



TRƯỜNG ĐẠI HỌC FPT

# Capstone Project Document

## BAKERY STORE ONLINE

Bakery Store Online		
<b>Group member</b>	Phạm Thị Minh Phương	SE03773
	Cao Anh Tuấn	SE03737
	Hoàng Anh Phương	SE03715
	Ngô Trung Kiên	SE03736
	Nguyễn Xuân Hà	SE03484
<b>Supervisor</b>	Mr. Nguyễn Văn Sang	
<b>Project code</b>	BSO	

## TABLE CONTENTS

<b>1 INTRODUCTION .....</b>	<b>2</b>
1.1 Purpose.....	2
1.2 Acronyms and Definitions.....	2
1.3 Abstract.....	2
1.4 Literature Review.....	2
1.5 Proposal .....	4
1.6 Benefits from Project .....	7
1.7 Critical Assumption and Constraints .....	8
1.8 Potential Risks.....	8
<b>2 PROJECT MANAGEMENT .....</b>	<b>9</b>
2.1 Introduction.....	9
2.2 Project Overview.....	9
2.3 Project Development Approach.....	15
2.4 Estimation .....	18
2.5 Project Organization.....	23
2.6 Communication & Reporting .....	25
<b>3 SOFTWARE REQUIREMENT.....</b>	<b>27</b>
3.1 Introduction.....	27
3.2 Software Requirements Specification.....	28
<b>4 SOFTWARE DESIGN .....</b>	<b>94</b>
4.1 Introduction.....	94
4.2 Architecture design .....	94
4.3 Detail Design.....	108
4.4 Data design.....	180
4.5 Screen Design .....	185
<b>5 SOFTWARE TESTING .....</b>	<b>224</b>
5.1 Introduction.....	224
5.2 Test plan.....	226
5.3 Test case.....	233
5.4 Test Report.....	233
<b>6 USER MANUAL .....</b>	<b>235</b>
6.1 Introduction.....	235
6.2 Installation Guideline .....	235
6.3 User Guideline .....	245
6.4 Staff.....	266
<b>7 PROJECT RESULT REPORT .....</b>	<b>270</b>
7.1 Product Judging.....	270
7.2 Project Summary .....	271
7.3 Lesson Learnt.....	272
<b>8 REFERENCES .....</b>	<b>273</b>

## 1 INTRODUCTION

### 1.1 Purpose

This document is created as the introduction for project Bakery Store Online— our Capstone Project at FPT University. In this document, we will describe the overview of some existing systems, the initial idea for our project, a brief description about our expected system and some potential risks, critical assumptions, constraints. Moreover, this document also shows opportunities what it offers for users.

### 1.2 Acronyms and Definitions

Acronym & Abbreviation	Definition
<b>BSO</b>	Bakery Store Online
<b>FU</b>	FPT University
<b>Q&amp;A</b>	Question and Answer

**Table 1-1: Definitions and Acronyms**

### 1.3 Abstract

Bakery Store Online a standalone application which is based on ordering and selling the cake and other items and generating bill. The main principle behind the need of Bakery Store Online is easy supervision of shop. It has user friendly & modular approach. The modular approach of the software increases the flexibility of the software. Data storing is easier. It will be able to check any report at any time. Paper work and manual work is reduced. The system is user friendly and easy to use. The record of each customer is stored that is customer's name and contact details are added for reference. Next the item is selected, flavours, pounds and quantity is also added. At the end of the day, report is generated to calculate the payment for each user in each day..

### 1.4 Literature Review

There are quite a few major platform for Sales Online your project idea. However just have one Sales Online in Vietnam and that major platform just aim to project. For example: parisgateaux.vn, madamehuong.vn

### 1.4.1 Paris Gateaux

**PARIS BAGUETTE**

MENU ABOUT US CONTACT US FRANCHISING ORDER NOW CAREERS

FIND A LOCATION

*Bread and Pastries*

Our chefs bake fresh loaves of bread daily at each location.  
Our talented chefs' skills and love for bread help make wonderfully fresh-baked bread every single day.

BREAD & PASTRIES | CAKES & DESSERTS | SPECIALTY DRINKS | SANDWICHES & SALADS | GIFT ITEMS

Mini Swirl Pain au Chocolate 340 Cal.	Mozzarella Onion Bread 710 Cal.	Sweet Veggie Bread 280 Cal.	Mocha Cream Cheese Bread 210 Cal.
Black & Blueberry Cream Pastry 640 Cal.	Apple Bavarian Cream Pastry 690 Cal.	Almond Meringue Pastry 420 Cal.	Cinnamon Sugar Coffee Bun 440 Cal.

- Advantages:
  - There are many different types of cakes
  - Friendly site, easy to use
- Disadvantages:
  - Can not see the price of the product
  - Can not place order on website
  - Not support online payment

### 1.4.2 Madame Huong

The screenshot shows the homepage of Madame Huong's website. At the top, there is a navigation bar with links to HOME, GIỚI THIỆU, QUÀ TẶNG, CẨU CHUYỆN, BỘ SƯU TẬP, HOTLINE: 0902544888, and CÁC ĐIỂM BÁN. The main title is "BÁNH TRUNG THU MADAME HUONG". Below it, a sub-section title reads "Madame Huong hân hạnh giới thiệu Bộ sưu tập bánh trung thu 2017 mang tên Hà Nội Phố!". There are three product cards displayed:

- ĐỒNG XUÂN PHỐ - GIÁ: 320.000 VNĐ**  
Hộp 4 bánh 120gr: Sen xát, Cốm xanh, Dưa Mỹ, Đậu xanh
- ĐỒNG XUÂN PHỐ - GIÁ: 320.000 VNĐ**  
Hộp 4 bánh 120gr: Sen xát, Cốm xanh, Dưa Mỹ, Đậu xanh
- ĐỒNG XUÂN PHỐ - GIÁ: 320.000 VNĐ**  
Hộp 4 bánh 120gr: Sen xát, Cốm xanh, Dưa Mỹ, Đậu xanh

- Advantages:
  - Friendly site, easy to use
  - Can see the price of product in the website
- Disadvantages:
  - Can not place order on website
  - Only moon cake

## 1.5 Proposal

### 1.5.1 The idea

The need to enjoy delicious cakes is the constant need of every person, now we make our work become art, the more stimulated the demand is increasing. Society is growing thanks to the internet, we do not have to run out to buy bread.

Bakery Store Online - is an online website that sells cakes. With a friendly interface and easy to use, anyone can buy the cake they need (for example, birthday cake, biscuits, diabetic cake ...). In addition, the site shares some simple recipes that people can reference and perform at home

### 1.5.2 The objective

This project is the Capstone Project in FPT University studying the program.

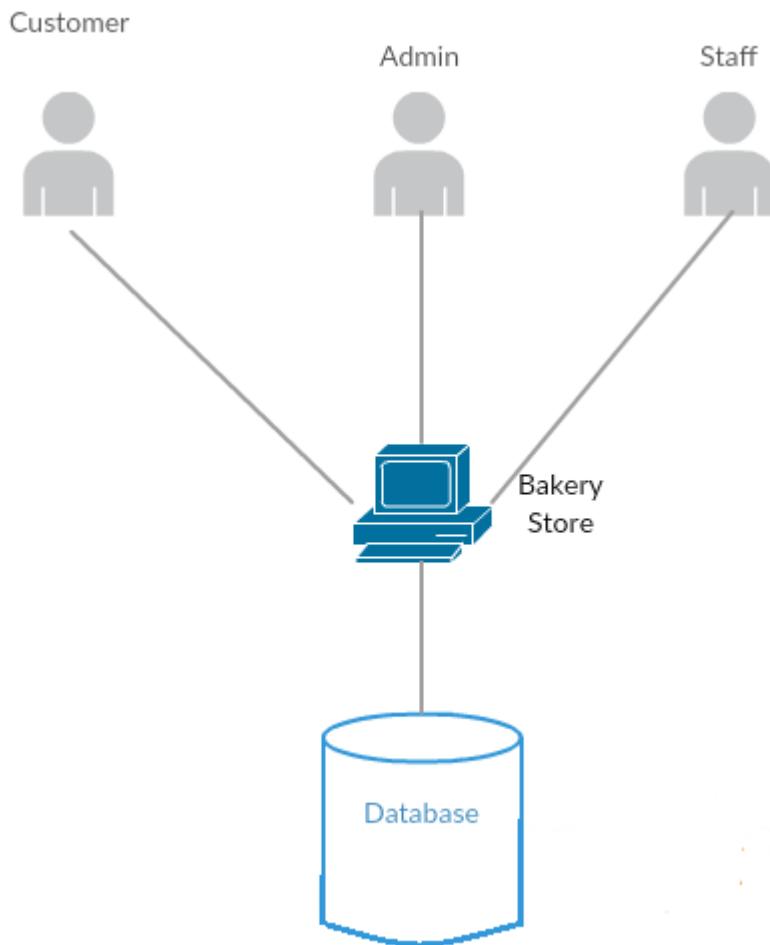
This project is responsible for all team members. So, we must complete all requirements from teacher and FU. During 3 months of capstone project, we hope that all member can get more knowledge about processing of software development, skill do document, coding, testing. Moreover, soft skill also very important, we can get skill management time, working in group, skill communication... It is not long time for project but our group hope that can do this project

successfully. This project also makes our CV more beautiful after graduating from FU. This is also an opportunity for us to prove capacity with recruiters.

### 1.5.3 Brief description about system

This site will promote the business and increase the store's sales. With a friendly and attractive interface, users can easily find or reference the cake samples.

Bakery Store Online uses familiar technologies as: Bootstraps, MVC5, NodeJS... And we are planning to develop a mobile version for easily connecting.



**Figure 1-1: Brief descriptions diagram of Bakery Store Online system**

### 1.5.4 System features

About the features of Bakery Store Online, we focus to provide for users main features as searching, login, order, etc. Admin can manage users, manage staff, manage database, manage cake and manage blog.

In here, we focus on features which can help users understand the best way what user want to have with features of Bakery Store Online. There are the detailed features of Bakery Store Online.

#### 1.5.4.1 Client features

- Register: User can register an account and login to use all features of BSO.
- Login/Logout: Users login/logout an account to use or exit system BSO .
- Search: User can search with name of cake and system will display all of cake relate to keywords.
- Filter: User can filter with category (Introduce, Product, News, Service, Blog, Contact). And system will display projects relate to these selected options.
- Cake:
  - Shopping cart: after choose the cake and order, the cake will be in the cart.
  - View order history: after order success, user can review all the cake ordered.
- Profile:
  - Change password: User can change password to keep security.
  - Update profile: User can change or update information.

#### 1.5.4.2 Admin features

- Login/Logout: Admin login/logout an account to use or exit system BSO.
- Manage customer:
  - Search customer: Admin can search with name of customer and system will display simple information about that customer(name, email, phone number...)
  - Active/Deactivate user: admin can set customer's account to active or deactivate.
- Manage staff:
  - Search staff: Admin can search with name of staff and system will display simple information about that staff(name, email, phone number...)
  - Active/Deactivate staff: admin can set staff's account to active or deactivate.
- Manage cake:
  - Search cake: Admin can search with name of cake and system will display information about cake (name, size, cost,...)
  - Update cake: admin can change information about cake (name, size, number,...)
  - Add cake: admin can add new cake
  - Block cake: admin can block cake when it will not sale
  - Unblock cake: admin can unblock cake when it will sale
- Manage blog:
  - Search blog: Admin can search with title of blog and system will display blog
  - Add blog: admin can add new post
  - Block blog: admin can block blog
  - Update blog: admin can update blog
- Manage event:
  - Search event: Admin can search with title of event and system will display event
  - Add event: admin can add new event

- Block event: admin can block event
- Update event: admin can update event
- Manage category:
  - Search category: Admin can search with title of category and system will display category
  - Add category: admin can add new category
  - Block category: admin can block category
  - Update category: admin can update category
- Manage benefit:
  - Search by date: admin can search benefit by date of BSO
  - Search by from date to date: admin can search benefit by from date to date of BSO
  - View benefit: admin can view detail benefit of BSO

#### 1.5.4.3 Staff features

- Login/Logout: Staff login/logout an account to use or exit system BSO.
- Forgot password: when staff forgot their password to login the system, this function will receiving their password to email.
- Manage order:
  - Search order: staff can search order of staff
  - Confirm order: staff can confirm order transaction successful
  - Unconfirm order: staff can confirm order transaction unsuccessful
  - Cancel order: staff can cancel order of customer
- Update profile: staff can edit profile

## 1.6 Benefits from Project

### 1.6.1 For our group

After developing and implementing this project, our group will get some benefits:

- Have more experiences of software project management how to manage plan, time, member and risk.
- Have more knowledge and skill about Node.js MVC5
- Know how to communicate with team members and how to teamwork more effective.

### 1.6.2 For Community

Just need to have a device that connect to the internet and any web browser, user can:

- Easy buy cake
- Easy receive cake

## 1.7 Critical Assumption and Constraints

- Critical assumption:
  - Training: All developers can train Node.js in 2 weeks. Before starting this project, just 1 member in our team have knowledge about Node.js, but with programming techniques which we studied at FU, we have to try the best to not missing any deadline in project plan.
  - Human resources: Assume that all members in team have a good healthy to do their tasks.
- Constraints:
  - Time and deadline: We must complete task on time. We work on 14 weeks, each member works 5 hours/day and 5 days/week. We do not have more time for us to complete developing and deliver application to teachers. Besides, we have to submit report documents before deadline to teacher can review.
  - Quality: The products must be run well when users perform main functions in Google Chrome 40 and Firefox 30 or higher.
  - Process: We have to follow the software processing of FPT Software
  - Human resources: There are 5 member in our team, each member have to study 2 subjects (Japanese and JFE) at school.

## 1.8 Potential Risks

After studying about this project, we find out some problem that we may be encountered:

- Under-estimate scope and time or miss deadline because lack of experience in group working, managing and controlling work.
- Loss data, source code before uploading data to server because of careless.
- Equipment got broken because of careless or accident.
- Human resources: Team member cannot complete their works because of health reasons, key member leave team or un-cooperating on team.
- Change requirements: Requirement changed when some functions cannot be completed or some technologies is not suitable.

## 2 PROJECT MANAGEMENT

### 2.1 Introduction

#### 2.1.1 Purpose

This part is the project management plan of Bakery Store Online (BSO) Project – our Capstone Project in FPT University. It is included the project overview, project organization, tools and infrastructures, schedule of this project.

#### 2.1.2 Definitions and Acronyms

Acronym	Definition	Note
CC	Infrastructure Configuration Controller	
CCB	Change Control Board	
CI	Configuration Item	
CM	Configuration Management	
CSCI	Computer Software Configuration Items	
DEV	Developer	
PIC	Person in charge	
PM	Project Manager	
PTL	Project Technical Leader	
QA	Quality Assurance Officer	
SRS	Software Requirement Specification	
TP	Test Plan	
TC	Test Case	

Table 2-1: Definitions and acronyms

### 2.2 Project Overview

#### 2.2.1 Project Description

<b>Project Code</b>	<b>BSO</b>	<b>Contract Type</b>	<b>None</b>
<b>Customer</b>	FPT University	<b>2nd Customer</b>	None
<b>Project Level</b>	Group	<b>Project Rank</b>	None
<b>Application Type</b>	Website	<b>Project Manager</b>	Pham Thi Minh Phuong
<b>Project Category</b>	Development	<b>Business Domain</b>	E-commerce

Table 2-2: Project Description

## 2.2.2 Scope and Purpose

### 2.2.2.1 Purpose

The need to enjoy delicious cakes is the constant need of every person, now we make our work become art, the more stimulated the demand is increasing. Society is growing thanks to the internet, we do not have to run out to buy bread.

Bakery Store Online- is an online website that sells cakes. With a friendly interface and easy to use, anyone can buy the cake they need (for example, birthday cake, biscuits, diabetic cake ...). In addition, the site shares some simple recipes that people can reference and perform at home.

### 2.2.2.2 Scope

The scope of this project contains: Requirement Analysis, Design, Coding and Testing (Unit Test, Integration Test, and System Test).

### 2.2.2.3 The functions of Project

These are the functions of BSO project:

#### 2.2.2.3.1 Client Module.

- Register: User can register an account and login to use all features of BSO.
- Login/Logout: Users login/logout an account to use or exit system BSO.
- Search: User can search with name of cake and system will display all of cake relate to keywords.
- Filter: User can filter with category (Introduce, Product, News, Service, Blog, Contact). And system will display projects relate to these selected options.
- Cake:
  - Shopping cart: after choose the cake and order, the cake will be in the cart.
  - View order history: after order success, user can review all the cake ordered.
- Profile:
  - Change password: User can change password to keep security.
  - Update profile: User can change or update information.

#### 2.2.2.3.2 Admin Module

- Login/Logout: Admin login/logout an account to use or exit system BSO.
- Manage customer:
  - Search customer: Admin can search with name of customer and system will display simple information about that customer(name, email, phone number...)
  - Active/Deactivate user: admin can set customer's account to active or deactivate.
- Manage staff:
  - Search staff: Admin can search with name of staff and system will display simple information about that staff(name, email, phone number...)
  - Active/Deactivate staff: admin can set staff's account to active or deactivate.
- Manage cake:
  - Search cake: Admin can search with name of cake and system will display information about cake (name, size, cost,...)
  - Update cake: admin can change information about cake (name, size, number,...)
  - Add cake: admin can add new cake

- Block cake: admin can block cake when it will not sale
- Unblock cake: admin can unblock cake when it will sale
- Manage blog:
  - Search blog: Admin can search with title of blog and system will display blog
  - Add blog: admin can add new post
  - Block blog: admin can block blog
  - Update blog: admin can update blog
- Manage event:
  - Search event: Admin can search with title of event and system will display event
  - Add event: admin can add new event
  - Block event: admin can block event
  - Update event: admin can update event
- Manage category:
  - Search category: Admin can search with title of category and system will display category
  - Add category: admin can add new category
  - Block category: admin can block category
  - Update category: admin can update category
- Manage benefit:
  - Search by date: admin can search benefit by date of BSO
  - Search by from date to date: admin can search benefit by from date to date of BSO
  - View benefit: admin can view detail benefit of BSO

### 2.2.2.3.3 Staff Module

- Login/Logout: Staff login/logout an account to use or exit system BSO.
- Forgot password: when staff forgot their password to login the system, this function will receiving their password to email.
- Manage order:
  - Search order: staff can search order of staff
  - Confirm order: staff can confirm order transaction successful
  - Unconfirm order: staff can confirm order transaction unsuccessful
  - Cancel order: staff can cancel order of customer
- Update profile: staff can edit profile

### 2.2.3 Constraints

No	Description	Note
<b>Constraints</b>		
1	This project must be completed and delivered before 25/12/2017	Schedule
2	In doing project processing, PM must submit report (include 6 reports) on certain date.	Schedule
3	Software Requirement Specification Document and Project Plan must be completed within 17 days since 09/09/2017 <b>Deadline:</b> 27/09/2017	Schedule
4	Design Document (include Architecture Design, Screen Design, Database Design) must be completed within 18 days since 25/09/2017 <b>Deadline:</b> 16/10/2017	Schedule
5	Integration Test Plan (include test plan and test case...) must be completed within 15 days since 25/10/2017 <b>Deadline:</b> 10/11/2017	Schedule
6	Completed coding activity and have unit test result within 22 days since 30/10/2017 <b>Deadline:</b> 28/11/2017	Schedule
7	Deliver report about User manual, software package and installation guide on 5 days since 08/12/2017 <b>Deadline:</b> 13/12/2017	Schedule
8	Complete all of document and application before finishing the project on 22/12/2017	Schedule
9	Project contains 5 members	Resource

**Table 2-3: Project Description****2.2.4 Project Objectives****2.2.4.1 Standard Objectives**

Metrics	Unit	Committe d	Re-committed	Note
Start Date	06/09/2017			
End Date	25/12/2017			
Duration	Day	81		
Maximum Team Size	Person	5		
Calendar effort	Person-day	405		1 Person-day = 5 hours
Effort Usage	Person-day	100		1 Person-day = 5 hours

**Table 2-4: Standard Objectives**

Metrics	Unit	Basic for setting Goals
Customer Satisfaction	Point	9.5
Leakage	def/mm	5
Effort Efficiency	%	80
Timeliness	%	100

**Table 2-5: Standard Objectives 2**

#### 2.2.4.2 Specific Objectives

Metrics	Unit	Basic for setting Goals	
		Plan	Actual
Training technology: MVC, Bootstrap, Mongod, Node.js	Person-day	15	15
Execute group review	Person-day	8	5
Training requirements, process before coding	Person-day	8	5

**Table 2-6: Specific Objectives**

#### 2.2.5 Critical Dependencies

No	Dependency	Expected delivery date	Note
1	This project must be completed and delivered to FPT University.	25/12/2017	
2	Project Plan and SRS must be completed and delivered to Supervisor.	27/09/2017	
3	User manual, Software Package and Installation Guide must be completed and delivered to Supervisor and FPT University.	13/12/2017	
4	Beside Capstone Project, Team members have to joining in Japanese class and Japanese Fundamental Exam class.	06/09/2017	

**Table 2-7: Critical Dependencies**

## 2.2.6 Project Risk

PM identifies risks in the Risk Management Plan. The document is updated to trigger each milestone, each event also. The document is updated weekly by the PM, Risk Management Plan will be notified to all of the stakeholders affected. Status of risk is reported to supervisor at Project Milestones Report.

Reference to [BSO\\_Risk Management Plan\\_v1.0\\_EN.xlsx](#)

## 2.3 Project Development Approach

### 2.3.1 Requirement Change Management

<b>Who logs the change request?</b>	<b>Any team members</b>
<b>Who reviews the change request?</b>	PM or who is PM assign
<b>Who approves the change request?</b>	PM by default. PTL if: <ul style="list-style-type: none"> <li>▪ Changes to project scope</li> <li>▪ Changes in delivery plan of project deliverable</li> <li>▪ Changes to assignment for key roles (PM, PTL)</li> </ul>

**Table 2-8: Requirement Change Management**

### 2.3.2 Quality Management

#### 2.3.2.1 Defect Prevention Strategy

Item (Process/Product)	Strategy	Expected Benefits
Requirement missing	List up all of requirement into SRS document.	10–20% reduction in defect injection rate and about 2% improvement in productivity
Careless mistake in Design Document Format/Template wrong	After designing, QA will review Document Format base on checklist review design	Improvement in quality as overall defect removal efficiency will improve; some benefits in productivity as defects will be detected early
Use wrong template	Have a meeting to disseminate all template that is used in this project for all member	All member will use right template when do document
Coding application does not match with User Requirement.	Develop Team must study about Requirement/Design within 1 weeks since project is assigned. PM and PTL has responsibility to review task results and explain User Requirement for Develop Team	Coding Application match with User Requirement.

