



TRƯỜNG ĐẠI HỌC FPT

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	10
Definition and Acronyms	10
I. Project Introduction	11
1. Overview	11
2. Product Background.....	11
3. Existing Systems	12
4. Business Opportunity.....	13
5. Software Product Vision.....	14
6. Project Scope & Limitations	14
II. Project Management Plan	17
1. Overview	17
2. Management Approach	23
3. Master Schedule	24
4. Project Organization	26
5. Project Communication	27
6. Configuration Management.....	28
III. Software Requirement Specification	30
1. Overall Description.....	30
2. User Requirements	31
3. Functional Requirements.....	110
4. Non-Functional Requirements.....	120
5. Other Requirements	121
IV. Software Design Document	124
1. Overall Description.....	124
2. System Architecture Design	124
3. System Detailed Design.....	129
4. Class Specifications	141
5. Data & Database Design	153
6. Algorithm	161
V. Software Testing Documentation	168
1. Overall Description.....	168
2. Test Plan	169

3. Test Cases.....	170
4. Test Reports	170
VI. Release Package & User Guides.....	171
1. Deliverable Package	171
2. Installation Guides	172
3. User Manual.....	181
4. 配達可能なパッケージ	210
5. 設置ガイド	211
6.ユーザーマニュアル	220
VII. Appendix.....	250
1. Glossary [Optional]	250
2. References [Optional]	250
3. Others [Optional]	251

List of Figures

Figure 1 Feature Mind Map	15
Figure 2 Project process.....	24
Figure 3 Project organization.....	26
Figure 4 Context Diagram	30
Figure 5 Overview Use Case Diagram	31
Figure 6 <Guest> Overview.....	34
Figure 7 <Guest> Register	34
Figure 8 <Guest> Login	37
Figure 9 <Customer> Overview.....	39
Figure 10 <Customer> Search photographers	40
Figure 11 <Customer> Get photographer's detail	41
Figure 12 <Customer> Book photographer	42
Figure 13 <Customer> Review photographers	45
Figure 14 <Customer> Get bookings.....	46
Figure 15 <Customer> Edit Booking.....	48
Figure 16 <Customer> Cancel booking	50
Figure 17 <Customer> Open thread	52
Figure 18 <Customer> Edit thread	54
Figure 19 <Customer> Comment thread	56
Figure 20 <Customer> Chat	58
Figure 21 <Photographer> Overview	60
Figure 22 <Photographer> Accept booking	60
Figure 23 <Photographer> Decline booking	62
Figure 24 <Photographer> Edit working weekdays	64
Figure 25 <Photographer> Add busy day.....	66
Figure 26 <Photographer> Get customer's review	68
Figure 27 <Photographer> Add package.....	69
Figure 28 <Photographer> Edit package.....	71
Figure 29 <Photographer> Remove package	73

Figure 30 <Photographer> Add album.....	75
Figure 31 <Photographer> Edit album.....	77
Figure 32 <Photographer> Remove album.....	79
Figure 33 <Authenticated User> Overview.....	81
Figure 34 <Authenticated User> Get profile.....	81
Figure 35 <Authenticated User> Edit profile	82
Figure 36 <Authenticated User> Change password.....	84
Figure 37 <Authenticated User> Logout.....	86
Figure 38 <Admin> Overview.....	87
Figure 39 <Admin> Block user	88
Figure 40 <Admin> Unblock user	89
Figure 41 <Admin> Add category.....	91
Figure 42 <Admin> Edit category.....	93
Figure 43 <Admin> Remove category.....	95
Figure 44 <Admin> Add returning type	96
Figure 45 <Admin> Edit returning type.....	98
Figure 46 <Admin> Remove returning type.....	100
Figure 47 <Admin> Change variables.....	101
Figure 48 <Admin> Ban thread	103
Figure 49 <<system>> Handler Overview	105
Figure 50 <<system>> Handler make booking notification	105
Figure 51 <<system>> Handler make scheduling notification.....	107
Figure 52 <<system>> Handler make weather warning	108
Figure 53 <<system>> Handler make time warning	109
Figure 54 Customer's screen flow.....	111
Figure 55 Photographer's screen flow	112
Figure 56 ER Diagram.....	119
Figure 57 Overall Architecture.....	124
Figure 58 Mobile App Architecture.....	125
Figure 59 Webapi Package Diagram	126
Figure 60 Package Diagram – Mobile app.....	128
Figure 61 Class Diagram.....	129
Figure 62 Sequence Diagram - Add Category	130
Figure 63 Sequence Diagram - Edit Category	130
Figure 64 Sequence Diagram - Remove Category.....	131
Figure 65 Sequence Diagram - Add returning type.....	131
Figure 66 Sequence Diagram - Edit returning type.....	132
Figure 67 Sequence Diagram - Remove returning type.....	132
Figure 68 Sequence Diagram - Block User	133
Figure 69 Sequence Diagram - Unblock User.....	133
Figure 70 Sequence Diagram - Change Variables	134
Figure 71 Activity Diagram - Make request	135
Figure 72 Activity Diagram – Edit Request.....	136
Figure 73 Activity Diagram - Cancel booking	137
Figure 74 Activity Diagram - Rating.....	138
Figure 75 Activity Diagram - Accept booking	139
Figure 76 Activity Diagram - Reject booking	140
Figure 77 State Machine Diagram - Booking.....	141
Figure 78 Database Design.....	153
Figure 79 Flowchart - Multiple factors sorting	163
Figure 80 Test Model	168

Figure 81 Setup files - 1.....	173
Figure 82 Setup files - 2.....	173
Figure 83 Installation - 1	174
Figure 84 Installation - 2	175
Figure 85 Installation - 3	176
Figure 86 Installation - 4	177
Figure 87 Server-side deployment – 1	178
Figure 88 Server-side deployment – 2	178
Figure 89 Server-side deployment – 3	178
Figure 90 Server-side deployment – 4	179
Figure 91 Server-side deployment -5.....	180
Figure 92 Server-side deployment - 6.....	180
Figure 93 Server-side deployment -7.....	181
Figure 94 Register	183
Figure 95 - Login.....	184
Figure 96 - Logout	185
Figure 97 Search photographers.....	186
Figure 98 - Make request for a day.....	187
Figure 99 - Make request for days	188
Figure 100 - Update booking for a day	188
Figure 101 - Update booking for days.....	189
Figure 102 - Decline /Accept /Cancel Booking.....	189
Figure 103 - View Reviews	190
Figure 104 - Write reviews.....	191
Figure 105 - Change Booking Status	192
Figure 106 - Shooting schedule.....	193
Figure 107 - Make changes to daily shooting schedule	194
Figure 108 - Add busy days	195
Figure 109 - Follow Album	196
Figure 110 - Add Album	197
Figure 111 - Edit Album.....	198
Figure 112 - View Services	199
Figure 113 - Add Service	199
Figure 114 - Update Service	200
Figure 115 - Select / View Categories	201
Figure 116 - Chat.....	202
Figure 117 - View Thread	203
Figure 118 - View Comment	204
Figure 119 - Add Comment	205
Figure 120 - Create Thread	206
Figure 121 - Select Photo Delivery Methods	207
Figure 122 - Update Profile	208
Figure 123 - Update Avatar/ Cover	209
図 124 設定ファイル – 1	212
図 125 設定ファイル – 2	212
図 126 インストール – 1	213
図 127 インストール – 2	214
図 128 インストール – 3	215
図 129 インストール – 4	216
図 130 サーバー側の展開 – 1	217
図 131 サーバー側の展開 – 2	217

図 132 サーバー側の展開 – 3	217
図 133 サーバー側の展開 – 4	218
図 134 サーバー側の展開 – 5	219
図 135 サーバー側の展開 – 6	219
図 136 サーバー側の展開 – 7	220
図137-「PBS」アカウントを作成する.....	223
図138-ログアウト.....	224
図139-フォトグラファー／サービスの検索.....	225
図140-一日の予約.....	226
図141-数日の予約.....	227
図142-一日の予約の変更.....	227
図143-数日の予約の変更.....	228
図144-予約の断り/受入 /キャンセル.....	228
図145-レビューを観ます.....	229
図146-レビューを書きます.....	230
図147-予約の状態を変更します.....	231
図148-撮るスケジュール	232
図149-毎日の撮るスケジュールを変更します	233
図150-都合がない日を追加します	234
図151-アルバムをフォローする.....	235
図152-アルバムを追加します.....	236
図153-アルバムを変更します.....	237
図154-サービスを観ます	238
図155-サービスを追加します.....	238
図156-サービスを変更します.....	239
図157-カテゴリーを観ます／選択します	240
図158-チャット.....	241
図159-スレッドを観ます	242
図160-コメントを観ます	243
図161-コメントを追加します	244
図162-スレッドを追加します.....	245
図163-配信方法を選びます	246
図164-プロファイルを編集します	247
図165-アバター／コバーを変更します	248

List of Tables

Table 1 Definition and Acronyms.....	10
Table 2 Team Members	11
Table 3 WBS & Estimation	22
Table 4 Project objectives	22
Table 5 Project risks	23
Table 6 Training plan.....	24
Table 7 Master schedule	25
Table 8 Roles & responsibilities	26
Table 9 Communication Plan	27
Table 10 FU Contacts	28
Table 11 Customer Contacts	28
Table 12 Tools & Infrastructures.....	28

Table 13 Business Rules	30
Table 14 System Actors.....	31
Table 15 Use Case List.....	33
Table 16 <Guest> Register Specification.....	36
Table 17 <Guest> Register Specification.....	38
Table 18 <Customer> Search Photographers Specification.....	41
Table 19 <Customer> Get photographer's detail specification	42
Table 20 <Customer> Book photographer specification	44
Table 21 <Customer> Review photographer specification.....	46
Table 22 <Customer> Get booking specification	48
Table 23 <Customer> Edit booking specification.....	50
Table 24 <Customer> Cancel booking specification	52
Table 25 <Customer> Open thread specification	54
Table 26 <Customer> Edit thread specification.....	56
Table 27 <Customer> Comment thread specification	58
Table 28 <Customer> Chat specification	59
Table 29 <Photographer> Accept booking specification	62
Table 30 <Photographer> Decline booking specification	64
Table 31 <Photographer> Edit working weekdays specification	66
Table 32 <Photographer> Add busy day specification	67
Table 33 <Photographer> Get customer's review specification.....	69
Table 34 <Photographer> Add package specification	71
Table 35 <Photographer> Edit package specification.....	73
Table 36 <Photographer> Remove package specification.....	75
Table 37 <Photographer> Add album specification.....	77
Table 38 <Photographer> Edit album specification.....	79
Table 39 <Photographer> Remove album specification.....	80
Table 40 <Authenticated User> Get profile specification.....	82
Table 41 <Authenticated User> Edit profile specification	84
Table 42 <Authenticated User> Change password specification	85
Table 43 <Authenticated User> Logout specification.....	87
Table 44 <Admin> Block user specification	89
Table 45 <Admin> Unblock user specification.....	91
Table 46 <Admin> Add category specification.....	92
Table 47 <Admin> Edit category specification.....	94
Table 48 <Admin> Remove category specification.....	96
Table 49 <Admin> Add returning type specification	98
Table 50 <Admin> Edit returning type specification	99
Table 51 <Admin> Remove returning type specification.....	101
Table 52 <Admin> Change variables specification.....	103
Table 53 <Admin> Ban thread specification	104
Table 54 <System Handler> Make booking notification specification.....	106
Table 55 <System Handler> Make scheduling notification specification	108
Table 56 <System Handler> Make weather warning specification.....	109
Table 57 <System Handler> Make time warning specification.....	110
Table 58 Customer's screen details	114
Table 59 Photographer's screen details.....	116
Table 60 Non-screen functions	118
Table 61 Entities List	119
Table 62 Messages List.....	123
Table 63 Package Diagram Description.....	127

Table 64 Package Diagram Description.....	129
Table 65 Class Diagram - User.....	142
Table 66 Class Diagram - Role	142
Table 67 Class Diagram - ERole	143
Table 68 Class Diagram - Package	143
Table 69 Class Diagram - Service.....	144
Table 70 Class Diagram - Booking	145
Table 71 Class Diagram - EBookingStatus	145
Table 72 Class Diagram - Album.....	146
Table 73 Class Diagram - Image	146
Table 74 Class Diagram - Category.....	147
Table 75 Class Diagram - DayOfWeek.....	147
Table 76 Class Diagram - Location	148
Table 77 Class Diagram - BookingComment	148
Table 78 Class Diagram - BookingDetail.....	149
Table 79 Class Diagram - BusyDay	149
Table 80 Class Diagram - Thread.....	150
Table 81 Class Diagram - ThreadTopic	151
Table 82 Class Diagram - ThreadComment.....	151
Table 83 Class Diagram - ReturningType	152
Table 84 Class Diagram - EReturningType	152
Table 85 Class Diagram – TimeLocationDetail.....	152
Table 86 Class Diagram - Variable.....	153
Table 87 Database - users	154
Table 88 Database - roles.....	154
Table 89 Database - photographer_packages	155
Table 90 Database - packages_services	155
Table 91 Database - services.....	155
Table 92 Database - bookings	156
Table 93 Database - albums	157
Table 94 Database - locations	157
Table 95 Database - categories	157
Table 96 Database - album_images	157
Table 97 Database - images	158
Table 98 Database - customer_albums	158
Table 99 Database - working_day_of_week	158
Table 100 Database - busy_days.....	158
Table 101 Database - returning_types.....	159
Table 102 Database - booking_details	159
Table 103 Database - time_location_details	159
Table 104 Database - booking_comments	160
Table 105 Database - threads	160
Table 106 Database - thread_comments.....	160
Table 107 Database - thread_topics	160
Table 108 Database - variables	161
Table 109 Date File Design.....	161
Table 110 Testing Levels	168
Table 111 Testing Types.....	169
Table 112 Test Stages	169
Table 113 Human Resources.....	169
Table 114 Environment.....	170

Table 115 Test Milestones	170
Table 116 Deliverables.....	170
Table 117 Test Cases.....	170
Table 118 Test Reports	170
Table 119 Deliverable Package	171
Table 120 PC Hardware requirements.....	172
Table 121 Android Hardware requirements	172
Table 122 Software requirements	173
表123 配達可能なパッケージ	211
表124 PCハードウェア要件	211
表 125ソフトウェア要件	212
表 126 ソフトウェア要件.....	212
Table 127 Glossary	250

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

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		Lecturer

b. Team Members

Full Name	Email	Mobile	Role
Trần Thiên Thảo	Thaottse62888@fpt.edu.vn	0903917982	Leader
Bùi 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 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

Currently, there are a myriad of platforms that help connect freelancers from various industries to the customers. However, there are only a few platforms specializing in photography. In addition, there are no systems or applications that provide photographers with time and business management.

Below are some employment and job-search systems that the team takes into consideration while doing research. Learning from their strengths and weaknesses can help better the team's product/system.

3.1 WeHelp

WeHelp is a platform which allows freelancers from a wide range of industries to market their skills and services that companies, organizations and others may need.

Link:

<https://wehelp.vn/>

<https://apps.apple.com/us/app/wehelp-tìm-người-tìm-việc/id1447795502/>

System actors:

- Customer.
- Freelancer.
- Admin.

Features:

- Allow freelancers to market their skills and services.
- Allow customers to state their requirements and book photographers.

Pros

- Good design, user-friendly

Cons

- Covering a wide range of job sectors, hence not well-support any sector.
- No time management for freelancers.
- No payment method.

3.2 vLance

A job-search website for freelancers, which helps companies or individuals reach the ones that can fulfil their needs.

Link: <https://www.vlance.vn/>

System actors:

- Customer.
- Freelancer.
- Admin.

Features:

- Allow customers to add job postings and desired commissions.
- E-wallet is supported.

Pros

- Allow freelancers to propose competitive commissions.
- E-wallet helps ensure both sides' benefits.

Cons

- Complex layouts, not user-friendly.

3.3 Tripo

Tripo is a young developed web application for photographer booking. Provide photographer and studio ways to approach potential customers and express their skills.

Link: <https://tripo.vn/>

System actors:

- Customer.
- Photographer.
- Admin.

Features:

- Allows customers to post their request as auction format.
- Photographers can offer the price after viewing customers' requests.

Pros

- Specialize in the photography domain.
- Easy to use design.

Cons

- Don't support mobile applications, lack of interactivity.
- Don't support schedule managing features for photographers.
- Customers can't book directly from their favoured photographer.

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

FE-01: Customers can search for photographers based on requirements.

FE-02: Customers can view photographers' albums, services and profiles.

FE-03: Customers can like photographers' albums.

FE-04: Customers can book photographers that meet requirements and be able to cancel bookings due to inconvenient time and weather.

- FE-05: Customers and photographers can view booking history.
- FE-06: Customers and photographers can view directions to booking history.
- FE-07: Customers and photographers can edit and cancel booking.
- FE-08: Customers can rate and write reviews for photographers.
- FE-09: Photographers can manage their schedule.
- FE-10: Photographers can create and edit their albums.
- FE-11: Photographers can manage services they provide.
- FE-12: Photographers can accept or decline customers' requests.
- FE-13: Photographers and admin can view customers' reviews.
- FE-14: Admin can disable customers' review.
- FE-15: Admin, photographers, and customers can sign up and sign in.
- FE-16: Admin, photographers, and customers can add and edit profiles and change passwords.
- FE-17: Admin, photographers, and customers can discuss in forums.
- FE-18: Admin can create and edit and forum's topics.
- FE-19: Admin, photographers, and customers can create, edit, and delete their threads.
- FE-20: Admin, photographers, and customers can create, edit, and delete their comments in threads.
- FE-21: Admin can delete inappropriate comments and threads.
- FE-22: Customers and photographers can send messages to each other.



Figure 1 Feature Mind Map

6.2 Limitations & Exclusions

- LI-01: The application has not supported payment.
- LI-02: The system has not had a photo verification feature.
- LI-03: The system has not provided a website version to both customers and photographers. Users may find it uncomfortable to use applications.
- LI-04: The system only supports Vietnamese language.

II. Project Management Plan

1. Overview

1.1 WBS & Estimation

#	WBS Item	Complexity	Est. Effort (man-days)
1	<i>Initiating</i>		5
1.1	<p>Kick-off project</p> <ul style="list-style-type: none"> • 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	<p>Develop business case</p> <ul style="list-style-type: none"> • 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	<p>Identify stake-holder</p> <ul style="list-style-type: none"> • Identifying • Analysing <ul style="list-style-type: none"> 1. Stakeholder Type 2. Contribution (value) 3. Legitimacy 4. Willingness to engage 5. Influence 6. Involvement • Mapping 	Medium	1

2	Planning		14
2.1	<p>Create business plan</p> <ul style="list-style-type: none"> ● Executive summary ● Opportunity ● Execution ● Company and management summary ● Financial plan ● Appendix 	Medium	2
2.2	Collect requirement	Complex	7
2.3	<p>Define scope</p> <ul style="list-style-type: none"> ● 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	<p>Define vision</p> <ul style="list-style-type: none"> ● Define outcome ● Define what unique brings to the above outcome ● Apply Add relatable, human, aspects 	Medium	1
2.5	<p>Create WBS</p> <ul style="list-style-type: none"> ● Identify the most important pieces of scope ● Break down the project ● Identify the known attributes for each activity 	Medium	1
2.6	<p>Create Risk Management Plan</p> <ul style="list-style-type: none"> ● 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	Simple	1

	<ul style="list-style-type: none"> ● 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 		
3	<i>Executing</i>		22
3.1	Analysis		
3.1.1	<p>Analyse Requirements</p> <ul style="list-style-type: none"> ● 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	<p>Create Use case diagram</p> <ul style="list-style-type: none"> ● 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	<p>Create Use case specification</p> <ul style="list-style-type: none"> ● 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	<p>Create SRS</p> <ul style="list-style-type: none"> ● Create an Outline 	Medium	2

	<ul style="list-style-type: none"> Start with a Purpose Give an Overview Detail Specific Requirements Get Approval for the SRS 		
3.2	Design		
3.2.1	Design System Architecture <ul style="list-style-type: none"> 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 <ul style="list-style-type: none"> 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 <ul style="list-style-type: none"> 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

		FINAL PRESENTATION
	Total Estimated Effort (man-days)	94

Table 3 WBS & Estimation

1.2 Project Objectives

#	Quality Stage	No. of Defects	% of Defect	Notes
1	Initiating	2	8.7	Must be completed in the first stage.
2	Planning	4	17.4	Plan to complete at least 80% of the project, and time distribution.
3	Analysis	4	17.4	Detailed project analysis.
4	Design	2	8.8	Design diagram and database.
5	Prototype	4	17.4	Make a prototype of a project.
6	Implementation	1	4.3	Implementation of the plans.
7	Perform quality assurance	2	8.8	Write test cases to check the quality of the project.
8	Support	1	4.3	Discuss unclear issues with the instructor.
9	Monitoring	1	4.3	Monitoring the steps taken.
10	Controlling	1	4.3	Track the progress of the project.
11	Closing	1	4.3	Review code and document.
Total		23	100%	

Table 4 Project objectives

1.3 Project Risks

#	Risk Description	Impact	Possibility	Response Plans
1	Most members find it unfamiliar to	High	30%	Hold training sessions to get familiar with them.

	use flutter or spring framework			
2	Scrum model is applied for the first time hence there may be difficulties in system development	High	30%	Hold daily meetings to avoid misunderstandings among members.
3	Wrong estimation of time needed	Medium	10%	Regularly track progress and update time estimation.
4	Unexpected scope change	High	10%	Be careful in planning to minimize impact of scope change.
5	Poor code quality	Medium	20%	Strictly follow code convention, carry out acceptance tests to ensure quality source code.

Table 5 Project risks

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.

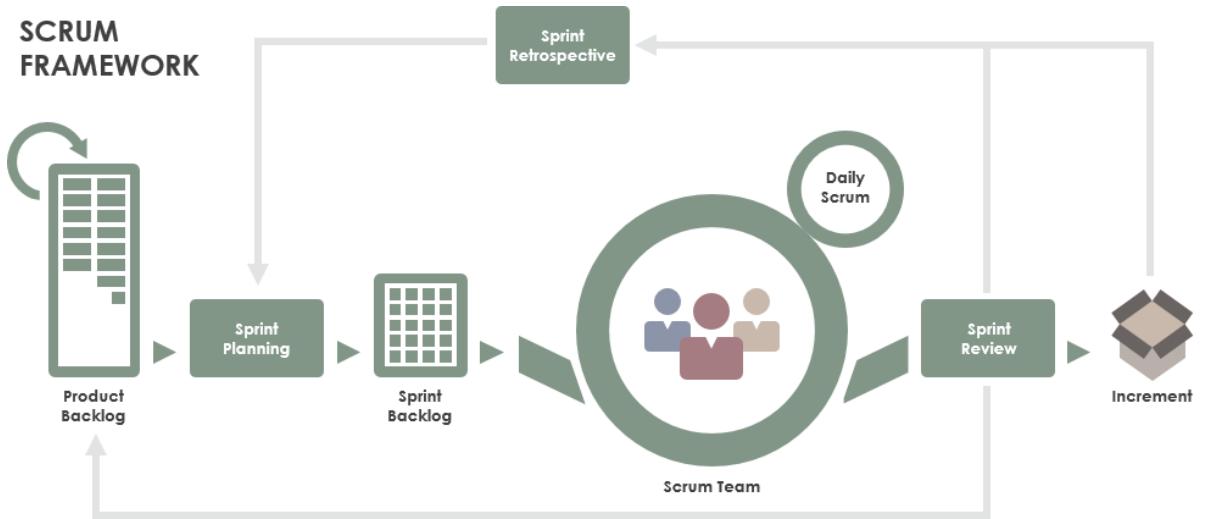


Figure 2 Project process

Reference: <https://www.scrum.org/resources/what-is-scrum>

2.2 Quality Management

To increase the pace of the project, the team take following measures:

- Observe similar available applications.
- Discuss all requirements together to achieve the best quality.
- Hold daily meetings so that all members are acknowledged of the progress and others' ability.

2.3 Training Plan

Training Area	Participants	When, Duration	Waiver Criteria
Java Spring Boot	All team members	3 days	Mandatory
Dart, Flutter	All team members	4 days	Mandatory
StarUML	All team members	3 days	Mandatory
MySQL	All team members	3 days	Mandatory
Git, Github	All team members	1 day	Mandatory

Table 6 Training plan

3. Master Schedule

#	Deliverable	Due Date	Deliverable Scope
1	Project Plan	17/09/2020	Make a goal plan, list tasks to do, time distribution
2	Use case diagram	24/09/2020	Design use case diagram

3	Conceptual diagram	24/09/2020	Design conceptual diagram
4	ER diagram	29/09/2020	Design ER diagram
5	Database	29/09/2020	Design database
6	Design	08/10/2020	Architecture Design, Detailed Design
7	Screen prototype	13/10/2020	Design screen of application
8	Report 1	20/10/2020	Project introduction
9	Report 2	20/10/2020	Project management plan
10	Report 3	20/10/2020	Software requirement specification
11	Report 4	20/10/2020	Software design document
12	Report 5	26/11/2020	Test case document, test documentation
13	Report 6	26/11/2020	Software user guides
14	Report 7	26/11/2020	Final project report
15	Code package 1	15/10/2020	Code & Unit test, System test cases
16	Code package 2	05/11/2020	Code & Unit test, System test cases
17	Code package 3	10/11/2020	Code & Unit test, System test cases
18	Code front-end 1	15/10/2020	Code & design screen
19	Code front-end 2	10/11/2020	Code & design screen
20	Code front-end 3	24/11/2020	Code & design screen
21	Final Package	08/12/2020	Final Codes & documents, User manual

Table 7 Master schedule

4. Project Organization

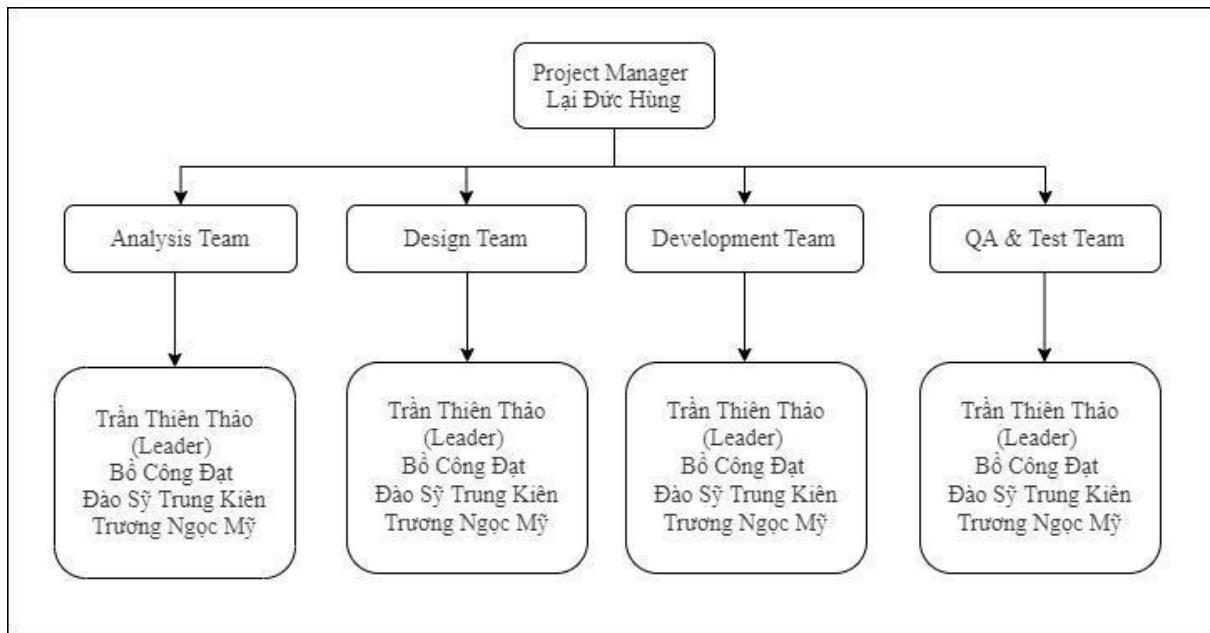


Figure 3 Project organization

4.1 Team & Structures

4.2 Roles & Responsibilities

Role	Responsibility
Project Manager	Give guidance, check, and help correct mistakes in the project.
Analysis Leader	Give guidance and adjust projects' requirements.
Analysis Member	Analyse and update projects' requirements.
Design Leader	Give guidance, design and implement user interface.
Design Member	Design and implement user interface.
Technical Leader	Introduce new techniques to the team.
Test Leader	Write test cases and test plan for the project.
Test Member	Write test cases and test plan for the project.

Table 8 Roles & responsibilities

5. Project Communication

5.1 Communication Plan

Communication Item	Who/ Target	Purpose	When, Frequency	Type, Tool, Method(s)
Kick off meeting	Project supervisor, project team	Introduce the project	Once	Meeting
Project status report	Project supervisor, Project team	Discuss project status, resolve potential issues or delays	Weekly	Meeting
Daily stand up meeting	Project team	Review each member's work status or any happening problems	Daily	Meeting
Meeting minute reports	Project supervisor	Inform supervisor about meeting content	Weekly	Email

Table 9 Communication Plan

5.2 External Interface

a. FU Contacts

Function	Contact Person (name, position)	Contact address (email, telephone)	Responsibility
Supervisor	Lại Đức Hùng	Hungld5@fe.edu.vn	<ul style="list-style-type: none"> - Provide document template - Give instruction to the project team - Review deliverables - Supervise project status - Answer questions about the project

Table 10 FU Contacts

b. Customer Contacts

Function	Contact Person (name, position)	Contact address (email, telephone)	Responsibility
Supervisor	Lại Đức Hùng	Hungld5@fe.edu.vn	<ul style="list-style-type: none"> - Provide document template - Give instruction to the project team - Review deliverables - Supervise project status - Answer questions about the project

Table 11 Customer Contacts

6. Configuration Management

6.1 Tools & Infrastructures

Programming languages	Java, Dart
Framework	Spring framework, Flutter
API	API
DBMS	MySQL
IDEs/Editors	Visual Studio Code, Android Studio, IntelliJ
UML tools	StarUML, Draw.io
Version Control	Github, Git
Deployment server	Hostinger VPS, AWS EC2 (backup)
Project management tool	Microsoft Excel

Table 12 Tools & Infrastructures

6.2 Document Management

<https://github.com/thienthao/pbs/tree/master/Documents>

https://drive.google.com/drive/folders/1A--J0H-8uMiggtQg93W7fExSaF_4xsl

6.3 Source Code Management

<https://github.com/thienthao/pbs>

III. Software Requirement Specification

1. Overall Description

1.1 Product Overview

Photographer Booking System is a new system that helps users find photographers effectively based on their needs. The context diagram below describes the 1.0 version of the system and external entities that interact with it.

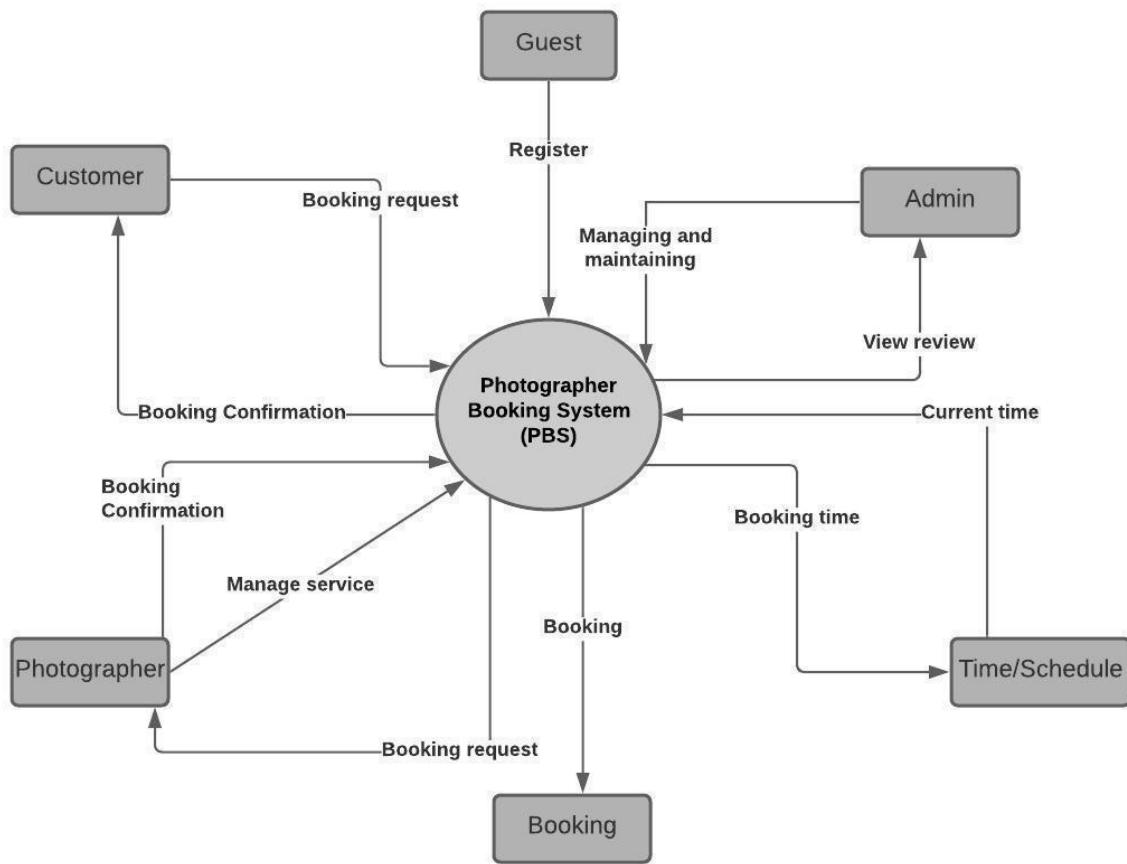


Figure 4 Context Diagram

1.2 Business Rules

ID	Rule Definition

Table 13 Business Rules

2. User Requirements

2.1 Overview

a. Use Case Diagram

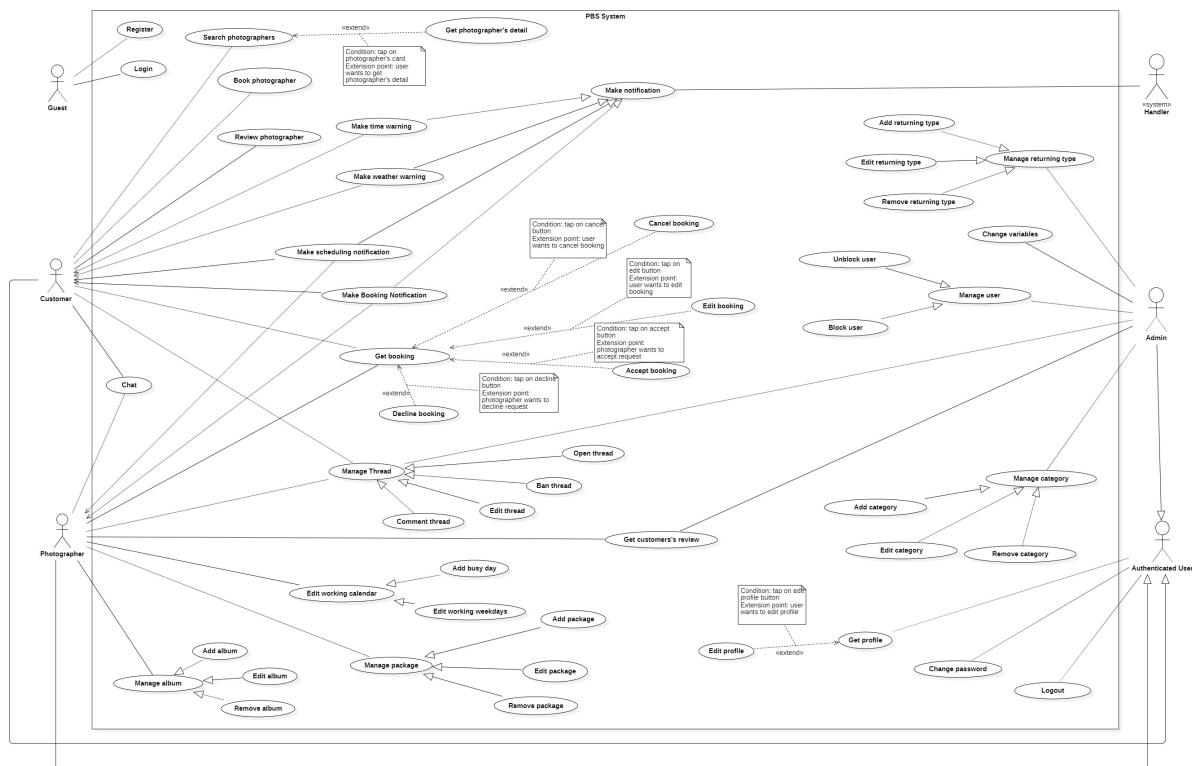


Figure 5 Overview Use Case Diagram

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 processes.

Table 14 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 15 Use Case List

2.2 List of use case

2.2.1 <Guest> Overview use case

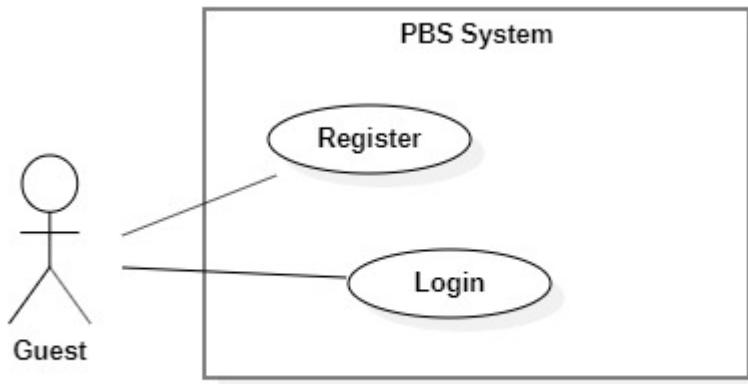


Figure 6 <Guest> Overview

a. <Guest> Register

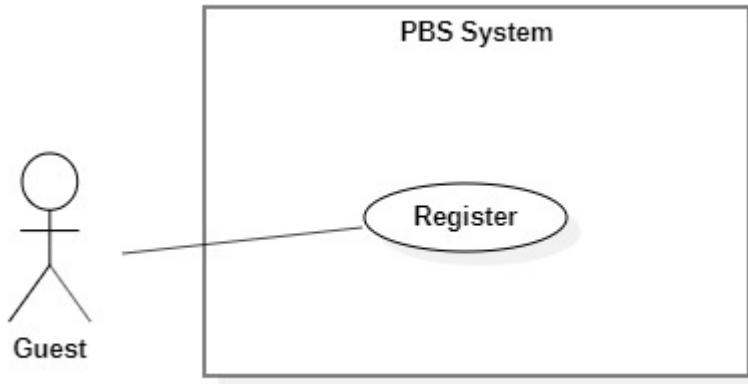


Figure 7 <Guest> Register

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:

			<ul style="list-style-type: none"> - “Họ và 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	<p>The system validates data, outputs message “Đăng ký thành công”</p> <p>[Exception 1]</p> <p>[Exception 2]</p> <p>[Exception 3]</p> <p>[Exception 4]</p>	
Alternative Flows:	N/A			
Exceptions:	<th>No</th> <th>Cause</th> <th>System Response</th>	No	Cause	System Response
	1	Required fields are empty.	<p>System shows error message:</p> <p>“Thông tin này là bắt buộc”</p>	
	2	Password and confirm password are not matched.	<p>System shows error message:</p> <p>“Mật khẩu không khớp”.</p>	
	3	Required fields are not valid.	<p>System shows error message corresponds to the field:</p> <p>“Tên đăng nhập không đúng”</p>	

			<p>“Email không đúng”</p> <p>“Mật khẩu chưa đủ mạnh”</p>
	4	Required information exists in the system (username, email).	<p>System shows error message corresponds to the field:</p> <p>“Tên đăng nhập đã có người sử dụng. Vui lòng nhập tên khác.”</p> <p>“Email đã tồn tại.”</p>
Priority:	Medium		
Frequency of Use:	Usually		
Business Rules:	<ul style="list-style-type: none"> - Needed information are: username, password, phone number, email, full name. - Username format should be 8-20 characters long, starts with character and underscore or dot can't be at the end or beginning. - Password format should be a minimum eight characters, contains at least one number, one letter and one special character. - Encrypt password using SHA256 hash function. - Email format should match regex: <code>^\\w+@[a-zA-Z_]+?\\.[a-zA-Z]{2,3}\$</code> - Phone should be in Vietnamese format. - All information must not be empty when submit. - After registering successfully, navigate back to login view. 		
Other Information:	N/A		
Assumptions:	N/A		

Table 16 <Guest> Register Specification

b. <Guest> Login

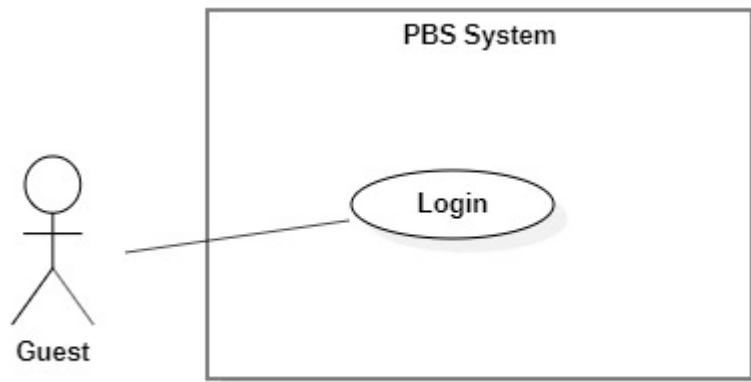


Figure 8 <Guest> Login

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:	<p>Success: Guest successfully logged into the system.</p> <p>Fail: System shows error message.</p>		
Normal Flow:	Step	Actor Action	System Response
	1	Guest goes to login view	<p>System requires information from Guest:</p> <ul style="list-style-type: none"> - “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	<p>System validates data, outputs message “Đăng nhập thành công”</p> <p>[Exception 1]</p> <p>[Exception 2]</p>
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:	<ul style="list-style-type: none"> - User login using username and password. - Username and password format match with the one they have registered. - Encrypt password to SHA256 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 17 <Guest> Register Specification

2.2.2 <Customer> Overview use case

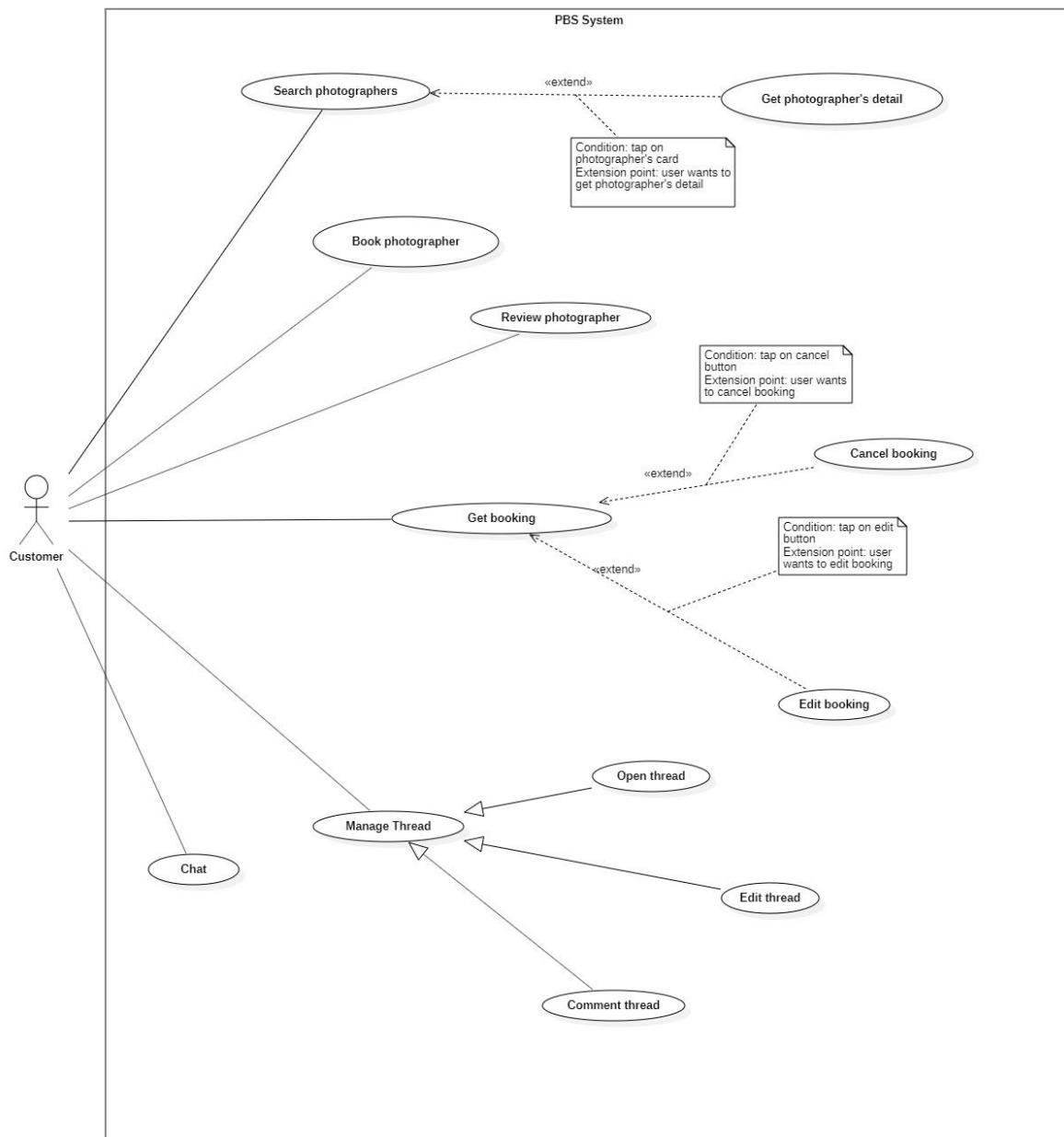


Figure 9 <Customer> Overview

a. <Customer> Search photographer

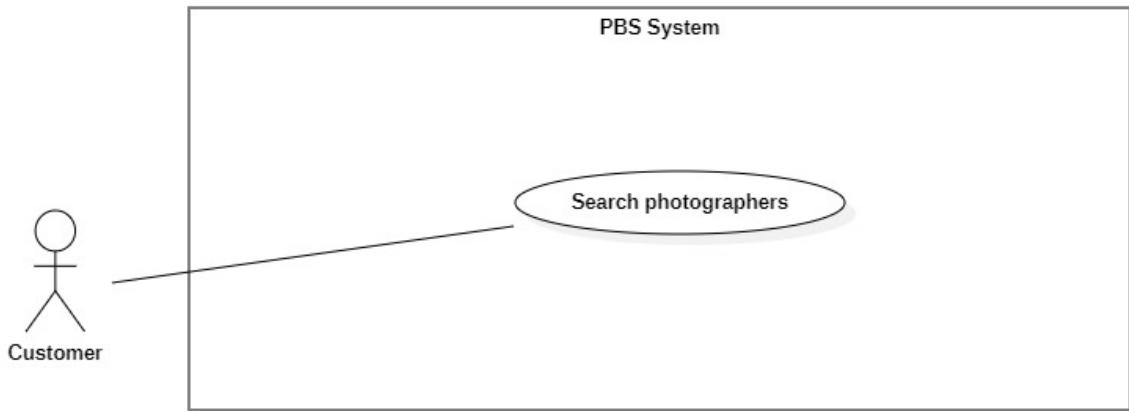


Figure 10 <Customer> Search photographers

UC ID and Name:	UC – 03 Search photographers														
Created By:	Trần Thiên Thảo	Date Created:	08/10/2020												
Primary Actor:	Customer	Secondary Actors:													
Trigger:	The customer sends a request to find photographers.														
Description:	This use case allows customers to find photographers and also packages.														
Preconditions:	User must be authenticated as Customer.														
Post-conditions:	<p>Success: A list of photographers is shown.</p> <p>Fail: System shows error messages.</p>														
Normal Flow:	<table border="1"> <thead> <tr> <th>Step</th> <th>Actor Action</th> <th>System Response</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Customer goes to photographers searching view.</td> <td>System requires information from Guest: - “Tìm kiếm”: text input.</td> </tr> <tr> <td>2</td> <td>Customer inputs information.</td> <td></td> </tr> <tr> <td>3</td> <td>Customer sends request to search photographers.</td> <td>System displays list of photographers and packages. [Exception 1]</td> </tr> </tbody> </table>	Step	Actor Action	System Response	1	Customer goes to photographers searching view.	System requires information from Guest: - “Tìm kiếm”: text input.	2	Customer inputs information.		3	Customer sends request to search photographers.	System displays list of photographers and packages. [Exception 1]		
Step	Actor Action	System Response													
1	Customer goes to photographers searching view.	System requires information from Guest: - “Tìm kiếm”: text input.													
2	Customer inputs information.														
3	Customer sends request to search photographers.	System displays list of photographers and packages. [Exception 1]													
Alternative Flows:	N/A														
Exceptions:	<table border="1"> <thead> <tr> <th>No</th> <th>Cause</th> <th>System Response</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>The process runs into an internal error.</td> <td>System shows error message: “Đã có lỗi xảy ra”.</td> </tr> </tbody> </table>	No	Cause	System Response	1	The process runs into an internal error.	System shows error message: “Đã có lỗi xảy ra”.								
No	Cause	System Response													
1	The process runs into an internal error.	System shows error message: “Đã có lỗi xảy ra”.													

