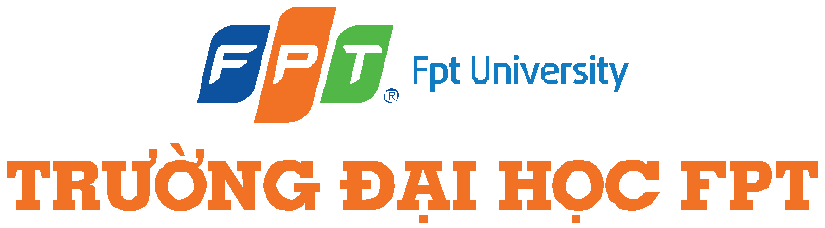
|  |  |
| --- | --- |
|  | **MINISTRY OF EDUCATION AND TRAINING** |

|  |
| --- |
| **FPT UNIVERSITY** |
| Capstone Project Document |
| Photographer Booking System |

|  |  |
| --- | --- |
| **FA20SE21** | |
| **Group Members** | Trần Thiên Thảo – Leader – SE62888  Bồ Công Đạt – Member – SE63154  Đào Sỹ Trung Kiên – Member – SE63208  Trương Ngọc Mỹ - Member – SE62050 |
| **Supervisor** | Lại Đức Hùng |
| **Ext Supervisor** | N/A |
| **Capstone Project code** | PBS |

* HoChiMinh September 2020 -



**CAPSTONE PROJECT REGISTER**

Class: Duration time: from …. …………To /….

(\*) Profession: <Software Engineer> Specialty: <JS> x

(\*) Kinds of person make registers: Lecturer x

Student

1. Register information for supervisor (if have)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Full name** | **Phone** | **E-Mail** | **Title** |
| Supervisor 1 | Lại Đức Hùng |  | HungLD5@fe.edu.vn | Mr. |

2. Register content of Capstone Project

(\*) 2.1. Capstone Project name:

English: Photographer booking system (PBS) Vietnamese: Tìm và đặt lịch thợ chụp hình.

Abbreviation:

This application will help you to find the best photographer with the suitable price and schedule. It makes the photographer more productive by making the booking process easier. Customers can also know exactly when he/she is available and they can book when they want.

Proposed Solution: PBS is a mobile application. It is an easy new way to find and book affordable photographers near you. PBS uses GPS and calendar syncing to connect you to local talent at incredibly friendly rates. Book one of the customer-rated photographers on demand or reserve them for a later date.

(\*) 2.2. Main proposal content (including result and product)

a) Theory and practice (document):

- . Student should apply the software development process and the UML

- . Software artifacts include User Requirement, Software Requirement Specification, Architecture Design, Detail Design, System Implementation, and Testing Document, Installation Guide, sources code, and deployable software packages

- . 3 tiers should be applied

b) Program:

Main functions:

* User functions:
* Find a photographer
* Book photographer
* Rate the service
* …
* Photographer functions
* Manage request
* Manage profile
* Manage schedule
* ….
* Admin functions
* Manage user
* Manage feedback
* …..

Non-functional requirements

* Usability
* Provide a convenient way to interact with system.
* Screen layout and navigation are clear and easy to use.
* Interface are simple and clear.
  + The system usability is easy to use that needs less than 2 days of training for company staffs to use the system.
* Reliability
* System notification success rate is less than 2 failed notifications per 1000 sent.
* Availability
* User connects to internet to login into the application.
* User uses offline after login and connect internet to push data to the server.
* The system updates every day at 00:01 am.
* Security
* Private: Each role of user has a specific permission to interact with the system.
* Only admin can manage user and feedback of user.
* All input data are validated before saving to database.
* Maintainability
* The code is easy to maintain and upgrade.
* Maintain the whole system every 1 months.
* Accidental problem is resolved within acceptable time.

c) Other products:

3. Other comment (propose all relative thing if have)

HCM city, date 14/08/2020

**Supervisor (If have) On behalf of Registers**

*(Sign and full name) (Sign and full name)*

**Table of Contents**

[Acknowledgement 13](#_Toc62259190)

[Definition and Acronyms 13](#_Toc62259191)

[I. Project Introduction 14](#_Toc62259192)

[1. Overview 14](#_Toc62259193)

[1.1 Project Information 14](#_Toc62259194)

[1.2 Project Team 14](#_Toc62259195)

[2. Product Background 14](#_Toc62259196)

[3. Existing Systems 15](#_Toc62259197)

[4. Business Opportunity 15](#_Toc62259198)

[5. Software Product Vision 15](#_Toc62259199)

[6. Project Scope & Limitations 16](#_Toc62259200)

[6.1 Major Features 16](#_Toc62259201)

[6.2 Limitations & Exclusions 16](#_Toc62259202)

[II. Project Management Plan 17](#_Toc62259203)

[1. Overview 17](#_Toc62259204)

[2. Management Approach 17](#_Toc62259205)

[2.1 Project Process 17](#_Toc62259206)

[2.2 Quality Management 18](#_Toc62259207)

[3. Master Schedule 18](#_Toc62259208)

[4. Project Organization 18](#_Toc62259209)

[4.1 Team & Structures 18](#_Toc62259210)

[4.2 Roles & Responsibilities 18](#_Toc62259211)

[5. Project Communication 18](#_Toc62259212)

[5.1 Communication Plan 18](#_Toc62259213)

[5.2 External Interface 18](#_Toc62259214)

[6. Configuration Management 18](#_Toc62259215)

[6.1 Tools & Infrastructures 18](#_Toc62259216)

[6.2 Document Management 18](#_Toc62259217)

[6.3 Source Code Management 18](#_Toc62259218)

[III. Software Requirement Specification 19](#_Toc62259219)

[1. Overall Description 19](#_Toc62259220)

[1.1 Product Overview 19](#_Toc62259221)

[1.2 Business Rules 19](#_Toc62259222)

[2. User Requirements 19](#_Toc62259223)

[2.1 Overview 19](#_Toc62259224)

[2.2 List of use case 20](#_Toc62259225)

[3. Functional Requirements 59](#_Toc62259226)

[3.1 System Functional Overview 59](#_Toc62259227)

[4. Non-Functional Requirements 60](#_Toc62259228)

[5. Other Requirements 60](#_Toc62259229)

[IV. Software Design Document 60](#_Toc62259230)

[1. Overall Description 60](#_Toc62259231)

[2. System Architecture Design 60](#_Toc62259232)

[2.1 Overall Architecture 60](#_Toc62259233)

[2.2 System Architecture 61](#_Toc62259234)

[2.3 Package Diagram 61](#_Toc62259235)

[3. System Detailed Design 61](#_Toc62259236)

[3.1 Overall 61](#_Toc62259237)

[4. Class Specifications 71](#_Toc62259238)

[5. Data & Database Design 71](#_Toc62259239)

[5.1 Database Design 71](#_Toc62259240)

[6. Algorithm 72](#_Toc62259241)

[6.1 Multiple-factors weighted sorting 72](#_Toc62259242)

[6.2 Naive Bayes Classifier 75](#_Toc62259243)

**List of Figures**

[Figure 1 Feature Mind Map 8](#_Toc62259608)

[Figure 2 Project process 9](#_Toc62259609)

[Figure 3 Overview Use Case Diagram 12](#_Toc62259610)

[Figure 4 <Guest> Register 12](#_Toc62259611)

[Figure 5 <Guest> Login 15](#_Toc62259612)

[Figure 6 <Guest> Forgot password 17](#_Toc62259613)

[Figure 7 <Customer> Overview 19](#_Toc62259614)

[Figure 8 <Customer> Book photographer 20](#_Toc62259615)

[Figure 9 <Customer> Get bookings 22](#_Toc62259616)

[Figure 10- <Customer> Make report 25](#_Toc62259617)

[Figure 11 <Customer> Submit cancellation 28](#_Toc62259618)

[Figure 12 <Photographer> Overview 31](#_Toc62259619)

[Figure 13 <Photographer> Accept booking 31](#_Toc62259620)

[Figure 14 <Photographer> Reject booking 34](#_Toc62259621)

[Figure 15 <Admin> Overview 37](#_Toc62259622)

[Figure 16 <Admin> Get cancellation 38](#_Toc62259623)

[Figure 17 <Admin> Approve cancellation 39](#_Toc62259624)

[Figure 18 <Admin> Get report 41](#_Toc62259625)

[Figure 19 <Admin> Block user 42](#_Toc62259626)

[Figure 20 <Admin> Unblock user 44](#_Toc62259627)

[Figure 21 <Admin> Warn user 46](#_Toc62259628)

[Figure 22 <<system>> Handler Overview 47](#_Toc62259629)

[Figure 23 <<system>> Handler make scheduling notification 48](#_Toc62259630)

[Figure 24 <<system>> Handler make weather warning 49](#_Toc62259631)

[Figure 25 ER Diagram 51](#_Toc62259632)

[Figure 26 Overall Architecture 52](#_Toc62259633)

[Figure 27 Class Diagram 53](#_Toc62259634)

[Figure 28 Sequence Diagram - Block user 54](#_Toc62259635)

[Figure 29 Sequence Diagram - Unblock User 55](#_Toc62259636)

[Figure 30 Sequence Diagram - warn user 56](#_Toc62259637)

[Figure 31 Sequence Diagram - approve cancellation 57](#_Toc62259638)

[Figure 32 Activity Diagram – Book photographer 58](#_Toc62259639)

[Figure 33 Activity Diagram – Edit Request 59](#_Toc62259640)

[Figure 34 Activity Diagram - Cancel booking 60](#_Toc62259641)

[Figure 35 Activity Diagram - Accept booking 61](#_Toc62259642)

[Figure 36 Activity Diagram - Reject booking 62](#_Toc62259643)

[Figure 37 State Machine Diagram - Booking 63](#_Toc62259644)

[Figure 38 Database Design 63](#_Toc62259645)

[Figure 39 Flowchart - Multiple factors sorting 66](#_Toc62259646)

**List of Tables**

[Table 1 Definition and Acronyms 4](#_Toc62259647)

[Table 2 Team Members 5](#_Toc62259648)

[Table 3 <Guest> Register Specification 14](#_Toc62259649)

[Table 4 <Guest> Register Specification 15](#_Toc62259650)

[Table 5 <Guest> Forgot password specification 17](#_Toc62259651)

[Table 6 <Customer> Book photographer specification 21](#_Toc62259652)

[Table 7 <Customer> Get booking specification 24](#_Toc62259653)

[Table 8 <Admin> Make report specification 27](#_Toc62259654)

[Table 9 <Customer> Submit cancellation specification 29](#_Toc62259655)

[Table 10 <Photographer> Accept booking specification 33](#_Toc62259656)

[Table 11 <Photographer> Reject booking specification 35](#_Toc62259657)

[Table 12 <Admin> Get cancellation specification 38](#_Toc62259658)

[Table 13 <Admin> Approve cancellation 40](#_Toc62259659)

[Table 14 <Admin> Get report 41](#_Toc62259660)

[Table 15 <Admin> Block user specification 43](#_Toc62259661)

[Table 16 <Admin> Unblock user specification 44](#_Toc62259662)

[Table 17 <Admin> Warn user specification 46](#_Toc62259663)

[Table 18 <System Handler> Make scheduling notification specification 48](#_Toc62259664)

[Table 19 <System Handler> Make weather warning specification 49](#_Toc62259665)

# Acknowledgement

# Definition and Acronyms

|  |  |
| --- | --- |
| **Acronym** | **Definition** |
| PBS | Photographer booking system |
| AWS | Amazon Web Services |
| BR | Business Rule |
| ERD | Entity Relationship Diagram |
| SRS | Software Requirement Specification |
| UC | Use Case |
| API | Application Program Interface |

Table 1 Definition and Acronyms

# I. Project Introduction

## 1. Overview

### 1.1 Project Information

* Project name: Photographer Booking System
* Project code: PBS
* Group name: FA20SE21
* Software type: Mobile Application

### 1.2 Project Team

#### a. Supervisor

|  |  |  |  |
| --- | --- | --- | --- |
| **Full Name** | **Email** | **Phone Number** | **Title** |
| Lại Đức Hùng | Hungld5@fe.edu.vn | 0976 710580 | Lecturer |

#### b. Team Members

|  |  |  |  |
| --- | --- | --- | --- |
| **Full Name** | **Email** | **Mobile** | **Role** |
| Trần Thiên Thảo | Thaottse62888@fpt.edu.vn | 0903 917 982 | Leader |
| Bồ Công Đạt | Datbcse63154@fpt.edu.vn | 0915 223 623 | Member |
| Đào Sỹ Trung Kiên | Kiendstse63208@fpt.edu.vn | 0769 606 867 | Member |
| Trương Ngọc Mỹ | Mytnse62050@fpt.edu.vn | 0906 969 071 | Member |

Table 2 Team Members

## 2. Product Background

Despite the ever-increasingly fast-paced development of science and technology, there are still difficulties in the connections between customers and photographers.

As for customers:

* They find it difficult to contact skilful photographers living nearby or near where they want to shoot.
* They cannot view photographers’ working schedule, so they are not able to book the ones they want.
* Without viewing photographers’ albums beforehand, they are not sure if photographers’ style meets their requirements before booking.
* They are unlikely to find the appropriate rates quickly.

As for photographers:

* They are likely to miss potential customers living nearby or near their current shooting sites because customers do not know where they are and how to contact them.
* Applications that allow them to arrange a working schedule and receive bookings from customers are scarce.

## 3. Existing Systems

## 4. Business Opportunity

The model that we want to build is not the one emerging in the current market, which freelancers seek websites such as vLance, FreeLancerViet in Vietnam or foreign ones such as Upwork, PeoplePerHour and so on. Despite their long-time existence, they still have limitations, which is a huge opportunity for the research team. Researching comparison websites and users’ feedback, the team realized that most job search websites allow customers to hire photographers, but they cannot arrange shooting time effectively. In addition, photography has various branches such as cosplay photography, portrait photography, product photography, etc. Each of these branches would have its techniques, which means that shooting prices vary. The team would like to improve the drawbacks of those job-seeking websites via this project.

In fact, there is also a need for contact between customers and photographers to help minimize difficulties in finding the most appropriate one. With the team’s product, customers could save a lot of time in searching and thanks to other users’ feedback, they no longer need to ask for advice from friends, family or social networks. Customers would be able to view photographers’ working schedule and then easily arrange desired shooting time. Photographers would not miss potential customers and have the chance to market their abilities via profile and users’ feedback. They could also be more flexible in their working time management.

## 5. Software Product Vision

Intending to help customers easily contact skilful photographers, the researchers came up with the idea of building two mobile applications, which serve as a platform for connecting customers and photographers and vice versa.

All things considered, this project is conducted to provide customers with a faster, more convenient and effective way to contact photographers. They now can view photographers’ working schedules and make their reservations easily without double-booking. Similarly, photographers can also efficiently arrange their schedule and manage their bookings.

Thanks to those applications, customers can reach the most appropriate photographers based on criteria such as shooting time, rates, and feedback from other users. Customers would also be provided with a pricing list and can avoid being overcharged. On the other hand, photographers can easily get booked, upload their album, efficiently manage their working schedule and develop their brand thanks to users’ feedback.

## 6. Project Scope & Limitations

### 6.1 Major Features

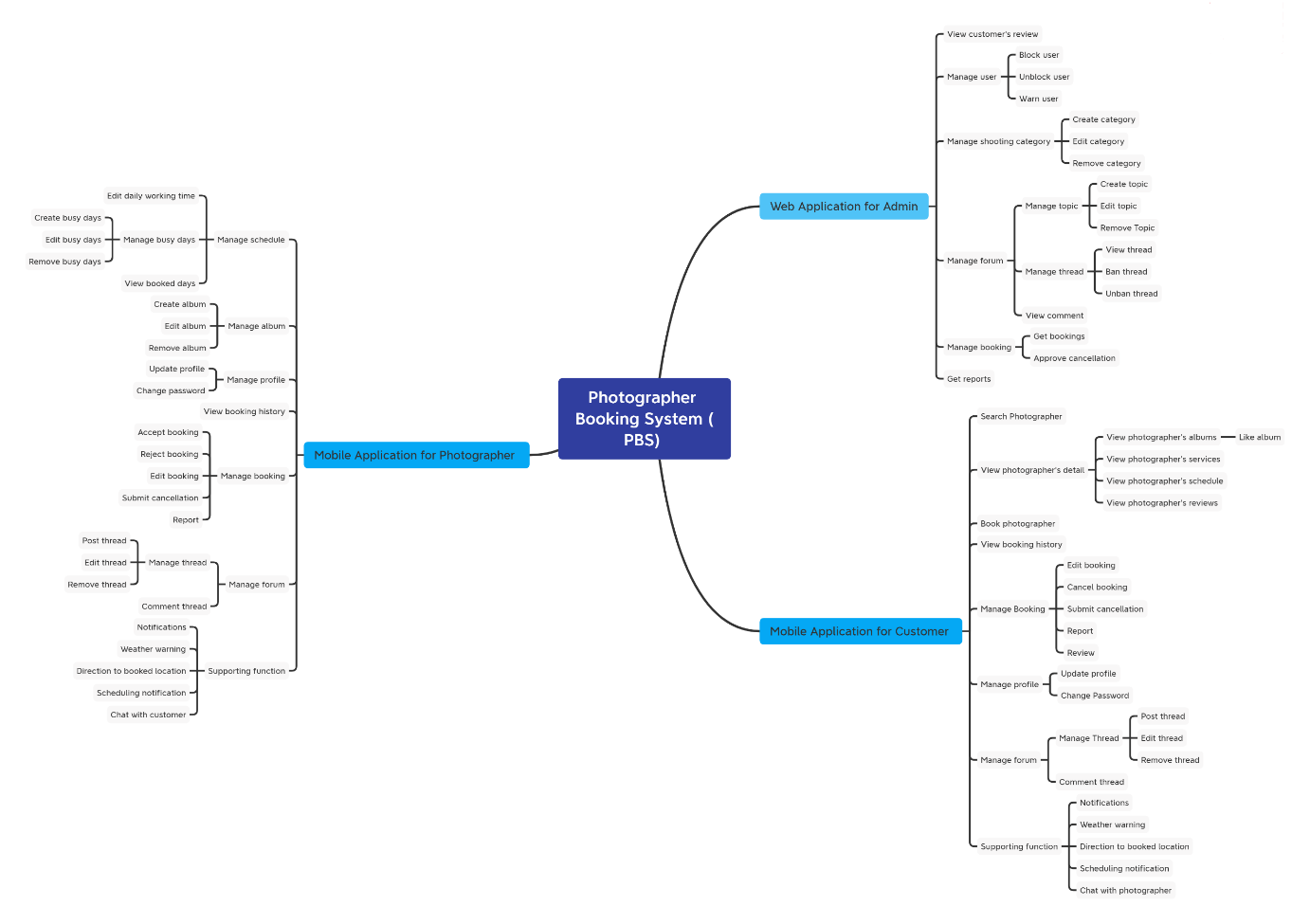


Figure 1 Feature Mind Map

### 6.2 Limitations & Exclusions

# II. Project Management Plan

## 1. Overview

## 2. Management Approach

Scrum model would be implemented in the development of this project because it is appropriate for the process.