**Table 2-9: Defect Prevention Strategy**

### 2.3.2.2 Review Strategy

Review Item	Reviewer	Review Type	Review Method	Completion Criteria
Project plan Project schedule CM Plan	PM, QA, Supervisor	Group review Group review One-person review	Use checklist and Self-review	
Business analysis and requirements specification document, Use Case catalog	PM, QA, Supervisor	Group review	Use checklist	
Design document, object model	Self-review, PM, QA, Supervisor	Group review	Use checklist	
Stage plans	PM, QA, Supervisor	One-person review	Use checklist	
Complex/first time generated program specs incl. test cases, interactive diagrams		Group review		
Code	Self-review or Team Lead review or Peer review	Group review	Self-review and use checklist	

**Table 2-10: Review Strategy**

### 2.3.2.3 Integration Testing Strategy

Item to be Integration Tested	Integration Test Type	Integration Test Technique	Tool Used	Completion Criteria
Do test by flow of functions and items which have concern each other	Black-Box Test		Checklist, Boundary	- Number of UTC/KLOC: 20  - Number of defects/KLOC: 2-3

**Table 2-11: Integration Testing Strategy**

### 2.3.2.4 System Testing Strategy

Item to be System Tested	System Test Type	System Test Technique	Tool Used	Completion Criteria

Test whole system	Black-Box Test		None	-Number of UTC/KLOC: 60 -Number of defects/KLOC: 4-6
-------------------	----------------	--	------	---

**Table 2-12: System Testing Strategy****2.3.2.5 Estimates of Defects to be detected**

Review/Testing Stage	Targeted No. of Defects to be detected	% of Defects to be detected	Basic for Estimation
Requirements review	10	12%	Referenced to similar project estimations
Design review	15	18%	Referenced to similar project estimations
Code review	30	35%	Referenced to similar project estimations
Integration Test	15	18%	Referenced to similar project estimations
System Test	10	12%	Referenced to similar project estimations
User Acceptance Test	5	6%	Referenced to similar project estimations
Total	85	100%	

**Table 2-13: Estimates of Defects****2.3.2.6 Measurements Program**

Data to be collected	Purpose	PIC	When
Size: No. of KLOC	Achieve target	PM	At the end of stages
Effort: No. person-day	Monitor and controlling team member to keep plan.	Team members	Daily
Quality: No. defects detected	Managing product's quality.	Reviewer Tester	Right after the review/test
Schedule	Monitor and controlling software developing processing keep plan.	PM	Weekly and at the end of stages

**Table 2-14: Measurements Program**

## 2.4 Estimation

### 2.4.1 Size

This project is performed and must complete all requirements from teacher and FPT University. So size of our project is in Capstone Project limit.

### 2.4.2 Effort

The Effort estimation is documented in the table below:

	Initiation	Solution	Construction	Termination	Total
<b>Effort(person/day)</b>	72	168	186	54	<b>480</b>
<b>Total % budgeted Effort Usage (%)</b>	100	100	100	100	

**Table 2-15: Effort Estimation**

### 2.4.3 Schedule

#### 2.4.3.1 Project Milestone & Deliverables

No	Stage	Committed Delivery date	Description of Deliverable	Delivery media
<b>Initiation</b>			Requirements agreed, Report 1 reviewed	
1	Develop project idea	07-09-2017	Project goals and scope defined, milestone description defined, resource committed	SmartGit
2	Q&A Management Sheet	13-09-2017	Criteria: Documentation reviewed	SmartGit
3	Submit report no.1 final	21-09-2017	Completed report no.1	SmartGit
4	Project Plan	21-09-2017	Criteria: Documentation reviewed	SmartGit
5	Submit report no.2 final	17-10-2017	Completed report no.2	SmartGit
<b>Solution</b>				
1	Screen Prototype	04-10-2017	Criteria: Documentation reviewed	SmartGit
2	Architecture Design	09-10-2017	Criteria: Documentation reviewed	SmartGit
3	Screen Design	11-10-2017	Criteria: Documentation reviewed	SmartGit
4	Class Design	16-10-2017	Criteria: Documentation reviewed	SmartGit
5	Detail Data Design	05-10-2017	Criteria: Documentation reviewed	SmartGit
6	SRS	27-09-2017	Criteria: Documentation reviewed	SmartGit
7	Submit Report no.3 Final	26-10-2017	Completed report no.3	SmartGit
8	Submit Test Plan Final	25-10-2017	Criteria: Documentation reviewed	SmartGit
9	Create Test Case	10-11-2017	Criteria: Documentation reviewed	SmartGit

10	Submit Report no.4 Final	21-11-2017	Completed report no.4	SmartGit
<b>Construction</b>			Product developed & tested and released to supervisor, documentation reviewed.	
1	Review Test Document	09-11-2017	Criteria: Documentation reviewed	SmartGit
2	Submit report no.5 Final	10-12-2017	Completed report no.5	SmartGit
3	Complete Coding and Unit Test	20-11-2017	Source code Acceptance criteria: Product unit tested	SmartGit
4	Complete Testing	04-12-2017	Completed Test	SmartGit
5	Submit report no.6 Final	12-12-2017	Completed report no.6	SmartGit
6	Submit the last document and CD source code	12-12-2017	Final Documents and Source Code	SmartGit
<b>Termination</b>			Project post-mortem is conducted, Project assets archived and released to supervisor	
1	Lesson learned	11-12-2017	Criteria: Completed	SmartGit
2	Complete Presentation Slide	13-12-2017	Criteria: Completed	SmartGit
3	Represent capstone project	22-12-2017	Criteria: Completed	SmartGit
4	Project Complete	20-12-2017	Criteria: Completed	SmartGit

**Table 2-16: Project Milestone and Deliverables**

### 2.4.3.2 Detail Schedule

The detail project schedule is available in <...>. The Project Schedule is weekly updated by the Project Manager.

No.	Activity	Start date	Responsible	Note
<b>Defect Prevention</b>				
1	Training coding convention Node.js	24-09-2017	24-09-2017	Cao Anh Tuan
2	Training for Q&A and tester to use checklist	15-10-2017	16-10-2017	Pham Thi Minh Phuong
<b>Quality Control</b>				
1	Group review requirement	14-10-2017	16-10-2017	Pham Thi Minh Phuong
2	Group review design	07-10-2017	09-10-2017	Ngo Trung Kien
3	Group review coding	22-11-2017	24-11-2017	Nguyen Xuan Ha
<b>Project Tracking</b>				
1	Solution: Milestone review meeting	27-10-2017	28-10-2017	Pham Thi Minh Phuong
2	Construction: Milestone review meeting	06-12-2017	08-12-2017	Pham Thi Minh Phuong
3	Transition: Milestone review meeting	14-12-2017	15-12-2017	Pham Thi Minh Phuong
<b>Configuration Management</b>				
1	Baseline code	17-10-2017	20-10-2017	Nguyen Xuan Ha
2	Base line test report, test case and test plan	22-10-2017	29-10-2017	Pham Thi Minh Phuong
<b>QA</b>				
1	Final Inspection: Report 1	07-09-2017	22-09-2017	Pham Thi Minh Phuong
2	Final Inspection: Report 2	22-09-2017	19-10-2017	Pham Thi Minh Phuong

3	Final Inspection: Report 3	20-10-2017	28-10-2017	Pham Thi Minh Phuong
4	Final Inspection: Report 4	29-10-2017	25-11-2017	Pham Thi Minh Phuong
5	Final Inspection: Report 5	26-11-2017	10-12-2017	Pham Thi Minh Phuong
6	Final Inspection: Report 6	11-12-2017	14-12-2017	Pham Thi Minh Phuong

#### 2.4.4 Resource

Specified as in the section [2.5.1 Project Team](#)

#### 2.4.5 Infrastructure

Item	Description	Expected Availability by	Note
<b>Development Environment</b>			
Operating System	Window 8.1 (32 bit, 64 bit)		
Browser	Chrome (40 or above), Firefox (30 or above)		
Development language	Node.js		
<b>Technology</b>			
Development language	Node.js		
Database	MongoDB		
<b>Hardware Requirement</b>			
Hardware Configuration	2GB workspaces on server		
Design	Rational Rose		
<b>Equipment &amp; Tools</b>			
Source Version Control	SmartGit	Definition stage	
Task Tracking	MS Project Professional 2013	Initiation stage	
SRS	Microsoft Office Word 2013, Microsoft Office Excel 2013, Microsoft Office Visio 2013	Initiation stage	

**Table 2-17: Infrastructure****2.4.6 Training Plan**

Training Area	Participants	Duration	Waiver Criteria
<b>Technical</b>			
MVC5	TuanCA, HaNX	1 week	Mandatory
NodeJS, jQuery	TuanCA, HaNX	1 week	Mandatory
Bootstrap	KienNT, PhuongHA	1 week	
<b>Process</b>			
Quality system		3 hours	If already trained
Configuration management		2 hours	If already trained for CC. For others, on-the-job training
Group review		2 hours	If already trained
Defect prevention		2 hours	Mandatory

**Table 2-18: Training Plan****2.4.7 Finance**

Because this project is non-business, it is a Capstone Project at FPT University. So we do not estimate about finance.

**2.5 Project Organization****2.5.1 Project Team**

Role	Responsibility	Full name	Effort (%)	Start date	End date
PM	Have overall responsibility of the project: - Project planning and scheduling - Task assignment and tracking processing - Review documents - Reporting to supervisor	Pham Thi Minh Phuong	50	07-09-2017	25-12-2017

PTL	PTL is responsible for the technical project execution	Nguyen Xuan Ha	50	07-09-2017	25-12-2017
Programmer #1	- Study technique (MVC, Node.js, etc.) - Coding functions and modules of system. - Peer-review source code of others members	Cao Anh Tuan, Nguyen Xuan Ha,	100	07-09-2017	25-12-2017
Programmer #2	- Support coding functions and modules of system.	KienNT	50	07-09-2017	25-12-2017
Programmer #3	- Support coding functions and modules of system.	PhuongHA	100	07-09-2017	30-10-2017
Test Leader	- Create test plan, test case, test report, quality report - Execute test.	Hoang Anh Phuong	100	30-10-2017	25-12-2017
Tester	- Support creating test plan, test case, test report, quality report Execute test.	Pham Thi Minh Phuong	100	30-10-2017	25-12-2017
Design	- Create screen design, prototype - Review design of others member - Review design of others member	Ngo Trung Kien	100	07-09-2017	25-12-2017

**Table 2-19: Project Team description**

### 2.5.2 External Interfaces

FPT University Interfaces

Function	Contact Person (name, position)	Contact address (email, telephone)	Responsibility
Teacher	Nguyen Van Sang	<u>SangNV@fpt.edu.vn</u>	- Review and accept documents during project - Review and accept products of the project.

			- Resolve escalated issues and receive project reports.
Training Department		<a href="mailto:acad.hn@fpt.edu.vn">acad.hn@fpt.edu.vn</a>	Management course of student

## 2.6 Communication & Reporting

Communication Type	Method/Tool	When	Information	Participants/ Responsible
<b>Project Task Tracking</b>				
Task scheduling	MS Project Professional 2013	At the beginning of every stage, and weekly Refinement and rescheduling as necessary		PM
Task assignment	MS Project Professional 2013	Weekly		PTL
Task status reporting	Daily Report	Daily		Project Team members
<b>Project Meeting</b>				
Kick-off Meeting	Face to face	Initiation stage	Project introduction; Project plan review; Risk identification; stakeholders identify.	PM, Project Team Members
Project Progress Review Meetings	Face to face	Weekly and on event	Communicate project status Communicate and resolve any open issue, risks, and changes	PM, Project Team Members

			Discuss any suggested improvement	
Milestone Meetings	Face to face	5 days after the completion of stages: Definition, Solution & Construction	Project objective review, evaluate project performance (quality, schedule, effort), Causal analysis, update project plan for next stage	PM, Project Team Members, QA, Supervisor
Transfer/Sharing of project documentation/information	TortoiseGit	When available	All project documentation and information	PM, Project Team Members, QA
<b>Customer Communication and Reporting:</b>				
Project Report	Agreed FPT Software and FU standard format	5pm Monday, Weekly	Project status report, Issue requiring clarifications, escalation, if any	PM
Project Meetings with supervisor	Face to face	12h30 Wednesday, Weekly	As above	PM
Requirement gathering/clarification	Face to face meeting	During requirement analysis phase	As in Q&A list	PM
<b>Communication with Senior Management</b>				
Review Project Plan & Project schedule	By attend project meeting	Significant changes to WO, PP and Project schedule (scope, objectives Organization, HR, major milestone,		PM

		deliverables )		
Project Progress Review	By email and/or via Operation meeting at Group/Division level	Weekly	Project status report, Issue requiring clarifications, escalation, if any	PM
Project Milestone Review	By email and via project milestone review meeting	End of every stage	Project objective review, evaluate project performance (quality, schedule, effort), Causal analysis, update project plan for next stage	PM
<b>Other Communication and Reporting:</b>				
Raise issue or request service/support of BA groups (IT, Admin, QA, HR, Training, Recruitment, etc)	Call log; email; phone	Upon request	Request content, expected completion date	PM

**Table 2-20: Communication and Reporting Plan**

### 3 SOFTWARE REQUIREMENT

#### 3.1 Introduction

##### 3.1.1 Purpose

This document is created as the introduction for project BSO – our Capstone Project at FPT University. In this document, we provide the describing of general requirement and non-functional requirements. These requirements will assure that the system will correctly and reliably perform its intended functionality. This specification will provide general, as well as specific requirements to be used in the design, testing and validation of the system. It is intended for both stakeholders and the project team of the system.

##### 3.1.2 Definitions and Acronyms

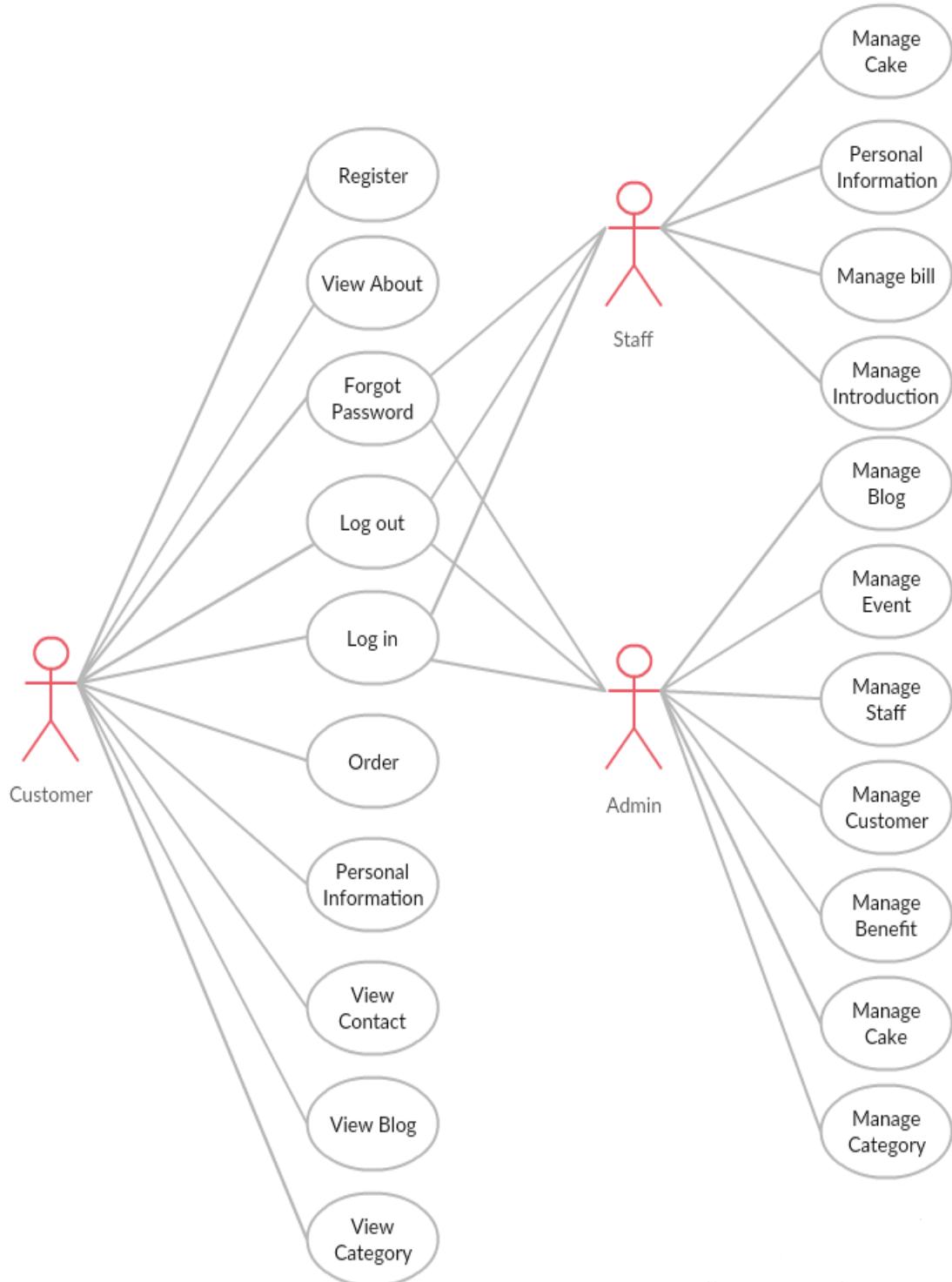
Acronym	Definition	Note
BSO	Bakery Store Online	
SRS	Software Requirements Specifications	

**Table 3-1: Definition and Acronyms**

## 3.2 Software Requirements Specification

### 3.2.1 Overall Description

#### 3.2.1.1 System use case



**Figure 3-1: Use case diagram of BSO system**

### 3.2.1.2 Product Features

UC No.	Group Of Functions	Function
<b>Administrator</b>		
UC001	Common	Admin Login
UC002		Admin Logout
UC003		Admin Forgot Password
UC004		View list staff
UC005		View staff profile
UC006		Activate staff account
UC007		Deactivate staff account
UC008		Search staff account
UC009		Update staff account
UC010		Insert a new staff account
UC011		View list customer
UC012	Manage Customer	Activate customer account

UC013		Deactivate customer account
UC014		View customer profile
UC015		Search customer account
UC016	Manage Benefit	Search benefit
UC017		View benefit
UC018	Manage Cake	Search cake
UC019		Add cake
UC020		Update cake
UC021		Block Cake
UC022		Unblock Cake
UC023		View information cake
UC024	Manage blog	View list blog
UC025		Search blog
UC026		Add blog
UC027		Block blog
UC028		Unblock blog
UC029	Manage Event	Search event
UC030		Add event

UC031		Block event
UC032		Unblock event
UC033	Manage Category	Add Category
UC034		Block Category
UC035		Unblock Category
<b>Staff</b>		
UC036	Common	Log in
UC037		Log out
UC038		Forgot Password
UC039	Manage Bill	Search order
UC040		Confirm Order
UC041		Unconfirm Order
UC042	Manage Cake	View Cake
UC043		Search Cake
UC044	Personal information	Update Information
UC045		Change Password
UC046	Manage Introduction	Create Introduction
UC047		Update Introduction

<b>Customer</b>		
UC048	Common Module	Register
UC049		Login
UC050		Logout
UC051		Forgot Password
UC052		View About
UC053		View Contact
UC054		View Blog
UC055		View Category
UC056	Order	Order Cake
UC057		Check Out
UC058	Personal Information	Edit Profile
UC059		Change Password
UC060		Purchase History

**Table 3-2: List of use case diagram**

### 3.2.1.3 User Characteristics

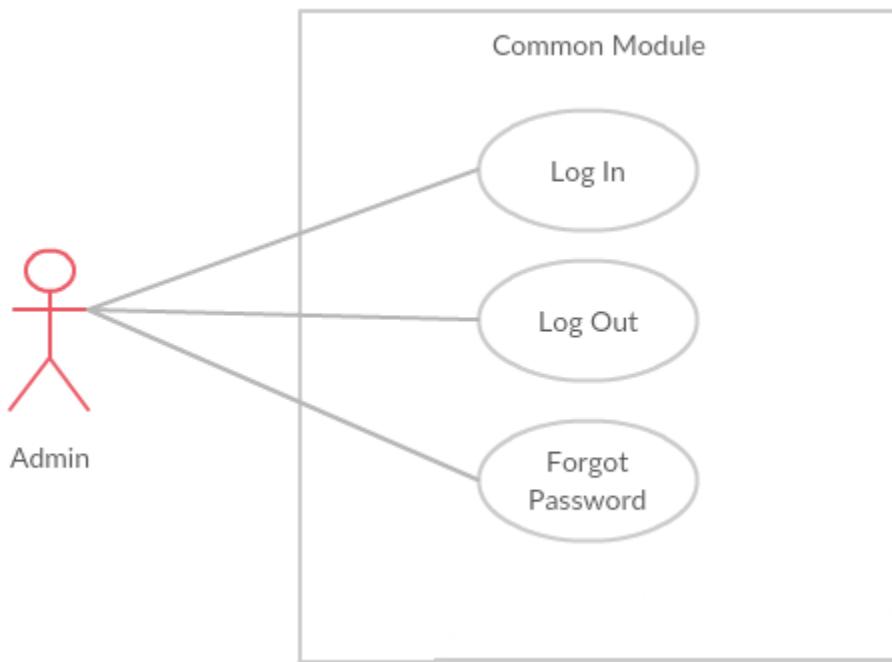
BSO website is the site sell cake online. The website requires users to have the ability to use a browser with an internet connection. User can easily use the functions in website, but if they want to use all function, they must be login. BSO has three objective users:

- Administrator: They can manage all information of BSO.
- Staff: They can manage information about order, contact,...
- Customer: they can use functions of BSO like Login, Logout, Order,...

