|  |  |
| --- | --- |
|  | **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 -

**Table of Contents**

[Acknowledgement 4](#_Toc58759045)

[Definition and Acronyms 4](#_Toc58759046)

[I. Project Introduction 5](#_Toc58759047)

[1. Overview 5](#_Toc58759048)

[1.1 Project Information 5](#_Toc58759049)

[1.2 Project Team 5](#_Toc58759050)

[a. Supervisor 5](#_Toc58759051)

[b. Team Members 5](#_Toc58759052)

[2. Product Background 5](#_Toc58759053)

[3. Existing Systems 6](#_Toc58759054)

[3.1 WeHelp 6](#_Toc58759055)

[3.2 vLance 6](#_Toc58759056)

[3.3 Tripo 7](#_Toc58759057)

[4. Business Opportunity 7](#_Toc58759058)

[5. Software Product Vision 8](#_Toc58759059)

[6. Project Scope & Limitations 8](#_Toc58759060)

[6.1 Major Features 8](#_Toc58759061)

[6.2 Limitations & Exclusions 10](#_Toc58759062)

[II. Project Management Plan 11](#_Toc58759063)

[1. Overview 11](#_Toc58759064)

[1.1 WBS & Estimation 11](#_Toc58759065)

[1.2 Project Objectives 16](#_Toc58759066)

[1.3 Project Risks 17](#_Toc58759067)

[2. Management Approach 17](#_Toc58759068)

[2.1 Project Process 17](#_Toc58759069)

[2.2 Quality Management 18](#_Toc58759070)

[2.3 Training Plan 18](#_Toc58759071)

[3. Master Schedule 18](#_Toc58759072)

[4. Project Organization 20](#_Toc58759073)

[4.1 Team & Structures 20](#_Toc58759074)

[4.2 Roles & Responsibilities 20](#_Toc58759075)

[5. Project Communication 21](#_Toc58759076)

[5.1 Communication Plan 21](#_Toc58759077)

[5.2 External Interface 21](#_Toc58759078)

[a. FU Contacts 21](#_Toc58759079)

[b. Customer Contacts 22](#_Toc58759080)

[6. Configuration Management 22](#_Toc58759081)

[6.1 Tools & Infrastructures 22](#_Toc58759082)

[6.2 Document Management 22](#_Toc58759083)

[6.3 Source Code Management 22](#_Toc58759084)

[VII. Appendix 24](#_Toc58759085)

[1. Glossary [Optional] 24](#_Toc58759086)

[2. References [Optional] 24](#_Toc58759087)

[3. Others [Optional] 24](#_Toc58759088)

**List of Figures**

[Figure 1 Feature mind map 14](#_Toc58760377)

[Figure 2 Project process 22](file:///C:\Users\ACER\Downloads\merge%20documents\Report7_Final%20Project%20Report.docx#_Toc58760378)

[Figure 3 Project organization 24](file:///C:\Users\ACER\Downloads\merge%20documents\Report7_Final%20Project%20Report.docx#_Toc58760379)

[Figure 4 Context diagram 27](file:///C:\Users\ACER\Downloads\merge%20documents\Report7_Final%20Project%20Report.docx#_Toc58760380)

[Figure 5 Overview Use Case Diagram 28](#_Toc58760381)

[Figure 6 <Guest> Overview 31](#_Toc58760382)

[Figure 7<Guest> Register 31](#_Toc58760383)

[Figure 8 <Guest> Register Specification 33](#_Toc58760384)

[Figure 9 <Guest> Login 34](#_Toc58760385)

[Figure 10 <Customer> Overview 36](#_Toc58760386)

[Figure 11 <Customer> Search photographers 37](#_Toc58760387)

[Figure 12 <Customer> Get photographer's detail 38](#_Toc58760388)

[Figure 13 <Customer> Book photographer 39](#_Toc58760389)

[Figure 14 <Customer> Review photographers 41](#_Toc58760390)

[Figure 15 <Customer> Get booking 43](#_Toc58760391)

[Figure 16 <Customer> Edit booking 44](file:///C:\Users\ACER\Downloads\merge%20documents\Report7_Final%20Project%20Report.docx#_Toc58760392)

[Figure 17 <Customer> Cancel booking 47](file:///C:\Users\ACER\Downloads\merge%20documents\Report7_Final%20Project%20Report.docx#_Toc58760393)

[Figure 18 <Customer> Open thread 49](file:///C:\Users\ACER\Downloads\merge%20documents\Report7_Final%20Project%20Report.docx#_Toc58760394)

[Figure 19<Customer> Edit thread 50](#_Toc58760395)

[Figure 20 <Customer> Comment thread 53](#_Toc58760396)

[Figure 21 <Customer> Chat 55](#_Toc58760397)

[Figure 22 <Photographer> Overview 57](#_Toc58760398)

[Figure 23 <Photographer> Accept booking 58](file:///C:\Users\ACER\Downloads\merge%20documents\Report7_Final%20Project%20Report.docx#_Toc58760399)

[Figure 24 <Photographer> Decline booking 60](file:///C:\Users\ACER\Downloads\merge%20documents\Report7_Final%20Project%20Report.docx#_Toc58760400)

[Figure 25 <Photographer> Edit working weekdays 62](#_Toc58760401)

[Figure 26 <Photographer> Add busy day 63](#_Toc58760402)

[Figure 27 <Photographer> Get customer's review 65](#_Toc58760403)

[Figure 28 <Photographer> Add package 66](#_Toc58760404)

[Figure 29<Photographer> Edit package 68](#_Toc58760405)

[Figure 30 <Photographer> Remove package 70](#_Toc58760406)

[Figure 31 <Photographer> Add album 72](#_Toc58760407)

[Figure 32 <Photographer> Edit album 74](#_Toc58760408)

[Figure 33 <Photographer> Remove album 76](#_Toc58760409)

[Figure 34 <Authenticated User> Overview 78](#_Toc58760410)

[Figure 35 <Authenticated User> Get profile 78](#_Toc58760411)

[Figure 36 <Authenticated User> Edit profile 79](file:///C:\Users\ACER\Downloads\merge%20documents\Report7_Final%20Project%20Report.docx#_Toc58760412)

[Figure 37 <Authenticated User> Change password 81](#_Toc58760413)

[Figure 38 <Authenticated User> Logout 83](#_Toc58760414)

[Figure 39 <Admin> Overview 84](#_Toc58760415)

[Figure 40 <Admin> Block user 85](#_Toc58760416)

[Figure 41 <Admin> Unblock user 86](#_Toc58760417)

[Figure 42 <Admin> Add category 88](#_Toc58760418)

[Figure 43 <Admin> Edit category 89](#_Toc58760419)

[Figure 44 <Admin> Remove category 91](#_Toc58760420)

[Figure 45 <Admin> Add returning type 93](#_Toc58760421)

[Figure 46 <Admin> Edit returning type 94](#_Toc58760422)

[Figure 47 <Admin> Remove returning type 96](#_Toc58760423)

[Figure 48 <Admin> Change variables 98](#_Toc58760424)

[Figure 49<Admin> Ban thread 99](#_Toc58760425)

[Figure 50 <System Handler> Overview 101](#_Toc58760426)

[Figure 51 <System Handler> Make booking notification 101](#_Toc58760427)

[Figure 52 <System Handler> Make scheduling notification 103](#_Toc58760428)

[Figure 53 <System Handler> Make weather warning 104](#_Toc58760429)

[Figure 54 <System Handler> Make time warning 105](#_Toc58760430)

[Figure 55 Customer's screen flow 107](#_Toc58760431)

[Figure 56 Photographer's screen flow 108](#_Toc58760432)

[Figure 57 Screen authorization 113](#_Toc58760433)

[Figure 58 ER Diagram 115](#_Toc58760434)

[Figure 59 Overall Architecture 120](#_Toc58760435)

[Figure 60 Mobile App Architecture 121](#_Toc58760436)

[Figure 61 Webapi Package Diagram 122](#_Toc58760437)

[Figure 62 Package Diagram – Mobile app 124](#_Toc58760438)

[Figure 63 Class Diagram 125](#_Toc58760439)

[Figure 64 Sequence Diagram - Add Category 126](#_Toc58760440)

[Figure 65 Sequence Diagram - Edit Category 126](#_Toc58760441)

[Figure 66 Sequence Diagram - Remove Category 127](#_Toc58760442)

[Figure 67 Sequence Diagram - Add returning type 127](#_Toc58760443)

[Figure 68 Sequence Diagram - Edit returning type 128](#_Toc58760444)

[Figure 69 Sequence Diagram - Remove returning type 128](#_Toc58760445)

[Figure 70 Sequence Diagram - Block User 129](#_Toc58760446)

[Figure 71 Sequence Diagram - Unblock User 129](#_Toc58760447)

[Figure 72 Sequence Diagram - Change Variables 130](#_Toc58760448)

[Figure 73 Activity Diagram - Make Request 131](file:///C:\Users\ACER\Downloads\merge%20documents\Report7_Final%20Project%20Report.docx#_Toc58760449)

[Figure 74 Activity Diagram – Edit Request 132](#_Toc58760450)

[Figure 75 Activity Diagram - Cancel booking 133](#_Toc58760451)

[Figure 76 Activity Diagram - Rating 134](file:///C:\Users\ACER\Downloads\merge%20documents\Report7_Final%20Project%20Report.docx#_Toc58760452)

[Figure 77 Activity Diagram - Accept booking 135](#_Toc58760453)

[Figure 78 Activity Diagram - Reject booking 136](file:///C:\Users\ACER\Downloads\merge%20documents\Report7_Final%20Project%20Report.docx#_Toc58760454)

[Figure 79 State Machine Diagram - Booking 137](#_Toc58760455)

[Figure 80 Database Design 149](#_Toc58760456)

[Figure 81 Flowchart - Multiple factors sorting 158](#_Toc58760457)

[Figure 82 Test Model 163](file:///C:\Users\ACER\Downloads\merge%20documents\Report7_Final%20Project%20Report.docx#_Toc58760458)

[Figure 83 Setup files - 1 168](#_Toc58760459)

[Figure 84 Setup files - 2 168](#_Toc58760460)

[Figure 85 Installation - 1 169](#_Toc58760461)

[Figure 86 Installation - 2 170](#_Toc58760462)

[Figure 87 Installation - 3 171](#_Toc58760463)

[Figure 88 Installation - 4 172](#_Toc58760464)

[Figure 89 Deployment - 1 172](#_Toc58760465)

[Figure 90 Deployment -2 173](#_Toc58760466)

[Figure 91 Deployment - 3 173](#_Toc58760467)

[Figure 92 Register 176](#_Toc58760468)

[Figure 93 - Login 177](#_Toc58760469)

[Figure 94 - Logout 178](#_Toc58760470)

[Figure 95 Search photographers 179](#_Toc58760471)

[Figure 96 - Make request for a day 180](#_Toc58760472)

[Figure 97 - Make request for days 181](#_Toc58760473)

[Figure 98 - Update booking for a day 181](#_Toc58760474)

[Figure 99 - Update booking for days 182](#_Toc58760475)

[Figure 100 - Decline /Accept /Cancel Booking 182](#_Toc58760476)

[Figure 101 - View Reviews 183](#_Toc58760477)

[Figure 102 - Write reviews 184](#_Toc58760478)

[Figure 103 - Change Booking Status 185](#_Toc58760479)

[Figure 104 - Shooting schedule 186](#_Toc58760480)

[Figure 105 - Make changes to daily shooting schedule 187](#_Toc58760481)

[Figure 106 - Add busy days 188](#_Toc58760482)

[Figure 107 - Follow Album 189](#_Toc58760483)

[Figure 108 - Add Album 190](#_Toc58760484)

[Figure 109 - Edit Album 191](#_Toc58760485)

[Figure 110 - View Services 192](#_Toc58760486)

[Figure 111 - Add Service 192](#_Toc58760487)

[Figure 112 - Update Service 193](#_Toc58760488)

[Figure 113 - Select / View Categories 194](#_Toc58760489)

[Figure 114 - Chat 195](#_Toc58760490)

[Figure 115 - View Thread 196](#_Toc58760491)

[Figure 116 - View Comment 197](#_Toc58760492)

[Figure 117 - Add Comment 198](#_Toc58760493)

[Figure 118 - Create Thread 199](#_Toc58760494)

[Figure 119 - Select Photo Delivery Methods 200](#_Toc58760495)

[Figure 120 - Update Profile 201](#_Toc58760496)

[Figure 121 - Update Avatar/ Cover 202](#_Toc58760497)

[図122 – ログアウト 208](#_Toc58760498)

[図123-フォトグラファー／サービスの検索 209](#_Toc58760499)

[図124 –一日の予約 210](#_Toc58760500)

[図125 –数日の予約 211](#_Toc58760501)

[図126 - 一日の予約の変更 211](#_Toc58760502)

[図127 -数日の予約の変更 212](#_Toc58760503)

[図128 – 予約の断り/受入 /キャンセル 212](#_Toc58760504)

[図129 – レビューをを観ます 213](#_Toc58760505)

[図130 – レビューを書きます 214](#_Toc58760506)

[図131 - 予約の状態を変更します 215](#_Toc58760507)

[図132 - 撮るスケジュール 216](#_Toc58760508)

[図133 – 毎日の撮るスケジュールを変更します 217](#_Toc58760509)

[図134 – 都合がない日を追加します 218](#_Toc58760510)

[図135 – アルバムをフォローする 219](#_Toc58760511)

[図136 - アルバムを追加します 220](#_Toc58760512)

[図137 – アルバムを変更します 221](#_Toc58760513)

[図138 – サービスを観ます 222](#_Toc58760514)

[図139 - サービスを追加します 222](#_Toc58760515)

[図140 – サービスを変更します 223](#_Toc58760516)

[図141 – カテゴリーを観ます／選択します 224](#_Toc58760517)

[図142 – チャット 225](#_Toc58760518)

[図143 – スレッドを観ます 226](#_Toc58760519)

[図144 – コメントを観ます 227](#_Toc58760520)

[図145 – コメントをを追加します 228](#_Toc58760521)

[図146 – スレッドを追加します 229](#_Toc58760522)

[図147 - 配信方法を選びます 230](#_Toc58760523)

[図148 – プロファイルを編集します 231](#_Toc58760524)

[図149 –アバター／コバーを変更します 232](#_Toc58760525)

**List of Tables**

[Table 1 Team members 12](#_Toc58760526)

[Table 2 WBS & Estimation 23](#_Toc58760527)

[Table 3 Project objectives 23](#_Toc58760528)

[Table 4 Project risks 24](#_Toc58760529)

[Table 5 Training plan 25](#_Toc58760530)

[Table 6 Master schedule 26](#_Toc58760531)

[Table 7 Roles & Responsibilities 27](#_Toc58760532)

[Table 8 Communication plan 28](#_Toc58760533)

[Table 9 FU Contacts 28](#_Toc58760534)

[Table 10 Customer contacts 29](#_Toc58760535)

[Table 11 Tools & Infrastructures 29](#_Toc58760536)

[Table 12 Business Rules 30](#_Toc58760537)

[Table 13 System Actors 31](#_Toc58760538)

[Table 14 Use Case List 33](#_Toc58760539)

[Table 15 <Guest> Register Specification 38](#_Toc58760540)

[Table 16 <Customer> Search Photographers Specification 41](#_Toc58760541)

[Table 17 <Customer> Get photographer's detail specification 42](#_Toc58760542)

[Table 18 <Customer> Book photographer specification 44](#_Toc58760543)

[Table 19 <Customer> Review photographers speicification 46](#_Toc58760544)

[Table 20 <Customer> Get booking specification 47](#_Toc58760545)

[Table 21 <Customer> Edit booking specification 49](#_Toc58760546)

[Table 22 <Customer> Cancel booking specification 51](#_Toc58760547)

[Table 23 <Customer> Open thread specification 53](#_Toc58760548)

[Table 24 <Customer> Edit thread specification 55](#_Toc58760549)

[Table 25 <Customer> Comment thread specification 57](#_Toc58760550)

[Table 26 <Customer> Chat specification 59](#_Toc58760551)

[Table 27 <Photographer> Accept booking specification 62](#_Toc58760552)

[Table 28 <Photographer> Decline booking specification 64](#_Toc58760553)

[Table 29 Edit working weekdays specification 66](#_Toc58760554)

[Table 30 <Photographer> Add busy day specification 68](#_Toc58760555)

[Table 31<Photographer> Get customer's review specification 69](#_Toc58760556)

[Table 32 <Photographer> Add package specification 71](#_Toc58760557)

[Table 33 <Photographer> Edit package specification 73](#_Toc58760558)

[Table 34 <Photographer> Remove package specification 75](#_Toc58760559)

[Table 35 <Photographer> Add album specification 77](#_Toc58760560)

[Table 36 <Photographer> Edit album specification 79](#_Toc58760561)

[Table 37<Photographer> Remove album specification 80](#_Toc58760562)

[Table 38 <Authenticated User> Get profile specification 82](#_Toc58760563)

[Table 39 <Authenticated User> Edit profile specification 84](#_Toc58760564)

[Table 40 <Authenticated User> Change password specification 85](#_Toc58760565)

[Table 41 <Authenticated User> Logout specification 87](#_Toc58760566)

[Table 42 <Admin> Block user specification 89](#_Toc58760567)

[Table 43 <Admin> Unblock user specification 91](#_Toc58760568)

[Table 44 <Admin> Add category specification 92](#_Toc58760569)

[Table 45 <Admin> Edit category specification 94](#_Toc58760570)

[Table 46 <Admin> Remove category specification 95](#_Toc58760571)

[Table 47 <Admin> Add returning type specification 97](#_Toc58760572)

[Table 48 <Admin> Edit returning type specification 99](#_Toc58760573)

[Table 49 <Admin> Remove returning type specification 100](#_Toc58760574)

[Table 50 <Admin> Change variables specification 102](#_Toc58760575)

[Table 51 <Admin> Ban thread specification 104](#_Toc58760576)

[Table 52 <System Handler> Make booking notification specification 106](#_Toc58760577)

[Table 53 <System Handler> Make scheduling notification specification 107](#_Toc58760578)

[Table 54 <System Handler> Make weather warning specification 108](#_Toc58760579)

[Table 55<System Handler> Make time warning specification 109](#_Toc58760580)

[Table 56 Customer's screen details 113](#_Toc58760581)

[Table 57 Photographer's screen details 115](#_Toc58760582)

[Table 58 Non-screen functions 117](#_Toc58760583)

[Table 59 Entities List 119](#_Toc58760584)

[Table 60 Messages List 122](#_Toc58760585)

[Table 61 Package Diagram Description 126](#_Toc58760586)

[Table 62 Package Diagram Description 128](#_Toc58760587)

[Table 63 Class Diagram - User 141](#_Toc58760588)

[Table 64 Class Diagram - Role 141](#_Toc58760589)

[Table 65 Class Diagram - ERole 142](#_Toc58760590)

[Table 66 Class Diagram - Package 142](#_Toc58760591)

[Table 67 Class Diagram - Service 143](#_Toc58760592)

