch will be hosted on GitHub and can only be changed by pulling from pull request by local dev branch.

∙ Pull request will be peer reviewed to determine whether it will be accepted or block.

28

**CHAPTER 3**

**SOFTWARE REQUIREMENT SPECIFICATION -** システム要求仕様

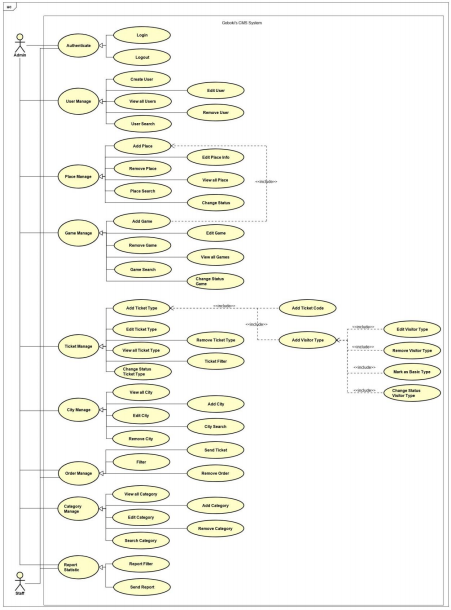
**1. Product Scope** （製品の範囲）

GoBoKi 1.0 is a web-based ticket booking online application that allows users to book, explore the amusement parks in VietNam through a simple and intuitive interface. This service will provide most functions without charge. At this time, the system is supporting the customer's payment with visa cards. However, the system does not support customers to refund, so there is a need to cancel tickets.

This project will also deliver a CMS system to manage some important entities. **2. Functional Requirements** （機能要求）

**2.1. Use case diagram** （ユースケース図）

29

*Figure 2.1-1: CMS use case diagram*

30

*Figure 2.1-2: Client’s use case diagram*

**2.2. Business Rules** （ビジネスルール）

|  |  |
| --- | --- |
| **ID** | **Description** |
| B01 | Account's email address must be valid. |
| B02 | Account's password must be from 8-20 characters in length and must contain at least 1 uppercase letter, lower case letter and digit. |
| B03 | When registering, or changing password, user must enter the new password twice. |
| B04 | A guest must provide their email address and password when registering an account. |
| B05 | A guest cannot register with an email that has already been registered. |
| B06 | After registering, guest must activate their account with the activation link sent to the account's email address. |
| B07 | User must provide their account's email address and password when logging into the website. |
| B08 | User cannot login to their account unless the account is activated. |
| B09 | User must provide their account's email address when resetting the account's password. |
| B10 | When changing password, new password must not be the same as the current |

31

|  |  |
| --- | --- |
|  | password. |
| B11 | An account must belong to one of the three roles: User, Admin, Staff. |
| B12 | Only Admin or Staff accounts can login to the CMS system. |
| B13 | Only Administrator accounts can manage User, Place, Game, Ticket in the system. |
| B14 | There cannot have two places with the same name. |
| B15 | User has to login before booking a ticket. |
| B16 | Park, game and ticket information of update or add mode must be different from details of data in the database. |
| B17 | Ticket type must be chosen. |
| B18 | Redemption date must be chosen. |
| B19 | Amount of ticket must be chosen. |
| B20 | All information about the park must be entered. |
| B21 | Price must be a positive number. |
| B22 | User must be received confirmation mail to go to the payment page. |
| B23 | A Place must have at least one game. |
| B24 | A game must belong to at least one Place. |
| B25 | User only can be received ticket when completed payment. |
| B26 | The search section will be on top of the screen as part of the one-page layout of the web version. |
| B27 | When payment, choose at least 1 payment method. |
| B28 | Purchased money cannot be refunded. |
| B29 | The searching process will only happen after the user hits Enter. |
| B30 | There cannot have two games with the same name. |
| B31 | All information of game must be entered. |
| B32 | A Place must have at least one ticket type. |
| B33 | A ticket type must belong to one place. |
| B34 | All information of ticket must be entered. |
| B35 | Uploaded files must be in ".XLSX" format |
| B36 | Each email address can be used for only one account |
| B37 | There cannot have two ticket types with the same name. |
| B38 | There cannot have two visitor types with the same name. |
| B39 | A ticket type must belong to one visitor type. |
| B40 | There cannot have two cities with the same name. |
| B41 | All information of city must be entered. |
| B42 | Must have at least one city in the system. |
| B43 | No need to fill in all the information fields when searching. |
| B44 | There cannot have two categories with the same name. |
| B45 | All information of category must be entered. |

*Table 2.2-1: Business Rules*

**2.3. Use Cases** （ユースケース）

The system consists of 62 use cases for four types of actors. Below are the lists of actors and use cases:

32

|  |  |
| --- | --- |
| **Actor** | **Description** |
| User | Customers who have an account and can use all functions of this website. |
| Guest | Everyone who does not have an account on the GoBoKi system. |
| Admin | A group of people who has the authority to manager the issuers and system’s status. |
| Staff | A staff that can view transaction information and send ticket to user’s email. |