### 2.1 Project Process

We choose Scrum model to do this project because of the following reasons:

* The feature can be changed during the developing process to be suitable with the Scrum model we can adapt to change it easily and help reduce risk.
* Scrum allows us to learn new frameworks or technology and develop projects at the same time.
* Because dividing the project allows the team to check partial inspection, identify and fix problems faster.

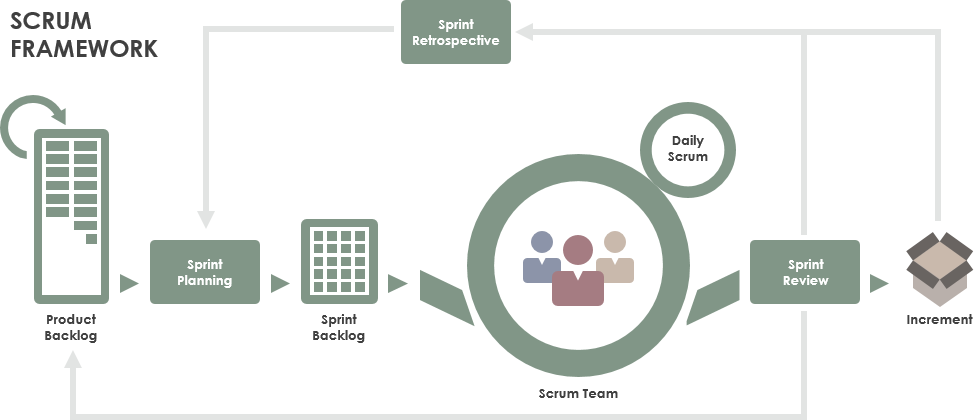
*Reference:* [*https://www.scrum.org/resources/what-is-scrum*](https://www.scrum.org/resources/what-is-scrum)

Figure 2 Project process

*Figure SEQ Figure \\* ARABIC 1 Project process*

### 2.2 Quality Management

## 3. Master Schedule

## 4. Project Organization

### 4.1 Team & Structures

*Figure SEQ Figure \\* ARABIC 2 Project organization*

### 4.2 Roles & Responsibilities

## 5. Project Communication

### 5.1 Communication Plan

### 5.2 External Interface

## 6. Configuration Management

### 6.1 Tools & Infrastructures

### 6.2 Document Management

*https://github.com/thienthao/pbs/tree/master/Documents*

*https://drive.google.com/drive/folders/1A--J0H-8uMiggtnQg93W7fExSaF\_4xsl*

### 6.3 Source Code Management

*https://github.com/thienthao/pbs*

# III. Software Requirement Specification

## 1. Overall Description

### 1.1 Product Overview

### 1.2 Business Rules

*Figure SEQ Figure \\* ARABIC 1 Context Diagram*

## 2. User Requirements

### 2.1 Overview

#### a. Use Case Diagram

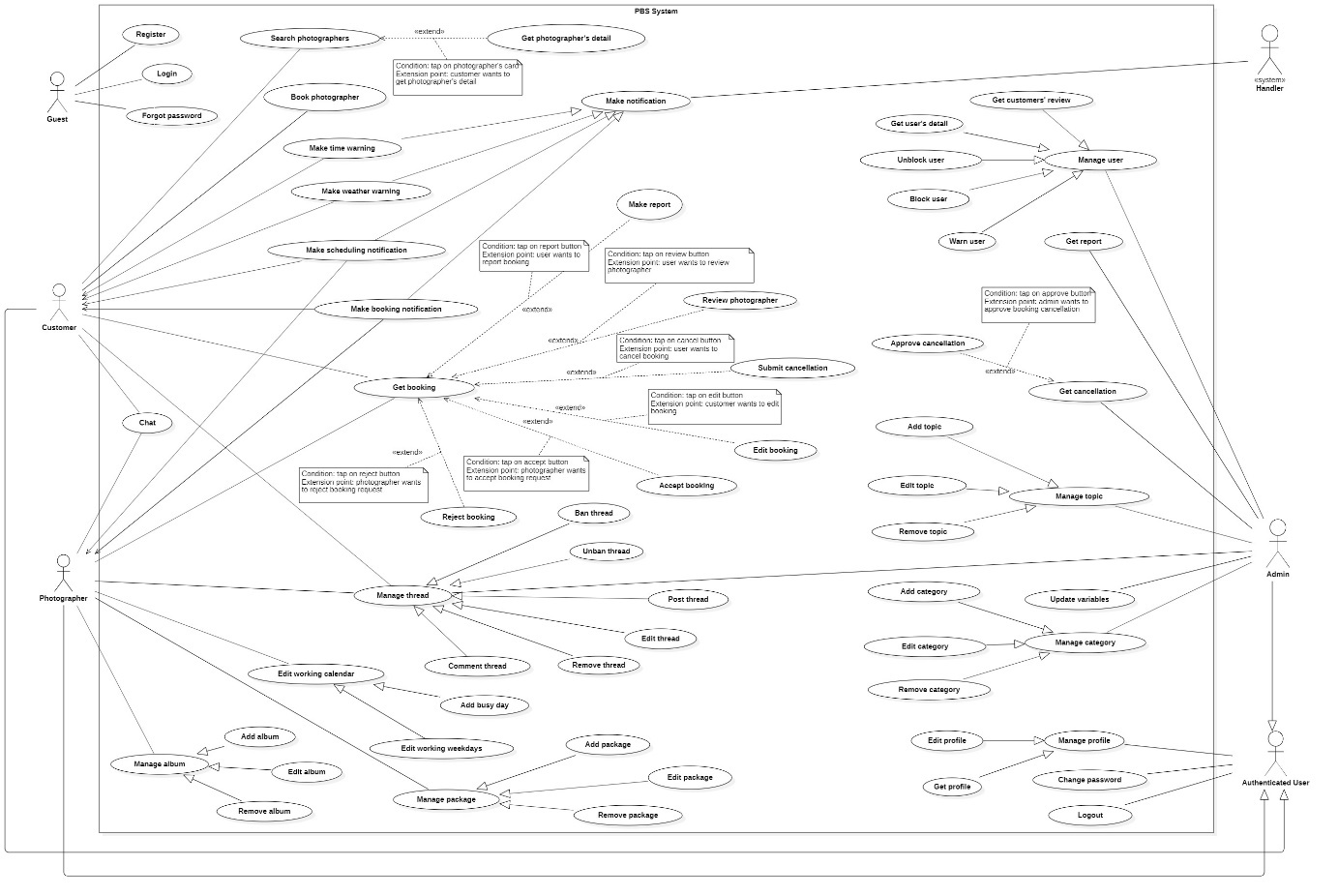


Figure 3 Overview Use Case Diagram

#### b. System Actors

#### c. Use Cases List

### 2.2 List of use case

#### 2.2.1 <Guest> Overview use case

##### a. <Guest> Register

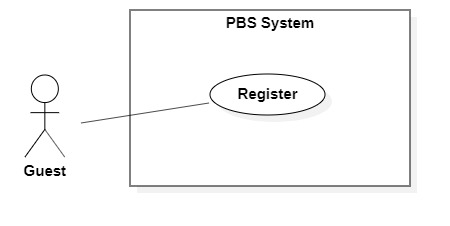


Figure 4 <Guest> Register

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| UC ID and Name: | **UC – 01 Register** | | | | |
| Created By: | Trần Thiên Thảo | | Date Created: | 08/10/2020 | |
| Primary Actor: | Guest | | Secondary Actors: |  | |
| Trigger: | The Guest sends a request to register an account. | | | | |
| Description: | This use case allows Guest to register an account in the system. | | | | |
| Preconditions: | User is not logged in. | | | | |
| Post-conditions: | **Success:** Guest successfully registered an account.  **Fail:** System shows error message. | | | | |
| Normal Flow: | **Step** | **Actor Action** | | | **System Response** |
| 1 | Guest goes to register view | | | System requires information from Guest:  - “Họ & tên”: text input, required.  - “Tên đăng nhập”: text input, required  - “Mật khẩu”: password input, required.  - “Xác nhận mật khẩu”: password input, required.  - “Số điện thoại”: phone number input.  - “Email”: text input, required. |
| 2 | Guest inputs required information | | |  |
| 3 | Guest send request to register in the system | | | The system validates data, outputs message “Đăng ký thành công”  [Exception 1]  [Exception 2]  [Exception 3]  [Exception 4] |
| Alternative Flows: | N/A | | | | |
| Exceptions: | **No** | **Cause** | | | **System Response** |
| 1 | Required fields are empty. | | | System shows error message:  “Thông tin này là bắt buộc” |
| 2 | Password and confirm password are not matched. | | | System shows error message:  “Mật khẩu không khớp”. |
| 3 | Required fields are not valid. | | | System shows error message corresponds to the field:  Username:  “Tên đăng nhập phải từ 8 - 20 kí tự.”  “Không thể có kí tự "\_" hoặc "." ở đầu hoặc cuối.”  Email:  “Email phải theo định dạng. Ví dụ: pbs@fpt.edu.vn”  Password:  “Tối thiểu 8 kí tự, 1 chữ cái, 1 số và 1 kí tự đặc biệt.” |
| 4 | Required information exists in the system (username, email). | | | System shows error message corresponds to the field:  “Tên đăng nhập đã có người sử dụng. Vui lòng nhập tên khác.”  “Email đã tồn tại.” |
| Priority: | Medium | | | | |
| Frequency of Use: | Usually | | | | |
| Business Rules: | * Needed information are: username, password, phone number, email, full name. * Username format should be 8-20 characters long, starts with character, underscore or dot can’t be at the end or beginning. Username must not contain special characters. * Password format should be a minimum eight characters, contains at least one number, one letter and one special character. * Password must be encrypted. * Fullname must not contains number, dot, underscore, special characters (except Vietnamese name’s characters). * Email should be in email format. * Phone should be in Vietnamese format. * All information must not be empty when submit. * After registering successfully, navigate back to login view. User account will have status unenabled. User will receive verification token through the registered email, after user clicks on the link the account will be enabled. * Verification token expired after 24 hours. | | | | |
| Other Information: | N/A | | | | |
| Assumptions: | N/A | | | | |

Table 3 <Guest> Register Specification

##### b. <Guest> Login

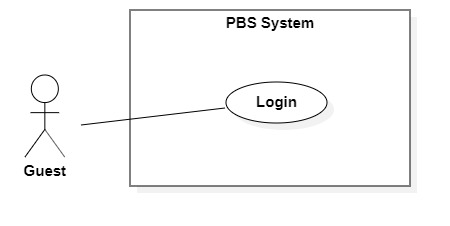


Figure 5 <Guest> Login

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| UC ID and Name: | **UC – 02 Login** | | | | |
| Created By: | Trần Thiên Thảo | | Date Created: | 08/10/2020 | |
| Primary Actor: | Guest | | Secondary Actors: |  | |
| Trigger: | The Guest sends a request to login into the system. | | | | |
| Description: | This use case allows Guest to login into the system. | | | | |
| Preconditions: | User is not logged in. | | | | |
| Post-conditions: | **Success**: Guest successfully logged into the system.  **Fail:** System shows error message. | | | | |
| Normal Flow: | **Step** | **Actor Action** | | | **System Response** |
| 1 | Guest goes to login view | | | System requires information from Guest:  - “Tên đăng nhập”: text input, required  - “Mật khẩu”: password input, required |
| 2 | Guest inputs required information | | |  |
| 3 | Guest sends request to login into the system | | | System validates data, outputs message “Đăng nhập thành công”  [Exception 1]  [Exception 2] |
| Alternative Flows: | N/A | | | | |
| Exceptions: | **No** | **Cause** | | | **System Response** |
| 1 | Required fields are empty | | | System shows error message: “Thông tin này là bắt buộc” |
| 2 | Username or password is incorrect | | | System shows error message: “Thông tin đăng nhập không chính xác” |
| Priority: | Medium | | | | |
| Frequency of Use: | Usually | | | | |
| Business Rules: | * User login using username and password. * Username and password format match with the one they have registered. * Encrypt inputted password and compare to password in database. * New Json web token will be returned to client side. * New user session will be created. * After login successfully, navigate user to Home view. | | | | |
| Other Information: | N/A | | | | |
| Assumptions: | N/A | | | | |

Table 4 <Guest> Register Specification

##### c. <Guest> Forgot password

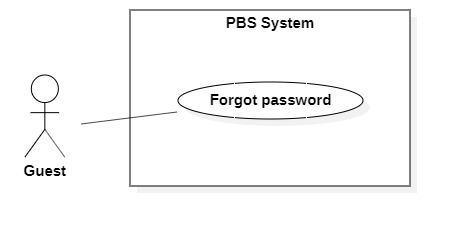


Figure 6 <Guest> Forgot password

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| UC ID and Name: | **UC – 03 Forgot Password** | | | | |
| Created By: | Trần Thiên Thảo | | Date Created: | 01/01/2021 | |
| Primary Actor: | Authenticated User | | Secondary Actors: |  | |
| Trigger: | User wants to recover password. | | | | |
| Description: | This use case allows User to recover password. | | | | |
| Preconditions: | User has account in the system. | | | | |
| Post-conditions: | **Success:** User has new password.  **Fail:** System shows error message. | | | | |
| Normal Flow: | **Step** | **Actor Action** | | | **System Response** |
| 1 | User clicks “Forgot password” button. | | | System shows “Forget password” detail with following information:  - “Email\*”: text  - “Xác nhận”: button |
| 2 | User inputs required information and send password recovery request. | | | System validates information, send password recovery request to the system and outputs message “Gửi yêu cầu thành công. Yêu cầu của sẽ được xử lý và phản hồi thông qua email này”.  [Exception 1]  [Exception 2] |
| Exceptions: | **No** | **Cause** | | | **System Response** |
| 1 | Required fields are empty. | | | System shows error message:  “Thông tin này là bắt buộc” |
| 2 | The process runs into an internal error. | | | System shows error message: “Đã có lỗi xảy ra”. |
| Priority: | Medium | | | | |
| Frequency of Use: | Sometimes | | | | |
| Business Rules: | * Email is required to send a password recovery request. * Check if email belongs to a user. * First, send an email to verify that user wants to update password, after user clicks on that link, another email will be sent to user with the generated password. * New password is generated following password format in register use case | | | | |
| Other Information: | N/A | | | | |
| Assumptions: | N/A | | | | |

Table 5 <Guest> Forgot password specification

#### 2.2.2 <Customer> Overview Use Case

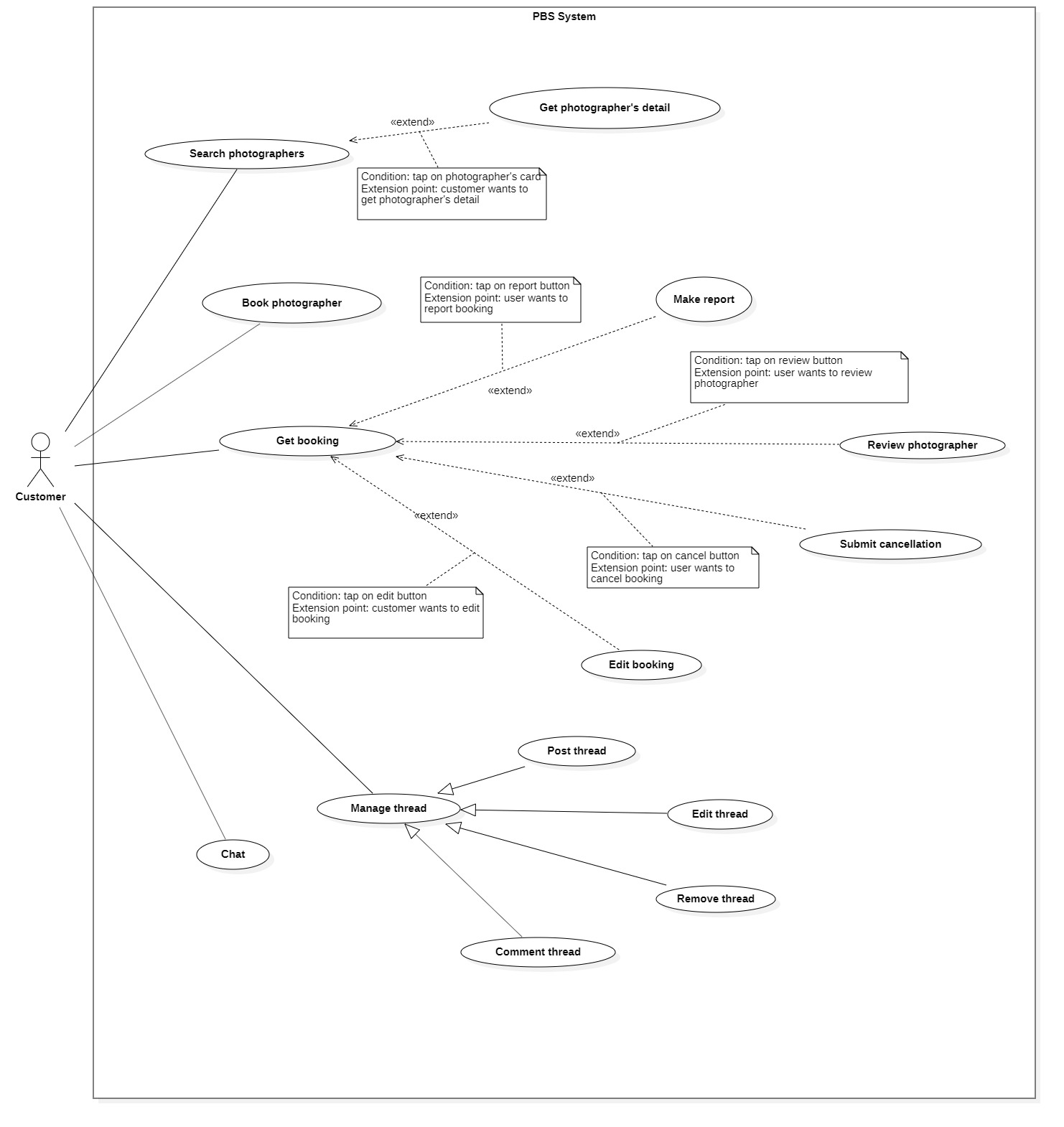


Figure 7 <Customer> Overview

##### <Customer> Book photographer

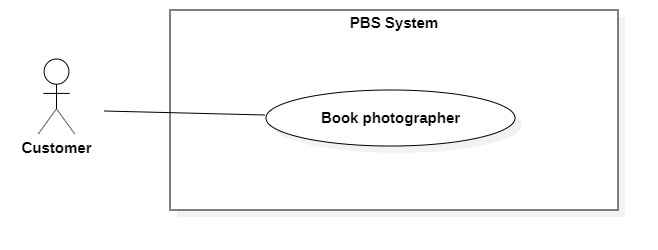


Figure 8 <Customer> Book photographer

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| UC ID and Name: | **UC – 06 Book photographer** | | | | |
| Created By: | Trần Thiên Thảo | | Date Created: | 08/10/2020 | |
| Primary Actor: | Customer | | Secondary Actors: |  | |
| Trigger: | The customer sends a request to make a booking. | | | | |
| Description: | This use case allows customers to make a booking for photographers. | | | | |
| Preconditions: | - Users must login into the system with the role Customer.  - The photographer must exist in the system with status is not blocked and is not deleted. | | | | |
| Post-conditions: | **Success:** A booking will be stored with status is “Pending” **Fail:** System outputs error message and sends to Customer. | | | | |
| Normal Flow: | **Step** | **Actor Action** | | | **System Response** |
| 1 | The customer goes to booking view | | | System returns the available photographer’s detail.  System requires information from customers:  - “Thời gian chụp”: date time picker, required  - “Thời gian nhận ảnh”: date time picker, required  - “Địa điểm”: map picker, required  - “Giao trả”: dropdownlist, required,  - “Gói dịch vụ: drop down list, required  - “Tổng cộng”: text |
| 2 | The customer inputs required information and send request to create new booking | | | - The system validates data  - The system adds new booking to the system, outputs “Đặt hẹn thành công” and send to photographer.  [Exception 1]  [Exception 2] |
| Alternative Flows: | N/A | | | | |
| Exceptions: | **No** | **Cause** | | | **System Response** |
| 1 | The customer inputs invalid data or leaves empty required fields | | | System outputs error message and send request to ask Customer to input correctly |
| 2 | The process run into an internal error | | | Error will be recorded in a log file. System outputs error message to customer. |
| Priority: | Medium | | | | |
| Frequency of Use: | Usually | | | | |
| Business Rules: | - Booking will be managed by the system with following status: PENDING, ONGOING, EDITING, DONE, REJECTED, CANCELLING\_PHOTOGRAPHER, CANCELLED\_PHOTOGRAPHER, CANCELLING\_CUSTOMER, CANCELLED\_CUSTOMER, CANCELLED\_ADMIN, EXPIRED.  - If photographer already have a booking in a time span, customer can’t book photographer in that time span.  - Booking information contains: package’s details, time and location details, editing deadline, photographer’s detail, customer’s detail, returning type.  - Package’s detail contains: package’s name, package’s description, package’s price, package’s services.  - Time and location detail contain: shooting date and time, shooting location.  - All information must not be empty.  - Shooting time and edit deadline must not be in the past. Editing deadline must be after shooting time at least 1 day.  - After booking successfully, a new booking will have PENDING status.  - If booking is failed, return fail message.  - After 24 hours, if photographer doesn’t accept or reject pending booking, status will become EXPIRED | | | | |
| Other Information: | N/A | | | | |
| Assumptions: | N/A | | | | |