[Table 68 Class Diagram - Booking 143](#_Toc58760593)

[Table 69 Class Diagram - EBookingStatus 144](#_Toc58760594)

[Table 70 Class Diagram - Album 145](#_Toc58760595)

[Table 71 Class Diagram - Image 145](#_Toc58760596)

[Table 72 Class Diagram - Category 146](#_Toc58760597)

[Table 73 Class Diagram - DayOfWeek 146](#_Toc58760598)

[Table 74 Class Diagram - Location 147](#_Toc58760599)

[Table 75 Class Diagram - BookingComment 147](#_Toc58760600)

[Table 76 Class Diagram - BookingDetail 148](#_Toc58760601)

[Table 77 Class Diagram - BusyDay 148](#_Toc58760602)

[Table 78 Class Diagram - Thread 149](#_Toc58760603)

[Table 79 Class Diagram - ThreadTopic 150](#_Toc58760604)

[Table 80 Class Diagram - ThreadComment 150](#_Toc58760605)

[Table 81 Class Diagram - ReturningType 151](#_Toc58760606)

[Table 82 Class Diagram - EReturningType 151](#_Toc58760607)

[Table 83 Class Diagram - TimeLocationDetail 151](#_Toc58760608)

[Table 84 Database - users 153](#_Toc58760609)

[Table 85 Database - roles 153](#_Toc58760610)

[Table 86 Database - photographer\_packages 153](#_Toc58760611)

[Table 87 Database - packages\_services 153](#_Toc58760612)

[Table 88 Database - services 154](#_Toc58760613)

[Table 89 Database - bookings 155](#_Toc58760614)

[Table 90 Database - albums 155](#_Toc58760615)

[Table 91 Database - locations 155](#_Toc58760616)

[Table 92 Database - categories 155](#_Toc58760617)

[Table 93 Database - album\_images 156](#_Toc58760618)

[Table 94 Database - images 156](#_Toc58760619)

[Table 95 Database - customer\_albums 156](#_Toc58760620)

[Table 96 Database - working\_day\_of\_week 156](#_Toc58760621)

[Table 97 Database - busy\_days 157](#_Toc58760622)

[Table 98 Database - returning\_types 157](#_Toc58760623)

[Table 99 Database - booking\_details 157](#_Toc58760624)

[Table 100 Database - time\_location\_details 158](#_Toc58760625)

[Table 101 Database - booking\_comments 158](#_Toc58760626)

[Table 102 Database - threads 158](#_Toc58760627)

[Table 103 Database - thread\_comments 158](#_Toc58760628)

[Table 104 Database - thread\_topics 159](#_Toc58760629)

[Table 105 Date File Design 159](#_Toc58760630)

[Table 106 Testing levels 166](#_Toc58760631)

[Table 107 Testing Types 167](#_Toc58760632)

[Table 108 Test Stages 167](#_Toc58760633)

[Table 109 Human Resources 167](#_Toc58760634)

[Table 110 Environment 168](#_Toc58760635)

[Table 111 Test Milestones 168](#_Toc58760636)

[Table 112 Deliverables 168](#_Toc58760637)

[Table 113 Test Cases 168](#_Toc58760638)

[Table 114 Test Reports 168](#_Toc58760639)

[Table 115 Deliverable Package 169](#_Toc58760640)

[Table 116 PC Hardware requirements 170](#_Toc58760641)

[Table 117 Android Hardware requirements 170](#_Toc58760642)

[Table 118 Software requirements 171](#_Toc58760643)

# Acknowledgement

# Definition and Acronyms

|  |  |
| --- | --- |
| **Acronym** | **Definition** |
| PWM | Psychology website |
| AWS | Amazon Web Services |
| BA | Business Analysis |
| BR | Business Rule |
| ERD | Entity Relationship Diagram |
| GUI | Graphical User Interface |
| PM | Project Manager |
| SDD | Software Design Description |
| SPMP | Software Project Management Plan |
| SRS | Software Requirement Specification |
| UAT | User Acceptance Test |
| UC | Use Case |
| API | Application Program Interface |

# 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 |  | Lecturer |

#### b. Team Members

|  |  |  |  |
| --- | --- | --- | --- |
| **Full Name** | **Email** | **Mobile** | **Role** |
| Trần Thiên Thảo | Thaottse62888@fpt.edu.vn | 0903917982 | Leader |
| Bồ Công Đạt | Datbcse63154@fpt.edu.vn | 0915223623 | Member |
| Đào Sỹ Trung Kiên | Kiendstse63208@fpt.edu.vn | 0769606867 | Member |
| Trương Ngọc Mỹ | Mytnse62050@fpt.edu.vn | 0906969071 | Member |

Table 1 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.

## 6. Project Scope & Limitations

### 6.1 Major Features



Figure 1 Feature mind map

# II. Project Management Plan

## 1. Overview

### 1.1 WBS & Estimation

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **WBS Item** | **Complexity** | **Est. Effort**  **(man-days)** |
| ***1*** | ***Initiating*** |  | ***5*** |
| 1.1 | Kick-off project   * Identify Project Goals & Vision * Get the Right Team Together * Define the Right Key Performance Indicators (KPIs) or Timeline * Project Tools & Methodologies * Project Kick-off Meeting Planning * Project Kick-off Meeting | Medium | 1 |
| 1.2 | Develop business case   * Confirm the opportunity * Analyse and develop shortlisted options * Evaluate the options * Implementation strategy * Recommendation | Medium | 2 |
| 1.3 | Assign Project Manager | Simple | 1 |
| 1.4 | Identify stake-holder   * Identifying * Analysing  1. Stakeholder Type 2. Contribution (value) 3. Legitimacy 4. Willingness to engage 5. Influence 6. Involvement  * Mapping | Medium | 1 |
| ***2*** | ***Planning*** |  | ***14*** |
| 2.1 | Create business plan   * Executive summary * Opportunity * Execution * Company and management summary * Financial plan * Appendix | Medium | 2 |
| 2.2 | Collect requirement | Complex | 7 |
| 2.3 | Define scope   * Identify the project needs * Confirm the objectives and goals of the Project * Project Scope description * Expectations and acceptance * Identify constraints * Identify necessary changes | Medium | 1 |
| 2.4 | Define vision   * Define outcome * Define what unique brings to the above outcome * Apply Add relatable, human, aspects | Medium | 1 |
| 2.5 | Create WBS   * Identify the most important pieces of scope * Break down the project * Identify the known attributes for each activity | Medium | 1 |
| 2.6 | Create Risk Management Plan   * Identify the Risk * Analyse the Risk * Evaluate or Rank the Risk * Treat the Risk * Monitor and Review the Risk | Medium | 1 |
| 2.7 | Create Human Resource Plan   * Analyse the objectives * Make an inventory of current human resources * Forecast your HR demand * Determine the number and extent of skills gaps * Draw up an action plan * Integrate and implement the plan * Monitoring, measurement, and feedback | Simple | 1 |
| ***3*** | ***Executing*** |  | ***22*** |
| **3.1** | **Analysis** |  |  |
| 3.1.1 | Analyse Requirements   * Identify customer's needs. * Evaluate system for feasibility. * Perform economic and technical analysis. * Allocate functions to system elements. * Establish schedule and constraints. * Create system definitions. | Medium | 2 |
| 3.1.2 | Create Use case diagram   * Identify the Actors * For each category of users, identify all roles played by the users relevant to the system * Identify what are the users required the system to be performed to achieve these goals * Create use cases for every goal * Structure the use cases | Medium | 1 |
| 3.1.3 | Create Use case specification   * Create a use case model showing the use cases and actors * Create an overview of the steps (content) of the use case * Write the use case specification | Complex | 1 |
| 3.1.4 | Create SRS   * Create an Outline * Start with a Purpose * Give an Overview * Detail Specific Requirements * Get Approval for the SRS | Medium | 2 |
| **3.2** | **Design** |  |  |
| 3.2.1 | Design System Architecture   * Analyse the requirements * Define use cases for the system * Identify processor/modules to implement the use cases * Select operating system and hardware platform * Define sequence diagrams | Medium | 2 |
| 3.2.2 | Design System Detail (Diagrams) | Medium | 3 |
| 3.2.3 | Design Database   * Determine the purpose * Find and organize the information required * Divide the information into tables * Turn information items into columns * Specify primary keys * Set up the table relationships * Refine your design * Apply the normalization rules | Medium | 3 |
| 3.2.4 | Design User Interface   * Briefing * Analytics * Wireframing * UX prototype * UI creation | Complex | 3 |
| **3.3** | **Prototype** |  |  |
| 3.3.1 | Define storyboard | Medium | 1 |
| 3.3.2 | Create paper prototype | Simple | 1 |
| 3.3.3 | Create digital prototype | Complex | 3 |
| **3.4** | **Implementation** |  |  |
| 3.4.1 | Coding/ implement every module defined in SRS | ***Medium*** | ***30*** |
| **3.5** | **Perform quality assurance** |  | ***7*** |
| 3.5.1 | Perform unit testing | Simple | 1 |
| 3.5.2 | Perform integration testing | Medium | 2 |
| 3.5.3 | Perform system testing | Medium | 2 |
| 3.5.4 | Perform acceptance testing | Medium | 2 |
| **3.6** | **Support** |  | ***7*** |
| 3.6.1 | Training | Medium | 2 |
| 3.6.2 | Documentation | Medium | 2 |
| 3.6.3 | User Support | Simple | 1 |
| 3.6.4 | Enhancement | Medium | 2 |
| ***4*** | ***Monitoring*** |  | ***3*** |
| 4.1 | Validate Scope | Medium | 2 |
| 4.2 | Monitor project work | Medium | 1 |
| ***5*** | ***Controlling*** |  | ***5*** |
| 5.1 | Control scope | *Medium* | *1* |
| 5.2 | Control quality | *Medium* | *1* |
| 5.3 | Control risks | *Medium* | *1* |
| 5.4 | Control schedule | *Medium* | *1* |
| 5.5 | Control communication | *Simple* | *1* |
| ***6*** | ***Closing*** |  | ***1*** |
| 6.1 | Close project | Simple | 1 |
| ***Total Estimated Effort (man-days)*** | | | FINAL PRESENTATION |
|  | | | ***94*** |

Table 2 WBS & Estimation

## 2. Management Approach

### 2.1 Project Process

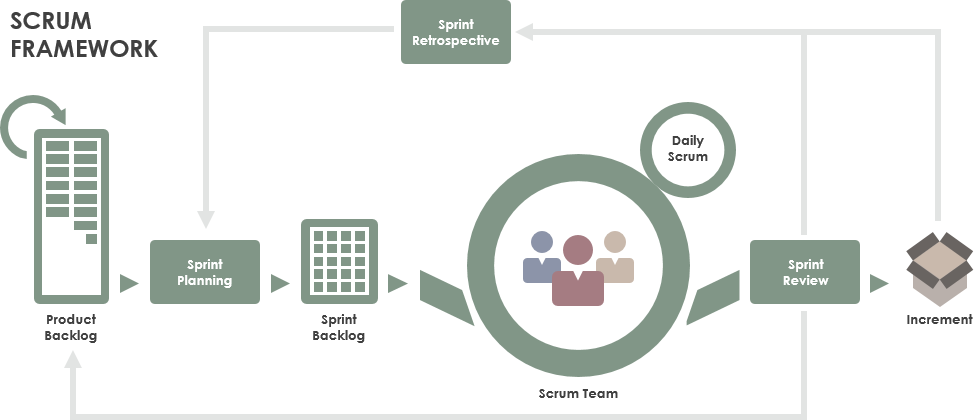
*Reference:* [*https://www.scrum.org/resources/what-is-scrum*](https://www.scrum.org/resources/what-is-scrum)

Figure 2 Project process

***Figure SEQ Figure \\* ARABIC 1-Scrum Framework***

# III. Software Requirement Specification

## 2. User Requirements

### 2.1 Overview

#### a. Use Case Diagram

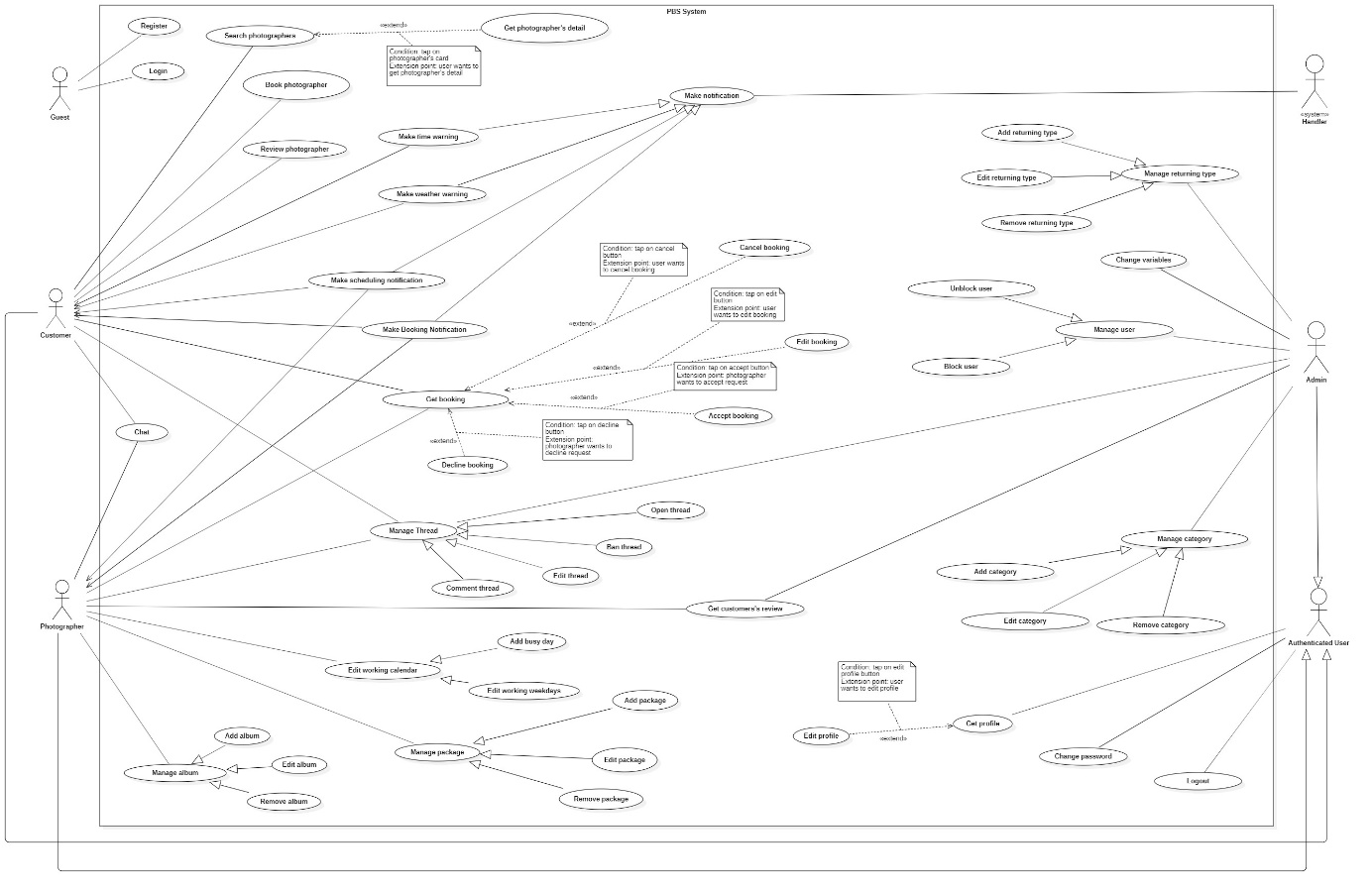


Figure 3 Overview Use Case Diagram

#### b. System Actors

|  |  |  |
| --- | --- | --- |
| **No** | **Actor** | **Description** |
| 1 | Administrator | Administrators manage the whole system |
| 2 | Photographer | Photographers offer shooting services and handle customers’ requests |
| 3 | Customer | Customers make bookings to use photographers’ services |
| 4 | Guest | Guests are the ones without accounts |
| 5 | Authenticated User | Authenticated Users are the ones have accounts (admins, photographers, customers) |
| 6 | <<system>> Handler | <<system>> Handler deals with internal process. |

Table 13 System Actors

#### c. Use Cases List

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Use Case** | **Primary Actors** | **Secondary Actors** |
| 01 | Register | Guest |  |
| 02 | Login | Guest |  |
| 03 | Search photographers | Customer |  |
| 04 | Get photographer’s detail | Customer |  |
| 05 | Book photographer | Customer |  |
| 06 | Review photographer | Customer |  |
| 07 | Get booking | Customer, Photographer |  |
| 08 | Edit booking | Customer |  |
| 09 | Cancel booking | Customer |  |
| 10 | Accept booking | Photographer |  |
| 11 | Decline booking | Photographer |  |
| 12 | Open thread | Customer, Photographer, Admin |  |
| 13 | Edit thread | Customer, Photographer,  Admin |  |
| 14 | Comment thread | Customer, Photographer, Admin |  |
| 15 | Chat | Customer, Photographer |  |
| 16 | Edit working weekdays | Photographer |  |
| 17 | Add busy day | Photographer |  |
| 18 | Get customer’s review | Photographer, Admin |  |
| 19 | Add package | Photographer |  |
| 20 | Edit package | Photographer |  |
| 21 | Remove package | Photographer |  |
| 22 | Add album | Photographer |  |
| 23 | Edit album | Photographer |  |
| 24 | Remove album | Photographer |  |
| 25 | Get profile | Authenticated User |  |
| 26 | Edit profile | Authenticated User |  |
| 27 | Change password | Authenticated User |  |
| 28 | Logout | Authenticated User |  |
| 29 | Block user | Admin |  |
| 30 | Unblock user | Admin |  |
| 31 | Add category | Admin |  |
| 32 | Edit category | Admin |  |
| 33 | Remove category | Admin |  |
| 34 | Add returning type | Admin |  |
| 35 | Edit returning type | Admin |  |
| 36 | Remove returning type | Admin |  |
| 37 | Change variables | Admin |  |
| 38 | Ban thread | Admin |  |
| 39 | Make booking notification | System Handler |  |
| 40 | Make scheduling notification | System Handler |  |
| 41 | Make weather warning | System Handler |  |
| 42 | Make time warning | System Handler |  |