*Table 2.3 – 1: User’s role*

|  |  |  |
| --- | --- | --- |
| **ID** | **Actor** | **Name** |
| UC-1 | Amin, Staff, User | Login |
| UC-2 | Amin, Staff, User | Logout |
| UC-3 | Admin | Create User |
| UC-4 | Admin | Edit User |
| UC-5 | Admin | View all Users |
| UC-6 | Admin | Remove User |
| UC-7 | Admin | User Search |
| UC-8 | Admin | Add Place |
| UC-9 | Admin | Edit Place Information |
| UC-10 | Admin | Remove Place |
| UC-11 | Admin | View all Places |
| UC-12 | Admin | Place Search |
| UC-13 | Admin | Change Status Place |
| UC-14 | Admin | Add Games |
| UC-15 | Admin | Edit Game |
| UC-16 | Admin | Remove Game |
| UC-17 | Admin | View all Games |
| UC-18 | Admin | Game Search |
| UC-19 | Admin | Change Status Game |
| UC-20 | Admin | Add Ticket Type |
| UC-21 | Admin | Add Ticket Code |
| UC-22 | Admin | Add Visitor Type |
| UC-23 | Admin | Remove Visitor Type |
| UC-24 | Admin | Edit Visitor Type |
| UC-25 | Admin | Mask as Basic Type |
| UC-26 | Admin | Change Status Visitor Type |
| UC-27 | Admin | Edit Ticket Type |
| UC-28 | Admin | Remove Ticket Type |
| UC-29 | Admin | View all Ticket Type |
| UC-30 | Admin | Change Status Ticker Type |

33

|  |  |  |
| --- | --- | --- |
| UC-31 | Admin | Ticket Filter |
| UC-32 | Admin | View all City |
| UC-33 | Admin | Add City |
| UC-34 | Admin | Edit City |
| UC-35 | Admin | City Search |
| UC-36 | Admin | Remove City |
| UC-37 | Admin, Staff | Send Ticket |
| UC-38 | Admin, Staff | Remove Order |
| UC-39 | Admin, Staff | Filter |
| UC-40 | Admin | View all Category |
| UC-41 | Admin | Add Category |
| UC-42 | Admin | Edit Category |
| UC-43 | Admin | Remove Category |
| UC-44 | Admin | Search Category |
| UC-45 | Admin, Staff | Report Filter |
| UC-46 | Admin, Staff | Send Report |
| UC-47 | User | Edit Profile |
| UC-48 | User | Change Password |
| UC-49 | User | My order |
| UC-50 | User | Forgot Password |
| UC-51 | User | Visa Card |
| UC-52 | User | Bank Transfer |
| UC-53 | User | Add Ticket to Order |
| UC-53.1 | User | Choose Ticket Type |
| UC-53.2 | User | Pick Redemption Date |
| UC-53.3 | User | Pick Amount of Ticket |
| UC-54 | User, Guest | View all Place in a Location |
| UC-55 | User, Guest | View Place's Detail |
| UC-56 | User, Guest | Search |
| UC-56.1 | User, Guest | Filter |
| UC-57 | Guest | Register |
| UC-57.1 | Guest | Verify account |

*Table 2.3 – 2: Use case*

**2.3.1. Admin & Staff** （管理者＆スタッフ）

34

**2.3.1.1. Login**

****

*Figure 2.3.1.1-1: Login activity diagram*

|  |  |  |  |
| --- | --- | --- | --- |
| **UC ID and Name** | **UC-1 Login** | | |
| **Create by** | DucPT | **Date created** | 04/06/2020 |
| **Primary Actors** | Admin | **Secondary Actors** | Staff, User |
| **Description** | Login to GoBoKi website | | |
| **Priority** | High | | |
| **Trigger** | Admin/Staff want to login website | | |
| **Pre-Condition(s)** | ∙ Admin has an account on GoBoKi system.  ∙ Admin accounts have been authorized.  ∙ Admin isn’t currently logged into the website.  ∙ Admin's device is connected to the internet when logging in. | | |
| **Post-Condition(s)** | ∙ Admin logged in the website successfully.  ∙ The system records the successful login activity in the Activity Log. ∙ Admin is redirected to Admin page. | | |
| **Basic Flow** | 1. Admin visits GoBoKi website.  2. Admin clicks Login or Register menu on the navigation bar. 3. System displays Login or Register form in the dialog box. 4. Admin enters email address and password on the login form. 5. Admin clicks Login button on the login form. | | |

35

|  |  |
| --- | --- |
|  | 6. System display the GoBoKi Admin page.  7. The system recognizes successful login activities in the Activity Log. |
| **Alternative Flow** | ∙ Only User |
| **Exception Flow** | ∙ The authentication system failed to log in and displayed an error message.  ∙ The user chooses forgot password command.  ∙ User enters wrong password or username.  ∙ Cannot communicate with API server. System displays error message |
| **Business Rules** | B01, B02, B07, B08, B11 |
| **Non-Functional Requirement** | ∙ When user checks the “Remember me” checkbox, next time they open the browser, user will be automatically signed in with the email used before. |

*Table 2.3.1.1-2: Login*

**Screen design**

****

*Figure 2.3.1.1-3: Login for admin*

36

*Figure 2.3.1.1-2: Login for user*

**2.3.1.2. Logout**

****

*Figure 2.3.1.2-1: Logout activity diagram*

|  |  |  |  |
| --- | --- | --- | --- |
| **UC ID and Name** | **UC-2 Logout** | | |
| **Create by** | DucPT | **Date created** | 04/06/2020 |