Priority:	Medium
Frequency of Use:	Sometimes
Business Rules:	<ul style="list-style-type: none">- User inputs text in search field, system returns photographers' name and packages' name contains the text.- The list of photographers will be ordered by rating count.- List of photographers is shown with following information: name, avatar, rating count.- List of packages is shown with following information: name, price, description, photographer's name.- If the list is empty, return empty text to customer.
Other Information:	N/A
Assumptions:	N/A

Table 18 <Customer> Search Photographers Specification

b. <Customer> Get photographer's detail

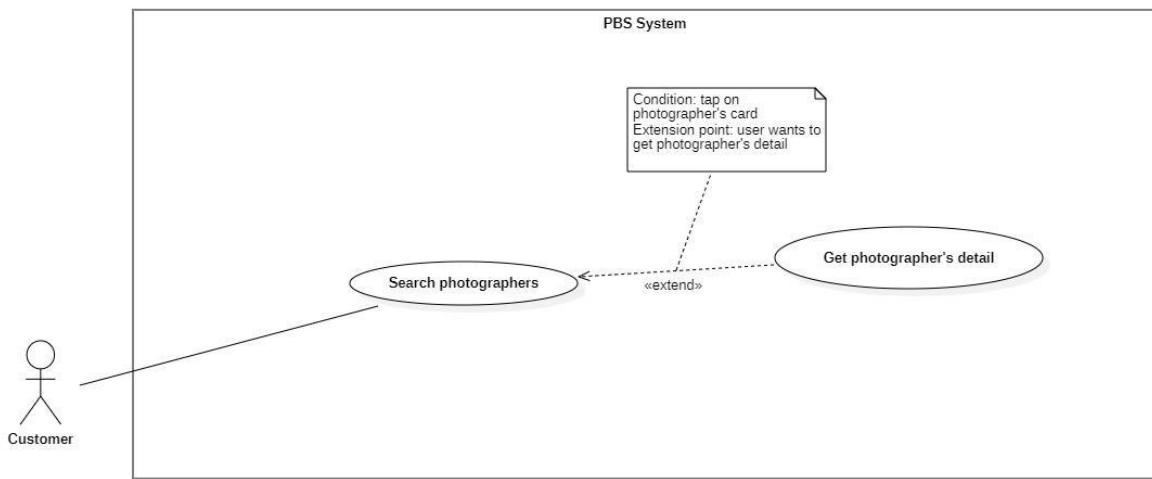


Figure 11 <Customer> Get photographer's detail

UC ID and Name:	UC – 04 Get photographer's detail		
Created By:	Trần Thiên Thảo	Date Created:	08/10/2020
Primary Actor:	Customer	Secondary Actors:	
Trigger:	The customer sends a request to get the photographer's details.		
Description:	This use case allows customers to get photographer's detail.		
Preconditions:	User has been authenticated as a Customer.		
Post-conditions:	Success: The system displays the photographer's detail. Fail: The system shows error message		
Normal Flow:	Step	Actor Action	System Response

	1	Customer sends request to get photographer's detail.	System returns photographer's detail [Exception 1]
Alternative Flows:	N/A		
Exceptions:	No	Cause	System Response
	1	The process runs into an internal error.	System shows error message: "Đã có lỗi xảy ra"
Priority:	Medium		
Frequency of Use:	Usually		
Business Rules:	<ul style="list-style-type: none"> - The photographer's detail return contains the following information: photographer's basic information like rating count, self-description, customers' reviews, photographer's working calendar, photographer's packages. - Photographer working calendar includes the date they have been booked and the date they are busy. - Photographer's packages contain the following information: package's name, package's price, package's description, package's services, package's time anticipation. - If any information is empty, return empty text. 		
Other Information:	N/A		
Assumptions:	N/A		

Table 19 <Customer> Get photographer's detail specification

c. <Customer> Book photographer

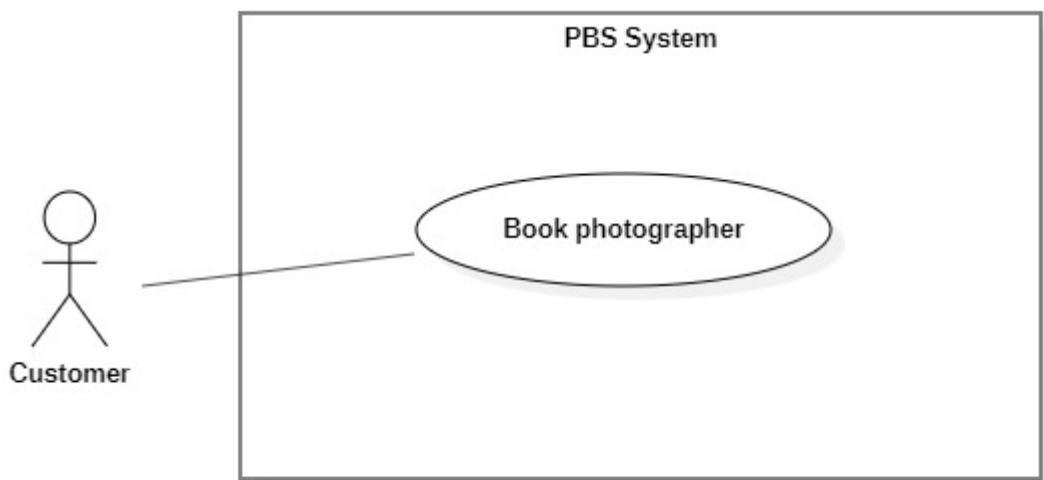


Figure 12 <Customer> Book photographer

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:	<ul style="list-style-type: none"> - Users must login into the system with the role Customer. - The photographer must exist in the system with status as “active”. 		
Post-conditions:	<p>Success: A booking will be stored with status is “Pending”</p> <p>Fail: System outputs error message and sends to Customer.</p>		
Normal Flow:	Step	Actor Action	System Response
	1	The customer goes to booking view	<p>System returns the available photographer's detail.</p> <p>System requires information from customers:</p> <ul style="list-style-type: none"> - “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	<ul style="list-style-type: none"> - The system validates data - The system adds new booking to the system, outputs “Đặt hẹn thành công” and send to photographer. <p>[Exception 1]</p> <p>[Exception 2]</p>

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
Priority:	Medium		
Frequency of Use:	Usually		
Business Rules:	<ul style="list-style-type: none"> - 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. - All information must not be empty. - 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. 		
Other Information:	N/A		
Assumptions:	N/A		

Table 20 <Customer> Book photographer specification

d. <Customer> Review photographers

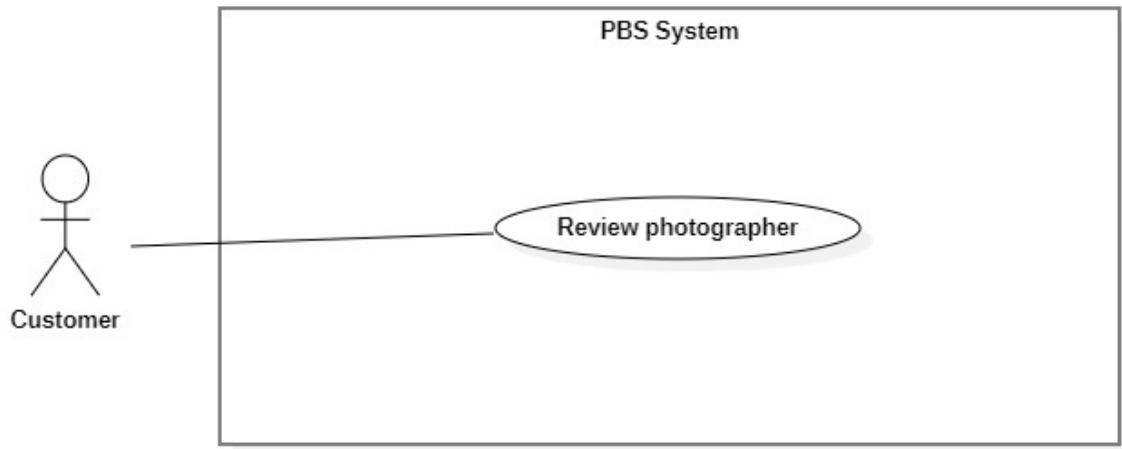


Figure 13 <Customer> Review photographers

UC ID and Name:	UC – 06 Review photographer		
Created By:	Trần Thiên Thảo	Date Created:	08/10/2020
Primary Actor:	Customer	Secondary Actors:	
Trigger:	The customer sends a request to review photographer.		
Description:	This use case allows customers to review photographer after booking is done.		
Preconditions:	<ul style="list-style-type: none"> - User has been authenticated as Customer - Booking status is “Đã hoàn thành” 		
Post-conditions:	<p>Success: The review will be stored in the system and response message to Customer.</p> <p>Fail: System shows error message.</p>		
Normal Flow:	Step	Actor Action	System Response
	1	The customer goes to review view	<p>System requires information from Customer:</p> <ul style="list-style-type: none"> - “Bình luận của bạn”: text - “Đánh giá dịch vụ”: slider
	2	The customer inputs information and sends requests	System creates new review and outputs message: “Đã gửi bình luận”

			[Exception 1] [Exception 2]
Alternative Flows:	N/A		
Exceptions:	No	Cause	System Response
	1	Customer inputs more than 150 characters for comment	System outputs error message: “Đã vượt quá 150 ký tự”
	2	The process run into internal error	System outputs error message: “Đã có lỗi xảy ra”
Priority:	Medium		
Frequency of Use:	Sometimes		
Business Rules:	<ul style="list-style-type: none"> - The customer can only review if a booking state is “DONE”. - The review contains of comment and rating. - Rating is double value from 0 to 5. - Customer can edit their rating. 		
Other Information:	N/A		
Assumptions:	N/A		

Table 21 <Customer> Review photographer specification

e. <Customer> Get booking

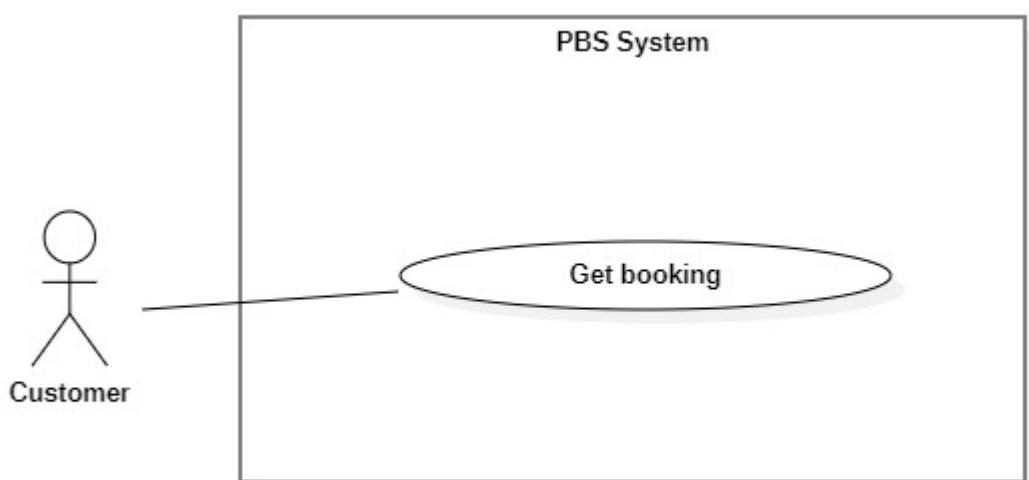


Figure 14 <Customer> Get bookings

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 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:	<p>Success: System displays list of bookings.</p> <p>Fail: System shows error message.</p>		
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:	<ul style="list-style-type: none"> - The list of bookings is displayed and ordered by most recent activity (System will save the time of every activity like accepting, cancelling, rejecting, etc..). - Booking has following status: <ul style="list-style-type: none"> • Pending • Ongoing • Editing • Done • Rejected • Cancelled - Booking has following information: Name of the person who booked, time details (Datetime), location details, booking status, booked package. - Time details have following information: shooting date, start time, end time, edit deadline, shooting time anticipated. 		

	<ul style="list-style-type: none"> - Location details have following information: location name, location's latitude and longitude. - Package has following information: package's name, package's price, package's services.
Other Information:	N/A
Assumptions:	N/A

Table 22 <Customer> Get booking specification

f. <Customer> Edit Booking

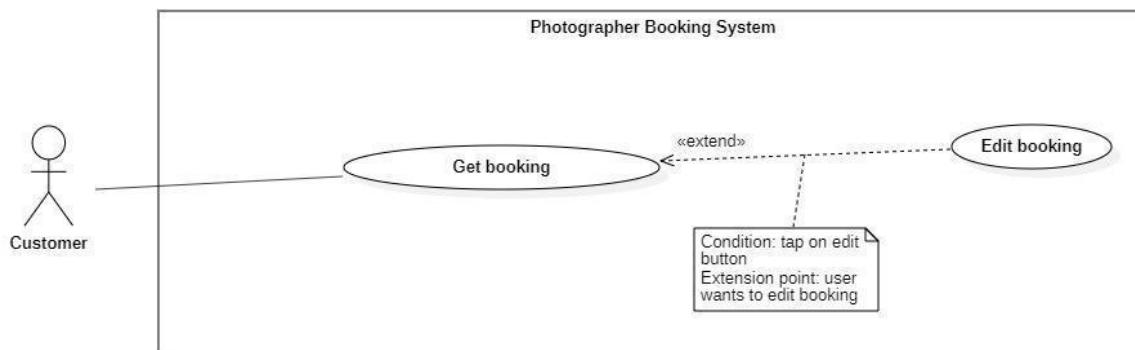


Figure 15 <Customer> Edit Booking

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 customers to edit their booking.		
Preconditions:	The User has been authenticated as Customer.		
Post-conditions:	<p>Success: Booking successfully edited and saved to the system.</p> <p>Fail: System shows error message.</p>		
Normal Flow:	Step	Actor Action	System Response
	1	Customer sends request to receive booking detail	<p>System returns booking detail with following information:</p> <ul style="list-style-type: none"> - "Trạng thái": text - "Thời gian": text - "Thời gian tác nghiệp dự kiến": text

			<ul style="list-style-type: none"> - “Thời gian trả ảnh”: text - “Địa điểm”: map - “Thông tin khách hàng”: <ul style="list-style-type: none"> • “Tên khách hàng”: text - “Tên gói dịch vụ”: <ul style="list-style-type: none"> • “Tên gói”: text • “Mô tả”: text • “Giá dịch vụ”: text • “Dịch vụ”: list of texts <p>[Exception 1]</p>
		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]	<ul style="list-style-type: none"> - The system validates data - The system adds new booking to the system, outputs “Cập nhật thành công” and send to photographer. <p>[Exception 2]</p> <p>[Exception 3]</p>
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 runs 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:	<ul style="list-style-type: none"> - Customer can only edit the booking if its status is “Pending”. - Customer can edit information like shooting time, shooting location, package, returning type and send request to photographer. - All information must not be empty. - Editing deadline must be after shooting time at least 1 day. - After edit, save editing time to the system. - After edit successfully, sends notification to the photographer. 		
Other Information:	N/A		
Assumptions:	N/A		

Table 23 <Customer> Edit booking specification

g. <Customer> Cancel booking

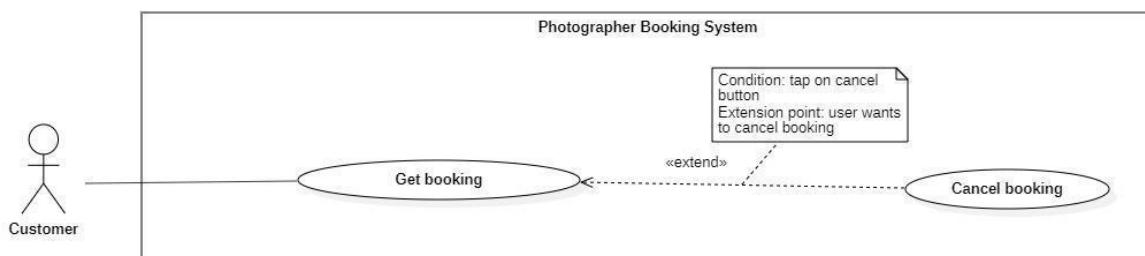


Figure 16 <Customer> Cancel booking

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	<p>System returns booking detail with following information:</p> <ul style="list-style-type: none"> - “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”: <ul style="list-style-type: none"> • “Tên khách hàng”: text - “Tên gói dịch vụ”: <ul style="list-style-type: none"> • “Tên gói”: text • “Mô tả”: text • “Giá dịch vụ”: text • “Dịch vụ”: list of texts <p>[Exception 1]</p>
	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]	<p>System outputs message: “Đã hủy cuộc hẹn”</p> <p>[Exception 2]</p>
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 show error message: "Đã có lỗi xảy ra"
Priority:	Medium		
Frequency of Use:	Sometimes		
Business Rules:	<ul style="list-style-type: none"> - Customers or Photographers can cancel if status is PENDING, ONGOING, EDITING. - When cancels, get the cancellation reason and save to system. - After cancelling, the status will be Cancelled. - After cancelling, increase the cancellation rate of the user. - After one side cancels, notification will be sent to the other person. 		
Other Information:	N/A		
Assumptions:	N/A		

Table 24 <Customer> Cancel booking specification

h. <Customer> Open thread

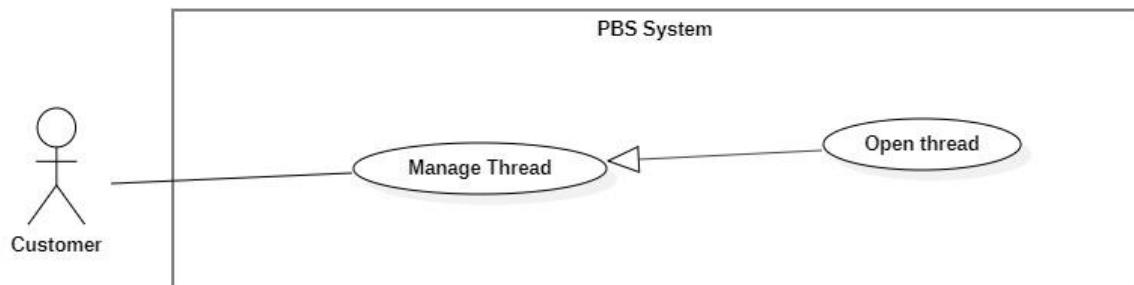


Figure 17 <Customer> Open thread

UC ID and Name:	UC – 12 Open thread		
Created By:	Trương Ngọc Mỹ	Date Created:	08/10/2020
Primary Actor:	Customer, Photographer, Admin	Secondary Actors:	
Trigger:	User sends a request to open a thread.		
Description:	This use case allows users to open new threads on forums.		
Preconditions:	The User has been authenticated.		
Post-conditions:	<p>Success: A new thread will be stored in the system.</p> <p>Fail: System outputs error message.</p>		

Normal Flow:	Step	Actor Action	System Response
	1	User goes to a new thread view.	<p>Systems requires information from Customer:</p> <ul style="list-style-type: none"> - “Chủ đề”: combo box, required - “Tiêu đề”: text, required - “Nội dung”: text, required - “Hình ảnh”: image picker
	2	The User inputs required information and sends a request to open a new thread.	<p>System validates information, add new thread to the system, outputs message “Đã tạo bài viết mới”</p> <p>[Exception 1]</p> <p>[Exception 2]</p>
Alternative Flows:	N/A		
Exceptions:	No	Cause	System Response
	1	User inputs information incorrectly	<p>System outputs error message corresponds to data fields:</p> <p>“Bạn chưa chọn chủ đề”</p> <p>“Tiêu đề không thể để trống”</p> <p>“Nội dung đã vượt quá 250 ký tự”</p>
	2	The process runs into an internal error.	System outputs error message: “Đã có lỗi xảy ra”
Priority:	Medium		
Frequency of Use:	Sometimes		
Business Rules:	<ul style="list-style-type: none"> - Needed information are: thread's name, thread's content, thread's topic. - User inputs thread's name and content as text. 		

	- Thread's topic will be picked from topic option list.
Other Information:	N/A
Assumptions:	N/A

Table 25 <Customer> Open thread specification

i. <Customer> Edit thread

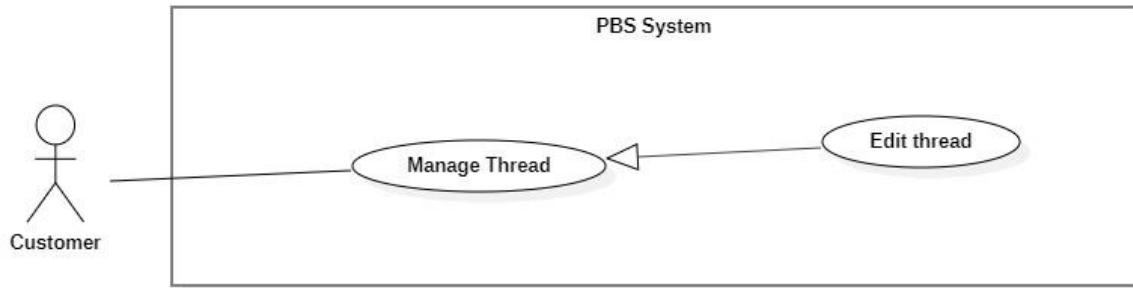


Figure 18 <Customer> Edit thread

UC ID and Name:	UC – 13 Edit thread		
Created By:	Trương Ngọc Mỹ	Date Created:	08/10/2020
Primary Actor:	Customer, Photographer, Admin	Secondary Actors:	
Trigger:	User sends a request to edit the thread.		
Description:	This use case allows users to edit their thread.		
Preconditions:	- The User has been authenticated.		
Post-conditions:	Success: Thread will be updated in the system with new content. Fail: System shows error message.		
Normal Flow:	Step	Actor Action	System Response
	1	User sends request to receive thread detail.	System return thread detail with following information: - "Chủ đề": combo box, required - "Tiêu đề": text, required - "Nội dung": text, required - "Hình ảnh": image picker

			<p>System requires information from customer:</p> <ul style="list-style-type: none"> - “Chủ đề”: combo box, required - “Tiêu đề”: text, required - “Nội dung”: text, required - “Hình ảnh”: image picker <p>[Exception 1]</p>
	2	The user inputs required information and sends a request to edit the thread.	<p>System outputs “Bạn muốn cập nhật bài viết này?”</p>
	3	User confirms “Có” [Alternative 1]	<p>System validates information, update thread to the system, outputs message “Đã cập nhật bài viết”</p> <p>[Exception 2]</p> <p>[Exception 3]</p>
Alternative Flows:	No	Actor Action	System Response
	1	User confirms “Không”	Back to thread detail view
Exceptions:	Step	Cause	System Response
	1	System can't get thread detail	System outputs “Không tìm được bài viết”
	2	User inputs information incorrectly	<p>System outputs error message corresponds to data fields:</p> <p>“Bạn chưa chọn chủ đề”</p> <p>“Tiêu đề không thể để trống”</p> <p>“Nội dung đã vượt quá 250 ký tự”</p>
	3	The process runs into an internal error.	System outputs error message: “Đã có lỗi xảy ra”

Priority:	Medium
Frequency of Use:	Sometimes
Business Rules:	<ul style="list-style-type: none"> - User can edit thread no matter how long they have posted it. - User can edit every information they have posted. Like thread's name, thread's content. - After edit successfully, save the datetime of editing. - User can't change the topic of the thread.
Other Information:	N/A
Assumptions:	N/A

Table 26 <Customer> Edit thread specification

j. <Customer> Comment thread

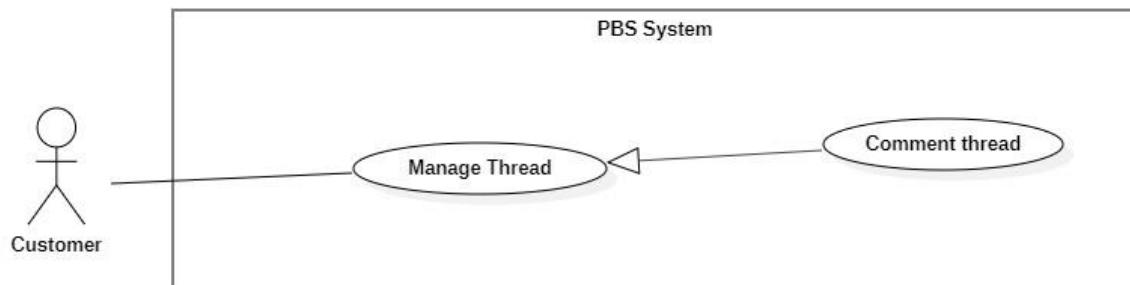


Figure 19 <Customer> Comment thread

UC ID and Name:	UC – 14 Comment thread		
Created By:	Trương Ngọc Mỹ	Date Created:	08/10/2020
Primary Actor:	Customer, Photographer, Admin	Secondary Actors:	
Trigger:	The user sends a request to the comment thread.		
Description:	This use case allows the user to comment on the thread.		
Preconditions:	- The User has been authenticated.		
Post-conditions:	<p>Success: A new comment will be stored in the system.</p> <p>Fail: System shows error message.</p>		
Normal Flow:	Step	Actor Action	System Response
	1	User sends request to receive thread detail.	<p>System return thread detail with following information:</p> <ul style="list-style-type: none"> - “Chủ đề”: combo box, required

		<ul style="list-style-type: none"> - “Tiêu đề”: text, required - “Nội dung”: text, required - “Hình ảnh”: image picker <p>System requires information from customer:</p> <ul style="list-style-type: none"> - “Bình luận của bạn”: text, required - “Hình ảnh”: image picker <p>[Exception 1]</p>	
	2	The user inputs required information and sends a request to the comment thread.	System outputs “Bạn muốn bình luận bài viết này?”
	3	User confirm “Có” [Alternative 1]	<p>System validates information, add new comment to the system, outputs message “Đã bình luận bài viết”</p> <p>[Exception 2]</p> <p>[Exception 3]</p>
Alternative Flows:	No	Actor Action	System Response
	1	User confirm “Không”	Back to thread detail view
Exceptions:	No	Cause	System Response
	1	System can't get thread detail	System outputs “Không tìm được bài viết”
	2	Customer inputs information incorrectly	<p>System outputs error message corresponds to data fields:</p> <p>“Bình luận đã vượt quá 100 ký tự”</p>
	3	The process runs into an internal error.	System outputs error message: “Đã có lỗi xảy ra”
Priority:	Medium		

Frequency of Use:	Sometimes
Business Rules:	<ul style="list-style-type: none"> - User inputs comment as text. Length is 350 words. - After the comment is made, notification will be sent to thread owner.
Other Information:	N/A
Assumptions:	N/A

Table 27 <Customer> Comment thread specification

k. <Customer> Chat

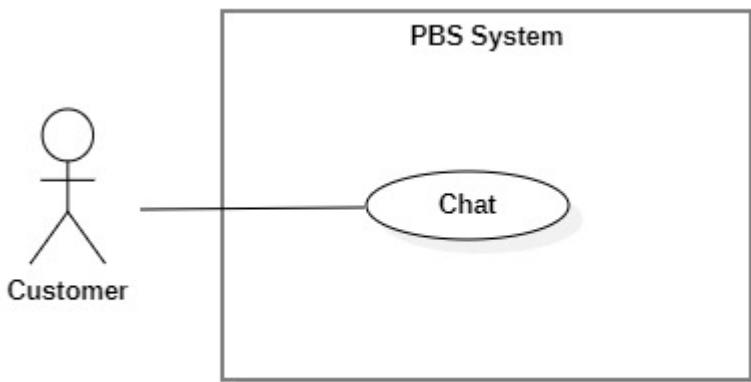


Figure 20 <Customer> Chat

UC ID and Name:	UC – 15 Chat		
Created By:	Trương Ngọc Mỹ	Date Created:	08/10/2020
Primary Actor:	Customer, Photographer	Secondary Actors:	
Trigger:	Customer or Photographer sends a request to chat.		
Description:	This use case allows the users to chat with each other.		
Preconditions:	- The User has been authenticated.		
Post-conditions:	<p>Success: user send message to another user successfully.</p> <p>Fail: System shows error message.</p>		
Normal Flow:	Step	Actor Action	System Response
	1	User goes to chat view.	<p>System return result with following information:</p> <ul style="list-style-type: none"> - Photographer's name: text - Old messages: text

			System receives chat input from customer. [Exception 1]
	2	User inputs messages and send request to chat with photographer.	System send messages to photographer and output in chat view. [Exception 2]
Alternative Flows:	N/A		
Exceptions:	No	Cause	System Response
	1	System can't get chat view with another user.	System outputs "Không tìm được фотографер/ Khách hàng"
	2	The process runs into an internal error.	System outputs error message: "Đã có lỗi xảy ra"
Priority:	Medium		
Frequency of Use:	Sometimes		
Business Rules:	<ul style="list-style-type: none"> - Chat view contains following information: Full name of two people who joined, previous messages. - After a message is sent, notification will be sent to the other person. 		
Other Information:	N/A		
Assumptions:	N/A		

Table 28 <Customer> Chat specification

2.2.3 <Photographer> Overview Use Case

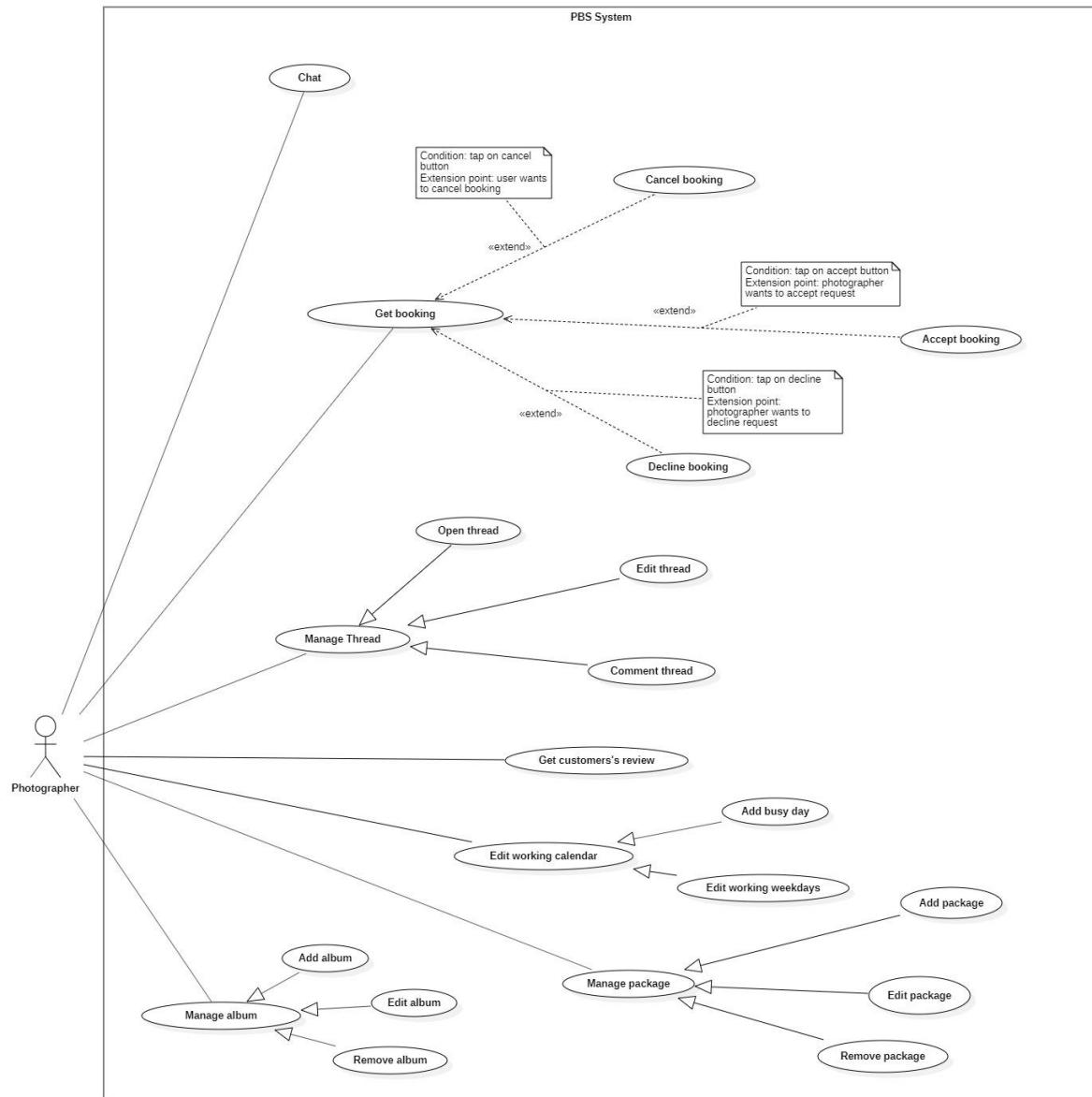


Figure 21 <Photographer> Overview

a. <Photographer> Accept booking

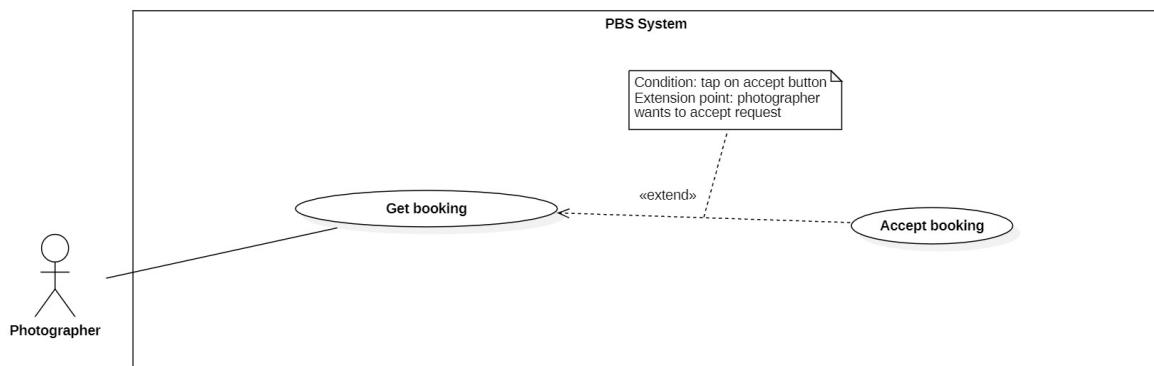


Figure 22 <Photographer> Accept booking

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:	<ul style="list-style-type: none"> - User has been authenticated as a Photographer. - The booking time is still valid. 		
Post-conditions:	<p>Success: The booking will be updated in the system with status change to “Ongoing”</p> <p>Fail: System shows error message</p>		
Normal Flow:	Step	Actor Action	System Response
	1	Photographer sends request to receive booking details.	<p>System returns booking detail with following information:</p> <ul style="list-style-type: none"> - “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”: <ul style="list-style-type: none"> • “Tên khách hàng”: text - “Tên gói dịch vụ”: <ul style="list-style-type: none"> • “Tên gói”: text • “Mô tả”: text • “Giá dịch vụ”: text • “Dịch vụ”: list of texts <p>[Exception 1]</p>
	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:	<ul style="list-style-type: none"> - If the booking time of the booking conflicts with old booking, accepting will be fail. Example: Photographer has an Ongoing booking at 9:00 am 20/10/2020, photographer can't accept another booking at 9:00 am. - After accepting, change status of booking to “Ongoing”. - After photographer accept, notification will be sent to customer. 		
Other Information:	N/A		
Assumptions:	N/A		

Table 29 <Photographer> Accept booking specification

b. <Photographer> Decline booking

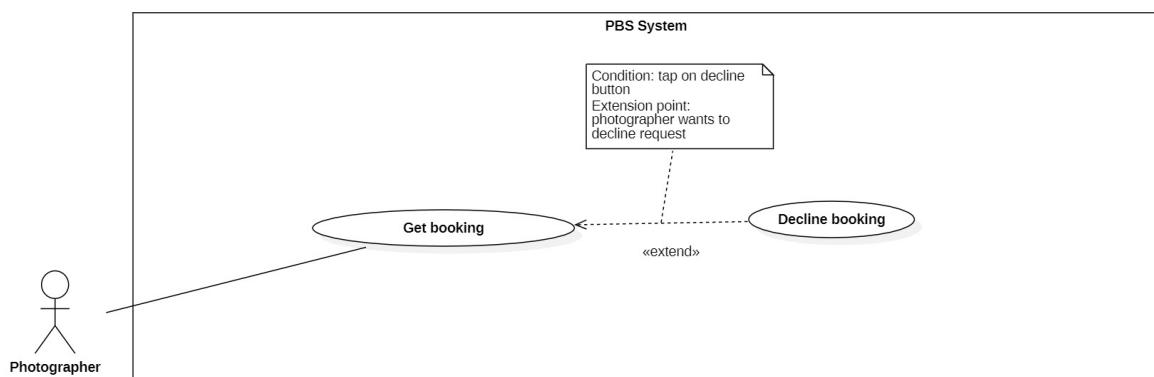


Figure 23 <Photographer> Decline booking

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:	<ul style="list-style-type: none"> - The User has been authenticated as a Photographer. - The booking time is still valid 		
Post-conditions:	<p>Success: The booking will be updated in the system and status change to “Rejected”</p> <p>Fail: System shows error message</p>		
Normal Flow:	Step	Actor Action	System Response
	1	Photographer sends request to receive booking details.	<p>System returns booking detail with following information:</p> <ul style="list-style-type: none"> - “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”: <ul style="list-style-type: none"> • “Tên khách hàng”: text - “Tên gói dịch vụ”: <ul style="list-style-type: none"> • “Tên gói”: text • “Mô tả”: text • “Giá dịch vụ”: text • “Dịch vụ”: list of texts <p>[Exception 1]</p>
	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:	<ul style="list-style-type: none"> - Photographer can only decline a booking with status “Pending”. - After declined, change booking status to “Rejected”. - When decline, get the rejection reason and save to system. - After photographer decline a booking, notification will be sent to customer. - Decline a booking will increase photographer's rejection rate, it could be used by admin to supervise photographers. 		
Other Information:	N/A		
Assumptions:	N/A		

Table 30 <Photographer> Decline booking specification

c. <Photographer> Edit working weekdays

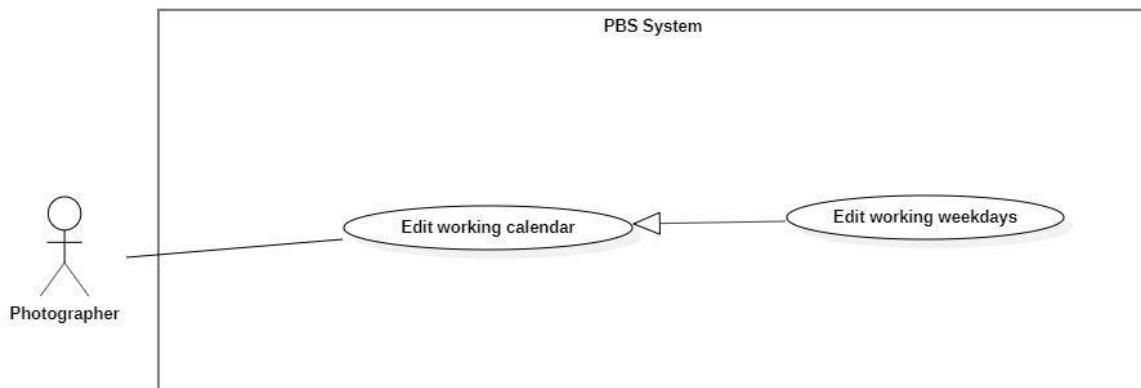


Figure 24 <Photographer> Edit working weekdays

UC ID and Name:	UC – 16 Edit working weekdays		
Created By:	Trương Ngọc Mỹ	Date Created:	08/10/2020

Primary Actor:	Photographer	Secondary Actors:	
Trigger:	The photographer sends requests to edit working weekdays.		
Description:	This use case allows photographers to edit their working weekday.		
Preconditions:	The User has been authenticated as a Photographer.		
Post-conditions:	<p>Success: Photographer's working weekday will be updated in the system.</p> <p>Fail: System outputs error message.</p>		
Normal Flow:	Step	Actor Action	System Response
	1	Photographer goes to work weekday editing view.	System requires information from photographer: - “Chọn ngày làm việc”: checkbox
	2	Photographer inputs required information and sends requests to add busy days.	System outputs “Xác nhận”
	3	Photographer confirm “Có” [Alternative 1]	System add busy day to system and outputs message “Đã điều chỉnh thời gian làm việc” [Exception 1] [Exception 2]
Alternative Flows:	Step	Actor Action	System Response
	1	Photographer confirms “Không”.	Back to working weekday editing view.
Exceptions:	No	Cause	System Response
	1	The process runs into an internal error	System outputs “Đã có lỗi xảy ra”
Priority:	Medium		
Frequency of Use:	Sometimes		
Business Rules:	<ul style="list-style-type: none"> - Weekday is: Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday. - Photographer can decide whether or not they want to work on a weekday. A boolean isWorking will be used to keep track of the photographer's working weekday. - Photographer can set starting time and ending time of a weekday. 		

	<ul style="list-style-type: none"> - All booking date will not be affected if they are accepted before photographer change the working weekday.
Other Information:	N/A
Assumptions:	N/A

Table 31 <Photographer> Edit working weekdays specification

d. <Photographer> Add busy day

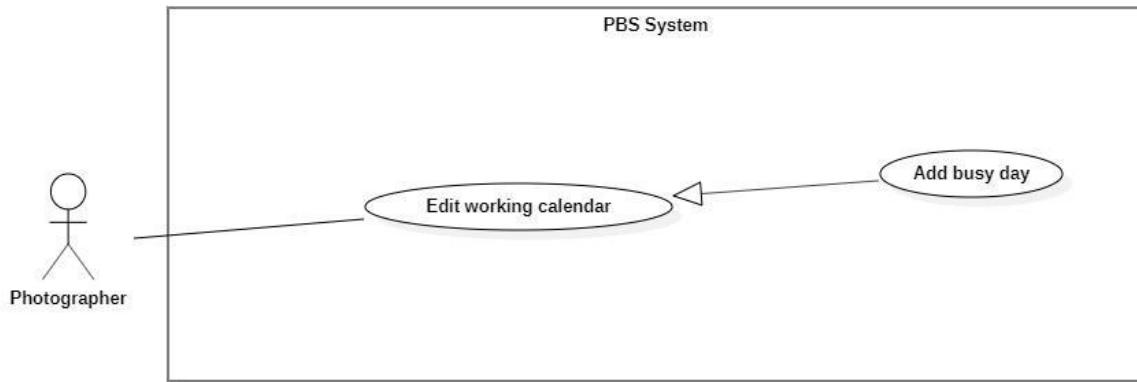


Figure 25 <Photographer> Add busy day

UC ID and Name:	UC – 17 Add busy day		
Created By:	Trương Ngọc Mỹ	Date Created:	08/10/2020
Primary Actor:	Photographer	Secondary Actors:	
Trigger:	The photographer sends a request to add a busy day.		
Description:	This use case allows photographers to add their busy day.		
Preconditions:	The User has been authenticated as a Photographer.		
Post-conditions:	<p>Success: New busy days will be stored in the system.</p> <p>Fail: System outputs error message.</p>		
Normal Flow:	Step	Actor Action	System Response
	1	Photographer goes on a busy day adding view.	<p>System requires information from photographer:</p> <ul style="list-style-type: none"> - “Tiêu đề”: text, required - “Mô tả”: text, required - “Ngày nghỉ”: datepicker, required

	2	Photographer inputs required information and sends requests to add busy days.	System outputs “Bạn muốn thêm ngày nghỉ?”
	3	Photographer confirm “Có” [Alternative 1]	System add busy day to system and outputs message “Đã thêm ngày nghỉ” [Exception 1] [Exception 2]
Alternative Flows:	Step	Actor Action	System Response
	1	Photographer confirms “Không”.	Back to a busy day adding views.
Exceptions:	No	Cause	System Response
	1	Photographer inputs data incorrectly or leave them empty	System outputs error message to notify photographer re-input.
	2	The process runs into an internal error	System outputs “Đã có lỗi xảy ra”
Priority:	Medium		
Frequency of Use:	Sometimes		
Business Rules:	<ul style="list-style-type: none"> - Busy day information contains: busy day note, busy datetime, description on busy day. - If the picked day has booking deal, system will return a message to warn photographer that they have to cancel the booking before set busy day. 		
Other Information:	N/A		
Assumptions:	N/A		

Table 32 <Photographer> Add busy day specification

e. <Photographer> Get customer's review

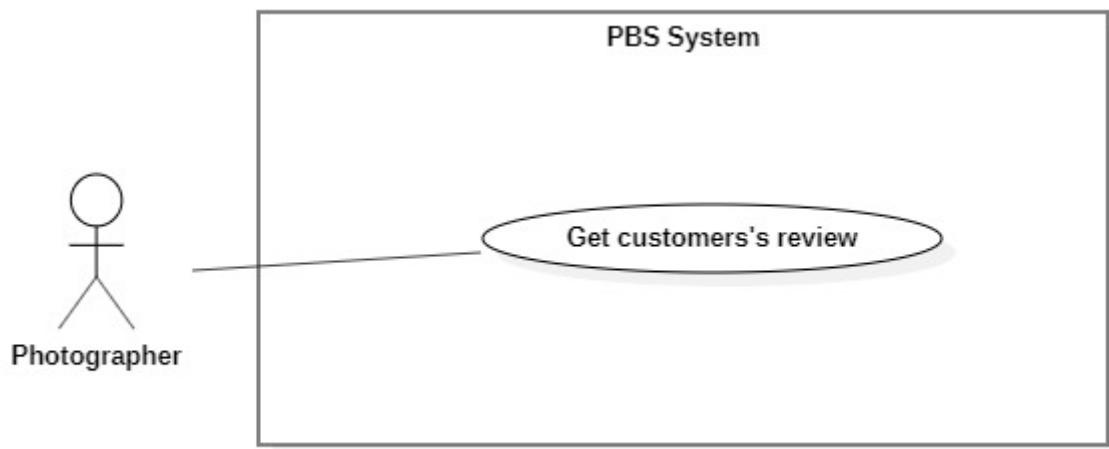


Figure 26 <Photographer> Get customer's review

UC ID and Name:	UC – 18 Get customer's review		
Created By:	Trương Ngọc Mỹ	Date Created:	08/10/2020
Primary Actor:	Photographer, Admin	Secondary Actors:	
Trigger:	The photographer or admin sends a request to get a customer's review.		
Description:	This use case allows photographers or admins to get customers' reviews.		
Preconditions:	The User has been authenticated as Photographer or Admin		
Post-conditions:	Success: The system returns customers' review of photographer Fail: System shows error message		
Normal Flow:	Step	Actor Action	System Response
	1	Photographer or Admin send requests to get their reviews from customers.	System displays list of reviews. [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:	<ul style="list-style-type: none"> - The results contain: customers' comments, customers' ratings, commented dates. 		

	<ul style="list-style-type: none"> - If the result is empty, returns empty text. - Admin can't delete customers's reviews
Other Information:	N/A
Assumptions:	N/A

Table 33 <Photographer> Get customer's review specification

f. <Photographer> Add package

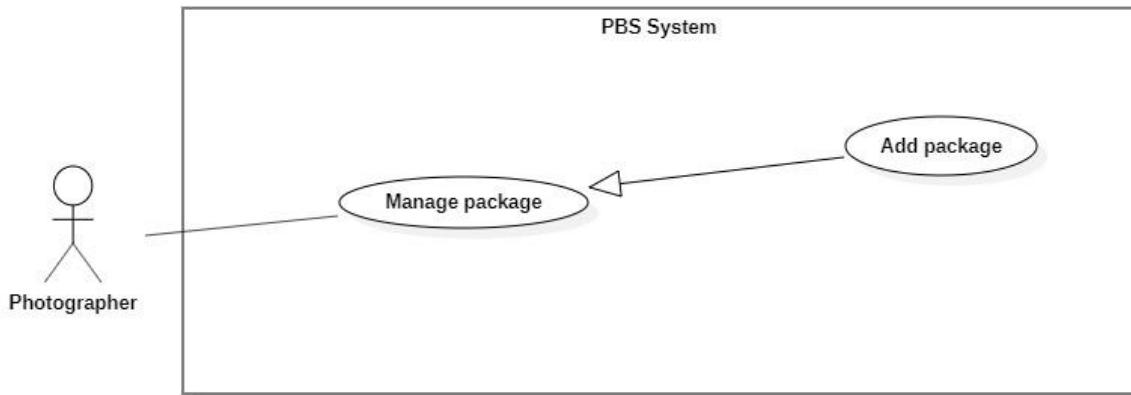


Figure 27 <Photographer> Add package

UC ID and Name:	UC – 19 Add package		
Created By:	Đào Sỹ Trung Kiên	Date Created:	08/10/2020
Primary Actor:	Photographer	Secondary Actors:	
Trigger:	Photographer sends a request to add a new package.		
Description:	This use case allows photographers to add new packages.		
Preconditions:	The User has been authenticated as Photographer		
Post-conditions:	<p>Success: Package will be stored with status active.</p> <p>Fail: System outputs error message and sends to Photographer.</p>		
Normal Flow:	Step	Actor Action	System Response
	1	Photographer goes to New Package View.	<p>System returns “phương thức giao hàng” for execution steps.</p> <p>System requires information from photographer:</p> <ul style="list-style-type: none"> - “Kiểu dịch vụ”: text, required.

			<ul style="list-style-type: none"> - “Tên dịch vụ”: text, required. - “Mô tả”: text, required. - “Giá thành dịch vụ”: text, required - “Thời gian tác nghiệp”: text, required - “Chi tiết dịch vụ”: text, required. - “Phương thức giao hàng”: drop list, required.
	2	Photographers input required information and send requests to create new packages.	<p>System validates information, adds new package to system, output message “Tạo gói dịch vụ thành công” and send to Photographer</p> <p>[Exception 1]</p> <p>[Exception 2]</p>
Alternative Flows:	N/A		
Exceptions:	No	Actor Action	System Response
	1	Photographer inputs invalid format text inputs or leaves empty required fields	System outputs error message and sends it to ask the photographer to input correct required fields.
	2	The create process encountered an internal error.	System outputs error message and sends to photographer and notify that create new package fails.
Priority:	Medium		
Frequency of Use:	Sometimes		
Business Rules:	<ul style="list-style-type: none"> - Package will be managed by the system with status active and deactivated. - Package information contains: Id, package category, package name, description, price, time anticipation, services, returning type, isSupportMultiDay. 		

	<ul style="list-style-type: none"> - Package id will be generated by the system. - Returning types and category option list drop returned by system. - IsSupportMultiDay is a boolean. - After creating a new package successfully, the new package will have status active and available for customers to choose.
Other Information:	N/A
Assumptions:	N/A

Table 34 <Photographer> Add package specification

g. <Photographer> Edit Package

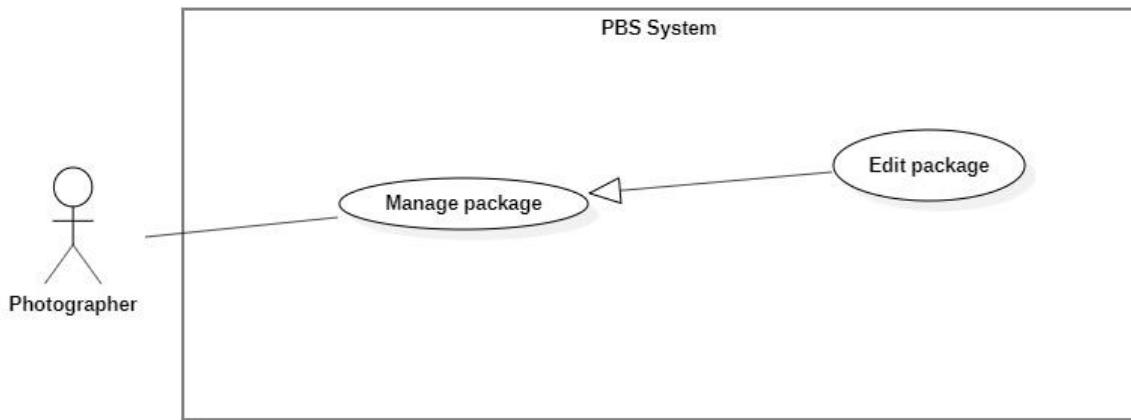


Figure 28 <Photographer> Edit package

UC ID and Name:	UC – 20 Edit package		
Created By:	Đào Sỹ Trung Kiên	Date Created:	08/10/2020
Primary Actor:	Photographer	Secondary Actors:	
Trigger:	Photographer sends request to edit package.		
Description:	This use case allows photographers to edit packages.		
Preconditions:	The User has been authenticated as Photographer		
Post-conditions:	<p>Success: Package is edited.</p> <p>Fail: System outputs error message and sends to Photographer.</p>		
Normal Flow:	No	Actor Action	System Response
	1	Photographers send requests to receive a list of packages.	System display list of packages. [Exception 1]
	2	Photographer sends request to receive package details	System returns “phương thức giao hàng” for execution steps.

			<p>System requires information from photographer:</p> <ul style="list-style-type: none"> - “Kiểu dịch vụ”: text, required. - “Tên dịch vụ”: text, required. - “Mô tả”: text, required. - “Giá thành dịch vụ”: text, required - “Thời gian tác nghiệp”: text, required - “Chi tiết dịch vụ”: text, required. - “Phương thức giao hàng”: drop list, required. <p>[Exception 2]</p>
	3	Photographer inputs package required information and send request to update package	<p>System validates information, updates package, sends output message: “Chỉnh sửa thành công” to photographer.</p> <p>[Exception 3]</p> <p>[Exception 4]</p>
Alternative Flows:	N/A		
Exceptions:	No	Actor Action	System Response
	1	System retrieves an empty list of available packages.	System outputs the error message “Không tìm thấy package” and sends it to the photographer.
	2	System retrieves an empty list of “Phương thức giao hàng”.	System outputs error message “Không tìm thấy phương thức giao hàng” and sends to photographer.
	3	Photographer inputs invalid format text inputs or leaves empty required fields.	System outputs error message and sends it to photographer to ask photographer to input correct required fields.

	4	Update package process encountered a system internal error.	System outputs error message to notify photographer that update package failed.
Priority:	Medium		
Frequency of Use:	Sometimes		
Business Rules:	<ul style="list-style-type: none"> - Photographers only access packages via package id. - Package will be updated in the system with inputted information. - Package id cannot be updated. - Package will be managed by the system with status active and deactivated. - Package information contains: Id, package category, package name, description, price, time anticipation, services, returning type, isSupportMultiDay. - IsSupportMultiDay is a boolean. - Returning type and category option list returned by system. - After updating the package successfully, the package will have new value and available for customers to choose. 		
Other Information:	N/A		
Assumptions:	N/A		

Table 35 <Photographer> Edit package specification

h. <Photographer> Remove Package

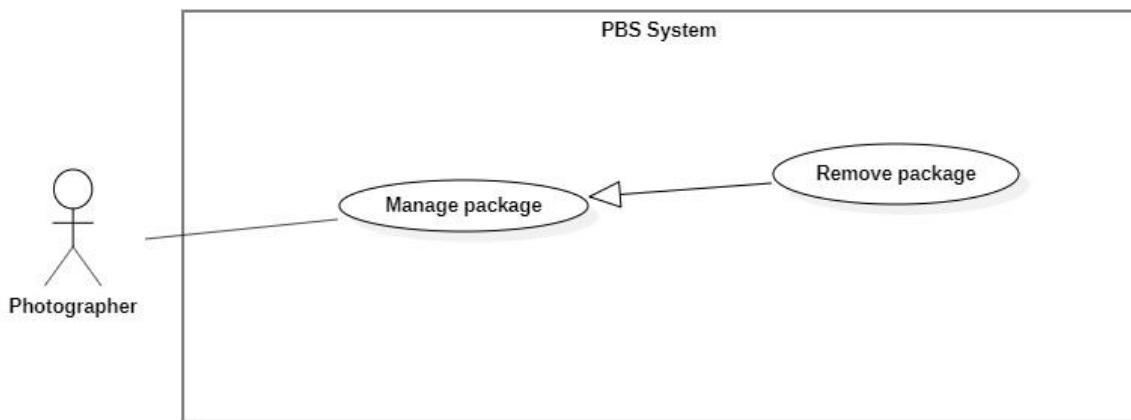


Figure 29 <Photographer> Remove package

UC ID and Name:	UC – 21 Remove package		
Created By:	Đào Sỹ Trung Kiên	Date Created:	08/10/2020
Primary Actor:	Photographer	Secondary Actors:	

Trigger:	Photographer sends request to delete package.		
Description:	This use case allows photographers to delete packages.		
Preconditions:	The User has been authenticated as Photographer		
Post-conditions:	<p>Success: Package is set to deactivated.</p> <p>Fail: System outputs error message and sends to Photographer.</p>		
Normal Flow:	Step	Actor Action	System Response
	1	Photographer sends a request to get a list package.	System returns and displays list package. [Exception 1]
	2	Photographer sends request to delete package.	System outputs confirm messages: “Bạn có muốn xóa gói dịch vụ này không ?” and requires Photographer to confirm.
Alternative Flows:	2	Photographer confirms “Đồng ý” to delete the package.	System outputs and sends message: “Xóa gói dịch vụ thành công” to photographer and updates list package. [Exception 2]
	N/A		
	No	Actor Action	System Response
Exceptions:	1	System retrieves an empty list package.	System outputs error message “Không có gói dịch vụ nào” and sends to photographer.
	2	The delete action process encountered an internal error.	System outputs error message and sends it to photographer to notify that the delete package failed.
Priority:	Medium		
Frequency of Use:	Sometimes		
Business Rules:	<ul style="list-style-type: none"> - Photographers only access packages via package id. - If package status is active: Update status to deactivate. 		