Table 14 Use Case List

### 2.2 List of use case

#### 2.2.2 <Customer> Overview use case

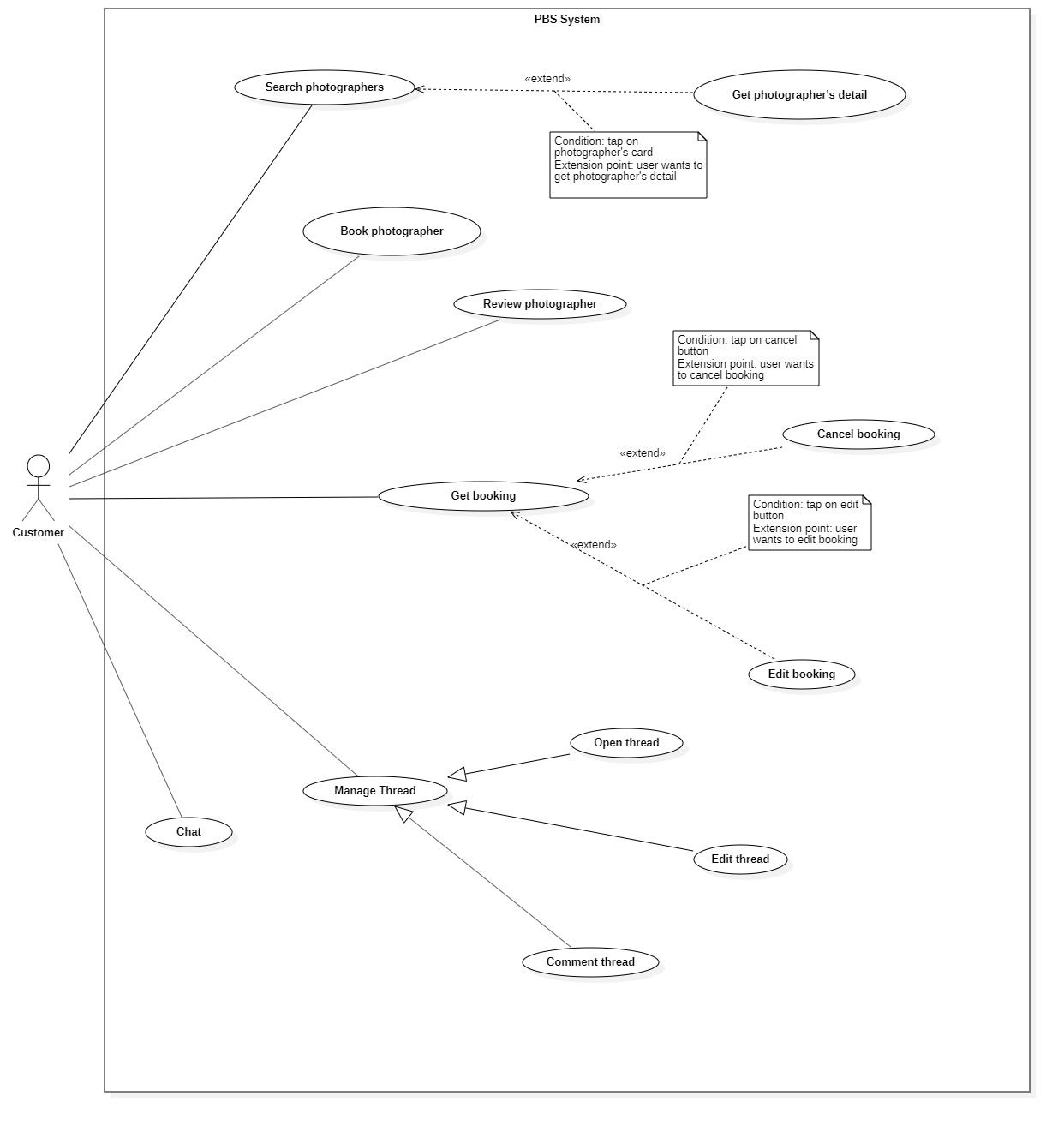


Figure 4 <Customer> Overview

##### <Customer> Book photographer

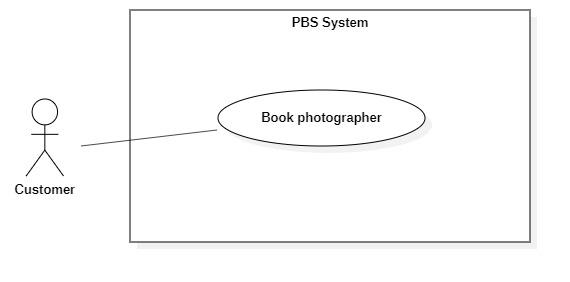
******

Figure 5 <Customer> Book photographer

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| UC ID and Name: | **UC – 05 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 as “active”. | | | | |
| 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  - “Phương thức nhận ảnh”: 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, Cancelled  - 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 price, package’s services  - Time and location detail contain: shooting date and time, shooting location  - Photographer’s detail: photographer’s name  - Customer’s detail: customer’s name, customer’s phone  - Photographer returned by the system will contain: photographer’s information, photographer’s package detail  - After booking successfully, a new booking will have “Pending” status. | | | | |
| Other Information: | N/A | | | | |
| Assumptions: | N/A | | | | |

Table 18 <Customer> Book photographer specification

##### <Customer> Get booking

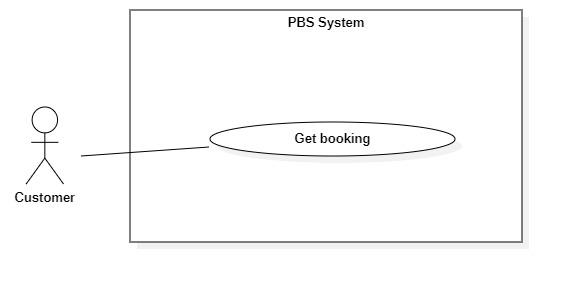
******

Figure 6 <Customer> Get booking

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 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 a booking list. | | | | |
| Description: | This use case allows customers or photographers to get their booking list of all status. | | | | |
| Preconditions: | - User has been authenticated as Customer / Photographer | | | | |
| Post-conditions: | **Success:** System displays list of bookings.  **Fail:** System shows error message. | | | | |
| Normal Flow: | Step | Actor Action | | | System Response |
| 1 | Customer / Photographer sends a request to get a list of bookings. | | | System returns list of bookings.  [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 list of bookings is displayed and ordered by most recent activity.  - Booking has following status:   * Pending * Ongoing * Editing * Done * Rejected * Cancelled   - Booking has following information: Name of the person who booked, time details (Datetime), location details, booking status, booked packages. | | | | |
| Other Information: | N/A | | | | |
| Assumptions: | N/A | | | | |

Table 20 <Customer> Get booking specification

##### <Customer> Edit Booking

Figure 7 <Customer> Edit booking

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| UC ID and Name: | **UC – 08 Edit booking** | | | | |
| Created By: | Trần Thiên Thảo | | Date Created: | 08/10/2020 | |
| Primary Actor: | Customer | | Secondary Actors: |  | |
| Trigger: | The Customer sends a request to edit booking. | | | | |
| Description: | This use case allows customer to edit their booking. | | | | |
| Preconditions: | - The User has been authenticated as Customer. | | | | |
| Post-conditions: | **Success:** Booking successfully edited and save to the system.  **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”: text  - “Thời gian tác nghiệp dự kiến”: text  - “Thời gian trả ảnh”: text  - “Địa điểm”: map  - “Thông tin khách hàng”:   * “Tên khách hàng”: text   - “Tên gói dịch vụ”:   * “Tên gói”: text * “Mô tả”: text * “Giá dịch vụ: text * “Dịch vụ”: list of texts   [Exception 1] |
|  | Customer inputs information and sends request to edit booking | | | System outputs message: “Bạn muốn thay đổi thông tin cuộc hẹn này?” |
|  | Customer confirm “Có”  [Alternative 1] | | | - The system validates data  - The system adds new booking to the system, outputs “Cập nhật thành công” and send to photographer.  [Exception 2]  [Exception 3] |
| 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 show error message: “Không tìm thấy cuộc hẹn” |
| 2 | The customer inputs invalid data or leaves empty required fields | | | System outputs error message and send request to ask Customer to input correctly |
| 3 | 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: | Sometimes | | | | |
| Business Rules: | * Customer can only edit the booking if its status is “Pending”. * Customer can edit information like shooting time, shooting location, package and send request to photographer. | | | | |
| Other Information: | N/A | | | | |
| Assumptions: | N/A | | | | |

Table 21 <Customer> Edit booking specification

##### <Customer> Cancel booking

Figure 8 <Customer> Cancel booking

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| UC ID and Name: | **UC – 09 Cancel 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 cancel booking. | | | | |
| Description: | This use case allows customers or photographers to cancel their booking. | | | | |
| Preconditions: | - The User has been authenticated as Customer / Photographer. | | | | |
| Post-conditions: | **Success:** The customer cancels 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”: text  - “Thời gian tác nghiệp dự kiến”: text  - “Thời gian trả ảnh”: text  - “Địa điểm”: map  - “Thông tin khách hàng”:   * “Tên khách hàng”: text   - “Tên gói dịch vụ”:   * “Tên gói”: text * “Mô tả”: text * “Giá dịch vụ: text * “Dịch vụ”: list of texts   [Exception 1] |
| 2 | Customer sends request to cancel booking | | | System outputs message: “Bạn muốn hủy cuộc hẹn này?” |
| 3 | Customer confirm “Có” and inputs cancelling reasons to cancel booking  [Alternative 1] | | | System outputs message: “Đã 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 show error message: “Không tìm thấy cuộc hẹn” |
| 2 | The process run into an internal error | | | System show error message: “Đã có lỗi xảy ra” |
| Priority: | Medium | | | | |
| Frequency of Use: | Sometimes | | | | |
| Business Rules: | * Customers or Photographers can cancel if status is PENDING, ONGOING, EDITING, but if their cancellation is not appropriate, the opposite side can report them. * If one side is being reported, increase their cancelling rate. * After one side cancels, notification will be sent to the other person. | | | | |
| Other Information: | N/A | | | | |
| Assumptions: | N/A | | | | |

Table 22 <Customer> Cancel booking specification

#### 2.2.3 <Photographer> Overview Use Case



Figure 9 <Photographer> Overview

##### <Photographer> Accept Booking

Figure 10 <Photographer> Accept booking

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| UC ID and Name: | **UC – 10 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 time is still valid. | | | | |
| 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”: text  - “Thời gian tác nghiệp dự kiến”: text  - “Thời gian trả ảnh”: text  - “Địa điểm”: map  - “Thông tin khách hàng”:   * “Tên khách hàng”: text   - “Tên gói dịch vụ”:   * “Tên gói”: text * “Mô tả”: text * “Giá dịch vụ: text * “Dịch vụ”: list of texts   [Exception 1] |
| 2 | Photographer sends request to accept booking | | | System outputs: “Chấp nhận cuộc hẹn này?” |
| 3 | Photographer confirms “Có”  [Alternative 1] | | | System outputs: “Nhận cuộc hẹn thành công”  [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 show error message: “Không tìm thấy cuộc hẹn” |
| 2 | The process run into an internal error | | | System show error message: “Đã có lỗi xảy ra” |
| Priority: | Medium | | | | |
| Frequency of Use: | Usually | | | | |
| Business Rules: | * Photographers can’t accept more than one booking with same starting time. * After photographer accept, notification will be sent to customer. | | | | |
| Other Information: | N/A | | | | |
| Assumptions: | N/A | | | | |

Table 27 <Photographer> Accept booking specification

##### <Photographer> Decline Booking

Figure 11 <Photographer> Decline booking

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| UC ID and Name: | **UC – 11 Decline booking** | | | | |
| Created By: | Trương Ngọc Mỹ | | Date Created: | 08/10/2020 | |
| Primary Actor: | Photographer | | Secondary Actors: |  | |
| Trigger: | The photographer sends a request to cancel booking. | | | | |
| Description: | This use case allows photographer to cancel booking from customer | | | | |
| Preconditions: | - The User has been authenticated as a Photographer.  - The booking time is still valid | | | | |
| 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”: text  - “Thời gian tác nghiệp dự kiến”: text  - “Thời gian trả ảnh”: text  - “Địa điểm”: map  - “Thông tin khách hàng”:   * “Tên khách hàng”: text   - “Tên gói dịch vụ”:   * “Tên gói”: text * “Mô tả”: text * “Giá dịch vụ: text * “Dịch vụ”: list of texts   [Exception 1] |
| 2 | Photographer sends request to decline 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 show error message: “Không tìm thấy cuộc hẹn” |
| 2 | The process run into an internal error | | | System show error message: “Đã có lỗi xảy ra” |
| Priority: | Usually | | | | |
| Frequency of Use: | Sometimes | | | | |
| Business Rules: | * Photographer can only decline a booking with status “Pending”. * After photographer decline a booking, notification will be sent to customer. * Decline a booking will increase photographer’s rejecting rate, it could be used by admin to supervise photographers. | | | | |
| Other Information: | N/A | | | | |
| Assumptions: | N/A | | | | |

Table 28 <Photographer> Decline booking specification

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

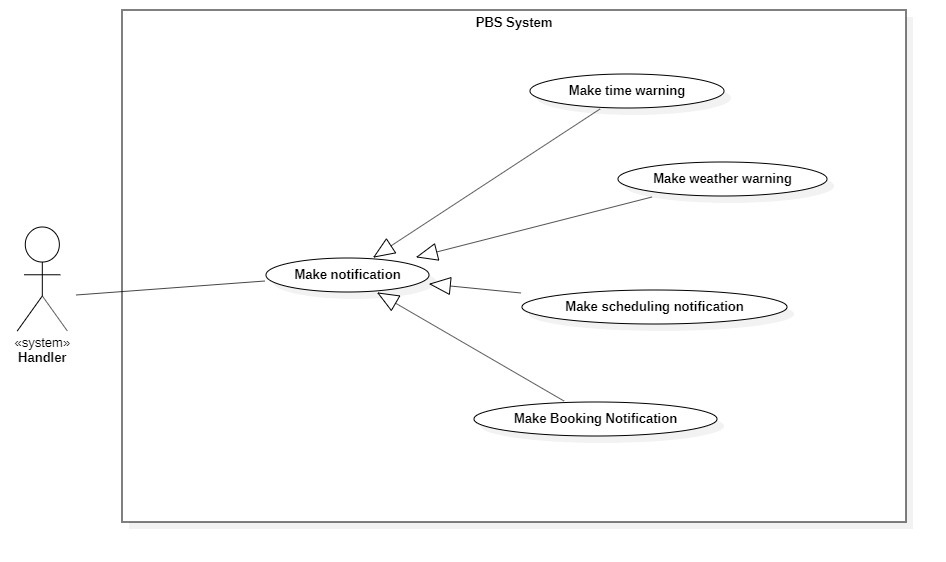


Figure 12 <System Handler> Overview

##### <<system>>Handler Make booking notification

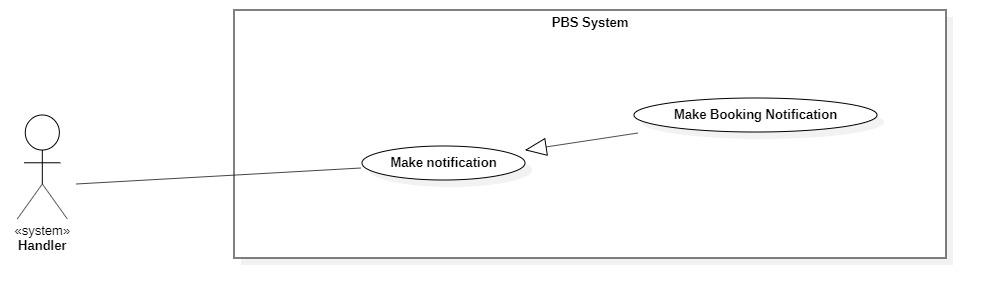
****

Figure 13 <System Handler> Make booking notification

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| UC ID and Name: | **UC – 39 Make booking notification** | | | | |
| Created By: | Bồ Công Đạt | | Date Created: | 08/10/2020 | |
| Primary Actor: | System Handler | | Secondary Actors: |  | |
| Trigger: | System handler sends request to notify photographer or customer whenever booking status is changed | | | | |
| Description: | This use case allows system handler to make booking notification | | | | |
| Preconditions: | N/A | | | | |
| Post-conditions: | **Success:** System make booking 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 booking request | | | System stores booking 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:   * When customer makes a new booking request. Notification will be sent to photographer. * When photographer accepts an incoming request. Notification will be sent to customer. * When photographer reject an incoming request. Notification will be sent to customer. * When customer cancels a booking. Notification will be sent to photographer. * When photographer cancels a booking. Notification will be sent to customer. * When photographer changes status of a booking from “Ongoing” to “Editing”. Notification will be sent to customer. * When photographer changes status of a booking from “Editing” to “Done”. Notification will be sent to customer. | | | | |
| Other Information: | N/A | | | | |
| Assumptions: | N/A | | | | |

Table 52 <System Handler> Make booking notification specification

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

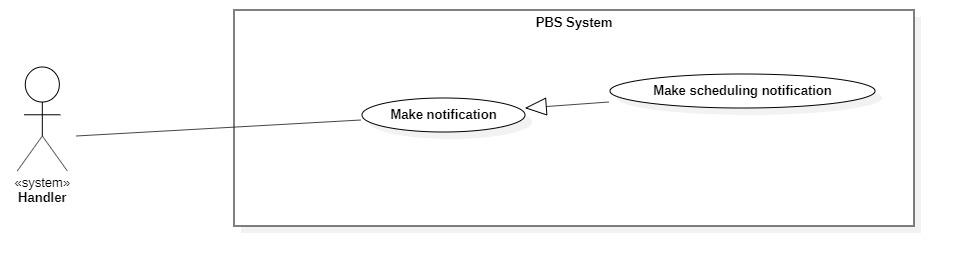
****

Figure 14 <System Handler> Make scheduling notification

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| UC ID and Name: | **UC – 40 Make scheduling notification** | | | | |
| Created By: | Bồ Công Đạt | | Date Created: | 08/10/2020 | |
| 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:   * When booking start time is current date. Notification will be sent to photographer and customer to notice them about on day booking. | | | | |
| Other Information: | N/A | | | | |
| Assumptions: | N/A | | | | |

Table 53 <System Handler> Make scheduling notification specification

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

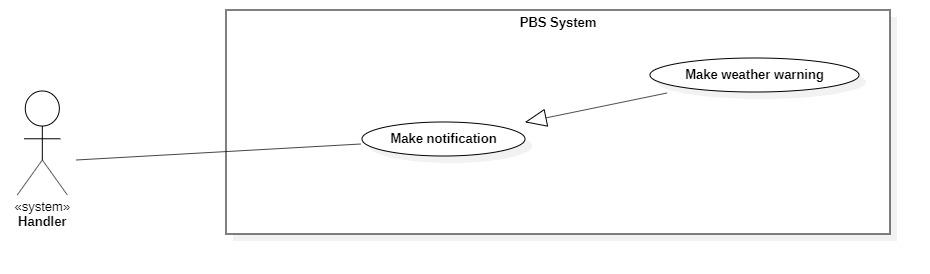
****

Figure 15 <System Handler> Make weather warning

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| UC ID and Name: | **UC – 41 Make weather warning** | | | | |
| Created By: | Bồ Công Đạt | | Date Created: | 08/10/2020 | |
| 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: |  | | | | |
| 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. | | | | |
| Other Information: | N/A | | | | |
| Assumptions: | N/A | | | | |