37

|  |  |  |  |
| --- | --- | --- | --- |
| **Primary Actors** | Admin | **Secondary Actors** | Staff, User |
| **Description** | Log out the GoBoKi system. | | |
| **Priority** | High | | |
| **Trigger** | User want to logout website | | |
| **Pre-Condition(s)** | ∙ User has logged into GoBoKi system. | | |
| **Post**  **Condition(s)** | ∙ Login status set to false and the access token is cleared on the browser. | | |
| **Basic Flow** | 1. User clicks on “Logout” at menu item.  2. System clears the access token on the browser.  3. System displays the GoBoKi home page. | | |
| **Alternative Flow** | N/A | | |
| **Exception Flow** | ∙ Cannot communicate with API server. System displays error message | | |
| **Business Rules** | N/A | | |
| **Non-Functional Requirement** | N/A | | |

*Table 2.3.1.2-1: Logout*

**Screen design**

****

*Figure 2.3.1.2-2: Logout for admin*

**

*Figure 2.3.1.2-3: Logout for user*

38

**2.3.1.3. Create User**

****

*Figure 2.3.1.3-1: Create User activity diagram*

|  |  |  |  |
| --- | --- | --- | --- |
| **UC ID and Name** | **UC-3 Create User** | | |
| **Create by** | ThaoVTP | **Date created** | 04/06/2020 |
| **Primary Actors** | Admin | **Secondary Actors** | N/A |
| **Description** | Allow admin to view user account. | | |
| **Priority** | Medium | | |
| **Trigger** | Admin want to create a new user. | | |
| **Pre-Condition(s)** | ∙ Admin has an admin account in GoBoKi system.  ∙ Admin isn’t currently logged into the website. | | |
| **Post**  **Condition(s)** | ∙ A user is created. | | |
| **Basic Flow** | 1. On the Side bar, the admin clicks on “User manager” tab. 2. System shows “User manager” page.  3. Admin clicks on “Add New User” button.  4. System shows a Sign Up form in the dialog box.  5. Admin fills information of User of account.  ∙ First name | | |

39

|  |  |
| --- | --- |
|  | ∙ Last name  ∙ Mail  ∙ Password  ∙ Phone number  ∙ Choose role(Admin/User/Staff)  6. Admin clicks on “Save User” button to complete.  7. Admin clicks on button Yes or No of the notification message. 8. System displays a message to notify that “Add User” successfully. |
| **Alternative Flow** | N/A |
| **Exception Flow** | ∙ Input data at step 5 is conflict with business rule.  ∙ Admin clicks “No” button of the confirmation dialog (Step 7), system hides the dialog box, and the use case ends.  ∙ Cannot communicate with API server. System displays error message |
| **Business Rules** | B11, B12, B13 |
| **Non-Functional Requirement** | N/A |

*Table 2.3.1.3-1: Create User*

**Screen design**

*****Figure 2.3.1.3-2: Create User*

40

*Figure 2.3.1.3-3: Create User*

41

**2.3.1.4. Edit User**

*****Figure 2.3.1.4-1: Edit User activity diagram*

|  |  |  |  |
| --- | --- | --- | --- |
| **UC ID and Name** | **UC-4 Edit User** | | |
| **Create by** | ThaoVTP | **Date created** | 04/06/2020 |
| **Primary Actors** | Admin | **Secondary Actors** | N/A |
| **Description** | Allow admin to edit account if needed | | |
| **Priority** | Medium | | |
| **Trigger** | Admin want to login website | | |
| **Pre-Condition(s)** | ∙ Admin has an admin account in GoBoKi system.  ∙ Admin isn’t currently logged into the website.  ∙ There is at least 1 user in the user list. | | |

42

|  |  |
| --- | --- |
| **Post**  **Condition(s)** | ∙ System update new data in user profile. |
| **Basic Flow** | 1. On the Side bar, the admin clicks on “User manager” tab. 2. System shows all User account list on “User manager” page. 3. Admin clicks on “Edit” button on each row.  4. Admin change information of User.  5. Admin clicks “Save User” to complete.  6. Admin clicks on button Yes or No of notification message 7. System displays edit successfully notification |
| **Alternative Flow** | N/A |
| **Exception Flow** | ∙ Cannot communicate with API server. System displays error message.  ∙ Admin clicks “No” button of the confirmation dialog (Step 6), system hides the dialog box, and the use case ends.  ∙ Input data at step 4 is conflict with business rule. |
| **Business Rules** | B11, B12, B13, B16 |
| **Non-Functional Requirement** | N/A |

*Table 2.3.1.4-1: Edit User*

**Screen design**

*****Figure 2.3.1.4-2: Edit User*

43

*Figure 2.3.1.4-3: Edit User*

**2.3.1.5. View all User**

****

*Figure 2.3.1.5-1: View all User activity diagram*

|  |  |  |  |
| --- | --- | --- | --- |
| **UC ID and Name** | **UC-5 View all User** | | |
| **Create by** | ThaoVTP | **Date created** | 04/06/2020 |
| **Primary Actors** | Admin | **Secondary Actors** | N/A |
| **Description** | Display a list of all user in GoBoKi system. | | |