### 3.2.2 Functional Specification

#### 3.2.2.1 Administrator Group Function

##### 3.2.2.1.1 Common Module



**Figure 3-2: Common use case**

###### 3.2.2.1.1.1 UC001- Admin login

Use Case ID	UC001	Version	1.0
Use Case Name	Admin login		
Created by:	PhuongPTM	Date Created:	28/09/2017
Primary Actor	Administrator(Admin)	Secondary Actor	
Description:	This function allows Admin logins into website		
Preconditions:	1. BSO website is available 2. Admin browsed BSO website 3. Admin are viewing Login page for admin		
Trigger:	Admin clicks on Login button		
Post conditions:	1. Log Admin into system 2. Redirect Admin to Admin home page		
<b>Normal Flow</b>			
Step	Actor	Action	

1	BSO	Display Admin Login page. If Admin choose Remember Me function before, fill all needed information
2	Admin	Type email address and password if Admin does not choose Remember Me function before
3	BSO	Verify: Email address Password
4	BSO	<ul style="list-style-type: none"> <li>– Log Admin into system</li> <li>– Redirect Admin to Admin home page</li> </ul>
<b>Alternative Flows: N/A</b>		
<b>Exceptions:</b>		
<b>EC1</b>	At step 2 in main flow, if Admin entered wrong Email address or Password	
Step	Actor	Action
2.1	BSO	Display Login popup message point out error field
2.2	BSO	Mark error fields
<b>Priority</b>		High
<b>Frequency of Use:</b>		High
<b>Business Rules:</b>		<a href="#">B1</a>
<b>Other Information:</b>		N/A

**Table 3-3: Admin login Use case****3.2.2.1.1.2 UC002- Admin logout**

<b>Use Case ID</b>	UC002	<b>Version</b>	1.0
<b>Use Case Name</b>	Admin logout		
<b>Created by:</b>	PhuongPTM	<b>Date Created:</b>	28/09/2017
<b>Primary Actor</b>	Administrator(Admin)	<b>Secondary Actor</b>	
<b>Description:</b>	This function allows Admin logout website		
<b>Preconditions:</b>	Admin logged website as Administrator		
<b>Trigger:</b>	Admin clicks on Sign out button		
<b>Post conditions:</b>	Admin logged out from BSO website		
<b>Normal Flow</b>			
Step	Actor	Action	

1	Admin	Choose sign out button
2	BSO	Sign out Admin Redirect to Admin Sign in page
<b>Alternative Flows:</b> N/A		
<b>Exceptions:</b> N/A		
<b>Priority</b>	High	
<b>Frequency of Use:</b>	High	
<b>Business Rules:</b>	<a href="#">B1</a>	
<b>Other Information:</b>	N/A	

**Table 3-4: Admin logout Use case****3.2.2.1.1.3 UC003 – Admin Forgot Password**

<b>Use Case ID</b>	UC003	<b>Version</b>	1.0		
<b>Use Case Name</b>	Admin forgot password				
<b>Created by:</b>	PhuongPTM	<b>Date Created:</b>	28/09/2017		
<b>Primary Actor</b>	Administrator(Admin)	<b>Secondary Actor</b>			
<b>Description:</b>	This function allows Admin receiving her/his password to mail				
<b>Preconditions:</b>	1. BSO website is available 2. Admin browsed BSO website 3. Admin are viewing Sign in page for admin				
<b>Trigger:</b>	Click Forgot password in Sign in page				
<b>Post conditions:</b>	Admin get new password on her/his email				
<b>Normal Flow</b>					
<b>Step</b>	<b>Actor</b>	<b>Action</b>			
1	Admin	Click Forgot password in Sign in page			
2	BSO	Display Email text box			
3	Admin	Enter registered email			
4	Admin	Click Forgot Password			
<b>Alternative Flows:</b> N/A					
<b>Exceptions:</b> N/A					
<b>Priority</b>	High				

<b>Frequency of Use:</b>	High
<b>Business Rules:</b>	<a href="#">B1</a>
<b>Other Information:</b>	N/A

Table 3-5: Admin Forgot Password Use case

### 3.2.2.1.2 Manage customer's account

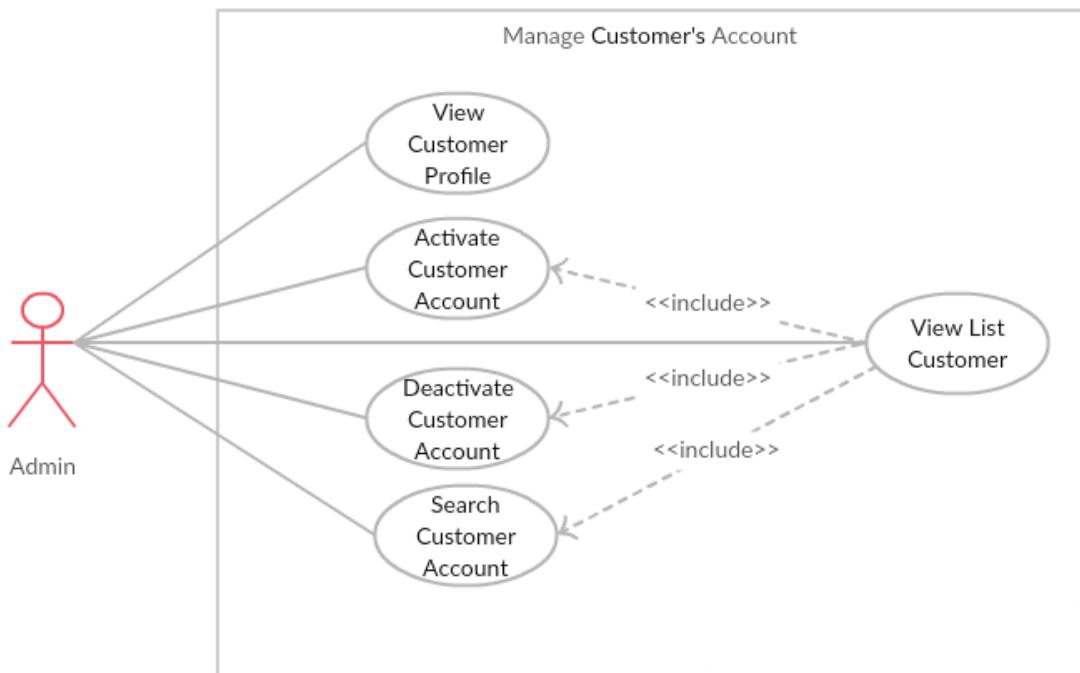


Figure 3-3: Manage customer's use case

#### 3.2.2.1.2.1 UC004- View List Customer

<b>Use Case ID</b>	UC004	<b>Version</b>	1.0
<b>Use Case Name</b>	View list customer		
<b>Created by:</b>	PhuongPTM	<b>Date Created:</b>	28/09/2017
<b>Primary Actor</b>	Administrator(Admin)	<b>Secondary Actor</b>	
<b>Description:</b>	This function allows Administrator to view list customer		
<b>Preconditions:</b>	1. BSO website is available 2. Admin browsed BSO website 3. Admin are viewing Login page for admin		
<b>Trigger:</b>	Admin click on Manage Account		
<b>Post conditions:</b>	Customer List is displayed		
<b>Normal Flow</b>			
<b>Step</b>	<b>Actor</b>	<b>Action</b>	

1	Admin	<ul style="list-style-type: none"> <li>– Click on “Choose role you want” drop down list</li> <li>– Click on Customer</li> </ul>
2	BSO	Display list of customer
<b>Alternative Flows:</b> N/A		
<b>Exceptions:</b> N/A		
<b>Priority</b>	High	
<b>Frequency of Use:</b>	High	
<b>Business Rules:</b>	<a href="#">B1</a>	
<b>Other Information:</b>	N/A	

**Table 3-6: View List Customer****3.2.2.1.2.2 UC005- View Customer Profile**

<b>Use Case ID</b>	UC005	<b>Version</b>	1.0
<b>Use Case Name</b>	View customer profile		
<b>Created by:</b>	PhuongPTM	<b>Date Created:</b>	28/09/2017
<b>Primary Actor</b>	Administrator(Admin)	<b>Secondary Actor</b>	
<b>Description:</b>	This function allows Administrator to view profile of a customer		
<b>Preconditions:</b>	1. BSO website is available 2. Admin browsed BSO website 3. Admin are viewing Login page for admin		
<b>Trigger:</b>	Admin click on Manage Account		
<b>Post conditions:</b>	Profile of this customer is displayed		
<b>Normal Flow</b>			
<b>Step</b>	<b>Actor</b>	<b>Action</b>	
1	Admin	<ul style="list-style-type: none"> <li>– Click on “Choose role you want” drop down list</li> <li>– Click on Customer</li> </ul>	
2	BSO	Display profile of this customer	
<b>Alternative Flows:</b> N/A			
<b>Exceptions:</b> N/A			
<b>Priority</b>	High		
<b>Frequency of Use:</b>	High		

<b>Business Rules:</b>	<a href="#"><u>B1</u></a>
<b>Other Information:</b>	N/A

**Table 3-7: View Customer Profile Use case**

## 3.2.2.1.2.3 UC006- Activate Customer Account

<b>Use Case ID</b>	UC006	<b>Version</b>	1.0		
<b>Use Case Name</b>	Activate customer's account				
<b>Created by:</b>	PhuongPTM	<b>Date Created:</b>	28/09/2017		
<b>Primary Actor</b>	Administrator(Admin)	<b>Secondary Actor</b>			
<b>Description:</b>	This function allows Administrator to active a customer's account				
<b>Preconditions:</b>	<ul style="list-style-type: none"> <li>– Admin is viewing Customers List</li> <li>– Status of this customer is Lock</li> </ul>				
<b>Trigger:</b>	Admin click on Unlock button				
<b>Post conditions:</b>	Customer's account is activated				
<b>Normal Flow</b>					
<b>Step</b>	<b>Actor</b>	<b>Action</b>			
1	Admin	Click on Unlock button at Status column of the customer whom want to activate			
2	BSO	<ul style="list-style-type: none"> <li>– Change displayed status from Lock to Unlock</li> <li>– Change status of this customer's account to Unlock on database</li> </ul>			
<b>Alternative Flows: N/A</b>					
<b>Exceptions: N/A</b>					
<b>Priority</b>	High				
<b>Frequency of Use:</b>	High				
<b>Business Rules:</b>	<a href="#">B1</a>				
<b>Other Information:</b>	N/A				

**Table 3-8: Activate Customer Account Use case**

## 3.2.2.1.2.4 UC007 – Deactivate Customer Account

<b>Use Case ID</b>	UC007	<b>Version</b>	1.0
<b>Use Case Name</b>	Deactivate customer's account		
<b>Created by:</b>	PhuongPTM	<b>Date Created:</b>	28/09/2017
<b>Primary Actor</b>	Administrator(Admin)	<b>Secondary Actor</b>	
<b>Description:</b>	This function allows Administrator to deactivate a customer's account		
<b>Preconditions:</b>	<ul style="list-style-type: none"> <li>– Admin is viewing Customers List</li> </ul>		

– Status of this customer is Unlock		
<b>Trigger:</b>	Admin click on Lock button	
<b>Post conditions:</b>	Customer's account is deactivated	
<b>Normal Flow</b>		
Step	Actor	Action
1	Admin	Click on Lock button at Status column of the customer whom want to deactivate
2	BSO	<ul style="list-style-type: none"> <li>– Change displayed status from Unlock to Lock</li> <li>– Change status of this customer's account to Lock on database</li> </ul>
<b>Alternative Flows: N/A</b>		
<b>Exceptions: N/A</b>		
<b>Priority</b>	High	
<b>Frequency of Use:</b>	High	
<b>Business Rules:</b>	<a href="#">B1</a>	
<b>Other Information:</b>	N/A	

**Table 3-9: Deactivate Customer Account Use case****3.2.2.1.2.5 UC008 – Search Customer Account**

<b>Use Case ID</b>	UC008	<b>Version</b>	1.0
<b>Use Case Name</b>	Search customer's account		
<b>Created by:</b>	PhuongPTM	<b>Date Created:</b>	28/09/2017
<b>Primary Actor</b>	Administrator(Admin)	<b>Secondary Actor</b>	
<b>Description:</b>	This function allows Administrator to search a customer's account		
<b>Preconditions:</b>	Admin is viewing Customers List		
<b>Trigger:</b>	Admin types phone number of customer at Search textbox		
<b>Post conditions:</b>	Display list of searched customer		
<b>Normal Flow</b>			
Step	Actor	Action	
1	Admin	Type phone number of customer whom want to search	
2	BSO	Display list of customer who have the phone number is typed	
<b>Alternative Flows: N/A</b>			

Exceptions: N/A	
<b>Priority</b>	High
<b>Frequency of Use:</b>	High
<b>Business Rules:</b>	<a href="#">B1</a>
<b>Other Information:</b>	N/A

**Table 3-10:Search Customer Account Use case****3.2.2.1.3 Manage Staff's Account****3.2.2.1.3.1 UC009 – View List Staff**

<b>Use Case ID</b>	UC009	<b>Version</b>	1.0		
<b>Use Case Name</b>	View list staff				
<b>Created by:</b>	PhuongPTM	<b>Date Created:</b>	28/09/2017		
<b>Primary Actor</b>	Administrator(Admin)	<b>Secondary Actor</b>			
<b>Description:</b>	This function allows Administrator to view list staff				
<b>Preconditions:</b>	1. BSO website is available 2. Admin browsed BSO website 3. Admin are viewing Login page for admin				
<b>Trigger:</b>	Admin click on Manage Account				
<b>Post conditions:</b>	Staff List is displayed				
<b>Normal Flow</b>					
<b>Step</b>	<b>Actor</b>	<b>Action</b>			
1	Admin	<ul style="list-style-type: none"> <li>– Click on “Choose role you want” drop down list</li> <li>– Click on Staff</li> </ul>			
2	BSO	Display list of staff			
<b>Alternative Flows: N/A</b>					
<b>Exceptions: N/A</b>					
<b>Priority</b>	High				
<b>Frequency of Use:</b>	High				
<b>Business Rules:</b>	<a href="#">B1</a>				
<b>Other Information:</b>	N/A				

**Table 3-11:View List Staff Use case****3.2.2.1.3.2 UC010 – View Staff Profile**

<b>Use Case ID</b>	UC010	<b>Version</b>	1.0		
<b>Use Case Name</b>	View staff profile				
<b>Created by:</b>	PhuongPTM	<b>Date Created:</b>	28/09/2017		
<b>Primary Actor</b>	Administrator(Admin)	<b>Secondary Actor</b>			
<b>Description:</b>	This function allows Administrator to view profile of a staff				
<b>Preconditions:</b>	Admin is viewing Staff List				
<b>Trigger:</b>	Admin click on Manage Account				
<b>Post conditions:</b>	Profile of this staff is displayed				
<b>Normal Flow</b>					
Step	Actor	<b>Action</b>			
1	Admin	Click on Manage Account button at action column of the staff whom want to see profile			
2	BSO	Display profile of this staff			
<b>Alternative Flows: N/A</b>					
<b>Exceptions: N/A</b>					
<b>Priority</b>	High				
<b>Frequency of Use:</b>	High				
<b>Business Rules:</b>	<u>B1</u>				
<b>Other Information:</b>	N/A				

**Table 3-12:View Staff Profile Use case****3.2.2.1.3.3 UC011 – Activate Staff Account**

<b>Use Case ID</b>	UC011	<b>Version</b>	1.0
<b>Use Case Name</b>	Activate staff's account		
<b>Created by:</b>	PhuongPTM	<b>Date Created:</b>	28/09/2017
<b>Primary Actor</b>	Administrator(Admin)	<b>Secondary Actor</b>	
<b>Description:</b>	This function allows Administrator to active a staff's account		
<b>Preconditions:</b>	<ul style="list-style-type: none"> <li>– Admin is viewing Staff List</li> <li>– Status of this staff is Lock</li> </ul>		

<b>Trigger:</b>	Admin click on Unlock button	
<b>Post conditions:</b>	Staff's account is activated	
<b>Normal Flow</b>		
Step	Actor	Action
1	Admin	Click on Unlock button at Status column of the staff whom want to activate
2	BSO	<ul style="list-style-type: none"> <li>– Change displayed status from Lock to Unlock</li> <li>– Change status of this staff's account to Unlock on database</li> </ul>
<b>Alternative Flows:</b> N/A		
<b>Exceptions:</b> N/A		
<b>Priority</b>	High	
<b>Frequency of Use:</b>	High	
<b>Business Rules:</b>	<a href="#">B1</a>	
<b>Other Information:</b>	N/A	

**Table 3-13: Activate Staff Account Use case**

## 3.2.2.1.3.4 UC012 – Deactivate Staff Account

<b>Use Case ID</b>	UC012	<b>Version</b>	1.0
<b>Use Case Name</b>	Deactivate staff's account		
<b>Created by:</b>	PhuongPTM	<b>Date Created:</b>	28/09/2017
<b>Primary Actor</b>	Administrator(Admin)	<b>Secondary Actor</b>	
<b>Description:</b>	This function allows Administrator to deactivate a staff's account		
<b>Preconditions:</b>	<ul style="list-style-type: none"> <li>– Admin is viewing Staff List</li> <li>– Status of this staff is Unlock</li> </ul>		
<b>Trigger:</b>	Admin click on Lock button		
<b>Post conditions:</b>	Staff's account is deactivated		
<b>Normal Flow</b>			
Step	Actor	Action	
1	Admin	Click on Lock button at Status column of the staff whom want to deactivate	
2	BSO	<ul style="list-style-type: none"> <li>– Change displayed status from Unlock to Lock</li> </ul>	

	– Change status of this staff's account to Lock on database
<b>Alternative Flows:</b> N/A	
<b>Exceptions:</b> N/A	
<b>Priority</b>	High
<b>Frequency of Use:</b>	High
<b>Business Rules:</b>	<a href="#">B1</a>
<b>Other Information:</b>	N/A

**Table 3-14: Deactivate Staff Account Use case**

## 3.2.2.1.3.5 UC013 – Search Staff Account

<b>Use Case ID</b>	UC013	<b>Version</b>	1.0		
<b>Use Case Name</b>	Search staff's account				
<b>Created by:</b>	PhuongPTM	<b>Date Created:</b>	28/09/2017		
<b>Primary Actor</b>	Administrator(Admin)	<b>Secondary Actor</b>			
<b>Description:</b>	This function allows Administrator to search a staff's account				
<b>Preconditions:</b>	Admin is viewing Staff List				
<b>Trigger:</b>	Admin types name of staff at Search textbox				
<b>Post conditions:</b>	Display list of searched staff				
<b>Normal Flow</b>					
Step	Actor	<b>Action</b>			
1	Admin	Type name of staff whom want to search			
2	BSO	Display list of staff who have the name is typed			
<b>Alternative Flows:</b> N/A					
<b>Exceptions:</b> N/A					
<b>Priority</b>	High				
<b>Frequency of Use:</b>	High				
<b>Business Rules:</b>	<a href="#">B1</a>				
<b>Other Information:</b>	N/A				

**Table 3-15: Search Staff Account Use case**

## 3.2.2.1.3.6 UC014 – Update Staff Account

<b>Use Case ID</b>	UC014	<b>Version</b>	1.0		
<b>Use Case Name</b>	Update staff's account				
<b>Created by:</b>	PhuongPTM	<b>Date Created:</b>	28/09/2017		
<b>Primary Actor</b>	Administrator/Admin)	<b>Secondary Actor</b>			
<b>Description:</b>	This function allows Administrator to update a staff's account				
<b>Preconditions:</b>	Admin is viewing Staff List				
<b>Trigger:</b>	Admin click on Update button				
<b>Post conditions:</b>	Update staff's account page is displayed				
<b>Normal Flow</b>					
<b>Step</b>	<b>Actor</b>	<b>Action</b>			
1	Admin	Choose the staff who want to change information			
2	Admin	Click on Update button			
3	Admin	Update information			
4	Admin	Click on Save button			
5	BSO	Save change			
<b>Alternative Flows: N/A</b>					
<b>Exceptions:</b>					
<b>EC1</b>	At step 3 in the main flows, if user miss fill 1 or more information				
<b>Step</b>	<b>Actor</b>	<b>Action</b>			
3.1	BSO	Display Dialog to remind user			
<b>Priority</b>	High				
<b>Frequency of Use:</b>	High				
<b>Business Rules:</b>	<a href="#">B1</a>				
<b>Other Information:</b>	N/A				