Table 54 <System Handler> Make weather warning specification

##### <<system>>Handler Make time warning

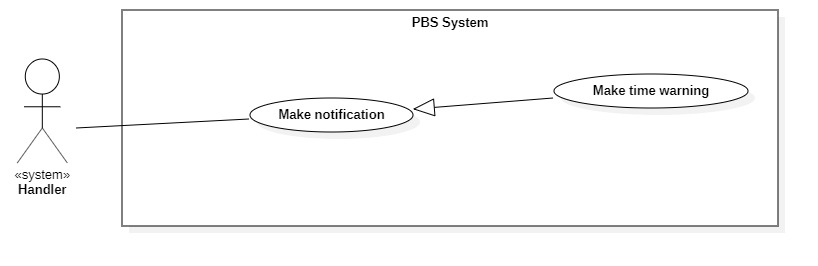
****

Figure 16 <System Handler> Make time warning

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| UC ID and Name: | **UC – 42 Make time warning** | | | | |
| Created By: | Bồ Công Đạt | | Date Created: | 08/10/2020 | |
| Primary Actor: | System Handler | | Secondary Actors: |  | |
| Trigger: | System handler sends requests to notify customers of bad time selection. | | | | |
| Description: | This use case allows the system handler to make time warnings. | | | | |
| Preconditions: | N/A | | | | |
| Post-conditions: | **Success:** System makes time warning and sends to customer.  **Fail:** System outputs error and sends to customer | | | | |
| Normal Flow: | **Step** | **Actor Action** | | | **System Response** |
| 1 | System handler retrieves bad time selection request | | | System stores bad time information and return notification to system handler |
| 2 | System handler retrieves notification information and sends to customer  [Exception 1] | | |  |
| Alternative Flows: |  | | | | |
| Exceptions: | **No** | **Cause** | | | **System Response** |
| 1 | The process runs into an internal error | | | System outputs error and sends to 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 the selected time is too tight for photographer. * For example, photographer have a booking between 9am – 12pm 10/10/2020, if customer book at 13pm 10/10/2020, the system will suggest the customer that photographer may not be able to accept this. | | | | |
| Other Information: | N/A | | | | |
| Assumptions: | N/A | | | | |

Table 55<System Handler> Make time warning specification

## 3. Functional Requirements

### 3.1 System Functional Overview

#### a. Screen Flow

1. **Customer’s screen flow**

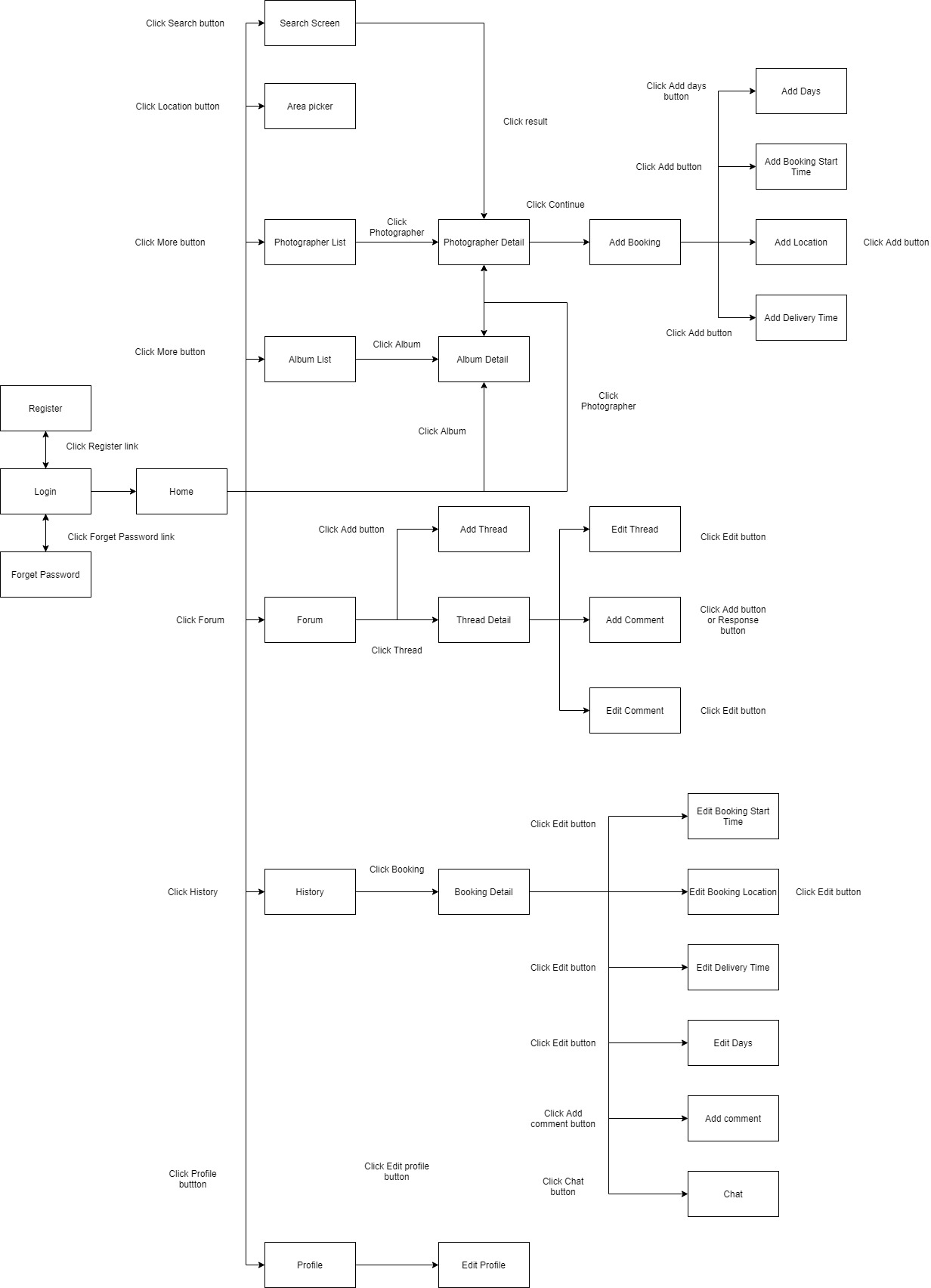


Figure 17 Customer's screen flow

1. **Photographer’s screen flow**

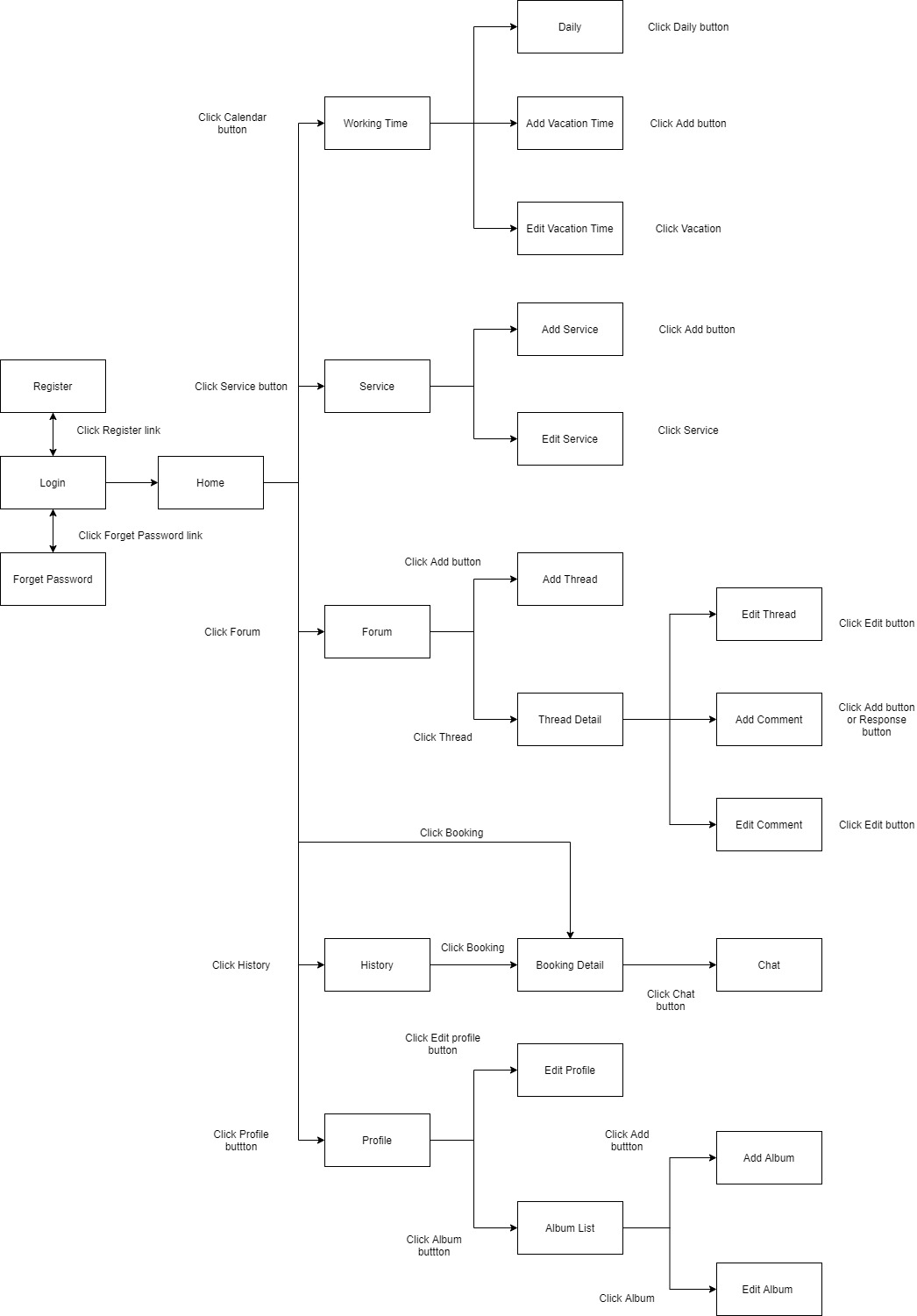


Figure 18 Photographer's screen flow

#### e. Entity Relationship Diagram

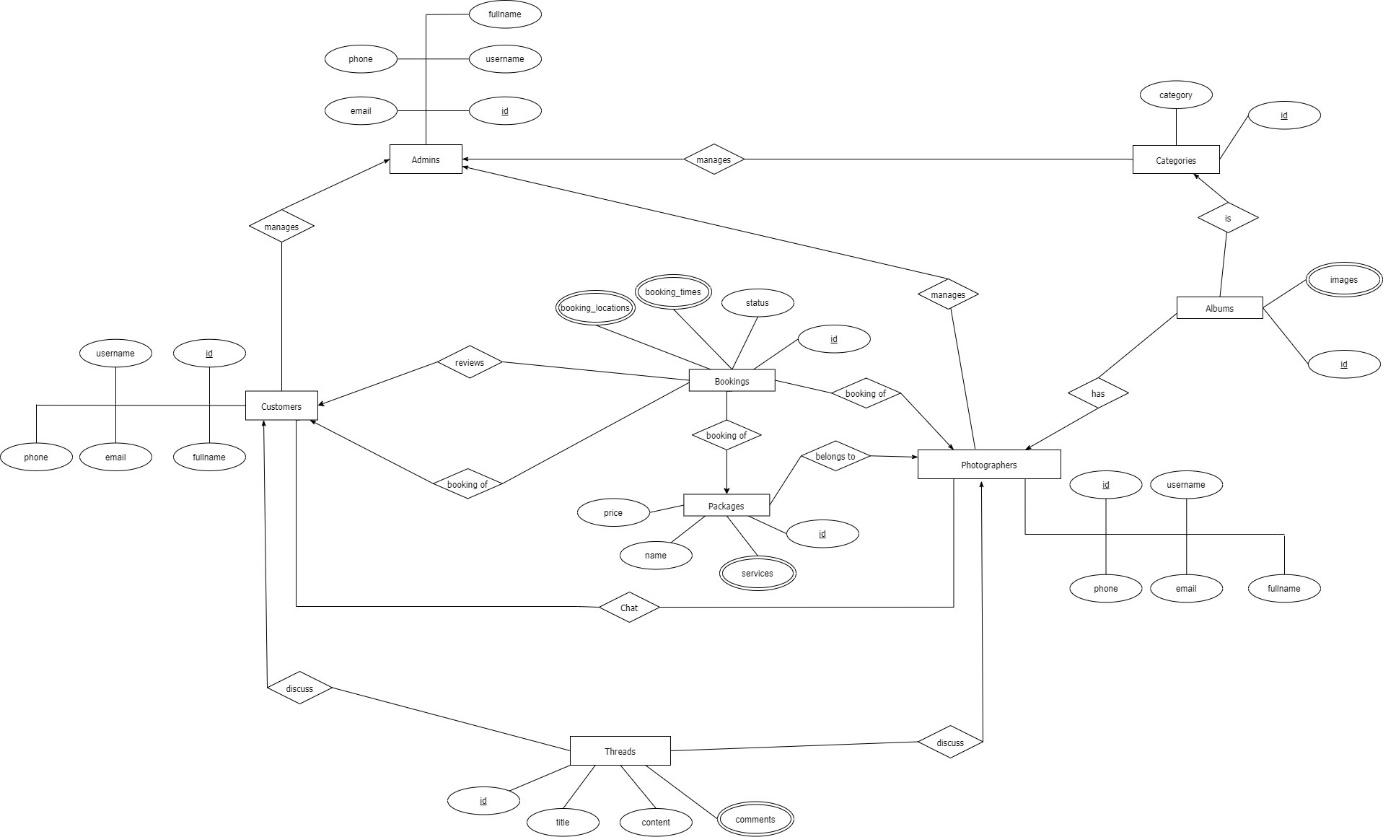


Figure 19 ER Diagram

**Entities List**

|  |  |  |
| --- | --- | --- |
| **#** | **Entity** | **Description** |
| 1 | Customer | Customer is an entity representing the person who purchases bookings from photographers: id, username, fullname, phone, email. |
| 2 | Admin | Admin is an entity represents the person who works for pbs system and manages activities of the system: id, username, fullname, phone, email |
| 3 | Booking | Booking is an entity represents the photography deal between photographer and customer: id, status, booking\_times, booking\_locations |
| 4 | Photographer | Photographer is an entity that represents the person who provides the photography services and takes customers’ deals. |
| 5 | Category | Category is an entity represents the type of albums (is it wedding or look book or anything else): id, category |
| 6 | Album | Album is an entity represents the images which photographer wants to show the customers to attract them: id, images |
| 7 | Package | Package is an entity represents the photography services which photographer provided: id, name, price, services |
| 8 | Thread | Thread is an entity represents a post of photographer or customer or admin on forum: id, title, content, comments |

Table 59 Entities List

# IV. Software Design Document

## 2. System Architecture Design

### 2.1 Overall Architecture

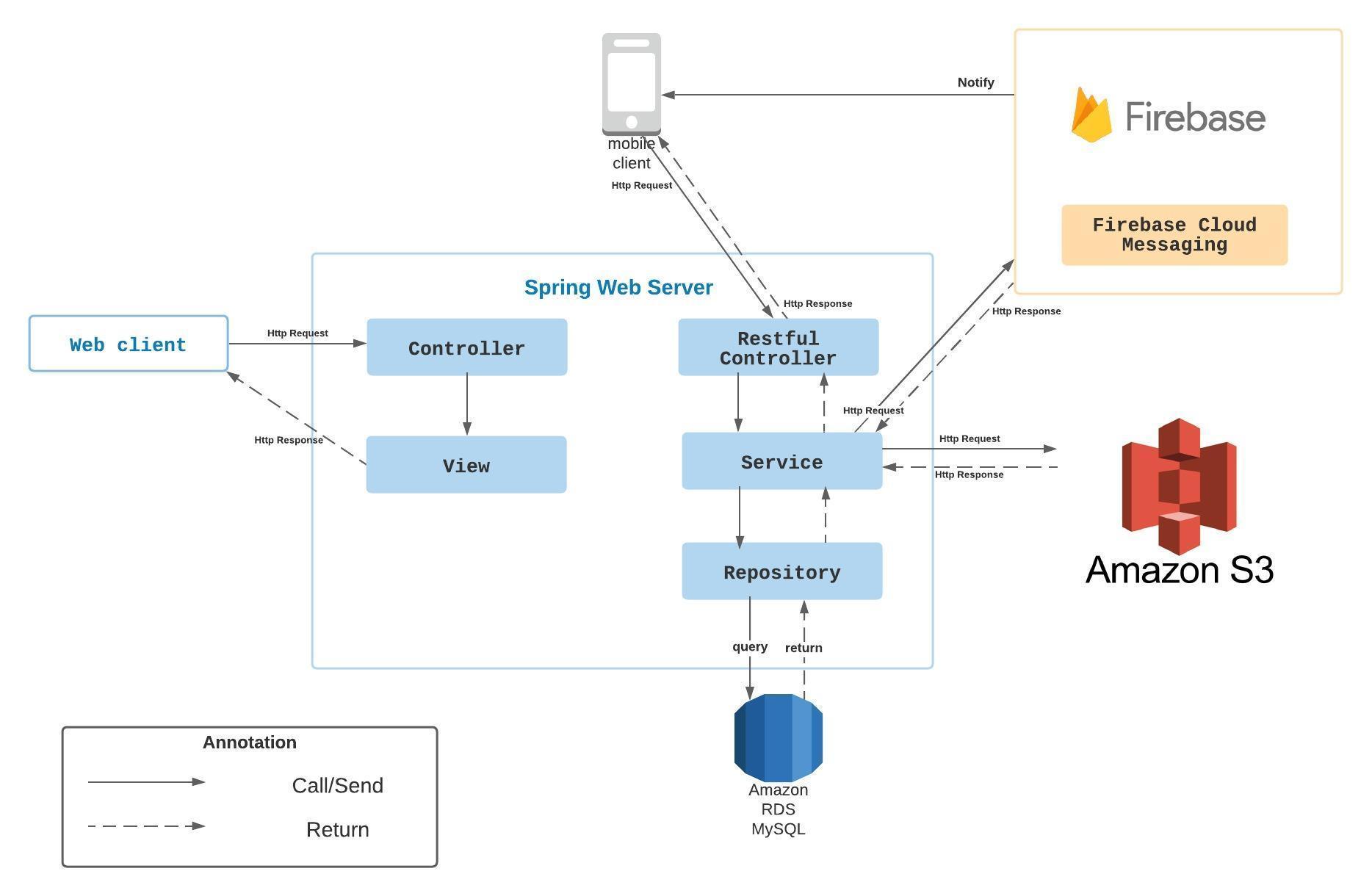


Figure 20 Overall Architecture

### 2.2 System Architecture

**2.2.2 Mobile app architecture**

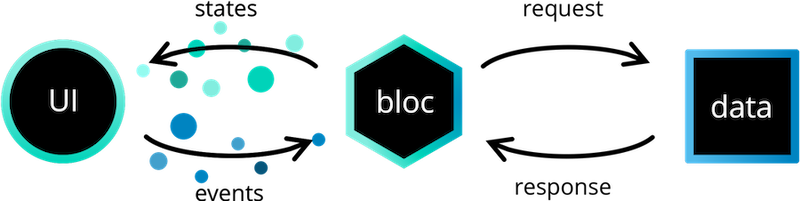


Figure 21 Mobile App Architecture

*Reference:* [*https://bloclibrary.dev/#/architecture*](https://bloclibrary.dev/#/architecture)