	- Package which currently has been booking can't not be removed.
Other Information:	N/A
Assumptions:	N/A

Table 36 <Photographer> Remove package specification

i. <Photographer> Add album

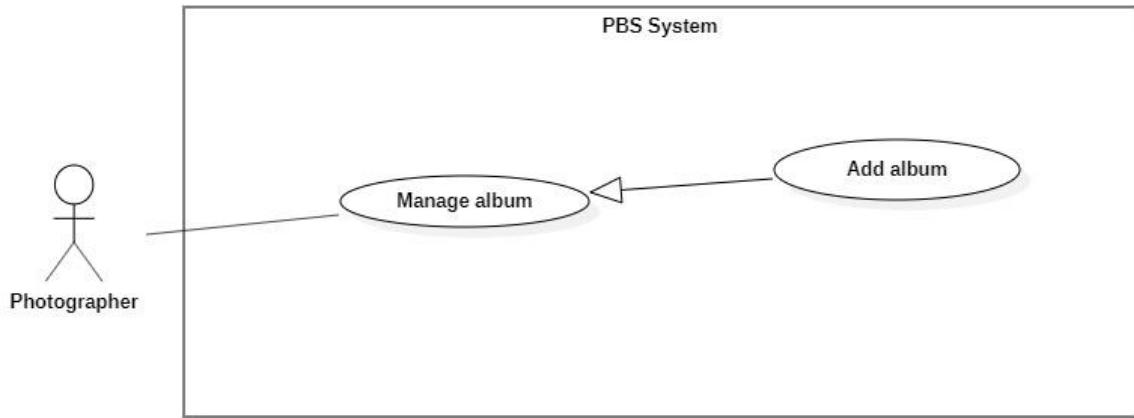


Figure 30 <Photographer> Add album

UC ID and Name:	UC – 22 Add album		
Created By:	Đào Sỹ Trung Kiên	Date Created:	08/10/2020
Primary Actor:	Photographer	Secondary Actors:	
Trigger:	The photographer sends request to add new album		
Description:	This use case allows Photographer to add new album		
Preconditions:	The User has been authenticated as Photographer		
Post-conditions:	<p>Success: Album will be added successfully</p> <p>Fail: Systems outputs error message and sends to Photographer</p>		
Normal Flow:	Step	Actor Action	System Response
	1	The Photographer goes to new Package View	<p>System returns “category” for execution steps.</p> <p>System requires information from photographer:</p> <ul style="list-style-type: none"> - “Ngày tạo”: datetime, required. - “Mô tả”: text, required.

		<ul style="list-style-type: none"> - “Tên album”: text, required. - “Hình ảnh”: picture, required. - “Category”: chips, required - “Địa điểm”: text, required. 	
	2	<p>Photographer inputs required information and send request to create new album</p>	<p>System validates information, adds new album to the system, outputs message “Tạo mới album thành công” and sends to the Photographer</p> <p>[Exception 1]</p> <p>[Exception 2]</p>
Alternative Flows:	N/A		
Exceptions:	No	Actor Action	System Response
	1	Photographer inputs invalid format text inputs or leaves empty required fields.	System outputs error message and sends to ask Photographer to input correct required fields.
	2	The create process encountered an internal error.	System outputs error message and sends to the photographer to notify that create new album failed.
Priority:	Medium		
Frequency of Use:	Sometimes		
Business Rules:	<ul style="list-style-type: none"> - Album will be managed by the system with status active and deactivated. - Album information contains: id, created date, description, album's name, images, category. - Album id will be generated by the system. - Category option list returned by system. - After creating new album successfully, new album will have status active and available for customers. 		

Other Information:	N/A
Assumptions:	N/A

Table 37 <Photographer> Add album specification

j. <Photographer> Edit album

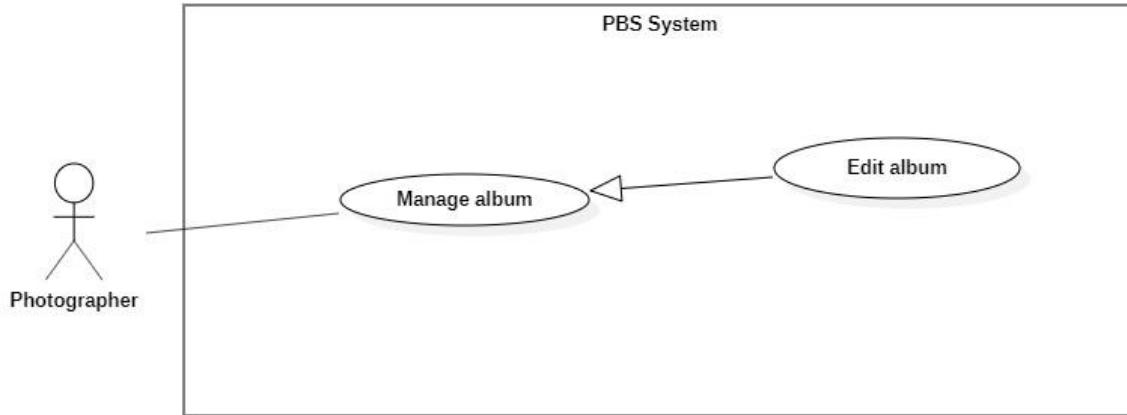


Figure 31 <Photographer> Edit album

UC ID and Name:	UC – 23 Edit album		
Created By:	Đào Sỹ Trung Kiên	Date Created:	08/10/2020
Primary Actor:	Photographer	Secondary Actors:	
Trigger:	The photographer sends request to edit album		
Description:	This use case allows Photographer to edit album		
Preconditions:	The User has been authenticated as Photographer		
Post-conditions:	<p>Success: Album will be updated</p> <p>Fail: Systems outputs error message and sends to Photographer</p>		
Normal Flow:	No	Actor Action	System Response
	1	Photographers send requests to receive a list of albums.	System display list of albums. [Exception 1]
	2	Photographer sends request to receive album details	System returns “category” for execution steps. System requires information from photographer:

			<ul style="list-style-type: none"> - “Ngày tạo”: datetime, required. - “Mô tả”: text, required. - “Tên album”: text, required. - “Hình ảnh”: picture, required. - “Category”: chips, required - “Địa điểm”: text, required. <p>[Exception 2]</p>
	3	Photographer inputs album required information and send request to update album	<p>System validates information, updates package, sends output message: “Chỉnh sửa thành công” to photographer.</p> <p>[Exception 3]</p> <p>[Exception 4]</p>
Alternative Flows:	N/A		
Exceptions:	No	Actor Action	System Response
	1	System retrieves an empty list of available albums.	System outputs the error message “Không tìm thấy album” and sends it to the photographer.
	2	System retrieves an empty list of “category”.	System outputs the error message “Không tìm thấy category” and sends it to the photographer.
	3	Photographer inputs invalid format text inputs or leaves empty required fields.	System outputs error message and sends it to photographer to ask photographer to input correct required fields.
	4	Update album process encountered a system internal error.	System outputs error message to notify

			photographer that update album failed.
Priority:	Medium		
Frequency of Use:	Sometimes		
Business Rules:	<ul style="list-style-type: none"> - Photographers only access albums by album id. - Albums will be updated in the system with inputted information. - Album id cannot be updated. - Albums will be managed by the system with status active and deactivated. - Album information contains: id, created date, description, album's name, images, category. - Category option list returned by system. - After updating the album successfully, the album will have new value and be available for customers. 		
Other Information:	N/A		
Assumptions:	N/A		

Table 38 <Photographer> Edit album specification

k. <Photographer> Remove album

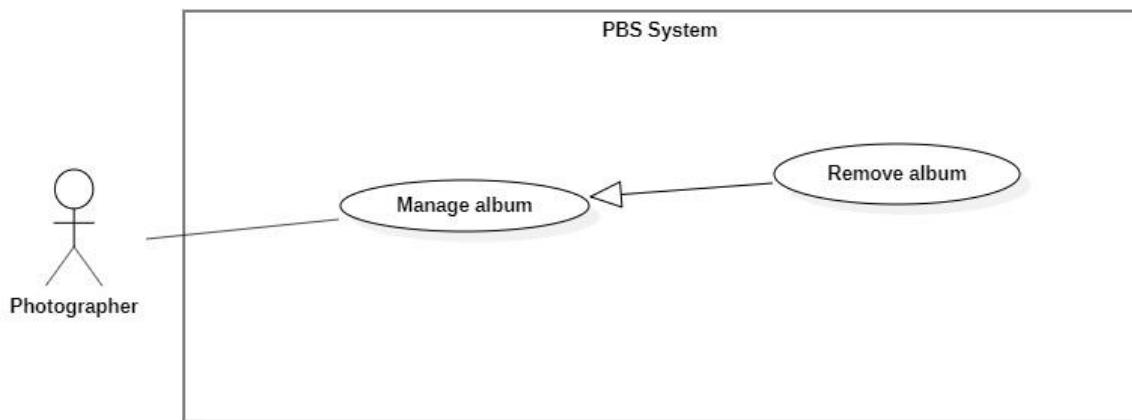


Figure 32 <Photographer> Remove album

UC ID and Name:	UC – 24 Remove album		
Created By:	Đào Sỹ Trung Kiên	Date Created:	08/10/2020
Primary Actor:	Photographer	Secondary Actors:	
Trigger:	Photographer sends request to delete album		
Description:	This use case allows photographers to delete albums.		
Preconditions:	The User has been authenticated as Photographer		

Post-conditions:	Success: Album is set to deactivated. Fail: System outputs error message and sends to Photographer.		
Normal Flow:	Step	Actor Action	System Response
	1	Photographer sends request to get list album.	System returns and displays the list album. [Exception 1]
	2	Photographer sends request to delete album.	System outputs confirm messages: “Bạn có muốn xóa album này không ?” and requires Photographer to confirm.
Alternative Flows:	N/A		
	No	Actor Action	System Response
	1	System retrieves an empty list album.	System outputs the error message “Không có album nào” and sends it to the photographer.
Exceptions:	2	The delete action process encountered an internal error.	System outputs error message and sends it to photographer to notify that the delete album failed.
	Medium		
	Sometimes		
Business Rules:	<ul style="list-style-type: none"> - Photographers only access albums via album id. - If album status is active: Update status to deactivate. 		
Other Information:	N/A		
Assumptions:	N/A		

Table 39 <Photographer> Remove album specification

2.2.4 <Authenticated User> Overview Use Case

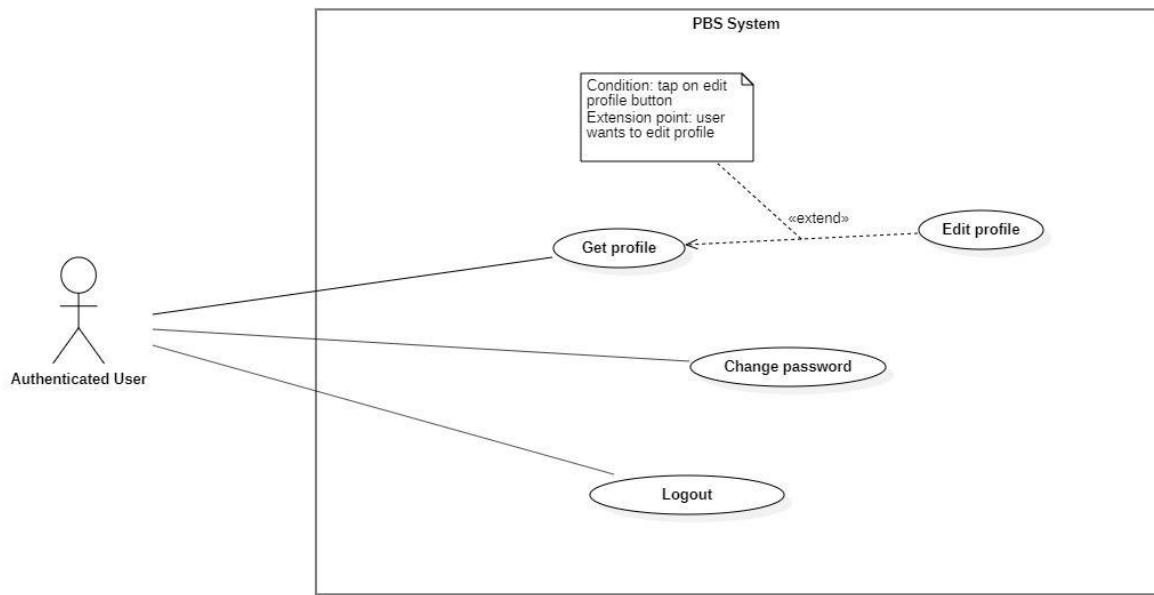


Figure 33 <Authenticated User> Overview

a. <Authenticated User> Get profile

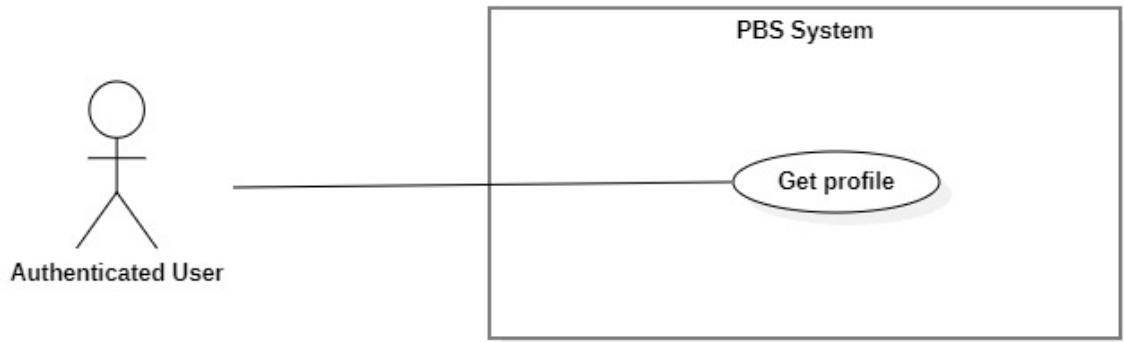


Figure 34 <Authenticated User> Get profile

UC ID and Name:	UC – 25 Get profile		
Created By:	Trương Ngọc Mỹ	Date Created:	08/10/2020
Primary Actor:	Authenticated User	Secondary Actors:	
Trigger:	The Authenticated User sends request to get profile		
Description:	This use case allows authenticated user to get their profile		
Preconditions:	User has been authenticated.		
Post-conditions:	Success: System returns authenticated user's profile. Fail: System shows error message.		

Normal Flow:	Step	Actor Action	System Response
	1	Authenticated users send requests to get their profile.	System outputs the result to the authenticated user. [Exception 1] [Exception 2]
Alternative Flows:	N/A		
Exceptions:	No	Cause	System Response
	1	System can't get the profile.	System outputs error message "Không thể tìm thấy thông tin".
	2	The process runs into an internal error.	System outputs error message "Đã có lỗi xảy ra"
Priority:	Medium		
Frequency of Use:	Sometimes		
Business Rules:	<ul style="list-style-type: none"> The results contain following information: user's avatar, user' full name, user's phone, user's email, user's description. 		
Other Information:	N/A		
Assumptions:	N/A		

Table 40 <Authenticated User> Get profile specification

b. <Authenticated User> Edit profile

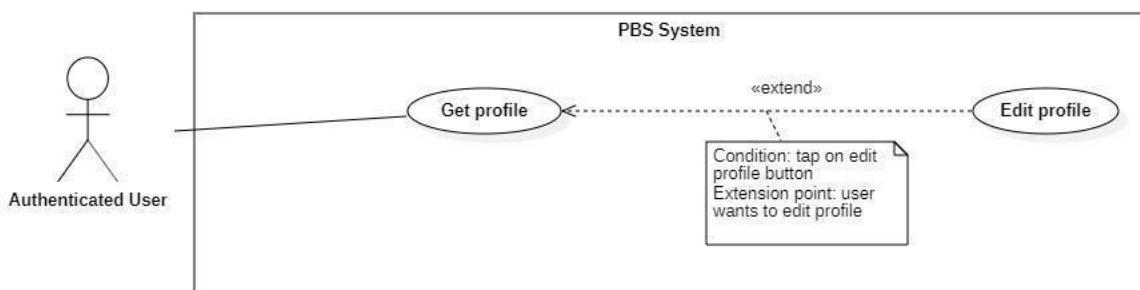


Figure 35 <Authenticated User> Edit profile

UC ID and Name:	UC – 26 Edit profile		
Created By:	Trương Ngọc Mỹ	Date Created:	08/10/2020

Primary Actor:	Authenticated User	Secondary Actors:	
Trigger:	The Authenticated User sends a request to edit the profile.		
Description:	This use case allows authenticated users to edit their profile.		
Preconditions:	User has been authenticated.		
Post-conditions:	<p>Success: The profile will be updated into the system.</p> <p>Fail: System outputs error message.</p>		
Normal Flow:	Step	Actor Action	System Response
	1	Authenticated users send requests to receive their profile.	<p>System return results and requires information from user:</p> <ul style="list-style-type: none"> - “Họ tên”: text, required - “Email”: text, required - “Số điện thoại”: number, required - “Mô tả”: text - “Ảnh đại diện”: image <p>[Exception 1]</p>
	2	Authenticated user input information and send requests to edit their profile.	System outputs “Bạn muốn cập nhật hồ sơ?”.
	3	Authenticated confirmation “Có”. [Alternative 1]	<p>System update data and outputs “Đã cập nhật hồ sơ”.</p> <p>[Exception 2]</p> <p>[Exception 3]</p>
Alternative Flows:	Step	Actor Action	System Response
	1	Authenticated confirm “Không”.	Back to profile view.
Exceptions:	No	Cause	System Response
	1	System can't get user's profile	System outputs “Không thể tìm thấy hồ sơ”
	2	User inputs information incorrectly or leaves it empty in the required field.	System outputs error messages to get user input correctly.

	3	System runs into an internal error.	System outputs error message “Đã có lỗi xảy ra”
Priority:	Medium		
Frequency of Use:	Sometimes		
Business Rules:	<ul style="list-style-type: none"> The return view contains following information: user's avatar, user's full name, user's phone, user's email, user's description User can view every information in their profile. 		
Other Information:	N/A		
Assumptions:	N/A		

Table 41 <Authenticated User> Edit profile specification

c. <Authenticated User> Change password

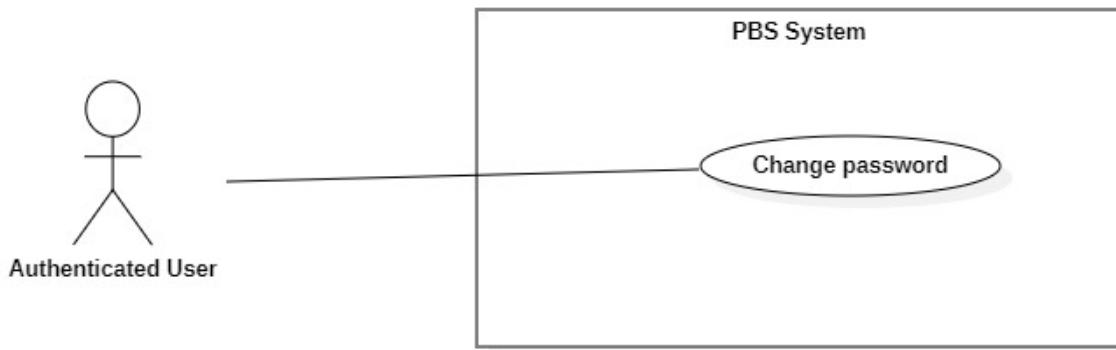


Figure 36 <Authenticated User> Change password

UC ID and Name:	UC – 27 Change password		
Created By:	Bồ Công Đạt	Date Created:	08/10/2020
Primary Actor:	Authenticated User	Secondary Actors:	
Trigger:	The Authenticated User sends a request to change password.		
Description:	This use case allows authenticated users to change their password.		
Preconditions:	User has been authenticated.		
Post-conditions:	<p>Success: User's password updated successfully in the system.</p> <p>Fail: System outputs error message.</p>		
Normal Flow:	Step	Actor Action	System Response
	1	Authenticated user sends request to get their password	System returns result and requires following information from user:

			<ul style="list-style-type: none"> - “Mật khẩu cũ”: password, required - “Mật khẩu mới”: password, required - “Nhập lại mật khẩu”: password, required <p>[Exception 1]</p>
	2	Authenticated user inputs required information and sends request to change password	System outputs “Bạn muốn cập nhật mật khẩu?”
	3	Authenticated user confirms “Có” [Alternative 1]	<ul style="list-style-type: none"> System update data and output “Cập nhật mật khẩu thành công” [Exception 2] [Exception 3]
Alternative Flows:	Step	Actor Action	System Response
	1	Authenticated user confirms “Không”	Back to password changing view
Exceptions:	No	Cause	System Response
	1	System can't get user's password	System outputs “Không thể tìm thấy thông tin”
	2	User inputs information incorrectly or leave required fields empty	System outputs error message to get user input information correctly
	3	The process runs into an internal error	System outputs “Đã có lỗi xảy ra”
Priority:	Medium		
Frequency of Use:	Sometimes		
Business Rules:	<ul style="list-style-type: none"> - User must input the current password before changing new password. - Password format should be a minimum eight characters, contains at least one number, one letter and one special character. - Save password using SHA256 hash function. 		
Other Information:	N/A		
Assumptions:	N/A		

Table 42 <Authenticated User> Change password specification

d. <Authenticated User> Logout

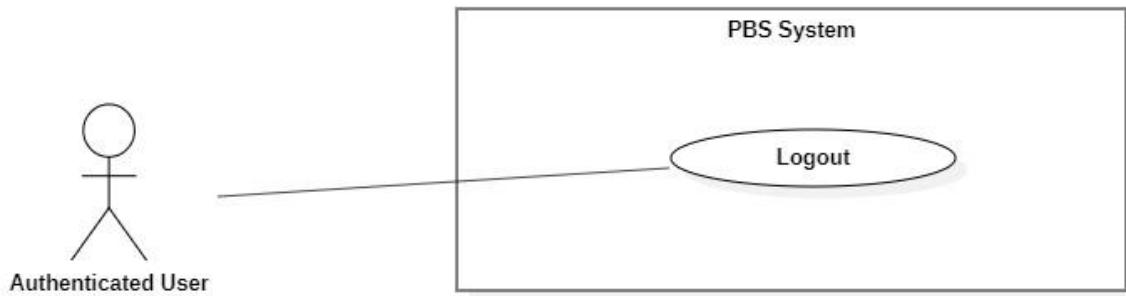


Figure 37 <Authenticated User> Logout

UC ID and Name:	UC – 28 Logout				
Created By:	Bồ Công Đạt	Date Created:	08/10/2020		
Primary Actor:	Authenticated User	Secondary Actors:			
Trigger:	The Authenticated User sends a request to log out from the system.				
Description:	This use case allows users to log out from the system.				
Preconditions:	User has been authenticated.				
Post-conditions:	Success: User logged out from the system. Fail: System outputs error message				
Normal Flow:	Step	Actor Action	System Response		
	1	User sends a request to log out from the system.	System outputs “Bạn muốn đăng xuất?”		
	2	User confirms “Có” [Alternative 1]	System logs the user out. [Exception 1]		
Alternative Flows:	Step	Actor Action	System Response		
	1	User confirms “Không”	Back to logout view		
	No	Cause	System Response		
	1	The process runs into an internal error	System outputs “Đã có lỗi xảy ra”		
	Priority:	Medium			
Frequency of Use:	Usually				
Business Rules:	<ul style="list-style-type: none"> - User's Json web token will be removed on client side. - User's session will be destroyed on client side. 				
Other Information:	N/A				

Assumptions:	N/A
--------------	-----

Table 43 <Authenticated User> Logout specification

2.2.5 <Admin> Overview Use Case



Figure 38 <Admin> Overview

a. <Admin> Block user

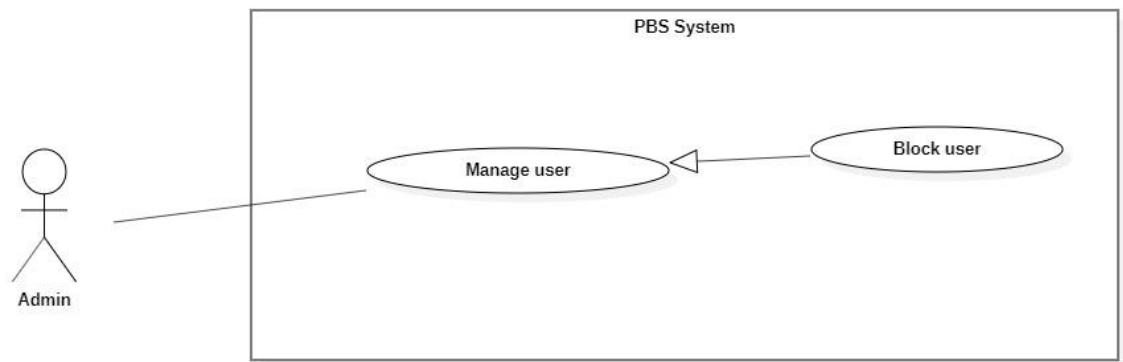


Figure 39 <Admin> Block user

UC ID and Name:	UC – 29 Block user		
Created By:	Đào Sỹ Trung Kiên	Date Created:	08/10/2020
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:	<p>Success: User will be blocked and unable to access the system.</p> <p>Fail: Systems outputs error message and sends to Admin</p>		
Normal Flow:	Step	Actor Action	System Response
	1	The Admin goes to block user view	System requires information from admin: - "User Id":text, required.
	2	Admin inputs required information and send request to block user	System validates information, block user and outputs message "Chặn người dùng thành công" 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 block user failed.
Priority:	Medium		
Frequency of Use:	Sometimes		
Business Rules:	<ul style="list-style-type: none"> - Users will be managed by the system with status active and deactivated. - Block user if their cancellation above 40%. If their cancellation rate is lower than 40%, warn the admin. - Cancellation rate = number of cancelled bookings / total of bookings. - Save the admin's name, admin's id and the datetime of blocking to log file. - Blocked user will have status deactivated. - User information contains: id. - After successfully blocking the user, log the user out if they are logged in. 		
Other Information:	N/A		
Assumptions:	N/A		

Table 44 <Admin> Block user specification

b. <Admin> Unblock user

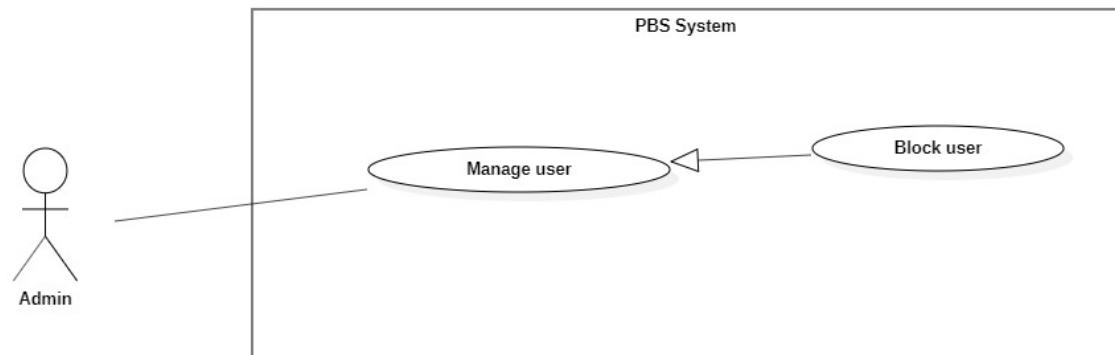


Figure 40 <Admin> Unblock user

UC ID and Name:	UC – 30 Unblock user		
Created By:	Đào Sỹ Trung Kiên	Date Created:	08/10/2020
Primary Actor:	Admin	Secondary Actors:	
Trigger:	The Admin sends request to unblock album		
Description:	This use case allows Admin to unblock user		
Preconditions:	The User has been authenticated as Admin		
Post-conditions:	<p>Success: User will be unblocked and enabled to access the system.</p> <p>Fail: Systems outputs error message and sends to Admin</p>		
Normal Flow:	Step	Actor Action	System Response
	1	The Admin goes to unblock user view	System requires information from admin: - "User Id":text, required.
	2	Admin inputs required information and send request to unblock user	System validates information, unblock user and outputs message “Gỡ chặn người dùng thành công” 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 unblock user failed.
Priority:	Medium		
Frequency of Use:	Sometimes		
Business Rules:	<ul style="list-style-type: none"> Users will be managed by the system with status active and deactivated. 		

	<ul style="list-style-type: none"> - After unblock user will have status change from deactivated to active. - User information contains: id. - After unblocking the user successfully, this user will be enabled to access the system.
Other Information:	N/A
Assumptions:	N/A

Table 45 <Admin> Unblock user specification

c. <Admin> Add Category

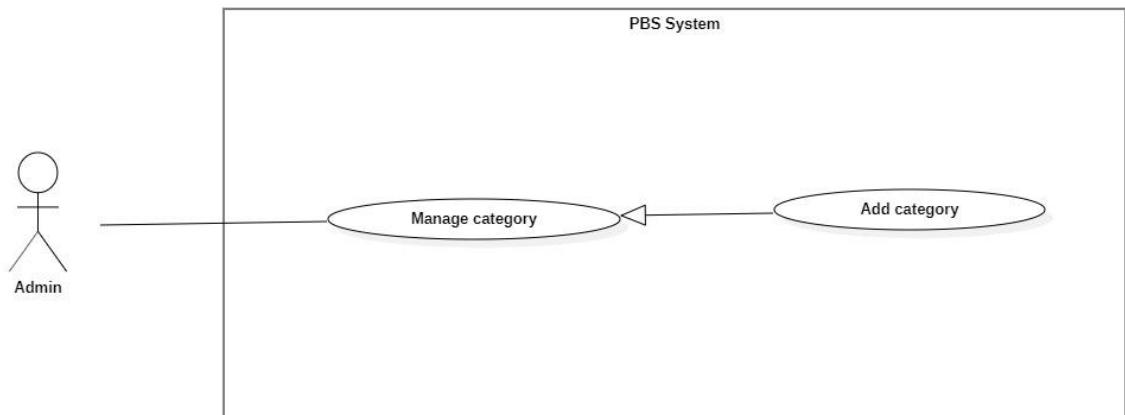


Figure 41 <Admin> Add category

UC ID and Name:	UC – 31 Add category		
Created By:	Đào Sỹ Trung Kiên	Date Created:	08/10/2020
Primary Actor:	Admin	Secondary Actors:	
Trigger:	The admin sends request to add new category		
Description:	This use case allows Admin to add new category		
Preconditions:	The User has been authenticated as Admin		
Post-conditions:	<p>Success: Category will be added successfully</p> <p>Fail: Systems outputs error message and sends to Admin</p>		
Normal Flow:	Step	Actor Action	System Response
	1	The Admin goes to new Category View	System requires information from admin: - "Tên category": text, required.
	2	Admin inputs required information and send request to create category	System validates information, adds new

			category to the system, outputs message “Tạo mới category thành công” 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.
Priority:	Medium		
	Sometimes		
Business Rules:	<ul style="list-style-type: none"> - Category will be managed by the system with status active and deactivated. - Category information contains: id, category name. - Category id will be generated by the system. - After creating a new category successfully, the new category will have status active and available for customer and photographer. 		
Other Information:	N/A		
Assumptions:	N/A		

Table 46 <Admin> Add category specification

d. <Admin> Edit category

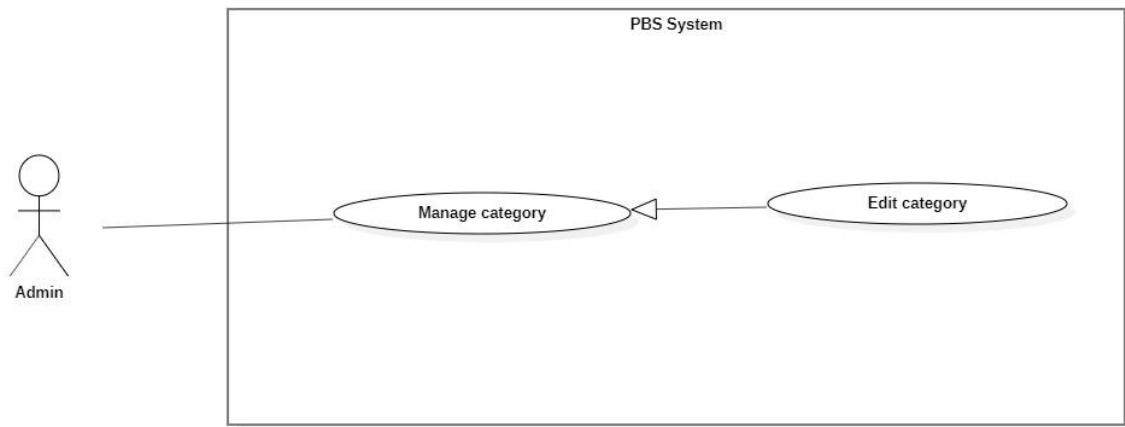


Figure 42 <Admin> Edit category

UC ID and Name:	UC – 32 Edit category		
Created By:	Đào Sỹ Trung Kiên	Date Created:	08/10/2020
Primary Actor:	Admin	Secondary Actors:	
Trigger:	The admin sends request to edit category		
Description:	This use case allows Admin to edit category		
Preconditions:	The User has been authenticated as Admin		
Post-conditions:	<p>Success: Category will be updated</p> <p>Fail: Systems outputs error message and sends to Admin</p>		
Normal Flow:	No	Actor Action	System Response
	1	Admin sends a request to receive a list of categories.	System display list of categories. [Exception 1]
	2	Admin sends request to receive category details	System requires information from admin: - “Tên category”: text, required.
	3	Admin inputs category required information and send request to update category	System validates information, updates category, sends output message: “Chỉnh sửa thành công” to admin. [Exception 2] [Exception 3]

Alternative Flows:	N/A		
Exceptions:	No	Actor Action	System Response
	1	System retrieves an empty list of available categories.	System outputs the error message “Không tìm thấy category” and sends it to the admin.
	2	Admin inputs invalid format text inputs or leaves empty required fields.	System outputs error message and send to ask admin to input correct required fields
	3	Update category process encountered a system internal error.	System outputs error message to notify admin that update category failed.
Priority:	Medium		
Frequency of Use:	Sometimes		
Business Rules:	<ul style="list-style-type: none"> - Admin only access category by category id. - Category will be updated in the system with inputted information. - Category id cannot be updated. - Category will be managed by the system with status active and deactivated. - Category information contains: id, category name. - After updating Category successfully, category will have new value and available for customer and photographer. 		
Other Information:	N/A		
Assumptions:	N/A		

Table 47 <Admin> Edit category specification

e. <Admin> Remove category

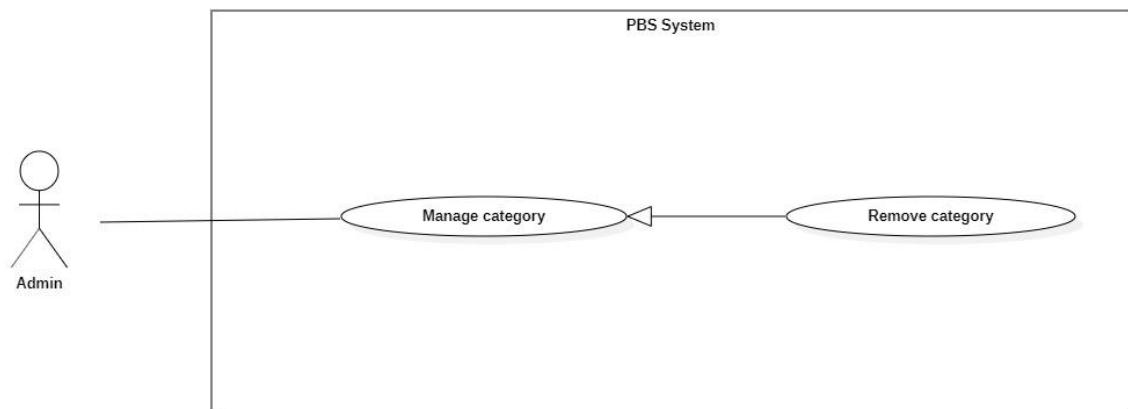


Figure 43 <Admin> Remove category

UC ID and Name:	UC – 33 Remove category		
Created By:	Đào Sỹ Trung Kiên	Date Created:	08/10/2020
Primary Actor:	Admin	Secondary Actors:	
Trigger:	Admin sends a request to delete the category.		
Description:	This use case allows the admin to delete categories.		
Preconditions:	The User has been authenticated as Admin		
Post-conditions:	<p>Success: Category is set to deactivated.</p> <p>Fail: System outputs error message and sends to Admin.</p>		
Normal Flow:	Step	Actor Action	System Response
	1	Admin sends a request to get a list category.	System returns and displays list category. [Exception 1]
	2	Admin sends a request to delete category.	System outputs confirm messages: “Bạn có muốn xóa category này không ?” and requires Admin to confirm.
	3	Admin confirms “Đồng ý” to delete the category.	System outputs and sends a message: “Xóa category thành công” to the admin and updates list category. [Exception 2]
Alternative Flows:	N/A		

Exceptions:	No	Actor Action	System Response
	1	System retrieves an empty list category.	System outputs the error message “Không có category nào” and sends it to the admin.
	2	The delete action process encountered an internal error.	System outputs error message and sends it to admin to notify that delete category failed.
Priority:	Medium		
Frequency of Use:	Sometimes		
Business Rules:	<ul style="list-style-type: none"> - Admin only access category by category id. - If category status is active: Update status to deactivate. - Customers and photographers will not use this category. 		
Other Information:	N/A		
Assumptions:	N/A		

Table 48 <Admin> Remove category specification

f. <Admin> Add returning type

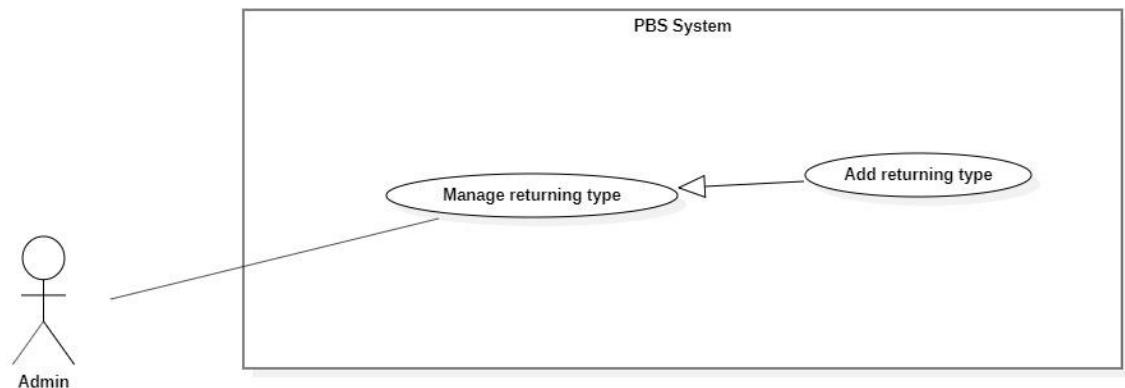


Figure 44 <Admin> Add returning type

UC ID and Name:	UC – 34 Add returning type		
Created By:	Đào Sỹ Trung Kiên	Date Created:	08/10/2020
Primary Actor:	Admin	Secondary Actors:	
Trigger:	The admin sends request to add new returning type		
Description:	This use case allows Admin to add new returning type		
Preconditions:	The User has been authenticated as Admin		

Post-conditions:	Success: Returning type will be added successfully Fail: Systems outputs error message and sends to Admin		
Normal Flow:	Step	Actor Action	System Response
	1	The Admin goes to new Returning type View	System requires information from admin: - “Tên returning type”: text, required.
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.
Priority:	Medium		
Frequency of Use:	Sometimes		
Business Rules:	<ul style="list-style-type: none"> - Returning type will be managed by the system with status active and deactivated. - Returning type information contains: id, returning type name. - Returning type id will be generated by the system. - After creating a new returning type successfully, the new returning type will have status active and available for customer and photographer. 		

Other Information:	N/A
Assumptions:	N/A

Table 49 <Admin> Add returning type specification

g. <Admin> Edit returning type

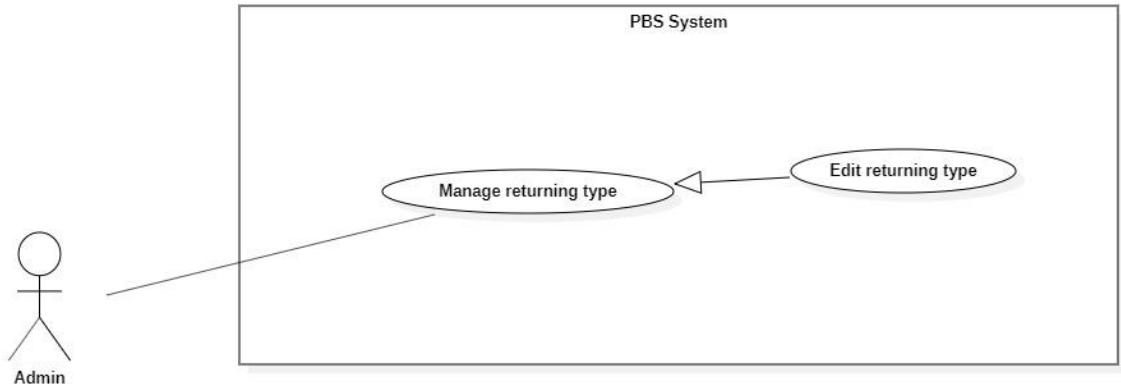


Figure 45 <Admin> Edit returning type

UC ID and Name:	UC – 35 Edit returning type		
Created By:	Đào Sỹ Trung Kiên	Date Created:	08/10/2020
Primary Actor:	Admin	Secondary Actors:	
Trigger:	The admin sends request to edit returning type		
Description:	This use case allows Admin to edit returning type		
Preconditions:	The User has been authenticated as Admin		
Post-conditions:	Success: Returning type will be updated Fail: Systems outputs error message and sends to Admin		
Normal Flow:	No	Actor Action	System Response
	1	Admin sends a request to receive a list of returning types.	System display list of returning type. [Exception 1]
	2	Admin sends request to receive returning type details	System requires information from admin: - “Tên returning type”: text, required.
	3	Admin inputs returning type required information and send request to update returning type	System validates information, updates returning type, sends output message: “Chỉnh

			sửa thành công" to admin. [Exception 2] [Exception 3]
Alternative Flows:	N/A		
Exceptions:	No	Actor Action	System Response
	1	System retrieves an empty list of available returning types.	System outputs the error message "Không tìm thấy returning type" and sends it to the admin.
	2	Admin inputs invalid format text inputs or leaves empty required fields.	System outputs error message and send to ask admin to input correct required fields
	3	Update returning type process encountered a system internal error.	System outputs error message to notify admin that update returning type failed.
Priority:	Medium		
Frequency of Use:	Sometimes		
Business Rules:	<ul style="list-style-type: none"> - Admin only access returning type via returning type id. - Returning type will be updated in the system with inputted information. - Returning type id cannot be updated. - Returning type will be managed by the system with status active and deactivated. - Returning type information contains: id, returning type name. - After updating the returning type successfully, the returning type will have new value and available for customer and photographer. 		
Other Information:	N/A		
Assumptions:	N/A		

Table 50 <Admin> Edit returning type specification

h. <Admin> Remove returning type

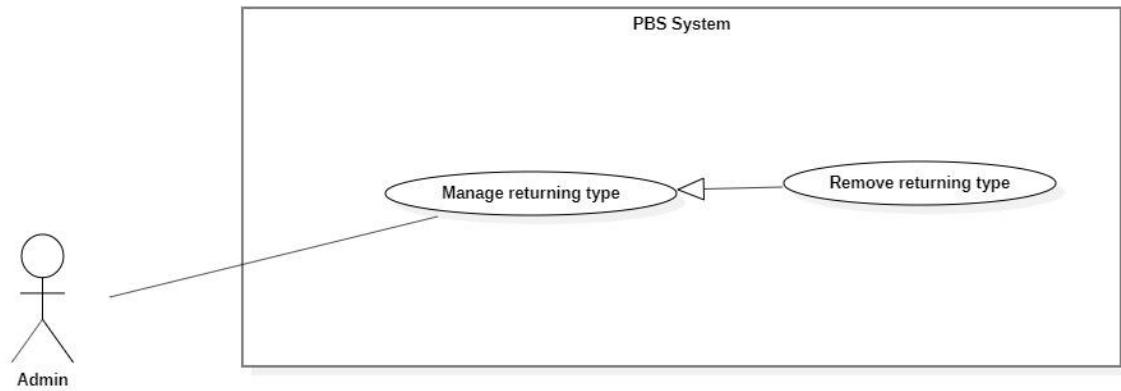


Figure 46 <Admin> Remove returning type

UC ID and Name:	UC – 36 Remove returning type		
Created By:	Đào Sỹ Trung Kiên	Date Created:	08/10/2020
Primary Actor:	Admin	Secondary Actors:	
Trigger:	Admin sends a request to delete the returning type.		
Description:	This use case allows admin to delete returning type.		
Preconditions:	The User has been authenticated as Admin		
Post-conditions:	<p>Success: Returning type is set to deactivated.</p> <p>Fail: System outputs error message and sends to Admin.</p>		
Normal Flow:	Step	Actor Action	System Response
	1	Admin sends a request to get a list returning type.	System returns and displays list returning type. [Exception 1]
	2	Admin sends request to delete returning type	System outputs confirm messages: “Bạn có muốn xóa returning type này không ?” and requires Admin to confirm.
	3	Admin confirms “Đồng ý” to delete the returning type.	System outputs and sends a message: “Xóa returning type thành công” to admin and updates list returning type. [Exception 2]

Alternative Flows:	N/A		
Exceptions:	No	Actor Action	System Response
	1	System retrieves an empty list returning type.	System outputs the error message “Không có returning type nào” and sends it to the admin.
	2	The delete action process encountered an internal error.	System outputs error message and sends it to admin to notify that delete returning type failed.
Priority:	Medium		
Frequency of Use:	Sometimes		
Business Rules:	<ul style="list-style-type: none"> - Admin only access returning type via returning type id. - If returning type status is active: Update status to deactivate. - Customer and photographer will not use this returning type. 		
Other Information:	N/A		
Assumptions:	N/A		

Table 51 <Admin> Remove returning type specification

i. <Admin> Change variables

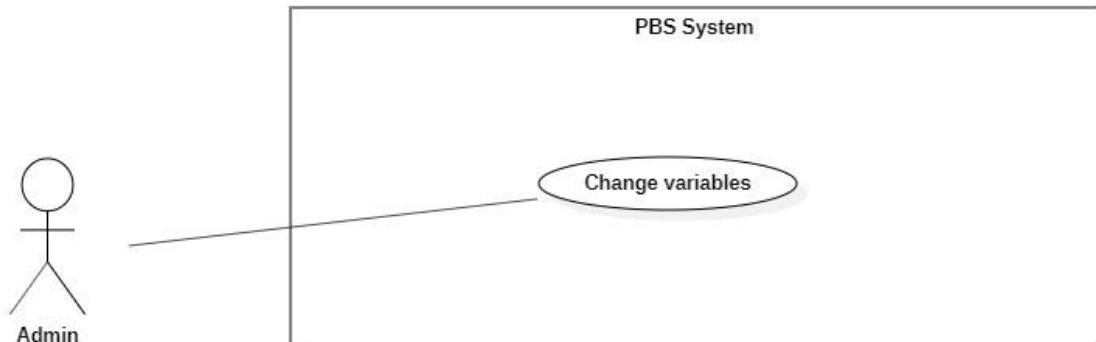


Figure 47 <Admin> Change variables

UC ID and Name:	UC – 37 Change variables		
Created By:	Đào Sỹ Trung Kiên	Date Created:	08/10/2020
Primary Actor:	Admin	Secondary Actors:	
Trigger:	The admin sends request to change variables		

Description:	This use case allows Admin to change variables		
Preconditions:	The User has been authenticated as Admin		
Post-conditions:	<p>Success: Variables will be updated</p> <p>Fail: Systems outputs error message and sends to Admin</p>		
Normal Flow:	No	Actor Action	System Response
	1	Admin sends request to receive variables details	<p>System requires information from admin:</p> <ul style="list-style-type: none"> - “Khoảng cách”: number, required. - “Trung bình giá dịch vụ”: number, required. - “Đánh giá từ người dùng”: number, required. <p>[Exception 1]</p>
	2	Admin inputs variables required information and send request to change variables	<p>System validates information, change variables, sends output message: “Chỉnh sửa thành công” to admin.</p> <p>[Exception 2]</p> <p>[Exception 3]</p>
Alternative Flows:	N/A		
Exceptions:	No	Actor Action	System Response
	1	System retrieves an empty variable.	System outputs error message “Không tìm thấy trọng số nào” and sends to admin.
	2	Admin inputs invalid format text inputs or leaves empty required fields.	System outputs error message and send to ask admin to input correct required fields
	3	Change variables process encountered a system internal error.	System outputs error message to notify admin that change variables failed.
Priority:	Medium		

Frequency of Use:	Sometimes
Business Rules:	<ul style="list-style-type: none"> - Variables will be updated in the system with inputted information. - Variables id cannot be updated. - Shooting category information contains: id, variable name. - After changing the variables category successfully, variables will have new value. - The sum of the variable is equal to 1
Other Information:	N/A
Assumptions:	N/A

Table 52 <Admin> Change variables specification

j. <Admin> Ban thread

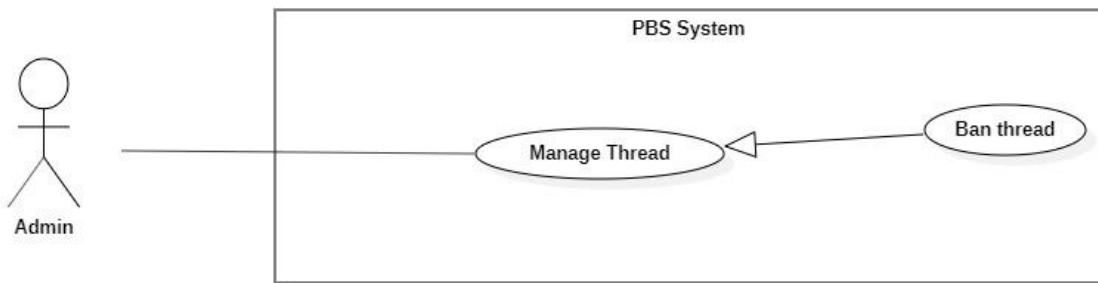


Figure 48 <Admin> Ban thread

UC ID and Name:	UC – 38 Ban thread								
Created By:	Bồ Công Đạt	Date Created:	08/10/2020						
Primary Actor:	Admin	Secondary Actors:							
Trigger:	The Admin sends request to ban thread								
Description:	This use case allows Admin to ban thread								
Preconditions:	The User has been authenticated as Admin								
Post-conditions:	<p>Success: Thread will be banned successfully.</p> <p>Fail: Systems outputs error message and sends to Admin</p>								
Normal Flow:	<table border="1"> <thead> <tr> <th>Step</th> <th>Actor Action</th> <th>System Response</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>The Admin goes to ban thread view.</td> <td>System requires information from admin: - “Thread id”: text, required.</td> </tr> </tbody> </table>	Step	Actor Action	System Response	1	The Admin goes to ban thread view.	System requires information from admin: - “Thread id”: text, required.		
Step	Actor Action	System Response							
1	The Admin goes to ban thread view.	System requires information from admin: - “Thread id”: text, required.							