Table 6 <Customer> Book photographer specification

##### <Customer> Get booking

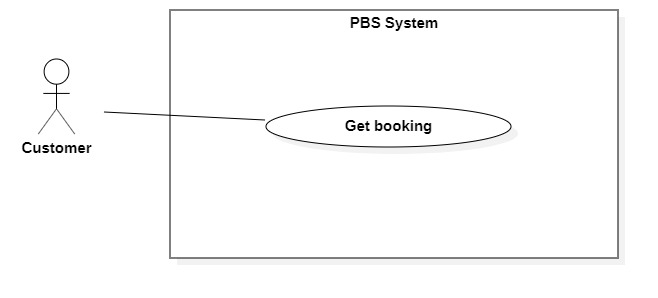


Figure 9 <Customer> Get bookings

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| UC ID and Name: | **UC – 07 Get booking** | | | | |
| Created By: | Trần Thiên Thảo | | Date Created: | 08/10/2020 | |
| Primary Actor: | Customer, Photographer | | Secondary Actors: |  | |
| Trigger: | Customer or Photographer sends a request to get booking detail. | | | | |
| Description: | This use case allows customers or photographers to get their booking detail. | | | | |
| Preconditions: | User has been authenticated as Customer / Photographer | | | | |
| Post-conditions: | **Success:** System displays detail of a booking.  **Fail:** System shows error message. | | | | |
| Normal Flow: | Step | Actor Action | | | System Response |
| 1 | Customer / Photographer sends a request to get detail of a booking. | | | System returns booking detail with following information:  - “Trạng thái”: text  - “Thời gian chụp”: text  - “Thời gian nhận ảnh”: text  - “Thời gian tác nghiệp dự kiến”: text  - “Địa điểm”:   * “Địa chỉ”: text * “Latitude”: double * “Longitude”: double   - “Thông tin photographer”:   * “Tên photographer”: text * “Điểm đánh giá”: number * “Số điện thoại photographer”: number * Avatar: image   - “Thông tin khách hàng”:   * “Tên khách hàng”: text * “Số điện thoại photographer”: number   “Avatar: image  - “Thông tin gói dịch vụ”:   * “Tên gói”: text * “Mô tả”: text * “Tổng cộng”: text * “Bao gồm các dịch vụ”: list of texts   - “Phương thức trả ảnh”: text  [Exception 1] |
| Alternative Flows: | N/A | | | | |
| Exceptions: | No | Cause | | | System Response |
| 1 | System run into internal error | | | System shows error message: “Đã có lỗi xảy ra” |
| Priority: | Medium | | | | |
| Frequency of Use: | Usually | | | | |
| Business Rules: | - The booking detail is displayed with following information: booking status, service’s time anticipation, edit deadline, time and location detail, photographer’s detail, package’s detail, returning type, rating and comment, qrcode scanner, cancelled reason, rejected reason, image returning link (after done).  - Booking has following status:   * PENDING * ONGOING * EDITING * DONE * REJECTED * CANCELLING\_CUSTOMER * CANCELLING\_PHOTOGRAPHER * CANCELLED\_CUSTOMER * CANCELLED\_PHOTOGRAPHER * CANCELLED\_ADMIN * EXPIRED   - Time and location details have the following information: shooting location, shooting date, shooting time, qr check in code.  - Customer’s information: customer’s avatar, customer’s name, customer’s phone.  - Photographer’s information: photographer’s avatar, photographer’s name, photographer’s rating count photographer’s phone,  - Rating’s information: customer’s avatar, customer’s rating, createdDate, comment. Only display rating section if booking’s status is DONE.  - Package has following information: package’s name, package’s description, package’s price, package’s services.  - Only show qrcode scanner if booking status is ONGOING.  - Only show rejected reason if booking status is REJECTED.  - Only show cancelled reason if booking status is CANCELLING\_CUSTOMER, CANCELLING\_PHOTOGRAPHER, CANCELLED\_CUSTOMER, CANCELLED\_PHOTOGRAPHER, CANCELLED\_ADMIN.  - Booking status will be changed with following actions:   * Photographer accepts: PENDING will become ONGOING. * Photographer rejects: PENDING will be come REJECTED. * Photographer sends request to change status from ONGOING to EDITING, condition is: all booking date have been checked in, in case photographer forgot to check in, they can send a confirmation request to customer, after customer confirms, status will be changed to EDITING. * Photographer sends request to change status from EDITING to DONE. * If booking status is ONGOING / EDITING will become CANCELLING\_PHOTOGRAPHER / CANCELLING\_CUSTOMER. * Admin approves cancellation: CANCELLING\_PHOTOGRAPHER / CANCELLING\_CUSTOMER will become CANCELLED\_PHOTOGRAPHER / CANCELLED\_CUSTOMER. * In case admin blocks user or cancel booking after view user’s report: status will become CANCELLED\_ADMIN. | | | | |
| Other Information: | N/A | | | | |
| Assumptions: | N/A | | | | |

Table 7 <Customer> Get booking specification

##### <Customer > Make report

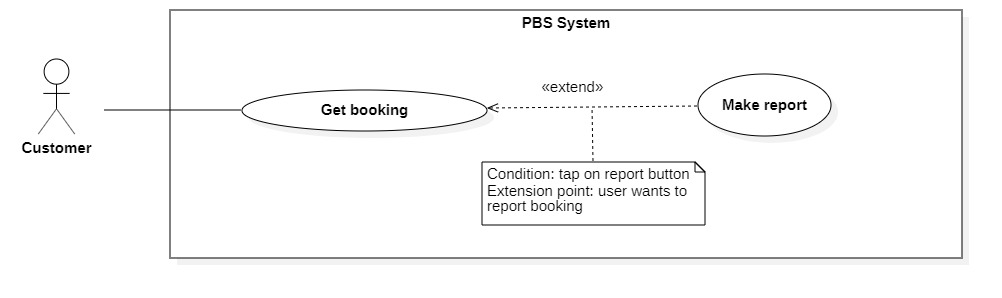


Figure 10- <Customer> Make report

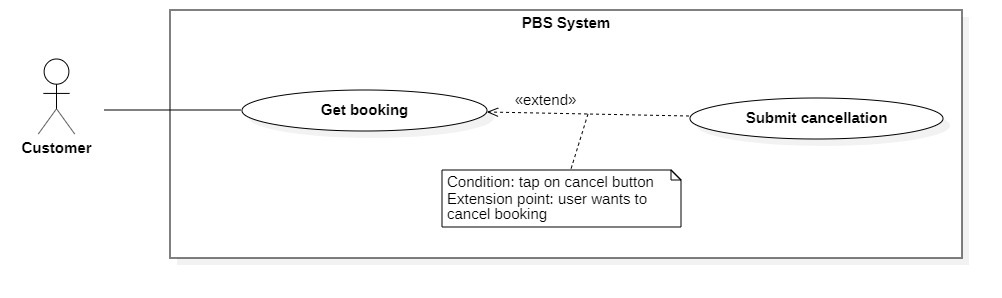
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| UC ID and Name: | **UC – 08 Make report** | | | | | |
| Created By: | Trần Thiên Thảo | | | Date Created: | 01/01/2021 | |
| Primary Actor: | Customer, Photographer | | | Secondary Actors: |  | |
| Trigger: | Customer wants to report booking with Photographer and vice versa. | | | | | |
| Description: | This use case allows Customer and Photographer to report the booking. | | | | | |
| Preconditions: | The User has been authenticated as Customer / Photographer. | | | | | |
| Post-conditions: | **Success:** The customer report photographer via booking successfully.  **Fail:** System shows error message | | | | | |
| Normal Flow: | **Step** | | **Actor Action** | | | **System Response** |
| 1 | | Customer sends request to receive booking detail | | | System returns booking detail with following information:  - “Trạng thái”: text  - “Thời gian chụp”: text  - “Thời gian nhận ảnh”: text  - “Thời gian tác nghiệp dự kiến”: text  - “Địa điểm”:   * “Địa chỉ”: text * “Latitude”: double * “Longitude”: double   - “Thông tin photographer”:   * “Tên photographer”: text * “Điểm đánh giá”: number * “Số điện thoại photographer”: number * Avatar: image   - “Thông tin khách hàng”:   * “Tên khách hàng”: text * “Số điện thoại photographer”: number   “Avatar: image  - “Thông tin gói dịch vụ”:   * “Tên gói”: text * “Mô tả”: text * “Tổng cộng”: text * “Bao gồm các dịch vụ”: list of texts   - “Phương thức trả ảnh”: text  [Exception 1] |
| 2 | | Customer sends request to report photographer | | | System outputs:   * “Lý do”: radio button list * “Chi tiết”: text field |
| 3 | | Customer select an option, input reason and tap “Xác nhận”  [Alternative 1] | | | System outputs message: “Gửi báo cáo thành công”  [Exception 2] |
| Alternative Flows: | **Step** | **Actor Action** | | | | **System Response** |
| 1 | Customer confirm “Hủy” | | | | Back to booking detail view |
| Exceptions: | **No** | | **Cause** | | | **System Response** |
| 1 | | System can’t get the booking detail | | | System shows error message: “Không tìm thấy cuộc hẹn” |
| 2 | | The process run into an internal error | | | System shows error message: “Đã có lỗi xảy ra” |
| Priority: | Medium | | | | | |
| Frequency of Use: | Sometimes | | | | | |
| Business Rules: | * The report information contain: reporter information, booking’s information, report reason, reported time. * Send notification to customer / photographer after admin handle the report. * User can only report booking, not a user. | | | | | |
| Other Information: | N/A | | | | | |
| Assumptions: | N/A | | | | | |

Table 8 <Admin> Make report specification

##### <Customer> Submit cancellation

Figure 11 <Customer> Submit cancellation

*Figure SEQ Figure \\* ARABIC 13 <Customer> Cancel booking*



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| UC ID and Name: | **UC – 10 Submit cancellation** | | | | |
| Created By: | Trần Thiên Thảo | | Date Created: | 01/01/2021 | |
| Primary Actor: | Customer, Photographer | | Secondary Actors: |  | |
| Trigger: | Customer or Photographer sends request to cancel booking. | | | | |
| Description: | This use case allows customers or photographers to cancel their booking. | | | | |
| Preconditions: | * The User has been authenticated as Customer / Photographer. * Booking status is ONGOING / EDITING. | | | | |
| Post-conditions: | **Success:** The cancellation request is submitted to the system for admin to handle.  **Fail:** System shows error message | | | | |
| Normal Flow: | **Step** | **Actor Action** | | | **System Response** |
| 1 | Customer sends request to receive booking detail | | | System returns booking detail with following information:  - “Trạng thái”: text  - “Thời gian chụp”: text  - “Thời gian nhận ảnh”: text  - “Thời gian tác nghiệp dự kiến”: text  - “Địa điểm”:   * “Địa chỉ”: text * “Latitude”: double * “Longitude”: double   - “Thông tin photographer”:   * “Tên photographer”: text * “Điểm đánh giá”: number * “Số điện thoại photographer”: number * Avatar: image   - “Thông tin khách hàng”:   * “Tên khách hàng”: text * “Số điện thoại photographer”: number   “Avatar: image  - “Thông tin gói dịch vụ”:   * “Tên gói”: text * “Mô tả”: text * “Tổng cộng”: text * “Bao gồm các dịch vụ”: list of texts   - “Phương thức trả ảnh”: text  [Exception 1] |
| 2 | Customer sends request to cancel booking | | | System outputs: “Bạn muốn hủy cuộc hẹn này?” with textfield to input cancelling reason. |
| 3 | Customer confirm “Có” and inputs cancelling reasons to cancel booking  [Alternative 1] | | | System outputs message: “Đã gửi yêu cầu hủy cuộc hẹn”  [Exception 2] |
| Alternative Flows: | **Step** | **Actor Action** | | | **System Response** |
| 1 | Customer confirm “Không” | | | Back to booking detail view |
| Exceptions: | **No** | **Cause** | | | **System Response** |
| 1 | System can’t get the booking detail | | | System shows error message: “Không tìm thấy cuộc hẹn” |
| 2 | The process run into an internal error | | | System shows error message: “Đã có lỗi xảy ra” |
| Priority: | Medium | | | | |
| Frequency of Use: | Sometimes | | | | |
| Business Rules: | * Customers or Photographers can cancel if status is ONGOING, EDITING. * When cancels, customer send the cancellation request to admin, update status to CANCELLING\_CUSTOMER / CANCELLING\_PHOTOGRAPHER and notify Photographer / Customer. * The cancellation request information includes: cancelling reason, cancelled user information, booking information, created date. * - Customers can directly cancel pending bookings without approval from admin. | | | | |
| Other Information: | N/A | | | | |
| Assumptions: | N/A | | | | |