### 2.3 Package Diagram

**2.3.1 Webapi**

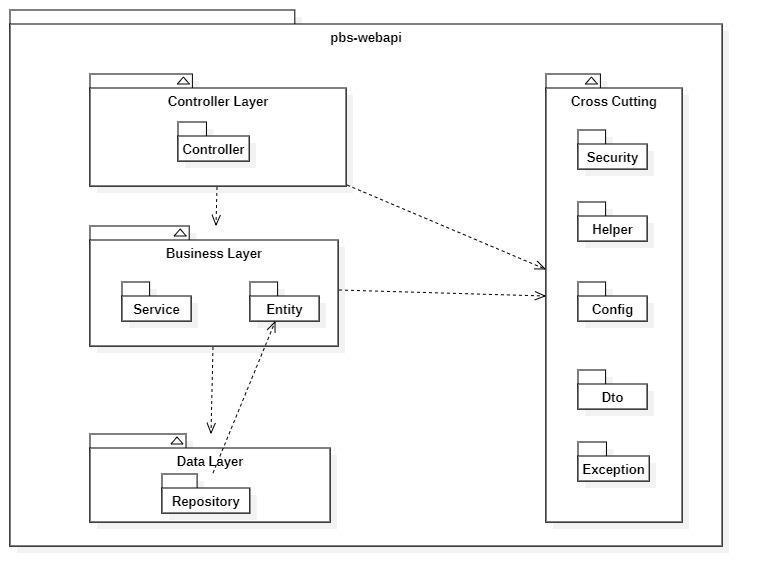
****

Figure 22 Webapi Package Diagram

|  |  |  |
| --- | --- | --- |
| **No** | **Package** | **Description** |
| 01 | Controller | Package contains classes responsible for handling Http request and return responses with values |
| 02 | Service | Package contains classes responsible for handling business logic, validation and communicate with repository |
| 03 | Entity | Package contains classes represents the business logic entities, also responsible for code first approach with hibernate |
| 04 | Repository | Package contains classes responsible for executing query and communicate with database |
| 05 | Security | Package contains classes responsible for performing security includes jwt handling, authentication and authorization |
| 06 | Helper | Package contains classes responsible for utilizing like map helper, datetime helper, String helper |
| 07 | Config | Package contains classes responsible for configuring spring boot app, from security, application properties to firebase notification |
| 08 | Exception | Package contains classes responsible for handling exception, including status code mapping or error messages returning |
| 09 | Dto | Package contains classes responsible for sharping the data to return to clients |

Table 61 Package Diagram Description

**2.3.2 Mobile App**

Both mobile apps share a same project structuring, so we use one diagram to represents both of them here:

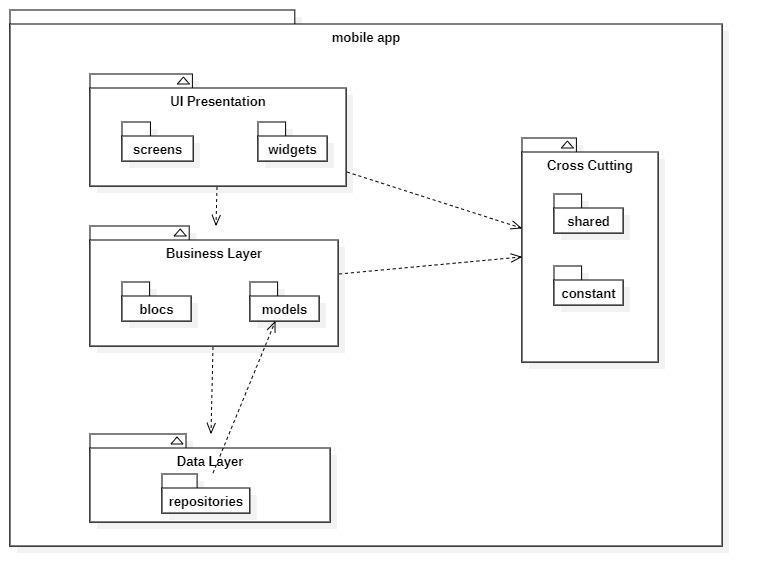


Figure 23 Package Diagram – Mobile app

|  |  |  |
| --- | --- | --- |
| **No** | **Package** | **Description** |
| 01 | screens | Package contains classes for UI representation |
| 02 | widgets | Package contains classes to form the widgets used by screens |
| 03 | blocs | Package contains classes to implement the bloc architecture including bloc’s state, bloc’s event |
| 04 | models | Package contains classes to form an object extracted from api |
| 05 | repositories | Package contains classes responsible for communicating with api server |
| 06 | shared | Package contains classes responsible for utilizing and environment variables |
| 07 | constant | Package contains classes represents the constants used by application |