	2	Admin inputs required information and send request to ban thread	System validates information, ban thread and outputs message “Chặn bài đăng thành công” 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 ban thread failed.
Priority:	Medium		
Frequency of Use:	Sometimes		
Business Rules:	<ul style="list-style-type: none"> - Thread will be managed by the system with status active and banned. - After ban thread, thread's status will be changed to banned. - Thread information contains: id. - After successfully ban the thread, save the admin name and time who have banned the thread to log file. 		
Other Information:	N/A		
Assumptions:	N/A		

Table 53 <Admin> Ban thread specification

2.2.6 <<system>>Handler Overview use case

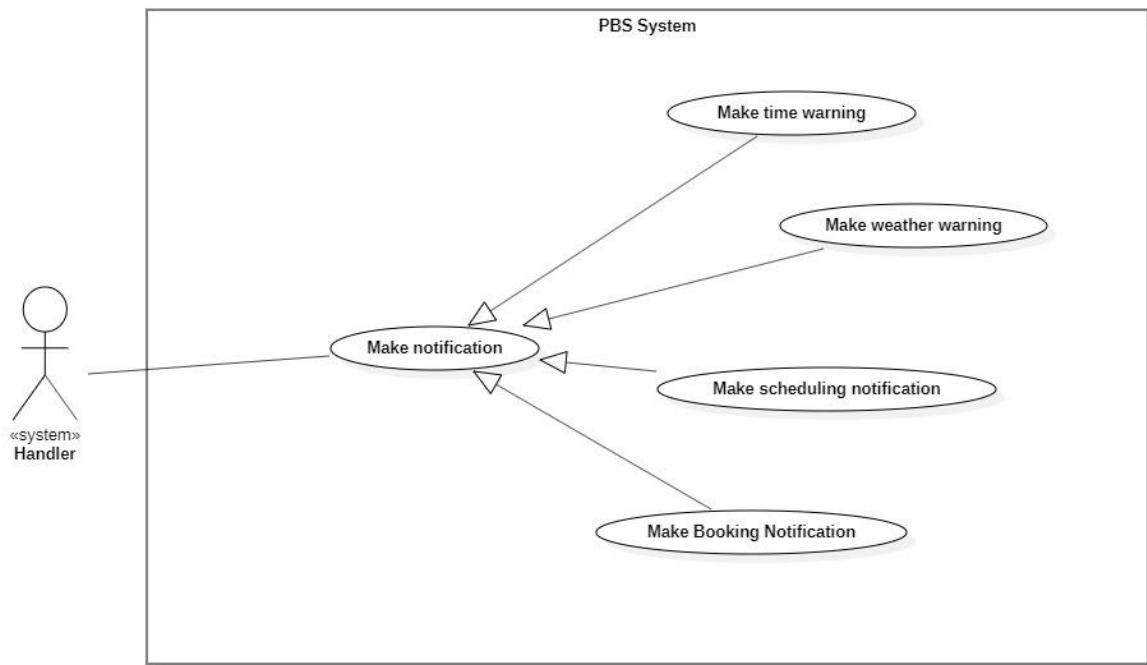


Figure 49 <<system>> Handler Overview

a. <<system>>Handler Make booking notification

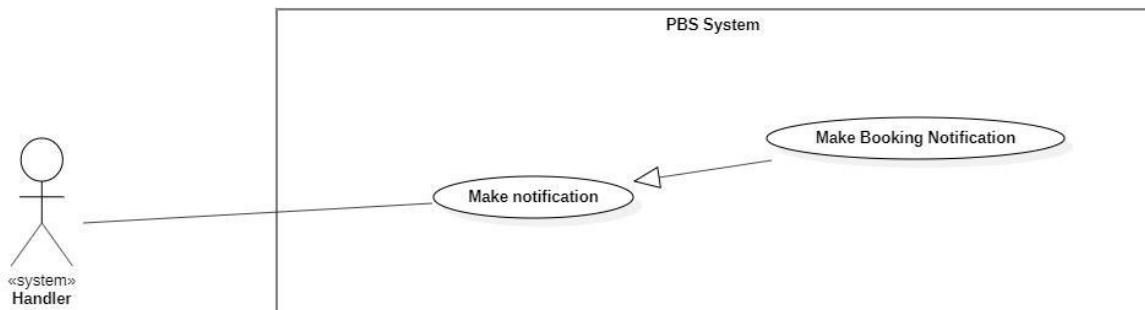


Figure 50 <<system>> Handler make booking notification

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
Alternative Flows:	N/A		
	Exceptions:	No	Cause
Priority:		1	The process runs into an internal error
System outputs error and sends to photographer or customer			
Frequency of Use:	Usually		
Business Rules:	<p>This use case will be triggered in following situations:</p> <ul style="list-style-type: none"> - 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 54 <System Handler> Make booking notification specification

b. <<system>>Handler Make scheduling notification

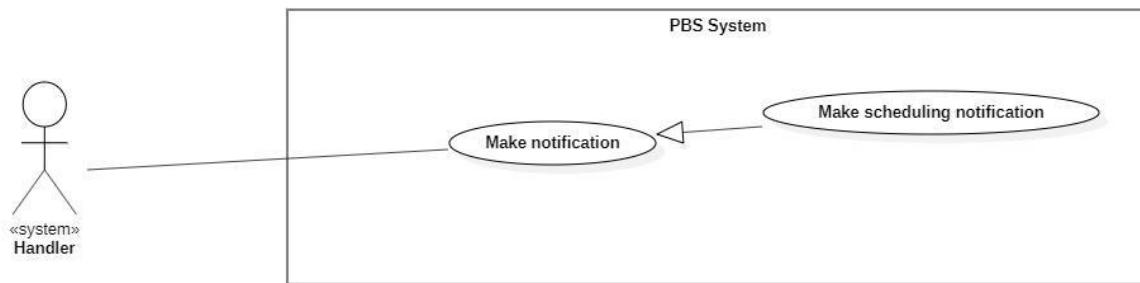


Figure 51 <<system>> Handler make scheduling notification

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:	<p>Success: System make scheduling notification and sends to photographer or customer</p> <p>Fail: System outputs error and sends to photographer or customer</p>											
Normal Flow:	<table border="1"> <thead> <tr> <th>Step</th> <th>Actor Action</th> <th>System Response</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>System handler retrieves scheduling changing request</td> <td>System stores scheduling information and return notification to system handler</td> </tr> <tr> <td>2</td> <td>System handler retrieves notification information and sends to photographer or customer [Exception 1]</td> <td></td> </tr> </tbody> </table>	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]			
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:	<table border="1"> <thead> <tr> <th>No</th> <th>Cause</th> <th>System Response</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>The process runs into an internal error</td> <td>System outputs error and sends to photographer or customer</td> </tr> </tbody> </table>	No	Cause	System Response	1	The process runs into an internal error	System outputs error and sends to photographer or customer					
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: <ul style="list-style-type: none"> - 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 55 <System Handler> Make scheduling notification specification

c. <>Handler Make weather warning

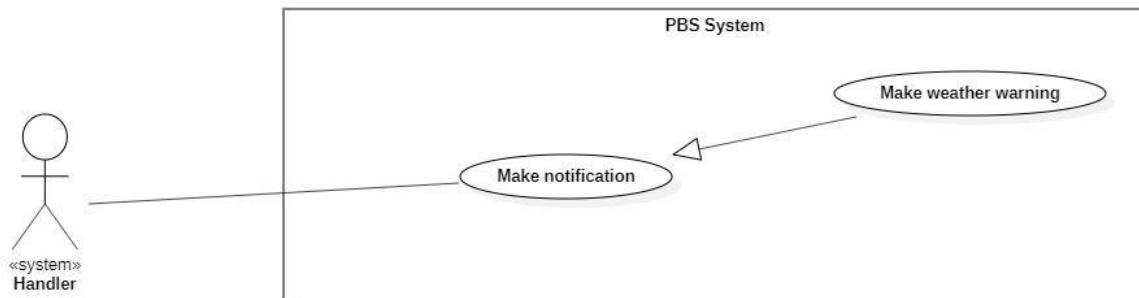


Figure 52 <>Handler make weather warning

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: <ul style="list-style-type: none">- 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 56 <System Handler> Make weather warning specification

d. <>system>>Handler Make time warning

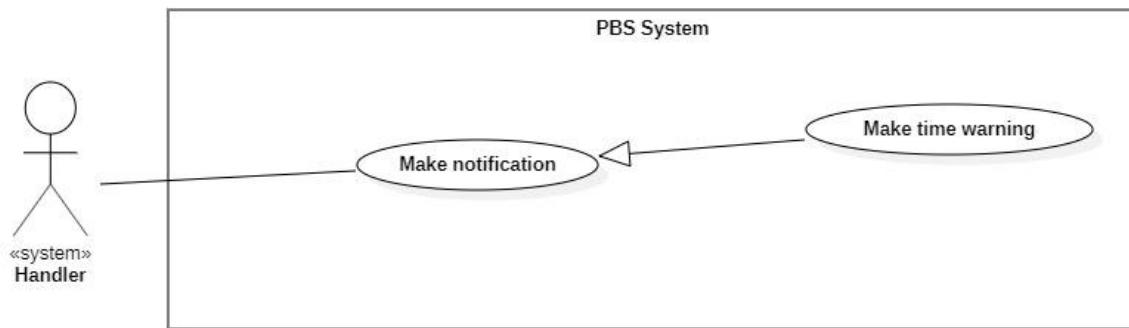


Figure 53 <>system>> Handler make time warning

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:	<p>This use case will be triggered in following situations:</p> <ul style="list-style-type: none"> - 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 57 <System Handler> Make time warning specification

3. Functional Requirements

3.1 System Functional Overview

a. Screen Flow

1. Customer's screen flow

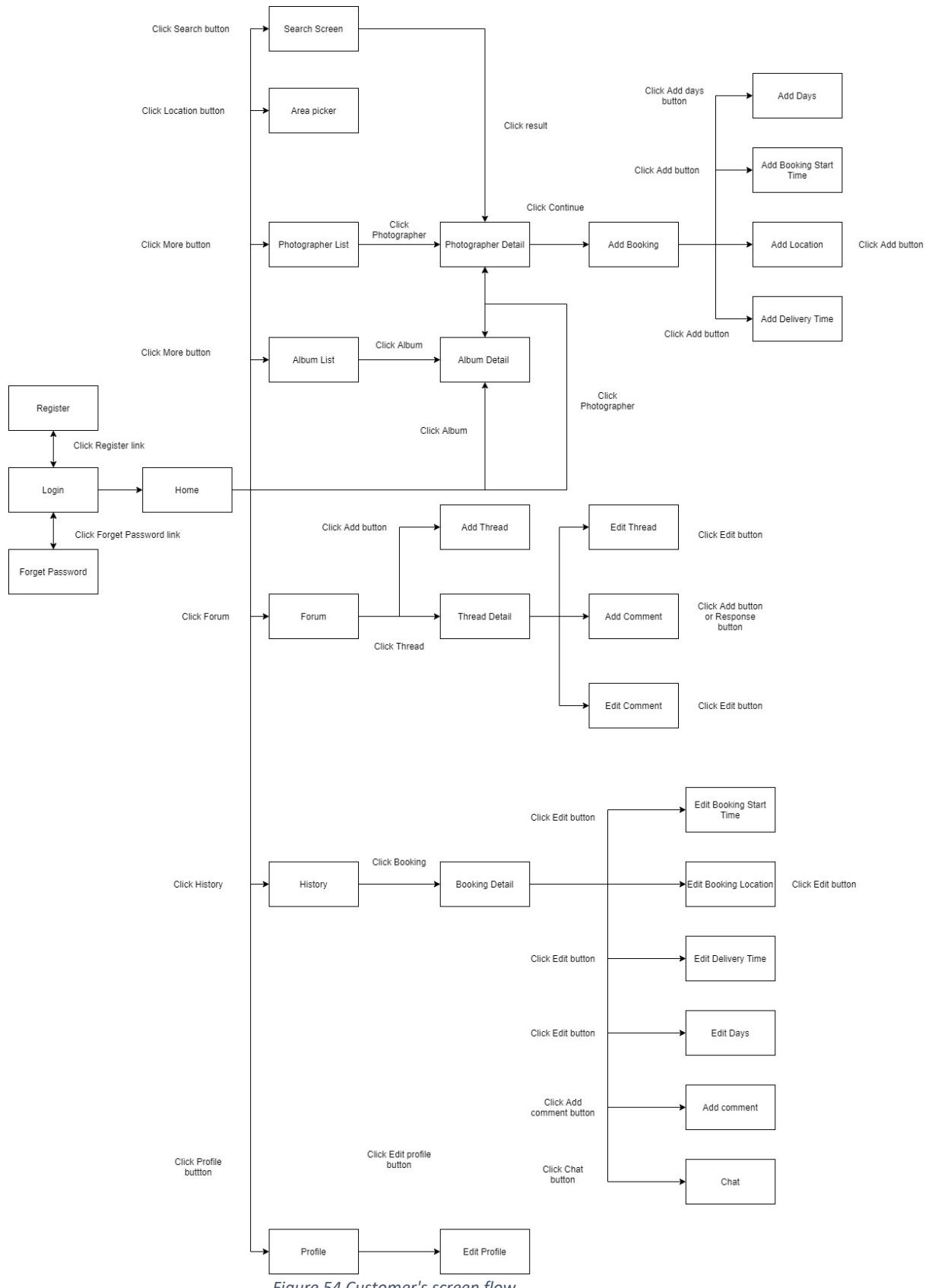


Figure 54 Customer's screen flow

2. Photographer's screen flow

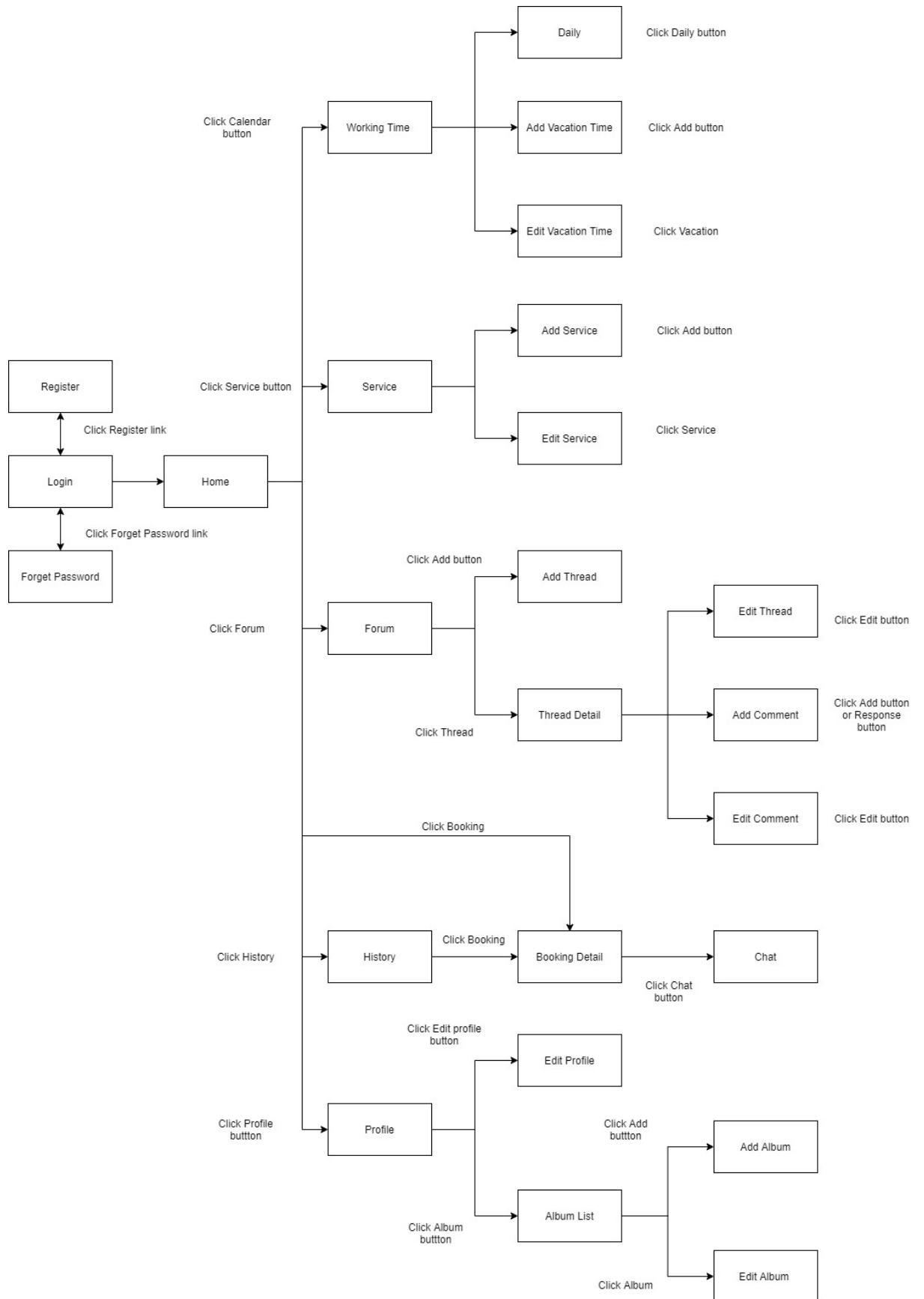


Figure 55 Photographer's screen flow

b. Screen Details

1. Customer's screen details

#	Feature	Screen	Description
1	Register	Register	Register screen included: username textfield, password textfield, re-password textfield
2	Login	Login	Login screen included: username textfield, password textfield
3	Forgot password	Forgot password	Forgot password screen included: email textfield
4	View application	Home	Main screen included: Search icon, category list, photographer list, album list
5	View profile	Profile	Profile screen included: fullname, username, phone number, email.
6	Update profile	Edit profile	Edit profile screen included: fullname textfield, phone number textfield, email textfield.
7	View photographer	Photographer List	Photographer list screen included: photographer name, price, working location, rate.
		Photographer Detail	Detail request screen included: photographer name, describe, album, review, schedule, package.
9	Booking	Add Booking	Add booking screen included: photographer name, start time button, delivery time button, location button, package drop list, delivery method drop list.
		Add Days	Add days screen included: photographer name, days button, delivery time button, location button, package drop list, delivery method drop list.
		Add Booking Start Time	Add booking start time screen included: calendar picker, time picker.
		Add Location	Add location include: map picker
		Add Delivery Time	Add delivery time screen included: calendar picker, time picker.
14	View album	Album List	Album list screen included: name, rating.
		Album Detail	Album detail screens include: name, rating, describe, date, location.
16	View forum	Forum	Forum screen included: list thread

17	Add thread	Add Thread	Add thread screen included: name textfield , topic button, detail textfield
18	View thread	Thread Detail	Thread detail screen included: thread, comment
19	Edit thread	Edit Thread	Edit thread screen included: name textfield, detail textfield
20	Add comment in forum	Add Comment	Add comment screen included: comment textfield
21	Edit comment in forum	Edit Comment	Edit comment screen included: comment textfield
22	View booking	History	History screen included: booking list
		Booking Detail	Booking detail screen included: photographer name, chat, location, start time, delivery time, delivery method, package
24	Edit booking	Edit Booking Start Time	Edit booking start time screen included: time picker, calendar picker
		Edit Booking Location	Edit booking location screen included: map picker
		Edit Delivery Time	Edit delivery screen included: time picker, calendar picker
		Edit Days	Edit days screen included: days button
27	Add comment	Add Comment	Add comment screen included: rating slider, comment textfield
28	Chat	Chat	Chat screen included: chat bubble, chat textfield
29	Pick shooting area	Area Picker	Area picker screen included: map picker
30	Search photographer and package	Search	Search screen included: search area, result area

Table 58 Customer's screen details

2. Photographer's screen details

#	Feature	Screen	Description
1	Register	Register	Register screen included: username textfield, password textfield, re-password textfield
2	Login	Login	Login screen included: username textfield, password textfield

3	Forgot password	Forgot password	Forgot password screen included: email textfield
4	View application	Home	Main screen included: calendar icon, pending task list, to do task list
5	View profile	Profile	Profile screen included: fullname, username, phone number, working location, email.
6	Update profile	Edit profile	Edit profile screen included: fullname textfield, phone number textfield, working location picker, email textfield.
7	View album	Album List	Album list screen included: album picture
8	Add album	Add Album	Add album screen included: album name textfield, date picker, location picker, describe textfield, category picker, photo picker
9	Edit album	Edit Album	Edit album screen included: album name textfield, date picker, location picker, describe textfield, category picker, photo picker
10	View forum	Forum	Forum screen included: list thread
11	Add thread	Add Thread	Add thread screen included: name textfield, topic button, detail textfield
12	View thread	Thread Detail	Thread detail screen included: thread, comment
13	Edit thread	Edit Thread	Edit thread screen included: name textfield, detail textfield
14	Add comment in forum	Add Comment	Add comment screen included: comment textfield
15	Edit comment in forum	Edit Comment	Edit comment screen included: comment textfield
16	Booking	History	History screen included: booking list
		Booking Detail	Booking detail screen included: customer name, chat, location, start time, delivery time, delivery method, package
17	Chat	Chat	Chat screen included: chat bubble, chat textfield
18	View working time	Working time	Working time screen included: daily button, add vacation button, vacation day list
19	Manage daily working time	Daily	Daily screen included: day of the week picker, begin time picker, end time picker
20	Add vacation time	Add Vacation time	Add vacation time included: calendar picker, name textfield

21	Edit vacation time	Edit Vacation time	Edit vacation time included: calendar picker, name textfield
22	View Service	Service	Service screen included: service list
23	Add Service	Add Service	Add service included: service type radio button, service name textfield, category drop list, describe textfield, duration textfield, price textfield, delivery type picker, detail list
24	Edit Service	Edit Service	Edit service included: service type radio button, service name textfield, category drop list, describe textfield, duration textfield, price textfield, delivery type picker, detail list

Table 59 Photographer's screen details

c. Screen Authorization

Screen	Guest	Customer	Photographer	Admin	Authenticated User
Register	X				
Login		X	X	X	X
Forgot password		X	X	X	X
Home		X	X	X	
Profile		X	X	X	X
Edit profile		X	X	X	X
Photographer List		X			
Photographer Detail		X			
Add Booking		X			
Add Days		X			
Add Booking Start Time		X			
Add Location		X			
Add Delivery Time		X			
Album List		X	X		
Album Detail		X	X		
Add Album			X		
Edit Album			X		
Forum		X	X	X	

Add Thread		X	X	X	
Thread Detail		X	X	X	
Edit Thread		X	X	X	
Add Comment in Forum		X	X	X	
Edit Comment in Forum		X	X	X	
History		X	X		
Booking Detail		X	X		
Edit Booking Start Time		X			
Edit Booking Location		X			
Edit Delivery Time		X			
Edit Days		X			
Add Comment		X			
Chat		X	X		
Area Picker		X			
Working time			X		
Daily			X		
Add Vacation time			X		
Edit Vacation time			X		
Service			X		
Add Service			X		
Edit Service			X		

Figure 53 Screen authorization

In which:

- Guest: The user has not registered for an account.
- Customer: The user who used this app for booking photographer services.
- Photographer: The user who used this app for providing services.
- Admin: The user who manages the whole system.
- Authenticated user: The user who has logged in to the system successfully.

d. Non-Screen Functions

No	Feature	System Function	Description
1	Authentication	Register	Function allows user to register into the system to be either photographer or customer.

		Login	Function allows user to log into the system if they have account.
2	Booking	Book photographer	Function allows customer to book a photographer.
		Accept booking	Function allows photographer to accept incoming request.
		Reject booking	Function allows photographer to reject incoming request.
		Cancel booking	Function allows photographer or customer to cancel booking. But if their cancel is inappropriate, they could be ban by admin.
		Edit booking	Function allows customer to edit their request, they can edit information like time, location or package.
		Review booking	Function allows customer to review booking after it done.
3	Photographer	Manage package	Function allows photographer to manage their package service.
		Manage album	Function allows photographer to manage their album.
		Manage working calendar	Function allows photographer to manage their working schedule including add busy day, edit working weekday.
4	Forum	Open thread	Function allows user to open new thread.
		Edit thread	Function allows user to edit their posted thread.
		Comment thread	Function allows user to comment on a thread.
		Ban thread	Function allows admin to ban a thread.
5	Profile	Edit profile	Function allows user to change their basic information like full name, avatar, etc.
		Change password	Function allows user to change their password.
6	Notification	Make weather warning	Function allows system to warn customer about bad weather.
		Make scheduling notification	Function allows system to notify customer or photographer on their schedule.
		Make booking notification	Function allows system to notify customer or photographer about booking.

Table 60 Non-screen functions

e. Entity Relationship Diagram

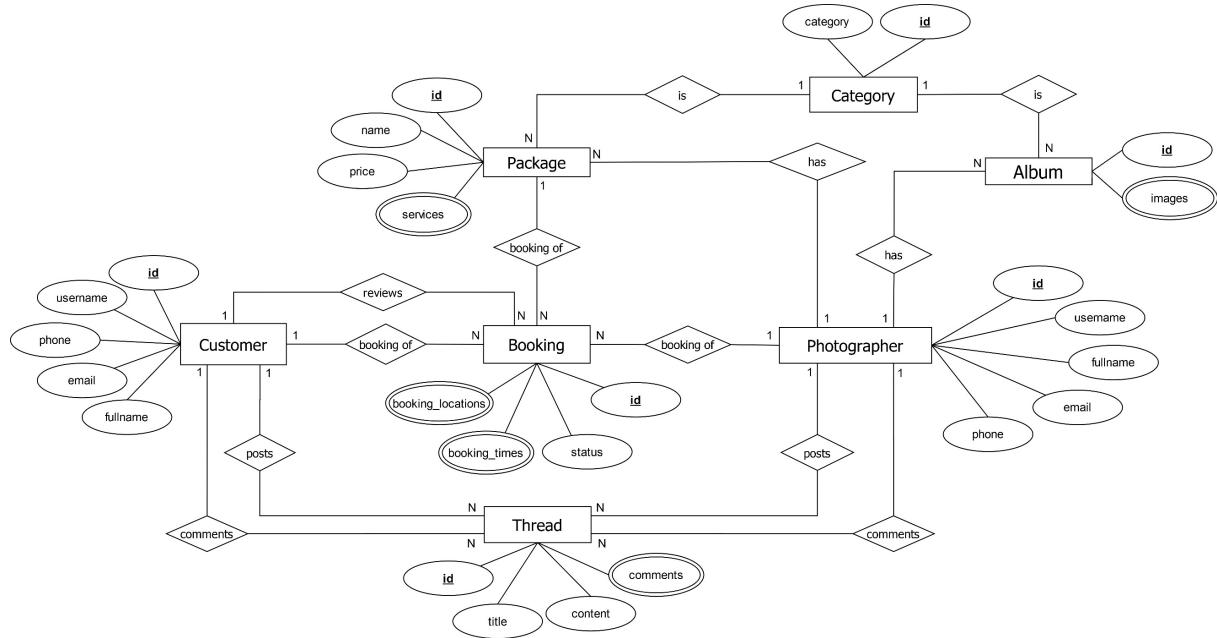


Figure 56 ER Diagram

Entities List

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

Table 61 Entities List

4. Non-Functional Requirements

4.1 External Interfaces

a. User Interfaces

For web application:

UI-1: GUI has to be simple and easy to use.

UI-2: The language used in the web application is Vietnamese.

UI-3: The web application has to be responsive on multiple devices: laptop, tablet and mobile.

For mobile application:

UI-1: User interface is pleasant to look at and easy to use.

UI-2: The language is Vietnamese.

UI-3: The user interface can be responsive on multiple screen sizes.

UI-4: The functionalities are arranged clearly.

b. Software Interfaces

This system will have required software MySQL to program and create and manage the database for the system. Use for source control is Git and GitHub. Test browsers use Chrome 96 or above.

Specification for developing the web application is Java EE 11.0.8. Program tools are Visual Studio Code, Android Studio, IntelliJ.

The system will require software window 7 and window 10 to program and manage photographer booking systems. Booking process of the system that organizes the distribution of users in the mobile app and able to do this manage review.

c. Hardware Interfaces

Just like any other system, this system requires a basic computer that consists of CPU, monitor, keyboard and mouse or laptop and smartphone for input and output. Smartphones could be used to do booking and use the system. Besides, for direction the system connected with a GPS device in every smartphone to get the location detail.

d. Communications Interfaces

The system communication interface is completely dependent on correctly sending and retrieving data from the database. Other than that is an online protocol to connect between GPS and the mobile.

4.2 Quality Attributes

a. Usability

- The system does not require its users to have any technical knowledge before using its functionalities.
- Layout and navigation are clear and easy to use.
- Customers can use PSB easily even if they have never used it before. They can find a photographer and make a request within 5 minutes.
- Photographers can use PSB easily even if they have never used it before. They can view and confirm requests received from the customers within 5 minutes. They can also use other

functionalities quickly and effectively with easy-to-understand instructions and simple operation.

b. Reliability

- The availability of the system is 80%. The system can be used within 60 minutes and will automatically log the user out if there is no operation within 5 minutes.

c. Performance

All requests will be processed within 10 seconds on an Internet connection above 25Mbps.

d. Dependability

d1. Security

- All user's passwords will be encrypted before being saved into the database.
- Accounts will be verified by email.
- Username must be unique.
- Photographers' email and phone number will be secured.

d2. Safety

- A service price a photographer has posted cannot be changed when a customer chooses that service. If there is a surcharge, it must be negotiated with the customer.
- When the photos return day has come, if the photographer does not return the photos, the request will be cancelled automatically.

e. Supportability

- Online bookings in real time: The customer will be able to see all the available dates directly, and make a booking. If there is no further availability on one date, this will be blocked in real time to prevent further bookings.
- Intelligent calendar: An intelligent and easy-to-understand booking or occupancy calendar provides the user with a summary of all bookings, capacity and turnover for the required periods at a glance.
- Online and chat support for booking systems.
- Booking management via smartphone.

f. Design Constraints

- The software language is Vietnamese.
- The software process is Scrum.

g. Support Documents

The system does not use any references to support document writing.

h. Purchased Components

To support the system operation, we have used some third-party services such as:

- Google Map API.
- AWS S3.
- ClimaCell weather api.

5. Other Requirements

5.1 Messages List

#	Message code	Message Type	Context	Content
1	MSG01	In line	There is not any search result	Không có kết quả.
2	MSG02	In red, under the text box	Input-required fields are empty	Thông tin này là bắt buộc.
3	MSG03	Toast message	Updating asset(s) information successfully	Cập nhật thành công.
4	MSG04	Toast message	Adding new asset successfully	Thêm * thành công.
5	MSG05	Toast message	Confirming email of asset hand-over is sent successfully	A confirmation email has been sent to {email_address}.
6	MSG06	Toast message	Resetting asset information successfully	Thay đổi * thành công
7	MSG07	Toast message	Deleting asset information successfully	Xóa * thành công.
8	MSG08	In red, under the text box	Input value length > max length	Bạn vui lòng nhập đủ {max_length} ký tự.
9	MSG09	In line	Username or password is not correct when clicking sign-in	Thông tin đăng nhập không chính xác.
10	MSG10	In line	Password and Confirm password are not matched.	Mật khẩu nhập lại không khớp.
11	MSG11	Toast message	Register successfully	Đăng ký thành công
12	MSG12	In red, under the text box	Email is invalid	Email mail này không hợp lệ. Vui lòng nhập email khác
13	MSG13	In red, under the text box	Username is existing	Tên đăng nhập của bạn đã có người sử dụng. Vui lòng nhập tên khác
14	MSG14	Toast message	The customer book photographer successfully.	Gửi yêu cầu thành công.
15	MSG15	In red, under the text box	The customer does not choose a calendar and address.	Bạn chưa chọn thông tin.
16	MSG16	Notifications	The system sends notifications to the customer when photographers accept requests.	Yêu cầu của bạn đã được xác nhận
17	MSG17	Notifications	The system sends notification to the customer when a photographer declines a request.	Yêu cầu của bạn đã bị từ chối.

18	MSG18	In red, under the text box	Current password is invalid	Mật khẩu nhập lại chưa chính xác
19	MSG19	Toast message	The system shows notifications to the admin when blocking users.	Bạn có muốn khóa người dùng này?
20	MSG20	Toast message	The admin block user successfully.	Người dùng đã được khóa thành công.
21	MSG21	Toast message	The system sends notifications to the user when the account has been blocked.	Tài khoản của bạn đã bị vi phạm. Vui lòng liên hệ với quản trị viên
22	MSG22	Toast message	The system shows notifications to the admin when unblocking the user.	Bạn có muốn mở khóa người dùng này?
23	MSG23	Toast message	The system sends notifications to the user when the account has unblocked.	Người dùng đã được mở khóa thành công
24	MSG24	In red, under the text box	Information is duplicated	* đã tồn tại.

Table 62 Messages List

IV. Software Design Document

1. Overall Description

1.1 Assumptions

Assuming all the users for this system have basic mobile knowledge and knows how to operate the system smoothly.

1.2 Design Constraints

Hard drive space: The system will need at least 8GB storage space and minimum 1GB Storage space in order to run the booking system.

2. System Architecture Design

2.1 Overall Architecture

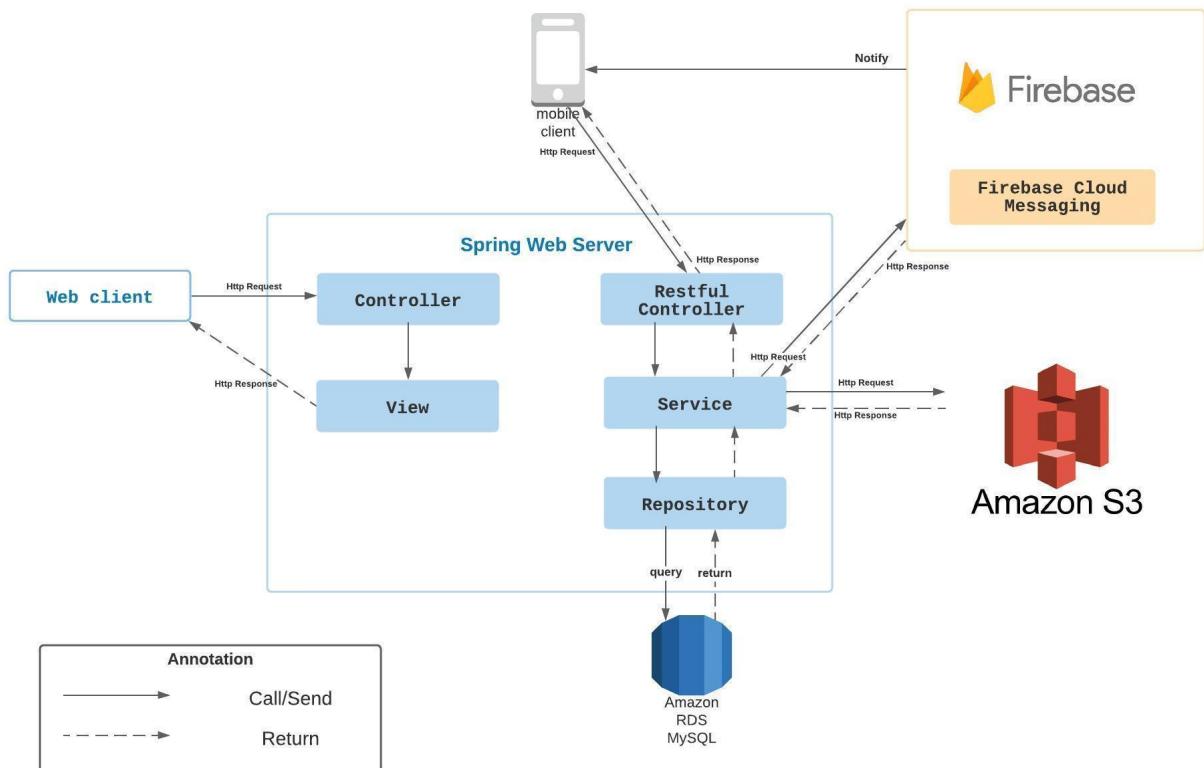


Figure 57 Overall Architecture

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 and Amazon RDS for MySQL instances.

2.2 System Architecture

2.2.1 Web server architecture

We build our web server following n-tier architecture. We decided to use this pattern for several reasons:

- It has been proven to be a scalable architecture model.
- Provide reusability, maintainability and fault tolerance.

- Helps our members to create reusable and flexible functionality.

2.2.2 Mobile app architecture

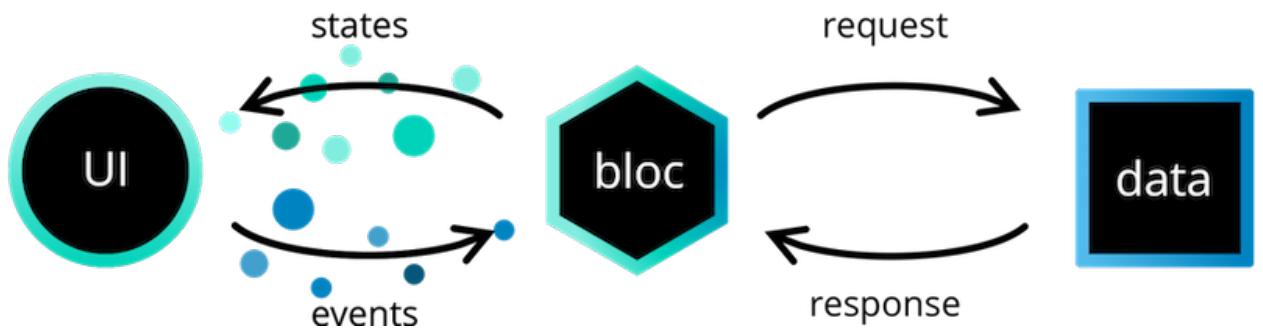


Figure 58 Mobile App Architecture

Reference: <https://bloclibrary.dev/#/architecture>

Bloc pattern allow us to separate our mobile app into 3 layers:

- Presentation: UI
- Business Logic Component: bloc (where we can manage events – click, tap, etc. - send a request to data layer, then retrieve data from data layer and response to input from UI through states).
- Data Layer: Repository (where we can retrieve data from sources).

2.3 Package Diagram

2.3.1 Webapi

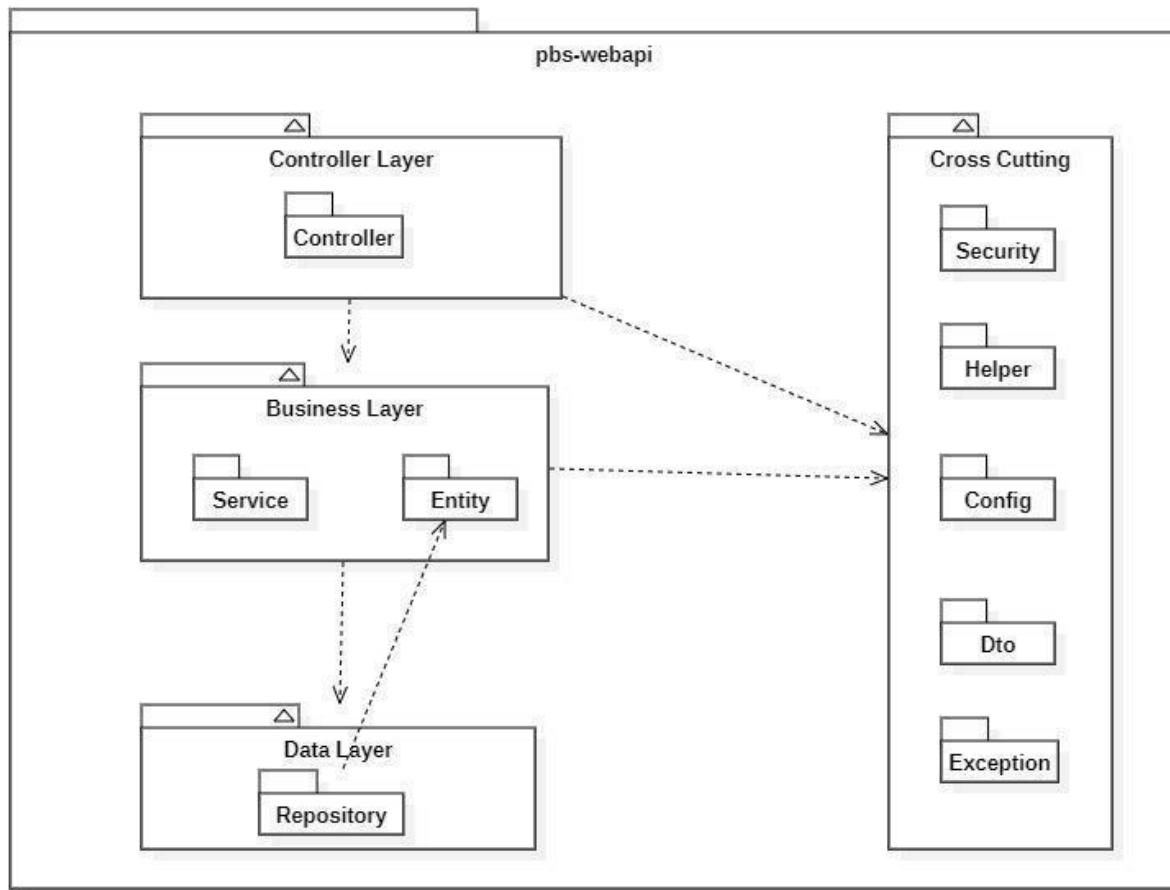


Figure 59 Webapi Package Diagram

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 sharpening the data to return to clients

Table 63 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:

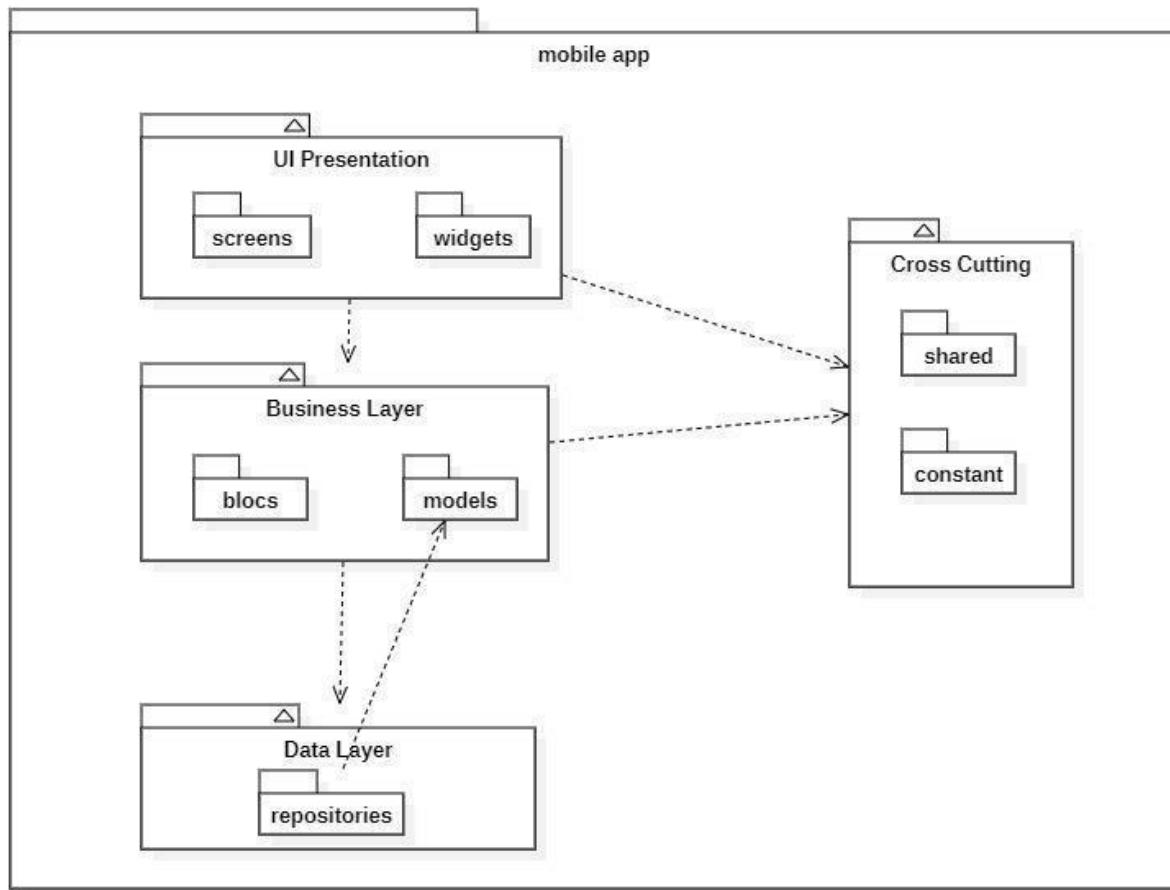


Figure 60 Package Diagram – Mobile app

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 64 Package Diagram Description

3. System Detailed Design

3.1 Overall

a. Class Diagram

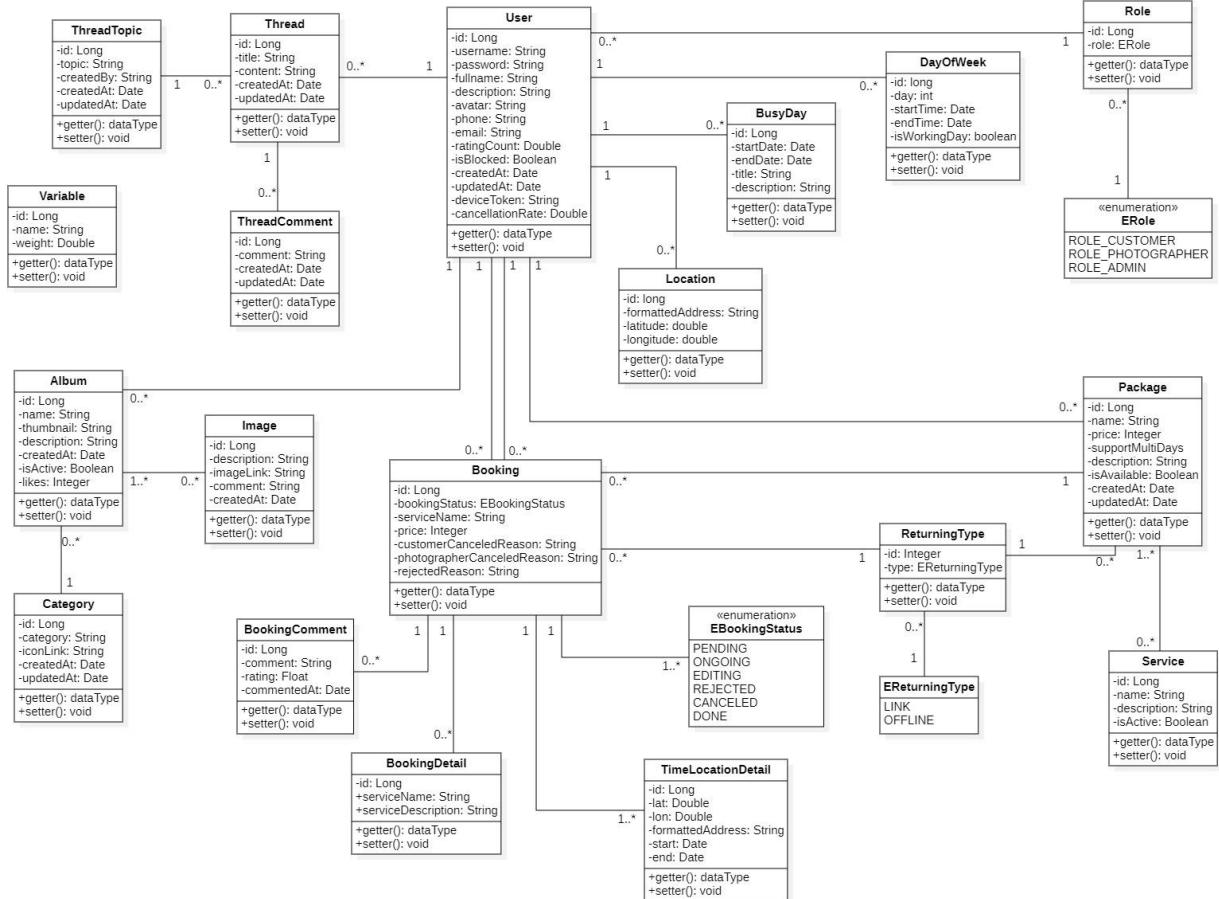


Figure 61 Class Diagram

b. Sequence Diagram(s)