Table 9 <Customer> Submit cancellation specification

#### 2.2.3 <Photographer> Overview Use Case

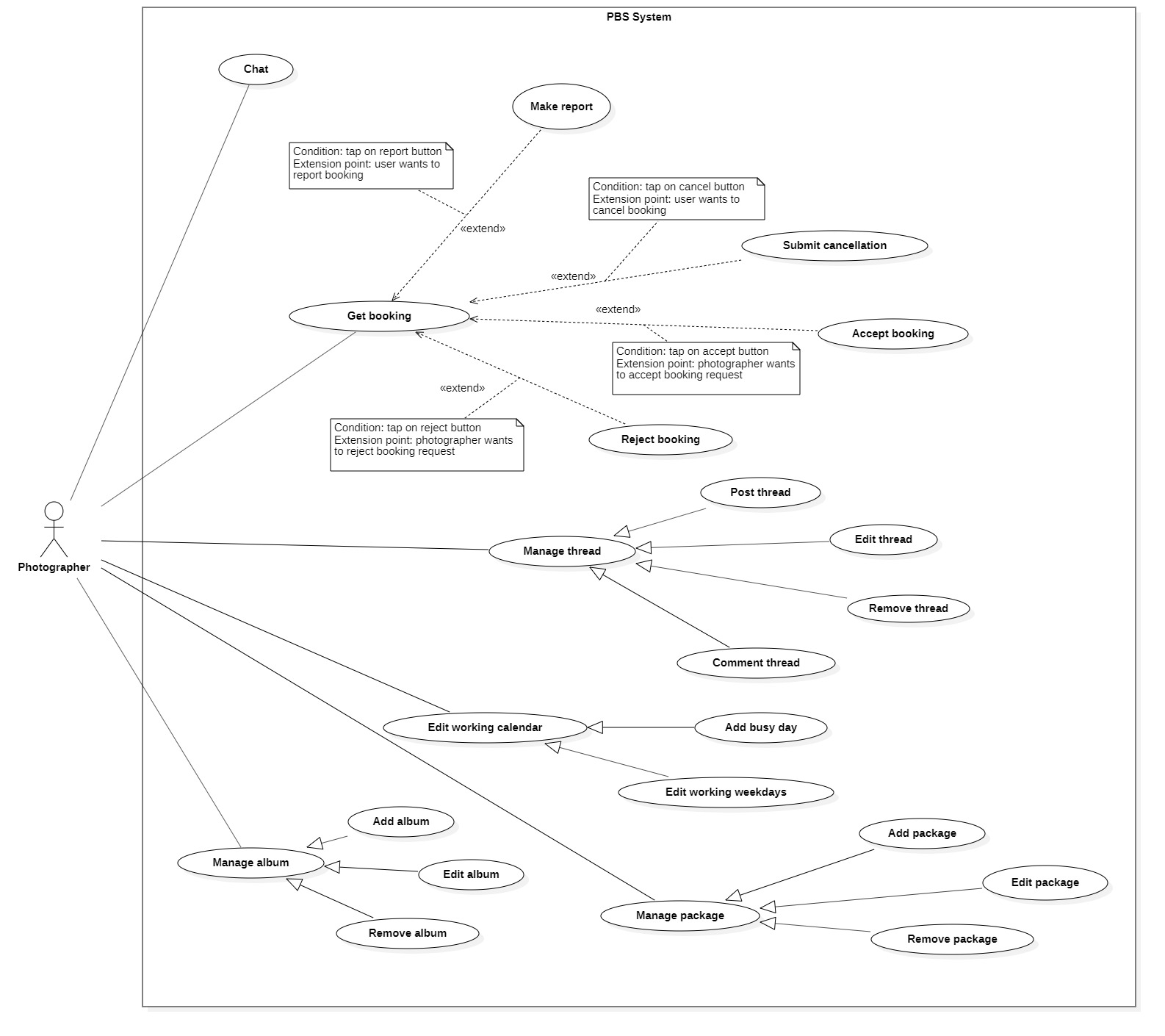


Figure 12 <Photographer> Overview

##### <Photographer> Accept booking

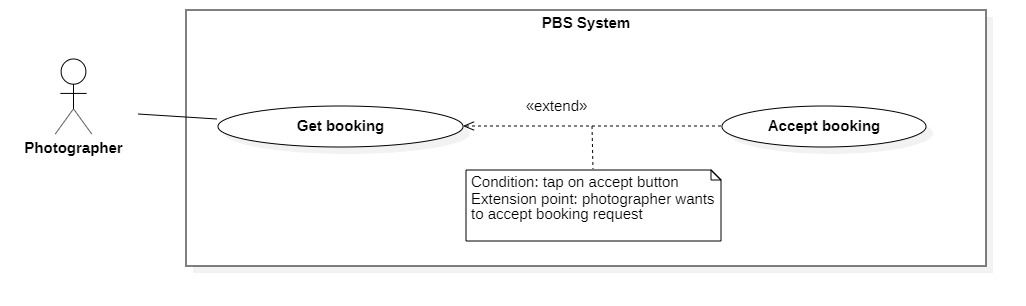


Figure 13 <Photographer> Accept booking

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| UC ID and Name: | **UC – 12 Accept booking** | | | | |
| Created By: | Trần Thiên Thảo | | Date Created: | 08/10/2020 | |
| Primary Actor: | Photographer | | Secondary Actors: |  | |
| Trigger: | The photographer sends a request to accept booking. | | | | |
| Description: | This use case allows photographers to accept booking from customer. | | | | |
| Preconditions: | - User has been authenticated as a Photographer.  - The booking status is PENDING. | | | | |
| Post-conditions: | **Success:** The booking will be updated in the system with status change to ONGOING.  **Fail:** System shows error message | | | | |
| Normal Flow: | **Step** | **Actor Action** | | | **System Response** |
| 1 | Photographer sends request to receive booking details. | | | System returns booking detail with following information:  - “Trạng thái”: text  - “Thời gian chụp”: text  - “Thời gian nhận ảnh”: text  - “Thời gian tác nghiệp dự kiến”: text  - “Địa điểm”:   * “Địa chỉ”: text * “Latitude”: double * “Longitude”: double   - “Thông tin photographer”:   * “Tên photographer”: text * “Điểm đánh giá”: number * “Số điện thoại photographer”: number * Avatar: image   - “Thông tin khách hàng”:   * “Tên khách hàng”: text * “Số điện thoại photographer”: number   “Avatar: image  - “Thông tin gói dịch vụ”:   * “Tên gói”: text * “Mô tả”: text * “Tổng cộng”: text * “Bao gồm các dịch vụ”: list of texts   - “Phương thức trả ảnh”: text  [Exception 1] |
| 2 | Photographer tap “Đồng ý”  [Alternative 1] | | | System changes booking status to ONGOING.  [Exception 2] |
|  |  | | |  |
| Alternative Flows: | **Step** | **Actor Action** | | | **System Response** |
| 1 | Photographer confirm “Không” | | | Back to booking detail view |
| Exceptions: | **No** | **Cause** | | | **System Response** |
| 1 | System can’t get the booking detail | | | System shows error message: “Không tìm thấy cuộc hẹn” |
| 2 | The process run into an internal error | | | System shows error message: “Đã có lỗi xảy ra” |
| Priority: | Medium | | | | |
| Frequency of Use: | Usually | | | | |
| Business Rules: | * After accepting, change status of booking to ONGOING. * After photographer accepts, notification will be sent to customer. | | | | |
| Other Information: | N/A | | | | |
| Assumptions: | N/A | | | | |

Table 10 <Photographer> Accept booking specification

##### <Photographer> Reject booking

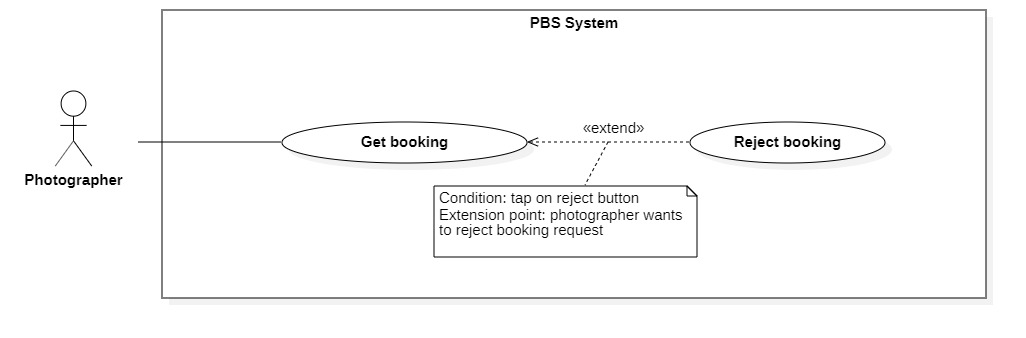


Figure 14 <Photographer> Reject booking

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| UC ID and Name: | **UC – 13 Reject booking** | | | | |
| Created By: | Trần Thiên Thảo | | Date Created: | 08/10/2020 | |
| Primary Actor: | Photographer | | Secondary Actors: |  | |
| Trigger: | The photographer sends a request to reject booking. | | | | |
| Description: | This use case allows photographer to reject booking from customer | | | | |
| Preconditions: | - The User has been authenticated as a Photographer.  - The booking status is PENDING. | | | | |
| Post-conditions: | **Success:** The booking will be updated in the system and status change to REJECTED  **Fail:** System shows error message | | | | |
| Normal Flow: | **Step** | **Actor Action** | | | **System Response** |
| 1 | Photographer sends request to receive booking details. | | | System returns booking detail with following information:  - “Trạng thái”: text  - “Thời gian chụp”: text  - “Thời gian nhận ảnh”: text  - “Thời gian tác nghiệp dự kiến”: text  - “Địa điểm”:   * “Địa chỉ”: text * “Latitude”: double * “Longitude”: double   - “Thông tin photographer”:   * “Tên photographer”: text * “Điểm đánh giá”: number * “Số điện thoại photographer”: number * Avatar: image   - “Thông tin khách hàng”:   * “Tên khách hàng”: text * “Số điện thoại photographer”: number   “Avatar: image  - “Thông tin gói dịch vụ”:   * “Tên gói”: text * “Mô tả”: text * “Tổng cộng”: text * “Bao gồm các dịch vụ”: list of texts   - “Phương thức trả ảnh”: text  [Exception 1] |
| 2 | Photographer sends request to reject booking | | | System outputs: “Từ chối cuộc hẹn này?” |
| 3 | Photographer confirms “Có”  [Alternative 1] | | | System outputs: “Đã từ chối cuộc hẹn”  [Exception 2] |
| Alternative Flows: | **Step** | **Actor Action** | | | **System Response** |
| 1 | Photographer confirm “Không” | | | Back to booking detail view |
| Exceptions: | **No** | **Cause** | | | **System Response** |
| 1 | System can’t get the booking detail | | | System shows error message: “Không tìm thấy cuộc hẹn” |
| 2 | The process run into an internal error | | | System shows error message: “Đã có lỗi xảy ra” |
| Priority: | Usually | | | | |
| Frequency of Use: | Sometimes | | | | |
| Business Rules: | * After rejected, change booking status to REJECTED. * When photographer sends request to reject booking, get the rejection reason and save to system. * After photographer reject a booking, notification will be sent to customer. | | | | |
| Other Information: | N/A | | | | |
| Assumptions: | N/A | | | | |

Table 11 <Photographer> Reject booking specification

#### 2.2.5 <Admin> Overview Use Case



Figure 15 <Admin> Overview

##### <Admin> Get cancellation

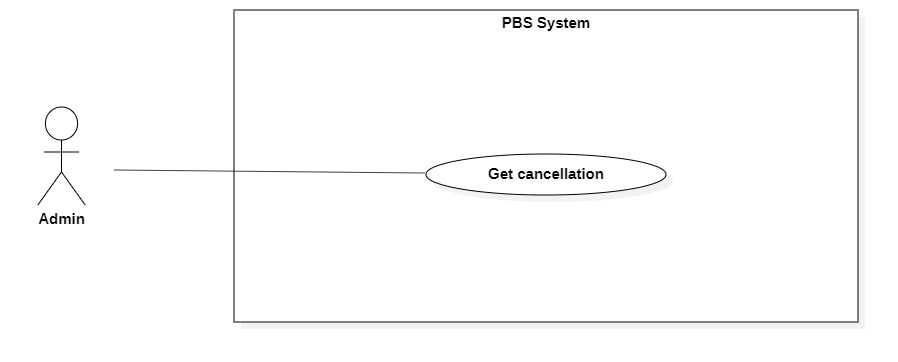


Figure 16 <Admin> Get cancellation

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| UC ID and Name: | **UC – 40 Get cancellation** | | | | |
| Created By: | Trần Thiên Thảo | | Date Created: | 01/01/2021 | |
| Primary Actor: | Admin | | Secondary Actors: |  | |
| Trigger: | Admin sends a request to get cancellation. | | | | |
| Description: | This use case allows admin to get cancellation. | | | | |
| Preconditions: | User has been authenticated as Admin. | | | | |
| Post-conditions: | **Success:** System displays detail of a cancellation.  **Fail:** System shows error message. | | | | |
| Normal Flow: | Step | Actor Action | | | System Response |
| 1 | Admin sends a request to get detail of a cancellation. | | | System returns cancellation detail with following information:  - “Người hủy”: text  - “Lý do hủy”: text  - “Thời gian hủy”: datetime  - “Lý do hủy”: text  - Booking information:   * “Khách hàng”: text * “Thợ ảnh”: text * “Trạng thái”: text   [Exception 1] |
| Alternative Flows: | N/A | | | | |
| Exceptions: | No | Cause | | | System Response |
| 1 | System run into internal error | | | System shows error message: “Đã có lỗi xảy ra” |
| Priority: | Medium | | | | |
| Frequency of Use: | Usually | | | | |
| Business Rules: | * In cancellation view, admin can approve cancellation, warn user or navigate to user detail and block them there. * Admin can navigate to booking detail view to check information. | | | | |
| Other Information: | N/A | | | | |
| Assumptions: | N/A | | | | |

Table 12 <Admin> Get cancellation specification

##### <Admin> Approve cancellation

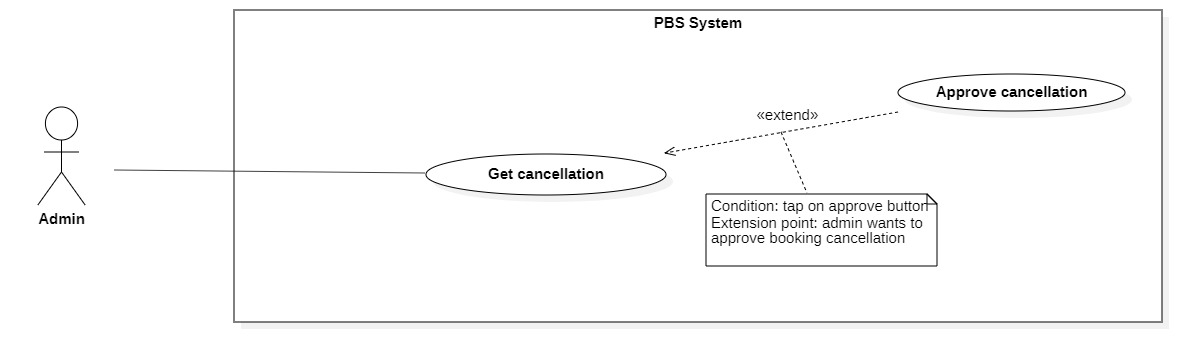


Figure 17 <Admin> Approve cancellation

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| UC ID and Name: | **UC – 41 Approve cancellation** | | | |
| Created By: | Đào Sỹ Trung Kiên | Date Created: | | 01/01/2021 |
| Primary Actor: | Admin | Secondary Actors: | |  |
| Trigger: | The Admin sends request to approve cancellation | | | |
| Description: | This use case allows Admin to approve cancellation | | | |
| Preconditions: | The User has been authenticated as Admin | | | |
| Post-conditions: | **Success:** User's cancellation request is approved.  **Fail:** Systems outputs error message and sends to Admin | | | |
| Normal Flow: | **Step** | **Actor Action** | **System Response** | |
| 1 | Admin send requests to receive cancellation view. | System returns cancellation view with following information:  - “Người hủy”: text  - “Lý do hủy”: text  - “Khách hàng”: text  - “Thợ ảnh”: text  - “Thời gian hẹn”: text  - “Thời gian hủy”: text  - “Trạng thái đơn hẹn”: text  [Exception 1] | |
| 2 | Admin send request to approve cancellation. | System outputs “Bạn muốn hủy đơn này ?”. | |
|  | 3 | Admin confirms “Có”.  [Alternative 1] | System update data and outputs “Đã hủy đơn”.  [Exception 2] | |
| Alternative Flows: | **Step** | **Actor Action** | **System Response** | |
| 1 | Admin confirms “Không”. | Back to cancellation view. | |
| Exceptions: | **No** | **Actor Action** | **System Response** | |
| 1 | System can’t get cancellation view. | System shows error message: “Không tìm thấy yêu cầu hủy” | |
|  | 2 | The process run into an internal error | System shows error message: “Đã có lỗi xảy ra” | |
| Priority: | Medium | | | |
| Frequency of Use: | Sometimes | | | |
| Business Rules: | * After approval, change status of booking to CANCELLED\_PHOTOGRAPHER / CANCELLED\_CUSTOMER. * After admin approval, notification will be sent to customer and photographer. | | | |
| Other Information: | N/A | | | |
| Assumptions: | N/A | | | |

Table 13 <Admin> Approve cancellation

##### <Admin> Get report

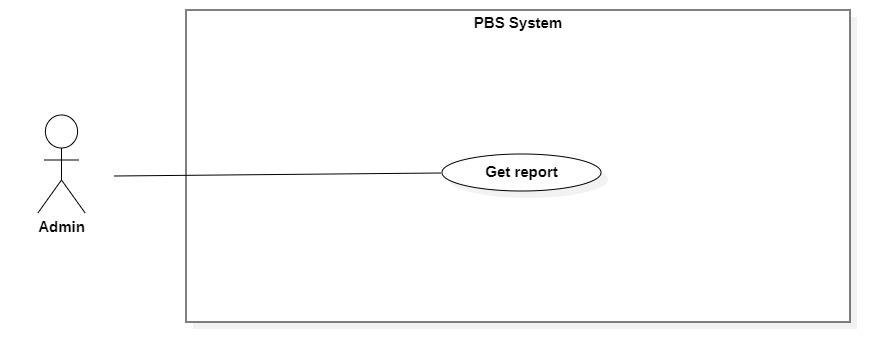


Figure 18 <Admin> Get report

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| UC ID and Name: | **UC – 39 Get report** | | | | |
| Created By: | Đào Sỹ Trung Kiên | | Date Created: | 08/10/2020 | |
| Primary Actor: | Admin | | Secondary Actors: |  | |
| Trigger: | The admin sends a request to get user’s report. | | | | |
| Description: | This use case allows admins to get get user’s report. | | | | |
| Preconditions: | The User has been authenticated as admin | | | | |
| Post-conditions: | **Success:** The system returns get user’s report  **Fail:** System shows error message | | | | |
| Normal Flow: | **Step** | **Actor Action** | | | **System Response** |
| 1 | Admin send requests to get user’s report. | | | System displays following information:   * “Người báo cáo”: text. * “Lý do”: text. * “Khách hàng”: text. * “Thợ ảnh”: text. * “Thời gian”: text.   [Exception 1] |
| Alternative Flows: | N/A | | | | |
| Exceptions: | **No** | **Cause** | | | **System Response** |
| 1 | The process runs into an internal error. | | | System outputs error message: “Đã có lỗi xảy ra” |
| Priority: | Medium | | | | |
| Frequency of Use: | Sometimes | | | | |
| Business Rules: | * Admin can check seen the report. * Admin can warn user via notification and email. * Admin can navigate to user’s detail and block them there. | | | | |
| Other Information: | N/A | | | | |
| Assumptions: | N/A | | | | |

Table 14 <Admin> Get report

##### <Admin> Block user

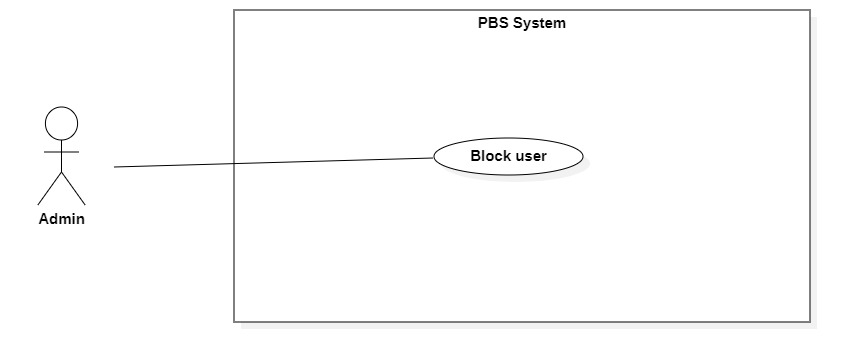
******

Figure 19 <Admin> Block user

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| UC ID and Name: | **UC – 34 Block user** | | | | |
| Created By: | Đào Sỹ Trung Kiên | | Date Created: | 11/11/2020 | |
| Primary Actor: | Admin | | Secondary Actors: |  | |
| Trigger: | The Admin sends request to block user. | | | | |
| Description: | This use case allows Admin to block user. | | | | |
| Preconditions: | The User has been authenticated as Admin. | | | | |
| Post-conditions: | **Success:** User will be blocked and unable to access the system.  **Fail:** Systems outputs error message and sends to Admin. | | | | |
| Normal Flow: | **Step** | **Actor Action** | | | **System Response** |
| 1 | The Admin goes to user view. | | | System returns user information include:  - Avatar: image  - Role: text  - All booking number: number  - Number of cancelled booking: number  - Cancellation rate: number  - Number of done booking: number  - Emai: text  - Phone: text  - Account status: text |
| 2 | Admin send request to block user. | | | System shows confirmation message: “Bạn có muốn chặn người dung này?” |
| 3 | Admin confim “Có”  [Alternative 1] | | | System block user and outputs message “Chặn người dùng thành công” and sends to the admin.  [Exception 1] |
| Alternative Flows: | **Step** | **Actor Action** | | | **System Response** |
| 1 | Admin confim “Không”. | | | Back to user view. |
| Exceptions: | **No** | **Actor Action** | | | **System Response** |
| 1 | The create process encountered an internal error. | | | System outputs error message and sends admin to notify that block user failed. |
| Priority: | Medium | | | | |
| Frequency of Use: | Sometimes | | | | |
| Business Rules: | * User’s status will be changed to blocked. * System will log the user out of the system. * If the user login after being blocked, system shows “Tài khoản của bạn đã bị vô hiệu”. * If user has booking, the system will notice admin that this user has bookings and show confirmation alert. If the admin confirms yes the system will block the user and set all PENDING / EDITING bookings’ status to CANCELLED\_ADMIN then notify the opposite Customer / Photographer. * Mail to blocked user. | | | | |
| Other Information: | N/A | | | | |
| Assumptions: | N/A | | | | |