Table 62 Package Diagram Description

## 3. System Detailed Design

### 3.1 Overall

#### a. Class Diagram

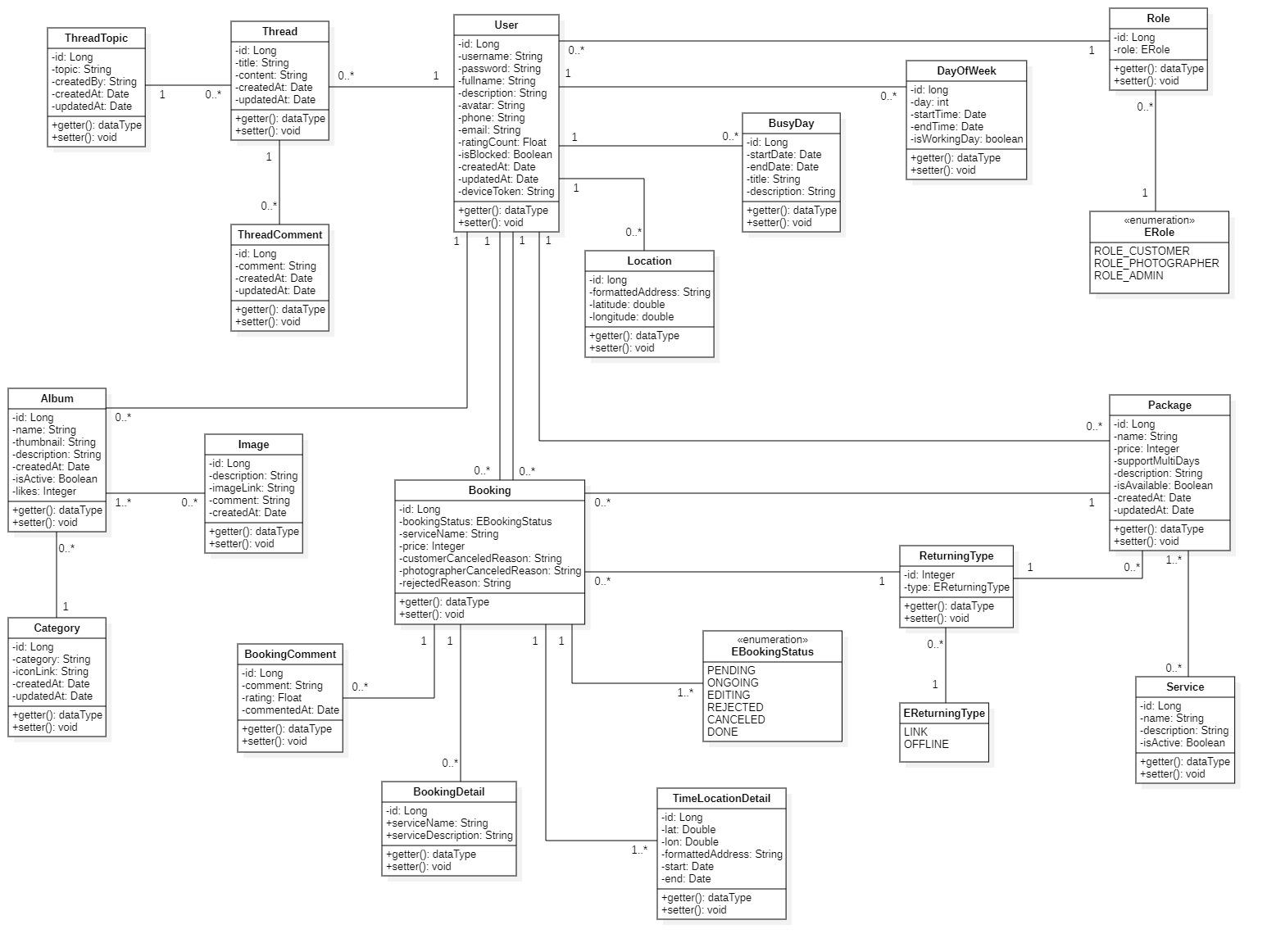


Figure 24 Class Diagram

#### b. Sequence Diagram(s)

1. Add Category

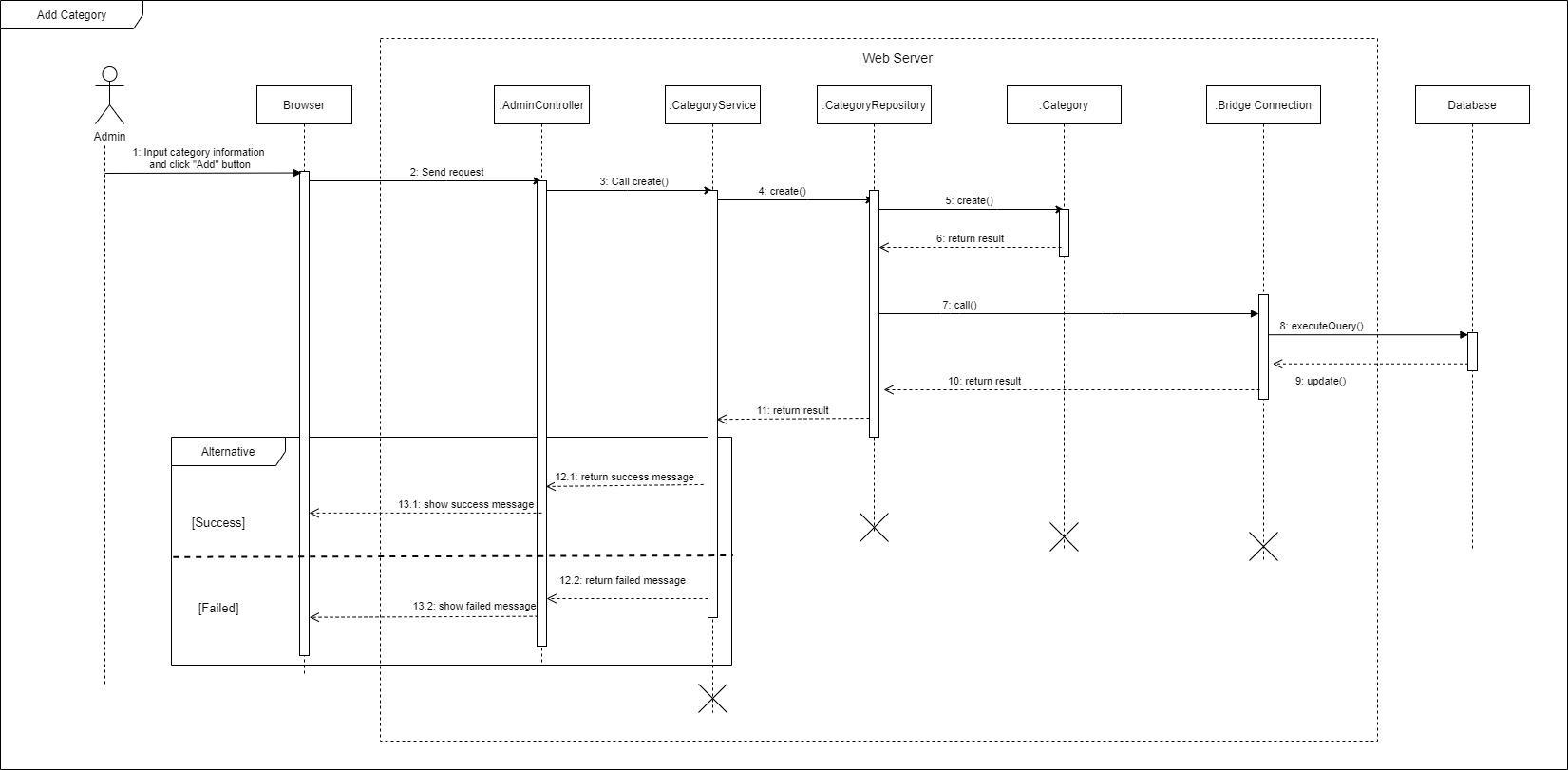


Figure 25 Sequence Diagram - Add Category

1. Edit Category

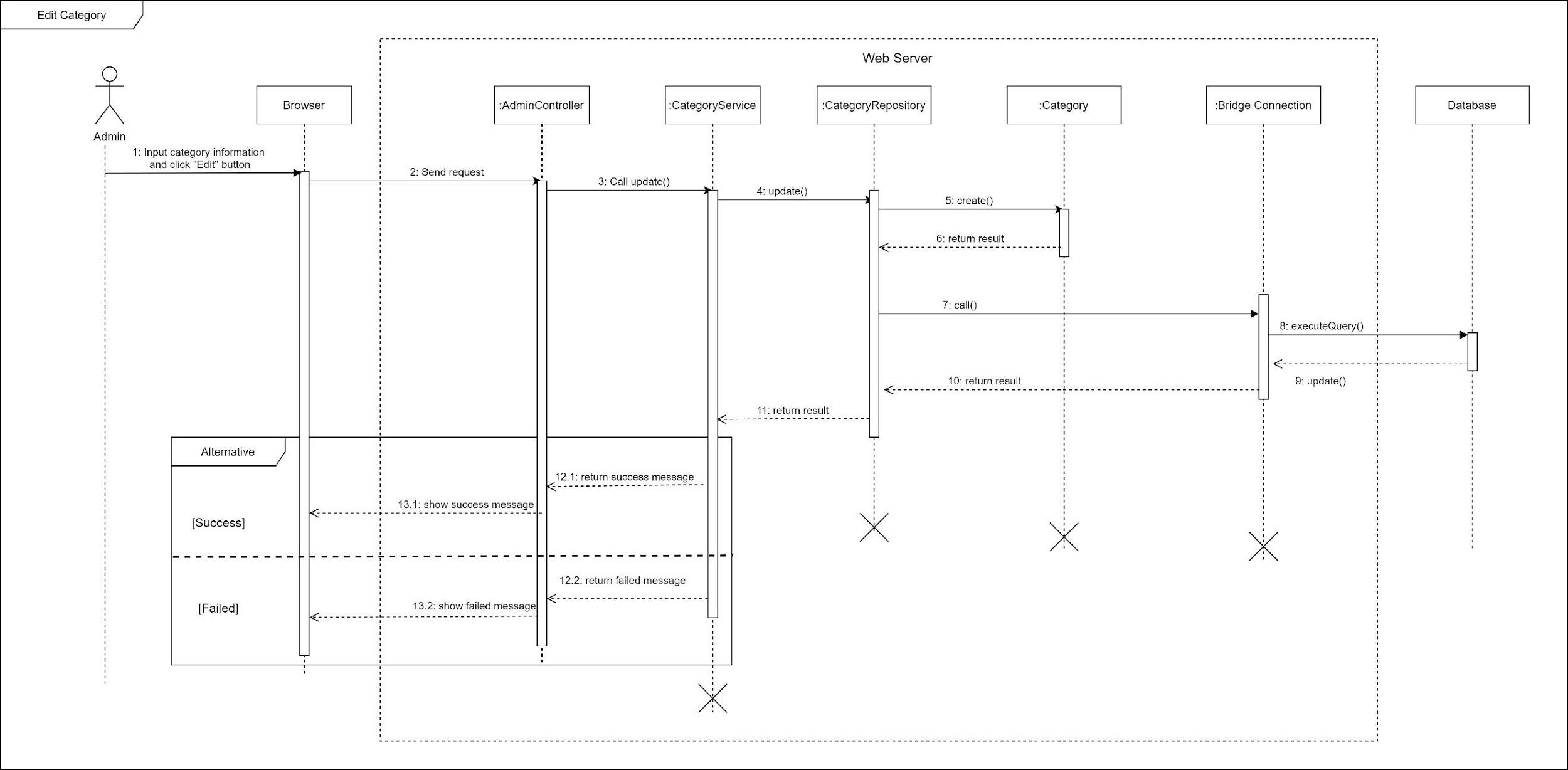


Figure 26 Sequence Diagram - Edit Category

1. Remove category

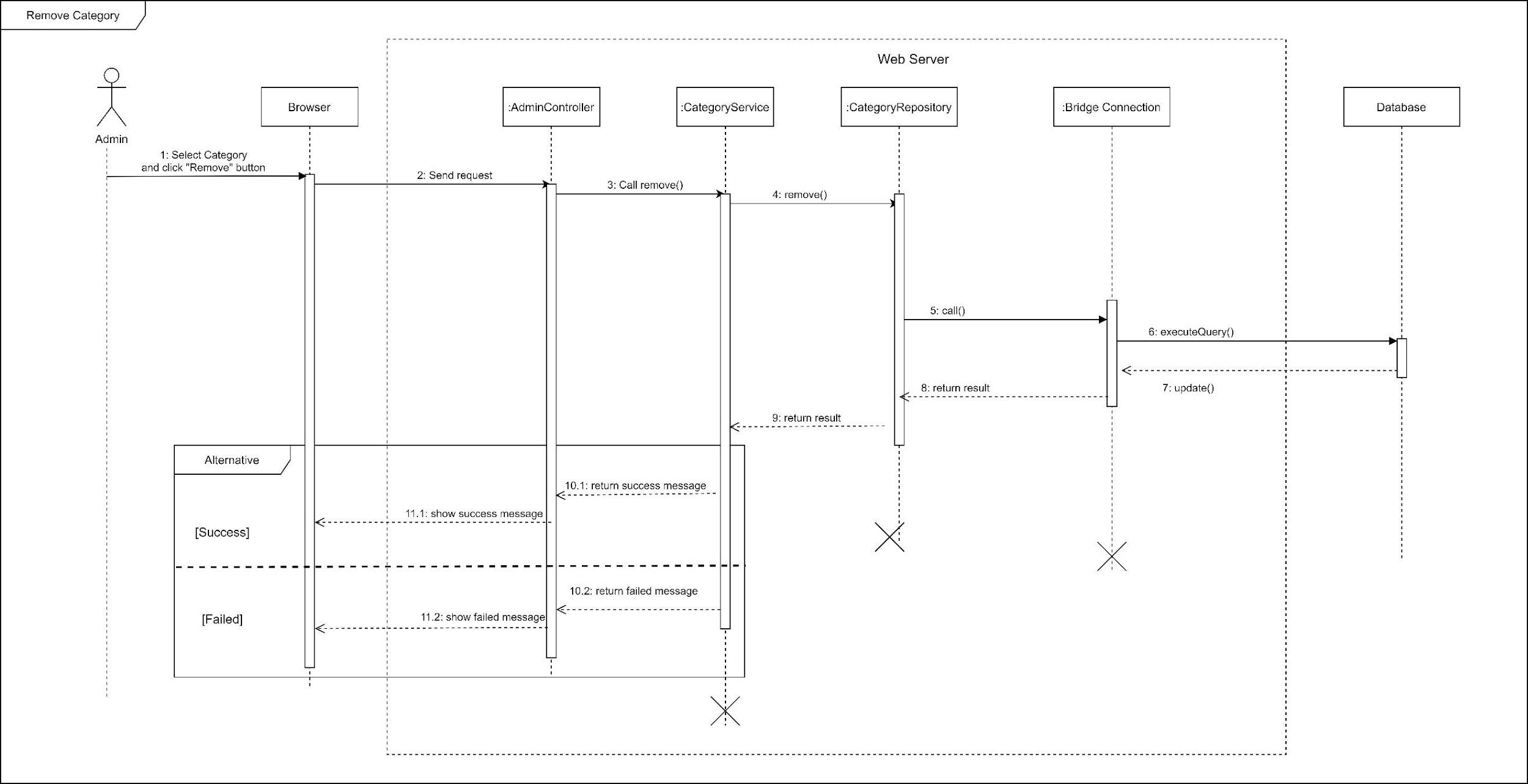


Figure 27 Sequence Diagram - Remove Category

1. Add returning type

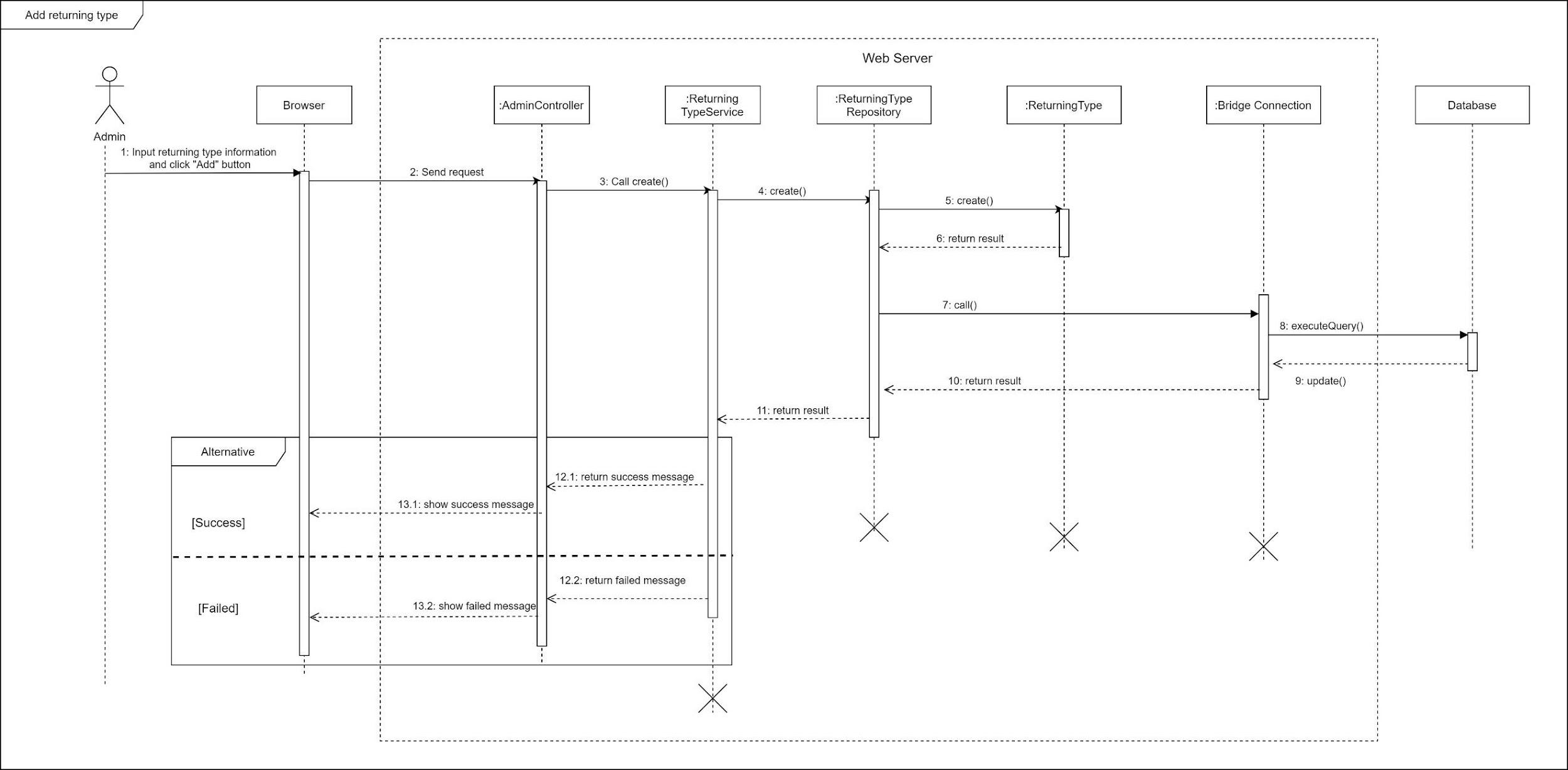


Figure 28 Sequence Diagram - Add returning type

1. Edit returning type

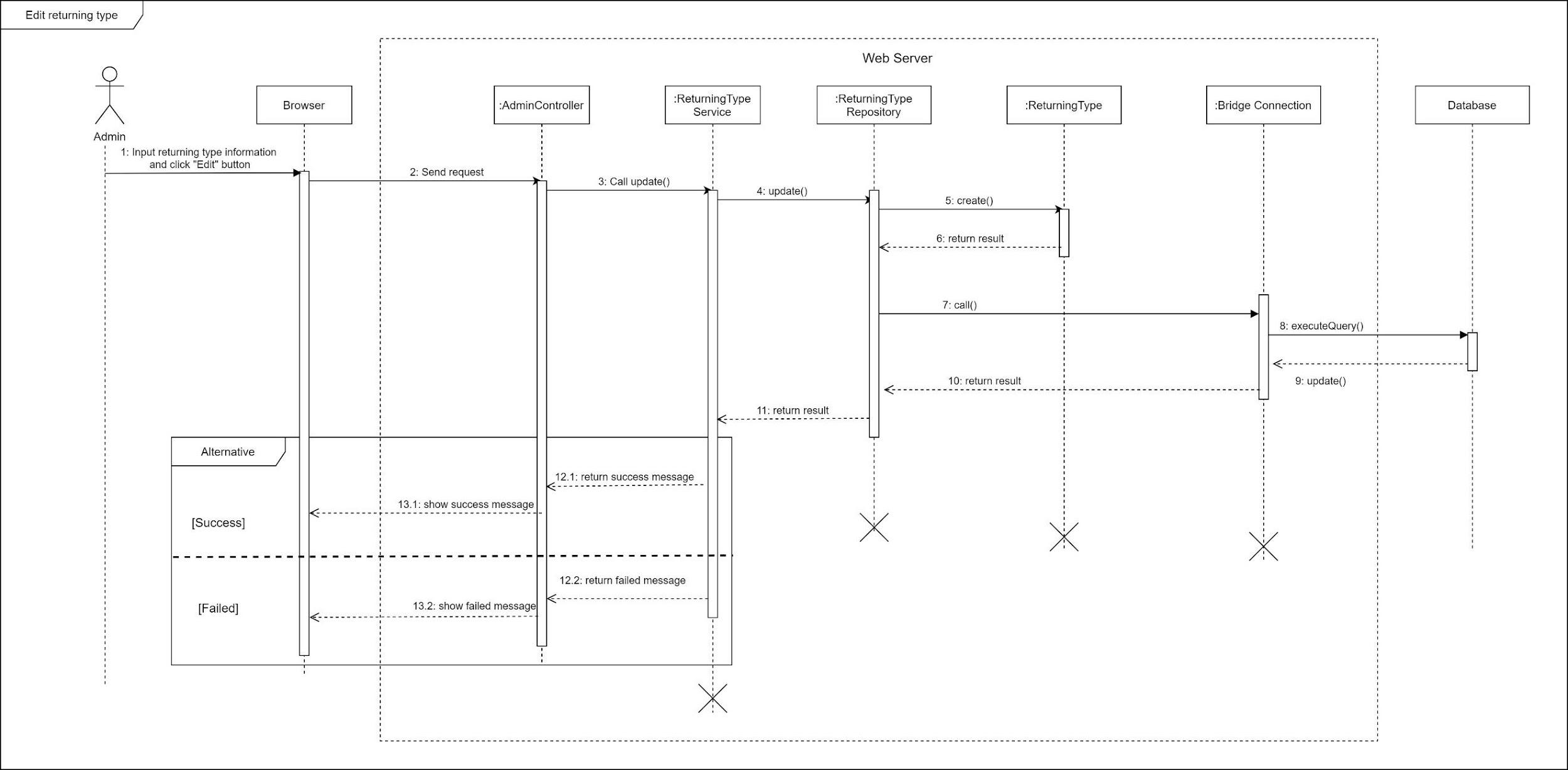


Figure 29 Sequence Diagram - Edit returning type

1. Remove returning type

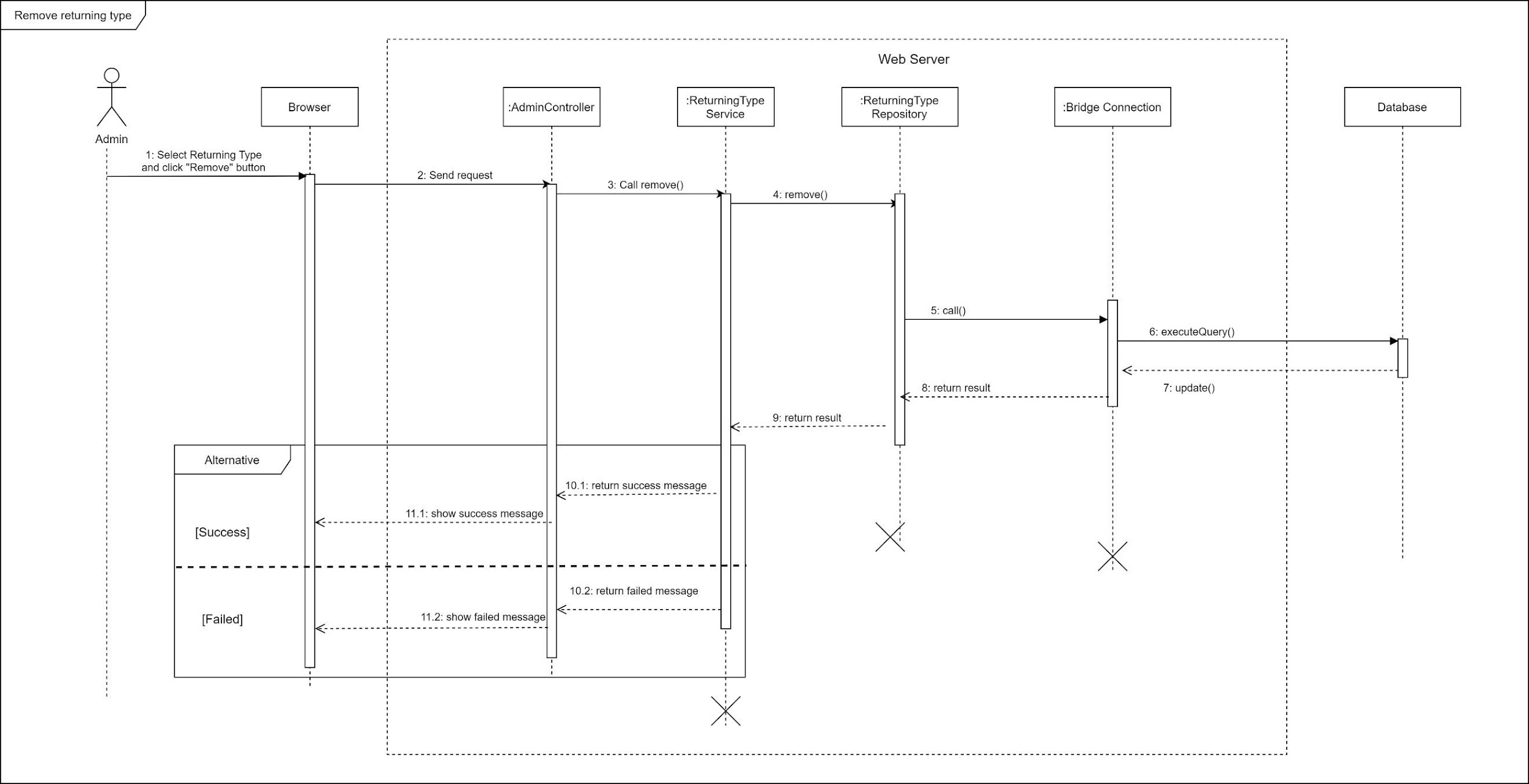


Figure 30 Sequence Diagram - Remove returning type

1. Block user

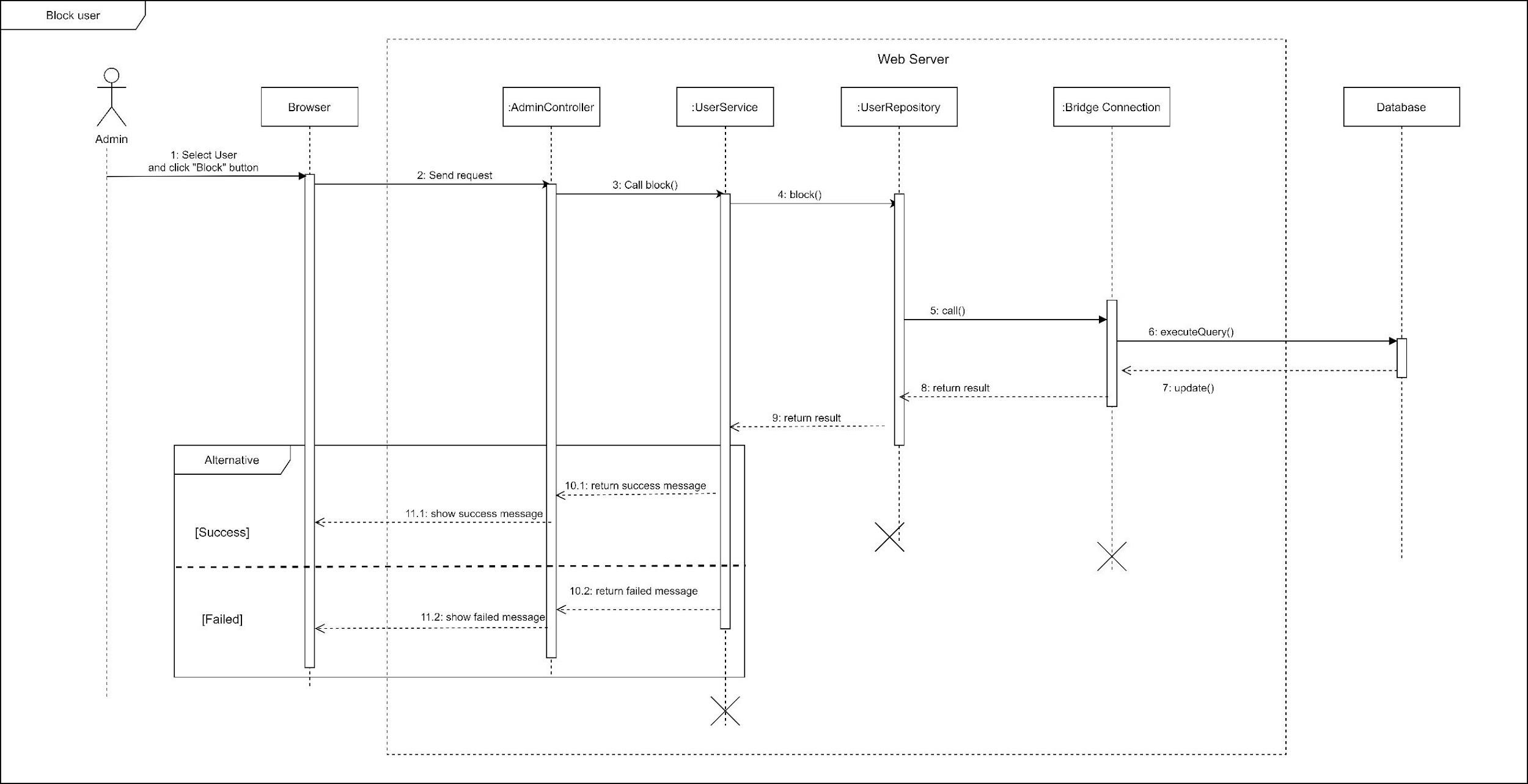


Figure 31 Sequence Diagram - Block User

1. Unblock user