44

|  |  |
| --- | --- |
| **Priority** | Medium |
| **Trigger** | N/A |
| **Pre-Condition(s)** | ∙ Admin has a GoBoKi admin account.  ∙ Admin is currently logged into the website by admin account. ∙ There is at least 1 user in the user list. |
| **Post**  **Condition(s)** | N/A |
| **Basic Flow** | 1. Login into website by admin account.  2. System displays admin page  3. Admin clicks on “User Manager” tab.  4. System displays the list of user, where each of them is presented with the following information:  ∙ First name  ∙ Last name  ∙ Mail  ∙ Phone number |
| **Alternative Flow** | N/A |
| **Exception Flow** | Cannot communicate with API server. System displays error message. |
| **Business Rules** | B12, B13 |
| **Non-Functional Requirement** | N/A |

*Table 2.3.1.5-1: View all User*

**Screen design**

*****Figure 2.3.1.5-2: View all User*

45

**2.3.1.6. Remove User**

****

*Figure 2.3.1.6-1: Remove User activity diagram*

|  |  |  |  |
| --- | --- | --- | --- |
| **UC ID and Name** | **UC-6 Remove User** | | |
| **Create by** | ThaoVTP | **Date created** | 09/06/2020 |
| **Primary Actors** | Admin | **Secondary Actors** | N/A |
| **Description** | Allow admin to delete account if needed | | |
| **Priority** | Medium | | |
| **Trigger** | Admin want to login website | | |
| **Pre-Condition(s)** | ∙ Admin has an admin account in GoBoKi system.  ∙ Admin isn’t currently logged into the website.  ∙ There is at least 1 user in the user list. | | |
| **Post**  **Condition(s)** | When the normal flow completes successfully, user account will be removed from the database. | | |
| **Basic Flow** | 1. On the Side bar, the admin clicks on “User manager”. 2. System shows all User account list on “User manager” page. 3. Admin clicks on “Remove” button on each row.  4. Admin clicks on button Yes or No of the notification message. | | |

46

|  |  |
| --- | --- |
|  | 5. System displays a message to notify that “User” has been deleted. |
| **Alternative Flow** | N/A |
| **Exception Flow** | ∙ Admin clicks “No” button of the confirmation dialog (Step 4), system hides the dialog box, and the use case ends.  ∙ Cannot communicate with API server. System displays error message. |
| **Business Rules** | B11, B12, B13 |
| **Non-Functional Requirement** | N/A |

*Table 2.3.1.6-1: Remove User*

**Screen design**

*****Figure 2.3.1.6-2: Remove User*

47

**2.3.1.7. User Search**

*****Figure 2.3.1.7-1: User Search activity diagram*

|  |  |  |  |
| --- | --- | --- | --- |
| **UC ID and Name** | **UC-7 User Search** | | |
| **Create by** | ThaoVTP | **Date created** | 09/06/2020 |
| **Primary Actors** | Admin | **Secondary Actors** | N/A |
| **Description** | Allow admin to search account if needed | | |
| **Priority** | Medium | | |
| **Trigger** | Admin want to login website | | |
| **Pre-Condition(s)** | ∙ Admin has an admin account in GoBoKi system.  ∙ Admin isn’t currently logged into the website.  ∙ There is at least 1 user in the user list. | | |
| **Post**  **Condition(s)** | When the normal flow completes successfully, user account by keyword will be displayed on the "User manager" page. | | |
| **Basic Flow** | 1. On the Side bar, the admin clicks on “User manager”. 2. System shows “User manager” page. | | |

48

|  |  |
| --- | --- |
|  | 3. In “User manager” page, fills in the information of user want to search.  ∙ First name  ∙ Last name  ∙ Mail  ∙ Phone number  ∙ Choose role(Admin/User/Staff)  4. Admin clicks on “Search” button.  5. System will show all user list according to the search keyword, where each of them is presented with the following information:  ∙ First name  ∙ Last name  ∙ Mail  ∙ Phone number  ∙ Role |
| **Alternative Flow** | N/A |
| **Exception Flow** | ∙ Input data at step 3 is conflict with business rule.  ∙ Cannot communicate with API server. System displays error message. |
| **Business Rules** | B11, B12, B13, B43 |
| **Non-Functional Requirement** | N/A |

*Table 2.3.1.7-1: User Search*

**Screen design**

*****Figure 2.3.1.7-2: User Search*

49

**2.3.1.8. Add Place**

****

*Figure 2.3.1.8-1: Add Place activity diagram*

|  |  |  |  |
| --- | --- | --- | --- |
| **UC ID and Name** | **UC-8 Add Place** | | |
| **Create by** | ThaoVTP | **Date created** | 04/06/2020 |
| **Primary Actors** | Admin | **Secondary Actors** | N/A |
| **Description** | Allows admin can add new Place information into GoBoKi website | | |
| **Priority** | High | | |
| **Trigger** | N/A | | |
| **Pre-Condition(s)** | ∙ Admin has a GoBoKi admin account.  ∙ Admin is currently logged into the website by admin account | | |

50

|  |  |
| --- | --- |
| **Post**  **Condition(s)** | ∙ New Place will be saved in database and the table of added location will be shown. |
| **Basic Flow** | 1. Login into website by admin account.  2. Admin clicks on “Place manager” tab.  3. Admin clicks on “Add New Place” button.  4. System shows add place form in the dialog box.  5. Fills in.  ∙ Place name  ∙ Choose Category  ∙ Choose City  ∙ Address  ∙ Phone Number  ∙ Mail  ∙ Short description  ∙ Detail Description  ∙ Choose image file  6. Admin clicks on “Save Place” button.  7. Admin clicks on button Yes or No of the notification message. 8. System displays notification of add successfully. |
| **Alternative Flow** | N/A |
| **Exception Flow** | ∙ Input data at step 5 is conflict with business rule.  ∙ Admin clicks “No” button of the confirmation dialog (Step 7), system hides the dialog box, and the use case ends.  ∙ Cannot communicate with API server. System displays error message. |
| **Business Rules** | B12, B13, B14, B20, B23 |
| **Non-Functional Requirement** | N/A |