Table 3-16: Update Staff Account Use case

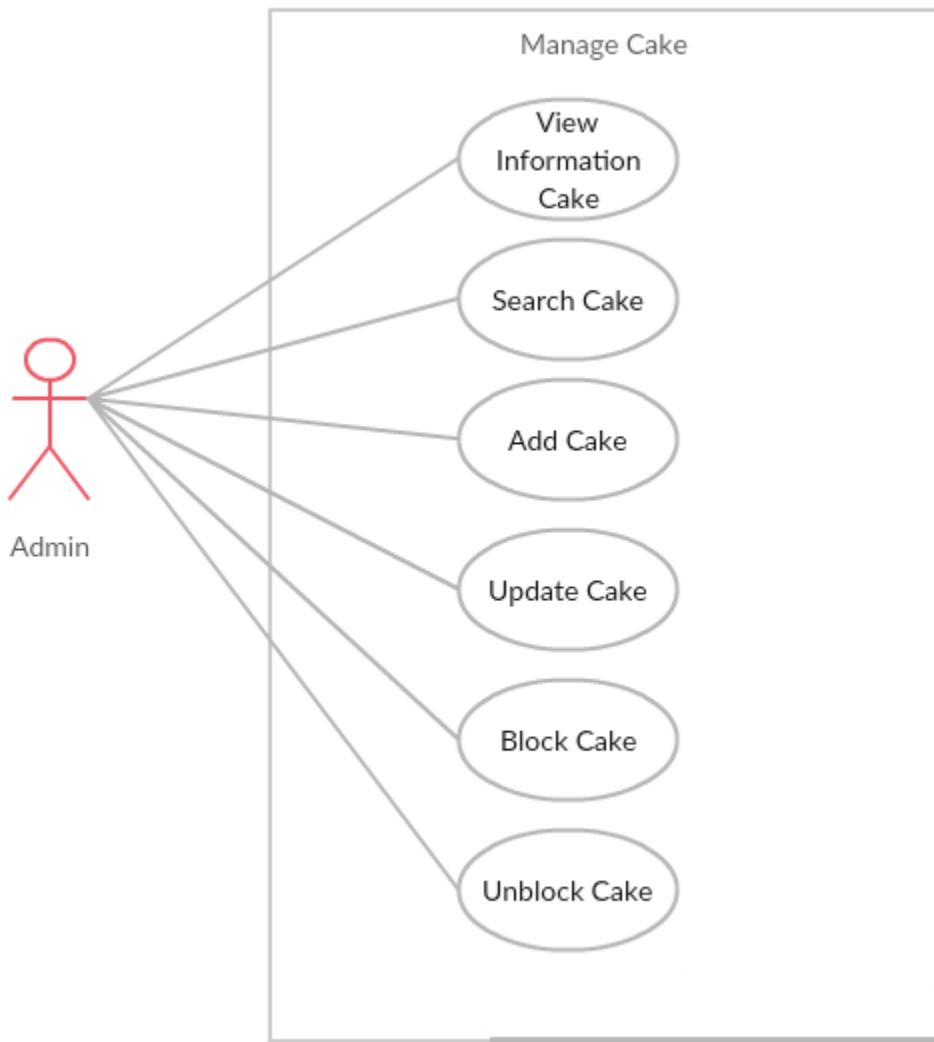
## 3.2.2.1.3.7 UC015 – Insert A New Staff Account

<b>Use Case ID</b>	UC015	<b>Version</b>	1.0
--------------------	-------	----------------	-----

<b>Use Case Name</b>	Insert a new staff's account				
<b>Created by:</b>	PhuongPTM	<b>Date Created:</b>	28/09/2017		
<b>Primary Actor</b>	Administrator(Admin)	<b>Secondary Actor</b>			
<b>Description:</b>	This function allows Administrator to insert a new staff's account				
<b>Preconditions:</b>	1. BSO website is available 2. Admin browsed BSO website 3. Admin are viewing Login page for admin				
<b>Trigger:</b>	Admin click on Insert Staff				
<b>Post conditions:</b>	Insert staff page is displayed				
<b>Normal Flow</b>					
Step	Actor	<b>Action</b>			
1	Admin	Click on Insert Staff			
2	BSO	Insert staff page is displayed			
3	Admin	<ul style="list-style-type: none"> <li>- Type Staff name</li> <li>- Type Password</li> <li>- Type Address</li> <li>- Type Phone number</li> <li>- Type Email</li> <li>- Choose image</li> <li>- Click Add button</li> </ul>			
<b>Alternative Flows: N/A</b>					
<b>Exceptions:</b>					
<b>EC1</b>	At step 3 in the main flows, if user miss fill 1 or more information				
Step	Actor	<b>Action</b>			
<b>3.1</b>	BSO	Display Dialog to remind user			
<b>Priority</b>	High				
<b>Frequency of Use:</b>	High				
<b>Business Rules:</b>	<u>B1</u>				
<b>Other Information:</b>	N/A				

**Table 3-17: Insert A New Staff Account Use case**

### 3.2.2.1.4 Manage Cake



**Figure 3-4: Manage Cake Use case**

#### 3.2.2.1.4.1 UC016 – View Information Cake

<b>Use Case ID</b>	UC016	<b>Version</b>	1.0
<b>Use Case Name</b>	View information cake		
<b>Created by:</b>	PhuongPTM	<b>Date Created:</b>	28/09/2017
<b>Primary Actor</b>	Administrator(Admin)	<b>Secondary Actor</b>	
<b>Description:</b>	This function allows Administrator to view list cake		
<b>Preconditions:</b>	1. BSO website is available 2. Admin browsed BSO website 3. Admin are viewing Login page for admin		
<b>Trigger:</b>	Admin click on Manage Cake		
<b>Post conditions:</b>	Cake List is displayed		

<b>Normal Flow</b>		
<b>Step</b>	<b>Actor</b>	<b>Action</b>
1	Admin	Click on Manage Cake
2	BSO	Display list of cake
<b>Alternative Flows: N/A</b>		
<b>Exceptions: N/A</b>		
<b>Priority</b>	High	
<b>Frequency of Use:</b>	High	
<b>Business Rules:</b>	<a href="#">B1</a>	
<b>Other Information:</b>	N/A	

**Table 3-18: View Information Cake Use case****3.2.2.1.4.2 UC017 – Search Cake**

<b>Use Case ID</b>	UC017	<b>Version</b>	1.0
<b>Use Case Name</b>	Search cake		
<b>Created by:</b>	PhuongPTM	<b>Date Created:</b>	28/09/2017
<b>Primary Actor</b>	Administrator(Admin)	<b>Secondary Actor</b>	
<b>Description:</b>	This function allows Administrator to search cake		
<b>Preconditions:</b>	Admin is viewing Cake List		
<b>Trigger:</b>	Admin types name of cake at Search textbox		
<b>Post conditions:</b>	Display list of searched cake		
<b>Normal Flow</b>			
<b>Step</b>	<b>Actor</b>	<b>Action</b>	
1	Admin	Click on Manage Cake	
2	BSO	List cake is displayed	
3	Admin	Type name of cake what admin want to search	
4	BSO	Display list of cake what have the name is typed	
<b>Alternative Flows: N/A</b>			
<b>Exceptions: N/A</b>			

<b>Priority</b>	High
<b>Frequency of Use:</b>	High
<b>Business Rules:</b>	<a href="#">B1</a>
<b>Other Information:</b>	N/A

**Table 3-19: Search Cake Use case****3.2.2.1.4.3 UC018 – Add Cake**

<b>Use Case ID</b>	UC018	<b>Version</b>	1.0		
<b>Use Case Name</b>	Add cake				
<b>Created by:</b>	PhuongPTM	<b>Date Created:</b>	28/09/2017		
<b>Primary Actor</b>	Administrator(Admin)	<b>Secondary Actor</b>			
<b>Description:</b>	This function allows Administrator to add a new cake				
<b>Preconditions:</b>	1. BSO website is available 2. Admin browsed BSO website 3. Admin are viewing Login page for admin				
<b>Trigger:</b>	Admin click on Insert Cake				
<b>Post conditions:</b>	Insert cake page is displayed				
<b>Normal Flow</b>					
<b>Step</b>	<b>Actor</b>	<b>Action</b>			
1	Admin	Click on Insert Cake			
2	BSO	Insert cake page is displayed			
3	Admin	<ul style="list-style-type: none"> <li>- Choose Cake type</li> <li>- Type Cake name</li> <li>- Type Material</li> <li>- Type Description</li> <li>- Type Size</li> <li>- Type Price</li> <li>- Type Quantity</li> <li>- Choose image</li> <li>- Click Add button</li> </ul>			
<b>Alternative Flows: N/A</b>					
<b>Exceptions:</b>					

<b>EC1</b>	At step 3 in the main flows, if user miss fill 1 or more information	
<b>Step</b>	<b>Actor</b>	<b>Action</b>
3.1	BSO	Display Dialog to remind user
<b>Priority</b>	High	
<b>Frequency of Use:</b>	High	
<b>Business Rules:</b>	<u>B1</u>	
<b>Other Information:</b>	N/A	

**Table 3-20: Add Cake Use case****3.2.2.1.4.4 UC019 – Update Cake**

<b>Use Case ID</b>	UC019	<b>Version</b>	1.0		
<b>Use Case Name</b>	Update cake				
<b>Created by:</b>	PhuongPTM	<b>Date Created:</b>	28/09/2017		
<b>Primary Actor</b>	Administrator/Admin)	<b>Secondary Actor</b>			
<b>Description:</b>	This function allows Administrator to update a cake				
<b>Preconditions:</b>	Admin is viewing Cake List				
<b>Trigger:</b>	Admin click on Update button				
<b>Post conditions:</b>	Update cake page is displayed				
<b>Normal Flow</b>					
<b>Step</b>	<b>Actor</b>	<b>Action</b>			
1	Admin	Choose the cake what admin want to change information			
2	Admin	Click on Update button			
3	Admin	Update information			
4	Admin	Click on Save button			
5	BSO	Save change			
<b>Alternative Flows: N/A</b>					
<b>Exceptions:</b>					
<b>EC1</b>	At step 3 in the main flows, if user miss fill 1 or more information				
<b>Step</b>	<b>Actor</b>	<b>Action</b>			

<b>3.1</b>	BSO	Display Dialog to remind user
<b> </b>		
<b>Priority</b>		High
<b>Frequency of Use:</b>		High
<b>Business Rules:</b>		<a href="#">B1</a>
<b>Other Information:</b>		N/A

**Table 3-21: Update Cake Use case****3.2.2.1.4.5 UC020 – Block Cake**

<b>Use Case ID</b>	UC020	<b>Version</b>	1.0		
<b>Use Case Name</b>	Block cake				
<b>Created by:</b>	PhuongPTM	<b>Date Created:</b>	28/09/2017		
<b>Primary Actor</b>	Administrator(Admin)	<b>Secondary Actor</b>			
<b>Description:</b>	This function allows Administrator to block cake when the cake was not sold				
<b>Preconditions:</b>	<ul style="list-style-type: none"> <li>– Admin is viewing Cake List</li> <li>– Status of this cake is Selling</li> </ul>				
<b>Trigger:</b>	Admin click on Not Sell button				
<b>Post conditions:</b>	Cake is block				
<b>Normal Flow</b>					
<b>Step</b>	<b>Actor</b>	<b>Action</b>			
1	Admin	Click on Not Sell button at Status column of the cake what want to block			
2	BSO	<ul style="list-style-type: none"> <li>– Change displayed status from Selling to Not Sell</li> <li>– Change status of this cake to Not Sell on database</li> </ul>			
<b>Alternative Flows: N/A</b>					
<b>Exceptions: N/A</b>					
<b>Priority</b>		High			
<b>Frequency of Use:</b>		High			
<b>Business Rules:</b>		<a href="#">B1</a>			
<b>Other Information:</b>		N/A			

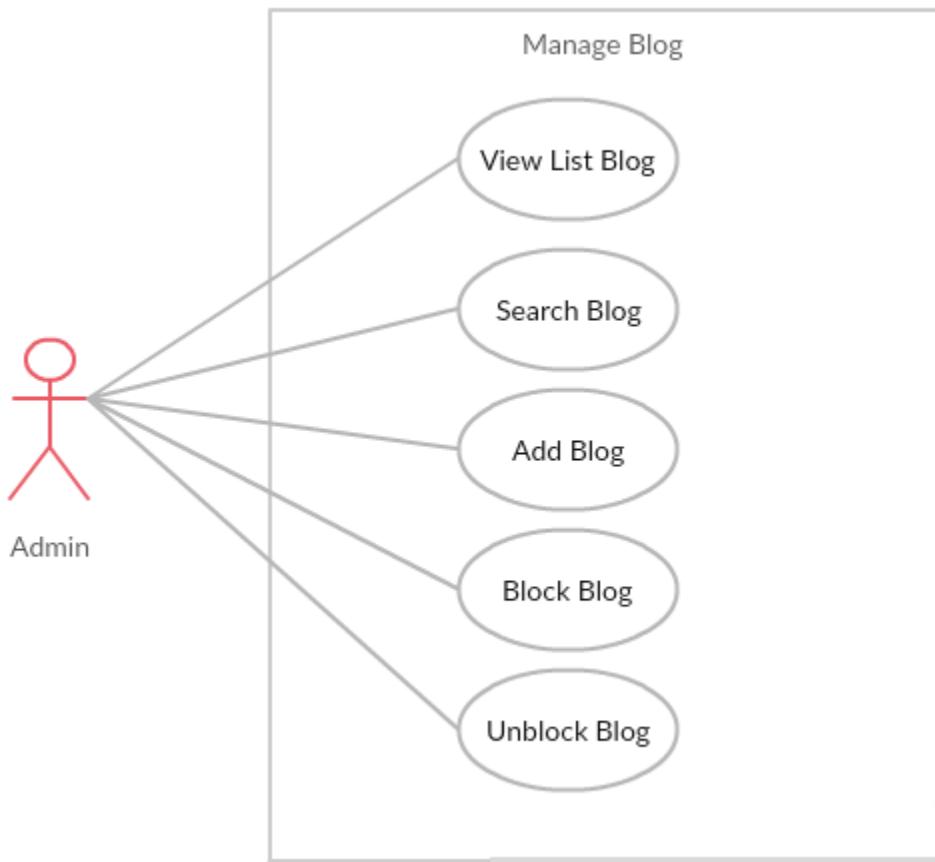
**Table 3-22: Block Cake Use case**

## 3.2.2.1.4.6 UC021 – Unblock Cake

<b>Use Case ID</b>	UC021	<b>Version</b>	1.0		
<b>Use Case Name</b>	Unblock cake				
<b>Created by:</b>	PhuongPTM	<b>Date Created:</b>	28/09/2017		
<b>Primary Actor</b>	Administrator(Admin)	<b>Secondary Actor</b>			
<b>Description:</b>	This function allows Administrator to unblock cake when the cake is selling				
<b>Preconditions:</b>	<ul style="list-style-type: none"> <li>– Admin is viewing Cake List</li> <li>– Status of this cake is Not Sell</li> </ul>				
<b>Trigger:</b>	Admin click on Sell Again button				
<b>Post conditions:</b>	Cake is unblock				
<b>Normal Flow</b>					
<b>Step</b>	<b>Actor</b>	<b>Action</b>			
1	Admin	Click on Sell Again button at Status column of the cake what want to unblock			
2	BSO	<ul style="list-style-type: none"> <li>– Change displayed status from Not Sell to Selling</li> <li>– Change status of this cake to Selling on database</li> </ul>			
<b>Alternative Flows: N/A</b>					
<b>Exceptions: N/A</b>					
<b>Priority</b>	High				
<b>Frequency of Use:</b>	High				
<b>Business Rules:</b>	<a href="#">B1</a>				
<b>Other Information:</b>	N/A				

**Table 3-23: Unblock Cake Use case**

### 3.2.2.1.5 Manage Blog



**Figure 3-5: Manage blog use case**

#### 3.2.2.1.5.1 UC022 – View List Blog

<b>Use Case ID</b>	UC022	<b>Version</b>	1.0
<b>Use Case Name</b>	View list blog		
<b>Created by:</b>	PhuongPTM	<b>Date Created:</b>	28/09/2017
<b>Primary Actor</b>	Administrator(Admin)	<b>Secondary Actor</b>	
<b>Description:</b>	This function allows Administrator to view list blog		
<b>Preconditions:</b>	1. BSO website is available 2. Admin browsed BSO website 3. Admin are viewing Login page for admin		
<b>Trigger:</b>	Admin click on Manage Blog		
<b>Post conditions:</b>	Blog List is displayed		
<b>Normal Flow</b>			
<b>Step</b>	<b>Actor</b>	<b>Action</b>	
1	Admin	Click on Manage Blog	

2	BSO	Display list of blog
<b>Alternative Flows:</b> N/A		
<b>Exceptions:</b> N/A		
<b>Priority</b>	High	
<b>Frequency of Use:</b>	High	
<b>Business Rules:</b>	<a href="#">B1</a>	
<b>Other Information:</b>	N/A	

Table 3-24: View List Blog Use case

3.2.2.1.5.2 [UC023 – Add Blog](#)

<b>Use Case ID</b>	UC023	<b>Version</b>	1.0		
<b>Use Case Name</b>	Add blog				
<b>Created by:</b>	PhuongPTM	<b>Date Created:</b>	28/09/2017		
<b>Primary Actor</b>	Administrator(Admin)	<b>Secondary Actor</b>			
<b>Description:</b>	This function allows Administrator to add a new blog				
<b>Preconditions:</b>	1. BSO website is available 2. Admin browsed BSO website 3. Admin are viewing Login page for admin				
<b>Trigger:</b>	Admin click on Insert Blog				
<b>Post conditions:</b>	Insert blog page is displayed				
<b>Normal Flow</b>					
<b>Step</b>	<b>Actor</b>	<b>Action</b>			
1	Admin	Click on Insert Blog			
2	BSO	Insert blog page is displayed			
3	Admin	<ul style="list-style-type: none"> <li>- Type title</li> <li>- Type Content</li> <li>- Choose image</li> <li>- Click Post button</li> </ul>			
<b>Alternative Flows:</b> N/A					
<b>Exceptions:</b>					
<b>EC1</b>	At step 3 in the main flows, if user miss fill 1 or more information				
<b>Step</b>	<b>Actor</b>	<b>Action</b>			
3.1	BSO	Display Dialog to remind user			

<b>Priority</b>	High
<b>Frequency of Use:</b>	High
<b>Business Rules:</b>	<a href="#">B1</a>
<b>Other Information:</b>	N/A

**Table 3-25: Add Blog Use case****3.2.2.1.5.3 UC024 – Block Blog**

<b>Use Case ID</b>	UC024	<b>Version</b>	1.0		
<b>Use Case Name</b>	Block blog				
<b>Created by:</b>	PhuongPTM	<b>Date Created:</b>	28/09/2017		
<b>Primary Actor</b>	Administrator(Admin)	<b>Secondary Actor</b>			
<b>Description:</b>	This function allows Administrator to block blog				
<b>Preconditions:</b>	<ul style="list-style-type: none"> <li>– Admin is viewing Blog List</li> <li>– Status of this blog is Unlock</li> </ul>				
<b>Trigger:</b>	Admin click on Lock button				
<b>Post conditions:</b>	Blog is block				
<b>Normal Flow</b>					
<b>Step</b>	<b>Actor</b>	<b>Action</b>			
1	Admin	Click on Lock button at Status column of the blog what want to block			
2	BSO	<ul style="list-style-type: none"> <li>– Change displayed status from Unlock to Lock</li> <li>– Change status of this blog to Lock on database</li> </ul>			
<b>Alternative Flows: N/A</b>					
<b>Exceptions: N/A</b>					
<b>Priority</b>	High				
<b>Frequency of Use:</b>	High				
<b>Business Rules:</b>	<a href="#">B1</a>				
<b>Other Information:</b>	N/A				

**Table 3-26:Block Blog Use case**

## 3.2.2.1.5.4 UC025 – Unblock Blog

<b>Use Case ID</b>	UC025	<b>Version</b>	1.0		
<b>Use Case Name</b>	Unblock blog				
<b>Created by:</b>	PhuongPTM	<b>Date Created:</b>	28/09/2017		
<b>Primary Actor</b>	Administrator(Admin)	<b>Secondary Actor</b>			
<b>Description:</b>	This function allows Administrator to unblock blog				
<b>Preconditions:</b>	<ul style="list-style-type: none"> <li>– Admin is viewing Blog List</li> <li>– Status of this blog is Lock</li> </ul>				
<b>Trigger:</b>	Admin click on Unlock button				
<b>Post conditions:</b>	Blog is unblock				
<b>Normal Flow</b>					
Step	Actor	<b>Action</b>			
1	Admin	Click on Unlock button at Status column of the blog what want to unblock			
2	BSO	<ul style="list-style-type: none"> <li>– Change displayed status from Lock to Unlock</li> <li>– Change status of this blog to Unlock on database</li> </ul>			
<b>Alternative Flows: N/A</b>					
<b>Exceptions: N/A</b>					
<b>Priority</b>	High				
<b>Frequency of Use:</b>	High				
<b>Business Rules:</b>	<a href="#">B1</a>				
<b>Other Information:</b>	N/A				

**Table 3-27:Unblock Blog Use case**

## 3.2.2.1.5.5 UC026 – Search Blog

<b>Use Case ID</b>	UC026	<b>Version</b>	1.0
<b>Use Case Name</b>	Search blog		
<b>Created by:</b>	PhuongPTM	<b>Date Created:</b>	28/09/2017
<b>Primary Actor</b>	Administrator(Admin)	<b>Secondary Actor</b>	

<b>Description:</b>	This function allows Administrator to search blog	
<b>Preconditions:</b>	Admin is viewing Blog List	
<b>Trigger:</b>	Admin types name of blog at Search textbox	
<b>Post conditions:</b>	Display list of searched blog	
<b>Normal Flow</b>		
Step	Actor	Action
1	Admin	Click on Manage blog
2	BSO	List blog is displayed
3	Admin	Type name of blog what admin want to search
4	BSO	Display list of blog what have the name is typed
<b>Alternative Flows: N/A</b>		
<b>Exceptions: N/A</b>		
<b>Priority</b>	High	
<b>Frequency of Use:</b>	High	
<b>Business Rules:</b>	<a href="#">B1</a>	
<b>Other Information:</b>	N/A	

**Table 3-28:Search Blog Use case****3.2.2.1.6 Manage Event****3.2.2.1.6.1 UC027 – Search Event**

<b>Use Case ID</b>	UC027	<b>Version</b>	1.0
<b>Use Case Name</b>	Search event		
<b>Created by:</b>	PhuongPTM	<b>Date Created:</b>	28/09/2017
<b>Primary Actor</b>	Administrator(Admin)	<b>Secondary Actor</b>	
<b>Description:</b>	This function allows Administrator to search event		
<b>Preconditions:</b>	Admin is viewing Event List		
<b>Trigger:</b>	Admin types name of event at Search textbox		
<b>Post conditions:</b>	Display list of searched event		
<b>Normal Flow</b>			

Step	Actor	Action
1	Admin	Click on Manage event
2	BSO	List event is displayed
3	Admin	Type name of event what admin want to search
4	BSO	Display list of event what have the name is typed
<b>Alternative Flows:</b> N/A		
<b>Exceptions:</b> N/A		
<b>Priority</b>		High
<b>Frequency of Use:</b>		High
<b>Business Rules:</b>		<a href="#">B1</a>
<b>Other Information:</b>		N/A