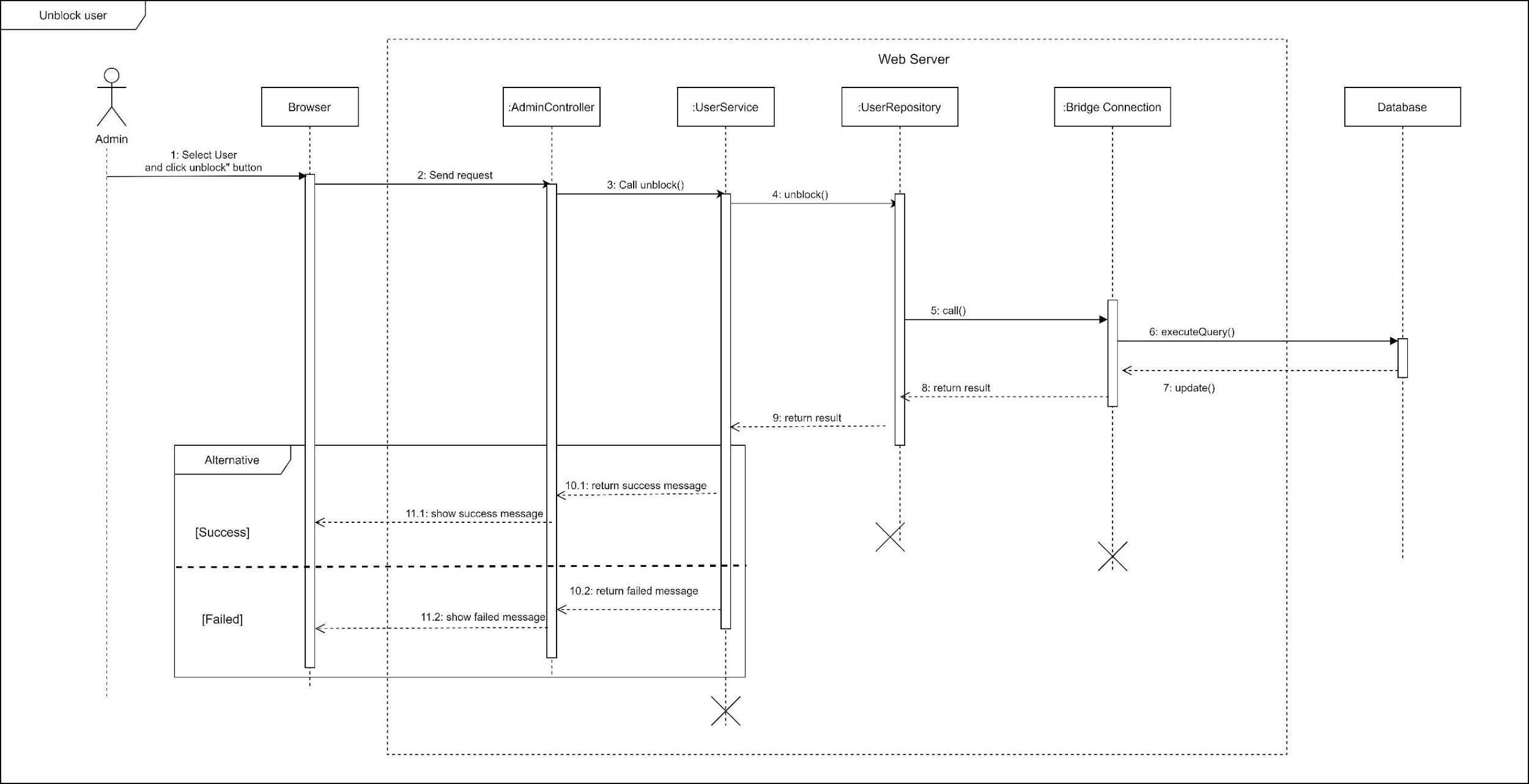


Figure 32 Sequence Diagram - Unblock User

1. Change variable

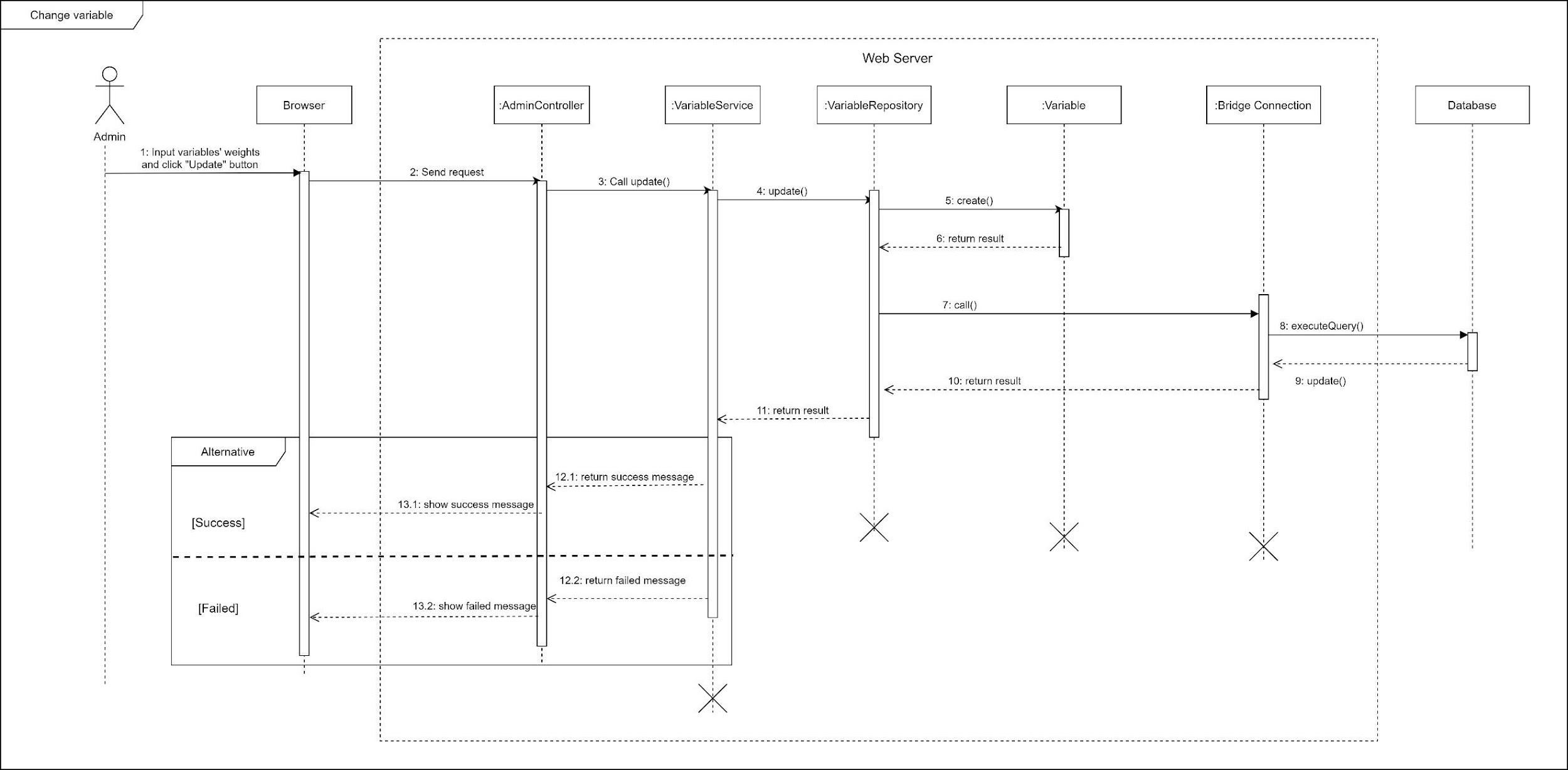
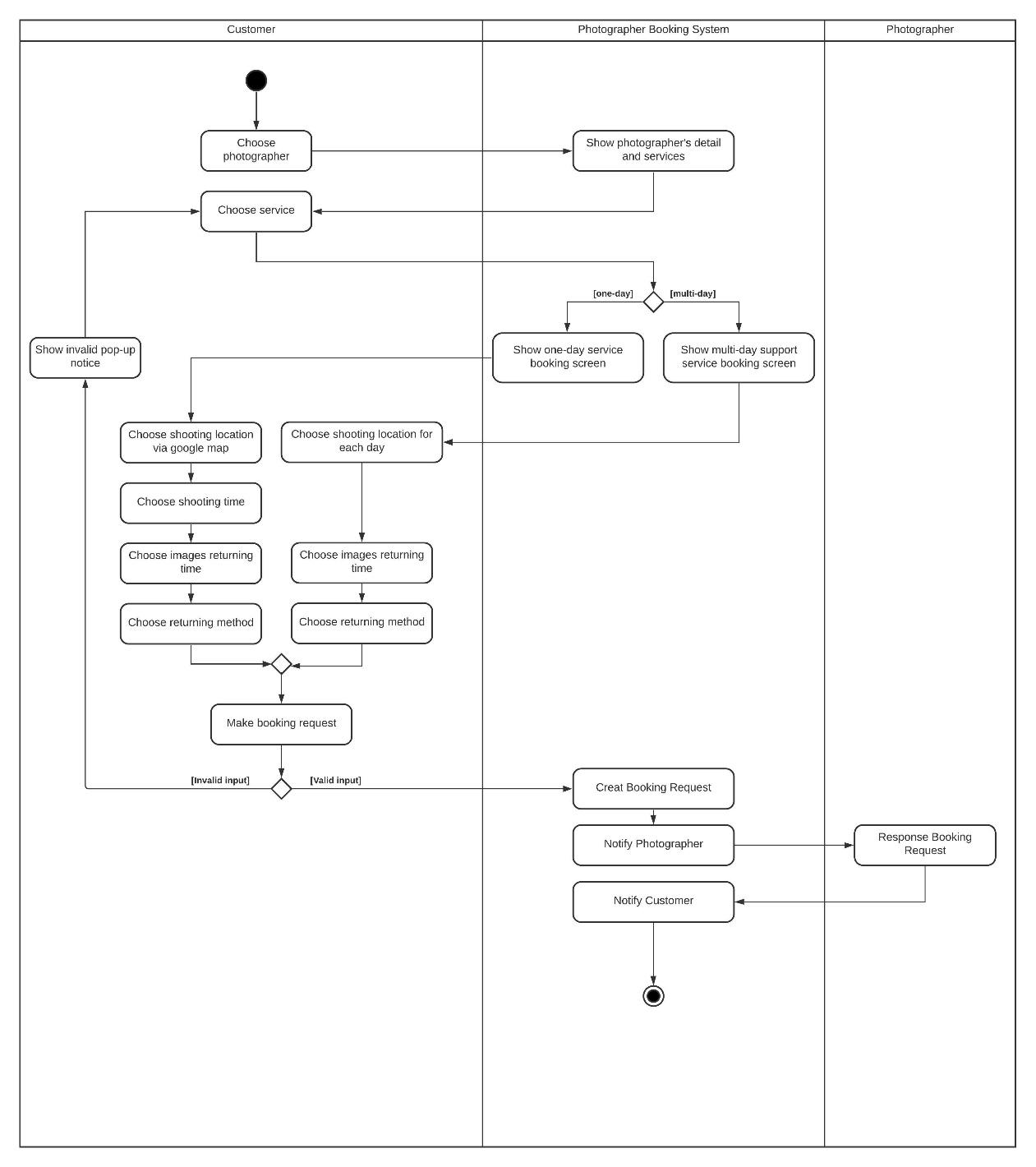


Figure 33 Sequence Diagram - Change Variables

#### c. Activity Diagram(s)

1. **<Customer>** **Make request**

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



*Figure SEQ Figure \\* ARABIC 8 Activity Diagram - Make Booking Request*

Figure 34 Activity Diagram - Make Request

1. **<Customer> Edit request**

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

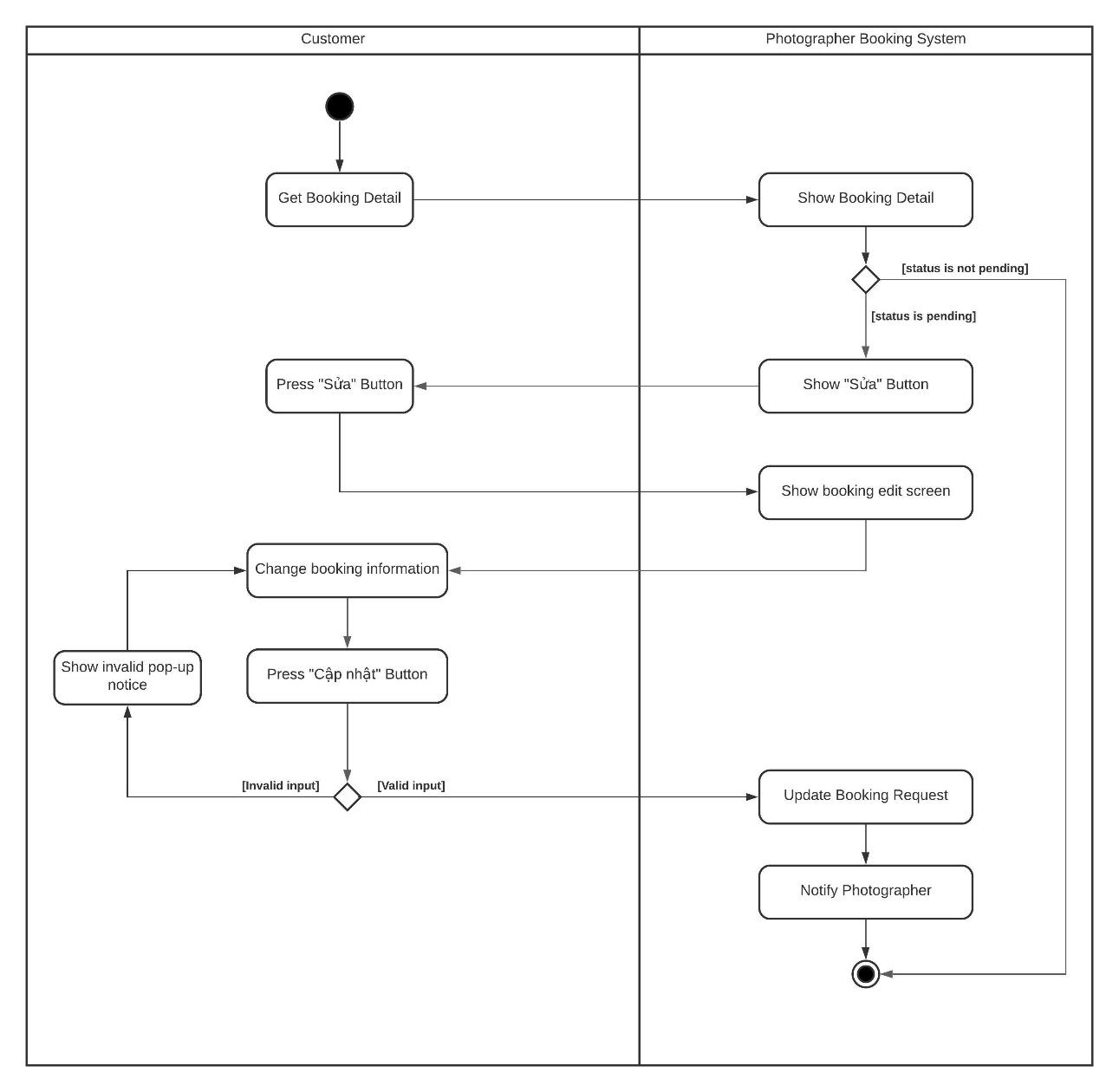


Figure 35 Activity Diagram – Edit Request

*Figure SEQ Figure \\* ARABIC 13 Activity Diagram - Edit Request*

1. **<Customer> Cancel booking**

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

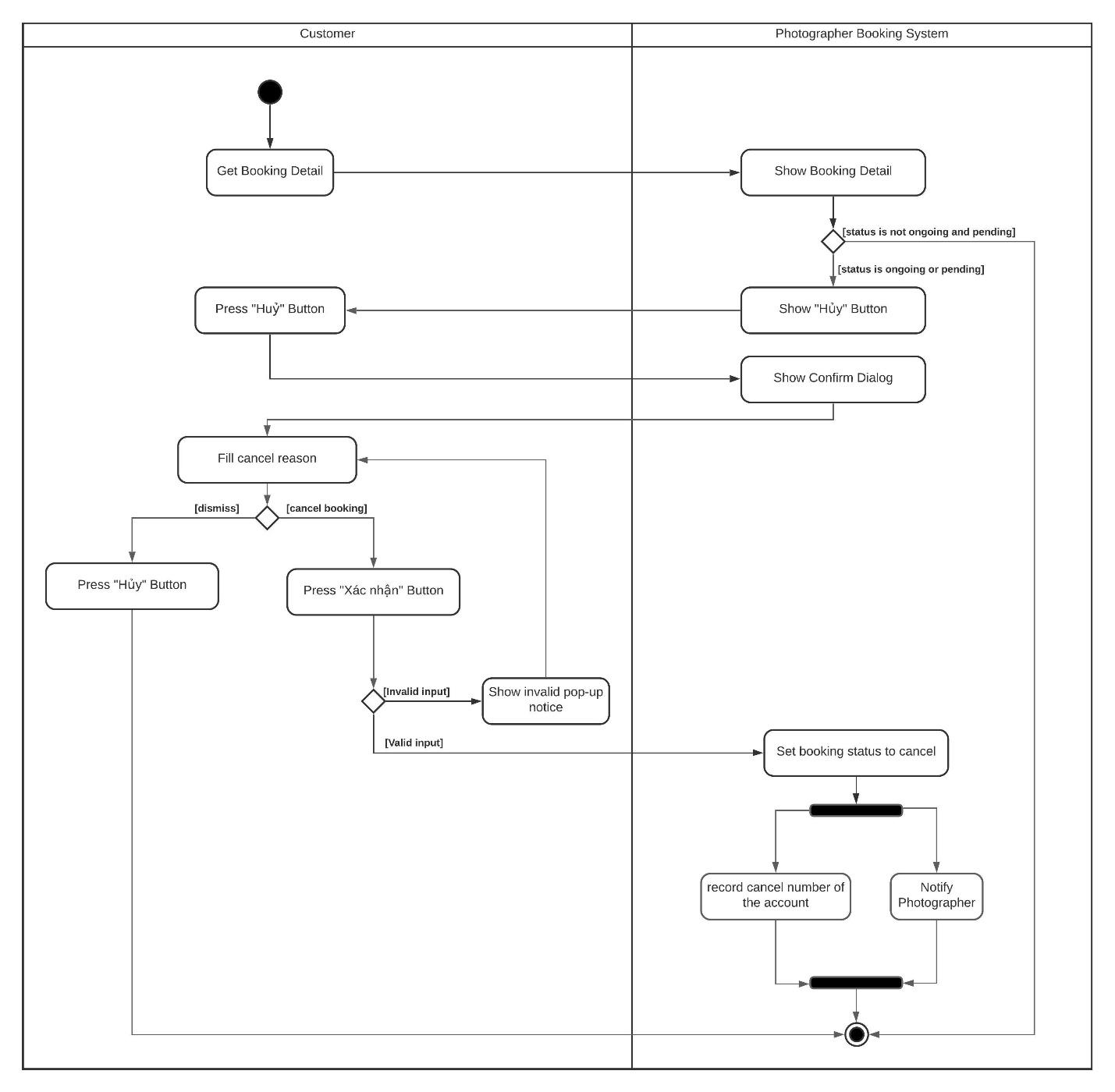


Figure 36 Activity Diagram - Cancel booking

1. **<Customer> Rating**

Summary: This diagram shows the process when a customer rates a photographer after booking finishes.