1. Add Category

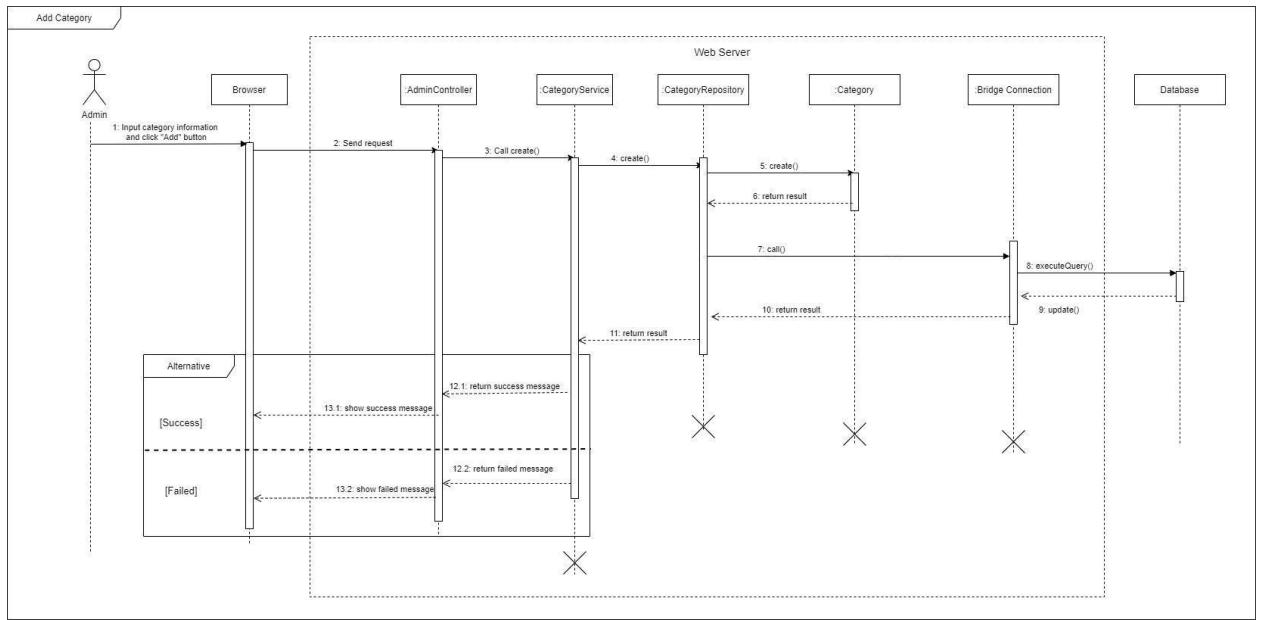


Figure 62 Sequence Diagram - Add Category

2. Edit Category

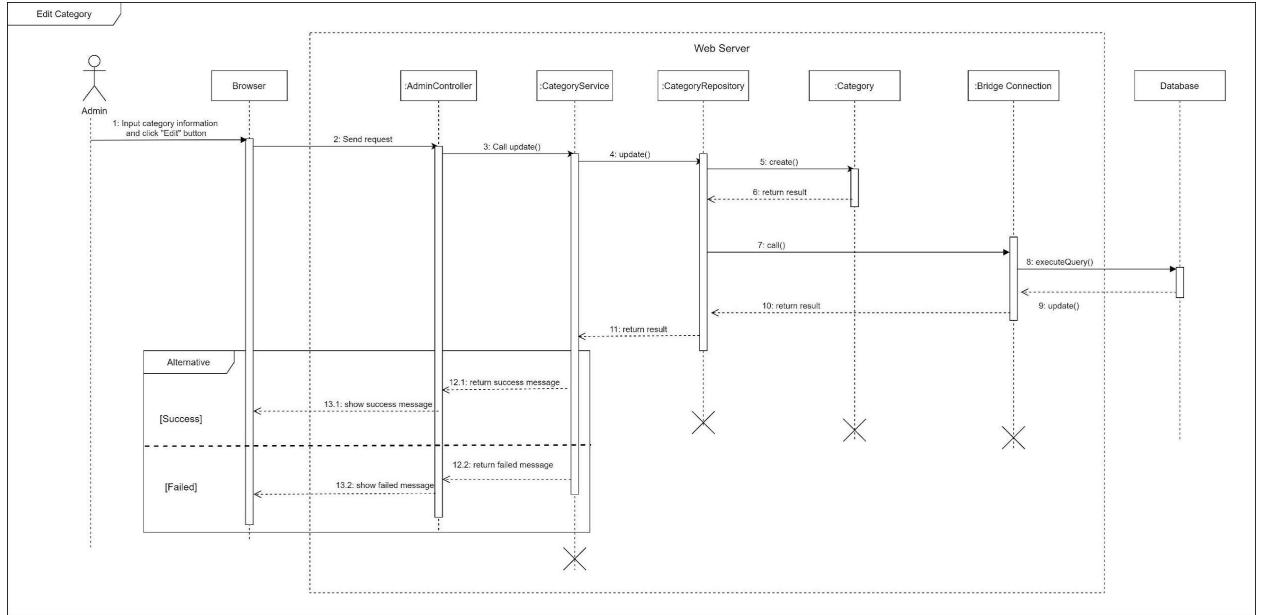


Figure 63 Sequence Diagram - Edit Category

3. Remove category

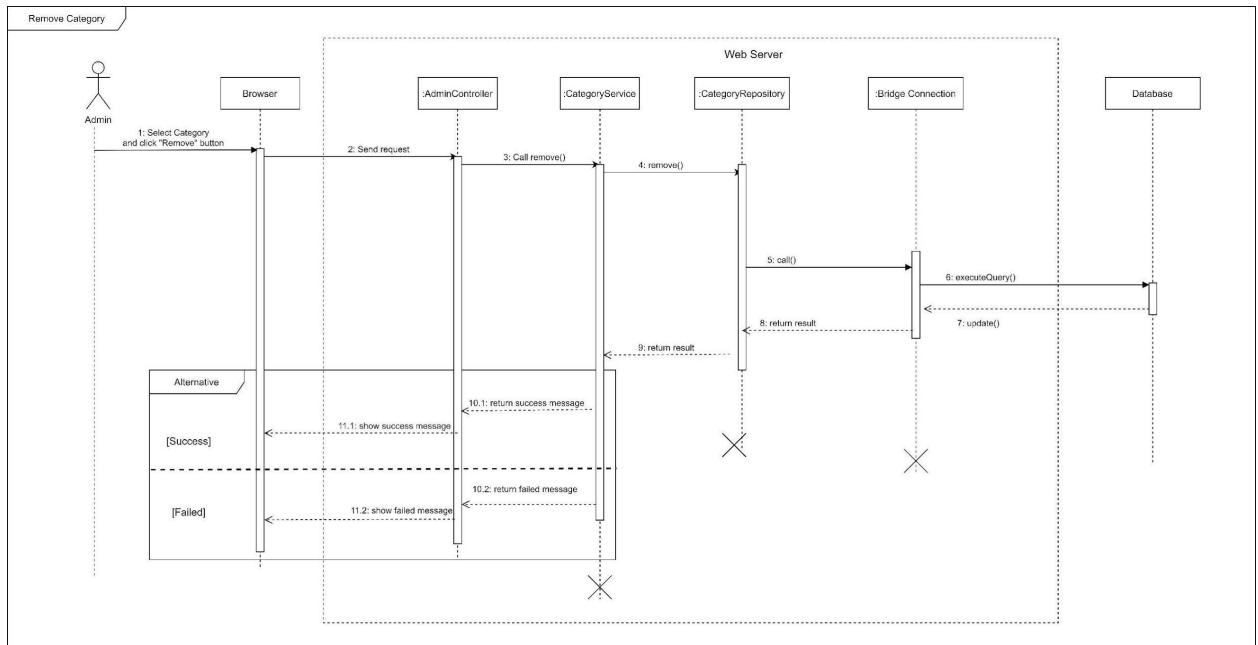


Figure 64 Sequence Diagram - Remove Category

4. Add returning type

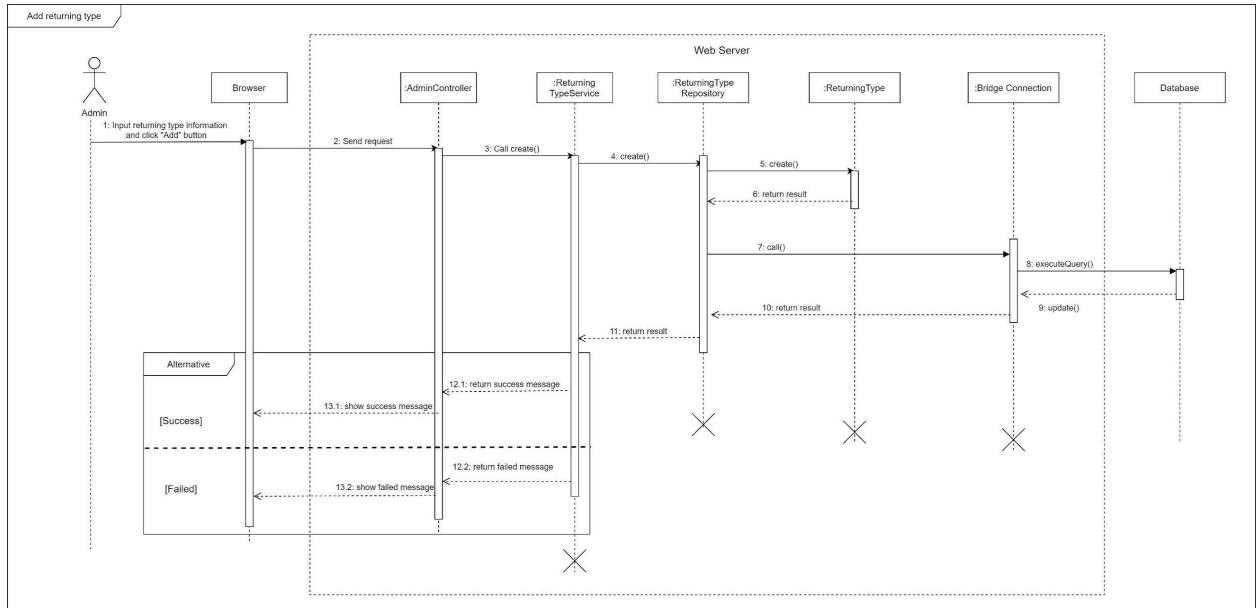


Figure 65 Sequence Diagram - Add returning type

5. Edit returning type

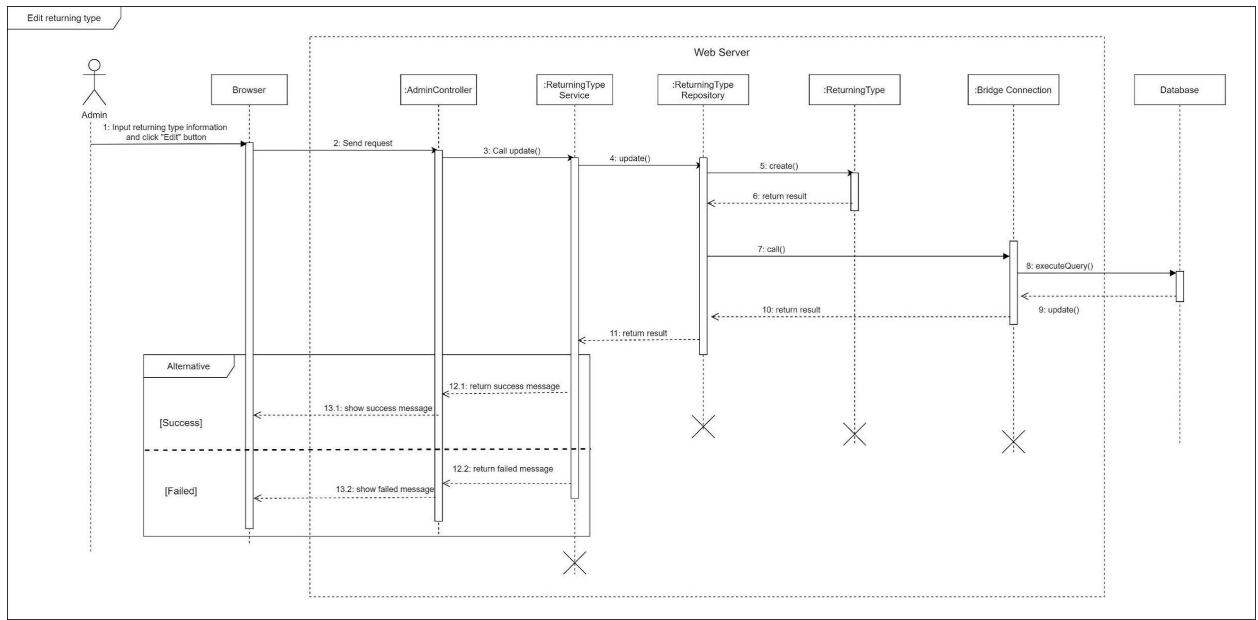


Figure 66 Sequence Diagram - Edit returning type

6. Remove returning type

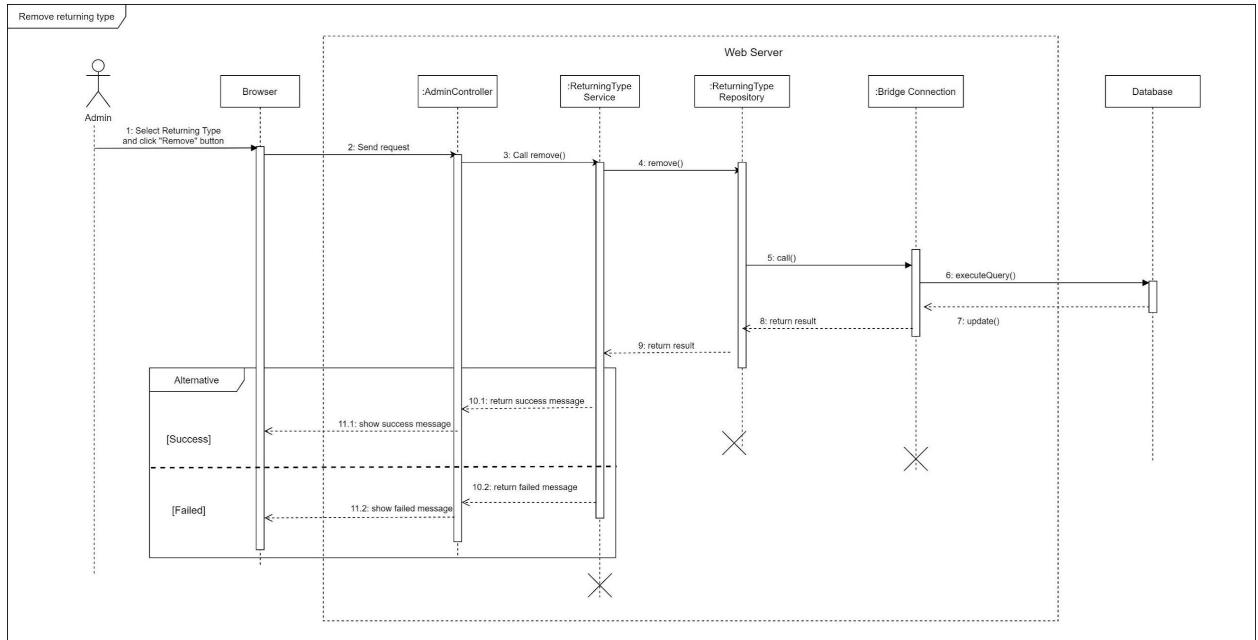


Figure 67 Sequence Diagram - Remove returning type

7. Block user

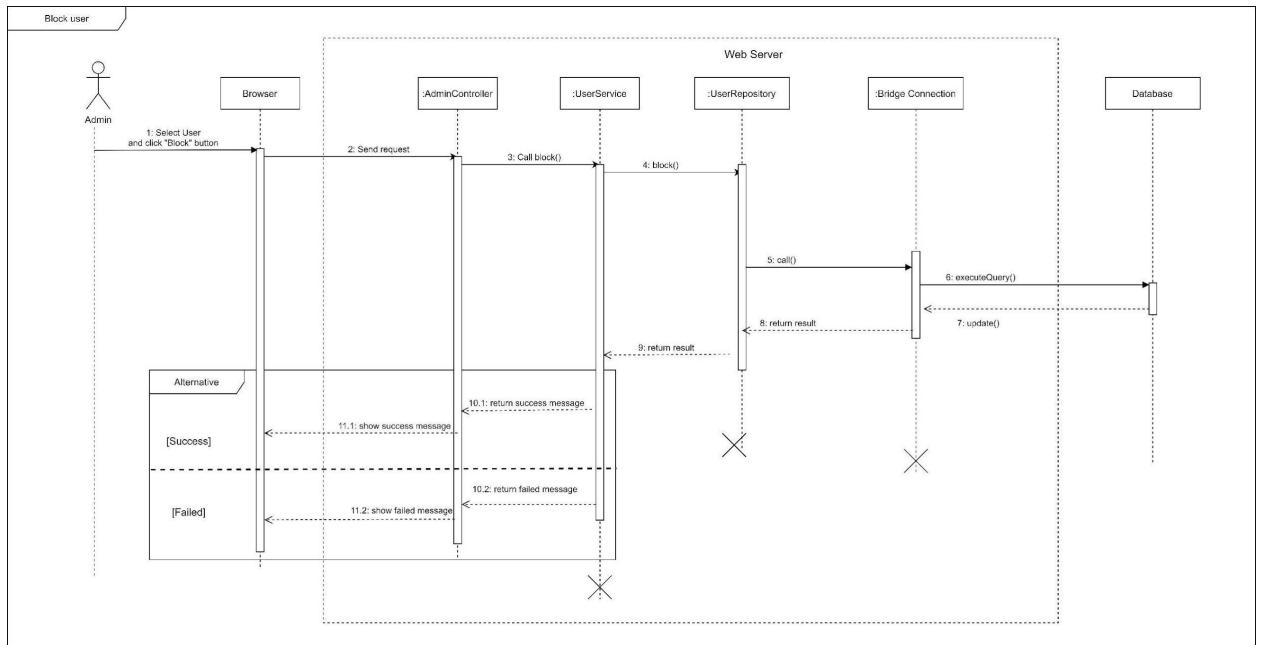


Figure 68 Sequence Diagram - Block User

8. Unblock user

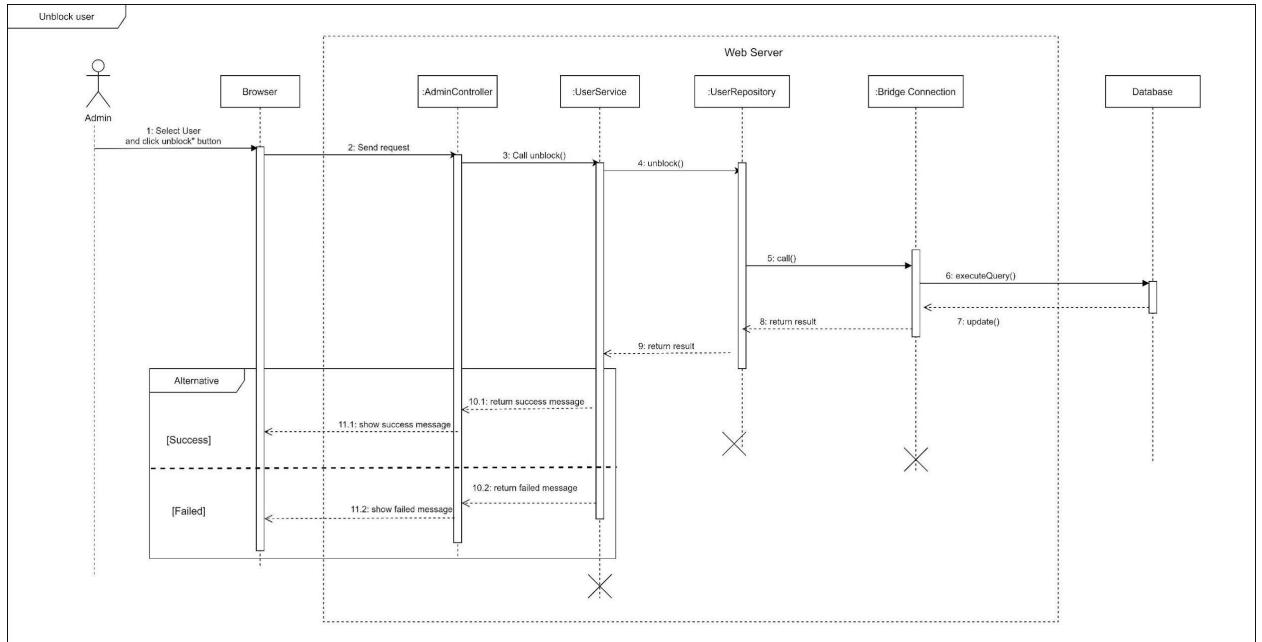
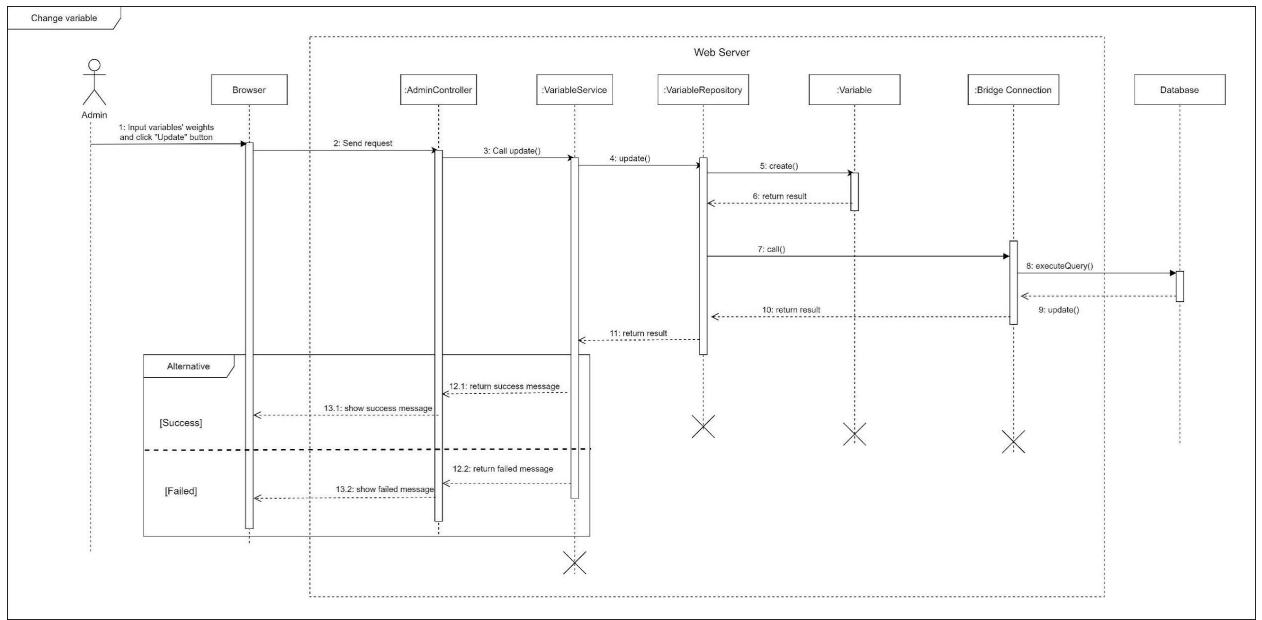


Figure 69 Sequence Diagram - Unblock User

9. Change variable



c. Activity Diagram(s)

1. <Customer> Make request

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

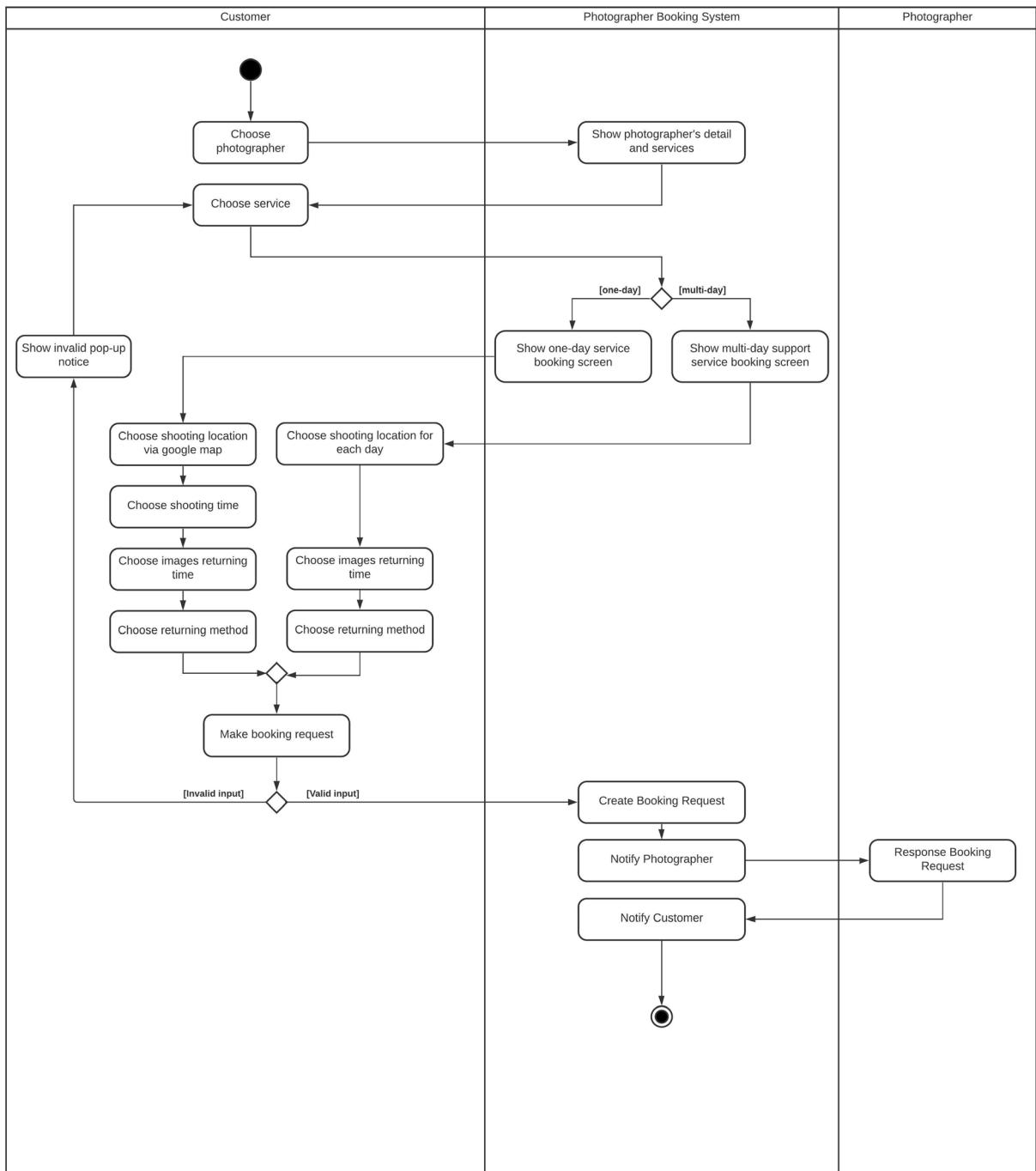


Figure 71 Activity Diagram - Make request

2. <Customer> Edit request

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

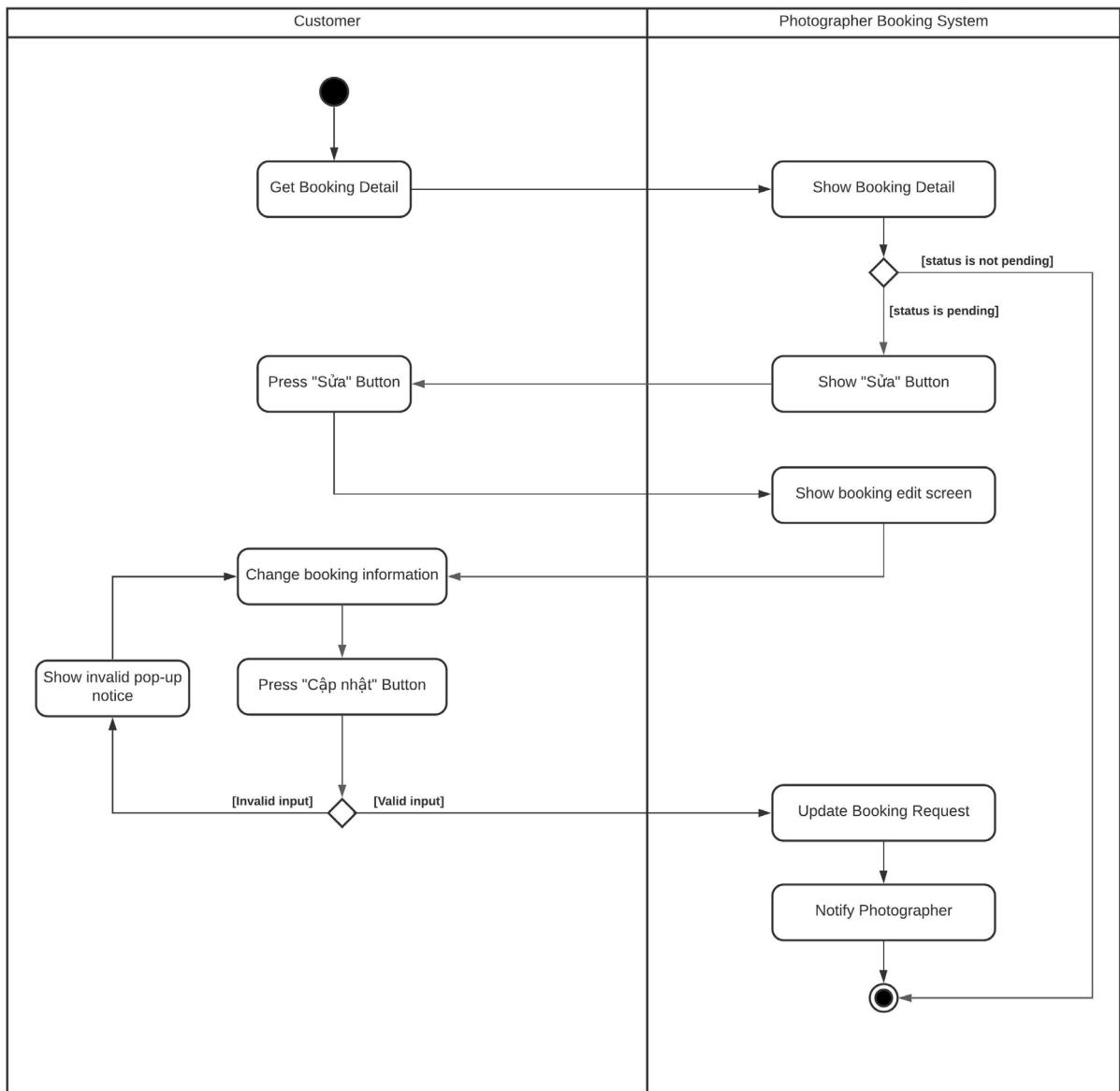


Figure 72 Activity Diagram – Edit Request

3. <Customer> Cancel booking

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

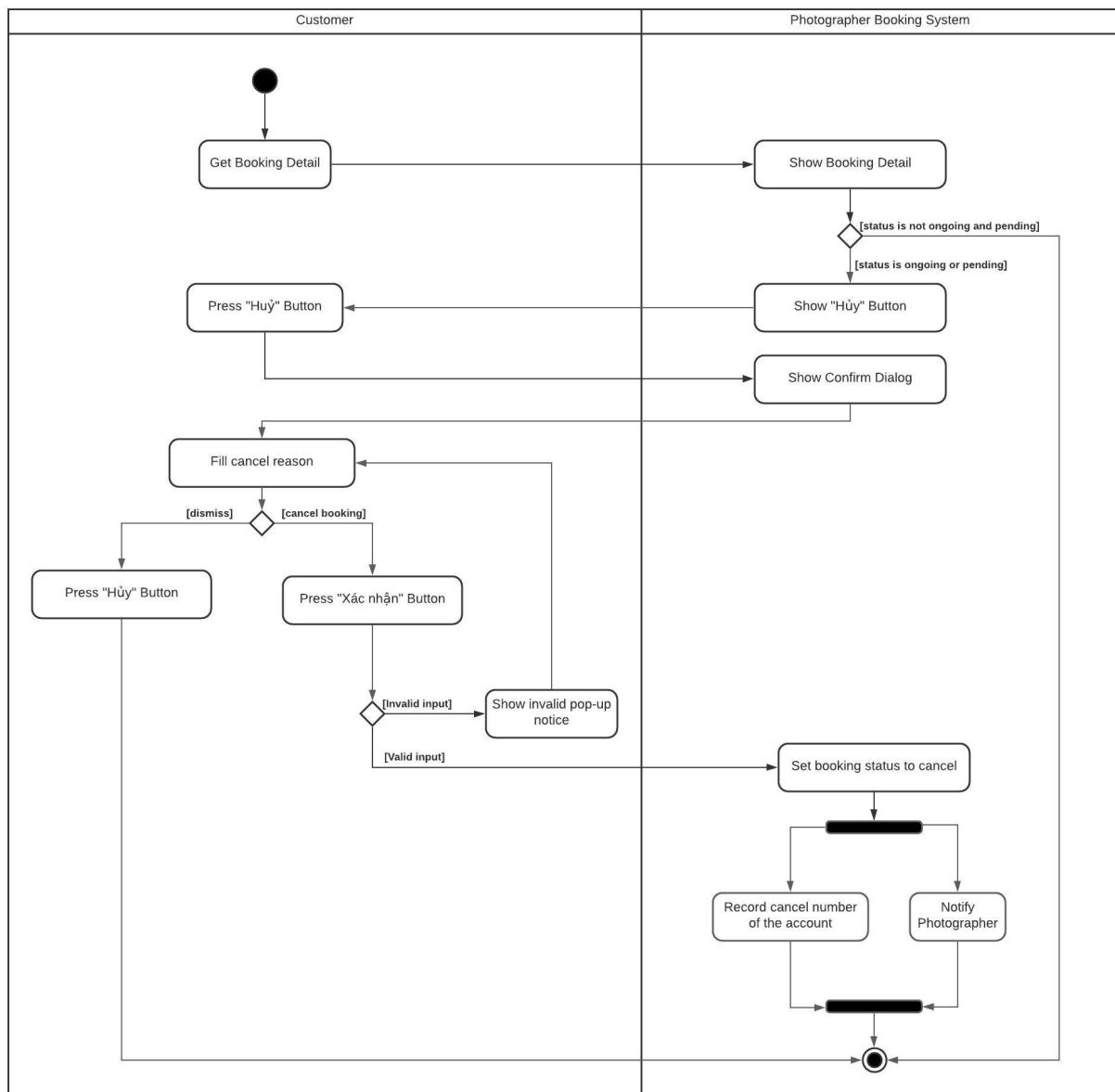


Figure 73 Activity Diagram - Cancel booking

4. <Customer> Rating

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

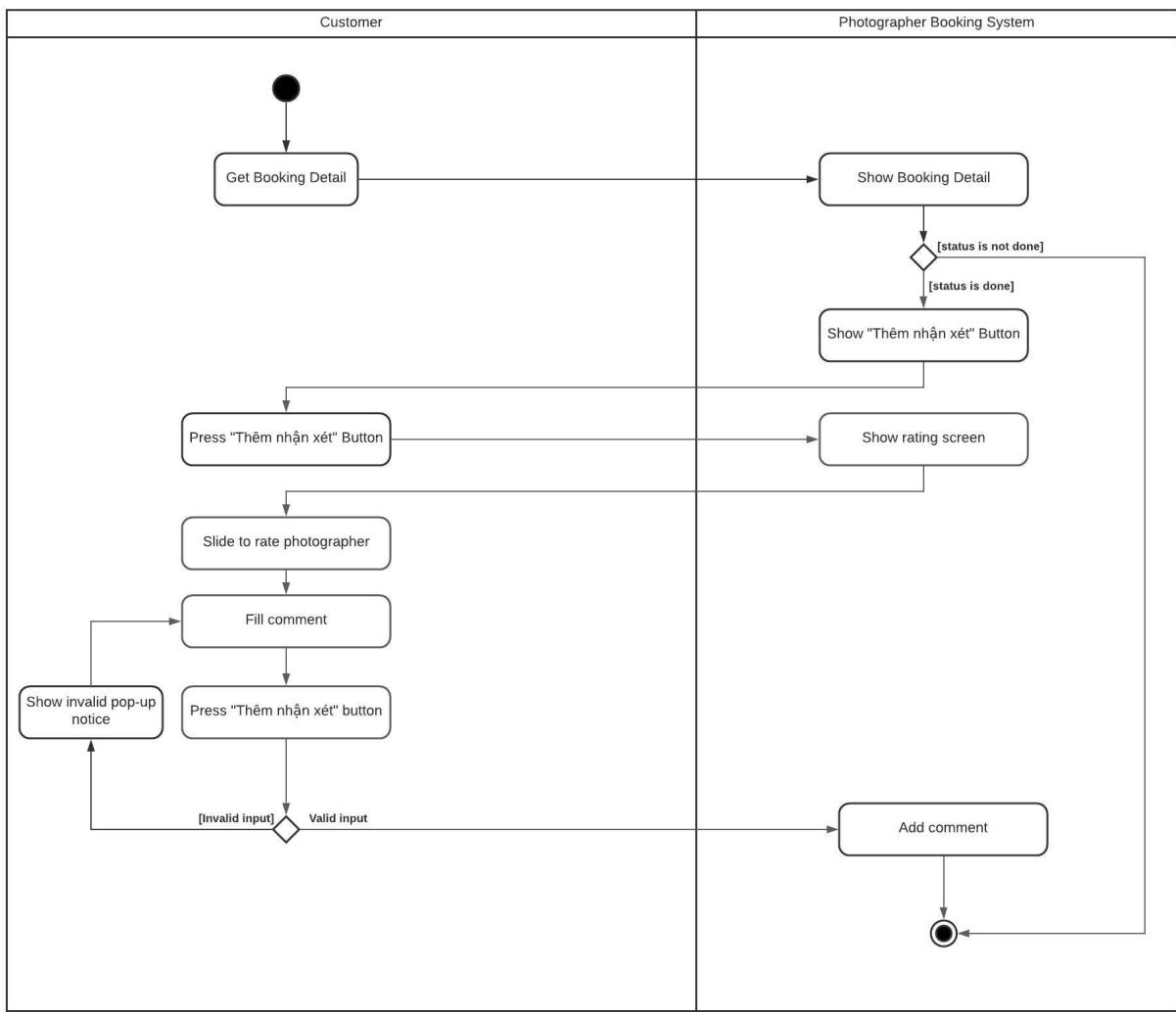


Figure 74 Activity Diagram - Rating

5. <Photographer> Accept booking

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

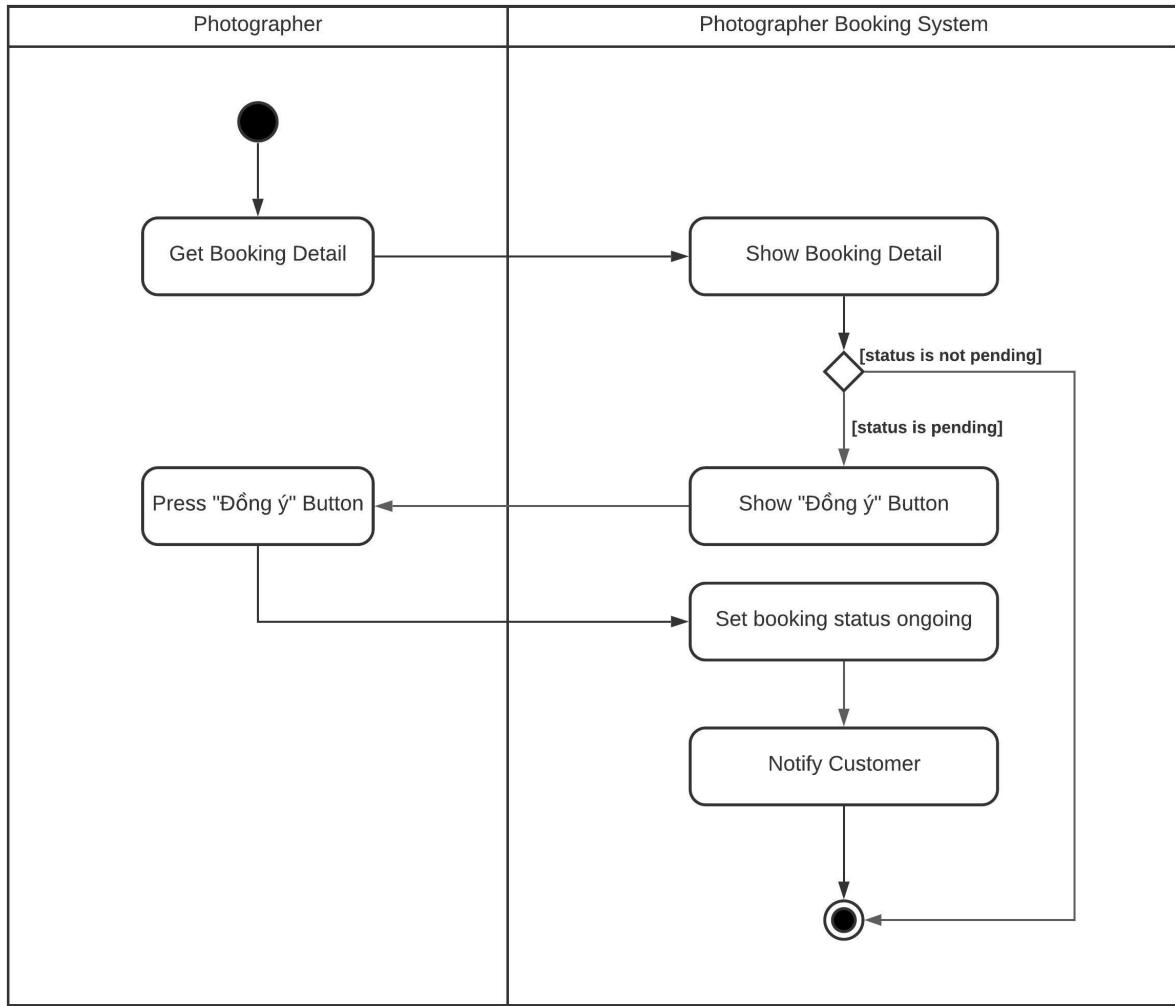


Figure 75 Activity Diagram - Accept booking

6. <Photographer> Reject booking

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

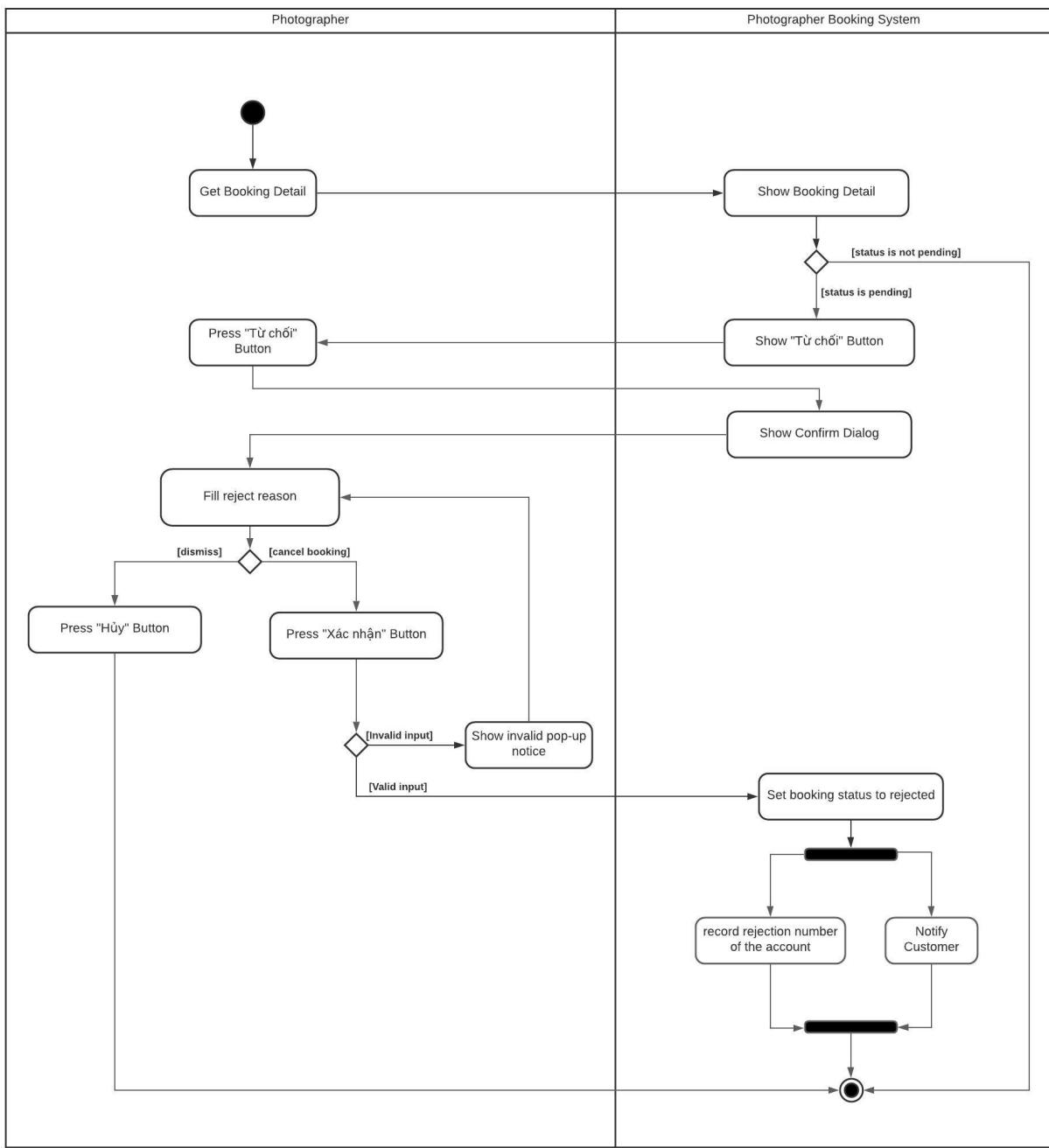


Figure 76 Activity Diagram - Reject booking

d. State Machine Diagram(s)

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

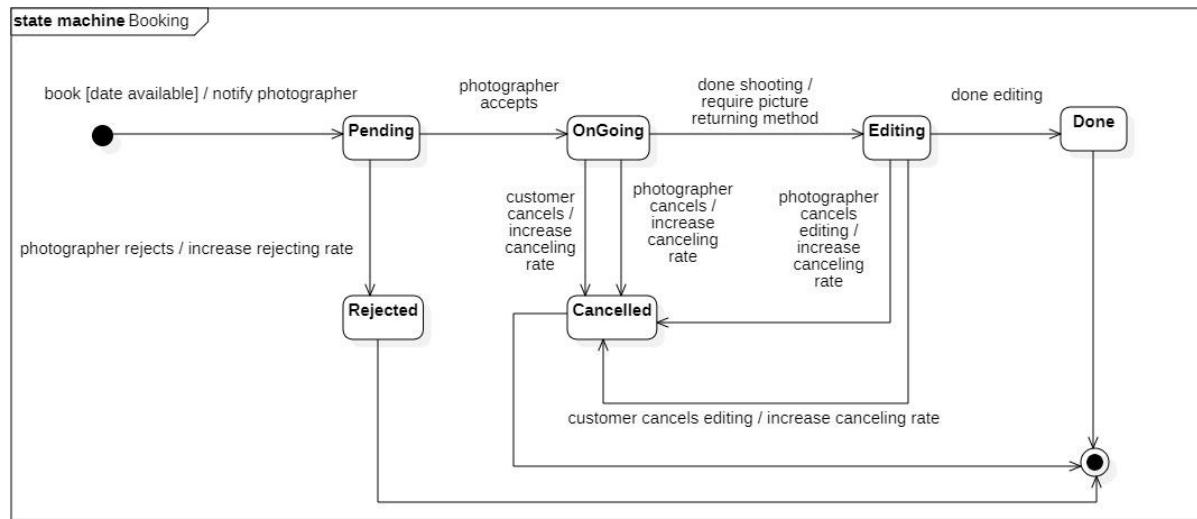


Figure 77 State Machine Diagram - Booking

4. Class Specifications

4.1 User

Attribute	Type	Visibility	Description
id	Long	private	Unique identifier of a user
username	String	private	The attribute represents username of a user
password	String	private	The attribute represents password of a user
fullname	String	private	The attribute represents full name of a user
description	String	private	The attribute represents user self-description
avatar	String	private	The attribute represents avatar link of a user
phone	String	private	The attribute represents phone of a user
email	String	private	The attribute represents email of an user
ratingCount	Float	private	The attribute represents the rating count of a user whose role is photographer
isBlocked	Boolean	private	The attribute represents if user has been blocked
createdAt	Date	private	The attribute represents created date of a user object

updatedAt	Date	private	The attribute represents updated date of a user object
deviceToken	String	private	The attribute represents user's firebase device token
cancellationRate	Double	private	The attribute represents user's cancellation rate

No	Method	Visibility	Return Type	Description
1	getter	public	Attribute Type	get attribute value
2	setter	public	void	set attribute value

Table 65 Class Diagram - User

4.2 Role

Attribute	Type	Visibility	Description
id	Long	private	Unique identifier of a role
role	ERole	private	The attribute represents role name

No	Method	Visibility	Return Type	Description
1	getter	public	Attribute Type	get attribute value
2	setter	public	void	set attribute value

Table 66 Class Diagram - Role

4.3 ERole

Attribute	Type	Visibility	Description
ROLE_CUSTOMER			Enum attribute
ROLE_PHOTOGRAPHER			Enum attribute
ROLE_ADMIN			Enum attribute

No	Method	Visibility	Return Type	Description
1	getter	public	Attribute Type	get attribute value
2	setter	public	void	set attribute value

Table 67 Class Diagram - ERole

4.4 Package

Attribute	Type	Visibility	Description
id	Integer	private	Unique identifier of a package
name	String	private	The attribute represents the package name
price	Integer	private	The attribute represents the package price
supportMultiDays	Boolean	private	The attribute represents if this package support multi days deal
description	String	private	The attribute represents the package description
isAvailable	Boolean	private	The attribute represents if the package still available
createdAt	Date	private	The attribute represents created date of a package object
updatedAt	Date	private	The attribute represents updated date of a package object

No	Method	Visibility	Return Type	Description
1	getter	public	Attribute Type	get attribute value
2	setter	public	void	set attribute value

Table 68 Class Diagram - Package

4.5 Service

Attribute	Type	Visibility	Description
id	Long	private	Unique identifier of a package

name	String	private	The attribute represents the name of a service
description	String	private	The attribute represents the service's description
isActive	Boolean	private	The attribute represents if the service is still available

No	Method	Visibility	Return Type	Description
1	getter	public	Attribute Type	get attribute value
2	setter	public	void	set attribute value

Table 69 Class Diagram - Service

4.6 Booking

Attribute	Type	Visibility	Description
id	Long	private	Unique identifier of a booking
bookingStatus	EBooking Status	private	The attribute represents the status of a booking
serviceName	String	private	The attribute represents the service of a booking
price	Integer	private	The attribute represents the service price of a booking
customerCancelledReason	String	private	The attribute represents the reason if customer decided to cancel a booking
photographerCancelledReason	String	private	The attribute represents the reason if photographer decided to cancel a booking
rejectedReason	String	private	The attribute represents the reason if photographer decided to reject a booking

No	Method	Visibility	Return Type	Description
1	getter	public	Attribute Type	get attribute value
2	setter	public	void	set attribute value

Table 70 Class Diagram - Booking

4.7 EBookingStatus

Attribute	Type	Visibility	Description
Pending			Enum attribute
OnGoing			Enum attribute
Editing			Enum attribute
Rejected			Enum attribute
Cancelled			Enum attribute
Done			Enum attribute

No	Method	Visibility	Return Type	Description
1	getter	public	Attribute Type	get attribute value
2	setter	public	void	set attribute value

Table 71 Class Diagram - EBookingStatus

4.8 Album

Attribute	Type	Visibility	Description
id	Long	private	Unique identifier of album
name	String	private	The attribute represents the name album
thumbnail	String	private	The attribute represents the album's thumbnail link
description	String	private	The attribute represents the album's description
createdAt	Date	private	The attribute represents the created date of album object
isActive	Boolean	private	The attribute represents if album still active
likes	Integer	private	The attribute represents the album like count

No	Method	Visibility	Return Type	Description
1	getter	public	Attribute Type	get attribute value
2	setter	public	void	set attribute value

Table 72 Class Diagram - Album

4.9 Image

Attribute	Type	Visibility	Description
id	Long	private	Unique identifier of image
description	String	private	The attribute represents the description of album
imageLink	String	private	The attribute represents the album's image link
comment	String	private	The attribute represents the album's comment
createdAt	Date	private	The attribute represents the created date of image object

No	Method	Visibility	Return Type	Description
1	getter	public	Attribute Type	get attribute value
2	setter	public	void	set attribute value

Table 73 Class Diagram - Image

4.10 Category

Attribute	Type	Visibility	Description
id	Long	private	Unique identifier of category
category	String	private	The attribute represents the category name
iconLink	String	private	The attribute represents the category icon link
createdAt	Date	private	The attribute represents the created date of category object

updatedAt	Date	private	The attribute represents the updated date of category object
-----------	------	---------	--

No	Method	Visibility	Return Type	Description
1	getter	public	Attribute Type	get attribute value
2	setter	public	void	set attribute value

Table 74 Class Diagram - Category

4.11 DayOfWeek

Attribute	Type	Visibility	Description
id	Long	private	Unique identifier of day of week
day	Int	private	The attribute represents the day of the week
startTime	Date	private	The attribute represents the start time of photographer's working time
endTime	Date	private	The attribute represents the end time of photographer's working time
isWorkingDay	Boolean	private	The attribute represents if this day of week is working day

No	Method	Visibility	Return Type	Description
1	getter	public	Attribute Type	get attribute value
2	setter	public	void	set attribute value

Table 75 Class Diagram - DayOfWeek

4.12 Location

Attribute	Type	Visibility	Description
id	Long	private	Unique identifier of location
formattedAddress	String	private	The attribute represents the location full address

longitude	String	private	The attribute represents the longitude of a location
latitude	Double	private	The attribute represents the latitude of a location

No	Method	Visibility	Return Type	Description
1	getter	public	Attribute Type	get attribute value
2	setter	public	void	set attribute value

Table 76 Class Diagram - Location

4.13 BookingComment

Attribute	Type	Visibility	Description
id	Long	private	Unique identifier of booking comment
comment	String	private	The attribute represents the comment of user
rating	Float	private	The attribute represents the rating of user
commentedAt	Date	private	The attribute represents the commented date of a booking

No	Method	Visibility	Return Type	Description
1	getter	public	Attribute Type	get attribute value
2	setter	public	void	set attribute value

Table 77 Class Diagram - BookingComment

4.14 BookingDetail

Attribute	Type	Visibility	Description
id	Long	private	Unique identifier of booking detail
serviceName	String	private	The attribute represents the booked service's name

serviceDescription	String	private	The attribute represents the booked service's description
--------------------	--------	---------	---

No	Method	Visibility	Return Type	Description
1	getter	public	Attribute Type	get attribute value
2	setter	public	void	set attribute value

Table 78 Class Diagram - BookingDetail

4.15 BusyDay

Attribute	Type	Visibility	Description
id	Long	private	Unique identifier of busy day
startDate	Date	private	The attribute represents the start date of photographer's busy event
endDate	Date	private	The attribute represents the end date of photographer's busy event
title	String	private	The attribute represents the title of photographer's busy event
description	String	private	The attribute represents the description of this busy event

No	Method	Visibility	Return Type	Description
1	getter	public	Attribute Type	get attribute value
2	setter	public	void	set attribute value

Table 79 Class Diagram - BusyDay

4.16 Thread

Attribute	Type	Visibility	Description
id	Long	private	Unique identifier of thread
title	String	private	The attribute represents the title of thread
content	String	private	The attribute represents the content of a thread
createdAt	Date	private	The attribute represents the created date of a thread
updatedAt	Date	private	The attribute represents the updated date of a thread

No	Method	Visibility	Return Type	Description
1	getter	public	Attribute Type	get attribute value
2	setter	public	void	set attribute value

Table 80 Class Diagram - Thread

4.17 ThreadTopic

Attribute	Type	Visibility	Description
id	Long	private	Unique identifier of thread topic
topic	String	private	The attribute represents the topic name of a thread
createdBy	String	private	The attribute represents the name of the person who created this topic
createdAt	Date	private	The attribute represents the created date of a thread
updateAt	Date	private	The attribute represents the updated date of a thread

No	Method	Visibility	Return Type	Description

1	getter	public	Attribute Type	get attribute value
2	setter	public	void	set attribute value

Table 81 Class Diagram - ThreadTopic

4.18 ThreadComment

Attribute	Type	Visibility	Description
id	Long	private	Unique identifier of thread comment
comment	String	private	The attribute represents the comment of a thread
createdAt	Date	private	The attribute represents the created date of this record
updateAt	Date	private	The attribute represents the updated date of this record

No	Method	Visibility	Return Type	Description
1	getter	public	Attribute Type	get attribute value
2	setter	public	void	set attribute value

Table 82 Class Diagram - ThreadComment

4.19 ReturningType

Attribute	Type	Visibility	Description
id	Long	private	Unique identifier of returning type
type	EReturningType	private	The attribute represents the type

No	Method	Visibility	Return Type	Description
1	getter	public	Attribute Type	get attribute value
2	setter	public	void	set attribute value

Table 83 Class Diagram - ReturningType

4.20 EReturningType

Attribute	Type	Visibility	Description
LINK			Enum attribute
OFFLINE			Enum attribute

No	Method	Visibility	Return Type	Description
1	getter	public	Attribute Type	get attribute value
2	setter	public	void	set attribute value

Table 84 Class Diagram - EReturningType

4.21 TimeLocationDetail

This table used for saving time and location details of a booking

Attribute	Type	Visibility	Description
id	Long	private	Unique identifier of time location detail
lat	Double	private	The attribute represents the latitude of a user
lon	Double	private	The attribute represents the longitude of a user
formattedAddress	String	private	The attribute represents the formatted address of a user
start	Date	private	The attribute represents start time of a booking
end	Date	private	The attribute represents end time of a booking

No	Method	Visibility	Return Type	Description
1	getter	public	Attribute Type	get attribute value
2	setter	public	void	set attribute value

Table 85 Class Diagram – TimeLocationDetail

4.22 Variable

Attribute	Type	Visibility	Description
id	Long	private	Unique identifier of variable
name	String	private	The attribute represents the name of the variable
weight	Double	private	The attribute represents the weight of the variable

No	Method	Visibility	Return Type	Description
1	getter	public	Attribute Type	get attribute value
2	setter	public	void	set attribute value

Table 86 Class Diagram - Variable

5. Data & Database Design

5.1 Database Design

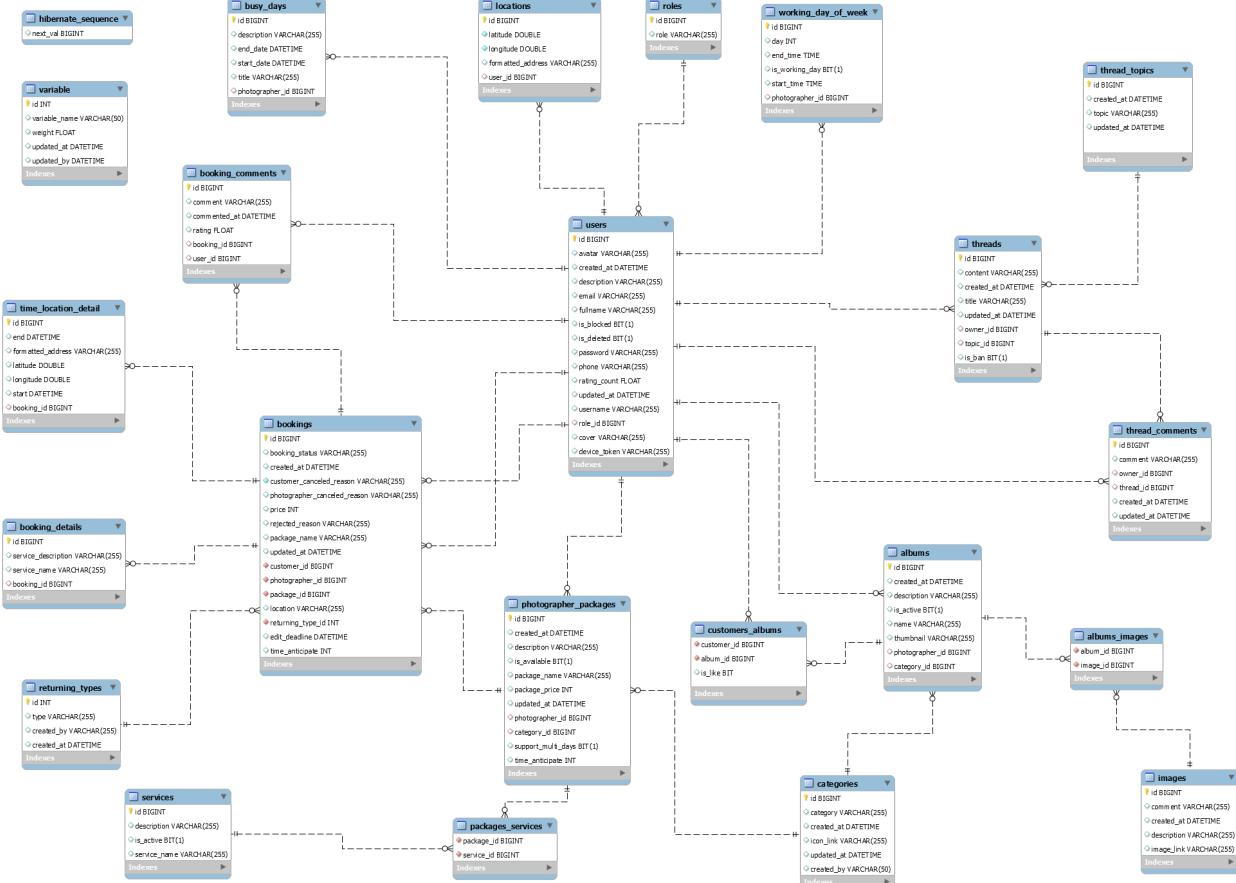


Figure 78 Database Design

5.1.1 users

#	Field name	Type	Size	Unique	Not Null	PK/FK	Notes
1	id	bigint		X	X	X	
2	username	varchar	255	X	X		
3	password	varchar	255		X		
4	fullname	varchar	255		X		
5	description	varchar	255				
6	avatar	varchar	255				Avatar image link of user
7	phone	varchar	255				
8	email	varchar	255				
9	device_token	varchar	255				The firebase token of device, using for firebase messaging.
9	is_blocked	bit					
10	is_active	bit					
11	created_at	datetime					
12	updated_at	datetime					
13	role_id	bigint			X	X	
14	cover	varchar	255				

Table 87 Database - users

5.1.2 roles

#	Field name	Type	Size	Unique	Not Null	PK/FK	Notes
1	id	int		X	X	X	
2	role	varchar	255		X		

Table 88 Database - roles

5.1.3 photographer_packages

#	Field name	Type	Size	Unique	Not Null	PK/FK	Notes
1	id	bigint		X	X	X	
2	package_name	varchar	255		X		
3	package_price	int			X		
4	is_available	bit	1				

5	description	varchar	255				
6	photographer_id	bigint			X	X	
7	category_id	bigint			X	X	
8	support_multiple_days	bit	1				This field represents whether or not a package is supporting multiple day booking.
9	time_anticipate	int					The shooting time of a package (in seconds).
10	created_at	datetime					
11	updated_at	datetime					

Table 89 Database - photographer_packages

5.1.4 packages_services

#	Field name	Type	Size	Unique	Not Null	PK/FK	Notes
1	package_id	Int			X	X	
2	service_id	Int			X	X	

Table 90 Database - packages_services

5.1.5 services

#	Field name	Type	Size	Unique	Not Null	PK/FK	Notes
1	id	bigint		X	X	X	
2	name	varchar	255		X		
3	description	varchar	255				
4	is_active	bit	1				

Table 91 Database - services

5.1.6 bookings

#	Field name	Type	Size	Unique	Not Null	PK/FK	Notes
1	id	bigint		X	X	X	
2	booking_status	varchar	255		X		There will be 6 status: Pending, Ongoing, Editing, Done, Rejected, Canceled
3	package_name	varchar	255		X		Package's name
4	price	int			X		Package's price

5	customer_canceled_reason	varchar	255				In case customers cancel and provide their reason
6	photographer_canceled_reason	varchar	255				In case photographers cancel and provide their reason
7	rejected_reason	varchar	255				In case photographers reject the request and provide reason
8	customer_id	bigint			X	X	
9	photographer_id	bigint			X	X	
10	returning_type_id	int			X	X	
11	package_id	bigint			X	X	
12	edit_deadline	datetime					The datetime customer wants to have their pictures done.
13	time_anticipate	int					Package's shooting time (in seconds)
14	created_at	datetime					
15	updated_at	datetime					

Table 92 Database - bookings

5.1.7 albums

#	Field name	Type	Size	Unique	Not Null	PK/FK	Notes
1	id	bigint		X	X	X	
2	name	varchar	255		X		
3	description	varchar	255				
4	created_at	datetime					
5	thumbnail	varchar	255				The thumbnail link of the album.
6	is_active	bit	1				
7	photographer_id	bigint			X	X	

8	category_id	bigint				X	
---	-------------	--------	--	--	--	---	--

Table 93 Database - albums

5.1.8 locations

#	Field name	Type	Size	Unique	Not Null	PK/FK	Notes
1	id	bigint		X	X	X	
2	formatted_address	varchar	255		X		Location full address.
3	longitude	double			X		
4	latitude	double			X		
5	user_id	bigint			X	X	

Table 94 Database - locations

5.1.9 categories

#	Field name	Type	Size	Unique	Not Null	PK/FK	Notes
1	id	bigint		X	X	X	
2	category	varchar	255		X		
3	icon_link	varchar	255		X		Link for the icon of category
4	created_by	varchar	255				Name of the admin who created the category
5	created_at	datetime					
6	updated_at	datetime					

Table 95 Database - categories

5.1.10 album_images

#	Field name	Type	Size	Unique	Not Null	PK/FK	Notes
1	album_id	bigint			X	X	
2	image_id	bigint			X	X	

Table 96 Database - album_images

5.1.11 images

#	Field name	Type	Size	Unique	Not Null	PK/FK	Notes
1	id	bigint		X	X	X	
2	image_link	varchar	255		X		Link of the image.
3	description	varchar	255				
4	created_at	datetime					

5	comment	varchar	255		X	X	
---	---------	---------	-----	--	---	---	--

Table 97 Database - images

5.1.12 *customers_albums*

#	Field name	Type	Size	Unique	Not Null	PK/FK	Notes
1	is_like	bit					represent if customer has liked the album
2	album_id	bigint			X	X	
3	customer_id	bigint			X	X	

Table 98 Database - customer_albums

5.1.13 *working_day_of_week*

#	Field name	Type	Size	Unique	Not Null	PK/FK	Notes
1	id	bigint		X	X	X	
2	day	int					Day represents week day from Monday to Sunday. 1 is Sunday and 7 is Saturday.
3	start_time	time					
4	end_time	time					
5	is_working_day	bit	1				
6	photographer_id	bigint			X	X	

Table 99 Database - working_day_of_week

5.1.14 *busy_days*

#	Field name	Type	Size	Unique	Not Null	PK/FK	Notes
1	id	bigint		X	X	X	
2	description	varchar	255				
3	start_date	datetime					
4	end_date	datetime					
5	title	varchar	255				
6	photographer_id	bigint			X	X	

Table 100 Database - busy_days

5.1.15 *returning_types*

#	Field name	Type	Size	Unique	Not Null	PK/FK	Notes
1	id	int		X	X	X	
2	type	varchar	255		X		
3	created_at	datetime					
4	created_by	varchar	255		X		Name of the admin who created the type.