*Table 2.3.1.8-1: Add Place*

**Screen design**

****51

*Figure 2.3.1.8-2: Add Place*

*Figure 2.3.1.8-3: Add Place*

*Figure 2.3.1.8-4: Add Place*

52

**2.3.1.9. Edit Place Information**

****

*Figure 2.3.1.9-1: Edit Place activity diagram*

|  |  |  |  |
| --- | --- | --- | --- |
| **UC ID and Name** | **UC-9 Edit Place Information** | | |
| **Create by** | ThaoVTP | **Date created** | 04/06/2020 |
| **Primary Actors** | Admin | **Secondary Actors** | N/A |
| **Description** | Allows admin to enter information and edit for a Place into GoBoKi system. | | |
| **Priority** | Medium | | |
| **Trigger** | N/A | | |

53

|  |  |
| --- | --- |
| **Pre-Condition(s)** | ∙ Admin has a GoBoKi admin account.  ∙ Admin is currently logged into the website by admin account. ∙ There is at least 1 Place in the Place list. |
| **Post**  **Condition(s)** | ∙ Information of Place will be saved in database and the table of list location will be shows. |
| **Basic Flow** | 1. Login into website by admin account  2. Admin clicks on “Place manager” tab.  3. System shows All Place list.  4. Admin clicks on “Edit” button in each row.  5. Fills in.  6. Admin clicks on “Save Place” button.  7. Admin clicks button Yes or No of notification message. 8. System displays edit successfully notification |
| **Alternative Flow** | N/A |
| **Exception Flow** | ∙ Input data at step 5 is conflict with business rule.  ∙ Admin clicks “No” button of the confirmation dialog (Step 7), system hides the dialog box, and the use case ends.  ∙ Cannot communicate with API server. System displays error message. |
| **Business Rules** | B12, B13, B14, B16, B20, B23, B32 |
| **Non-Functional Requirement** | N/A |

*Table 2.3.1.9-1: Edit Place information*

**Screen design**

*****Figure 2.3.1.9-2: Edit Place Information*

54

*Figure 2.3.1.9-3: Edit Place Information*

*Figure 2.3.1.9-4: Edit Place Information*

55

**2.3.1.10. Remove Place**

****

*Figure 2.3.1.10-1: Remove Place activity diagram*

|  |  |  |  |
| --- | --- | --- | --- |
| **UC ID and Name** | **UC-10 Remove Place** | | |
| **Create by** | ThaoVTP | **Date created** | 09/06/2020 |
| **Primary Actors** | Admin | **Secondary Actors** | N/A |
| **Description** | Allows admin to enter information and delete for a park into GoBoKi system. | | |
| **Priority** | Medium | | |
| **Trigger** | N/A | | |
| **Pre-Condition(s)** | ∙ Admin has a GoBoKi admin account.  ∙ Admin is currently logged into the website by admin account. ∙ There is at least 1 park in the park list. | | |
| **Post**  **Condition(s)** | When the normal flow completes successfully, park information will removed from the database. | | |
| **Basic Flow** | 1. Login into website by admin account  2. Admin clicks on “Place manager” tab.  3. System shows All Place list | | |

56

|  |  |
| --- | --- |
|  | 4. Admin clicks on “Remove” button in each row.  5. Admin clicks on button Yes or No of notification message. 6. System displays a message to notify that “Place” has been deleted. |
| **Alternative Flow** | N/A |
| **Exception Flow** | ∙ Admin clicks “No” button of the confirmation dialog (Step 5), system hides the dialog box, and the use case ends.  ∙ Cannot communicate with API server. System displays error message. |
| **Business Rules** | B12, B13 |
| **Non-Functional Requirement** | N/A |

*Table 2.3.1.10-1: Remove Place*

**Screen design**

*****Figure 2.3.1.10-2: Remove Place*

57

**2.3.1.11. View all Places**

****

*Figure 2.3.1.11-1: View all Places activity diagram*

|  |  |  |  |
| --- | --- | --- | --- |
| **UC ID and Name** | **UC-11 View all Places** | | |
| **Create by** | ThaoVTP | **Date created** | 04/06/2020 |
| **Primary Actors** | Admin | **Secondary Actors** | N/A |
| **Description** | Display a list of all Place in GoBoKi system. | | |
| **Priority** | Medium | | |
| **Trigger** | N/A | | |
| **Pre-Condition(s)** | ∙ Admin has a GoBoKi admin account.  ∙ Admin is currently logged into the website by admin account. | | |
| **Post**  **Condition(s)** | ∙ Table of added Place will be shows. | | |
| **Basic Flow** | 1. Login into website by admin account  2. Admin clicks on “Place manager” tab.  3. System shows all Place list.  4. System displays the list of Places, where each of them is presented with the following information:  ∙ Place name  ∙ City  ∙ Mail  ∙ Phone number  ∙ Category  ∙ Short Description  ∙ Status | | |