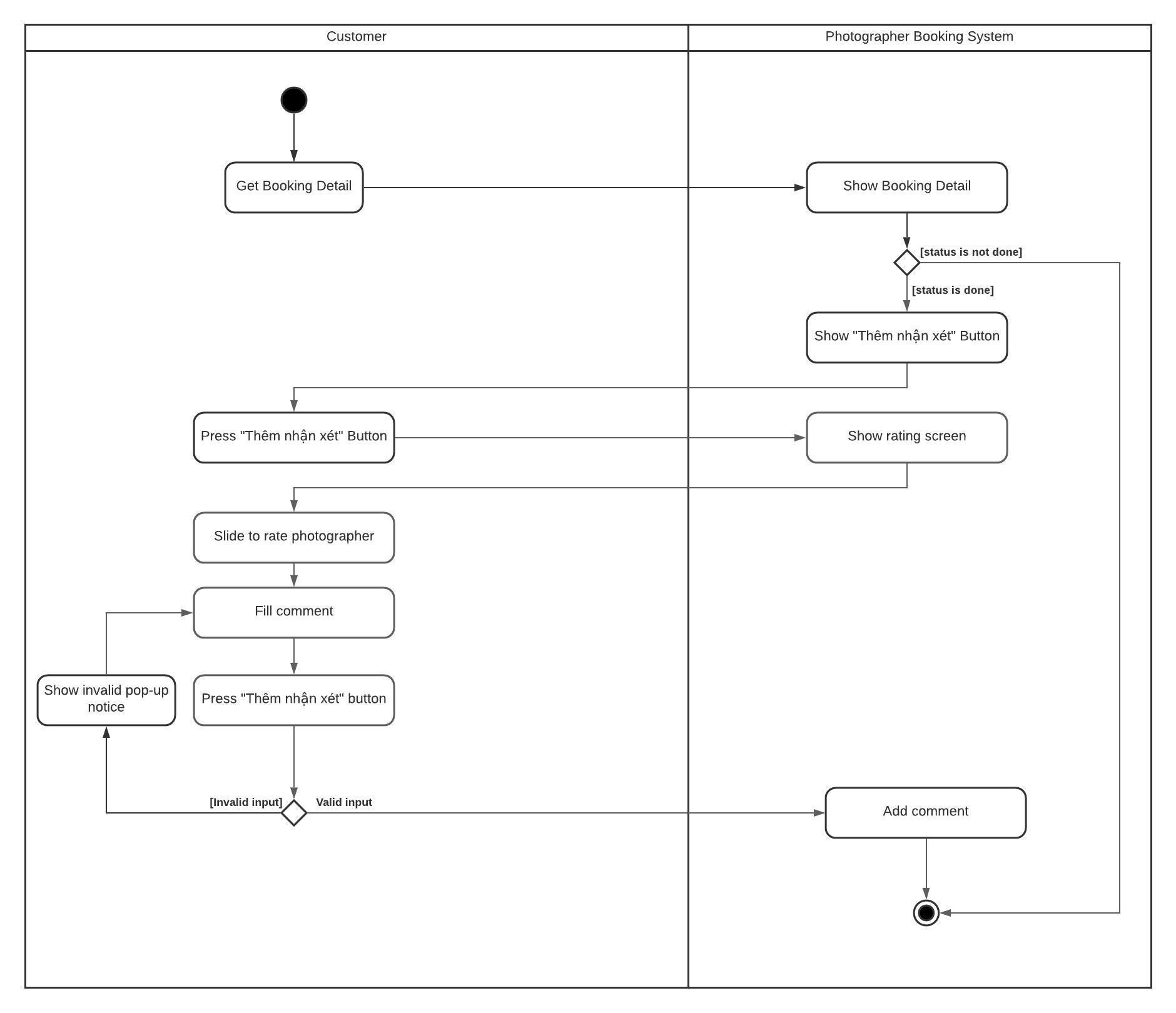


Figure 37 Activity Diagram - Rating

*Figure SEQ Figure \\* ARABIC 15 Activity Diagram - Rating*

*Figure SEQ Figure \\* ARABIC 15 Activity Diagram - Rating*

1. **<Photographer> Accept booking**

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

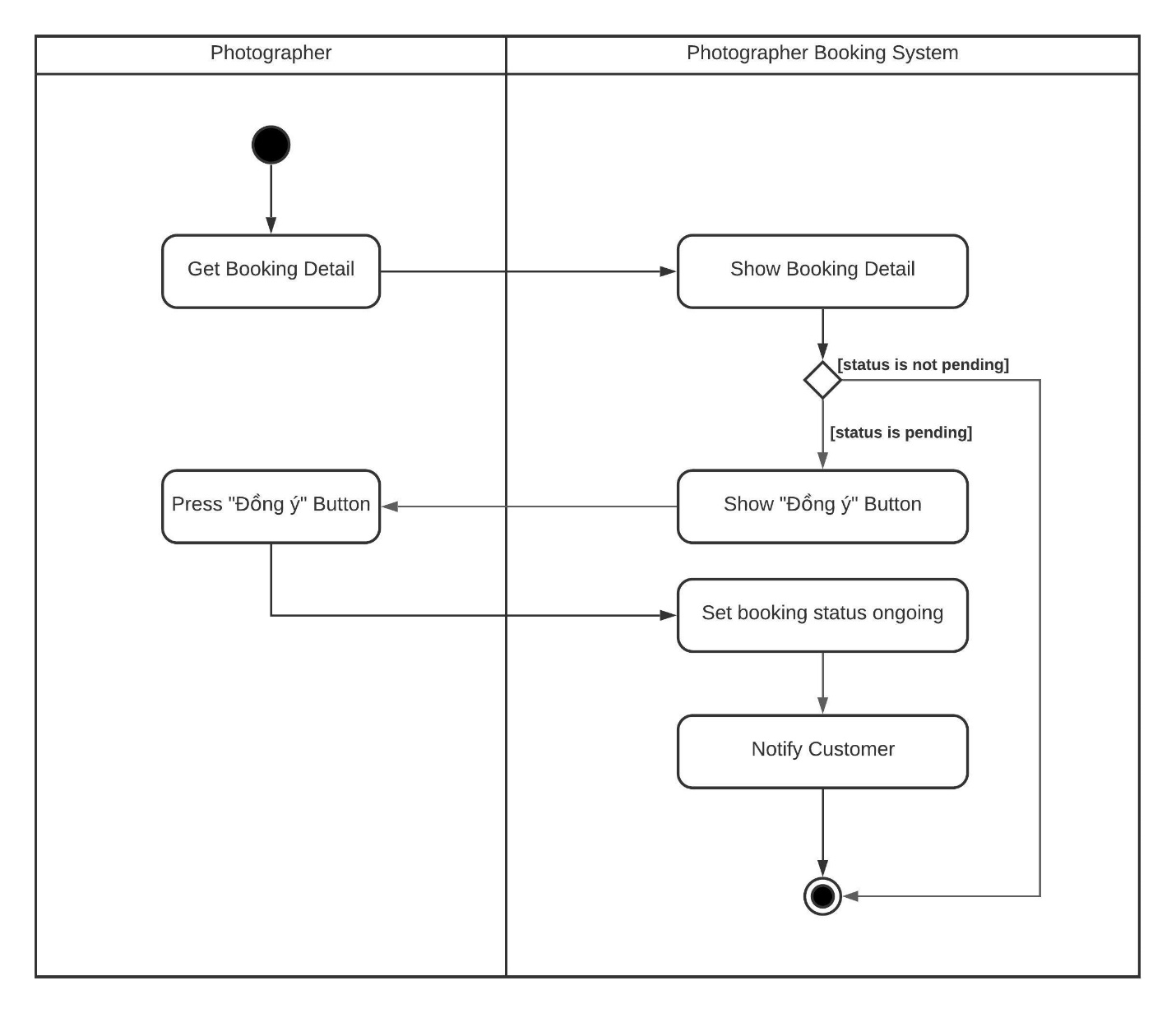


Figure 38 Activity Diagram - Accept booking

*Figure SEQ Figure \\* ARABIC 16 Activity Diagram - Accept Booking*

*Figure SEQ Figure \\* ARABIC 16 Activity Diagram - Accept Booking*

1. **<Photographer> Reject booking**

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

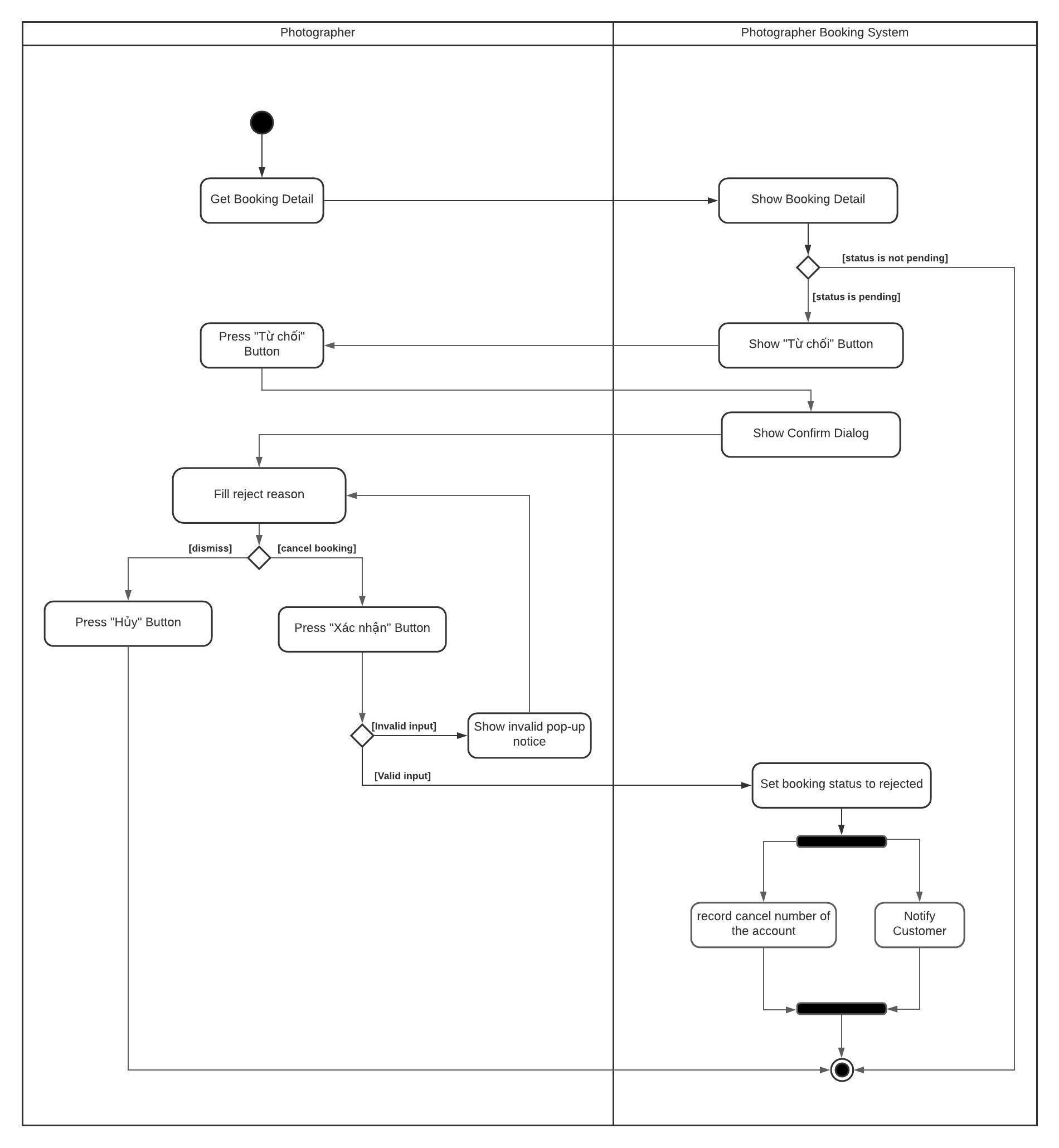


Figure 39 Activity Diagram - Reject booking

*Figure SEQ Figure \\* ARABIC 17 Activity Diagram - Reject Booking*

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

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

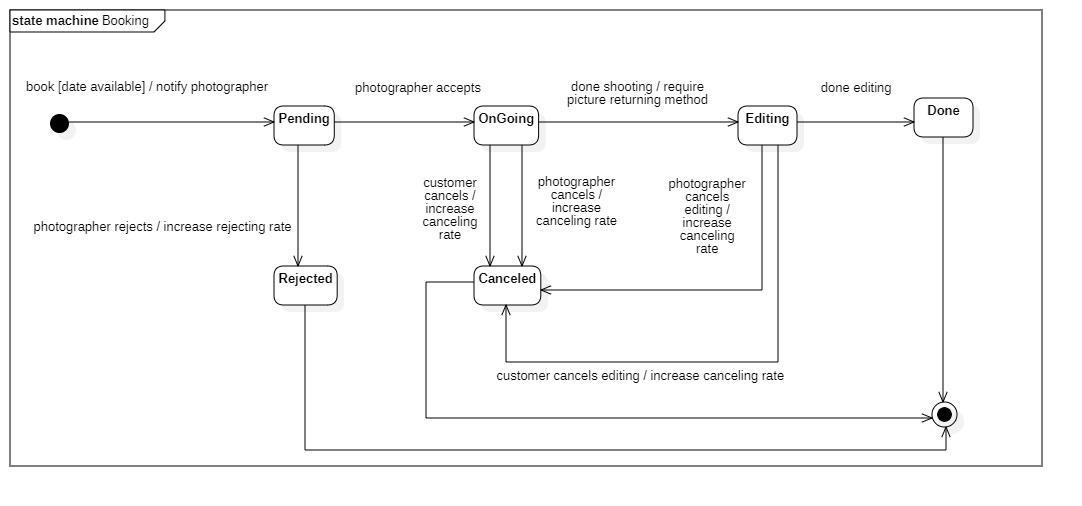


Figure 40 State Machine Diagram - Booking

## 5. Data & Database Design

### 5.1 Database Design

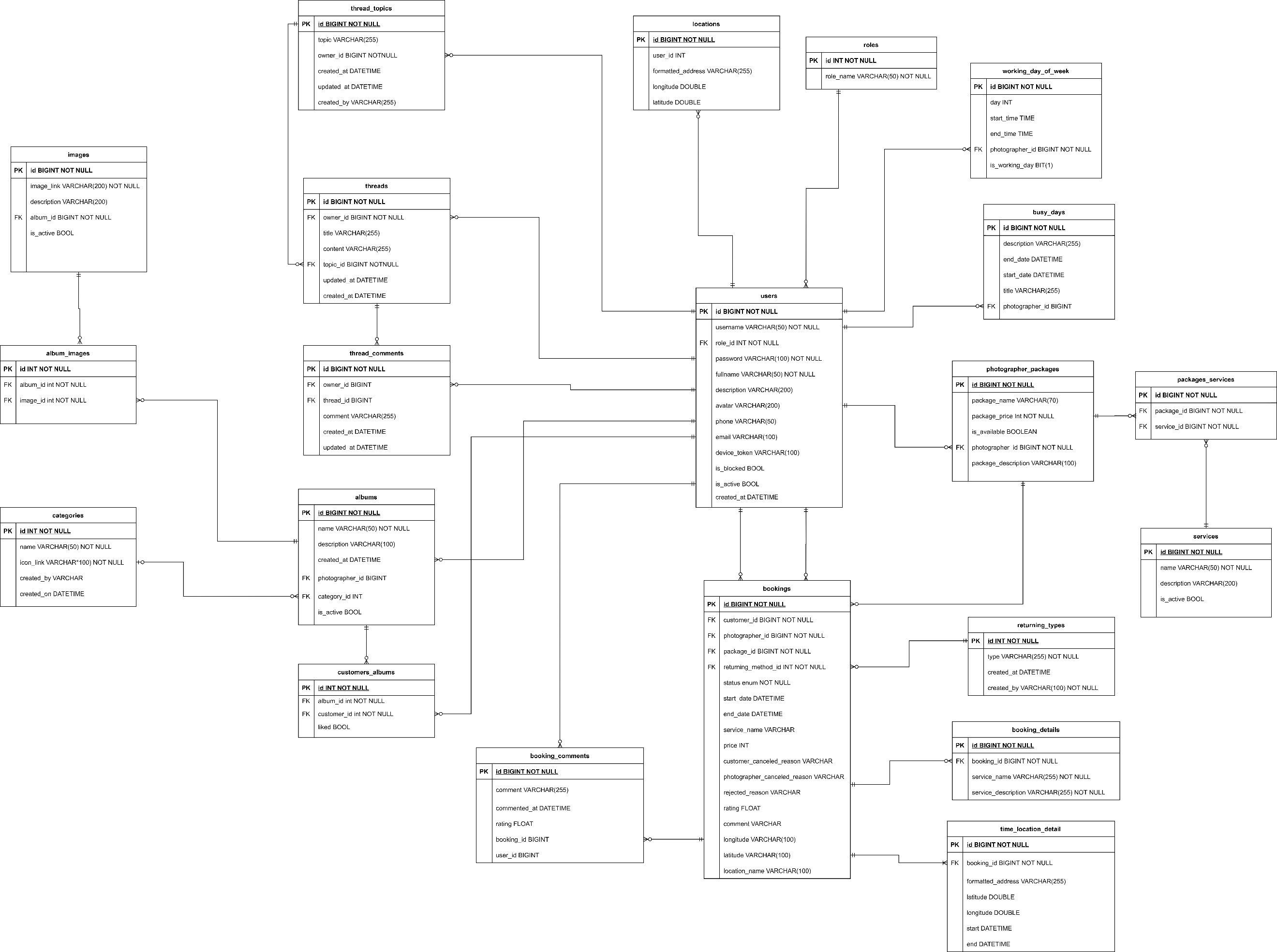


Figure Database Design

## 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.**

#### 6.1.4 Complexity

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

#### 6.1.5 Flowchart

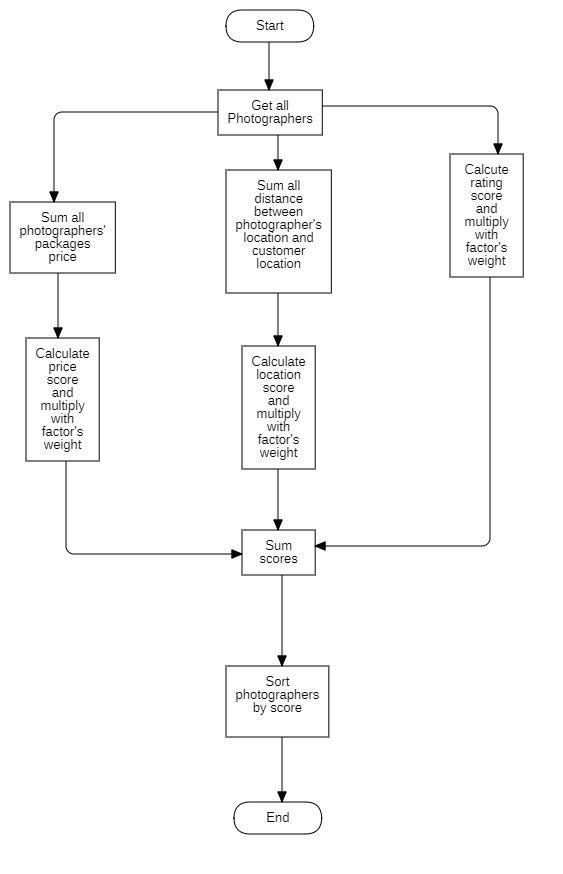


Figure 42 Flowchart - Multiple factors sorting

### 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 best suited to solve problems which has categorizable variables Some application of naive bayes includes: Spam e-mail 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 |

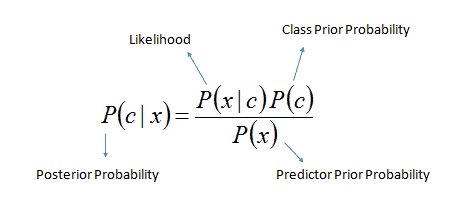
|  |  |  |
| --- | --- | --- |
| **Class** | **Temperature** | |
| High | Normal |
| Yes | 0 | 4 |
| No | 4 | 0 |

|  |  |  |
| --- | --- | --- |
| **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.