Table 15 <Admin> Block user specification

##### <Admin> Unblock user

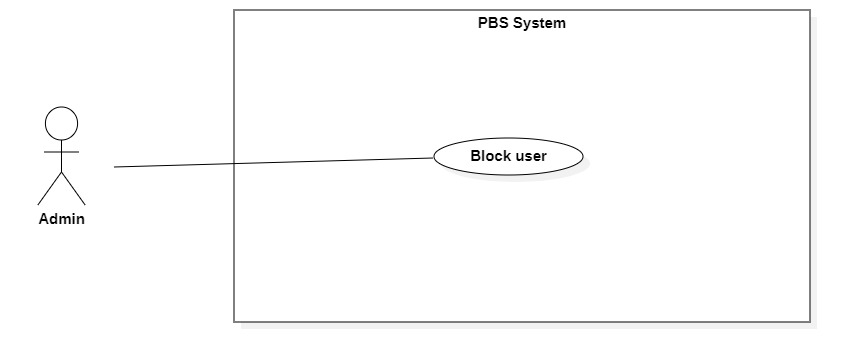
****

Figure 20 <Admin> Unblock user

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| UC ID and Name: | **UC – 35 Unblock user** | | | | |
| Created By: | Đào Sỹ Trung Kiên | | Date Created: | 11/11/2020 | |
| Primary Actor: | Admin | | Secondary Actors: |  | |
| Trigger: | The Admin sends request to unblock user. | | | | |
| Description: | This use case allows Admin to unblock user. | | | | |
| Preconditions: | The User has been authenticated as Admin. | | | | |
| Post-conditions: | **Success:** User will be unblocked and enabled to access the system.  **Fail:** Systems outputs error message and sends to Admin. | | | | |
| Normal Flow: | **Step** | **Actor Action** | | | **System Response** |
| 1 | The Admin goes to user view. | | | System returns user information include:  - Avatar: image  - Role: text  - All booking number: number  - Number of cancelled booking: number  - Cancellation rate: number  - Number of done booking: number  - Emai: text  - Phone: text  - Account status: text |
| 2 | Admin send request to unblock user. | | | System shows confirmation message: “Bạn có muốn gỡ chặn người dung này?” |
|  | 3 | Admin confim “Có”  [Alternative 1] | | | System unblock user and outputs message “Gỡ chặn người dùng thành công” and sends to the admin.  [Exception 1] |
| Alternative Flows: | **Step** | **Actor Action** | | | **System Response** |
| 1 | Admin confim “Không”. | | | Back to user view. |
| Exceptions: | **No** | **Actor Action** | | | **System Response** |
| 1 | The create process encountered an internal error. | | | System outputs error message and sends admin to notify that unblock user failed. |
| Priority: | Medium | | | | |
| Frequency of Use: | Sometimes | | | | |
| Business Rules: | * User’s status will be changed to unblocked. * Mail to inform user. * User’s booking, profile, password, information will be remained same. | | | | |
| Other Information: | N/A | | | | |
| Assumptions: | N/A | | | | |

Table 16 <Admin> Unblock user specification

##### <Admin> Warn User

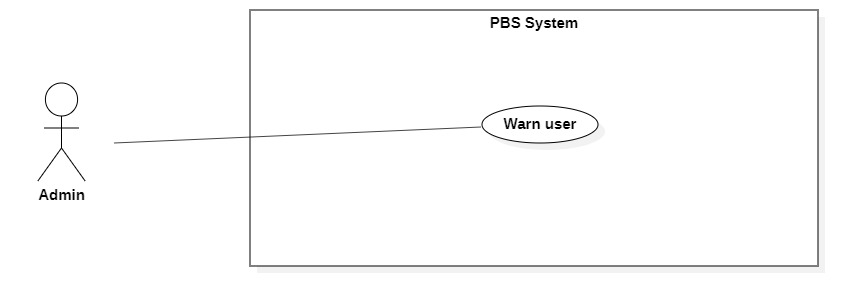


Figure 21 <Admin> Warn user

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| UC ID and Name: | **UC – 36 Warn user** | | | | |
| Created By: | Đào Sỹ Trung Kiên | | Date Created: | 01/01/2021 | |
| Primary Actor: | Admin | | Secondary Actors: |  | |
| Trigger: | The Admin sends request to warn user. | | | | |
| Description: | This use case allows Admin to warn user. | | | | |
| Preconditions: | The User has been authenticated as Admin. | | | | |
| Post-conditions: | **Success:** A warning will be sent to user.  **Fail:** Systems outputs error message and sends to admin. | | | | |
| Normal Flow: | **Step** | **Actor Action** | | | **System Response** |
| 1 | The Admin goes to warn user view | | | System requires information from admin:  - “Cảnh báo”: text, required. |
| 2 | Admin inputs required information and send request to warn user | | | System validates information, warn user and outputs message “Đã cảnh báo người dùng” and sends to the admin.  [Exception 1]  [Exception 2] |
| Alternative Flows: | N/A | | | | |
| Exceptions: | **No** | **Actor Action** | | | **System Response** |
| 1 | Admin inputs invalid format text inputs or leaves empty required fields. | | | System outputs error message and sends it to ask admin to input correct required fields. |
| 2 | The create process encountered an internal error. | | | System outputs error message and sends admin to notify that warn user failed. |
| Priority: | Medium | | | | |
| Frequency of Use: | Sometimes | | | | |
| Business Rules: | * This use case related to cancellation management and report management. * After warn user, set warned user’s booking to CANCELLED\_PHOTOGRAPHER or CANCELLED\_CUSTOMER depends on the role of warned user. * Send notify to the opposite user about the cancelled booking. * Send warning notification to warned user. Warning notification includes admin’s warning message, cancelled booking information. | | | | |
| Other Information: | N/A | | | | |
| Assumptions: | N/A | | | | |

Table 17 <Admin> Warn user specification

#### 2.2.6 <<system>>Handler Overview use case

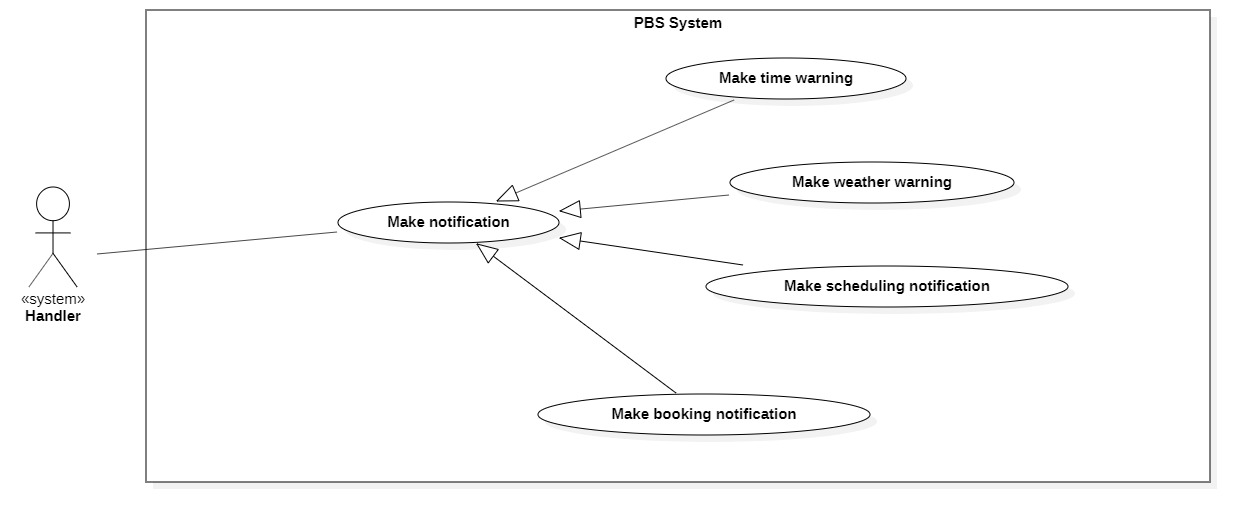


Figure 22 <<system>> Handler Overview

##### <<system>>Handler Make scheduling notification

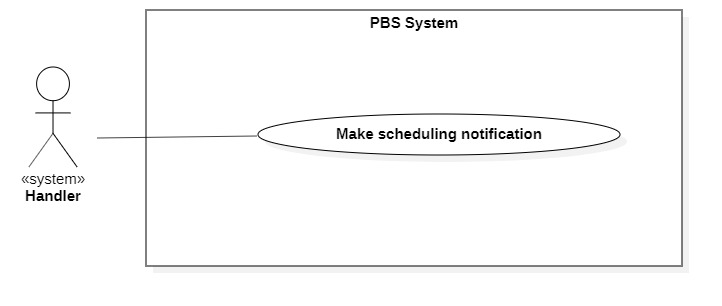


Figure 23 <<system>> Handler make scheduling notification

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| UC ID and Name: | **UC – 49 Make scheduling notification** | | | | |
| Created By: | Bồ Công Đạt | | Date Created: | 01/01/2021 | |
| Primary Actor: | System Handler | | Secondary Actors: |  | |
| Trigger: | System handler sends a request to notify photographer or customer on scheduling. | | | | |
| Description: | This use case allows system handler to make scheduling notification | | | | |
| Preconditions: | N/A | | | | |
| Post-conditions: | **Success:** System make scheduling notification and sends to photographer or customer  **Fail:** System outputs error and sends to photographer or customer | | | | |
| Normal Flow: | **Step** | **Actor Action** | | | **System Response** |
| 1 | System handler retrieves scheduling changing request | | | System stores scheduling information and return notification to system handler |
| 2 | System handler retrieves notification information and sends to photographer or customer  [Exception 1] | | |  |
| Alternative Flows: |  | | | | |
| Exceptions: | **No** | **Cause** | | | **System Response** |
| 1 | The process runs into an internal error | | | System outputs error and sends to photographer or customer |
| Priority: | Medium | | | | |
| Frequency of Use: | Usually | | | | |
| Business Rules: | This use case will be triggered in following situations:   * For ONGOING or EDITING booking, scheduling notification will be sent in following situation: * If booking has start time or editing deadline in between 00:00 – 12:00, system will notice user at 21:00 of the day before. * If booking has start time or editing deadline in between 12:01 – 24:00, system will notice user at 07:00 of that day. | | | | |
| Other Information: | N/A | | | | |
| Assumptions: | N/A | | | | |

Table 18 <System Handler> Make scheduling notification specification

##### <<system>>Handler Make weather warning

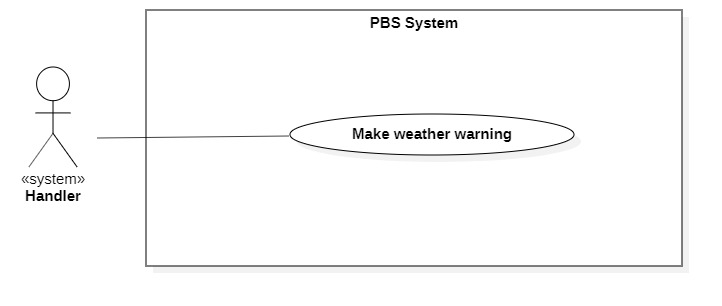


Figure 24 <<system>> Handler make weather warning

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| UC ID and Name: | **UC – 50 Make weather warning** | | | | |
| Created By: | Bồ Công Đạt | | Date Created: | 01/01/2021 | |
| Primary Actor: | System Handler | | Secondary Actors: |  | |
| Trigger: | System handler sends requests to notify customers of bad weather. | | | | |
| Description: | This use case allows the system handler to make weather warnings. | | | | |
| Preconditions: | N/A | | | | |
| Post-conditions: | **Success:** System makes weather warning and sends to customer.  **Fail:** System outputs error and sends to photographer or customer | | | | |
| Normal Flow: | **Step** | **Actor Action** | | | **System Response** |
| 1 | System handler retrieves bad weather request | | | System stores bad weather information and return notification to system handler |
| 2 | System handler retrieves notification information and sends to photographer or customer  [Exception 1] | | |  |
| Alternative Flows: | N/A | | | | |
| Exceptions: | **No** | **Cause** | | | **System Response** |
| 1 | The process runs into an internal error | | | System outputs error and sends to photographer or customer |
| Priority: | Medium | | | | |
| Frequency of Use: | Usually | | | | |
| Business Rules: | This use case will be triggered in following situations:   * Customer make a new booking, system checks whether if that day’s weather is suitable for shooting and send the warning notification back to her/his. * The return information will include: datetime, location, weather information. | | | | |
| Other Information: | N/A | | | | |
| Assumptions: | N/A | | | | |

Table 19 <System Handler> Make weather warning specification

## 3. Functional Requirements

### 3.1 System Functional Overview

#### a. Entity Relationship Diagram

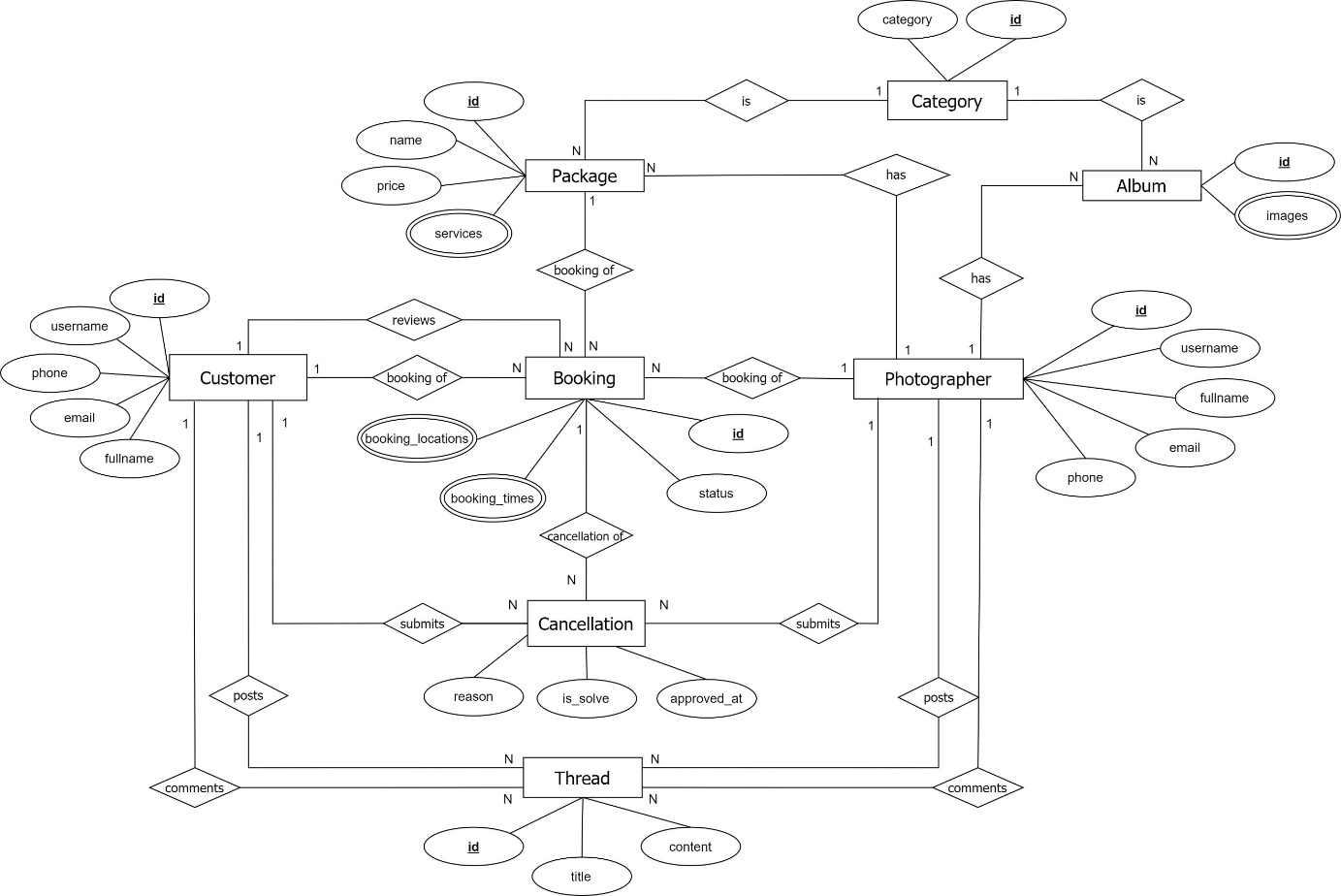


Figure 25 ER Diagram

## 4. Non-Functional Requirements

## 5. Other Requirements

# IV. Software Design Document

## 1. Overall Description

## 2. System Architecture Design

### 2.1 Overall Architecture

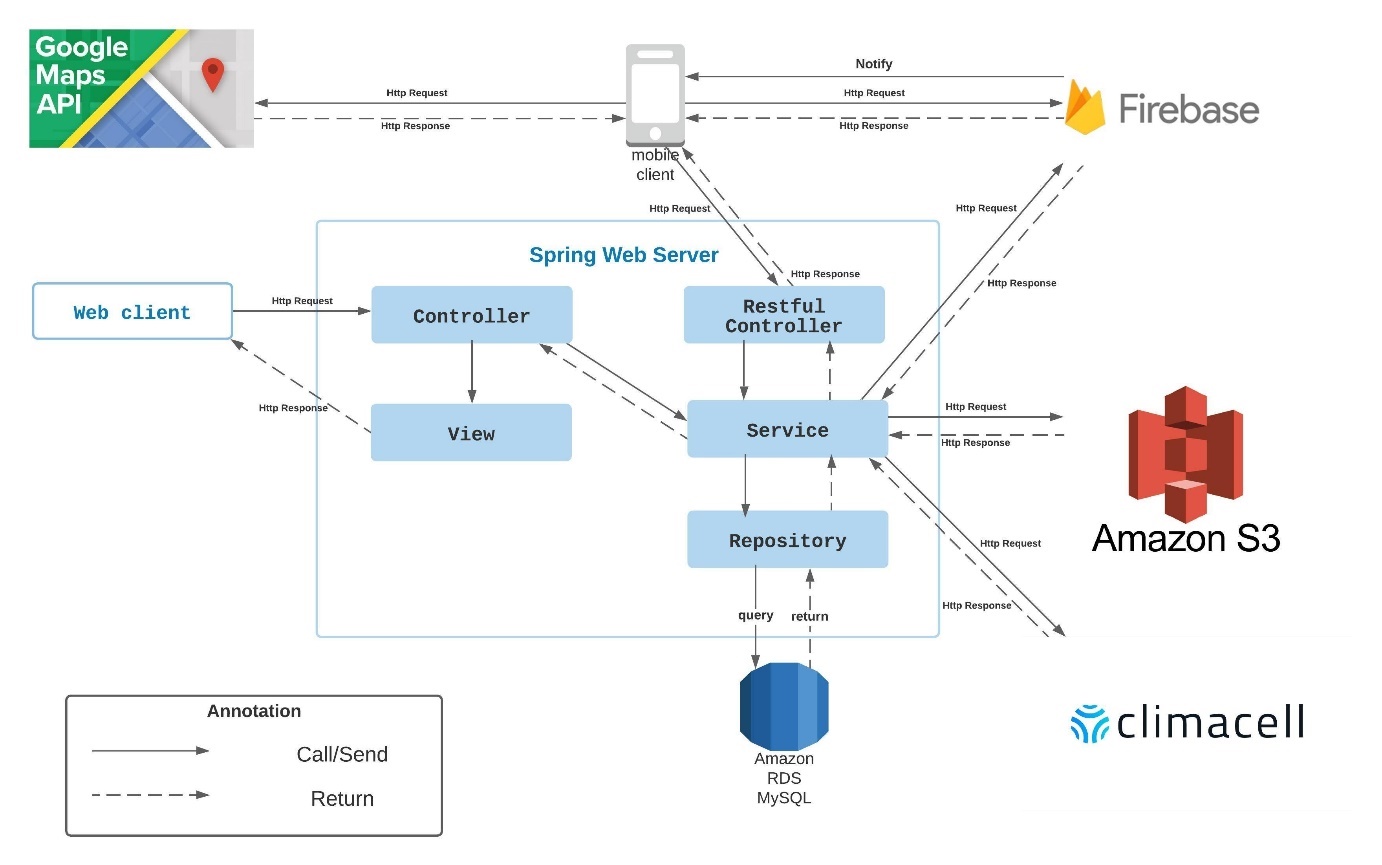


Figure 26 Overall Architecture

PBS system consists of three components, the spring boot web server provides APIs, two mobile applications for customers and photographers, and the web application for admin roles.