58

|  |  |
| --- | --- |
| **Alternative Flow** | N/A |
| **Exception Flow** | Cannot communicate with API server. System displays error message. |
| **Business Rules** | B12, B13 |
| **Non-Functional Requirement** | N/A |

*Table 2.3.1.11-1: View all Place*

**Screen design**

*****Figure 2.3.1.11-2: View all Place*

59

**2.3.1.12. Place Search**

*****Figure 2.3.1.12-1: Place search activity diagram*

|  |  |  |  |
| --- | --- | --- | --- |
| **UC ID and Name** | **UC-12 Place Search** | | |
| **Create by** | ThaoVTP | **Date created** | 09/06/2020 |
| **Primary Actors** | Admin | **Secondary Actors** | N/A |
| **Description** | Allow admin to search place if needed | | |
| **Priority** | Medium | | |
| **Trigger** | Admin want to login website | | |
| **Pre-Condition(s)** | ∙ Admin has an admin account in GoBoKi system.  ∙ Admin isn’t currently logged into the website.  ∙ There is at least 1 place in the place list. | | |
| **Post**  **Condition(s)** | When the normal flow completes successfully, Place by keyword will be displayed on the "Place manager" page. | | |
| **Basic Flow** | 1. On the Side bar, the admin clicks on “Place manager”. 2. System shows “Place manager” page. | | |

60

|  |  |
| --- | --- |
|  | 3. In “Place manager” page, Admin fills information of Place want search.  ∙ Place Name  ∙ Address  ∙ Choose City  ∙ Choose Place Type  4. Admin clicks on “Search” button.  5. System will show place list according to the search keyword, where each of them is presented with the following  information:  ∙ Place Name  ∙ City  ∙ Address  ∙ Mail  ∙ Phone Number  ∙ Category  ∙ Short Description  ∙ Status |
| **Alternative Flow** | N/A |
| **Exception Flow** | ∙ Input data at step 3 is conflict with business rule.  ∙ Cannot communicate with API server. System displays error message. |
| **Business Rules** | B11, B12, B13, B43 |
| **Non-Functional Requirement** | N/A |

*Table 2.3.1.12-1: Place Search*

**Screen design**

*****Figure 2.3.1.12-2: Place Search*

61

**2.3.1.13. Change Status Place**

****

*Figure 2.3.1.13-1: Change status place activity diagram*

|  |  |  |  |
| --- | --- | --- | --- |
| **UC ID and Name** | **UC-13 Change Status Place** | | |
| **Create by** | ThaoVTP | **Date created** | 09/06/2020 |
| **Primary Actors** | Admin | **Secondary Actors** | N/A |
| **Description** | Allows admin to enter activate or deactivate for a Place into GoBoKi system. | | |
| **Priority** | Medium | | |
| **Trigger** | N/A | | |
| **Pre-Condition(s)** | ∙ Admin has a GoBoKi admin account.  ∙ Admin is currently logged into the website by admin account. ∙ There is at least 1 Place in the Place list. | | |
| **Post**  **Condition(s)** | When the normal flow completes successfully, Place information will be changed from active to deactivate in the database. | | |
| **Basic Flow** | 1. Login into website by admin account  2. Admin clicks on “Place manager” tab.  3. System shows All Place list. | | |

62

|  |  |
| --- | --- |
|  | 4. Admin clicks on “Deactivate” button in each row.  5. Admin clicks on button Yes or No of notification message. 6. System displays a message to notify that “Place” has been deactivated. |
| **Alternative Flow** | N/A |
| **Exception Flow** | ∙ Admin clicks “No” button of the confirmation dialog (Step 5), system hides the dialog box, and the use case ends.  ∙ Cannot communicate with API server. System displays error message. |
| **Business Rules** | B12, B13 |
| **Non-Functional Requirement** | N/A |

*Table 2.3.1.13-1: Change Status Place*

**Screen design**

*****Figure 2.3.1.13-2 : Change Status Place*

63

**2.3.1.14. Add Games**

****

*Figure 2.3.1.14-1: Add games activity diagram*

|  |  |  |  |
| --- | --- | --- | --- |
| **UC ID and Name** | **UC-14 Add Games** | | |
| **Create by** | ThaoVTP | **Date created** | 04/06/2020 |
| **Primary Actors** | Admin | **Secondary Actors** | N/A |
| **Description** | Allow admin to add game into a Place in GoBoKi system. | | |
| **Priority** | Medium | | |
| **Trigger** | N/A | | |
| **Pre-Condition(s)** | ∙ Admin has a GoBoKi admin account.  ∙ Admin is currently logged into the website by admin account. | | |
| **Post**  **Condition(s)** | ∙ New game’s information will be saved in database. | | |

64

|  |  |
| --- | --- |
| **Basic Flow** | 1. Login into website by admin account  2. Admin clicks on “Game manager” tab.  3. In “Game page” clicks on “Add New Game” button. 4. System shows add new form in the dialog box.  5. Fills in:  ∙ Game name  ∙ Place name  ∙ Description  6. Admin clicks on “Save Game” button.  7. Admin clicks on button Yes or No of notification message. 8. System displays notification of add successfully. |
| **Alternative Flow** | N/A |
| **Exception Flow** | ∙ Input data at step 4 is conflict with business rule.  ∙ Admin clicks “No” button of the confirmation dialog (Step 7), system hides the dialog box, and the use case ends.  ∙ Cannot communicate with API server. System displays error message. |
| **Business Rules** | B12, B13, B24, B30, B31 |
| **Non-Functional Requirement** | N/A |