Table 101 Database - *returning_types*

5.1.16 *booking_details*

#	Field name	Type	Size	Unique	Not Null	PK/FK	Notes
1	id	bigint		X	X	X	
2	service_name	varchar	255		X		
3	service_descript ion	varchar	255		X		
4	booking_id	bigint			X	X	

Table 102 Database - *booking_details*

5.1.17 *time_location_details*

#	Field name	Type	Size	Unique	Not Null	PK/FK	Notes
1	id	bigint		X	X	X	
2	formatted_addr ess	varchar	255				Booking location full address.
3	latitude	double					
4	longitude	double					
5	start	datetime					Start time of the booking.
6	end	datetime					End time of the booking.
7	booking_id	bigint			X	X	

Table 103 Database - *time_location_details*

5.1.18 *booking_comments*

#	Field name	Type	Size	Unique	Not Null	PK/FK	Notes
1	id	bigint		X	X	X	
2	comment	varchar	255				
3	commented_at	datetime					
4	rating	float					Rating from 0 to 5.

5	user_id	bigint					
6	booking_id	bigint					

Table 104 Database - booking_comments

5.1.19 threads

#	Field name	Type	Size	Unique	Not Null	PK/FK	Notes
1	id	bigint		X	X	X	
2	title	varchar	255				
3	content	varchar	255				
4	update_at	datetime					
5	created_at	datetime					
6	topic_id	bigint			X	X	
7	owner_id	bigint			X	X	
8	is_ban	bit	1				

Table 105 Database - threads

5.1.20 thread_comments

#	Field name	Type	Size	Unique	Not Null	PK/FK	Notes
1	id	bigint		X	X	X	
2	comment	varchar	255				
3	update_at	datetime					
4	created_at	datetime					
5	thread_id	bigint				X	
6	owner_id	bigint				X	

Table 106 Database - thread_comments

5.1.21 thread_topics

#	Field name	Type	Size	Unique	Not Null	PK/FK	Notes
1	id	bigint		X	X	X	
2	topic	varchar	255				
3	created_by	varchar	255				
4	update_at	datetime					
5	created_at	datetime					

Table 107 Database - thread_topics

5.1.22 variables

#	Field name	Type	Size	Unique	Not Null	PK/FK	Notes
1	id	int		X	X	X	
2	variable_name	varchar	255				
3	weight	float					Weight from 0 to 1.
4	updated_at	datetime					
5	updated_by	datetime					Name of the admin who changed the variable.

Table 108 Database - variables

5.2 Data File Design

We use Aws S3 to store files and images.

#	File Name	Type	Notes
1	User avatar	Image	url: "{baseUrl}/api/users/{userId}/{userId}-avatar"
2	Photographer album thumbnail	Image	url: "{baseUrl}/api/photographers/{ptgId}/{albumId}/{albumId}-thumbnail"
3	Photographer album image	Image	url: "{baseUrl}/api/photographers/{ptgId}/{albumId}/{imageId}-image"
4	Photographer cover image	Image	url: "{baseUrl}/api/photographers/{ptgId}/{ptgId}-cover"
5	Category icon image	Image	url: "{baseUrl}/api/categories/{categoryId}-category"

Table 109 Date File 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 = $(1 - \text{distance} / \text{sum}) * \text{max scale}$
(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 = $(1 - \text{price} / \text{sum}) * \text{max scale}$
(Photographer with higher price will get lower score)

Step 5: Sum the scores

Score of a photographer = $0.3 * \text{rating score} + 0.3 * \text{distance score} + 0.4 * \text{price score}$

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

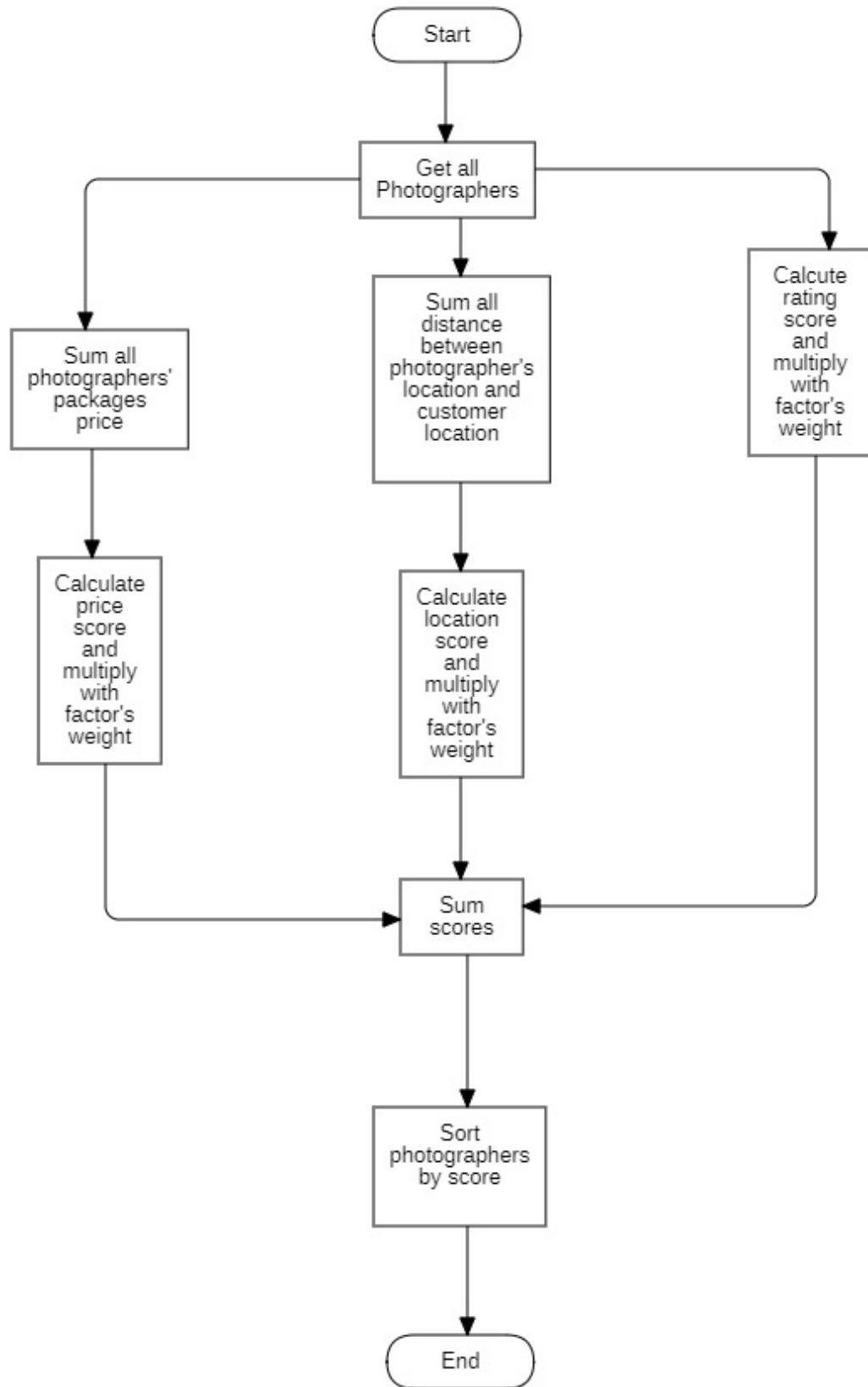


Figure 79 Flowchart - Multiple factors sorting

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:

$$\text{Bayes' Theorem: } P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

$P(A|B)$ - 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 $x = \{x_1, x_2, \dots, x_n\}$, in our case $x = \{\text{Outlook, Temperature, Humidity, Windy}\}$
- A fixed set of classes $C = \{c_1, c_2, \dots, c_n\}$, in our case $C = \{\text{Yes, No}\}$

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.

$$P(c|x) = \frac{P(x|c)P(c)}{P(x)}$$

Or we can simply calculate in expanded form:

$$P(c | X) = P(x_1 | c) \times P(x_2 | c) \times \cdots \times P(x_n | c) \times P(c)$$

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:

$$P(\text{Yes}) = \frac{3}{7}$$

$$P(\text{No}) = \frac{4}{7}$$

$$P(\text{Sunny} | \text{Yes}) = \frac{1}{3}$$

$$P(\text{Sunny} | \text{No}) = \frac{1}{4}$$

$$P(\text{Temperature} = \text{High} | \text{Yes}) = \frac{0}{3}$$

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

$$\rightarrow P(\text{Temperature} = \text{High} | \text{Yes}) = \frac{1}{3}$$

$$P(\text{Temperature} = \text{High} | \text{No}) = \frac{4}{4}$$

$$P(\text{Humidity} = \text{High} | \text{Yes}) = \frac{1}{3}$$

$$P(\text{Humidity} = \text{High} | \text{No}) = \frac{2}{4}$$

$$P(\text{Windy} = \text{Strong} | \text{Yes}) = \frac{0}{3}$$

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

$$\rightarrow P(\text{Windy} = \text{Strong} | \text{Yes}) = \frac{1}{3}$$

$$P(\text{Windy} = \text{Normal} | \text{No}) = \frac{4}{4}$$

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

- $P(\text{Yes} | X)$:

$$\begin{aligned} & P(\text{Yes}) \times P(\text{Yes}) \times P(\text{HighHumidity} | \text{Yes}) \times P(\text{StrongWindy} | \text{Yes}) \times P(\text{Yes}) \\ &= \frac{1}{3} \times \frac{1}{3} \times \frac{1}{3} \times \frac{1}{3} \times \frac{3}{7} \sim 0.00529 \end{aligned}$$

- $P(No | X)$:

$$\begin{aligned} & P(No) \times P(No) \times P(HighHumidity|No) \times P(StrongWindy|No) \times P(No) \\ & = \frac{1}{4} \times \frac{4}{4} \times \frac{2}{4} \times \frac{4}{4} \times \frac{4}{7} \sim 0.0714 \end{aligned}$$

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.

V. Software Testing Documentation

1. Overall Description

1.1 Test Model

We use a software development process model is V-model because this model is suitable for project conditions. For each stage, there will be a corresponding check it helps us to detect errors from the beginning.

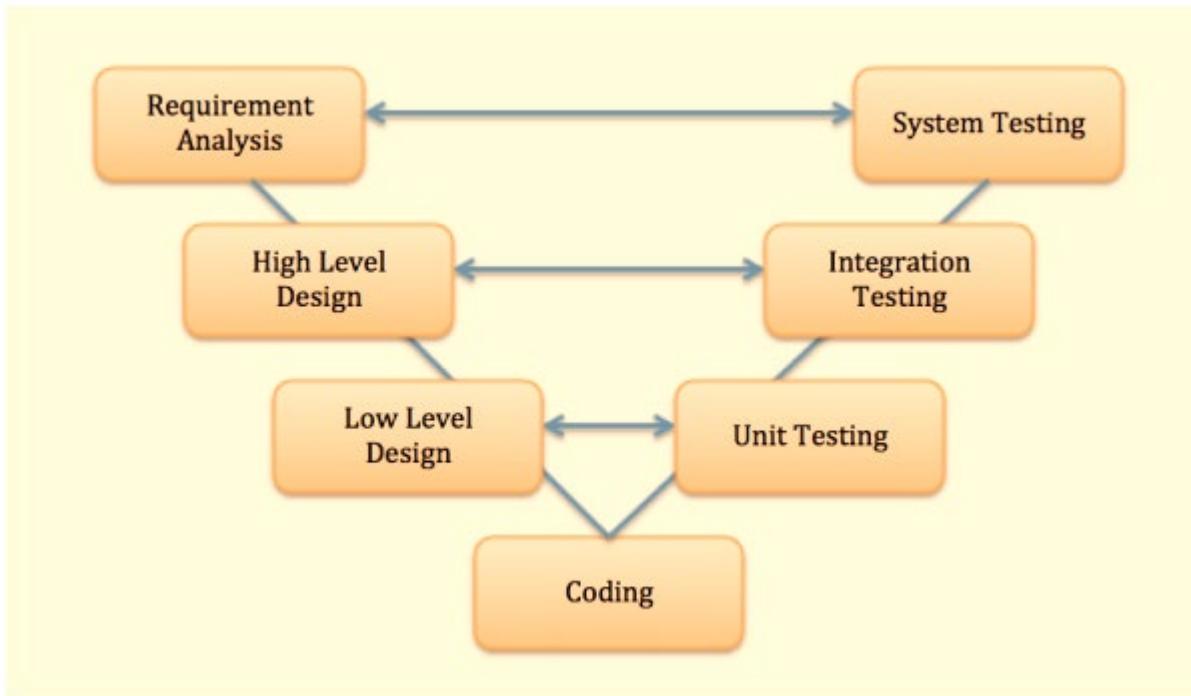


Figure 80 Test Model

References: [https://en.wikipedia.org/wiki/V-Model_\(software_development\)](https://en.wikipedia.org/wiki/V-Model_(software_development))

1.2 Testing Levels

Testing Level	Description
Unit Testing	Checks if software components are fulfilling functionalities or not.
Integration Testing	Checks the data flow from one module to other modules.
System Testing	Evaluates both functional and non-functional needs for the testing.
Acceptance Testing	Checks the requirements of a specification or contract are met as per its delivery.

Table 110 Testing Levels

1.3 Testing Types

Testing types	Description
Functional Testing	Include: <ul style="list-style-type: none">• Unit Testing

	<ul style="list-style-type: none"> • Integration Testing • System Testing • Interface Testing • Regression Testing
Non-functional Testing	<p>Include:</p> <ul style="list-style-type: none"> • Performance Testing • Load Testing

Table 111 Testing Types

2. Test Plan

2.1 Test Stages

Type of Test	Stage of Test			
	Unit	Integration	System	Acceptance
Interface test	X		X	
Performance Testing	X	X		
Load Test			X	X

Table 112 Test Stages

2.2 Resources

a. Human Resources

Worker/Doer	Role	Specific Responsibilities/Comments
Trần Thiên Thảo	Leader	<ul style="list-style-type: none"> - Functional Testing - Non-functional Testing
Bồ Công Đạt	Member	<ul style="list-style-type: none"> - Functional Testing - Non-functional Testing
Đào Sỹ Trung Kiên	Member	<ul style="list-style-type: none"> - Functional Testing - Non-functional Testing
Trương Ngọc Mỹ	Member	<ul style="list-style-type: none"> - Functional Testing - Non-functional Testing

Table 113 Human Resources

b. Environment

Purpose	Tool	Provider	Version
Unit Test	Postman	Postman, Inc.	7.36.0
Unit Testing	Chrome	Google LLC	87.0.4280.66
Integration Testing	Android phone	Sony Corporation	XZ1
System Testing			

Interface Testing			
Regression Testing			
Performance Testing			
Load Testing			

Table 114 Environment

2.3 Test Milestones

Milestone Task	Effort (md)	Start Date	End Date
Unit test case	25	20/11/2020	15/12/2020
System test case	25	20/11/2020	15/12/2020
Test procedures	8	05/12/2020	13/12/2020
Defect reports	5	15/12/2020	20/12/2020

Table 115 Test Milestones

2.4 Deliverables

No	Deliverables	Due Date
1	Unit test case	15/12/2020
2	System Test case	15/12/2020

Table 116 Deliverables

3. Test Cases

Test cases are included in these two excel files:

No	Filename
1	Unit Test Cases: Report5_Unit_Test_Case.xls
2	System Test case

Table 117 Test Cases

4. Test Reports

Test reports are included in this file:

No	Filename
1	Report5_Test_Case_Document.xlsx

Table 118 Test Reports

VI. Release Package & User Guides

1. Deliverable Package

1.1 Source codes & documents

No.	Items	Sub-Items	Type	Version
Code Package				
1	Web API	pbs_webapi	New	1.0.1
2	Mobile Apps	photographer_mobile_app	New	1.0.1
		customer_mobile_app	New	1.0.1
Database				
1	Tables	pbs_users.sql	New	1.0.1
		pbs_roles.sql	New	1.0.1
		pbs_photographer_packages.sql	New	1.0.1
		pbs_packages_services.sql	New	1.0.1
		pbs_services.sql	New	1.0.1
		pbs_returning_types.sql	New	1.0.1
		pbs_bookings.sql	New	1.0.1
		pbs_booking_details.sql	New	1.0.1
		pbs_booking_time_location_detail.sql	New	1.0.1
		pbs_booking_comments.sql	New	1.0.1
		pbs_albums.sql	New	1.0.1
		pbs_customers_albums.sql	New	1.0.1
		pbs_album_images.sql	New	1.0.1
		pbs_images.sql	New	1.0.1
		pbs_categories.sql	New	1.0.1
		pbs_threads.sql	New	1.0.1
		pbs_thread_comments.sql	New	1.0.1
		pbs_thread_topics.sql	New	1.0.1
		pbs_locations.sql	New	1.0.1
		pbs_working_day_of_week.sql	New	1.0.1
		pbs_busy_days.sql	New	1.0.1
Documents				
1	Introduction	Report1_Project-Introduction.docx	New	1.0.1
2	Management Plan	Report2_Project-Management-Plan.docx	New	1.0.1
3	Requirement	Report3_Software Requirement Specification.docx	New	1.0.1
4	Design	Report4_Software Design Document.docx	New	1.0.1
5	Testing	Report5_Test Documentation.docx	New	1.0.1
		Report5_Test Case Document.xlsx	New	1.0.1
		Report5_Unit Test Case.xls	New	1.0.1
6	User guides	Report6_Software User Guides.docx	New	1.0.1
7	Final	Report7_Final Project Report.docx	New	1.0.1

Table 119 Deliverable Package

1.2 Known Issues, Limitations & Restrictions

- The application has not supported payment.
- The system does not have a photo verification feature.
- The system has not provided a website version to both customers and photographers. Users may find it uncomfortable to use applications.

2. Installation Guides

2.1 System Requirements

2.1.1 Hardware requirements

For PC

PC	Minimum Requirements	Recommended
Internet Connection	Cable, Wi-Fi (4 Mbps)	Cable, Wi-Fi (8 Mbps)
Operating System	Windows 7	Windows 7 or more.
Computer Processor	Intel® Core i3 1.4GHz	Intel® Core i5 2.50GHz
Computer Memory	1GB RAM	2GB RAM or more
Web Browser	Chromes (v42 or higher)	Chrome latest stable version

Table 120 PC Hardware requirements

For Android Devices

Operating system	Android 8.0 or higher
Processor	Intel Atom® Processor Z2520 1.2 GHz, or faster processor
Storage	Between 850 MB and 1.2 GB, depending on the language version
RAM	Minimum of 512 MB, 2 GB is recommended
Hard Disk	<ul style="list-style-type: none">2 GB of available hard-disk space for installation; extra free space is required during installation.You cannot install using a removable flash storage device.
Video	1920 x 1080 pixels or higher
Browser/Internet	<ul style="list-style-type: none">This application is designed to work online.To download and launch Google Play Store apps within the application, a high-speed Internet connection is recommended.

Table 121 Android Hardware requirements

2.1.2 Software requirements

Software	Name / Version	Description
Operation System	Windows 7, Windows 10	Operating system and platform for development
Web browser	Chrome 69 or above, Mozilla Firefox	For web app
Android System	Android 4.0 or higher	For mobile app

Table 122 Software requirements

2.2 Setup Files

- PBS for photographer.apk

 photographer.apk	12/1/2020 9:44 AM	BlueStacks.Apk	48,802 KB
--	-------------------	----------------	-----------

Figure 81 Setup files - 1

- PBS for customer.apk

 customer.apk	12/1/2020 9:44 AM	BlueStacks.Apk	48,802 KB
--	-------------------	----------------	-----------

Figure 82 Setup files - 2

2.3 Installation Instruction

The installation steps of 2 android apps are the same, so we provide one guideline for both of them.

1. Install PBS by pressing the “Install” button.

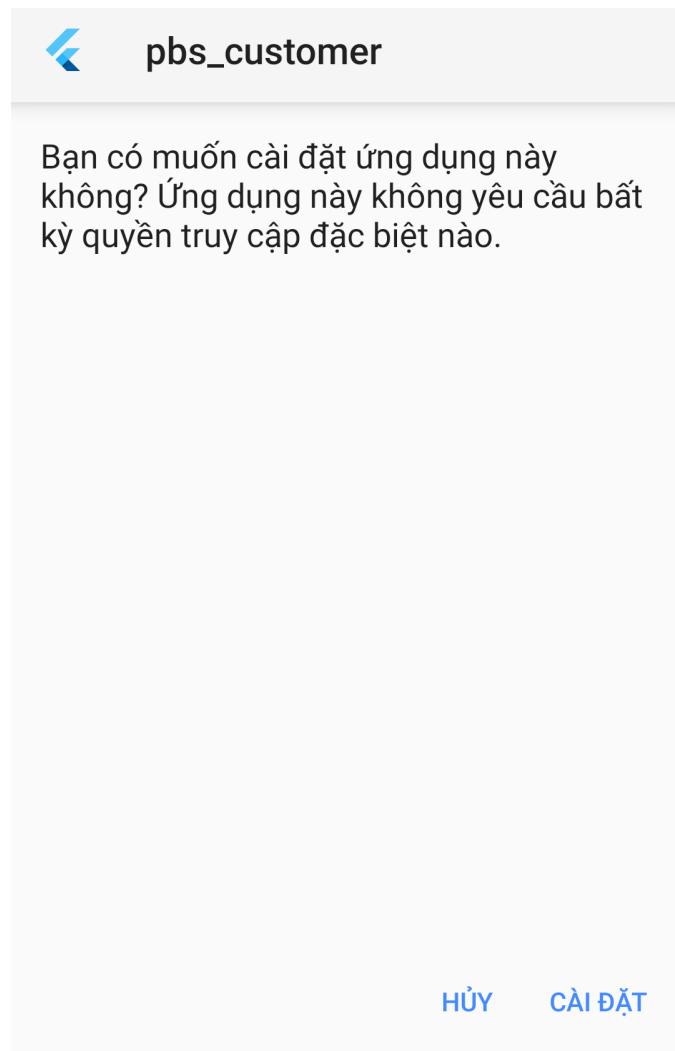


Figure 83 Installation - 1

2. Press “Accept” to accept application permissions.

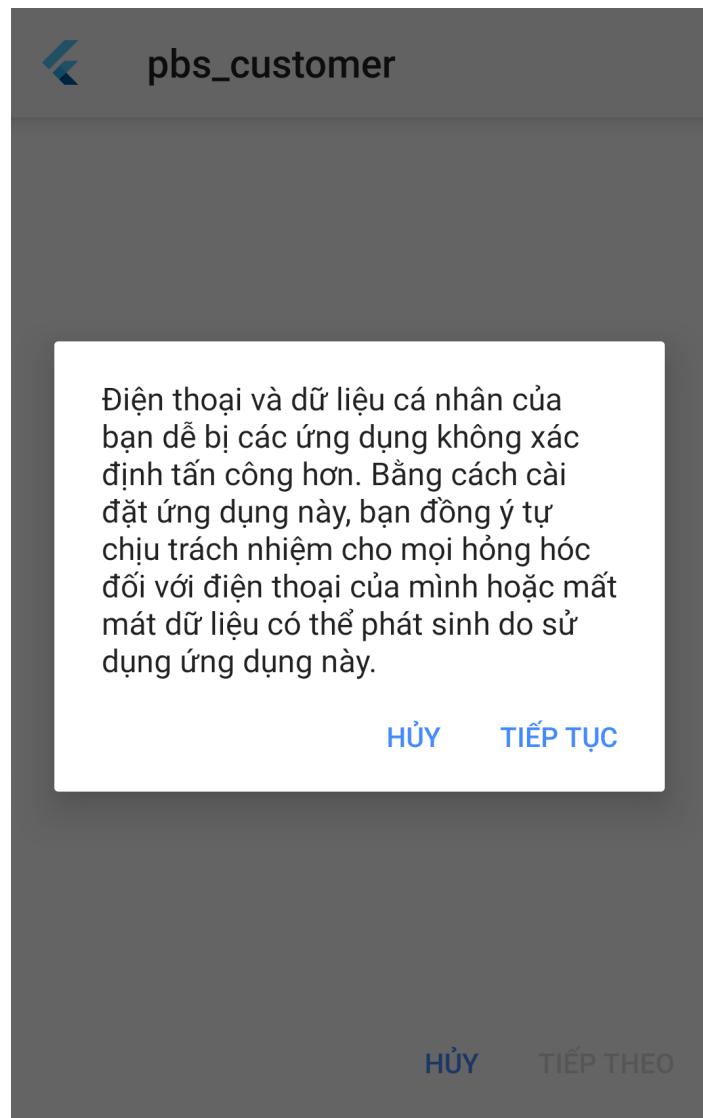


Figure 84 Installation - 2

3. Wait for installation of the PBS App.

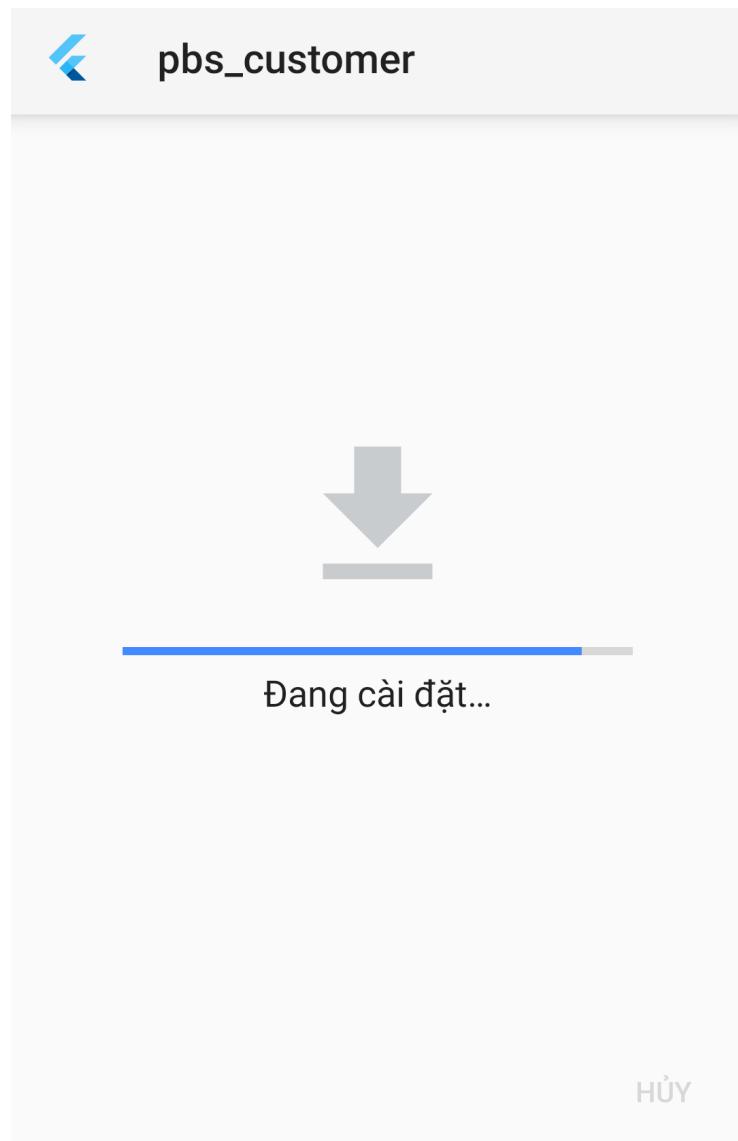


Figure 85 Installation - 3

4. Press “Mở” to start the main App.

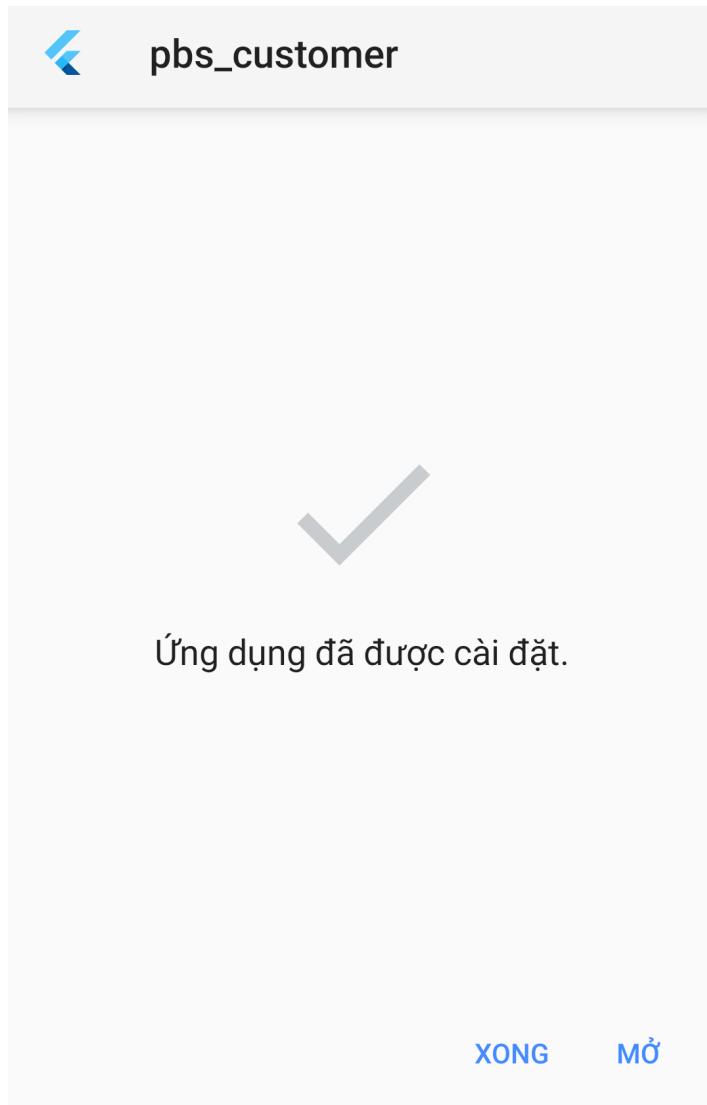


Figure 86 Installation - 4

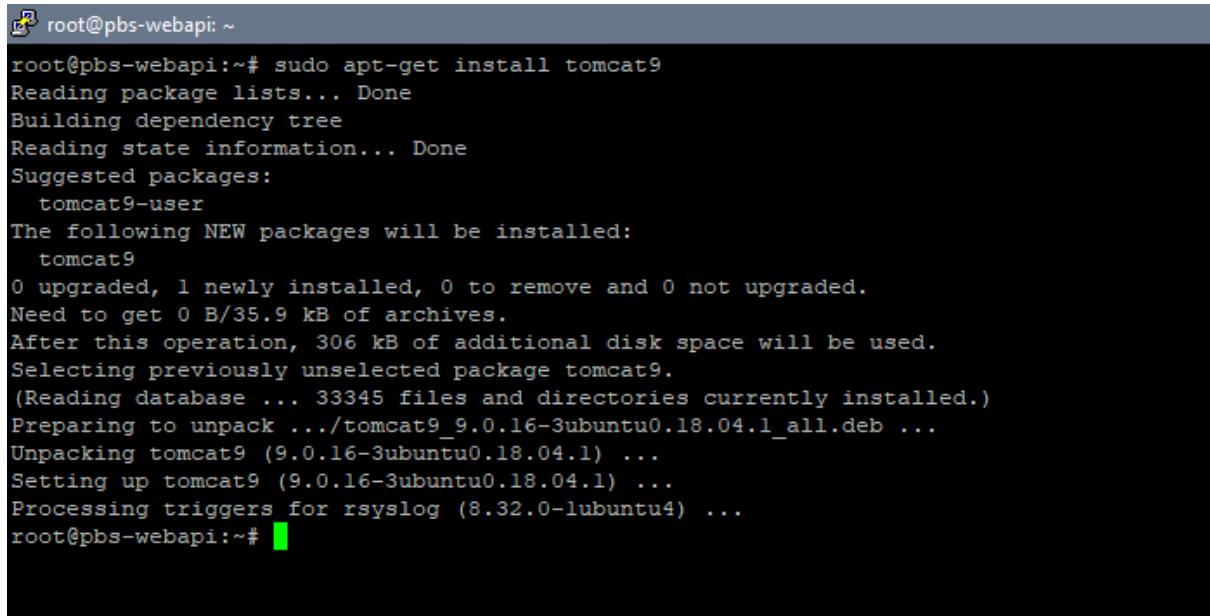
5. Read the manual for more information.

2.4 Server-side deployment

We deploy our webapi to Hostinger's VPS server. Server's operating system is Ubuntu 18.04 LTS.

Step 1: Install Tomcat and JDK. Openjdk will be install as dependency with command:

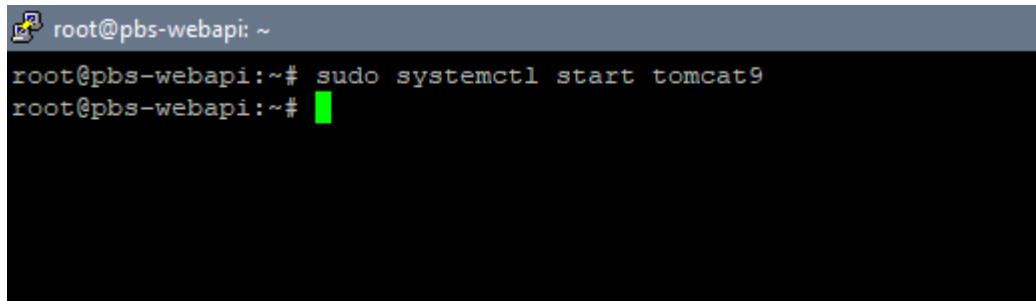
```
sudo apt-get install tomcat9
```



```
root@pbs-webapi:~# sudo apt-get install tomcat9
Reading package lists... Done
Building dependency tree
Reading state information... Done
Suggested packages:
  tomcat9-user
The following NEW packages will be installed:
  tomcat9
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 0 B/35.9 kB of archives.
After this operation, 306 kB of additional disk space will be used.
Selecting previously unselected package tomcat9.
(Reading database ... 33345 files and directories currently installed.)
Preparing to unpack .../tomcat9_9.0.16-3ubuntu0.18.04.1_all.deb ...
Unpacking tomcat9 (9.0.16-3ubuntu0.18.04.1) ...
Setting up tomcat9 (9.0.16-3ubuntu0.18.04.1) ...
Processing triggers for rsyslog (8.32.0-lubuntu4) ...
root@pbs-webapi:~#
```

Figure 87 Server-side deployment – 1

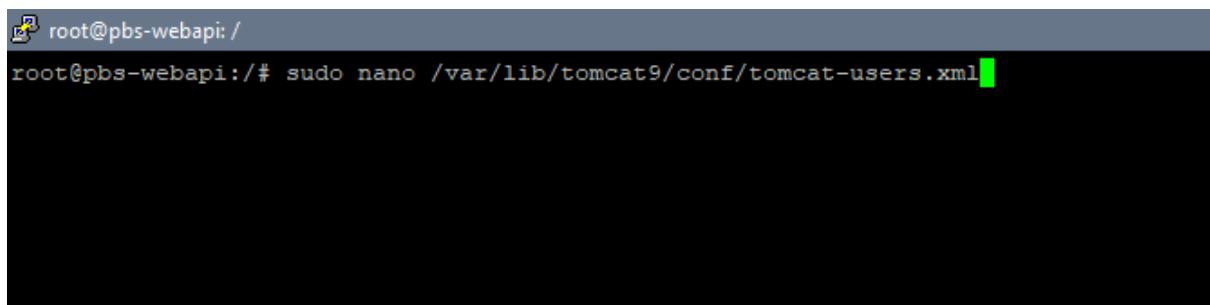
Step 2: Start tomcat service with command: sudo systemctl start tomcat9.



```
root@pbs-webapi:~# sudo systemctl start tomcat9
root@pbs-webapi:~#
```

Figure 88 Server-side deployment – 2

Step 3: Configure tomcat admin account at /var/lib/tomcat9/conf/tomcat-users.xml:



```
root@pbs-webapi:~#
root@pbs-webapi:/# sudo nano /var/lib/tomcat9/conf/tomcat-users.xml
```

Figure 89 Server-side deployment – 3

```
root@pbs-webapi: /  
GNU nano 2.9.3 /var/lib/tomcat9/conf/tomc  
  
<!--  
NOTE: By default, no user is included in the "manager-gui" role required  
to operate the "/manager/html" web application. If you wish to use this app,  
you must define such a user - the username and password are arbitrary. It is  
strongly recommended that you do NOT use one of the users in the commented out  
section below since they are intended for use with the examples web  
application.  
-->  
<!--  
NOTE: The sample user and role entries below are intended for use with the  
examples web application. They are wrapped in a comment and thus are ignored  
when reading this file. If you wish to configure these users for use with the  
examples web application, do not forget to remove the <!... ...> that surrounds  
them. You will also need to set the passwords to something appropriate.  
-->  
<!--  
<role rolename="tomcat"/>  
<role rolename="role1"/>  
<user username="tomcat" password=<must-be-changed> roles="tomcat"/>  
<user username="both" password=<must-be-changed> roles="tomcat,role1"/>  
<user username="role1" password=<must-be-changed> roles="role1"/>  
-->  
<role rolename="manager-gui"/>  
<role rolename="admin-gui"/>  
<user username="username" password="password" roles="manager-gui,admin-gui"/>  
</tomcat-users>
```

Figure 90 Server-side deployment – 4

Step 4: Export webapi war file using “mvn package clean”:

```

C:\Windows\System32\cmd.exe - mvn clean package
Microsoft Windows [Version 10.0.18363.1256]
(c) 2019 Microsoft Corporation. All rights reserved.

D:\Capstone\pbs\webapi>mvn clean package
[INFO] Scanning for projects...
[WARNING]
[WARNING] Some problems were encountered while building the effective model for fpt.university:pbs-webapi:war:0
[WARNING] The expression ${artifactId} is deprecated. Please use ${project.artifactId} instead.
[WARNING]
[WARNING] It is highly recommended to fix these problems because they threaten the stability of your build.
[WARNING]
[WARNING] For this reason, future Maven versions might no longer support building such malformed projects.
[WARNING]
[INFO]
[INFO] -----< fpt.university:pbs-webapi >-----
[INFO] Building pbs-webapi 0.0.1-SNAPSHOT
[INFO] -----[ war ]-----
[INFO]
[INFO] --- maven-clean-plugin:3.1.0:clean (default-clean) @ pbs-webapi ---
[INFO] Deleting D:\Capstone\pbs\webapi\target
[INFO]
[INFO] --- maven-resources-plugin:3.1.0:resources (default-resources) @ pbs-webapi ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] Copying 1 resource
[INFO] Copying 1088 resources
[INFO]
[INFO] --- maven-compiler-plugin:3.8.1:compile (default-compile) @ pbs-webapi ---
[INFO] Changes detected - recompiling the module!
[INFO] Compiling 118 source files to D:\Capstone\pbs\webapi\target\classes
[INFO]
[INFO] --- maven-resources-plugin:3.1.0:testResources (default-testResources) @ pbs-webapi ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] skip non existing resourceDirectory D:\Capstone\pbs\webapi\src\test\resources
[INFO]
[INFO] --- maven-compiler-plugin:3.8.1:testCompile (default-testCompile) @ pbs-webapi ---
[INFO] Changes detected - recompiling the module!
[INFO] Compiling 1 source file to D:\Capstone\pbs\webapi\target\test-classes
[INFO]
[INFO] --- maven-surefire-plugin:2.22.2:test (default-test) @ pbs-webapi ---
[INFO]
[INFO] -----

```

Figure 91 Server-side deployment -5

```

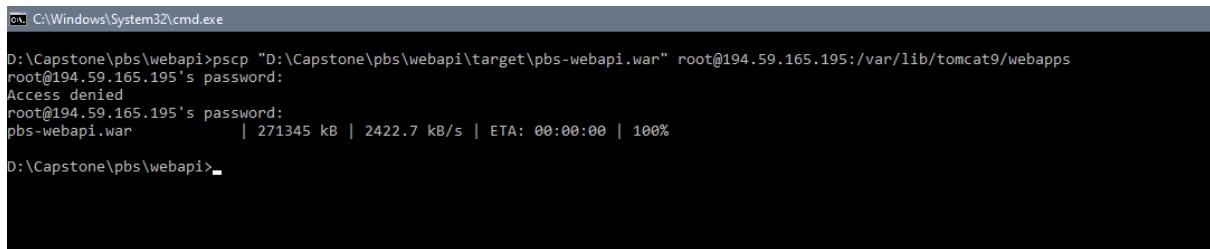
[INFO] --- maven-war-plugin:3.2.3:war (default-war) @ pbs-webapi ---
[INFO] Packaging webapp
[INFO] Assembling webapp [pbs-webapi] in [D:\Capstone\pbs\webapi\target\pbs-webapi]
[INFO] Processing war project
[INFO] Webapp assembled in [7010 msecs]
[INFO] Building war: D:\Capstone\pbs\webapi\target\pbs-webapi.war
[INFO]
[INFO] --- spring-boot-maven-plugin:2.3.3.RELEASE:repackage (repackage) @ pbs-webapi ---
[INFO] Replacing main artifact with repackaged archive
[INFO]
[INFO] BUILD SUCCESS
[INFO]
[INFO] -----
[INFO] Total time: 02:26 min
[INFO] Finished at: 2020-12-13T22:40:22+07:00
[INFO]

```

D:\Capstone\pbs\webapi>

Figure 92 Server-side deployment - 6

Step 5: Last step is to transfer war file to server using pscp via SSH:



```

C:\Windows\System32\cmd.exe
D:\Capstone\pbs\webapi>pscp "D:\Capstone\pbs\webapi\target\pbs-webapi.war" root@194.59.165.195:/var/lib/tomcat9/webapps
root@194.59.165.195's password:
Access denied
root@194.59.165.195's password:
pbs-webapi.war | 271345 kB | 2422.7 kB/s | ETA: 00:00:00 | 100%
D:\Capstone\pbs\webapi>_

```

Figure 93 Server-side deployment -7

3. User Manual

3.1 Terms and definitions

- Accounts are the accounts that you register to sign in as users.
- Photographers' application is an application provided by PBS for photographers to contact customers.
- Customers' application is an application provided by PBS for users to contact photographers.
- Mobile devices are your mobile phones, tablets, or other devices that can be used to download and run Photographers' application and Customers' application.
- System is the system provided by PBS to support photographers' and customers' contact, including the Photographers' application, Customers' application, website, and platform.
- Photographers are independent third-party who provide shooting services that customers can book via the system.
- Customers are the ones who have Customers' application's accounts.
- Website is the website run by PBS.

3.2 System requirements

3.2.1 Hardware requirements

For PC

PC	Minimum Requirements	Recommended
Internet Connection	Cable, Wi-Fi (4 Mbps)	Cable, Wi-Fi (8 Mbps)
Operating System	Windows 7	Windows 7 or higher.
Computer Processor	Intel® Core i3 1.4GHz	Intel® Core i5 2.50GHz
Computer Memory	1GB RAM	2GB RAM or more
Web Browser	Chromes (v42 or higher)	Chrome latest stable version

For Android devices

Operating system	Android 8.0 or higher
Processor	Intel Atom® Processor Z2520 1.2 GHz, or faster processor
Storage	Between 850 MB and 1.2 GB, depending on the language version

RAM	Minimum of 512 MB, 2 GB is recommended
Hard Disk	<ul style="list-style-type: none"> 2 GB of available hard-disk space for installation; extra free space is required during installation. You cannot install using a removable flash storage device.
Video	1920 x 1080 pixels or higher
Browser/Internet	<ul style="list-style-type: none"> This application is designed to work online. To download and launch Google Play Store apps within the application, a high-speed Internet connection is recommended.

3.2.2 Software requirements

Software	Name / Version	Description
Operation System	Windows 7, Windows 10	Operating system and platform for development
Web browser	Chrome 69 or above, Mozilla Firefox	For web app
Android System	Android 4.0 or higher	For mobile app

3.3 Application Usage

3.3.1. Overview

Customers' application allows customers to search for and book photographers. Then customers can rate photographers based on their experiences. The rates are displayed as stars and they can also write reviews, which would be shown on photographers' profiles. To give users more information about photography as well as trendy styles, there is also a forum on the system where they can share information and images.

Photographers' application helps photographers reach a wide range of potential customers. In addition, they can manage their shooting schedule and receive customers' bookings. They can also edit their profile, add albums and services that they offer. They can join the forums as well.

The system also provides admins with a website so that they can easily control the system. Admins can add, delete, and edit shooting categories, and manage photo delivery methods. They can block users who violate community standards.