We also use several third-party services such as Aws S3 for image storage, Firebase cloud messaging for push notification functions, Cloud Firestore for chat functions, Firebase Realtime database and Amazon RDS for MySQL instances.

### 2.2 System Architecture

### 2.3 Package Diagram

## 3. System Detailed Design

### 3.1 Overall

#### a. Class Diagram

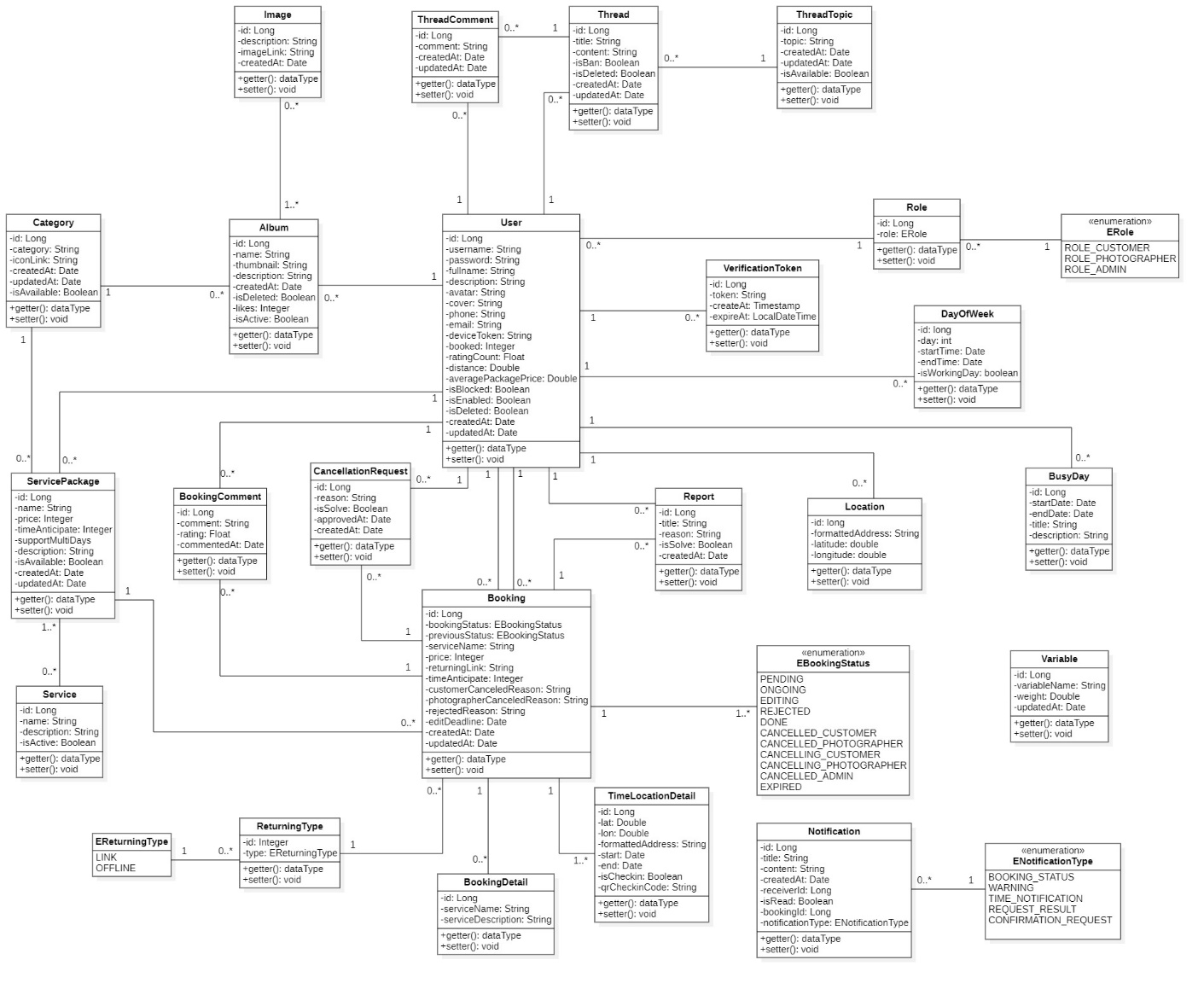


Figure 27 Class Diagram

#### b. Sequence Diagram(s)

1. Block user

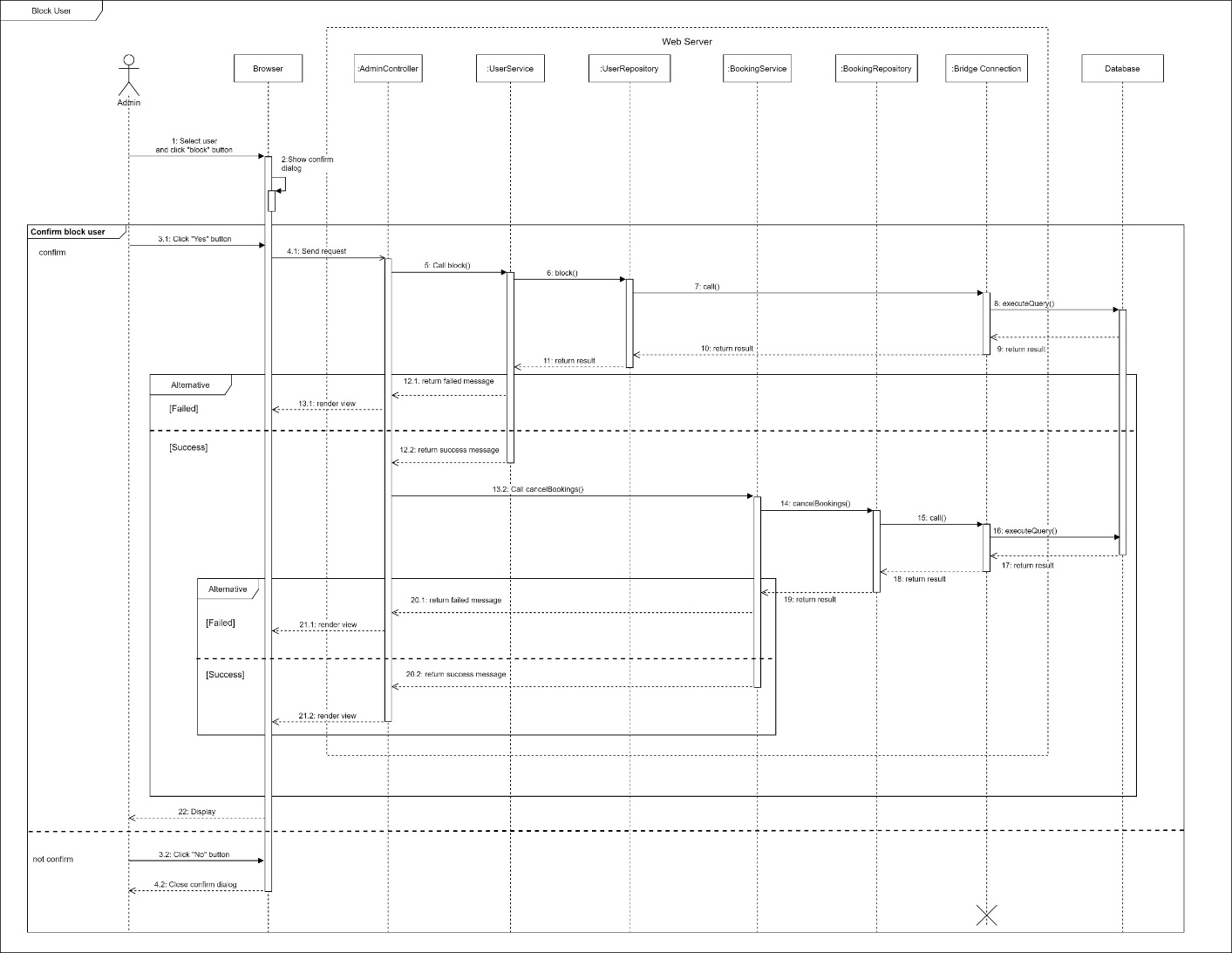


Figure 28 Sequence Diagram - Block user

1. Unblock user

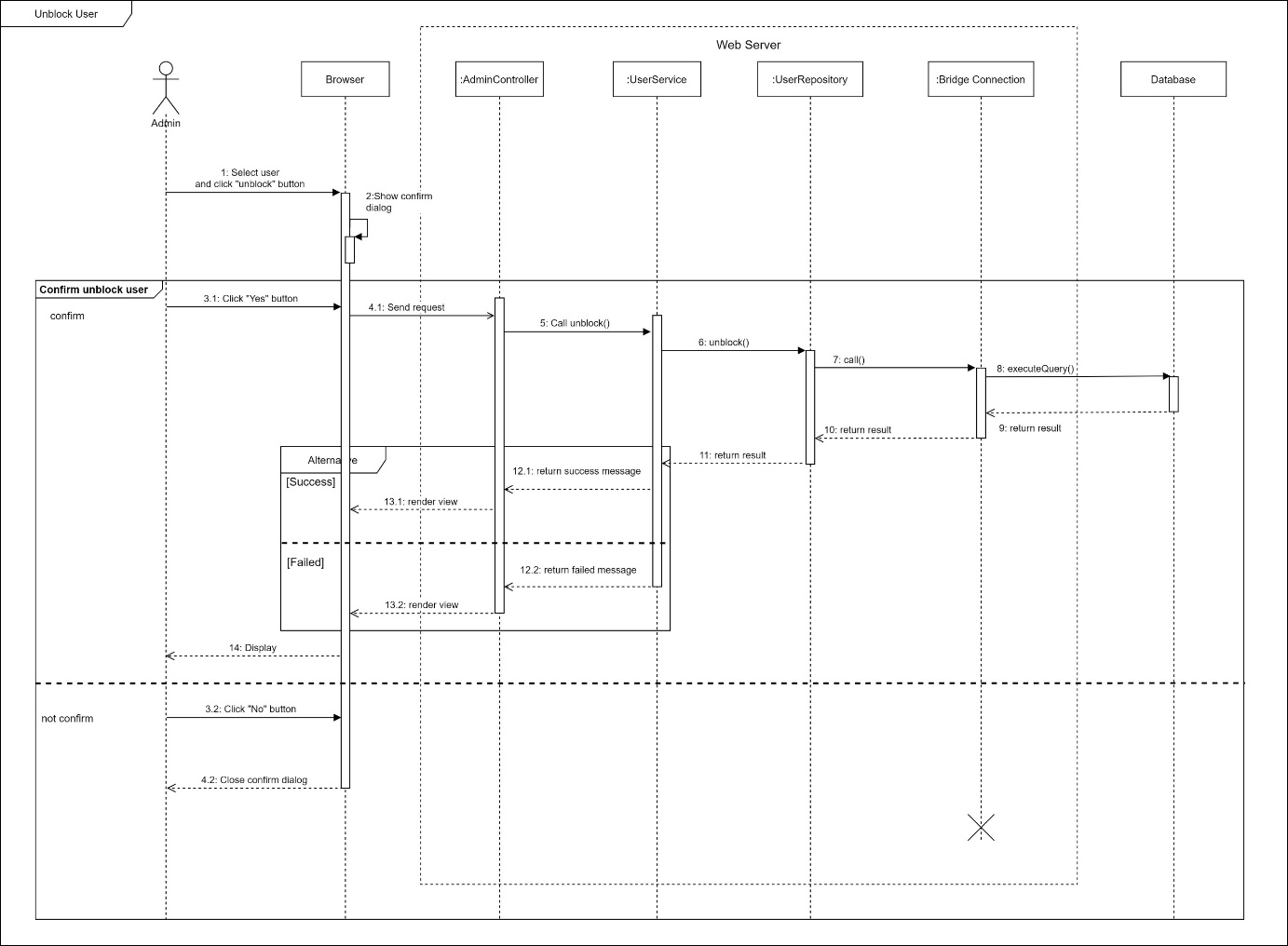


Figure 29 Sequence Diagram - Unblock User

1. Warn user

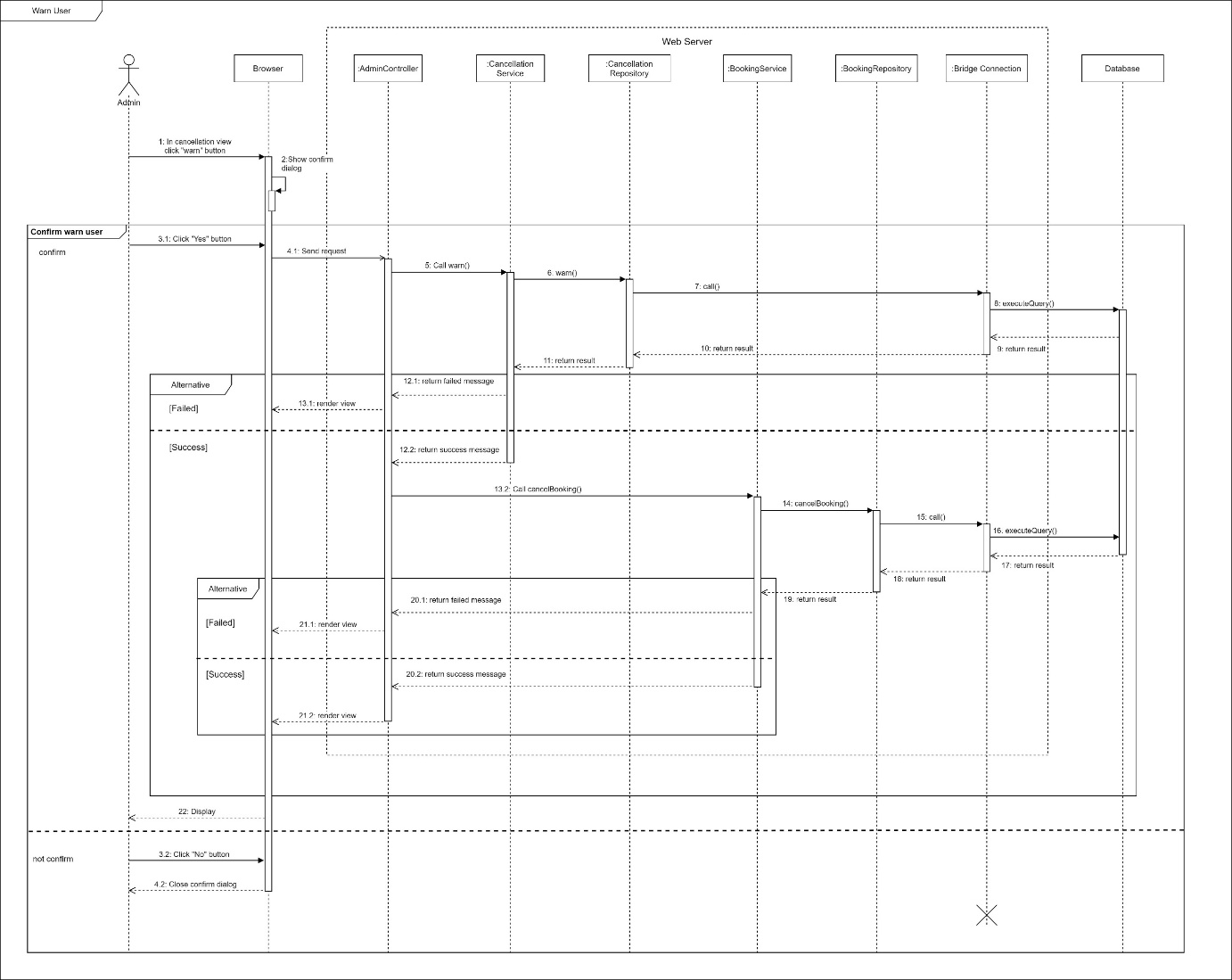


Figure 30 Sequence Diagram - warn user

1. Approve cancellation

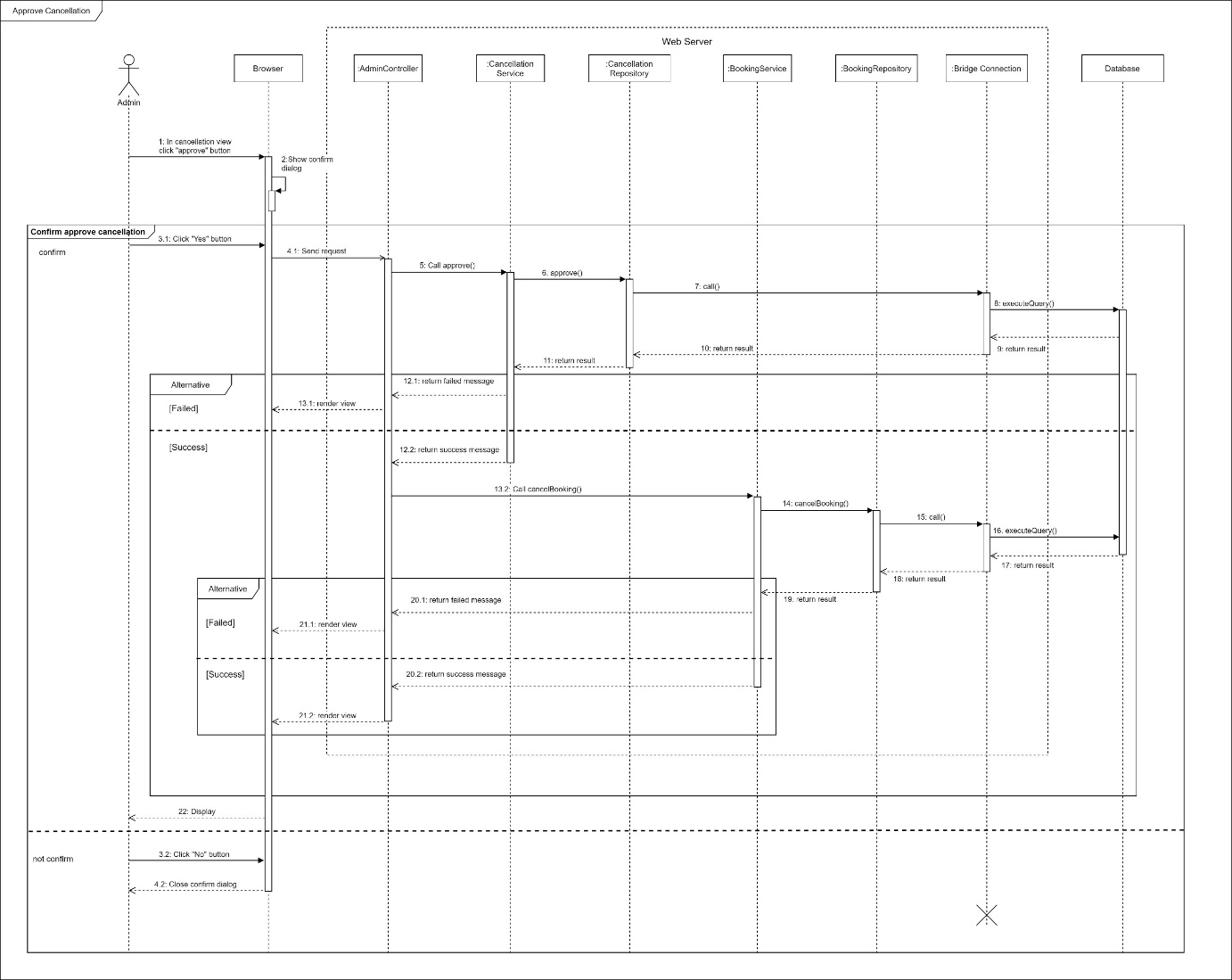


Figure 31 Sequence Diagram - approve cancellation

#### c. Activity Diagram(s)

1. **<Customer>** **Book photographer**

Summary: This diagram shows the process when customer books a photographer.