*Table 2.3.1.14-1: Add games*

**Screen design**

*****Figure 2.3.1.14-2: Add games*

65

*Figure 2.3.1.14-3: Add games*

66

**2.3.1.15. Edit Game**

****

*Figure 2.3.1.15-1: Edit game activity diagram*

|  |  |  |  |
| --- | --- | --- | --- |
| **UC ID and Name** | **UC-15 Edit Game** | | |
| **Create by** | ThaoVTP | **Date created** | 04/06/2020 |
| **Primary Actors** | Admin | **Secondary Actors** | N/A |
| **Description** | Allows admin to enter information and edit a game of a Place into GoBoKi system | | |
| **Priority** | Medium | | |
| **Trigger** | N/A | | |

67

|  |  |
| --- | --- |
| **Pre-Condition(s)** | ∙ Admin has a GoBoKi admin account.  ∙ Admin is currently logged into the website by admin account. ∙ There is at least 1 game in amusement Place. |
| **Post**  **Condition(s)** | ∙ Information of Place will be changed and saved in database and the table of list location will be shown. |
| **Basic Flow** | 1. Login into website by admin account  2. Admin clicks on “Game manager” tab.  3. System shows all games list.  4. Admin clicks on “Edit” button in each row.  5. Fills in.  6. Admin clicks on “Submit” button.  7. Admin clicks on button Yes or No of the notification message. 8. System displays edit successfully notification |
| **Alternative Flow** | N/A |
| **Exception Flow** | ∙ Input data at step 5 is conflict with business rule.  ∙ Admin clicks “No” button of the confirmation dialog (Step 7), system hides the dialog box, and the use case ends.  ∙ Cannot communicate with API server. System displays error message. |
| **Business Rules** | B12, B13, B16, B24, B30, B31 |
| **Non-Functional Requirement** | N/A |

*Table 2.3.1.15-1: Edit game*

68

**Screen design**

*****Figure 2.3.1.15-2: Edit games*

69

**2.3.1.16. Remove Game**

****

*Figure 2.3.1.16-1: Remove game activity diagram*

|  |  |  |  |
| --- | --- | --- | --- |
| **UC ID and Name** | **UC-16 Remove Game** | | |
| **Create by** | ThaoVTP | **Date created** | 04/06/2020 |
| **Primary Actors** | Admin | **Secondary Actors** | N/A |
| **Description** | Allows admin to enter information and delete a game of a park into GoBoKi system | | |
| **Priority** | Medium | | |
| **Trigger** | N/A | | |
| **Pre-Condition(s)** | ∙ Admin has a GoBoKi admin account.  ∙ Admin is currently logged into the website by admin account. ∙ There is at least 1 game in amusement park. | | |
| **Post**  **Condition(s)** | ∙ Game information not displayed into the "Park" page. ∙ Park information will be delete in database and the table of list location will be shows. | | |
| **Basic Flow** | 1. Login into website by admin account  2. Admin clicks on “Game manager” tab. | | |

70

|  |  |
| --- | --- |
|  | 3. System shows games list.  4. Admin clicks on “Remove” button in one row.  5. Admin clicks on button Yes or No notification message. 6. System displays a message to notify that “Game” has been deleted. |
| **Alternative Flow** | N/A |
| **Exception Flow** | ∙ Admin clicks “No” button of the confirmation dialog (Step 5), system hides the dialog box, and the use case ends.  ∙ Cannot communicate with API server. System displays error message. |
| **Business Rules** | B12, B13, B23, B24 |
| **Non-Functional Requirement** | N/A |

*Table 2.3.1.16-1: Remove game*

**Screen design**

*****Figure 2.3.1.16-2: Remove games*

71

**2.3.1.17. View all Games**

****

*Figure 2.3.1.17-1: View all game activity diagram*

|  |  |  |  |
| --- | --- | --- | --- |
| **UC ID and Name** | **UC-17 View all Games** | | |
| **Create by** | ThaoVTP | **Date created** | 04/06/2020 |
| **Primary Actors** | Admin | **Secondary Actors** | N/A |
| **Description** | Display a list all games of an amusement Place in the GoBoKi system. | | |
| **Priority** | Medium | | |
| **Trigger** | N/A | | |
| **Pre-Condition(s)** | ∙ Admin has a GoBoKi admin account.  ∙ Admin is currently logged into the website by admin account. | | |
| **Post**  **Condition(s)** | All games will be shows on screen. | | |
| **Basic Flow** | 1. Login into website by admin account  2. Admin clicks on “Game manager” tab.  3. System shows all Place list on “Game manager” tab. 4. System displays the list of games, where each of them is presented with the following information:  ∙ Game name  ∙ Description  ∙ Place name  ∙ Status | | |
| **Alternative Flow** | N/A | | |
| **Exception Flow** | Cannot communicate with API server. System displays error message. | | |

72

|  |  |
| --- | --- |
| **Business Rules** | B12, B13 |
| **Non-Functional Requirement** | N/A |