**Table 3-29:Search Event Use case****3.2.2.1.6.2 UC028 – Add Event**

<b>Use Case ID</b>	UC028	<b>Version</b>	1.0
<b>Use Case Name</b>	Add event		
<b>Created by:</b>	PhuongPTM	<b>Date Created:</b>	28/09/2017
<b>Primary Actor</b>	Administrator(Admin)	<b>Secondary Actor</b>	
<b>Description:</b>	This function allows Administrator to add a new event		
<b>Preconditions:</b>	1. BS website is available 2. Admin browsed BS website 3. Admin are viewing Login page for admin		
<b>Trigger:</b>	Admin click on Insert Event		
<b>Post conditions:</b>	Insert event page is displayed		
<b>Normal Flow</b>			
Step	Actor	<b>Action</b>	
1	Admin	Click on Insert Event	
2	BSO	Insert event page is displayed	
3	Admin	- Type title - Type Content	

		- Choose image - Click Add button
<b>Alternative Flows:</b> N/A		
<b>Exceptions:</b>		
<b>EC1</b>	At step 3 in the main flows, if user miss fill 1 or more information	
Step	Actor	Action
<b>3.1</b>	BSO	Display Dialog to remind user
<b>Priority</b>	High	
<b>Frequency of Use:</b>	High	
<b>Business Rules:</b>	<a href="#">B1</a>	
<b>Other Information:</b>	N/A	

**Table 3-30: Add Event Use case****3.2.2.1.6.3 UC029 – Block Event**

<b>Use Case ID</b>	UC029	<b>Version</b>	1.0		
<b>Use Case Name</b>	Block event				
<b>Created by:</b>	PhuongPTM	<b>Date Created:</b>	28/09/2017		
<b>Primary Actor</b>	Administrator(Admin)	<b>Secondary Actor</b>			
<b>Description:</b>	This function allows Administrator to block event				
<b>Preconditions:</b>	<ul style="list-style-type: none"> <li>– Admin is viewing Event List</li> <li>– Status of this event is Unlock</li> </ul>				
<b>Trigger:</b>	Admin click on Lock button				
<b>Post conditions:</b>	Event is block				
<b>Normal Flow</b>					
Step	Actor	<b>Action</b>			
1	Admin	Click on Lock button at Status column of the event what want to block			
2	BSO	<ul style="list-style-type: none"> <li>– Change displayed status from Unlock to Lock</li> <li>– Change status of this event to Lock on database</li> </ul>			
<b>Alternative Flows:</b> N/A					
<b>Exceptions:</b> N/A					

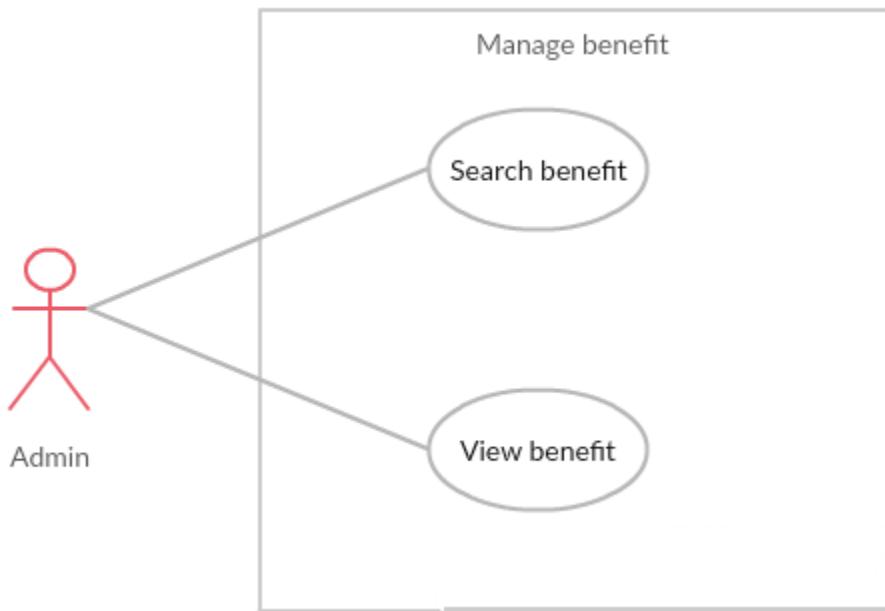
<b>Priority</b>	High
<b>Frequency of Use:</b>	High
<b>Business Rules:</b>	<a href="#">B1</a>
<b>Other Information:</b>	N/A

**Table 3-31:Block Event Use case****3.2.2.1.6.4 UC030 – Unblock Event**

<b>Use Case ID</b>	UC030	<b>Version</b>	1.0		
<b>Use Case Name</b>	Unblock event				
<b>Created by:</b>	PhuongPTM	<b>Date Created:</b>	28/09/2017		
<b>Primary Actor</b>	Administrator(Admin)	<b>Secondary Actor</b>			
<b>Description:</b>	This function allows Administrator to unblock event				
<b>Preconditions:</b>	<ul style="list-style-type: none"> <li>– Admin is viewing Event List</li> <li>– Status of this event is Lock</li> </ul>				
<b>Trigger:</b>	Admin click on Unlock button				
<b>Post conditions:</b>	Event is unblock				
<b>Normal Flow</b>					
<b>Step</b>	<b>Actor</b>	<b>Action</b>			
1	Admin	Click on Unlock button at Status column of the event what want to unblock			
2	BSO	<ul style="list-style-type: none"> <li>– Change displayed status from Lock to Unlock</li> <li>– Change status of this event to Unlock on database</li> </ul>			
<b>Alternative Flows: N/A</b>					
<b>Exceptions: N/A</b>					
<b>Priority</b>	High				
<b>Frequency of Use:</b>	High				
<b>Business Rules:</b>	<a href="#">B1</a>				
<b>Other Information:</b>	N/A				

**Table 3-32:Unblock Event Use case**

### 3.2.2.1.7 Manage Benefit



**Figure 3-6: Manage Benefit Use case**

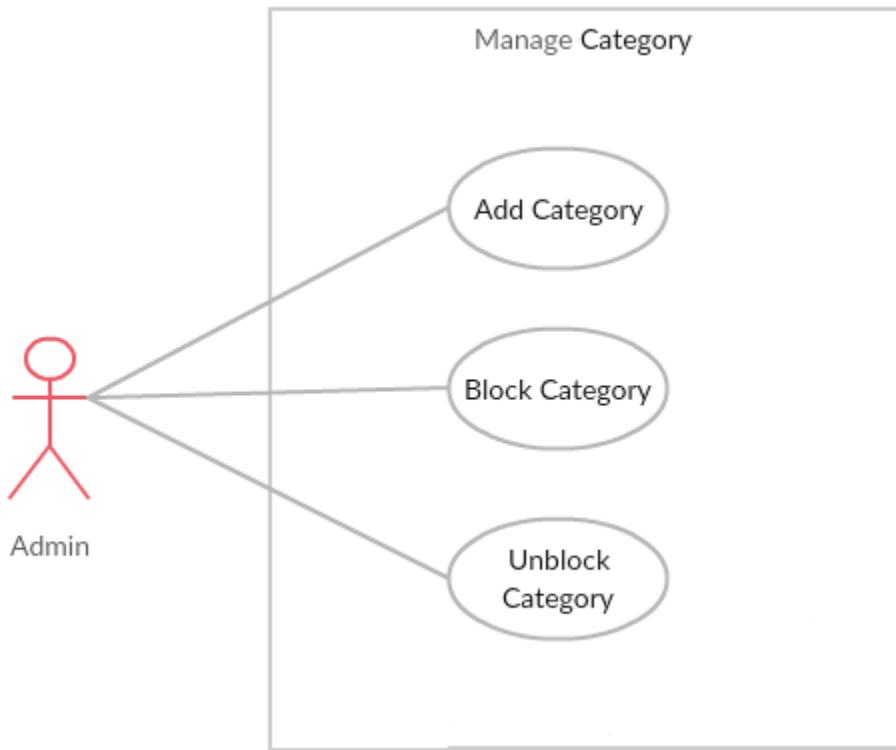
#### 3.2.2.1.7.1 UC031 – Search Benefit

<b>Use Case ID</b>	UC031	<b>Version</b>	1.0
<b>Use Case Name</b>	Search blog		
<b>Created by:</b>	PhuongPTM	<b>Date Created:</b>	28/09/2017
<b>Primary Actor</b>	Administrator(Admin)	<b>Secondary Actor</b>	
<b>Description:</b>	This function allows Administrator to search benefit		
<b>Preconditions:</b>	1. BS website is available 2. Admin browsed BS website 3. Admin are viewing Login page for admin		
<b>Trigger:</b>	- Admin click Revenue - Choose date		
<b>Post conditions:</b>	Display list of searched bill		
<b>Normal Flow</b>			
<b>Step</b>	<b>Actor</b>	<b>Action</b>	
1	Admin	Click on Revenue	
2	BSO	List bill is displayed	
3	Admin	Choose date	
4	BSO	Display list of bill	

<b>Alternative Flows:</b> N/A	
<b>Exceptions:</b> N/A	
<b>Priority</b>	High
<b>Frequency of Use:</b>	High
<b>Business Rules:</b>	<a href="#">B1</a>
<b>Other Information:</b>	N/A

**Table 3-33:Search Benefit Use case****3.2.2.1.7.2 UC032 - View Benefit**

<b>Use Case ID</b>	UC032	<b>Version</b>	1.0		
<b>Use Case Name</b>	View benefit				
<b>Created by:</b>	PhuongPTM	<b>Date Created:</b>	28/09/2017		
<b>Primary Actor</b>	Administrator(Admin)	<b>Secondary Actor</b>			
<b>Description:</b>	This function allows Administrator to check benefit of BS				
<b>Preconditions:</b>	1. BS website is available 2. Admin browsed BS website 3. Admin are viewing Login page for admin				
<b>Trigger:</b>	Admin click on Manage Bill				
<b>Post conditions:</b>	Bill list is displayed				
<b>Normal Flow</b>					
<b>Step</b>	<b>Actor</b>	<b>Action</b>			
1	Admin	Click on Manage bill			
2	BSO	Display list of bill			
<b>Alternative Flows:</b> N/A					
<b>Exceptions:</b> N/A					
<b>Priority</b>	High				
<b>Frequency of Use:</b>	High				
<b>Business Rules:</b>	<a href="#">B1</a>				
<b>Other Information:</b>	N/A				

**Table 3-34: View Benefit Use case****3.2.2.1.8 Manage Categories****Figure 3-7: Manage Categories Use case****3.2.2.1.8.1 UC033 - Add Category**

<b>Use Case ID</b>	UC033	<b>Version</b>	1.0
<b>Use Case Name</b>	Add category		
<b>Created by:</b>	PhuongPTM	<b>Date Created:</b>	28/09/2017
<b>Primary Actor</b>	Administrator(Admin)	<b>Secondary Actor</b>	
<b>Description:</b>	This function allows Administrator to add a new category		
<b>Preconditions:</b>	1. BS website is available 2. Admin browsed BS website 3. Admin are viewing Login page for admin		
<b>Trigger:</b>	Admin click on Insert Category		
<b>Post conditions:</b>	Insert category page is displayed		
<b>Normal Flow</b>			
<b>Step</b>	<b>Actor</b>	<b>Action</b>	
1	Admin	Click on Insert Category	

2	BS	Insert category page is displayed
3	Admin	<ul style="list-style-type: none"> <li>- Type category name</li> <li>- Click Post button</li> </ul>
<b>Alternative Flows: N/A</b>		
<b>Exceptions:</b>		
<b>EC1</b>	At step 3 in the main flows, if user miss fill 1 or more information	
Step	Actor	Action
<b>3.1</b>	BSO	Display Dialog to remind user
<b>Priority</b>	High	
<b>Frequency of Use:</b>	High	
<b>Business Rules:</b>	<a href="#">B1</a>	
<b>Other Information:</b>	N/A	

**Table 3-35: Add Category Use case****3.2.2.1.8.2 UC034 – Block Category**

<b>Use Case ID</b>	UC034	<b>Version</b>	1.0
<b>Use Case Name</b>	Block category		
<b>Created by:</b>	PhuongPTM	<b>Date Created:</b>	28/09/2017
<b>Primary Actor</b>	Administrator(Admin)	<b>Secondary Actor</b>	
<b>Description:</b>	This function allows Administrator to block category		
<b>Preconditions:</b>	<ul style="list-style-type: none"> <li>– Admin is viewing Category List</li> <li>– Status of this category is Unlock</li> </ul>		
<b>Trigger:</b>	Admin click on Lock button		
<b>Post conditions:</b>	Category is block		
<b>Normal Flow</b>			
Step	Actor	<b>Action</b>	
1	Admin	Click on Lock button at Status column of the category what want to block	
2	BSO	<ul style="list-style-type: none"> <li>– Change displayed status from Unlock to Lock</li> <li>– Change status of this category to Lock on database</li> </ul>	

<b>Alternative Flows:</b> N/A	
<b>Exceptions:</b> N/A	
<b>Priority</b>	High
<b>Frequency of Use:</b>	High
<b>Business Rules:</b>	<a href="#">B1</a>
<b>Other Information:</b>	N/A

**Table 3-36: Block Category Use case**

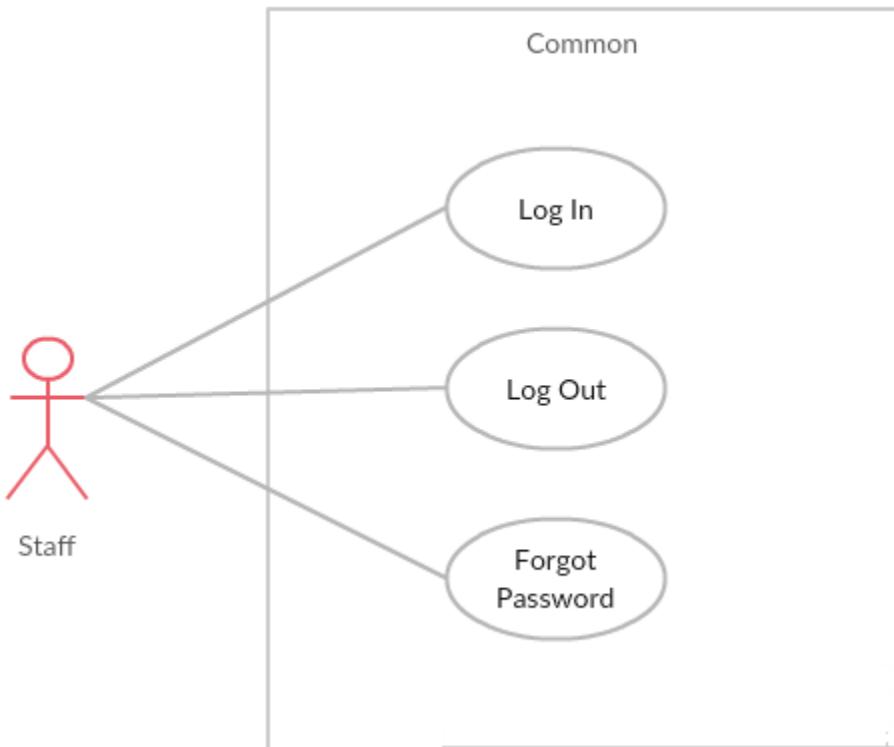
## 3.2.2.1.8.3 UC035 – Unblock Category

<b>Use Case ID</b>	UC035	<b>Version</b>	1.0		
<b>Use Case Name</b>	Unblock category				
<b>Created by:</b>	PhuongPTM	<b>Date Created:</b>	28/09/2017		
<b>Primary Actor</b>	Administrator(Admin)	<b>Secondary Actor</b>			
<b>Description:</b>	This function allows Administrator to unblock category				
<b>Preconditions:</b>	<ul style="list-style-type: none"> <li>– Admin is viewing Category List</li> <li>– Status of this category is Lock</li> </ul>				
<b>Trigger:</b>	Admin click on Unlock button				
<b>Post conditions:</b>	Category is unblock				
<b>Normal Flow</b>					
<b>Step</b>	<b>Actor</b>	<b>Action</b>			
1	Admin	Click on Unlock button at Status column of the category what want to unblock			
2	BSO	<ul style="list-style-type: none"> <li>– Change displayed status from Lock to Unlock</li> <li>– Change status of this category to Unlock on database</li> </ul>			
<b>Alternative Flows:</b> N/A					
<b>Exceptions:</b> N/A					
<b>Priority</b>	High				
<b>Frequency of Use:</b>	High				
<b>Business Rules:</b>	<a href="#">B1</a>				
<b>Other Information:</b>	N/A				

**Table 3-37: Unblock Category Use case**

### 3.2.2.2 Staff Group Function

#### 3.2.2.2.1 Common Module

**Figure 3-8: Register Use case**

##### 3.2.2.2.1.1 UC036 – Staff login

<b>Use Case ID</b>	UC036	<b>Version</b>	1.0
<b>Use Case Name</b>	Staff login		
<b>Created by:</b>	PhuongPTM	<b>Date Created:</b>	28/09/2017
<b>Primary Actor</b>	Staff	<b>Secondary Actor</b>	
<b>Description:</b>	This function allows Staff logins into website		
<b>Preconditions:</b>	1. BSO website is available 2. Staff browsed BSO website 3. Staff are viewing Login page for staff		
<b>Trigger:</b>	Staff clicks on Login button		
<b>Post conditions:</b>	1. Log Staff into system		

		2. Redirect Staff to Staff home page
<b>Normal Flow</b>		
Step	Actor	Action
1	BSO	Display Staff Login page. If Staff choose Remember Me function before, fill all needed information
2	Staff	Type email address and password if Staff does not choose Remember Me function before
3	BSO	Verify: Email address Password
4	BSO	– Log Staff into system – Redirect Staff to Staff home page
<b>Alternative Flows: N/A</b>		
<b>Exceptions:</b>		
<b>EC1</b>	At step 2 in main flow, if Staff entered wrong Email address or Password	
Step	Actor	Action
2.1	BSO	Display Login popup message point out error field
2.2	BSO	Mark error fields
<b>Priority</b>		High
<b>Frequency of Use:</b>		High
<b>Business Rules:</b>		<u>B1</u>
<b>Other Information:</b>		N/A

Table 3-38: Staff login Use case

## 3.2.2.2.1.2 UC037 – Staff logout

<b>Use Case ID</b>	UC037	<b>Version</b>	1.0
<b>Use Case Name</b>	Staff logout		
<b>Created by:</b>	PhuongPTM	<b>Date Created:</b>	28/09/2017
<b>Primary Actor</b>	Staff	<b>Secondary Actor</b>	
<b>Description:</b>	This function allows Staff logout website		
<b>Preconditions:</b>	Staff logged website		
<b>Trigger:</b>	Staff clicks on Sign out button		
<b>Post conditions:</b>	Staff logged out from BS website		

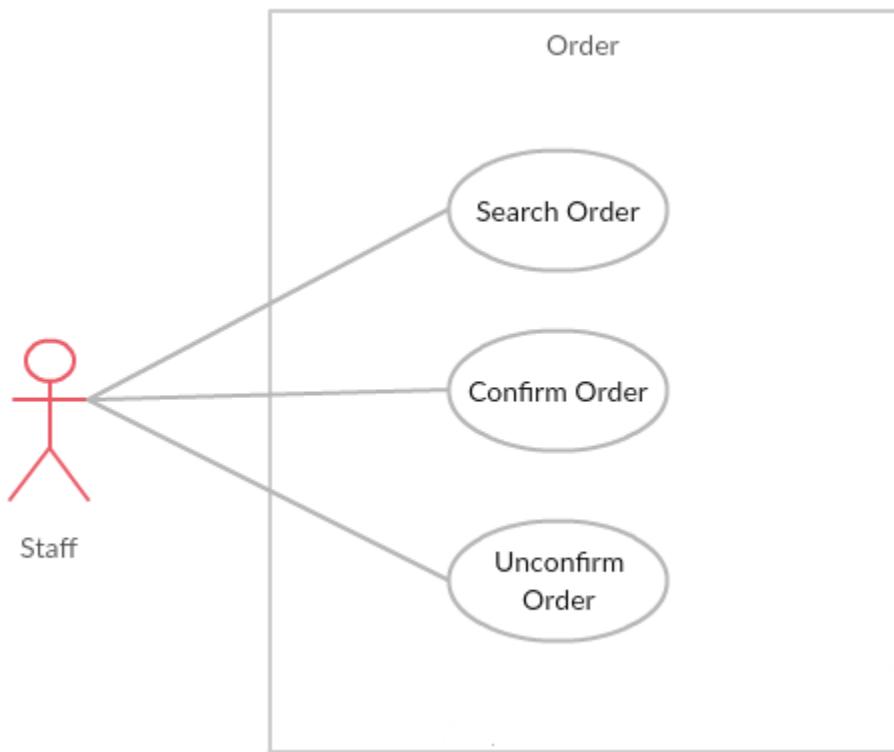
Normal Flow		
Step	Actor	Action
1	Staff	Choose sign out button
2	BSO	Sign out Staff Redirect to Staff Sign in page
Alternative Flows: N/A		
Exceptions: N/A		
Priority	High	
Frequency of Use:	High	
Business Rules:	<a href="#">B1</a>	
Other Information:	N/A	

Table 3-39:Staff logout Use case

3.2.2.2.1.3 [UC038 – Forgot Password](#)

Use Case ID	UC038	Version	1.0		
Use Case Name	Staff forgot password				
Created by:	PhuongPTM	Date Created:	28/09/2017		
Primary Actor	Staff	Secondary Actor			
Description:	This function allows Staff receiving her/his password to mail				
Preconditions:	1. BSO website is available 2. Staff browsed BSO website 3. Staff are viewing Sign in page for staff				
Trigger:	Click Forgot password in Sign in page				
Post conditions:	Staff get new password on her/his email				
Normal Flow					
Step	Actor	Action			
1	Staff	Click Forgot password in Sign in page			
2	BSO	Display Email text box			
3	Staff	Enter registered email			
4	Staff	Click Forgot Password			
Alternative Flows: N/A					
Exceptions: N/A					

<b>Priority</b>	High
<b>Frequency of Use:</b>	High
<b>Business Rules:</b>	<a href="#">B1</a>
<b>Other Information:</b>	N/A

**Table 3-40:Forgot Password Use case****3.2.2.2.2 Manage Order****Figure 3-9: Manage Order Use case****3.2.2.2.2.1 UC039 – Search Order**