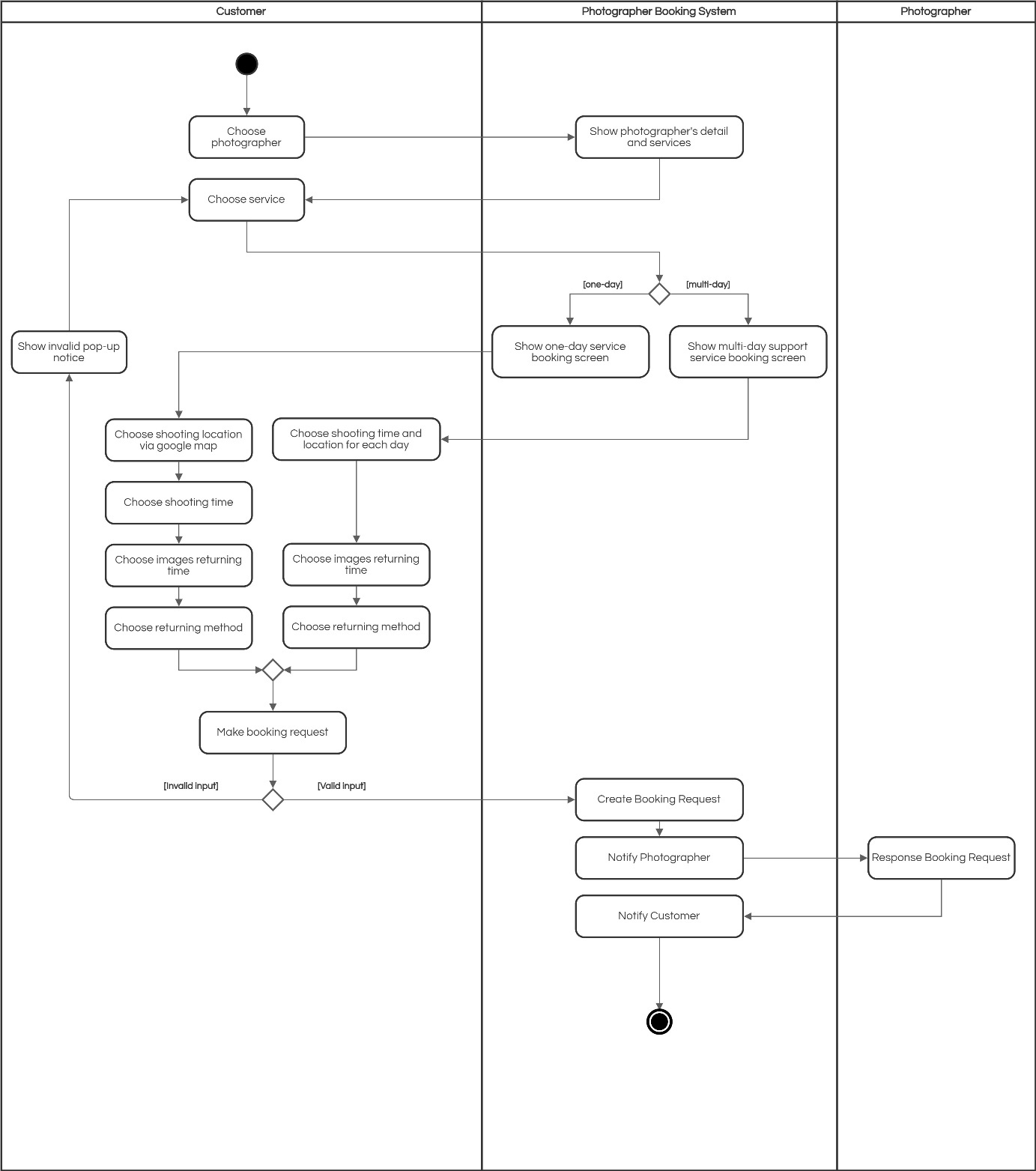


Figure 32 Activity Diagram – Book photographer

*Figure SEQ Figure \\* ARABIC 12 Activity Diagram - Make Request*

1. **<Customer> Edit booking**

Summary: This diagram shows the process when a customer edits a booking.

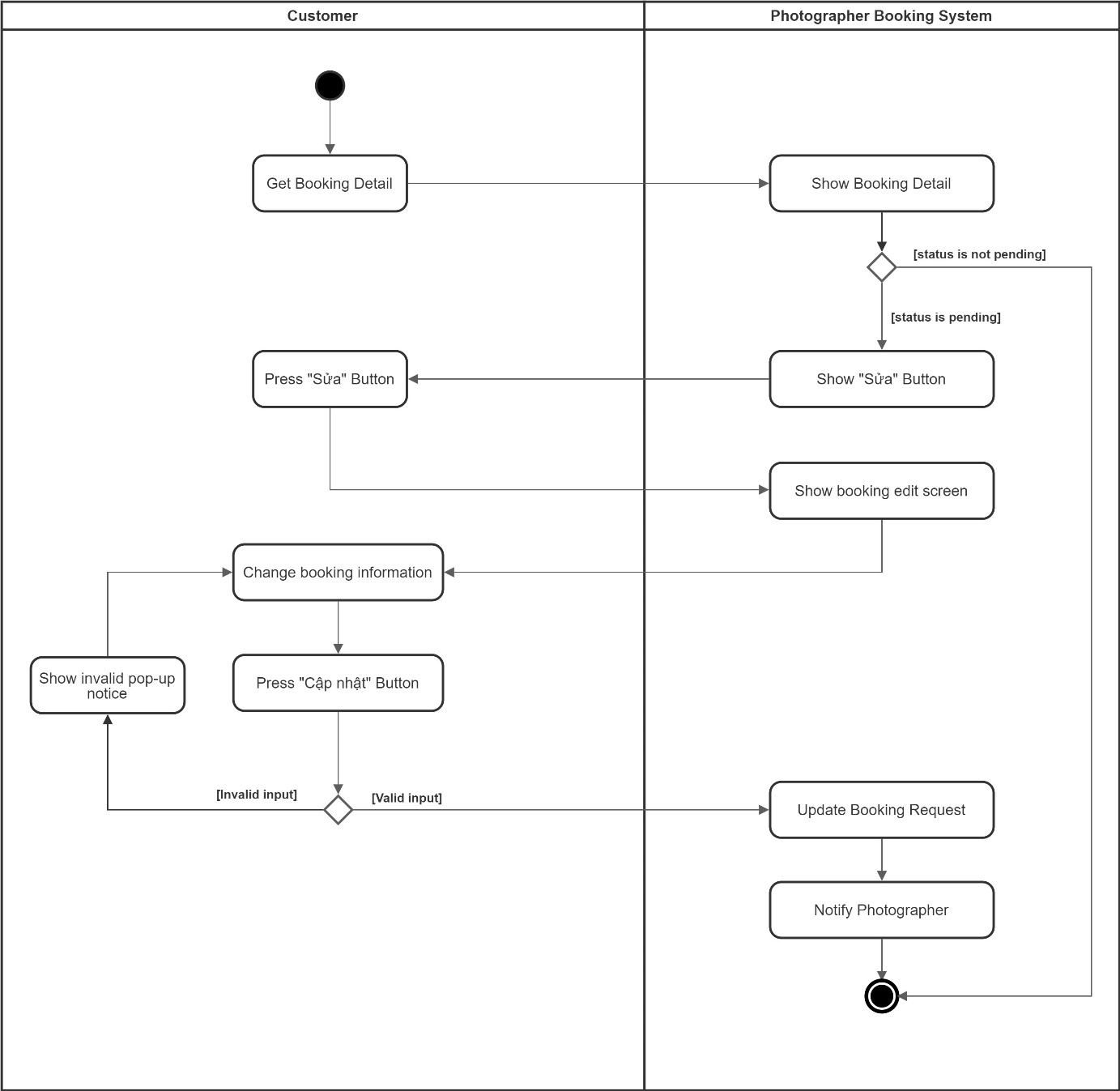


Figure 33 Activity Diagram – Edit Request

1. **<Customer> Submit booking cancellation**

Summary: This diagram shows the process when a customer cancels a booking.

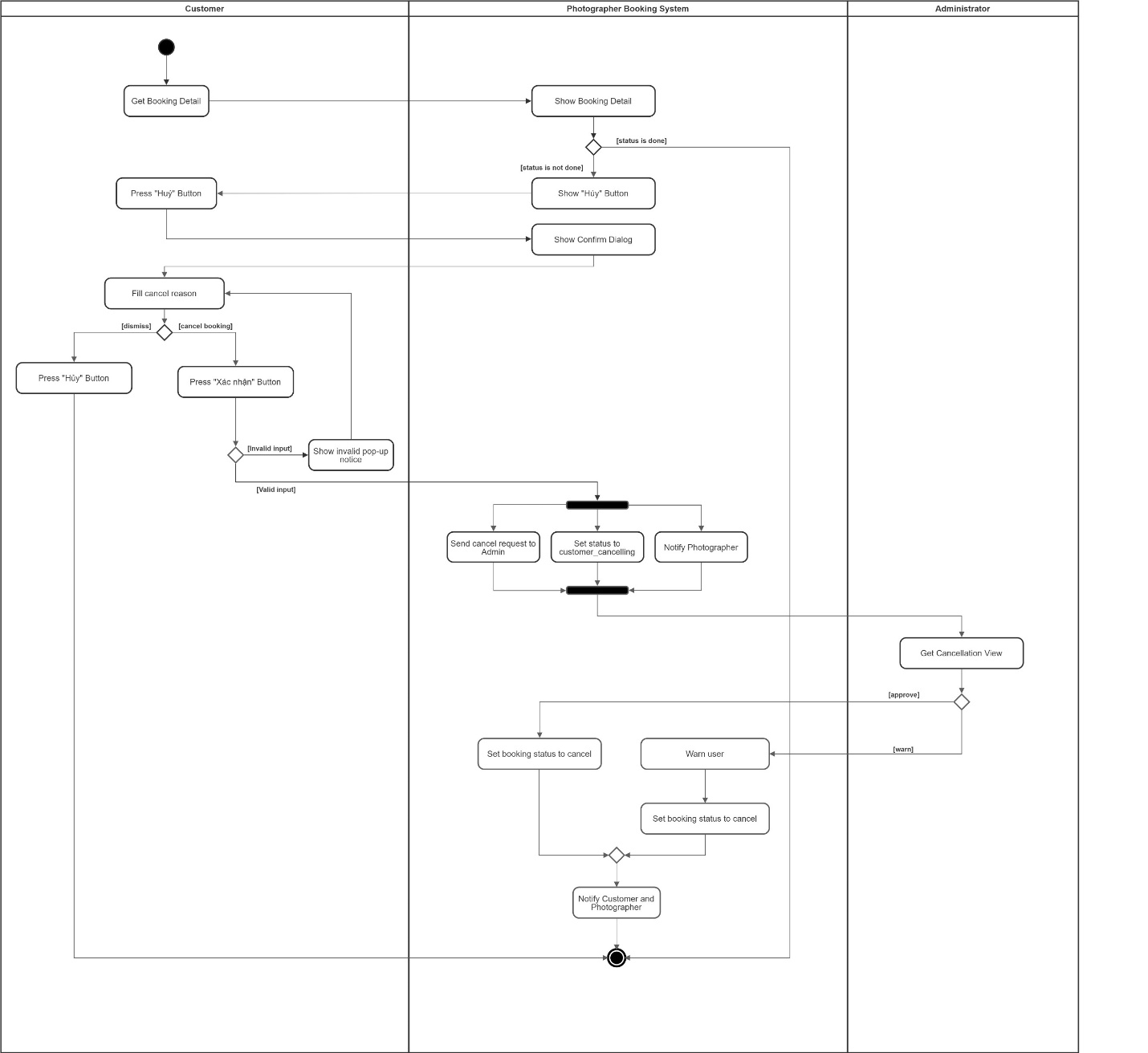


Figure 34 Activity Diagram – Submit booking cancellation

*Figure Activity Diagram - Rating*

1. **<Photographer> Accept booking**

Summary: This diagram shows the process when a photographer accepts a request.

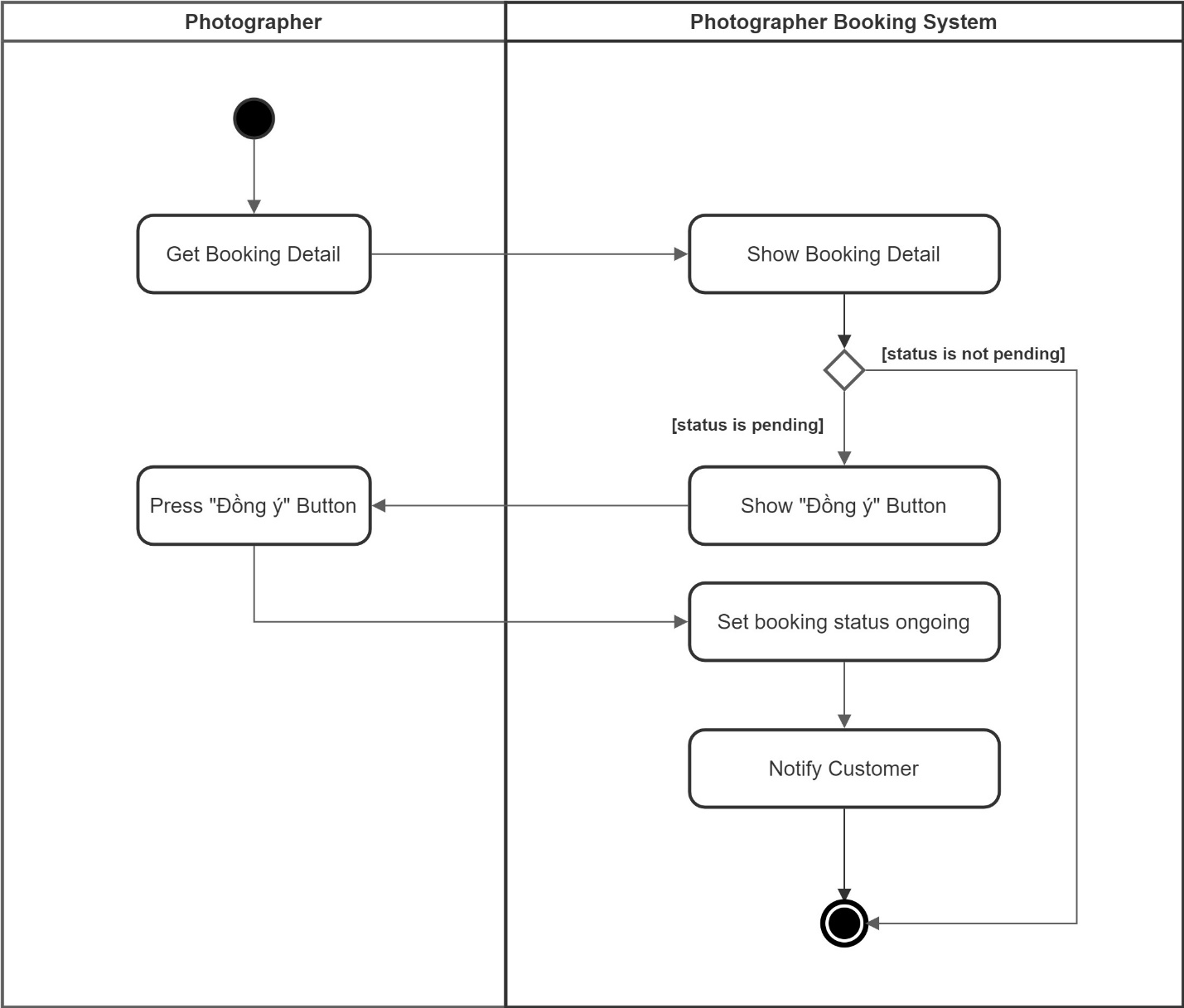


Figure 35 Activity Diagram - Accept booking

1. **<Photographer> Reject booking**

Summary: This diagram shows the process when a photographer rejects a request.