*Table 2.3.1.17-1: View all games*

**Screen design**

*****Figure 2.3.1.17-2: View all games*

73

**2.3.1.18. Game Search**

*****Figure 2.3.1.18-1: Game Search activity diagram*

|  |  |  |  |
| --- | --- | --- | --- |
| **UC ID and Name** | **UC-18 Game Search** | | |
| **Create by** | ThaoVTP | **Date created** | 09/06/2020 |
| **Primary Actors** | Admin | **Secondary Actors** | N/A |
| **Description** | Allow admin to search game if needed | | |
| **Priority** | Medium | | |
| **Trigger** | Admin want to login website | | |
| **Pre-Condition(s)** | ∙ Admin has an admin account in GoBoKi system.  ∙ Admin isn’t currently logged into the website.  ∙ There is at least 1 game in the game list. | | |
| **Post**  **Condition(s)** | When the normal flow completes successfully, Game by keyword will be displayed on the "Game manager" page. | | |
| **Basic Flow** | 1. On the Side bar, the admin clicks on “Game manager”. 2. System shows “Game manager” page. | | |

74

|  |  |
| --- | --- |
|  | 3. In “Game manager” page, Admin fills information of Game want search.  ∙ Game Name  ∙ Place Name  4. Admin clicks on “Search” button.  5. System will show all user list according to the search keyword, where each of them is presented with the following information:  ∙ Game Name  ∙ Description  ∙ Place Name  ∙ Status |
| **Alternative Flow** | N/A |
| **Exception Flow** | ∙ Input data at step 3 is conflict with business rule.  ∙ Cannot communicate with API server. System displays error message. |
| **Business Rules** | B11, B12, B13, B43 |
| **Non-Functional Requirement** | N/A |

*Table 2.3.1.18-1: Game Search*

**Screen design**

*****Figure 2.3.1.18-2: Games search*

75

**2.3.1.19. Change Status Game**

****

*Figure 2.3.1.19-1: Change status game activity diagram*

|  |  |  |  |
| --- | --- | --- | --- |
| **UC ID and Name** | **UC-19 Change Status Game** | | |
| **Create by** | ThaoVTP | **Date created** | 04/06/2020 |
| **Primary Actors** | Admin | **Secondary Actors** | N/A |
| **Description** | Allows admin to enter activate or deactivate a game of a Place into GoBoKi system | | |
| **Priority** | Medium | | |
| **Trigger** | N/A | | |
| **Pre-Condition(s)** | ∙ Admin has a GoBoKi admin account.  ∙ Admin is currently logged into the website by admin account. ∙ There is at least 1 game in the game list. | | |
| **Post**  **Condition(s)** | ∙ Game information not displayed into the "Game" page. ∙ Place information will be changed from active to deactivate in the database. | | |

76

|  |  |
| --- | --- |
| **Basic Flow** | 1. Login into website by admin account  2. Admin clicks on “Game manager” tab.  3. System shows all games list.  4. Admin clicks on “Deactivate” button in each row.  5. Admin clicks on button Yes or No notification message. 6. System displays a message to notify that “Game” has been deactivated. |
| **Alternative Flow** | N/A |
| **Exception Flow** | ∙ Admin clicks “No” button of the confirmation dialog (Step 5), system hides the dialog box, and the use case ends.  ∙ Cannot communicate with API server. System displays error message. |
| **Business Rules** | B12, B13, B23, B24 |
| **Non-Functional Requirement** | N/A |

*Table 2.3.1.19-1: Change status games*

**Screen design**

*****Figure 2.3.1.19-2: Change status games*

77

**2.3.1.20. Add Ticket Type**

****

*Figure 2.3.1.20-1: Add ticket type activity diagram*

|  |  |  |  |
| --- | --- | --- | --- |
| **UC ID and Name** | **UC-20 Add Ticket Type** | | |
| **Create by** | ThaoVTP | **Date created** | 03/06/2020 |
| **Primary Actors** | Admin | **Secondary Actors** | N/A |
| **Description** | Admin wants to add a new ticket type into game. | | |
| **Priority** | High | | |
| **Trigger** | N/A | | |
| **Pre-Condition(s)** | ∙ Admin has a GoBoKi admin account.  ∙ Admin is currently logged into the website by admin account. ∙ There is at least 1 game of Place in GoBoKi system | | |

78

|  |  |
| --- | --- |
| **Post**  **Condition(s)** | ∙ Notification adding Ticket successfully.  ∙ New Ticket type save in database of GoBoKi system. |
| **Basic Flow** | 1. Login into website by admin account  2. Admin clicks on “Ticket manager” tab.  3. In “Ticket page” clicks on “Add Ticket Type” button. 4. Fills in:  ∙ Ticket Name  ∙ Effective Time  ∙ Description  ∙ Cancel Policy  ∙ Reservation Information  ∙ Conversion Method  5. Admin clicks on “Save Ticket Type” button.  6. Admin clicks on button Yes or No notification message. 7. System displays notification of add successfully. |
| **Alternative Flow** | N/A |
| **Exception Flow** | ∙ Input data at step 4 is conflict with business rule.  ∙ Admin clicks “No” button of the confirmation dialog (Step 6), system hides the dialog box, and the use case ends.  ∙ Cannot communicate with API server. System displays error message. |
| **Business Rules** | B12, B13, B21, B34, B37 |
| **Non-Functional Requirement** | N/A |

*Table 2.3.1.20-1: Add ticket type*

**Screen design**

*****Figure 2.3.1.20-2: Add ticket type*

79

*Figure 2.3.1.20-3: Add ticket type*

80