<b>Use Case ID</b>	UC039	<b>Version</b>	1.0
<b>Use Case Name</b>	Search order		
<b>Created by:</b>	PhuongPTM	<b>Date Created:</b>	28/09/2017
<b>Primary Actor</b>	Staff	<b>Secondary Actor</b>	
<b>Description:</b>	This function allows Staff to search order by date		
<b>Preconditions:</b>	Staff is viewing Manage Bills		
<b>Trigger:</b>	Staff choose date		
<b>Post conditions:</b>	Display list of searched order		

<b>Normal Flow</b>		
<b>Step</b>	<b>Actor</b>	<b>Action</b>
1	Staff	Click on Manage Bills
2	BSO	List order is displayed
3	Staff	Staff choose date
4	BSO	Display list of order what have the date is choose
<b>Alternative Flows:</b> N/A		
<b>Exceptions:</b> N/A		
<b>Priority</b>	High	
<b>Frequency of Use:</b>	High	
<b>Business Rules:</b>	<a href="#">B1</a>	
<b>Other Information:</b>	N/A	

**Table 3-41:Search Order Use case**

## 3.2.2.2.2.2 UC040 – Confirm Order

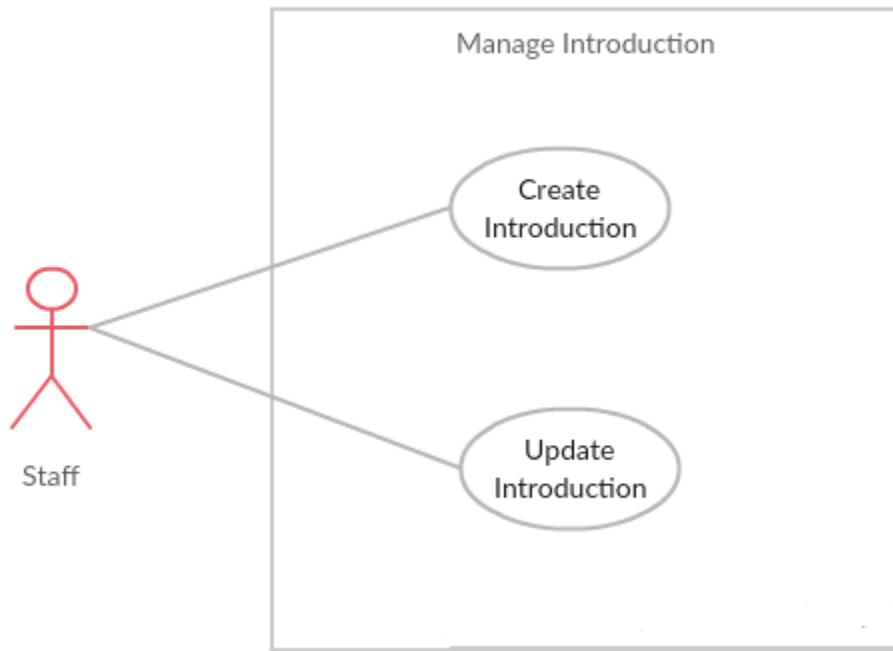
<b>Use Case ID</b>	UC040	<b>Version</b>	1.0
<b>Use Case Name</b>	Confirm order		
<b>Created by:</b>	PhuongPTM	<b>Date Created:</b>	28/09/2017
<b>Primary Actor</b>	Staff	<b>Secondary Actor</b>	
<b>Description:</b>	This function allows Staff to confirm order when order was completed		
<b>Preconditions:</b>	<ul style="list-style-type: none"> <li>– Staff is viewing Manage Bills</li> <li>– Status of this order is Uncofirm</li> </ul>		
<b>Trigger:</b>	Staff click Confirm button		
<b>Post conditions:</b>	Display order is confirm		
<b>Normal Flow</b>			
<b>Step</b>	<b>Actor</b>	<b>Action</b>	
1	Staff	Click on Confirm button at Status column of the order what want to confirm	

2	BSO	<ul style="list-style-type: none"> <li>– Change displayed status from Unconfirm to Confirm</li> <li>– Change status of this cake to Confirm on database</li> </ul>
<b>Alternative Flows:</b> N/A		
<b>Exceptions:</b> N/A		
<b>Priority</b>	High	
<b>Frequency of Use:</b>	High	
<b>Business Rules:</b>	<a href="#">B1</a>	
<b>Other Information:</b>	N/A	

**Table 3-42: Confirm Order Use case****3.2.2.2.3 UC041 – Unconfirm Order**

<b>Use Case ID</b>	UC041	<b>Version</b>	1.0		
<b>Use Case Name</b>	Unconfirm order				
<b>Created by:</b>	PhuongPTM	<b>Date Created:</b>	28/09/2017		
<b>Primary Actor</b>	Staff	<b>Secondary Actor</b>			
<b>Description:</b>	This function allows Staff to unconfirm order when order was not completed				
<b>Preconditions:</b>	– Staff is viewing Manage Bills				
<b>Trigger:</b>	Staff click Unconfirm button				
<b>Post conditions:</b>	Display order is unconfirm				
<b>Normal Flow</b>					
<b>Step</b>	<b>Actor</b>	<b>Action</b>			
1	Staff	Click on Unconfirm button at Status column of the order what want to unconfirm			
2	BSO	Change status of this cake to Unconfirm on database			
<b>Alternative Flows:</b> N/A					
<b>Exceptions:</b> N/A					
<b>Priority</b>	High				
<b>Frequency of Use:</b>	High				
<b>Business Rules:</b>	<a href="#">B1</a>				

<b>Other Information:</b>	N/A
---------------------------	-----

**Table 3-43:Unconfirm Order Use case****3.2.2.2.3 Manage Introduction****Figure 3-10: Manage Introduction Use case****3.2.2.2.3.1 UC042 – Create Introduction**

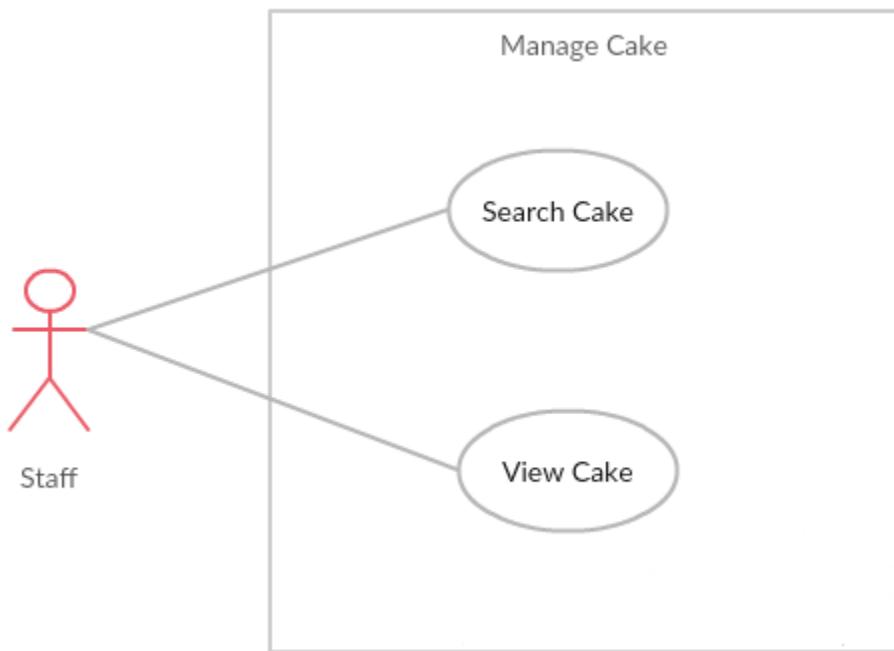
<b>Use Case ID</b>	UC042	<b>Version</b>	1.0
<b>Use Case Name</b>	Create introduction		
<b>Created by:</b>	PhuongPTM	<b>Date Created:</b>	28/09/2017
<b>Primary Actor</b>	Staff	<b>Secondary Actor</b>	
<b>Description:</b>	This function allows Staff to add information about BS website		
<b>Preconditions:</b>	1. BSO website is available 2. Staff browsed BSO website 3. Staff are viewing Sign in page for staff		
<b>Trigger:</b>	Staff click on Create Introduction		
<b>Post conditions:</b>	Create introduction page is displayed		
<b>Normal Flow</b>			
<b>Step</b>	<b>Actor</b>	<b>Action</b>	
1	Staff	Click on Create Introduction	

2	BSO	Create Introduction page is displayed
3	Staff	<ul style="list-style-type: none"> <li>- Type Address</li> <li>- Type Phone number</li> <li>- Choose image</li> <li>- Click Save button</li> </ul>
<b>Alternative Flows:</b> N/A		
<b>Exceptions:</b>		
<b>EC1</b>	At step 3 in the main flows, if user miss fill 1 or more information	
Step	Actor	Action
<b>3.1</b>	BSO	Display Dialog to remind user
<b>Priority</b>	High	
<b>Frequency of Use:</b>	High	
<b>Business Rules:</b>	<a href="#">B1</a>	
<b>Other Information:</b>	N/A	

**Table 3-44: Create Introduction Use case****3.2.2.2.3.2 UC043 – Update Introduction**

<b>Use Case ID</b>	UC043	<b>Version</b>	1.0
<b>Use Case Name</b>	Create introduction		
<b>Created by:</b>	PhuongPTM	<b>Date Created:</b>	28/09/2017
<b>Primary Actor</b>	Staff	<b>Secondary Actor</b>	
<b>Description:</b>	This function allows Staff to update information about BS website		
<b>Preconditions:</b>	1. BSO website is available 2. Staff browsed BSO website 3. Staff are viewing Sign in page for staff		
<b>Trigger:</b>	Staff click on Update Introduction		
<b>Post conditions:</b>	Update introduction page is displayed		
<b>Normal Flow</b>			
Step	Actor	Action	
1	Staff	Click on Update Introduction	

2	Staff	Update information
3	Staff	Click on Save button
4	BSO	Save change
<b>Alternative Flows: N/A</b>		
<b>Exceptions:</b>		
<b>EC1</b>	At step 3 in the main flows, if user miss fill 1 or more information	
Step	Actor	Action
<b>3.1</b>	BSO	Display Dialog to remind user
<b>Priority</b>	High	
<b>Frequency of Use:</b>	High	
<b>Business Rules:</b>	<u>B1</u>	
<b>Other Information:</b>	N/A	

**Table 3-45:Update Introduction Use case****3.2.2.2.4 Manage Cake****Figure 3-11: Manage Cake Use case**

## 3.2.2.2.4.1 UC044 – View Cake

<b>Use Case ID</b>	UC044	<b>Version</b>	1.0		
<b>Use Case Name</b>	View cake				
<b>Created by:</b>	PhuongPTM	<b>Date Created:</b>	28/09/2017		
<b>Primary Actor</b>	Staff	<b>Secondary Actor</b>			
<b>Description:</b>	This function allows Staff to check information about cake				
<b>Preconditions:</b>	1. BSO website is available 2. Staff browsed BSO website 3. Staff are viewing Sign in page for staff				
<b>Trigger:</b>	Staff click on Manage Cake				
<b>Post conditions:</b>	Cake list is displayed				
<b>Normal Flow</b>					
Step	Actor	<b>Action</b>			
1	Staff	Click on Manage Cake			
2	BSO	Display list of cake			
<b>Alternative Flows: N/A</b>					
<b>Exceptions: N/A</b>					
<b>Priority</b>	High				
<b>Frequency of Use:</b>	High				
<b>Business Rules:</b>	<a href="#">B1</a>				
<b>Other Information:</b>	N/A				

**Table 3-46:– View Cake Use case**

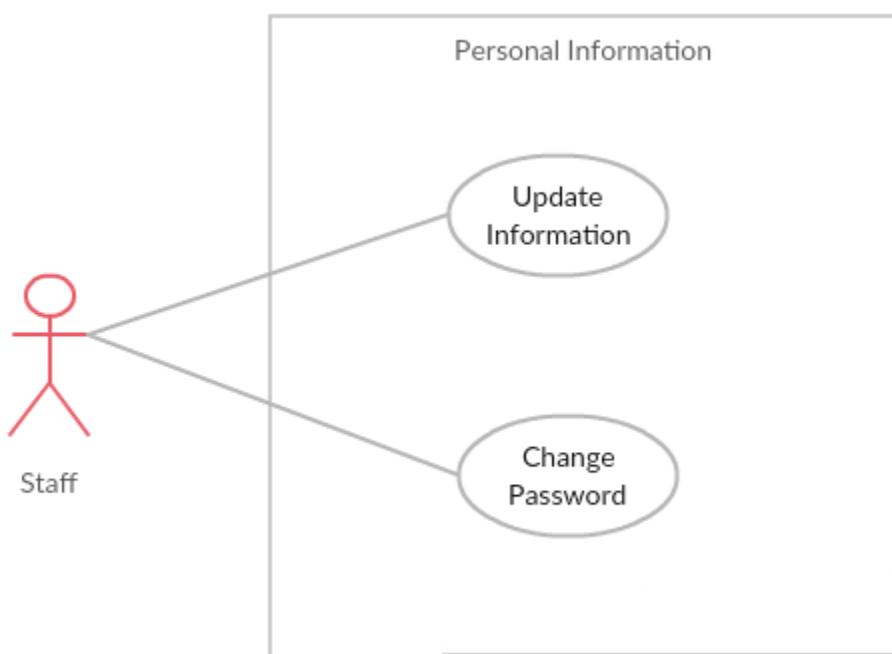
## 3.2.2.2.4.2 UC045 – Search Cake

<b>Use Case ID</b>	UC045	<b>Version</b>	1.0
<b>Use Case Name</b>	Search cake		
<b>Created by:</b>	PhuongPTM	<b>Date Created:</b>	28/09/2017
<b>Primary Actor</b>	Staff	<b>Secondary Actor</b>	
<b>Description:</b>	This function allows Staff to search cake		
<b>Preconditions:</b>	Staff is viewing Cake List		

<b>Trigger:</b>	Staff types name of cake at Search textbox	
<b>Post conditions:</b>	Display list of searched cake	
<b>Normal Flow</b>		
Step	Actor	Action
1	Staff	Click on Manage Cake
2	BSO	List cake is displayed
3	Staff	Type name of cake what admin want to search
4	BSO	Display list of cake what have the name is typed
<b>Alternative Flows:</b> N/A		
<b>Exceptions:</b> N/A		
<b>Priority</b>	High	
<b>Frequency of Use:</b>	High	
<b>Business Rules:</b>	<a href="#">B1</a>	
<b>Other Information:</b>	N/A	

Table 3-47: Search Cake Use case

### 3.2.2.2.5 Personal Information



**Figure 3-12: Personal Information Use case****3.2.2.2.5.1 UC046 – Update Information**

<b>Use Case ID</b>	UC046	<b>Version</b>	1.0		
<b>Use Case Name</b>	Update information				
<b>Created by:</b>	PhuongPTM	<b>Date Created:</b>	28/09/2017		
<b>Primary Actor</b>	Staff	<b>Secondary Actor</b>			
<b>Description:</b>	This function allows Staff to update profile				
<b>Preconditions:</b>	1. BSO website is available 2. Staff browsed BSO website 3. Staff are viewing Sign in page for staff				
<b>Trigger:</b>	Staff click on Update Information				
<b>Post conditions:</b>	Update information page is displayed				
<b>Normal Flow</b>					
<b>Step</b>	<b>Actor</b>	<b>Action</b>			
1	Staff	Click on Update Information			
2	Staff	Update information			
3	Staff	Click on Save button			
4	BSO	Save change			
<b>Alternative Flows: N/A</b>					
<b>Exceptions:</b>					
<b>EC1</b>	At step 3 in the main flows, if user miss fill 1 or more information				
<b>Step</b>	<b>Actor</b>	<b>Action</b>			
3.1	BSO	Display Dialog to remind user			
<b>Priority</b>	High				
<b>Frequency of Use:</b>	High				
<b>Business Rules:</b>	<u>B1</u>				
<b>Other Information:</b>	N/A				

**Table 3-48:Update Information**

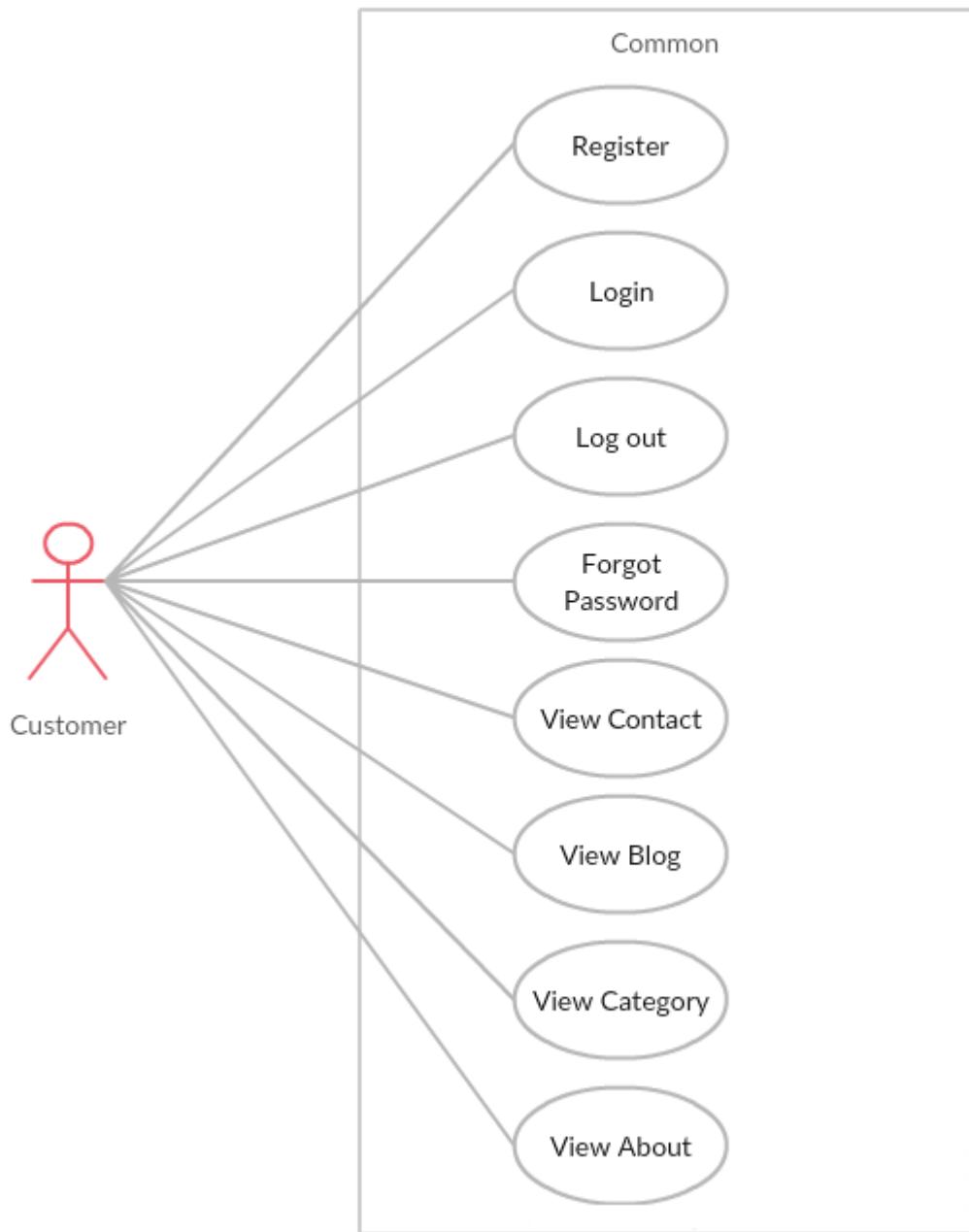
## 3.2.2.2.5.2 UC047 – Change Password

<b>Use Case ID</b>	UC047	<b>Version</b>	1.0		
<b>Use Case Name</b>	Change password				
<b>Created by:</b>	PhuongPTM	<b>Date Created:</b>	28/09/2017		
<b>Primary Actor</b>	Staff	<b>Secondary Actor</b>			
<b>Description:</b>	This function allows Staff to want change new password				
<b>Preconditions:</b>	1. BSO website is available 2. Staff browsed BSO website 3. Staff are viewing Sign in page for staff				
<b>Trigger:</b>	Staff click on Change password				
<b>Post conditions:</b>	Change password page is displayed				
<b>Normal Flow</b>					
<b>Step</b>	<b>Actor</b>	<b>Action</b>			
1	Staff	Click on Change password			
2	Staff	Fill information: - Old password - New password - Repeat password			
3	Staff	Click on Save button			
4	BSO	Save change			
<b>Alternative Flows: N/A</b>					
<b>Exceptions:</b>					
<b>EC1</b>	At step 3 in the main flows, if user miss fill 1 or more information				
<b>Step</b>	<b>Actor</b>	<b>Action</b>			
<b>3.1</b>	BSO	Display Dialog to remind user			
<b>Priority</b>	High				
<b>Frequency of Use:</b>	High				
<b>Business Rules:</b>	<u>B1</u>				
<b>Other Information:</b>	N/A				

Table 3-49: Change Password

### 3.2.2.3 Customer Group Function

#### 3.2.2.3.1 Common module



**Figure 3-13:** Registered common use case

##### 3.2.2.3.1.1 UC048 – Register

Use Case ID	UC048	Version	1.0
Use Case Name	Register		
Created by:	PhuongPTM	Date Created:	28/09/2017
Primary Actor	Customer	Secondary Actor	