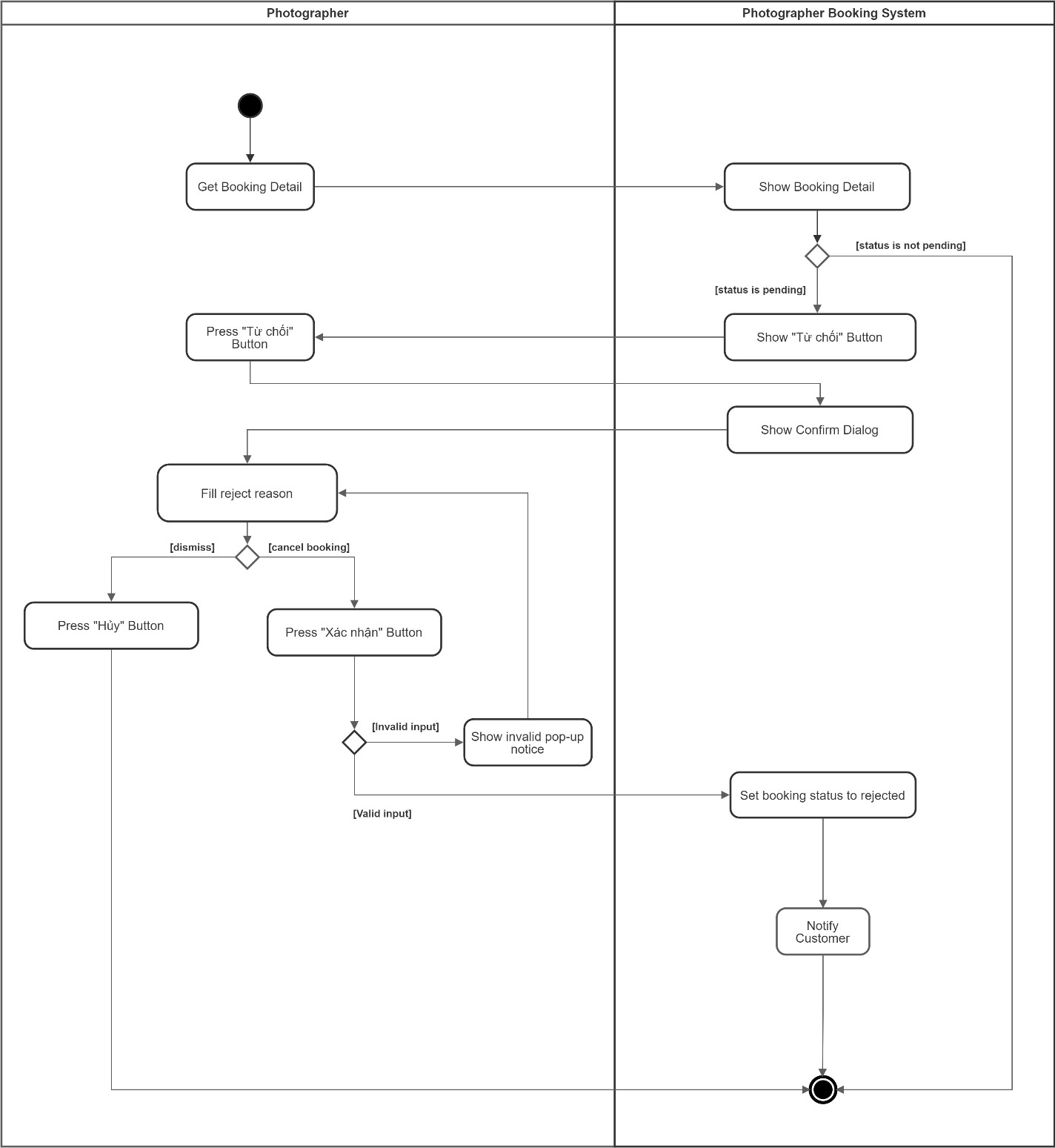


Figure 36 Activity Diagram - Reject booking

#### d. State Machine Diagram(s)

This state machine diagram describes states of a booking from start to done.

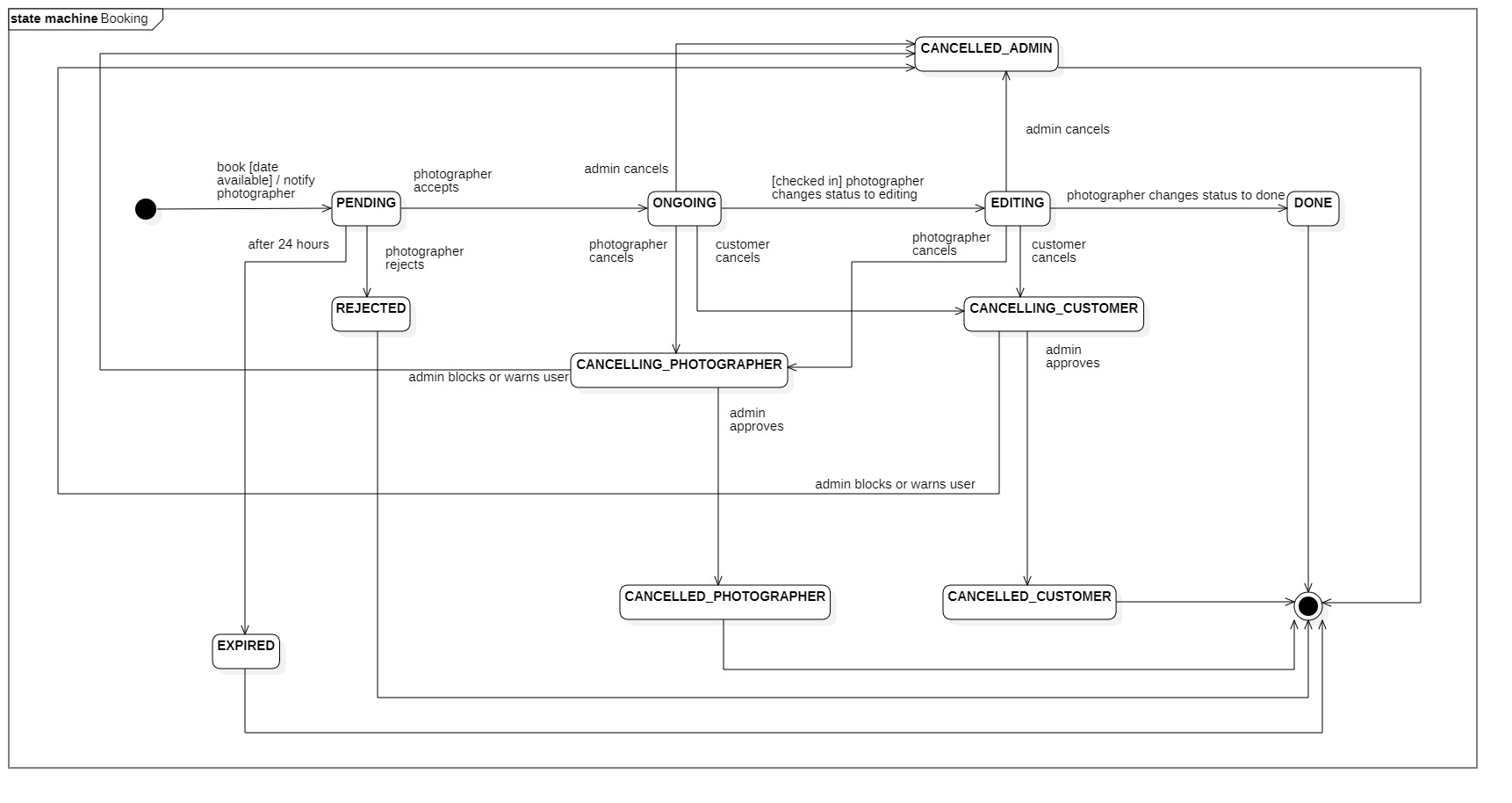


Figure 37 State Machine Diagram - Booking

## 4. Class Specifications

## 5. Data & Database Design

### 5.1 Database Design

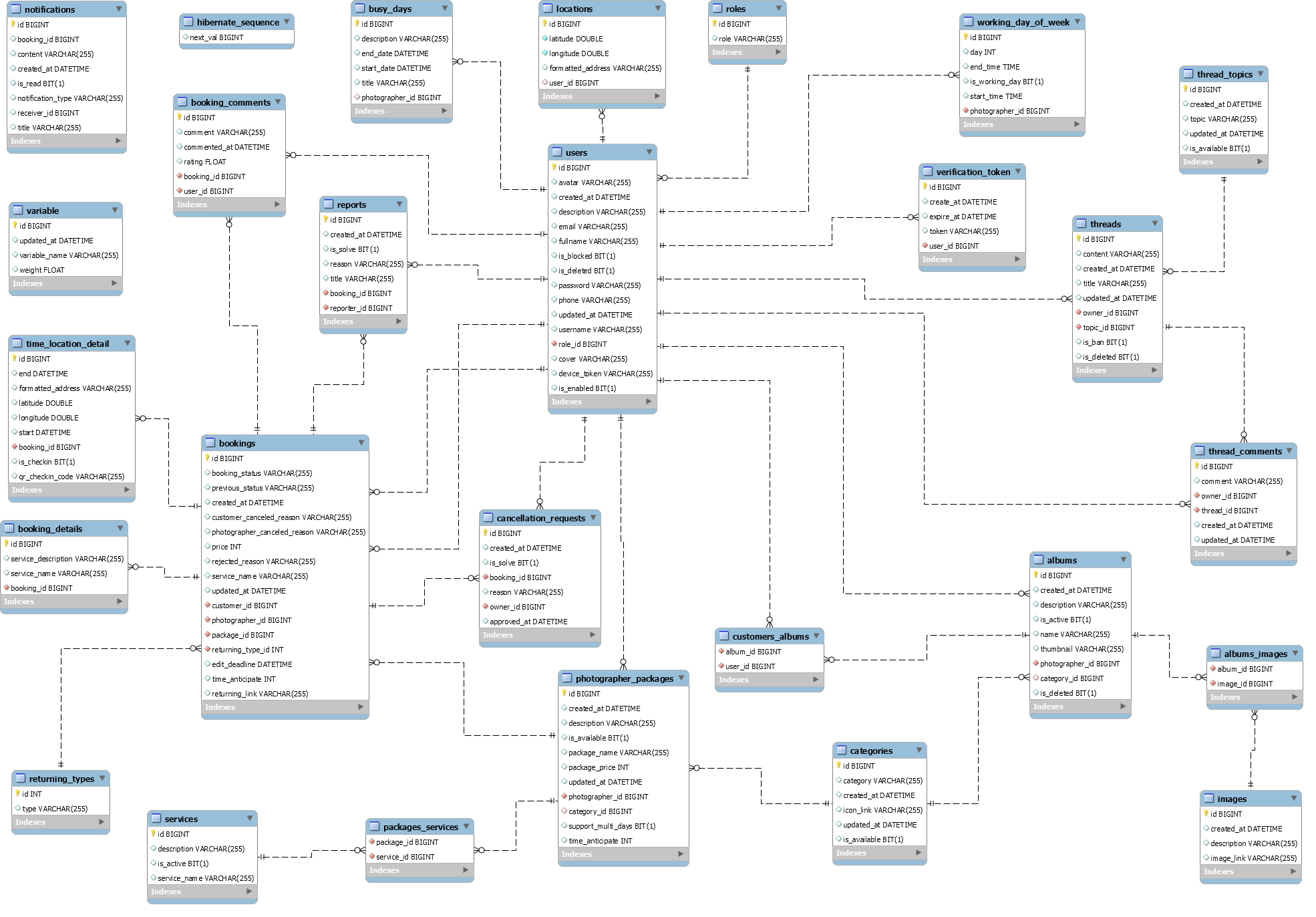


Figure 38 Database Design

## 6. Algorithm

### 6.1 Multiple-factors weighted sorting

#### 6.1.1 Definition

Multiple-factors weighted ranking is considered to be a simple yet effective algorithm for recommendation purpose.

#### 6.1.2 Define problem

A suggestion method is always needed in every kind of commercial application. Since it gives the customers an objective way to decide whether a photographer is trustable or not.

It also boosts the photographers’ motivation to give the best of them in order to connect to more customers.

#### 6.1.3 Solution

**Step 1: Choose suitable factors**

We will rank the photographers based on the following factors:

* Customers’ ratings.
* Photographer’s average service price.
* Distance (from photographer’s working location to the booked location).

**Step 2: Specify weight for each factor**

(Admin can change the weights)

|  |  |
| --- | --- |
| **Factor** | **Weight** |
| Rating | 0.3 |
| Distance | 0.3 |
| Price | 0.4 |

**Step 3: Set a common scale for factors**

Common scale = from 0 (worst) to 100 (best)

**Step 4: Transform each object’s factors score to the common scale**

* Rating: 1 rating point on scale of 5 will equal 20 points on scale of 100
* Distance:

Calculate total distances of all photographers in the list

Calculate score of a photographer’s distance factor = (1 – distance / sum) \* max scale

(photographers with higher distance will get lower score)

* Photographer’s average service price

Calculate total prices of all photographers in the list

Calculate score of a photographer’s price factor = (1 – price / sum) \* max scale

(Photographer with higher price will get lower score)

**Step 5: Sum the scores**

Score of a photographer = 0.3 \* rating score + 0.3 \* distance score + 0.4 \* price score

**Step 6: Sort photographers based on the score.**

**Note:**

For newcomers, when new photographers join, their rating would be 0.0, because of that they will have huge disadvantages because their rating will be lower than those who have bad rating (like 1.0 to 2.0).

For that reason, if photographer don’t have any rating yet, they will have the average rating of all photographers at that current time whenever the algorithm is triggered. After they have their booking rated, this will no longer affected.

#### 6.1.4 Complexity

The complexity of this algorithm is O(nlog(n)) for finding n photographers.

#### 6.1.5 Flowchart

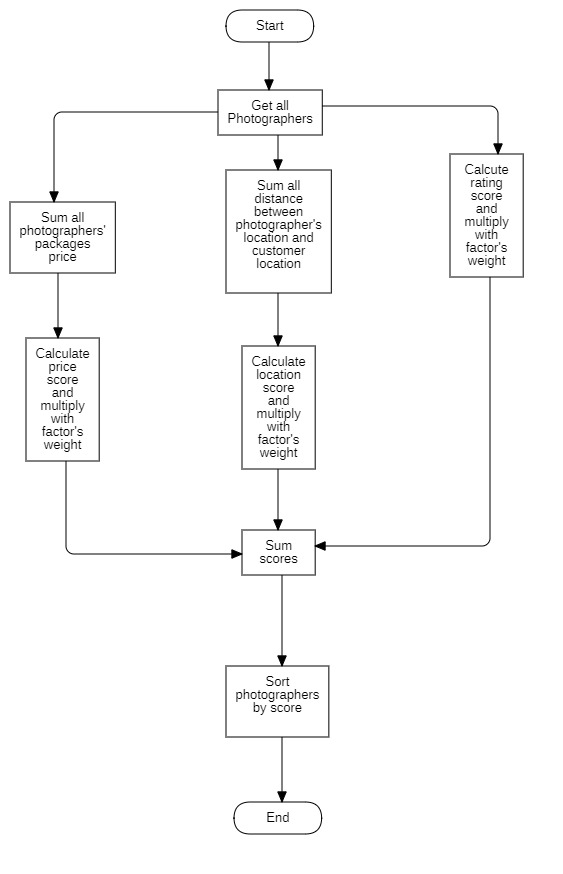


Figure 39 Flowchart - Multiple factors sorting

### 6.2 Naive Bayes Classifier

#### 6.2.1 Definition

Naive bayes classifiers is a simple technique mostly used for building classifiers based on Bayes’ theorem from probability theory and statistics.

It is best suited to solve problems which has categorizable variables Some application of naive bayes includes: Spam email filtering, sentiment prediction, or in our case, weather prediction.

#### 6.2.2 Define problem

Our application domain is photography, which has highly dependence on weather in order to make the most out of a shooting session. So, the problem here is we need a tool which could warn customer or photographer on bad weather days.

#### 6.2.3 Solution

Some definitions:

Bayes’ Theorem:

P(A|B) - Posterior: The probability of “A” being true given that “B” is already true.

P(B|A) - Likelihood: The probability of “B” being true given that “A” is already true.

P(A) - Prior: The probability of “A” being true.

P(B) - Evidence: The probability of “B” being true.

Input:

* A set of features x = {x1, x2… xn}, in our case x = {Outlook, Temperature, Humidity, Windy}
* A fixed set of classes C = {c1, c2… cn}, in our case C = {Yes, No}

Output

* A predicted class c from C.

For documenting purpose, we use a fictional example dataset below:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Outlook** | **Temperature** | **Humidity** | **Windy** | **Shooting** |
| Sunny | High | High | Strong | No |
| Overcast | Normal | Normal | Normal | Yes |
| Rainy | High | Normal | Strong | No |
| Sunny | Normal | Normal | Normal | Yes |
| Rainy | High | Normal | Strong | No |
| Overcast | Normal | High | Normal | Yes |
| Rainy | High | High | Strong | No |

We will convert this to Frequency tables:

|  |  |  |  |
| --- | --- | --- | --- |
| **Class** | **Outlook** | | |
| Sunny | Overcast | Rainy |
| Yes | 1 | 2 | 0 |
| No | 1 | 0 | 3 |

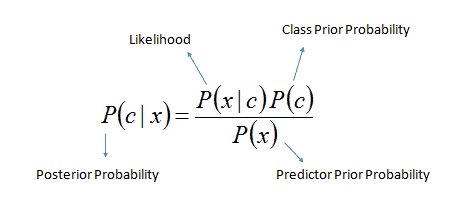
|  |  |  |
| --- | --- | --- |
| **Class** | **Temperature** | |
| High | Normal |
| Yes | 0 | 4 |
| No | 4 | 0 |

|  |  |  |
| --- | --- | --- |
| **Class** | **Humidity** | |
| High | Normal |
| Yes | 1 | 2 |
| No | 2 | 1 |

|  |  |  |
| --- | --- | --- |
| **Class** | **Windy** | |
| Strong | Normal |
| Yes | 0 | 4 |
| No | 4 | 0 |

For example, if we want to know if a set of features {Sunny, High Temperature, High Humidity, Strong Windy} is suitable for photographing or not.

According to Bayes theorem we need to calculate posterior probability.



Or we can simply calculate in expanded form:

posterior_probability_expanded

We calculate for each class (Yes and No) and then compare it to find out which gives higher score.

Since we want to classify sunny outlook, high temperature, high humidity and strong wind, we need to calculate following probabilities:

P (Yes) =

P (No) =

P (Sunny | Yes) =

P (Sunny | No) =

P (Temperature = High | Yes) =

In this case, to avoid the multiplication of probabilities become 0, we will add 1 to the value.

-> P (Temperature = High | Yes) =

P (Temperature = High | No) =

P (Humidity = High | Yes) =

P (Humidity = High | No) =

P (Windy = Strong | Yes) =

Same as the case above, we will add to 1 to the value.

-> P (Windy = Strong | Yes) =

P (Windy = Normal | No) =

Final step, we will find two posterior probabilities in expanded form (Without divide it for the evidence):

* P (Yes | X):

* P (No | X):

As 0.0714 > 0.00529 so we can predict that the weather is not suitable for shooting.

#### 6.2.4 Complexity

The complexity of Naive Bayes is O(nK), where n is number of features and K is number of classes.

#### 6.2.5 References

* <https://www.geeksforgeeks.org/naive-bayes-classifiers/>
* <https://en.wikipedia.org/wiki/Naive_Bayes_classifier>