3.3.2. Register

- Purpose**
 - Create users' accounts
- Function**

- Register for PBS account
- **User Interfaces**

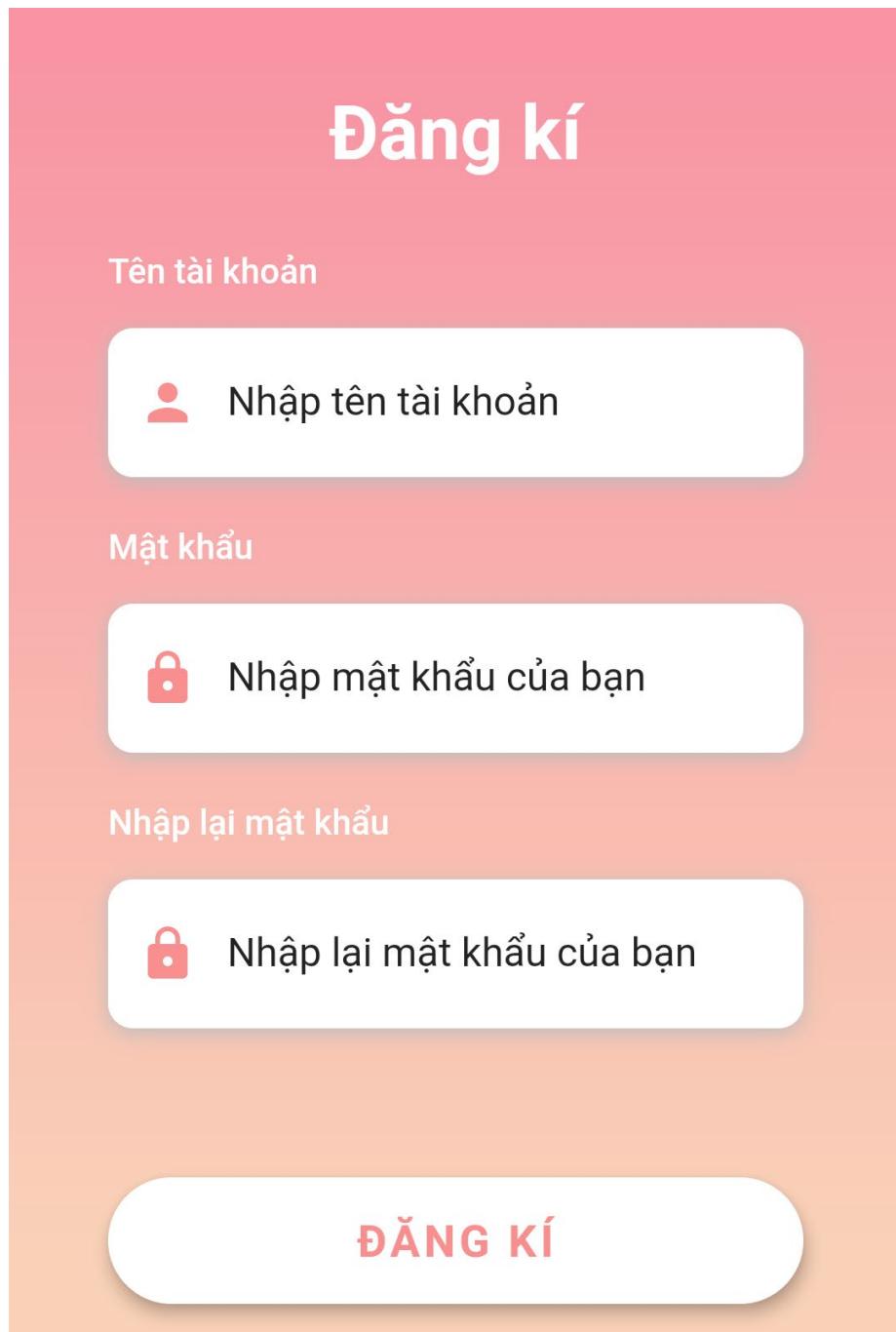


Figure 94 Register

3.3.3. Login

- **Purpose**
 - Allow users to sign in
- **Function**
 - Sign in with PBS account
- **User Interfaces**

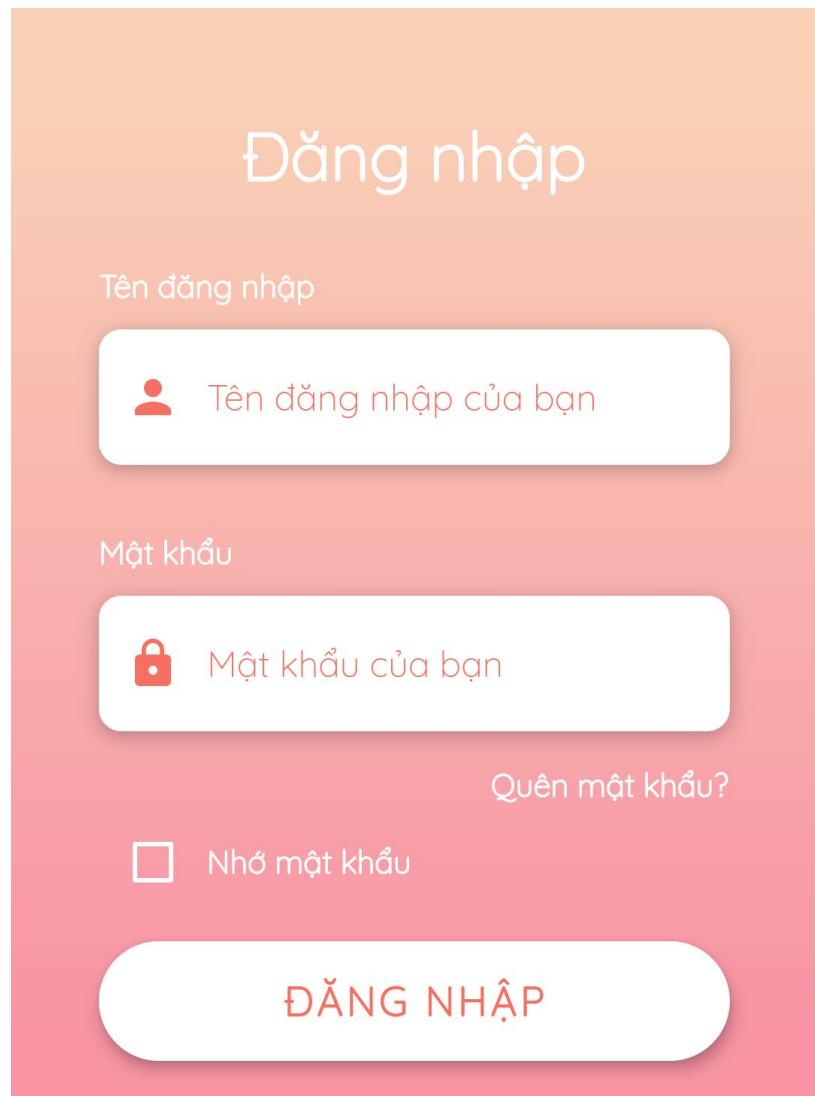


Figure 95 - Login

3.3.4. Logout

- **Purpose**
 - Allow users to sign out
- **Function**
 - Sign out
- **User Interfaces**

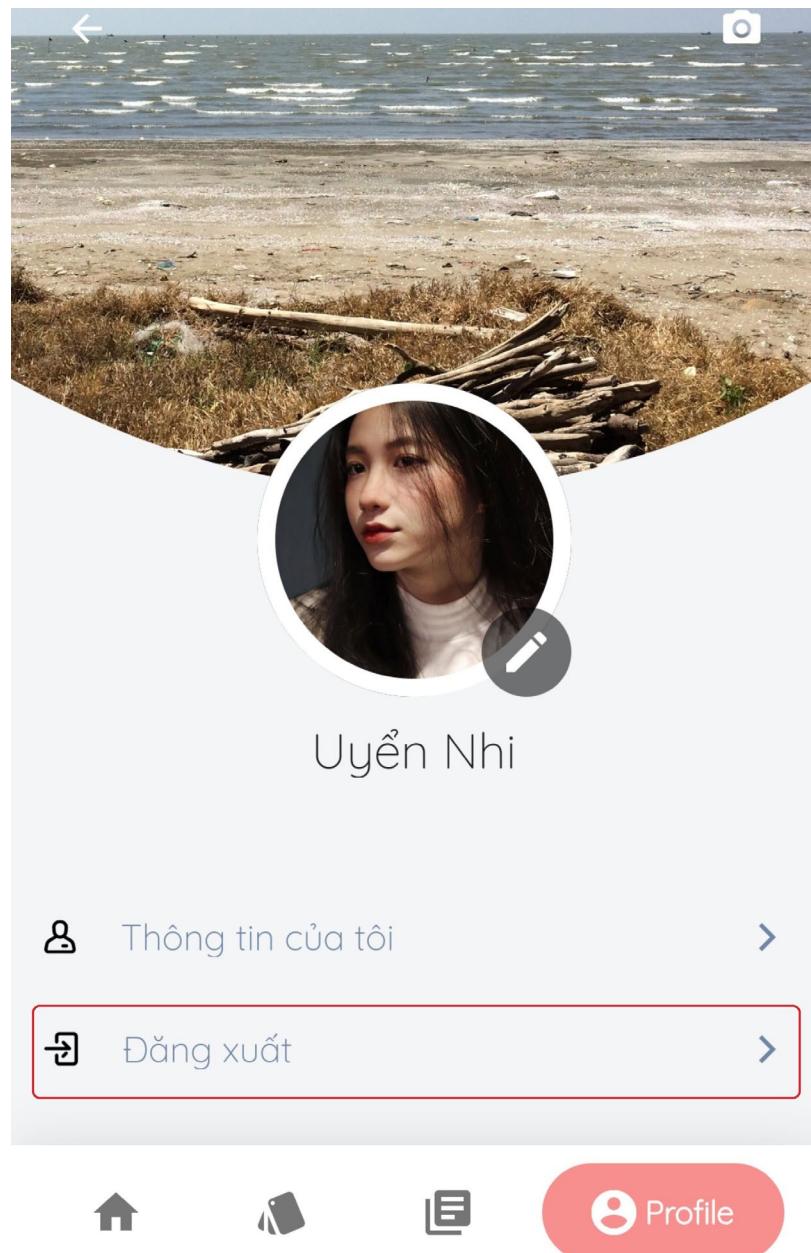


Figure 96 - Logout

3.3.5. Search photographer

- **Purpose**
 - Allow customers to find photographers and appropriate services
- **Function**
 - Search for photographers
 - Search for services
- **User Interfaces**

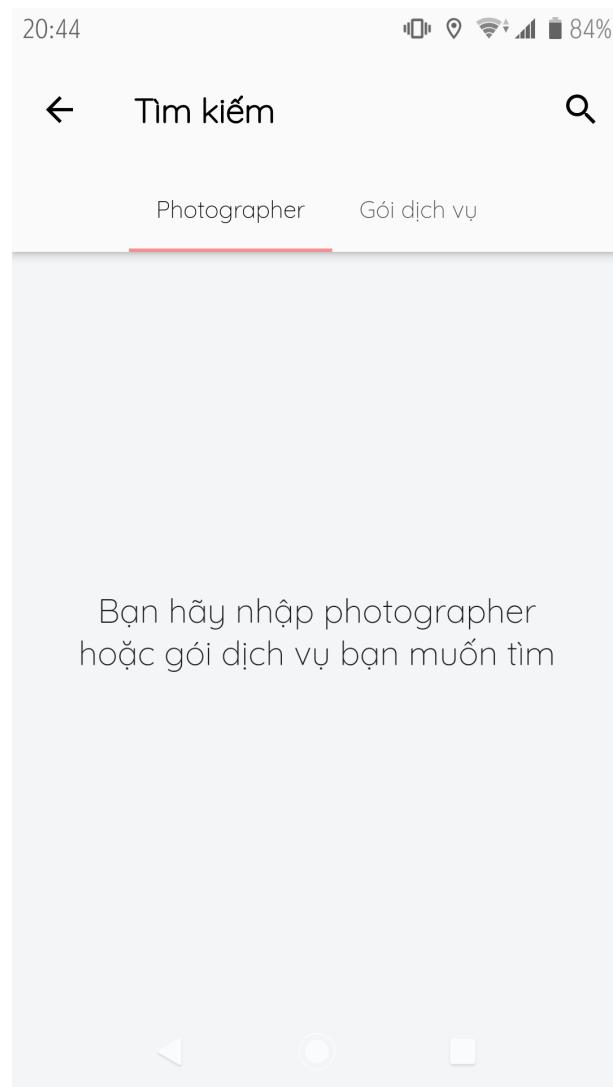


Figure 97 Search photographers

3.3.6. Manage request

- **Purpose**
 - Allow customers to manage their bookings
- **Function**
 - Make request for days (customers only)
 - Make request for a day (customers only)
 - Make changes to their bookings (customers only)
 - Cancel their bookings (customers only)
 - Accept bookings (photographers only)
 - Decline bookings (photographers only)
- **User Interfaces**



Figure 98 - Make request for a day

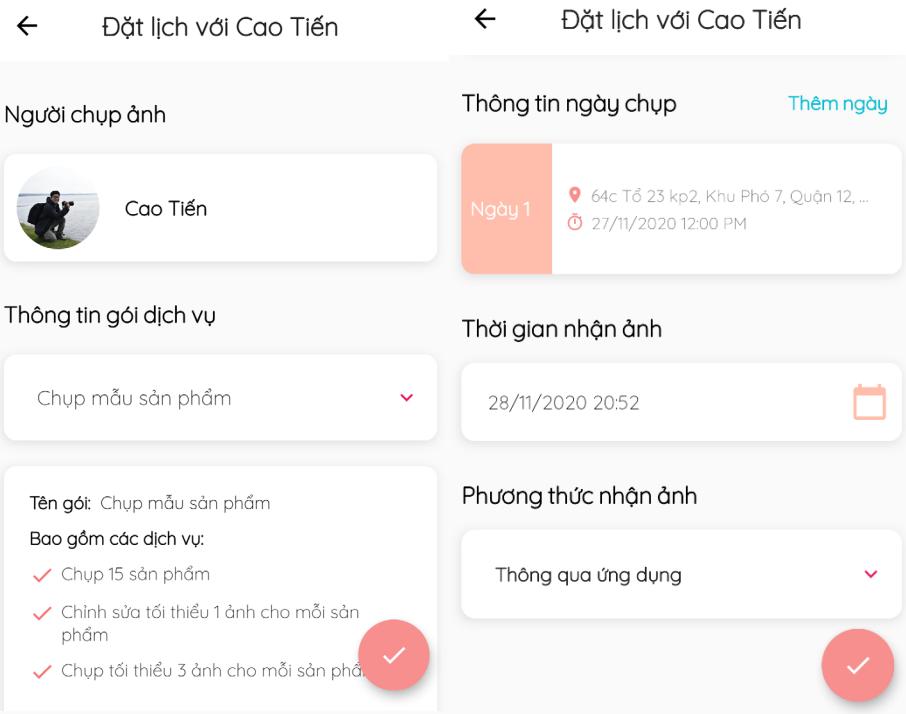


Figure 99 - Make request for days

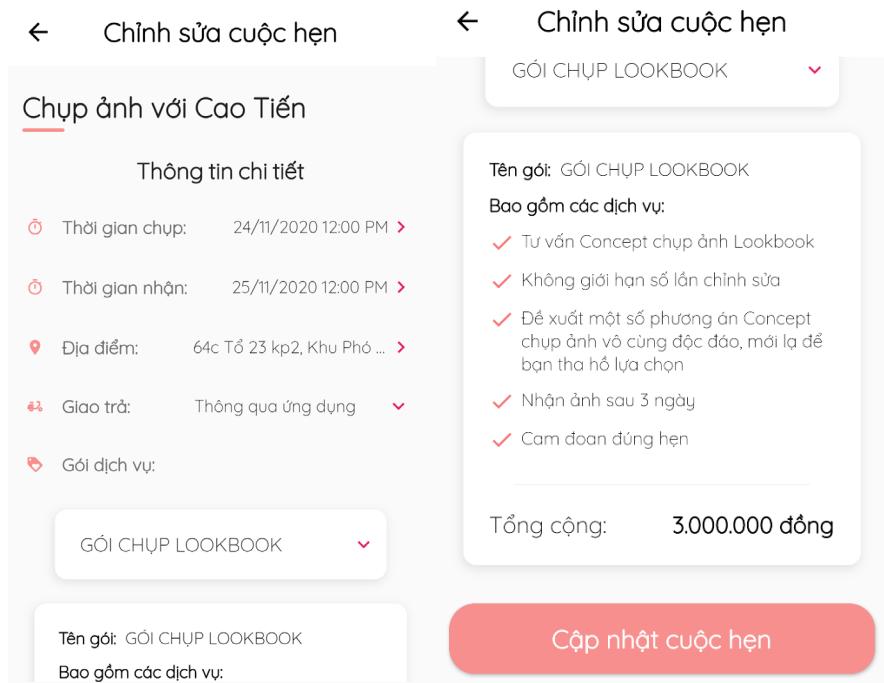


Figure 100 - Update booking for a day

← Đặt lịch với Cao Tiến

Người chụp ảnh

Cao Tiến

← Đặt lịch với Cao Tiến

Thông tin ngày chụp

Ngày 1
📍 88C Tô 21 KP2, Khu Phố 7, Quận 12,...

Thêm ngày

Thông tin gói dịch vụ

Chụp mẫu sản phẩm
▼

Thời gian nhận ảnh

25/11/2020 12:00 PM
📅

Tên gói: Chụp mẫu sản phẩm

Bao gồm các dịch vụ:

- Chụp 15 sản phẩm
- Chỉnh sửa tối thiểu 1 ảnh cho mỗi sản phẩm
- Chụp tối thiểu 3 ảnh cho mỗi sản phẩm

Phương thức nhận ảnh

Thông qua Ứng dụng
▼

Figure 101 - Update booking for days

← Thông tin chi tiết

Mô tả: Chụp ảnh cổ trang có rất nhiều dịch vụ chụp ảnh khác nhau, do đó bạn có thể chọn gói chụp theo nhu cầu hoặc mức kinh phí có sẵn

Bao gồm các dịch vụ:

- Khách được chụp 3 trang phục/người
- Khách nhận toàn bộ ảnh gốc (khoảng 100-200 ảnh) và 30 ảnh chỉnh sửa

Tổng cộng: 1.490.000 đồng

Phương thức giao hàng

- Giao hàng qua Ứng dụng

Hủy

Đồng ý

Figure 102 - Decline/Accept/Cancel Booking

189

3.3.7. Manage review

- **Purpose**
 - Allow customers to review shooting services
 - Manage reviews
- **Function**
 - View reviews
 - Write reviews (customers only)
- **User Interfaces**

Nhận xét



★★★★★
bởi: Uyển Nhi 19/11/2020

Bình luận: Amazing, good job anh



★★★★★
bởi: Uyển Nhi 22/08/2020

Bình luận: Chưa thấy anh chụp hình nào có tâm như a Tiến luôn á mọi người !!! Like mạnh :3



★★★★★
bởi: Uyển Đinh 22/08/2020

Bình luận: Thích anh Tiến nhất ở việc anh cực thân thiện và chuyên nghiệp!!

Xem thêm

Figure 103 - View Reviews



Figure 104 - Write reviews

3.3.8. Manage request status

- **Purpose**
 - Allow photographers to update request status
- **Function**
 - Update request status
- **User Interfaces**

← Thông tin chi tiết

Mô tả: Chụp ảnh cổ trang có rất nhiều dịch vụ chụp ảnh khác nhau, do đó bạn có thể chọn gói chụp theo nhu cầu hoặc mức kinh phí có sẵn

Bao gồm các dịch vụ:

- ✓ Khách được chụp 3 trang phục/người
- ✓ Khách nhận toàn bộ ảnh gốc (khoảng 100-200 ảnh) và 30 ảnh chỉnh sửa

Tổng cộng: 1.490.000 đồng

Phương thức giao hàng

- ✓ Giao hàng qua ứng dụng

Hủy

Đồng ý

Figure 105 - Change Booking Status

3.3.9. Manage calendar

- **Purpose**
 - Allow photographers to make changes to shooting schedule
- **Function**
 - Manage daily shooting schedule
 - Make changes to daily shooting schedule
 - Add unexpected unavailability
- **User Interfaces**

Lịch làm việc của tôi



Đang chờ

Việc cần làm

< tháng 11 năm 2020 >

Th 2 Th 3 Th 4 Th 5 Th 6 Th 7 CN

16 17 18 19 20 21 22

Chụp ảnh

Sắp diễn ra

Khách hàng: Uyển Nhi

Gói: Chụp mẫu sản phẩm

Địa điểm: QL20, Phường 10,
Thành phố Đà Lạt,
Lâm Đồng, Việt
Nam

Trang chủ



Figure 106 - Shooting schedule

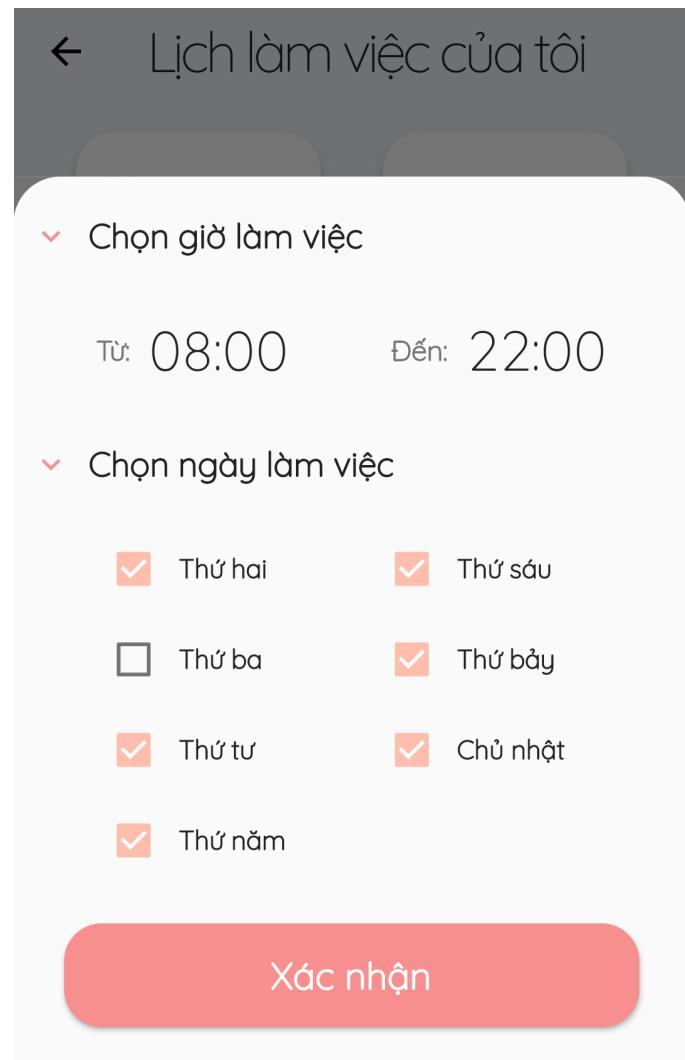


Figure 107 - Make changes to daily shooting schedule

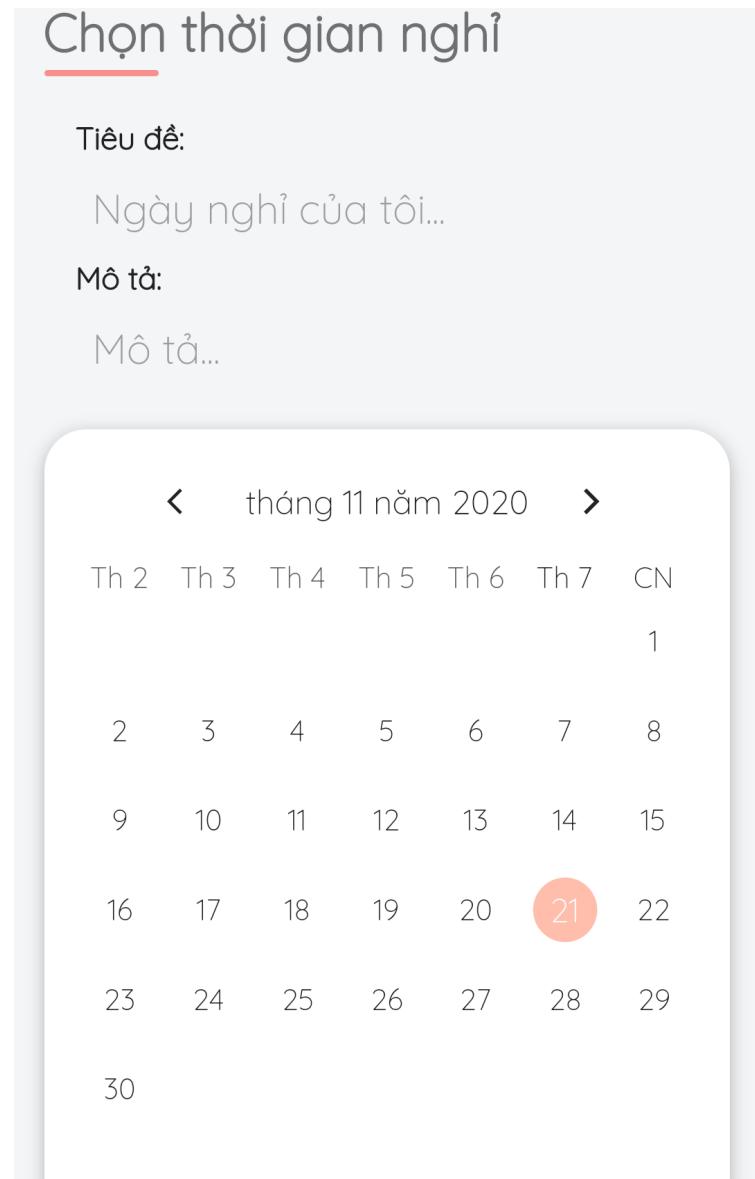


Figure 108 - Add busy days

3.3.10. Manage album

- **Purpose**
 - Allow photographers to manage their albums
 - Allow customers to follow their favourite albums
- **Function**
 - Add album (photographers only)
 - Edit album (photographers only)
 - Delete album (photographers only)
 - Follow album (customers only)
- **User Interfaces**

← Cô - Chú Lê Đồng



1 / 3



209 lượt yêu thích

• Ngày chụp: 20/12/2012

Figure 109 - Follow Album

✗ Tạo Album ✓

Tên Album: *

☰ Ví dụ: Album 01

Thời gian: *

📅 Ví dụ: 01/01/2011

Địa điểm: *

📍 Ví dụ: Hà Nội

Mô tả: *

☰ Ví dụ: Album này

Category: *

✚ ảnh cưới ✖ phong cảnh ✖ kỉ yếu

Figure 110 - Add Album

X Chi tiết Album ✓

Tên Album: *
Cô - Chú Lên ĐỒng

Thời gian: *
Ví dụ: 01/01/2011

Địa điểm: *
Sapa

Mô tả: *
Ví dụ: Album này

Category: *
+ ảnh cưới (X) phong cảnh (X) kỉ yếu

Figure 111 - Edit Album

3.3.11. Manage package

- Purpose
 - Allow photographers to manage their packages
 - Allow customers to view photographers' packages
- Function
 - Add package (photographers only)
 - Update package (photographers only)
 - Remove package (photographers only)
 - View packages
- User Interfaces

Dịch vụ của tôi

Tên gói: GÓI CHỤP 1 NGƯỜI

Bao gồm các dịch vụ:

- Khách được chụp 3 trang phục/người
- Khách nhận toàn bộ ảnh gốc (khoảng 100-200 ảnh) và 30 ảnh chỉnh sửa

Tổng cộng: **1.490.000 đồng**

Chỉnh sửa
Xóa

Tên gói: GÓI CHỤP NHÓM (Giá / người)

Bao gồm các dịch vụ:

- Giá in kỷ yếu photobook rẻ nhất TPHCM.

+

🏠
📷 Dịch vụ
📝
👤

Figure 112 - View Services

← **Tạo dịch vụ** ← **Tạo dịch vụ**

Kiểu dịch vụ: *

Trong ngày Nhiều ngày

Tên dịch vụ: *

Ví dụ: Gói 01

Mô tả:

Nhập mô tả cho gói dịch vụ của bạn

Thời gian tác nghiệp: *

Ví dụ: 3 giờ

Giá thành dịch vụ (đơn vị: đồng): *

Ví dụ: 10.000.000 đ

Phương thức giao hàng: *

Giao hàng qua ứng dụng

Gặp mặt khách hàng

Chi tiết dịch vụ: *

Thêm dịch vụ +

Figure 113 - Add Service

← Chính sửa dịch vụ ← Chính sửa dịch vụ

Tên dịch vụ: *

GÓI CHỤP 1 NGƯỜI

Mô tả:

Chụp ảnh cổ trang có rất nhiều dịch vụ chụp ảnh khác nhau, do đó bạn có thể chọn gói chụp theo nhu cầu hoặc mức kinh phí có sẵn

Giá thành dịch vụ: *

1,490.000 đ

Thời gian tác nghiệp: *

6 giờ

Thời gian tác nghiệp: *

6 giờ

Phương thức giao hàng: *

Giao hàng qua ứng dụng

Gặp mặt khách hàng

Chi tiết dịch vụ: *

Thêm dịch vụ +

Khách được chụp 3 trang phục/người ×

Khách nhận toàn bộ ảnh gốc (khoảng 100-200 ảnh) và 30 ảnh chỉnh sửa ×

Figure 114 - Update Service

3.3.12. Manage categories

- **Purpose**
 - Allow admins to manage shooting categories
 - Allow customers to view and select categories
- **Function**
 - View categories
 - Select categories
- **User Interfaces**

X Tạo Album ✓

Tên Album: *
☰ Ví dụ: Album 01

Thời gian: *
📅 Ví dụ: 01/01/2011

Địa điểm: *
📍 Ví dụ: Hà Nội

Mô tả: *
☰ Ví dụ: Album này ...

Category: *
+ ảnh cưới ✕ phong cảnh ✕ kỉ yếu

Figure 115 - Select / View Categories

3.3.13. Chat

- **Purpose**
 - Allow customers and photographers to easily contact each other
- **Function**
 - Chat
- **User Interfaces**

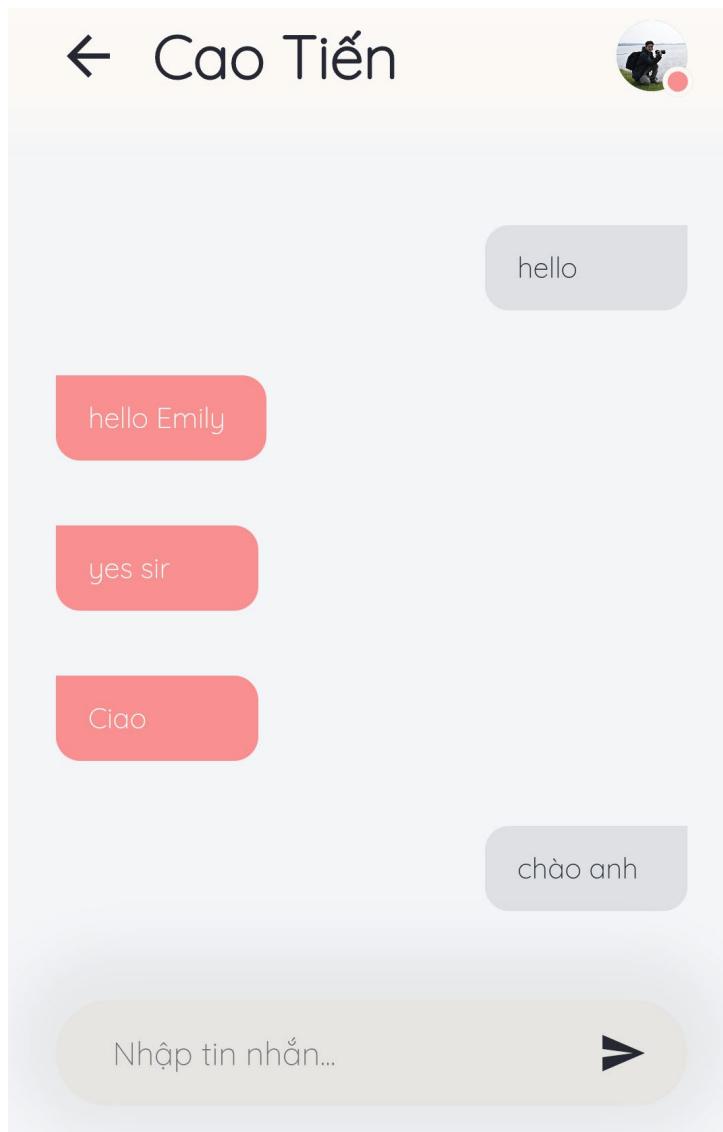


Figure 116 - Chat

3.3.14. Forum

- **Purpose**
 - Allow users to view forum's content
 - Comment in threads
 - Manage threads
 - Manage categories
- **Function**
 - View threads
 - View comments
 - Add threads
 - Edit threads
 - Delete threads
 - Add comments
 - Edit comments
 - Delete comments
- **User Interfaces**

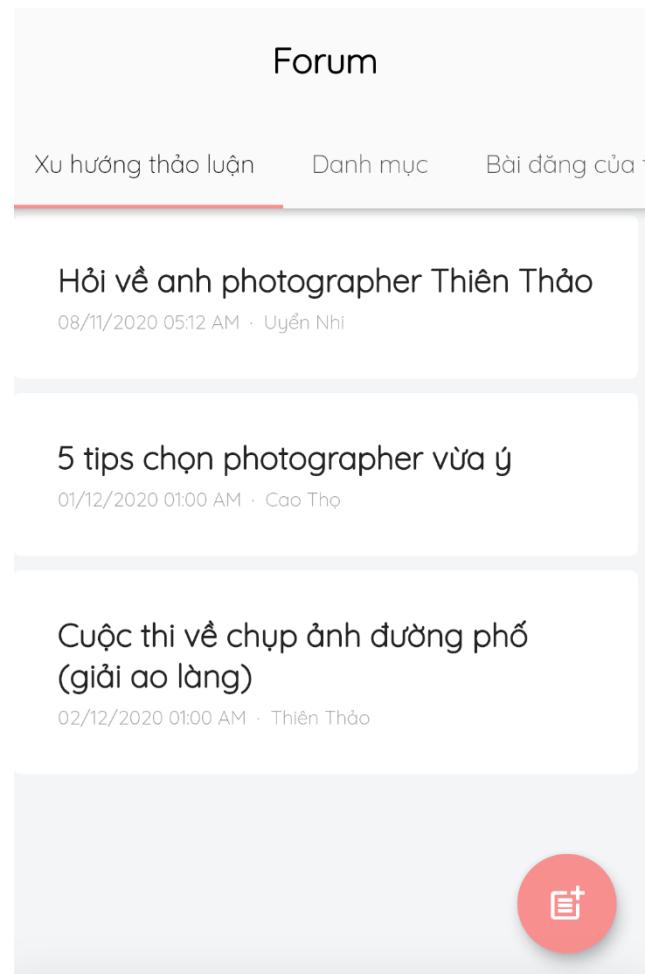


Figure 117 - View Thread

← Cuộc thi về chụp ảnh đường ...



Thiên Thảo

02/12/2020 01:00 AM

Cuộc thi về chụp ảnh đường phố (giải ao làng)

- 👉 Đơn vị tổ chức: Anh em hội PBS
- 👉 Đơn vị đồng tổ chức: Photographer.vn
- 👉 Đơn vị tài trợ: cafechupanh
- 👉 CƠ CẤU GIẢI THƯỞNG
- 👉 Giải 1 : 10 triệu
- 👉 Giải 2 : 6 triệu
- 👉 Giải 3 : 3 triệu
- 👉 Giải 4 : 1 triệu

TRÍCH · ĐIỀU HÀNH



Cao Thọ

02/12/2020 01:30 AM

Ao làng mà chất lượng thế anh

TRÍCH



Figure 118 - View Comment

×

Trả lời

✓

Viết bình luận của bạn...

Hình ảnh

+

Figure 119 - Add Comment



Figure 120 - Create Thread

3.3.15. Manage delivery method

- **Purpose**
 - Allow admins to manage delivery method
 - Allow customers and photographer to view and select delivery method
- **Function**
 - Select photo delivery methods (for photographers and customers)
- **User Interfaces**

← Chính sửa dịch vụ

Thời gian tác nghiệp: *

6 giờ

Phương thức giao hàng: *

Giao hàng qua Ứng dụng

Gặp mặt khách hàng

Chi tiết dịch vụ: *

Thêm dịch vụ +

Khách được chụp 3 trang phục/người ×

Khách nhận toàn bộ ảnh gốc (khoảng 100-200 ảnh) và 30 ảnh chỉnh sửa ×

Figure 121 - Select Photo Delivery Methods

3.3.16. Manage profile

- Purpose
 - Allow users to edit their profile
- Function
 - Edit profiles
 - Edit avatar and cover photos
- User Interfaces

✗ Chỉnh sửa trang cá nhân ✓

Họ tên: *



Ví dụ: Nguyễn Văn A

Email: *



Ví dụ: abc@xzy.com

Số điện thoại: *



Ví dụ: 012345678

Địa điểm làm việc: *



Ho chi minh



Ha noi



Da lat



Mô tả: *



Ví dụ: Tôi là....

Figure 122 - Update Profile

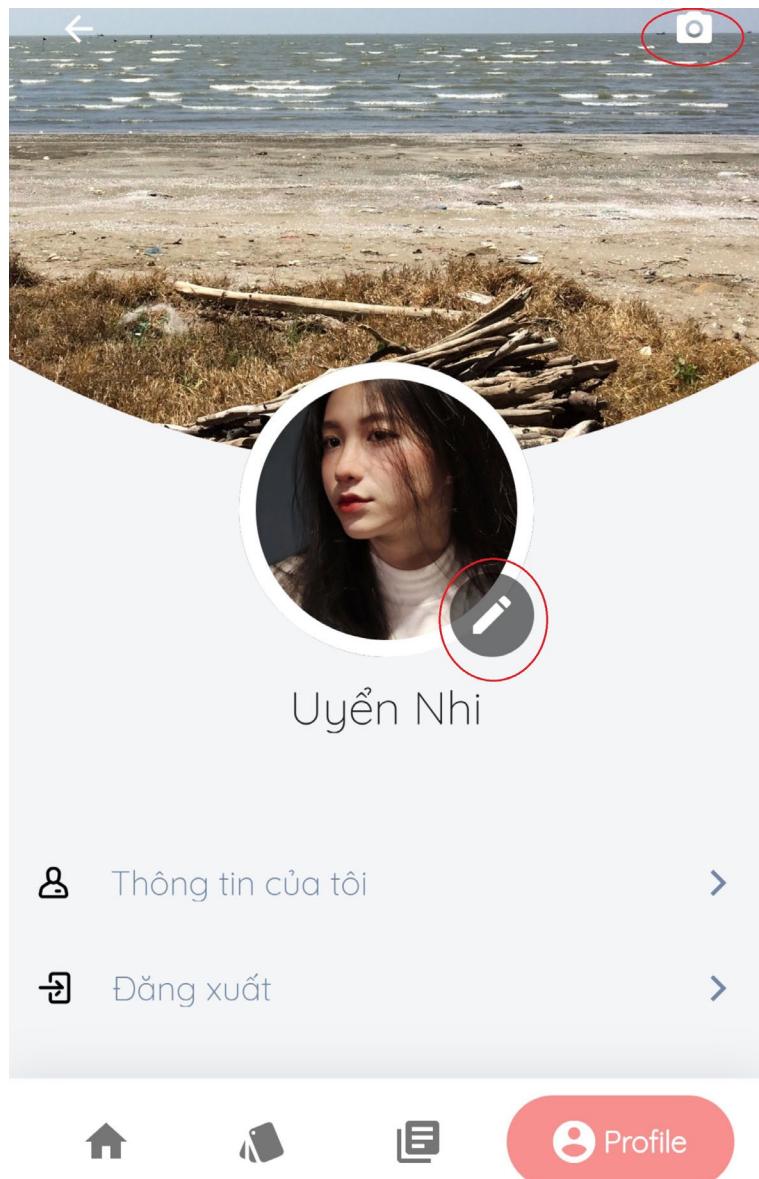


Figure 123 - Update Avatar/ Cover

3.4 Troubleshooting

1. Current location sometimes cannot be determined

First occurrence

- During testing time

Problem

- Current locations on phone are not allowed

Workaround

Do the following steps:

- Go to Setting > Current locations > On

2. Profile photos are sometimes unsuccessfully updated

First occurrence

- During testing time

Problem

- Image size exceeds maximum allowable size

Workaround

- Choose another photo smaller than ... size

4. 配達可能なパッケージ

4.1 ソースコードと文書

番号	項目	サブ項目	タイプ	バージョン
Code Package				
1	Web API	pbs_webapi	新しい	1.0.1
2	モバイルアプリ	photographer_mobile_app	新しい	1.0.1
		customer_mobile_app	新しい	1.0.1
Database				
1	テーブル	pbs_users.sql	新しい	1.0.1
		pbs_roles.sql	新しい	1.0.1
		pbs_photographer_packages.sql	新しい	1.0.1
		pbs_packages_services.sql	新しい	1.0.1
		pbs_services.sql	新しい	1.0.1
		pbs_returning_types.sql	新しい	1.0.1
		pbs_bookings.sql	新しい	1.0.1
		pbs_booking_details.sql	新しい	1.0.1
		pbs_booking_time_location_detail.sql	新しい	1.0.1
		pbs_booking_comments.sql	新しい	1.0.1
		pbs_albums.sql	新しい	1.0.1
		pbs_customers_albums.sql	新しい	1.0.1
		pbs_album_images.sql	新しい	1.0.1
		pbs_images.sql	新しい	1.0.1
		pbs_categories.sql	新しい	1.0.1
		pbs_threads.sql	新しい	1.0.1
		pbs_thread_comments.sql	新しい	1.0.1
		pbs_thread_topics.sql	新しい	1.0.1
		pbs_locations.sql	新しい	1.0.1
		pbs_working_day_of_week.sql	新しい	1.0.1
		pbs_busy_days.sql	新しい	1.0.1
Documents				
1	概要	Report1_Project-Introduction.docx	新しい	1.0.1
2	経営計画	Report2_Project-Management-Plan.docx	新しい	1.0.1

3	要件	Report3_Software Requirement Specification.docx	新しい	1.0.1
4	設計	Report4_Software Design Document.docx	新しい	1.0.1
5	テスト	Report5_Test Documentation.docx	新しい	1.0.1
		Report5_Test Case Document.xlsx	新しい	1.0.1
		Report5_Unit Test Case.xls	新しい	1.0.1
6	ユーザーガイド	Report6_Software User Guides.docx	新しい	1.0.1
7	最終	Report7_Final Project Report.docx	新しい	1.0.1

表123 配達可能なパッケージ

4.2 既知の問題、制限および限定

- お支払いには対応しておりません。
- システムには写真検証機能はありません。
- このシステムは、顧客と写真家の両方にウェブサイト版を提供していません。ユーザーにとって不快になるアプリケーションの利

5. 設置ガイド

5.1 システム要件

5.1.1 ハードウェア要件

パソコン用

パソコン	最小要件	お勧め
Internet Connection	Cable, Wi-Fi (4 Mbps)	Cable, Wi-Fi (8 Mbps)
Operating System	Windows 7	Windows 7 or more.
Computer Processor	Intel® Core i3 1.4GHz	Intel® Core i5 2.50GHz
Computer Memory	1GB RAM	2GB RAM or more
Web Browser	Chromes (v42 or higher)	Chrome latest stable version

表124 PCハードウェア要件

Androidデバイス用

オペレーティン	Android 8.0 or higher
プロセッサ	Intel Atom® Processor Z2520 1.2 GHz, or faster processor
ストレージ	Between 850 MB and 1.2 GB, depending on the language version
RAM	Minimum of 512 MB, 2 GB is recommended
ハードディスク	<ul style="list-style-type: none"> 2 GB of available hard-disk space for installation; extra free space is required during installation. You cannot install using a removable flash storage device.

ビデオ	1920 x 1080 pixels or higher
ブラウザ/インターネット	<ul style="list-style-type: none"> This application is designed to work online. To download and launch Google Play Store apps within the application, a high-speed Internet connection is recommended.

表 125 ソフトウェア要件

5.1.2 ソフトウェア要件

ソフトウェア	名前/バージョン	説明
Operation System	Windows 7, Windows 10	Operating system and platform for development
Web browser	Chrome 69 or above, Mozilla Firefox	For web app
Android System	Android 4.0 or higher	For mobile app

表 126 ソフトウェア要件

5.2 設定ファイル

- 写真家のアプリのため

 photographer.apk	12/1/2020 9:44 AM	BlueStacks.Apk	48,802 KB
--	-------------------	----------------	-----------

図 124 設定ファイル - 1

- お客様のアプリのため

 customer.apk	12/1/2020 9:44 AM	BlueStacks.Apk	48,802 KB
--	-------------------	----------------	-----------

図 125 設定ファイル - 2

5.3 インストール手順

2つのandroidアプリのインストール手順は同じですので、私たちはそれらの両方のための一つのガイドライン。

1. "Install"ボタンを押してPBSをインストールします。

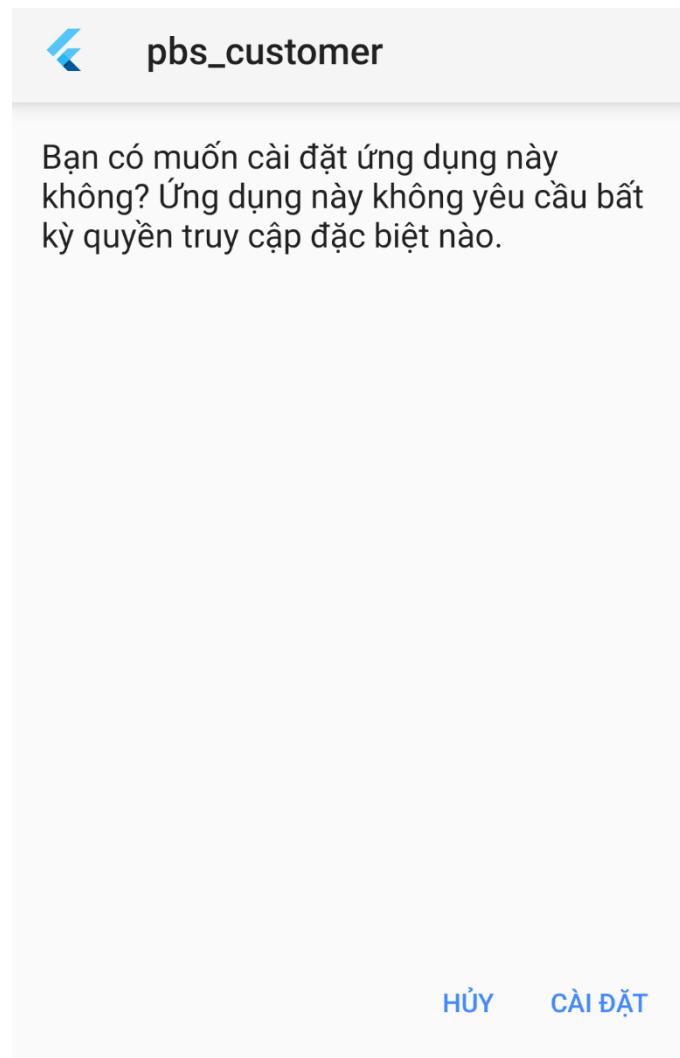


図126 インストール - 1

2. プレス"Accept"受け入れ申請のアクセス権に優先する。

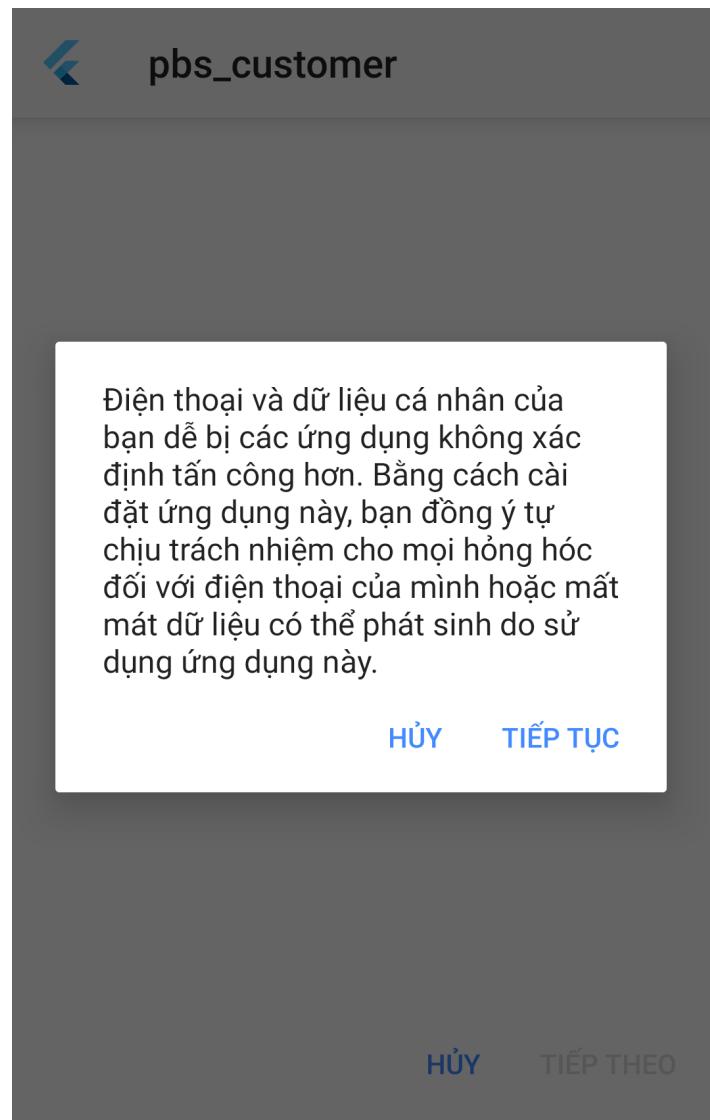


図 127 インストール - 2

3. PBSアプリのインストールを待ちます。

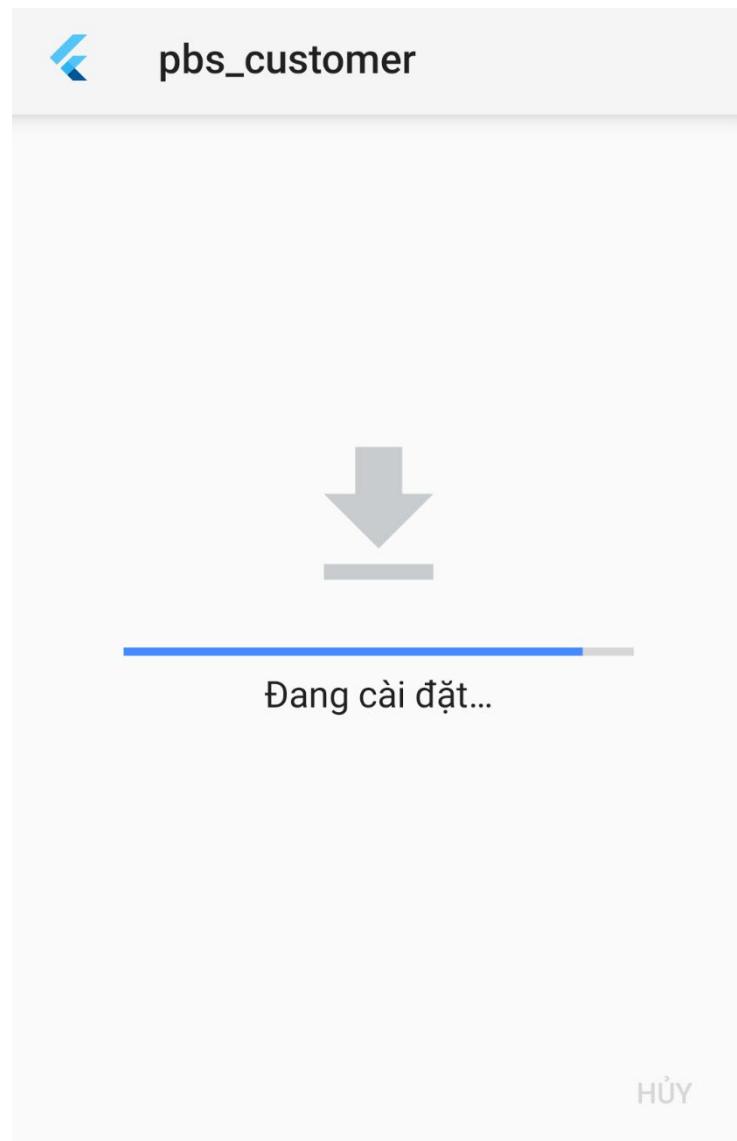


図 128 インストール - 3

4. インアプリを起動するには、"Mở"を押します。

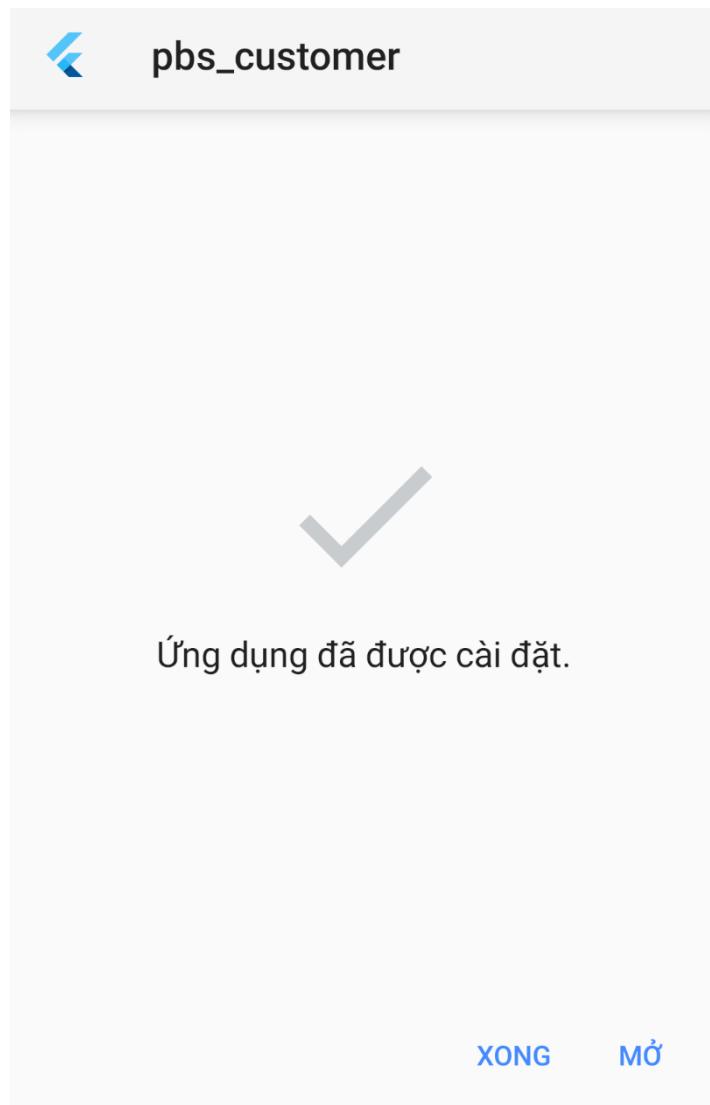


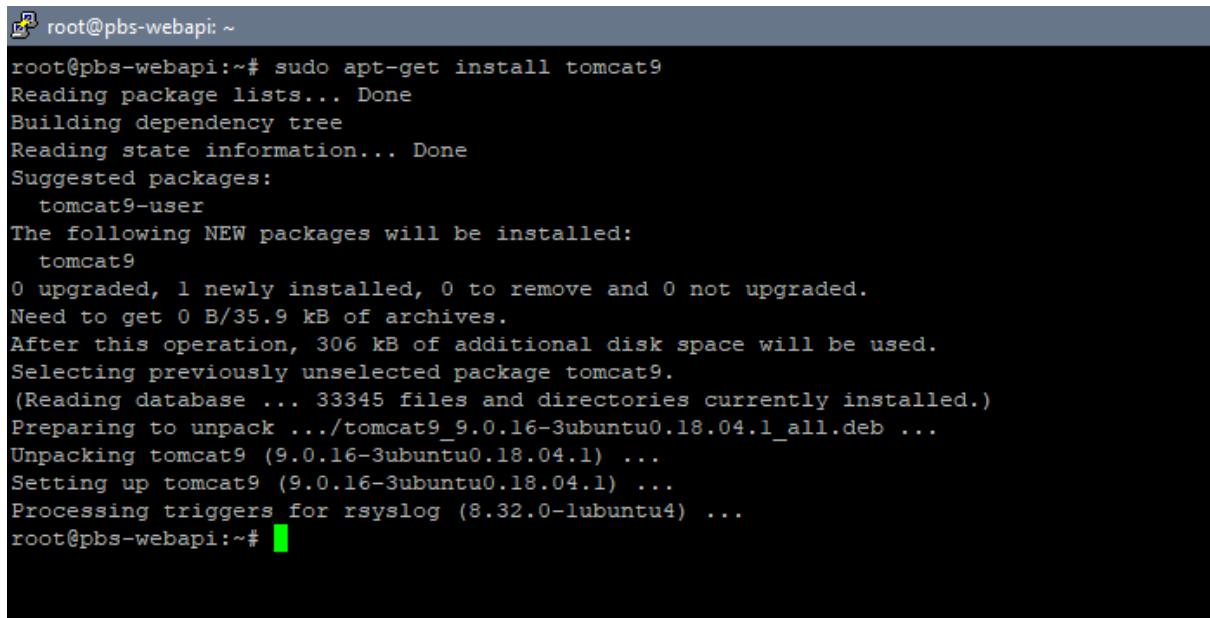
図129 インストール - 4

5. 詳細はマニュアルをご覧ください。

5.4 サーバー側の展開

WebapiをHostingerのVPSサーバーに展開します。サーバのオペレーティングシステムはUbuntu 18.04 LTSです。

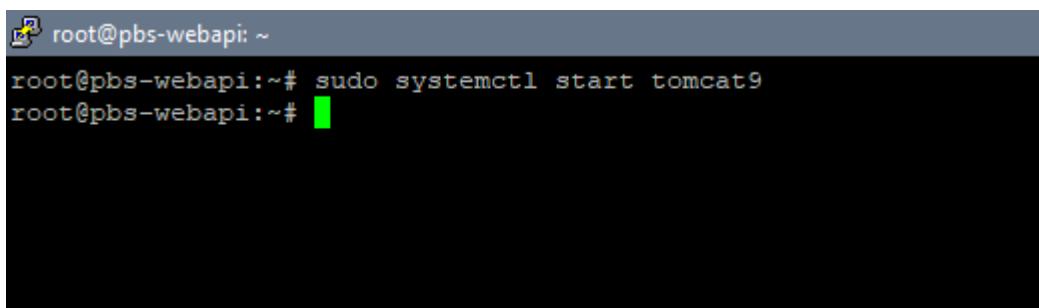
ステップ1: TomcatとJDKをインストールする。Openjdkはコマンドの依存関係としてインストールされます: `sudo apt-get install tomcat9`



```
root@pbs-webapi:~# sudo apt-get install tomcat9
Reading package lists... Done
Building dependency tree
Reading state information... Done
Suggested packages:
  tomcat9-user
The following NEW packages will be installed:
  tomcat9
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 0 B/35.9 kB of archives.
After this operation, 306 kB of additional disk space will be used.
Selecting previously unselected package tomcat9.
(Reading database ... 33345 files and directories currently installed.)
Preparing to unpack .../tomcat9_9.0.16-3ubuntu0.18.04.1_all.deb ...
Unpacking tomcat9 (9.0.16-3ubuntu0.18.04.1) ...
Setting up tomcat9 (9.0.16-3ubuntu0.18.04.1) ...
Processing triggers for rsyslog (8.32.0-lubuntu4) ...
root@pbs-webapi:~#
```