<b>Description:</b>	Visitor can create new account to use more function	
<b>Preconditions:</b>	<ul style="list-style-type: none"> <li>– BSO must be connected to the Internet</li> <li>– User browsed BSO website</li> </ul>	
<b>Trigger:</b>	Click on sign up link on the website	
<b>Post conditions:</b>	<ol style="list-style-type: none"> <li>1. Add account information to database</li> <li>2. Send confirm email</li> <li>3. Logs user into system</li> <li>4. Redirect user to Homepage and display as Member</li> </ol>	
<b>Normal Flow</b>		
Step	Actor	Action
1	Customer	Click Sign up in header
2	Customer	Display Register page includes: <ul style="list-style-type: none"> <li>– User name text box</li> <li>– Password text box</li> <li>– Repeat Password text box</li> <li>– Email address text box</li> <li>– Sign up button</li> </ul>
3	Customer	Enter data in fields
4	Customer	Click on Sign up button
5	BSO	Send confirm email
6	BSO	Logs user into system
7	BSO	Display Homepage as Member
<b>Alternative Flows: N/A</b>		
<b>Exceptions:</b>		
<b>EC1</b>	At step 3, required fields are not entered	
Step	Actor	Action
<b>3.1</b>	BSO	Display Dialog to remind user
<b>Priority</b>	High	
<b>Frequency of Use:</b>	High	
<b>Business Rules:</b>	<u>B1</u>	
<b>Other Information:</b>	N/A	

**Table 3-50: Register Use case****3.2.2.3.1.2 UC049 – Log In**

<b>Use Case ID</b>	UC049	<b>Version</b>	1.0		
<b>Use Case Name</b>	Login				
<b>Created by:</b>	PhuongPTM	<b>Date Created:</b>	28/09/2017		
<b>Primary Actor</b>	Customer	<b>Secondary Actor</b>			
<b>Description:</b>	This function allows Customer logins into website				
<b>Preconditions:</b>	1. BSO must be connected to the Internet 2. User browsed BSO website				
<b>Trigger:</b>	Customer click sign in button				
<b>Post conditions:</b>	3. Log Customer into system 4. Redirect Customer to home page				
<b>Normal Flow</b>					
<b>Step</b>	<b>Actor</b>	<b>Action</b>			
1	BSO	Display Customer Login page. If Customer choose Remember Me function before, fill all needed information			
2	Customer	Type email address and password if Customer does not choose Remember Me function before			
3	BSO	Verify: Email address Password			
4	BSO	<ul style="list-style-type: none"> <li>– Log Customer into system</li> <li>– Redirect Customer to home page</li> </ul>			
<b>Alternative Flows: N/A</b>					
<b>Exceptions:</b>					
<b>EC1</b>	At step 2 in main flow, if Customer entered wrong Email address or Password				
<b>Step</b>	<b>Actor</b>	<b>Action</b>			
2.1	BSO	Display Login popup message point out error field			
2.2	BSO	Mark error fields			
<b>Priority</b>		High			
<b>Frequency of Use:</b>		High			
<b>Business Rules:</b>		<u>B1</u>			
<b>Other Information:</b>		N/A			

**Table 3-51:Log In Use case****3.2.2.3.1.3 UC050 – Log Out**

<b>Use Case ID</b>	UC050	<b>Version</b>	1.0		
<b>Use Case Name</b>	Logout				
<b>Created by:</b>	PhuongPTM	<b>Date Created:</b>	28/09/2017		
<b>Primary Actor</b>	Customer	<b>Secondary Actor</b>			
<b>Description:</b>	This function allows Customer logout website				
<b>Preconditions:</b>	Customer logged website				
<b>Trigger:</b>	Customer clicks on Sign out button				
<b>Post conditions:</b>	Customer logged out from BSO website				
<b>Normal Flow</b>					
Step	Actor	<b>Action</b>			
1	Customer	Choose sign out button			
2	BSO	Sign out Customer Redirect to Homepage			
<b>Alternative Flows: N/A</b>					
<b>Exceptions: N/A</b>					
<b>Priority</b>	High				
<b>Frequency of Use:</b>	High				
<b>Business Rules:</b>	<a href="#">B1</a>				
<b>Other Information:</b>	N/A				

**Table 3-52:Log Out Use case****3.2.2.3.1.4 UC051 – Forgot Password**

<b>Use Case ID</b>	UC051	<b>Version</b>	1.0
<b>Use Case Name</b>	Forgot password		
<b>Created by:</b>	PhuongPTM	<b>Date Created:</b>	28/09/2017
<b>Primary Actor</b>	Customer	<b>Secondary Actor</b>	
<b>Description:</b>	This function allows Customer receiving her/his password to mail		
<b>Preconditions:</b>	1. BSO must be connected to the Internet		

	2. User browsed BSO website	
<b>Trigger:</b>	Click Forgot password in Sign in page	
<b>Post conditions:</b>	Customer get new password on her/his email	
<b>Normal Flow</b>		
Step	Actor	Action
1	Customer	Click Forgot password in Sign in page
2	BSO	Display Email text box
3	Customer	Enter registered email
4	Customer	Click Forgot Password
<b>Alternative Flows: N/A</b>		
<b>Exceptions: N/A</b>		
<b>Priority</b>	High	
<b>Frequency of Use:</b>	High	
<b>Business Rules:</b>	<a href="#">B1</a>	
<b>Other Information:</b>	N/A	

**Table 3-53:Forgot Password Out Use case****3.2.2.3.1.5 UC052 – View Blog**

<b>Use Case ID</b>	UC052	<b>Use Case ID</b>	UC051
<b>Use Case Name</b>	View blog		
<b>Created by:</b>	PhuongPTM	<b>Created by:</b>	PhuongPTM
<b>Primary Actor</b>	Customer	<b>Secondary Actor</b>	
<b>Description:</b>	This function allows Customer can view blog		
<b>Preconditions:</b>	1. BSO must be connected to the Internet 2. User browsed BSO website		
<b>Trigger:</b>	Click on the blog button on menu bar		
<b>Post conditions:</b>	User will go to Detail Blog page		
<b>Normal Flow</b>			
Step	Actor	Action	
1	Customer	Click on the blog button on menu bar	
2	BSO	Display Blog page	

<b>Alternative Flows:</b> N/A	
<b>Exceptions:</b> N/A	
<b>Priority</b>	High
<b>Frequency of Use:</b>	High
<b>Business Rules:</b>	<a href="#">B1</a>
<b>Other Information:</b>	N/A

**Table 3-54: View Blog Use case**

## 3.2.2.3.1.6 UC053 – View Category

<b>Use Case ID</b>	UC053	<b>Use Case ID</b>	UC051		
<b>Use Case Name</b>	View category				
<b>Created by:</b>	PhuongPTM	<b>Created by:</b>	PhuongPTM		
<b>Primary Actor</b>	Customer	<b>Secondary Actor</b>			
<b>Description:</b>	This function allows Customer can view category				
<b>Preconditions:</b>	1. BSO must be connected to the Internet 2. User browsed BSO website				
<b>Trigger:</b>	Click on the category button on menu bar				
<b>Post conditions:</b>	User will go to category page				
<b>Normal Flow</b>					
<b>Step</b>	<b>Actor</b>	<b>Action</b>			
1	Customer	Click on the category button on menu bar			
2	BSO	Display Category page			
<b>Alternative Flows:</b> N/A					
<b>Exceptions:</b> N/A					
<b>Priority</b>	High				
<b>Frequency of Use:</b>	High				
<b>Business Rules:</b>	<a href="#">B1</a>				
<b>Other Information:</b>	N/A				

**Table 3-55: View Category Use case**

## 3.2.2.3.1.7 UC054 – View Contact

<b>Use Case ID</b>	UC053	<b>Use Case ID</b>	UC051		
<b>Use Case Name</b>	View contact				
<b>Created by:</b>	PhuongPTM	<b>Created by:</b>	PhuongPTM		
<b>Primary Actor</b>	Customer	<b>Secondary Actor</b>			
<b>Description:</b>	This function allows Customer can view contact of BS website				
<b>Preconditions:</b>	1. BSO must be connected to the Internet 2. User browsed BSO website				
<b>Trigger:</b>	Click on the contact button on menu bar				
<b>Post conditions:</b>	User will go to contact page				
<b>Normal Flow</b>					
<b>Step</b>	<b>Actor</b>	<b>Action</b>			
1	Customer	Click on the contact button on menu bar			
2	BSO	Display contact page			
<b>Alternative Flows: N/A</b>					
<b>Exceptions: N/A</b>					
<b>Priority</b>	High				
<b>Frequency of Use:</b>	High				
<b>Business Rules:</b>	<a href="#">B1</a>				
<b>Other Information:</b>	N/A				

Table 3-56: View Contact Use case

## 3.2.2.3.1.8 UC055 – View About

<b>Use Case ID</b>	UC055	<b>Use Case ID</b>	UC051
<b>Use Case Name</b>	View about		
<b>Created by:</b>	PhuongPTM	<b>Created by:</b>	PhuongPTM
<b>Primary Actor</b>	Customer	<b>Secondary Actor</b>	
<b>Description:</b>	This function allows Customer can view information about BS website		
<b>Preconditions:</b>	1. BSO must be connected to the Internet 2. User browsed BSO website		
<b>Trigger:</b>	Click on the about button on menu bar		
<b>Post conditions:</b>	User will go to about page		

Normal Flow		
Step	Actor	Action
1	Customer	Click on the about button on menu bar
2	BSO	Display about page
Alternative Flows: N/A		
Exceptions: N/A		
Priority	High	
Frequency of Use:	High	
Business Rules:	<a href="#">B1</a>	
Other Information:	N/A	

Table 3-57:- View About Use case

### 3.2.2.3.2 Order

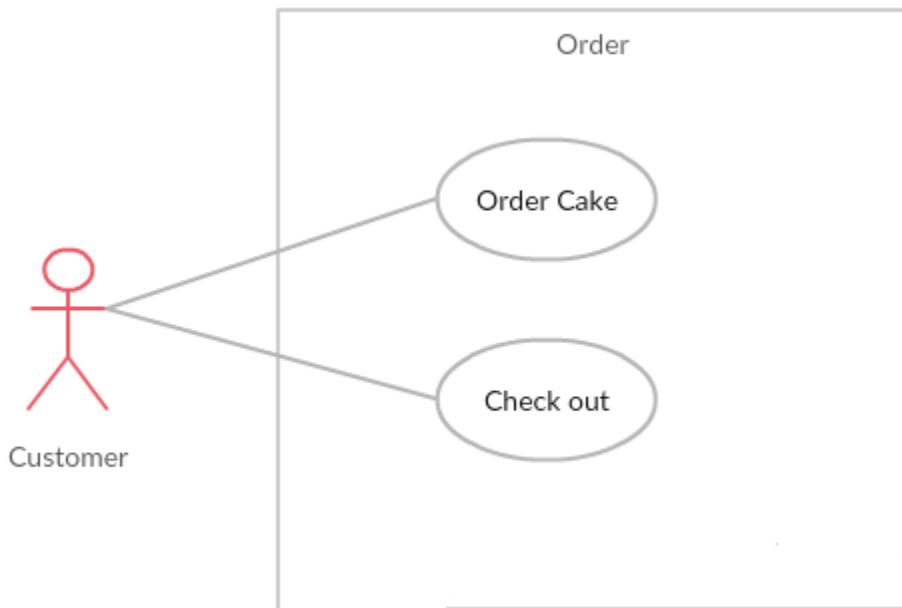


Figure 3-14: Order use cases

#### 3.2.2.3.2.1 UC056 – Order Cake

Use Case ID	UC056	Use Case ID	UC051
Use Case Name	Order Cake		
Created by:	PhuongPTM	Created by:	PhuongPTM

<b>Primary Actor</b>	Customer	<b>Secondary Actor</b>	
<b>Description:</b>	This function allows Customer can order cake		
<b>Preconditions:</b>	1. BSO must be connected to the Internet 2. User browsed BSO website		
<b>Trigger:</b>	Click on Add to cart		
<b>Post conditions:</b>	Display screen order		
<b>Normal Flow</b>			
<b>Step</b>	<b>Actor</b>	<b>Action</b>	
1	Customer	Choose a cake	
2	Customer	Click Add to cart	
3	BSO	Display screen order	
<b>Alternative Flows: N/A</b>			
<b>Exceptions: N/A</b>			
<b>Priority</b>	High		
<b>Frequency of Use:</b>	High		
<b>Business Rules:</b>	<a href="#">B1</a>		
<b>Other Information:</b>	N/A		

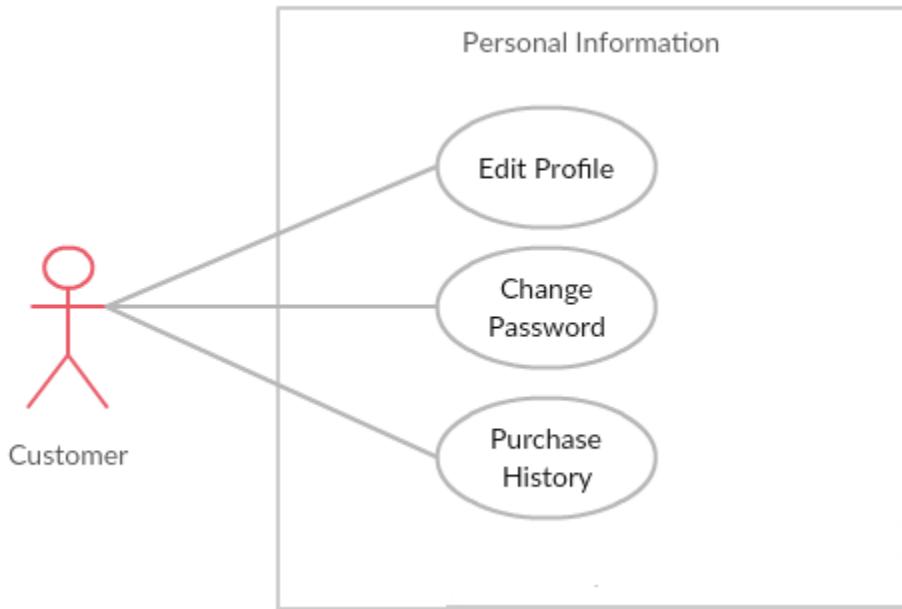
**Table 3-58: Order Cake Use case**

## 3.2.2.3.2.2 UC057 – Check Out

<b>Use Case ID</b>	UC057	<b>Use Case ID</b>	UC051		
<b>Use Case Name</b>	Check Out				
<b>Created by:</b>	PhuongPTM	<b>Created by:</b>	PhuongPTM		
<b>Primary Actor</b>	Customer	<b>Secondary Actor</b>			
<b>Description:</b>	This function allows Customer can check information about order				
<b>Preconditions:</b>	1. BSO must be connected to the Internet 2. User browsed BSO website				
<b>Trigger:</b>	Click on shopping cart				
<b>Post conditions:</b>	Display shopping cart screen				
<b>Normal Flow</b>					
<b>Step</b>	<b>Actor</b>	<b>Action</b>			
1	Customer	Click Cart			
2	BSO	Display shopping cart screen			
<b>Alternative Flows: N/A</b>					
<b>Exceptions: N/A</b>					
<b>Priority</b>	High				
<b>Frequency of Use:</b>	High				
<b>Business Rules:</b>	<a href="#">B1</a>				
<b>Other Information:</b>	N/A				

Table 3-59: Check Out Use case

### 3.2.2.3.3 Personal Information



**Figure 3-15:** Personal Information use cases

#### 3.2.2.3.3.1 UC058 – Edit Profile

<b>Use Case ID</b>	UC058	<b>Version</b>	1.0
<b>Use Case Name</b>	Update information		
<b>Created by:</b>	PhuongPTM	<b>Date Created:</b>	28/09/2017
<b>Primary Actor</b>	Customer	<b>Secondary Actor</b>	
<b>Description:</b>	This function allows Customer to update profile		
<b>Preconditions:</b>	1. BSO must be connected to the Internet 2. User browsed BSO website		
<b>Trigger:</b>	Click to Account Information		
<b>Post conditions:</b>	Update information page is displayed		
<b>Normal Flow</b>			
<b>Step</b>	<b>Actor</b>	<b>Action</b>	
1	Customer	Click to Account Information	
2	Customer	Update information	
3	Customer	Click on Save button	
4	BSO	Save change	

<b>Alternative Flows: N/A</b>		
<b>Exceptions:</b>		
<b>EC1</b>	At step 3 in the main flows, if user miss fill 1 or more information	
<b>Step</b>	<b>Actor</b>	<b>Action</b>
<b>3.1</b>	BSO	Display Dialog to remind user
<b>Priority</b>	High	
<b>Frequency of Use:</b>	High	
<b>Business Rules:</b>	<a href="#"><u>B1</u></a>	
<b>Other Information:</b>	N/A	

**Table 3-60: Edit Profile Use case**

## 3.2.2.3.3.2 UC059 – Change Password

<b>Use Case ID</b>	UC059	<b>Version</b>	1.0		
<b>Use Case Name</b>	Change password				
<b>Created by:</b>	PhuongPTM	<b>Date Created:</b>	28/09/2017		
<b>Primary Actor</b>	Customer	<b>Secondary Actor</b>			
<b>Description:</b>	This function allows Staff to want change new password				
<b>Preconditions:</b>	1. BSO must be connected to the Internet 2. User browsed BSO website				
<b>Trigger:</b>	Customer click on Change password				
<b>Post conditions:</b>	Change password page is displayed				
<b>Normal Flow</b>					
<b>Step</b>	<b>Actor</b>	<b>Action</b>			
1	Customer	Click on Change password			
2	Customer	Fill information: - Old password - New password - Repeat password			
3	Customer	Click on Save button			
4	BSO	Save change			
<b>Alternative Flows: N/A</b>					
<b>Exceptions:</b>					
<b>EC1</b>	At step 3 in the main flows, if user miss fill 1 or more information				
<b>Step</b>	<b>Actor</b>	<b>Action</b>			
<b>3.1</b>	BSO	Display Dialog to remind user			
<b>Priority</b>	High				
<b>Frequency of Use:</b>	High				
<b>Business Rules:</b>	<u>B1</u>				
<b>Other Information:</b>	N/A				

Table 3-61: Change Password use case

### 3.2.2.3.3.3 UC060 – Purchase History

<b>Use Case ID</b>	UC060	<b>Version</b>	1.0		
<b>Use Case Name</b>	Change password				
<b>Created by:</b>	PhuongPTM	<b>Date Created:</b>	28/09/2017		
<b>Primary Actor</b>	Customer	<b>Secondary Actor</b>			
<b>Description:</b>	This function allows Staff to want change new password				
<b>Preconditions:</b>	1. BSO must be connected to the Internet 2. User browsed BSO website				
<b>Trigger:</b>	Customer click on Purchase History				
<b>Post conditions:</b>	Purchase History page is displayed				
<b>Normal Flow</b>					
<b>Step</b>	<b>Actor</b>	<b>Action</b>			
1	Customer	Click on Purchase History			
2	Customer	Purchase History page is displayed			
3	BSO	User can view all the has bought.			
<b>Alternative Flows: N/A</b>					
<b>Exceptions:N/A</b>					
<b>Priority</b>	High				
<b>Frequency of Use:</b>	High				
<b>Business Rules:</b>	<a href="#">B1</a>				
<b>Other Information:</b>	N/A				

**Table 3-62: Purchase History Use case**

## 3.2.3 Non-Functional Specification

### 3.2.3.1 Reliability

- Mean time between failures: The mean time between each failure is expected to be about 480 hours of using.
- The maximum bugs per function are 2 bugs/a function when user uses software within 2 months. The average number of bugs in the software during system testing is about to be around 1 bugs/KLOC.
- The database must be backed up daily and can be recovered if necessary

### 3.2.3.2 Performance Requirements

#### 3.2.3.2.1 Availability

The application must be available 95% of time. Users can access to it everywhere from their Web browser with internet connection.

#### 3.2.3.2.2 Response time

In term of response time, waiting time of processing will be 8 to 10 seconds; time to process any functions will not exceed 15 seconds.

#### 3.2.3.2.3 Maintainability

- **Coding standards and naming conventions:**
  - Output of the project must include coding standards and naming conventions documentations. Implementation code must be easy to maintain.
  - If some components are reused, the documentations of those components must also be included.
- **Design:**
  - The design of the system must be loosely coupled that chances on some module will not affect others.
- **Logging:**
  - All the errors should be logged, supporting for bug fixing and maintenance.
  - All strange or sensitive situations should also be logged.
- **Mean time to repair:**
  - Immediately when Administrator finds out problem or website is attacked/ hacked by someone. Average 1 day.

#### 3.2.3.2.4 Usability

- Usability Requirements support the following from the perspective of its primary users:
  - Efficiency of use: user can complete each function in less than 12 actions
- Intuitiveness: User can understand approximately 90% help/ error messages

#### 3.2.3.2.5 Security Requirements

In order to avoid unwanted access by strangers, the security implemented in various ways:

- **User authorize:** Each function must be distributed system access and manipulate the individual parts. To edit information, user must declare their username and password before the operation. Functions will be show after users login dependencies user's role and permission of each user.
- **Manage Data Layer:** Only Administrator can modify database system.

#### 3.2.3.2.6 Hardware/Software Requirements

- **Hardware requirement:**
  - Server: Azure server, IIS8 or higher
  - Client: PC/Laptop (window 7, window 8, window 10)
- **Software requirements:**
  - Web browser (Chrome 40.0, Firefox 30.0) with internet connection

## 4 SOFTWARE DESIGN

### 4.1 Introduction

#### 4.1.1 Purpose

This chapter provides a guideline for the development team of what the system's architect, how to implement functions.

This section of document is the software design of the project in order to develop the software, it includes four main parts:

- Architecture Design.
- Detail Design.
- Database design.
- Screen design.

#### 4.1.2 Definitions, Acronyms, Abbreviations

Acronym	Definition	Note
BSO	Bakery Store Online	
DB	Database	
MVC	Model view control	
IDE	Integrated development environment	
Q&A	Question and answer	
GUI	Graphic user interface	
FU	FPT University	

**Table 4-1: Definitions and Acronyms**

### 4.2 Architecture design

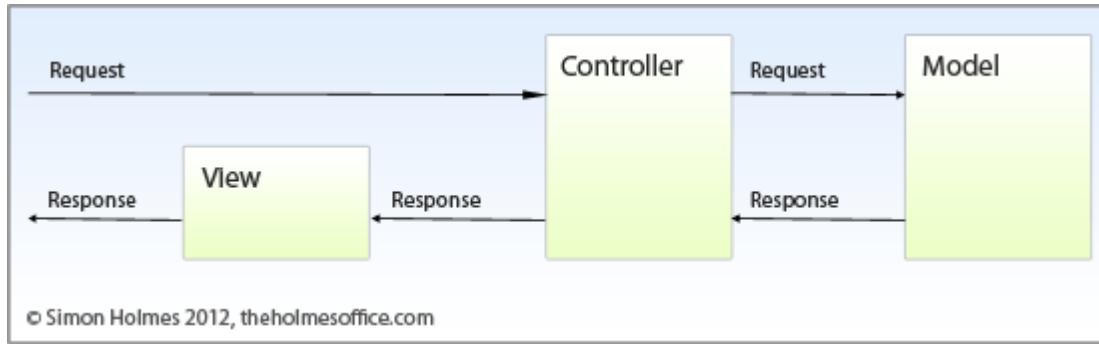
#### 4.2.1 Choice of Architecture Design

##### 4.2.1.1 MVC Model

###### 4.2.1.1.1 MVC Model Overview

The A very popular approach for web sites & web apps – and the one we'll be working with – is MVC, which stands for model, view, controller. This makes distinctions between the model (data), view (presentation) and controllers (logic).