図 130 サーバー側の展開 - 1

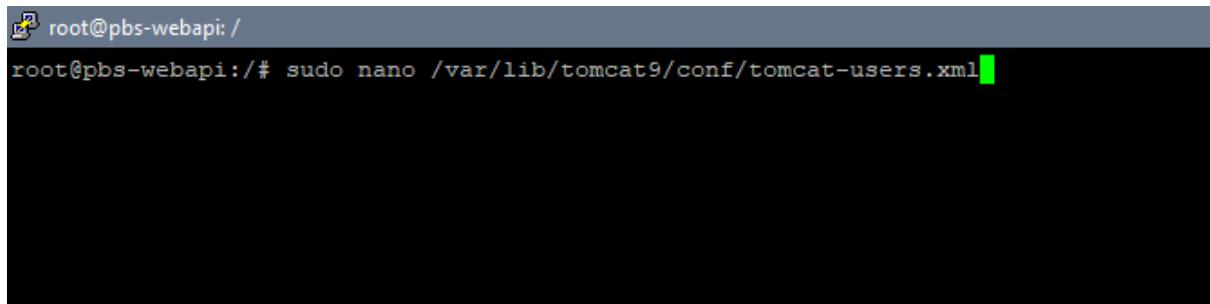
ステップ2:コマンドでtomcatサービスを開始します: sudo systemctl start tomcat9。



```
root@pbs-webapi:~# sudo systemctl start tomcat9
root@pbs-webapi:~#
```

図 131 サーバー側の展開 - 2

ステップ3:/var/lib/tomcat9/conf/tomcat-users.xmlでtomcat管理者アカウントを設定します:



```
root@pbs-webapi:/# sudo nano /var/lib/tomcat9/conf/tomcat-users.xml
```

図 132 サーバー側の展開 - 3

```
root@pbs-webapi: /  
GNU nano 2.9.3 /var/lib/tomcat9/conf/tomc  
  
<!--  
NOTE: By default, no user is included in the "manager-gui" role required  
to operate the "/manager/html" web application. If you wish to use this app,  
you must define such a user - the username and password are arbitrary. It is  
strongly recommended that you do NOT use one of the users in the commented out  
section below since they are intended for use with the examples web  
application.  
-->  
<!--  
NOTE: The sample user and role entries below are intended for use with the  
examples web application. They are wrapped in a comment and thus are ignored  
when reading this file. If you wish to configure these users for use with the  
examples web application, do not forget to remove the <!... ...> that surrounds  
them. You will also need to set the passwords to something appropriate.  
-->  
<!--  
<role rolename="tomcat"/>  
<role rolename="role1"/>  
<user username="tomcat" password=<must-be-changed> roles="tomcat"/>  
<user username="both" password=<must-be-changed> roles="tomcat,role1"/>  
<user username="role1" password=<must-be-changed> roles="role1"/>  
-->  
<role rolename="manager-gui"/>  
<role rolename="admin-gui"/>  
<user username="username" password="password" roles="manager-gui,admin-gui"/>  
</tomcat-users>
```

図133 サーバー側の展開 - 4

ステップ4:"mvn package clean"を使用してwebapi warファイルを書き出す:

```
C:\Windows\System32\cmd.exe - mvn clean package
Microsoft Windows [Version 10.0.18363.1256]
(c) 2019 Microsoft Corporation. All rights reserved.

D:\Capstone\pbs\webapi>mvn clean package
[INFO] Scanning for projects...
[WARNING]
[WARNING] Some problems were encountered while building the effective model for fpt.university:pbs-webapi:war:0
[WARNING] The expression ${artifactId} is deprecated. Please use ${project.artifactId} instead.
[WARNING]
[WARNING] It is highly recommended to fix these problems because they threaten the stability of your build.
[WARNING]
[WARNING] For this reason, future Maven versions might no longer support building such malformed projects.
[WARNING]
[INFO]
[INFO] -----< fpt.university:pbs-webapi >-----
[INFO] Building pbs-webapi 0.0.1-SNAPSHOT
[INFO] -----[ war ]-----
[INFO]
[INFO] --- maven-clean-plugin:3.1.0:clean (default-clean) @ pbs-webapi ---
[INFO] Deleting D:\Capstone\pbs\webapi\target
[INFO]
[INFO] --- maven-resources-plugin:3.1.0:resources (default-resources) @ pbs-webapi ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] Copying 1 resource
[INFO] Copying 1088 resources
[INFO]
[INFO] --- maven-compiler-plugin:3.8.1:compile (default-compile) @ pbs-webapi ---
[INFO] Changes detected - recompiling the module!
[INFO] Compiling 118 source files to D:\Capstone\pbs\webapi\target\classes
[INFO]
[INFO] --- maven-resources-plugin:3.1.0:testResources (default-testResources) @ pbs-webapi ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] skip non existing resourceDirectory D:\Capstone\pbs\webapi\src\test\resources
[INFO]
[INFO] --- maven-compiler-plugin:3.8.1:testCompile (default-testCompile) @ pbs-webapi ---
[INFO] Changes detected - recompiling the module!
[INFO] Compiling 1 source file to D:\Capstone\pbs\webapi\target\test-classes
[INFO]
[INFO] --- maven-surefire-plugin:2.22.2:test (default-test) @ pbs-webapi ---
[INFO]
[INFO] -----
```

図134 サーバー側の展開 - 5

```
[INFO] --- maven-war-plugin:3.2.3:war (default-war) @ pbs-webapi ---
[INFO] Packaging webapp
[INFO] Assembling webapp [pbs-webapi] in [D:\Capstone\pbs\webapi\target\pbs-webapi]
[INFO] Processing war project
[INFO] Webapp assembled in [7010 msecs]
[INFO] Building war: D:\Capstone\pbs\webapi\target\pbs-webapi.war
[INFO]
[INFO] --- spring-boot-maven-plugin:2.3.3.RELEASE:repackage (repackage) @ pbs-webapi ---
[INFO] Replacing main artifact with repackaged archive
[INFO]
[INFO] BUILD SUCCESS
[INFO]
[INFO] -----
```

D:\Capstone\pbs\webapi>

図135 サーバー側の展開 - 6

ステップ5:pscpを使用してssh経由でwarファイルをサーバーに転送します。

```

C:\Windows\System32\cmd.exe
D:\Capstone\pbs\webapi>pscp "D:\Capstone\pbs\webapi\target\pbs-webapi.war" root@194.59.165.195:/var/lib/tomcat9/webapps
root@194.59.165.195's password:
Access denied
root@194.59.165.195's password:
pbs-webapi.war | 271345 kB | 2422.7 kB/s | ETA: 00:00:00 | 100%
D:\Capstone\pbs\webapi>_

```

図 136 サーバー側の展開 - 7

6.ユーザーマニュアル

6.1 用語と定義

アカウントはユーザーとしてサインインするために登録するアカウントです。

フォトグラファーのアプリはフォトグラファーが客に連絡するための「PBS」によって提供されるアプリです。

客のアプリは客がフォトグラファーに連絡するための「PBS」によって提供されるアプリです。

モバイルデバイスはフォトグラファーのアプリや客のアプリをダウンロードして実行するために使えるケータイ、タブレット、他のデバイスです。

システムとはフォトグラファーのアプリ、客のアプリ、ウェブサイト、プラットフォームなど、フォトグラファーや客の連絡をサポートするために「PBS」によって提供するシステムです。

フォトグラファーは客がシステムで予約できる撮るサービスを提供する独立した第三者です。

客は客のアプリのアカウントを持っている人です。

ウェブサイトは「PBS」が運営するウェブサイトです。

6.2 システム要件

6.2.1 ハードウェア要件

パソコン

パソコン	最小要件	おすすめ
インターネット接続	ケーブル、ワイファイ (4Mbps)	ケーブル、ワイファイ (8Mbps)
オペレーティングシステム	WINDOWS7	WINDOWS7以上
コンピュータプロセッサ	インテル®コア i3 1.4GHz	インテル®コア i3 2.5GHz
コンピュータメモリ	1GB RAM	2GB RAM以上
ウェブブラウザ	グーグルクローム(v42以上)	グーグルクロームの最新安定版

アンドロイドデバイス

オペレーティング システム	Android 8.0以上
プロセッサ	インテル Atom® プロセッサ Z2520 1.2 GHz 以上のプロセッサ
ストレージ	言語バージョンに応じて 850 MB と 1.2 GB の間です。
RAM	最小は 512 MB です。 2 GB はおすすめです。
ハードディスク	インストール用は 2 GB のハードディスクのスペースが必要です。 フラッシュストレージデバイスを使用してインストールすることはできません。.
ビデオ	1920 x 1080 ピクセル以上。
ブラウザ/インターネット	このアプリは、オンラインで使うように設計されています。 「GOOGLEPLAY」ストアアプリをダウンロードして起動するには、高速インターネット接続をおすすめします。

6.2.2 ソフトウェア要件

ソフトウェア	名前/バージョン	説明
オペレーションシステム	Windows 7, Windows 10	開発用のオペレーティングシステムとプラットフォーム
ウェブブラウザ	クロム 69以上, ファイアーフォックス	ウェブアプリ用
Android システム	Android 4.0以上	モバイルアプリ用

6.3 アプリの使用

6.3.1. 概要

客のアプリでは客がフォトグラファーを検索し、予約することができます。その後、客は自分の体験に基づいてフォトグラファーを評価することができます。評価は星として表示され、フォトグラファーのプロフィールに表示されるレビューすることができます。ユーザーに写真や流行のスタイルについての情報を提供するために、システムにフォーラムもあります。

フォトグラファーのアプリはフォトグラファーが潜在的な客の広い範囲に達するのに役立ちます。また、撮るスケジュールを管理したり、客の予約を受け取ることができます。そして、プロフィールを編集したり、アルバムや提供するサービスを追加したりすることができます。フォーラムに参加することもできます。

システムには「ADMINISTRATOR」に簡単にシステムを制御できるように、ウェブサイトを提供します。「ADMINISTRATOR」には撮るカテゴリを追加、デリート、編集したり、写真配信方法を管理したりできます。コミュニティ標準に違反しているユーザーをロックできます。

6.3.2. 登録

- **目標**
 - ユーザーのアカウントを作成します
- **機能**
 - 「PBS」アカウントを作成します
 - グーグルでサインアップします
 - フェイスブックでサインアップします
- **ユーザインターフェイス**

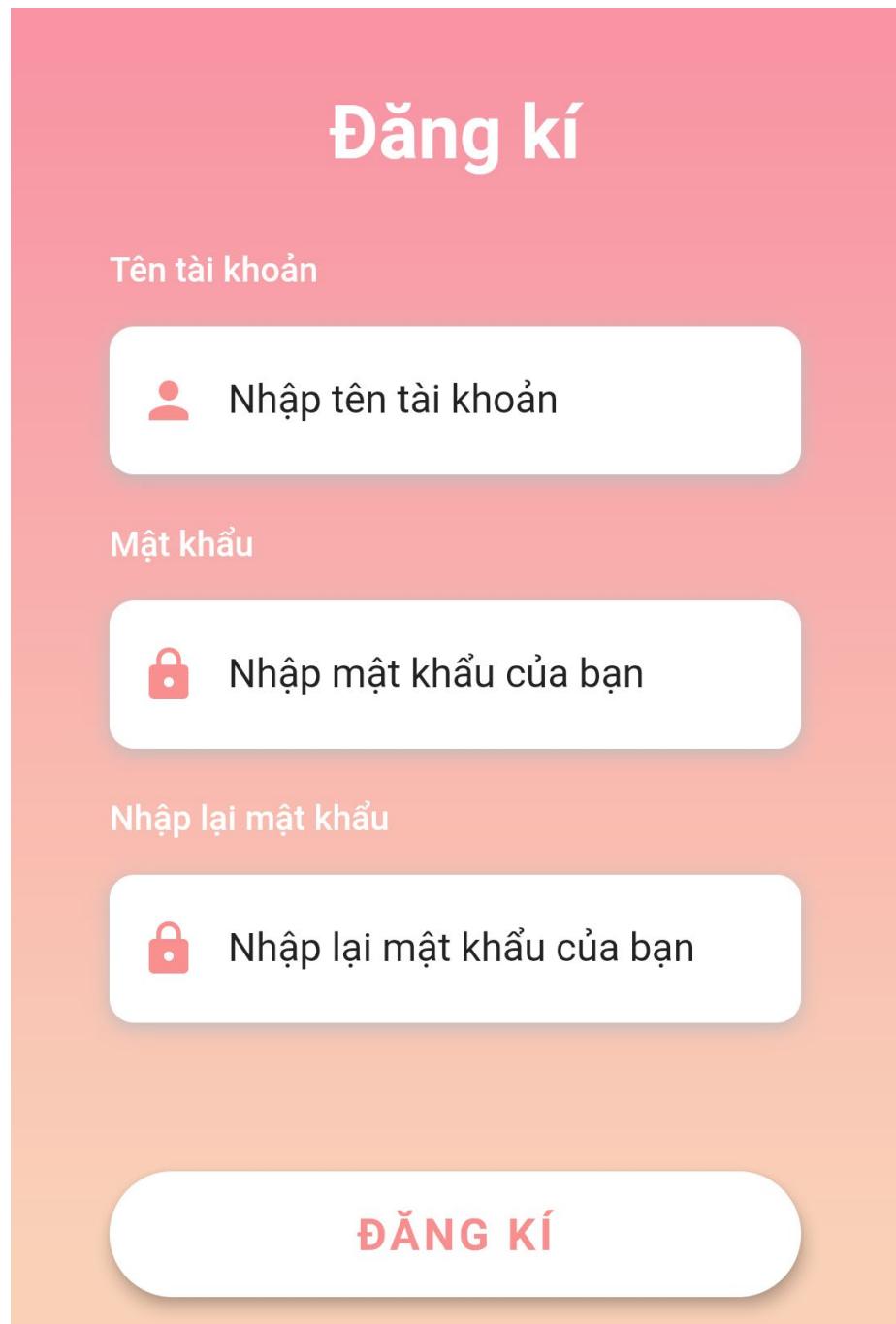


図137 「PBS」アカウントを作成する

6.3.3. ログイン

- **目標**
 - ユーザーがサインインします
- **機能**
 - 「PBS」アカウントでサインインします
 - グーグルでサインインします
 - フェイスブック アカウントでサインインします
- **ユーザーインターフェース**

6.3.4. ログアウト

- **目標**
 - ユーザーがサインアウトします
- **機能**
 - サインアウトします
- **ユーザーインターフェース**

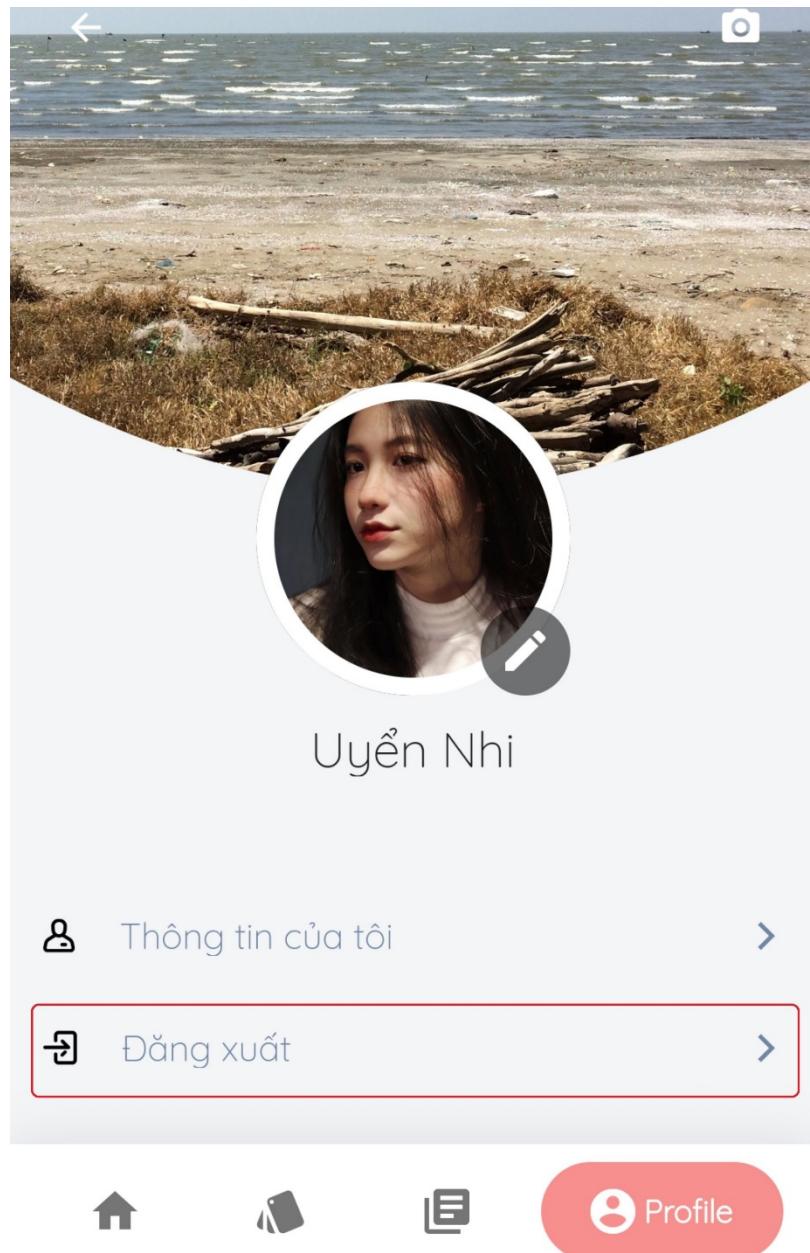


図138 – ログアウト

6.3.5. フォトグラファーを検索

- **目標**
 - 客が適用なフォトグラファーやサービスを見つけます
- **機能**
 - フォトグラファーを検索します
 - サービスを検索します

- ユーザーインターフェース

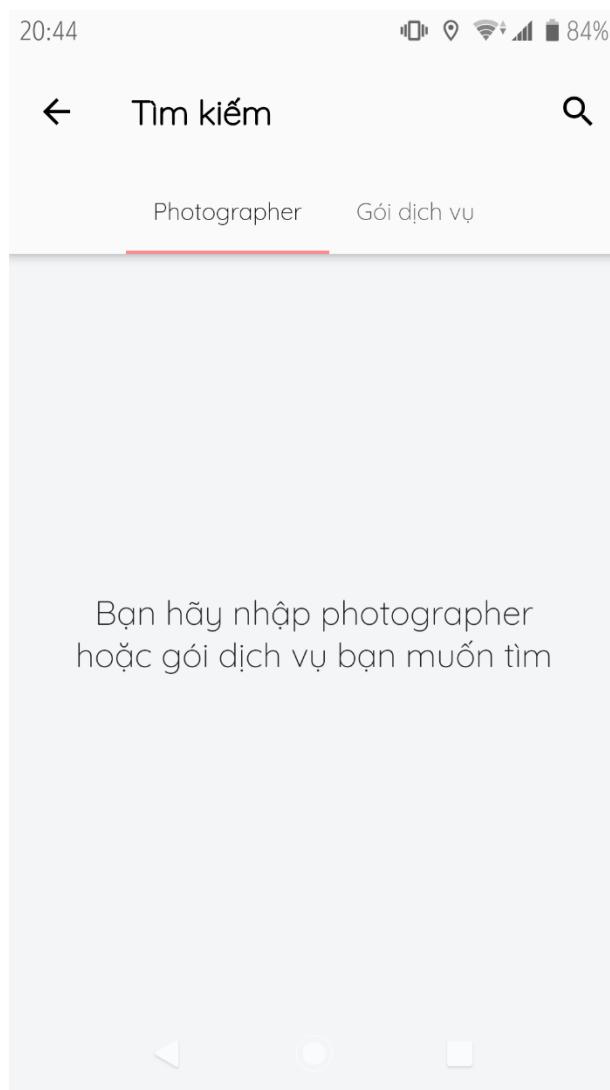


図139- フォトグラファー／サービスの検索

6.3.6. リクエストの管理

- 目標
 - 客が予約を管理できます
- 機能
 - 数日を予約します(客のみ)
 - 一日を予約します(客のみ)
 - 予約を変更します(客のみ)
 - 予約をキャンセルします(客のみ)
 - 予約の受け入れます(フォトグラファーのみ)
 - 予約を断ります(フォトグラファーのみ)
- ユーザーインターフェース

Đặt lịch với Cao Tiến

Thông tin chi tiết

⌚ Thời gian chụp: Hãy chọn thời gian ... >

⌚ Thời gian nhận: Hãy chọn thời gian ... >

📍 Địa điểm: Hãy chọn nơi bạn muốn... >

📍 Giao trả: Thông qua Ứng dụng <v>

🏷️ Gói dịch vụ:
GÓI CHỤP 1 NGƯỜI <v>

Tổng cộng: 1.490.000 đồng

Đặt dịch vụ

図140 - 一日の予約

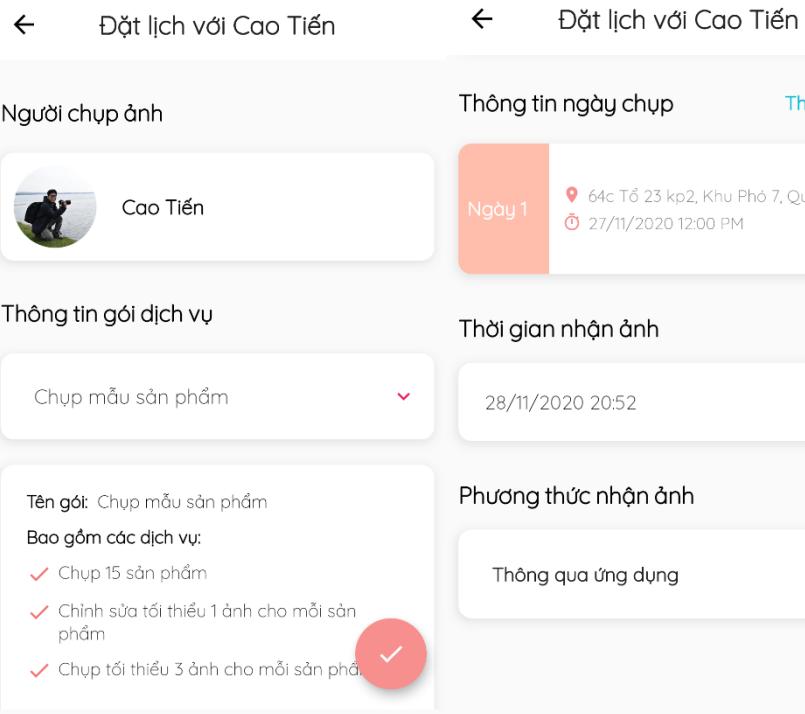


図141 - 数日の予約

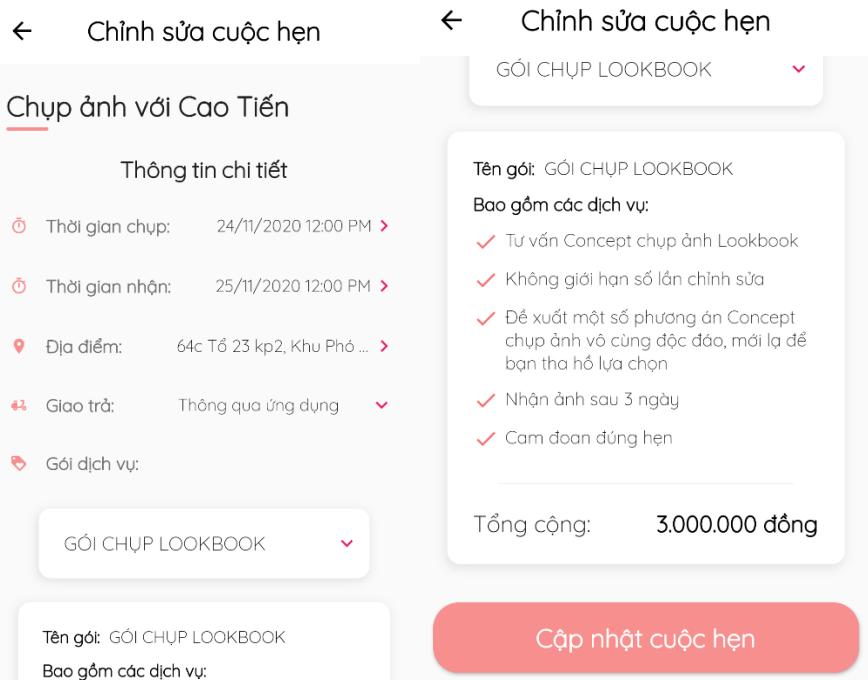


図142 - 一日の予約の変更

← Đặt lịch với Cao Tiến

Người chụp ảnh

Cao Tiến
Chuyên viên nhiếp ảnh

← Đặt lịch với Cao Tiến

Thông tin ngày chụp

[Thêm ngày](#)

Ngày 1

88C Tô 21 KP2, Khu Phố 7, Quận 12, Hồ Chí Minh
24/11/2020 12:00 PM

Thông tin gói dịch vụ

Chụp mẫu sản phẩm

Thời gian nhận ảnh

25/11/2020 12:00 PM

Tên gói: Chụp mẫu sản phẩm

Bao gồm các dịch vụ:

- Chụp 15 sản phẩm
- Chỉnh sửa tối thiểu 1 ảnh cho mỗi sản phẩm
- Chụp tối thiểu 3 ảnh cho mỗi sản phẩm

Phương thức nhận ảnh

Thông qua ứng dụng

図143 - 数日の予約の変更

← Thông tin chi tiết

Mô tả: Chụp ảnh cổ trang có rất nhiều dịch vụ chụp ảnh khác nhau, do đó bạn có thể chọn gói chụp theo nhu cầu hoặc mức kinh phí có sẵn

Bao gồm các dịch vụ:

- Khách được chụp 3 trang phục/người
- Khách nhận toàn bộ ảnh gốc (khoảng 100-200 ảnh) và 30 ảnh chỉnh sửa

Tổng cộng: **1.490.000 đồng**

Phương thức giao hàng

- Giao hàng qua ứng dụng

Hủy

Đồng ý

図144 - 予約の断り/受入/キャンセル

6.3.7. レビューの管理

- **目標**
 - 客がサービスをレビューします
 - レビューを管理します
- **機能**
 - レビューを観ます
 - レビューを書きます(客のみ)
- **ユーザーインターフェース**

Nhận xét

★★★★★19/11/2020
bởi: Uyển Nhi

Bình luận: Amazing, good job anh

★★★★★22/08/2020
bởi: Uyển Nhi

Bình luận: Chưa thấy anh chụp hình nào có tâm như a Tiến luôn á mọi người !!! Like mạnh :3

★★★★★22/08/2020
bởi: Uyển Đinh

Bình luận: Thích anh Tiến nhất ở việc anh cực thân thiện và chuyên nghiệp!!

[Xem thêm](#)

図145 – レビューを観ます



図146 - レビューを書きます

6.3.8. リクエストの状態の管理

- **目標**
 - フォトグラファーがリクエストの状態を変更します
- **機能**
 - リクエストの状態を変更します
- **ユーザーインターフェース**

← Thông tin chi tiết

Mô tả: Chụp ảnh cổ trang có rất nhiều dịch vụ chụp ảnh khác nhau, do đó bạn có thể chọn gói chụp theo nhu cầu hoặc mức kinh phí có sẵn

Bao gồm các dịch vụ:

- ✓ Khách được chụp 3 trang phục/người
- ✓ Khách nhận toàn bộ ảnh gốc (khoảng 100-200 ảnh) và 30 ảnh chỉnh sửa

Tổng cộng: 1.490.000 đồng

Phương thức giao hàng

- ✓ Giao hàng qua ứng dụng

Hủy

Đồng ý

図147 - 予約の状態を変更します

6.3.9. カレンダーの管理

- **目標**
 - フォトグラファーが撮るスケジュールを変更します
- **機能**
 - 毎日の撮るスケジュールを管理します
 - 毎日の撮るスケジュールを変更します
 - 都合がない日を追加します
- **ユーザーインターフェース**

Lịch làm việc của tôi



Đang chờ

Việc cần làm

< tháng 11 năm 2020 >

Th 2 Th 3 Th 4 Th 5 Th 6 Th 7 CN
16 17 18 19 20 21 22

Chụp ảnh

Sắp diễn ra

Khách hàng: Uyển Nhi

Gói: Chụp mẫu sản phẩm

Địa điểm: QL20, Phường 10,
Thành phố Đà Lạt,
Lâm Đồng, Việt
Nam

Trang chủ



図148 - 撮るスケジュール

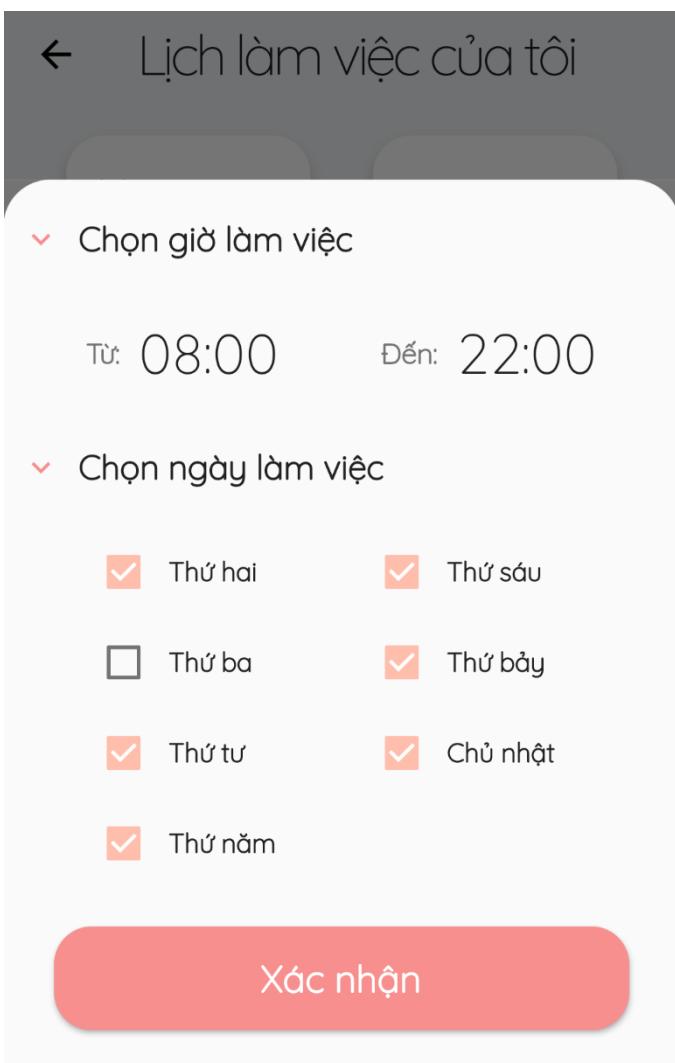


図149 – 毎日の撮るスケジュールを変更します

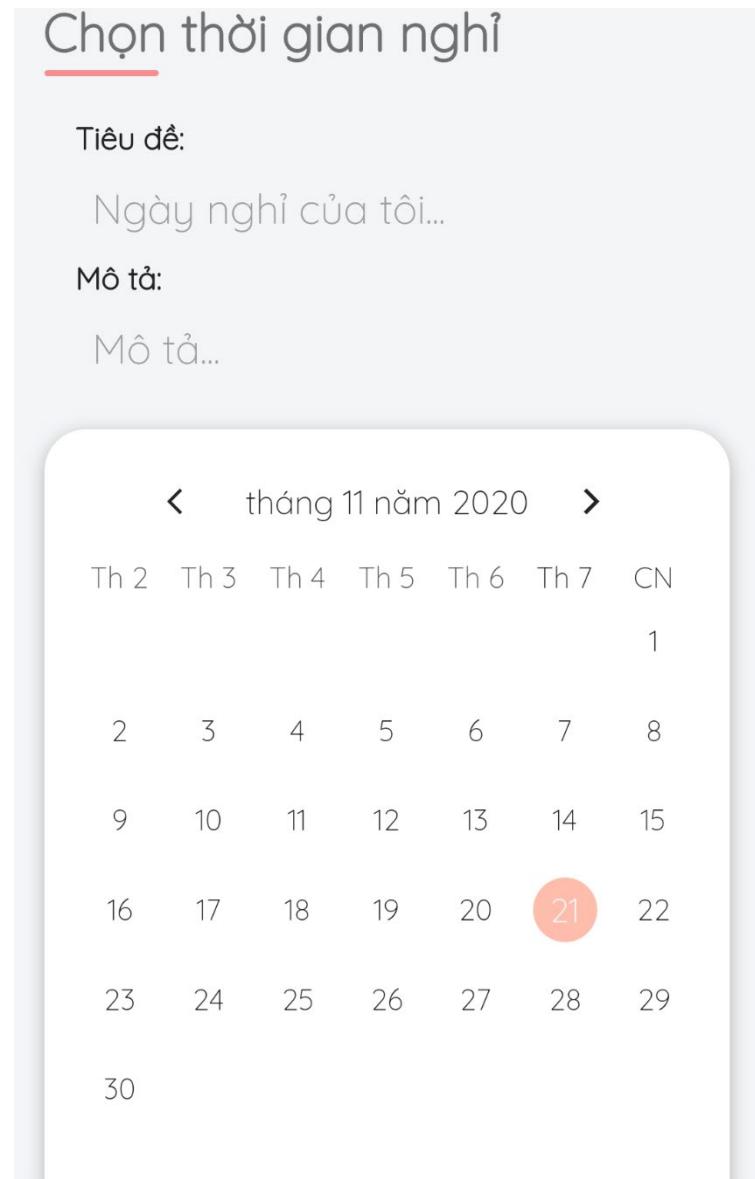


図150 – 都合がない日を追加します

6.3.10. アルバムの管理

- **目標**
 - フォトグラファーがアルバムを管理します
 - 客が好きなアルバムをフォローします
- **機能**
 - アルバムを追加します(フォトグラファーのみ)
 - アルバムを変更します(フォトグラファーのみ)
 - アルバムをデリートします(フォトグラファーのみ)
 - アルバムをフォローします(客のみ)
- **ユーザーインターフェース**

← Cô - Chú Lê Đồng



1 / 3



209 lượt yêu thích

• Ngày chụp: 20/12/2012

図151 – アルバムをフォローする

×

Tạo Album

✓

Tên Album: *



Ví dụ: Album 01

Thời gian: *



Ví dụ: 01/01/2011

Địa điểm: *



Ví dụ: Hà Nội

Mô tả: *



Ví dụ: Album này

Category: *



anh cưới



phong cảnh



kỉ yếu

図152 - アルバムを追加します

Chi tiết Album

Tên Album: *

Cô - Chú Lên ĐỒng

Thời gian: *

Ví dụ: 01/01/2011

Địa điểm: *

Sapa

Mô tả: *

Ví dụ: Album này

Category: *



ảnh cưới



phong cảnh



kỉ yếu

図153 – アルバムを変更します

6.3.11. サービスの管理

• 目標

- フォトグラファーがサービスを管理します
- 客がフォトグラファーのサービスを観ます

• 機能

- サービスを追加します(フォトグラファーのみ)
- サービスを変更します(フォトグラファーのみ)
- サービスを削除します(フォトグラファーのみ)
- サービスを観ます

• ユーザーインターフェース

Dịch vụ của tôi

Tên gói: GÓI CHỤP 1 NGƯỜI

Bao gồm các dịch vụ:

- ✓ Khách được chụp 3 trang phục/người
- ✓ Khách nhận toàn bộ ảnh gốc (khoảng 100-200 ảnh) và 30 ảnh chỉnh sửa

Tổng cộng:

1.490.000 đồng

Chỉnh sửa

Xóa

Tên gói: GÓI CHỤP NHÓM (Giá / người)

Bao gồm các dịch vụ:

- ✓ Giá in kỷ yếu photobook rẻ nhất TPHCM

+



図154 – サービスを観ます

← Tạo dịch vụ

Kiểu dịch vụ: *

Trong ngày Nhiều ngày

Tên dịch vụ: *

Ví dụ: Gói 01

Mô tả:

Nhập mô tả cho gói dịch vụ của bạn

Thời gian tác nghiệp: *

Ví dụ: 3 giờ

Giá thành dịch vụ (đơn vị: đồng): *

Ví dụ: 10.000.000 đ

Phương thức giao hàng: *

Giao hàng qua ứng dụng

Gặp mặt khách hàng

Chi tiết dịch vụ: *

Thêm dịch vụ +

図155 - サービスを追加します

← Chính sửa dịch vụ

Tên dịch vụ: *
GÓI CHỤP 1 NGƯỜI

Mô tả:
Chụp ảnh cổ trang có rất nhiều dịch vụ chụp ảnh khác nhau, do đó bạn có thể chọn gói chụp theo nhu cầu hoặc mức kinh phí có sẵn

Giá thành dịch vụ: *
1.490.000 đ

Thời gian tác nghiệp: *
6 giờ

← Chính sửa dịch vụ

Thời gian tác nghiệp: *
6 giờ

Phương thức giao hàng: *

Giao hàng qua ứng dụng

Gặp mặt khách hàng

Chi tiết dịch vụ: *

Thêm dịch vụ +

Khách được chụp 3 trang phục/người ×

Khách nhận toàn bộ ảnh gốc (khoảng 100-200 ảnh) và 30 ảnh chỉnh sửa ×

図156 – サービスを変更します

6.3.12. カテゴリーの管理

- **目標**
 - 客がカテゴリーを観たり、選択します
- **機能**
 - カテゴリーを観ます
 - カテゴリーを選択します
- **ユーザーインターフェース**

X Tạo Album ✓

Tên Album: *

☰ Ví dụ: Album 01

Thời gian: *

📅 Ví dụ: 01/01/2011

Địa điểm: *

📍 Ví dụ: Hà Nội

Mô tả: *

☰ Ví dụ: Album này ...

Category: *



anh cưới



phong cảnh



kỉ yếu

図157 – カテゴリーを観ます／選択します

6.3.13. チャット

- 目標
 - 客とフォトグラファーが簡単に連絡します
- 機能
 - チャット
- ユーザーインターフェース

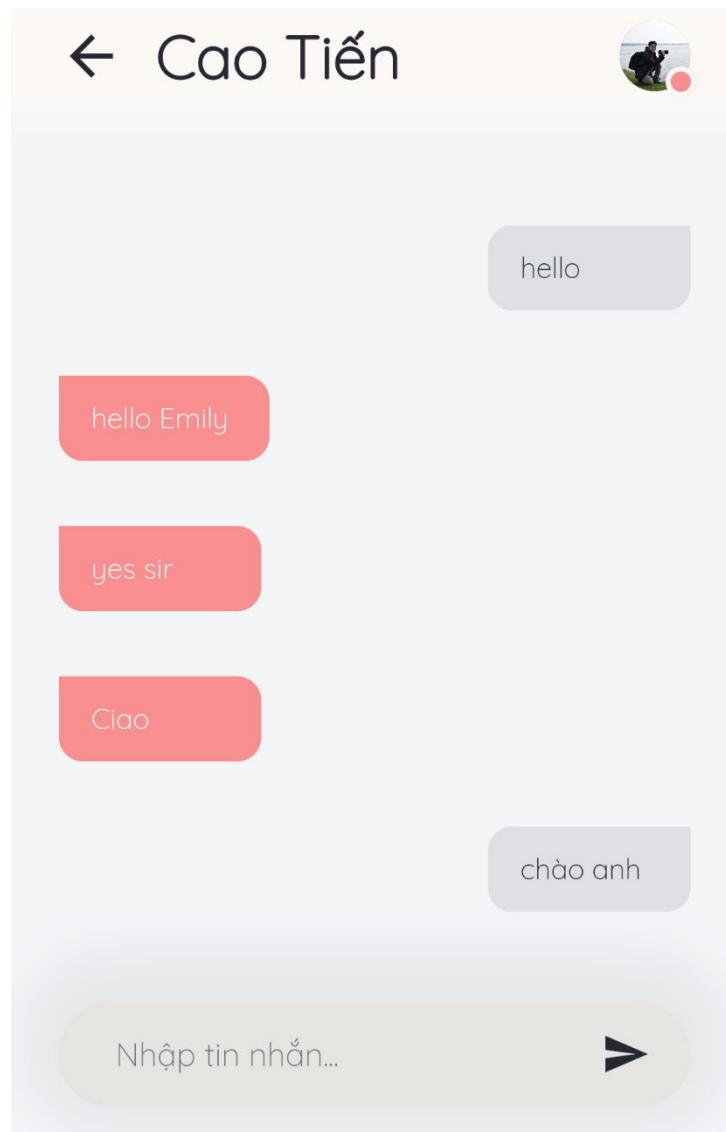


図158 – チャット

6.3.14. フォーラム

• 目標

- ユーザーがフォーラムのコンテンツを觀ます
- スレッドにコメントします
- スレッドを管理します
- カテゴリーを管理します

• 機能

- スレッドを觀ます
- コメントを觀ます
- スレッドを追加します
- スレッドを編集します
- スレッドをデリートします
- コメントを追加します
- コメントを編集します
- コメントをデリートします

- ユーザーインターフェース

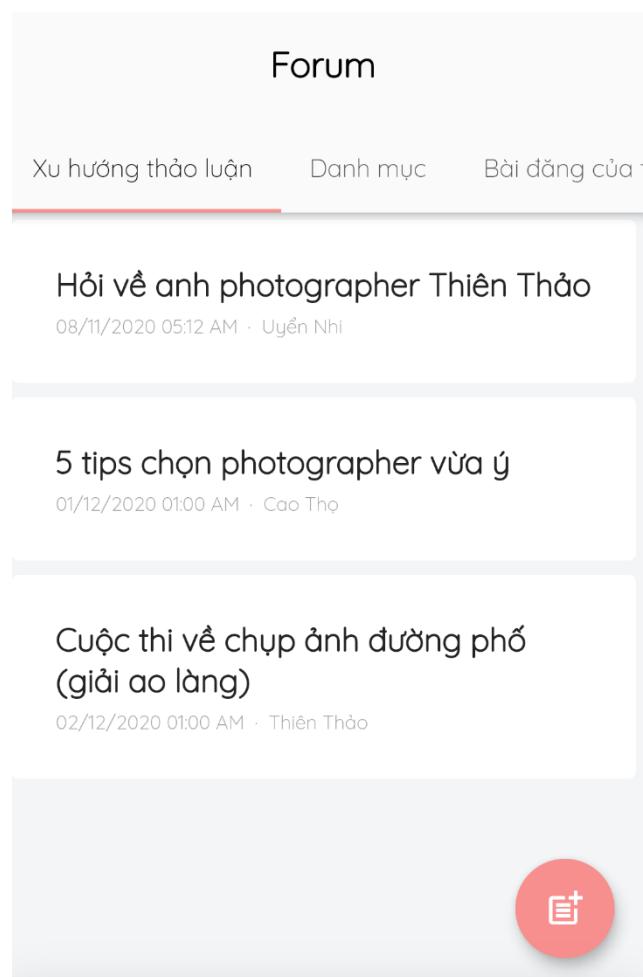


図159 – スレッドを觀ます

← Cuộc thi về chụp ảnh đường ...



Thiên Thảo

02/12/2020 01:00 AM

Cuộc thi về chụp ảnh đường phố (giải ao làng)

- 👉 Đơn vị tổ chức: Anh em hội PBS
- 👉 Đơn vị đồng tổ chức: Photographer.vn
- 👉 Đơn vị tài trợ: cafechupanh
- 👉 CƠ CẤU GIẢI THƯỞNG
- 🥇 Giải 1 : 10 triệu
- 🥈 Giải 2 : 6 triệu
- 🥉 Giải 3 : 3 triệu
- 🏅 Giải 4 : 1 triệu

TRÍCH · ĐIỀU HÀNH



Cao Thọ

02/12/2020 01:30 AM

Ao làng mà chất lượng thế anh

TRÍCH



図160 – コメントを觀ます

×

Trả lời

✓

Viết bình luận của bạn...

Hình ảnh

+

図161 – コメントを追加します



図162 – スレッドを追加します

6.3.15. 配信方法の管理

- **目標**
 - 管理者が配信方法の管理します
 - 客やフォトグラファーが配信方法を観たり、選択します
- **機能**
 - 配信方法を選択します(客やフォトグラファー)
- **ユーザーインターフェース**

← Chính sửa dịch vụ

Thời gian tác nghiệp: *

6 giờ

Phương thức giao hàng: *

Giao hàng qua ứng dụng

Gặp mặt khách hàng

Chi tiết dịch vụ: *

Thêm dịch vụ +

Khách được chụp 3 trang phục/người ×

Khách nhận toàn bộ ảnh gốc (khoảng 100-200 ảnh) và 30 ảnh chỉnh sửa ×

図163 - 配信方法を選びます

6.3.16. プロファイルの管理

- 目標
 - ユーザーがプロファイルを編集します
- 機能
 - プロファイルを編集します
 - アバターやカバーを変更します
- ユーザーインターフェース

✗ Chỉnh sửa trang cá nhân ✓

Họ tên: *



Ví dụ: Nguyễn Văn A

Email: *



Ví dụ: abc@xzy.com

Số điện thoại: *



Ví dụ: 012345678

Địa điểm làm việc: *



Ho chi minh



Ha noi



Da lat



Mô tả: *



Ví dụ: Tôi là....

図164 – プロファイルを編集します

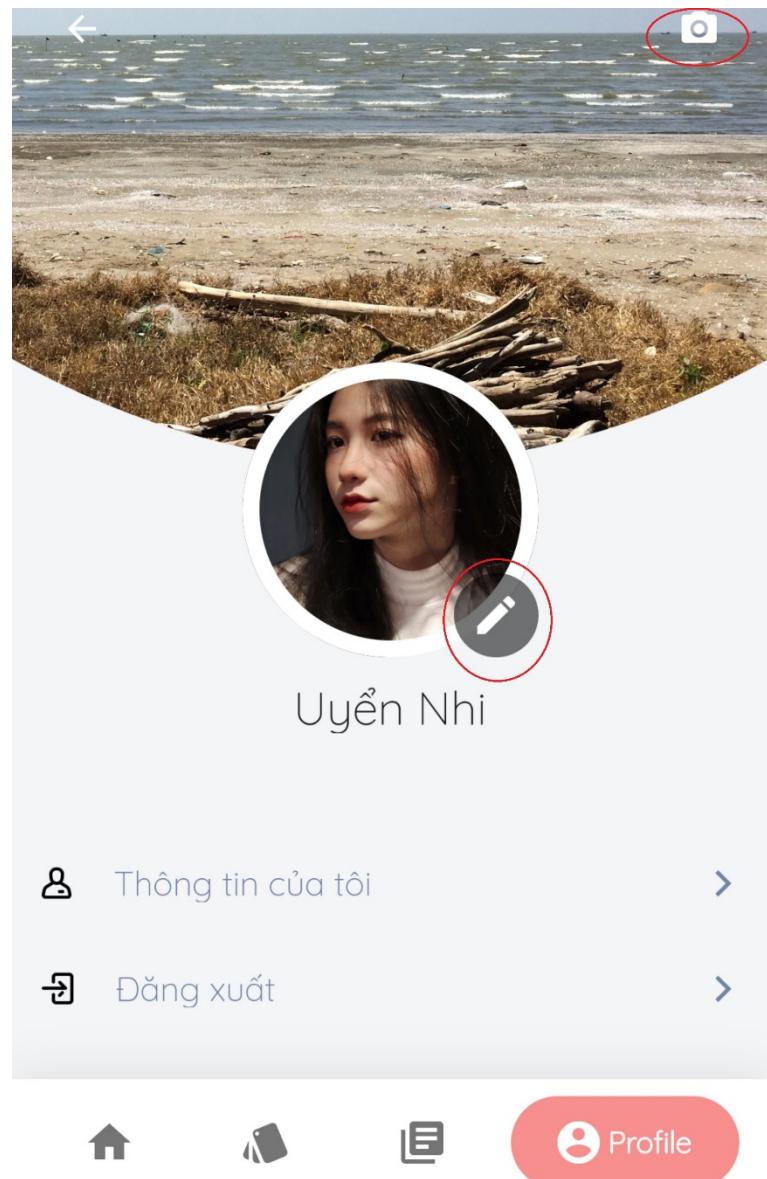


図165 – アバター／コバーを変更します

6.4トラブルシューティング

1.現在の場所を特定できない場合があります

最初の発生

- テストの間に

問題

- ケイタイの現在の場所は許可されていません

回避策

次のステップを実行します:

- 設定> 現在の場所 > オン
2. プロファイルのイメージを失敗しに変更する場合があります

最初の発生

- テストの間に

問題

- イメージのサイズが最大許容サイズを超えています

回避策

- 他の小さなイメージを選択します

VII. Appendix

1. Glossary [Optional]

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

Table 127 Glossary

2. References [Optional]

1. Pub.Dev: <https://pub.dev/>
2. Bloc Library: <https://bloclibrary.dev/#/>
3. Diagrams.net: <https://app.diagrams.net/>
4. Lucidchart: www.lucidchart.com
5. StarUML: <https://staruml.io/>
6. Xmind: <https://www.xmind.net/>
7. MySQL: <https://www.mysql.com/>
8. MySQL Workbench: <https://www.mysql.com/products/workbench/>
9. Android Studio: <https://developer.android.com/studio>
10. Visual Studio Code: <https://code.visualstudio.com/>
11. IntelliJ: <https://www.jetbrains.com/idea/>

12. Firebase Cloud Messaging: <https://firebase.google.com/docs/cloud-messaging>
13. Firebase Realtime Database: <https://firebase.google.com/docs/database>
14. Postman: <https://www.postman.com/>
15. Amazon Simple Storage Service: <https://aws.amazon.com/vi/s3/>

3. Others [Optional]