The diagram below shows how a standard MVC request/response loop looks:



**Figure 4-1: MVC Model**

In addition to dividing the application into three kinds of components, the MVC design defines the interactions between them:

- **Controllers:** Controller is a level which acts like a brain of the entire MVC system.. A controller also acts as a link between a user and the system. It provides the user with input by providing appropriate views to present it appropriately on the screen. The controller understands user output, converts it into the appropriate messages and passes the same to views.
- **Models:** This level is very important as it represents the data to the user. This level defines where the application's data objects are stored. The model doesn't know anything about views and controllers. So, whenever there are changes done in the model it will automatically notify observers that the changes are made. The model may be a single object or a structure of objects.
- **Views:** A view is a visual representation of the MVC model. This level creates an interface to show the actual output to the user. However, a view will not display anything itself. It is the controller or model that tells view what to display to the user. It also handles requests from the user and informs controller. A view is connected to its model and gets the data necessary for the presentation by asking certain questions. Sometimes, it also updates the model by sending appropriate messages. All these questions and messages are sent back to the model in such an easy terminology that it can easily understand the information sent by model or a controller.

#### 4.2.1.1.2 Advantages and disadvantages of MVC Model

- **Advantage:**
  - **Simultaneous development –** Multiple developers can work simultaneously on the model, controller and views..

- **Modification does not affect the entire model:** Modification does not affect the entire model because model part does not depend on the views part. Therefore, any changes in the Model will not affect the entire architecture..
- **MVC model returns the data without formatting:** MVC pattern returns data without applying any formatting so the same components can be used and called for use with any interface
- **Ease of modification :** Because of the separation of responsibilities, future development or modification is easier.
- **Disadvantage:**
  - For small projects that apply MVC model caused cumbersome, time consuming in development process.
  - Time consuming to transits data between components.
  - The MVC model takes time to learn and apply properly.

#### 4.2.1.1.3 The reasons for choosing MVC Model

- Better support for test-driven development (TDD)
- Helps to maintain, move, and organize applications more easily.
- Allow modification of code without affecting much of the other parts.

#### 4.2.1.2 Node.js

##### 4.2.1.2.1 Node.js Overview

Node.js is a platform built on Chrome's JavaScript runtime for easily building fast and scalable network applications. Node.js uses an event-driven, non-blocking I/O model that makes it lightweight and efficient, perfect for data-intensive real-time applications that run across distributed devices.

Node.js is an open source, cross-platform runtime environment for developing server-side and networking applications. Node.js applications are written in JavaScript, and can be run within the Node.js runtime on OS X, Microsoft Windows, and Linux.

Node.js also provides a rich library of various JavaScript modules which simplifies the development of web applications using Node.js to a great extent.

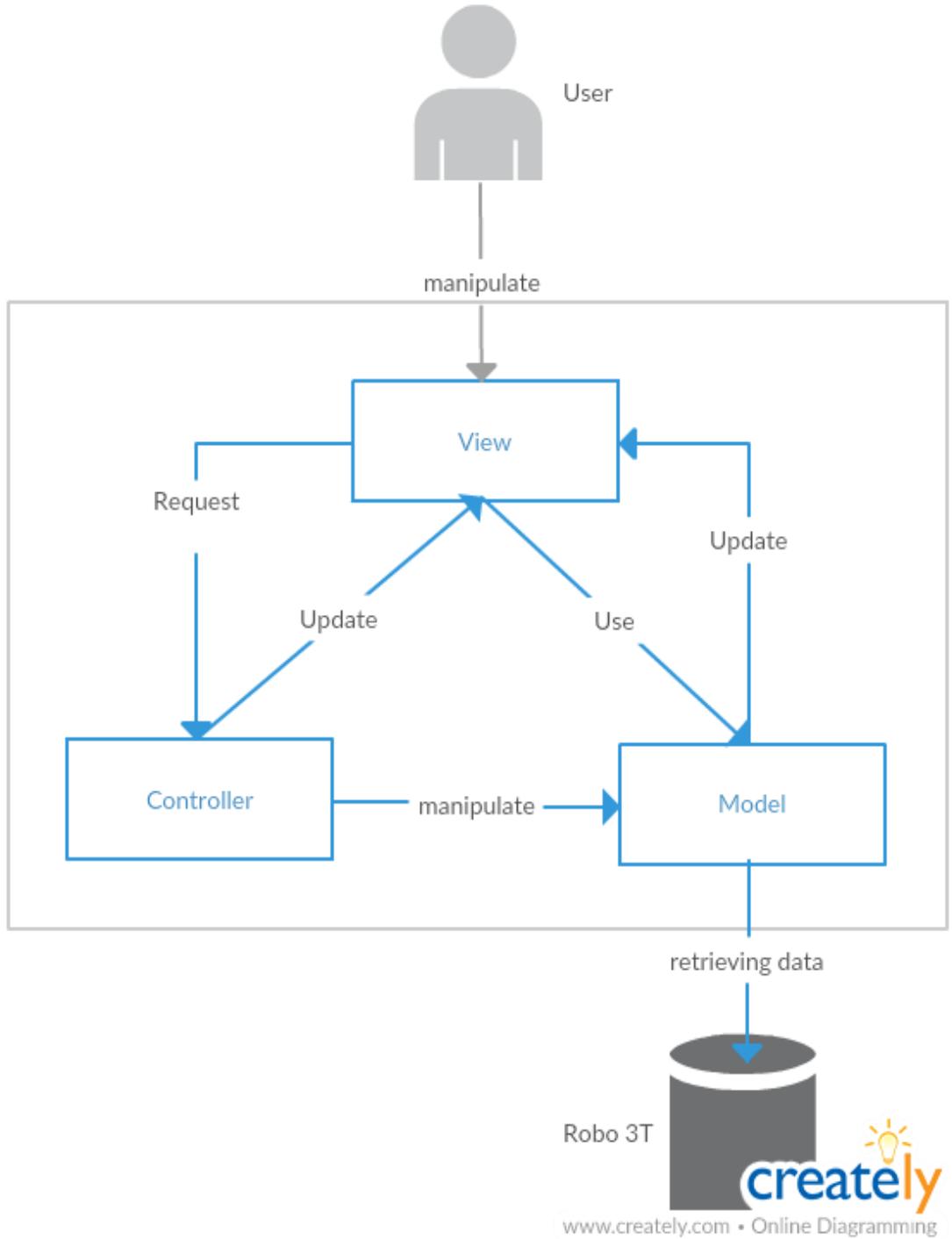
#### 4.2.1.2.2 Advantages and disadvantages of Node.js

- Advantages:
  - Node.js is an open-source framework under MIT license. (MIT license is a free software license originating at the Massachusetts Institute of Technology (MIT).)
  - Uses JavaScript to build entire server side application.
  - Lightweight framework that includes bare minimum modules. Other modules can be included as per the need of an application.
  - Asynchronous by default. So it performs faster than other frameworks.
  - Cross-platform framework that runs on Windows, MAC or Linux
- Disadvantages:
  - It doesn't support multi-threaded programming.
  - It doesn't support very high computational intensive tasks. When it executes long running task, it will queue all the incoming requests to wait for execution, since it follows JavaScript event loop which is single threaded.
  - Node good for executing synchronous and CPU intensive tasks.

#### 4.2.1.2.3 The reasons for choosing Node.js

- **Very Fast** – Being built on Google Chrome's V8 JavaScript Engine, Node.js library is very fast in code execution.
- **Single Threaded but Highly Scalable** – Node.js uses a single threaded model with event looping. Node.js uses a single threaded program and the same program can provide service to a much larger number of requests than traditional servers like Apache HTTP Server.
- **No Buffering** – Node.js applications never buffer any data. These applications simply output the data in chunks.
- **License** – Node.js is released under the MIT license

#### 4.2.2 Architecture Representation



**Figure 4-2: System Overview**

We follow MVC architecture to implement the BSO Project. MVC offers architectural benefits over standard Jquery and AngularJS — it helps us write better-organized and therefore more maintainable code.

- **DTO Model (data transfer object model)** is an object that defines how the data will be sent over the network to remove circular references from data model, Hide particular properties that clients are not supposed to view, omit some properties in order to reduce payload size, flatten object graphs that contain nested objects, to make them more convenient for clients, avoid “over-posting” vulnerabilities and decouple your service layer from your database layer.
- **View** is what is presented to the users and how users interact with the system. The view is expected to render the model in a meaningful way to the user. In BSO, the view is made with .cshtml file including css, AngularJS and jQuery, it sends user gestures to controller and allows controller to select view.
- **Controller** is the decision maker and the glue between the model and view; it handles user actions and gestures, and responds to user events. For example, in CMS, when a user clicks the “Create” button to create a new contract, the controller for that action is invoked. The controller will then make changes to the contract model. The view will then render the modified contract model to the display so that user can view the new contract he added in the contract list.
- **Data Model** is where the application’s data objects are stored. A model object is in charge of encapsulating application state and one object could be related to other objects establishing a one-to-one or one-to-many relationship.
- **Repository** is intermediate layer which used to separate the controller and the data access layer (database context). It queries the data source for data, maps it to DTO models, processes data and returns data to controller.

#### 4.2.3 Architectural Goals and Constraints

❖ **Availability:**

- The application must be available 95% of time. Users can access to it everywhere from there .Web browser with internet connection.

❖ **Maintainability:**

- Coding standards and naming conventions:
  - Output of the project must include coding standards and naming conventions documentations. Implementation code must be easy to maintain.
  - All code must be clearly commented, including class, method documentations.
  - If some components are reused, the documentations of those components must also be included.
- Design:
  - The design of the system must be loosely coupled that chances on some module will not affect others.

- Logging:
  - All the errors should be logged, supporting for bug fixing and maintenance.
  - All strange or sensitive situations should also be logged.
- ❖ Usability:
  - Intuitiveness: all help/error messages are simple to understand; user can know exactly how to do each feature after one time using it.
- ❖ Capacity and scalability:
  - Throughput, storage and growth requirements.

#### 4.2.4 Logical View

##### 4.2.4.1 Overview

Logical View includes Package diagram and Class diagram. Package diagram describes the organization of packages and elements. Class Diagram provides an overview of the target system by describing the objects and classes inside the system and the relationships between them. It provides a wide variety of usages; from modeling the domain-specific data structure to detailed design of the target system

- Controller contain the interface between:
  - Associated models
  - Associated views
  - The input devices (e.g., keyboard, pointing device, time).
  - Send commands to the model to update the model's state.
- Model is:
  - the domain-specific software simulation
  - Or implementation of the application's central structure.
- View: deal with everything graphical:
  - Requests data from their model
  - Display the data.
- Repository:
  - Create queries to DB.
  - Process data.
  - Return to controller.

#### 4.2.4.2 Architecturally Significant Design Packages

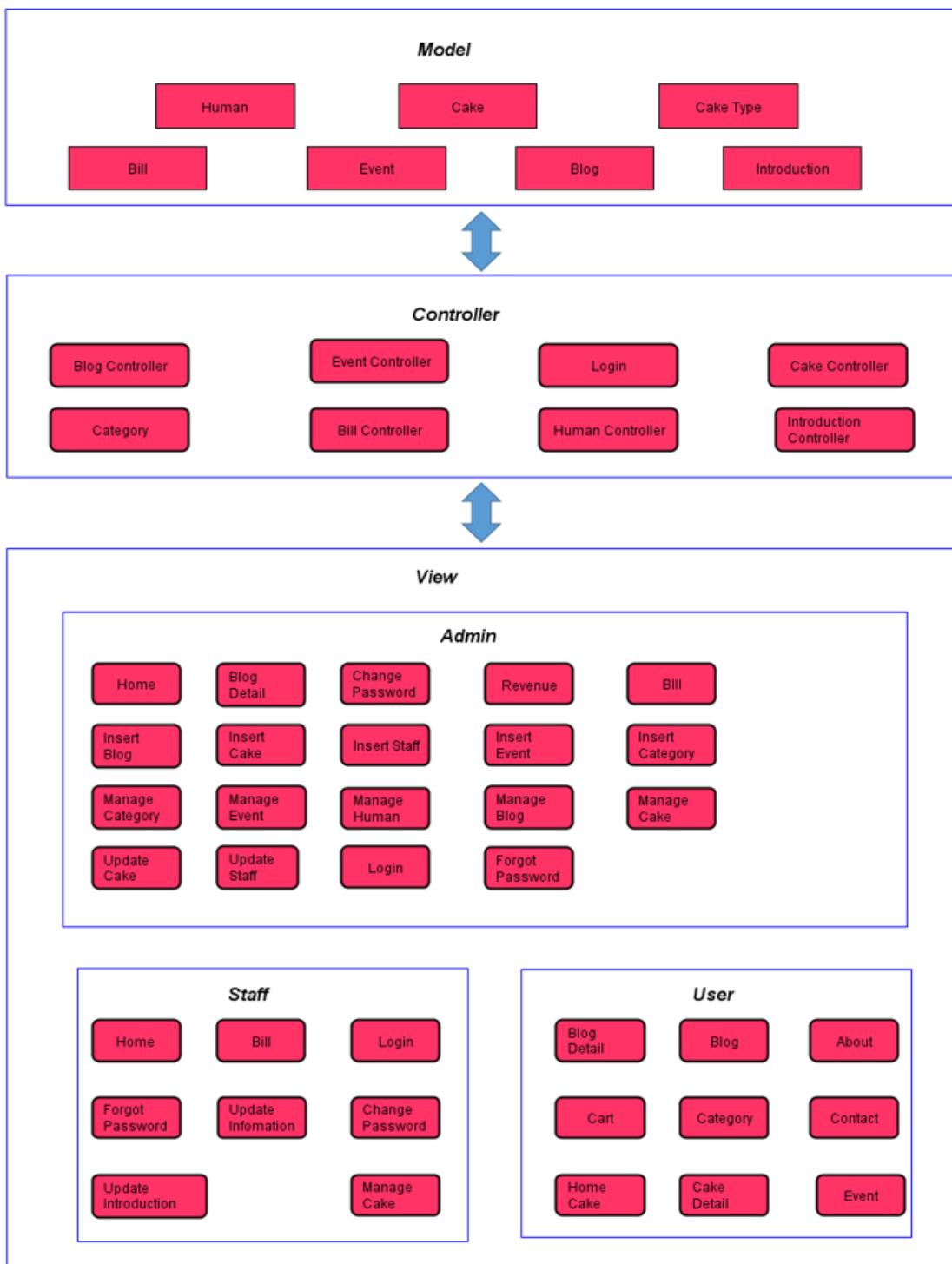


Figure 4-3: Package Diagram

❖ **Model:**

No	Model class	Role
----	-------------	------

<b>1</b>	Human	Description entity of User in database
<b>2</b>	CakeType	Description entity of Category in database
<b>3</b>	Cake	Description entity of Cake in database
<b>4</b>	Event	Description entity of Event in database
<b>5</b>	Blog	Description entity of Blog in database
<b>6</b>	Bill	Description entity of Bill in database
<b>7</b>	Introduction	Description entity of Introduction in database

**Table 4-2: Model Package**

❖ **Controller:**

No	Controller class	Role
1	LoginController	Receive request login, logout from client. Call method login, logout.  Respond login view and login, logout status.
2	HomeController	Receive request to home page from client. Handle request from client and call methods to get all data of home page.  Respond data back to Home View.
3	AccountAdminController	Receive requests (get, post, push, update) about Project from admin. Respond status, message and data to client.
4	CakeAdminController	Receive requests (get, post, push, update) about Project from admin. Respond status, message and data to client.
5	BillController	Receive request get admin partial views of from client. Generate partial views and respond to client.

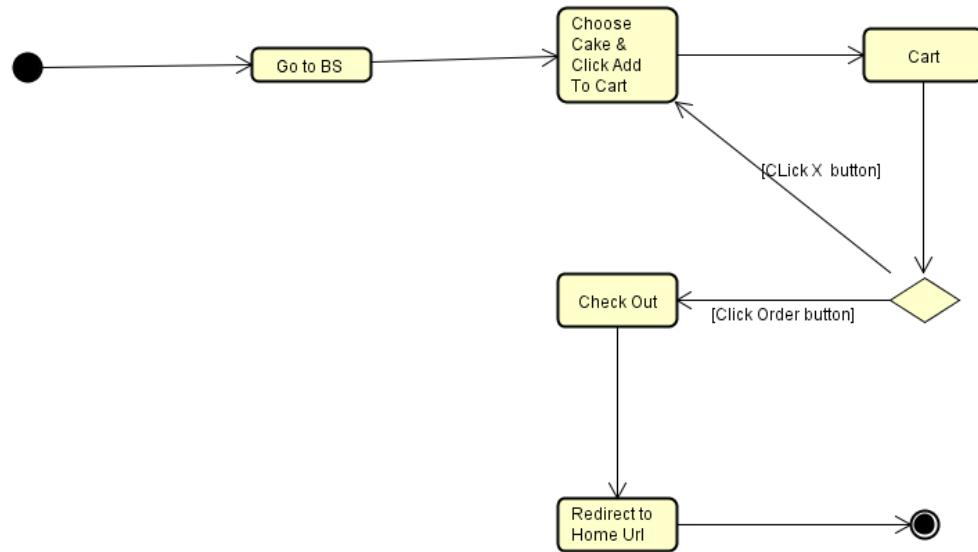
6	RevenueController	Receive request get admin partial views of from client. Generate partial views and respond to client.
7	ForgotPasswordController	Receive requests from client and send email for client and respond message
8	OrderCakeController	Receive requests from client and respond message

**Table 4-3:** Controller Package**❖ Repository:**

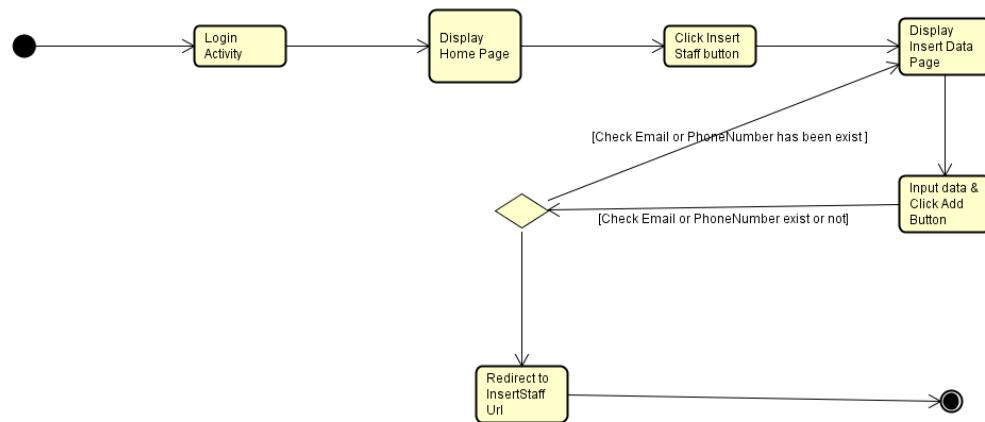
No	Repository class	Role
1	BSDataContext	Connect to DB, create queries to select, update, deletes data.
2	UserRepository	Process user data and return to controller.
3	CakeRepository	Process cake data and return to controller.
4	BlogRepository	Process blog data and return to controller.
5	EventRepository	Process event data and return to controller.
6	RevenueRepository	Process revenue data and return to controller.
7	BSDataContext	Connect to DB, create queries to select, update, deletes data.

**Table 4-4:** Repository Package

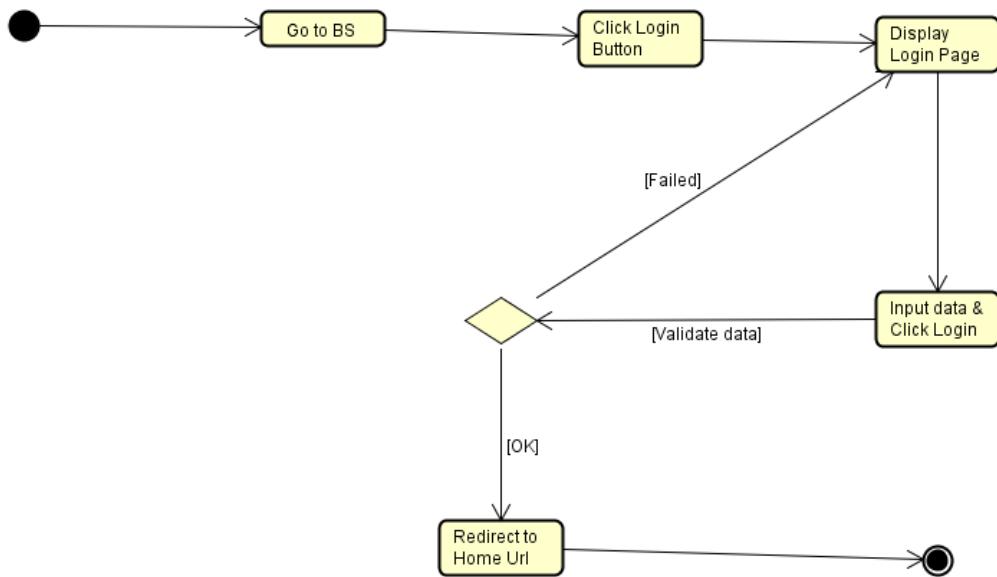
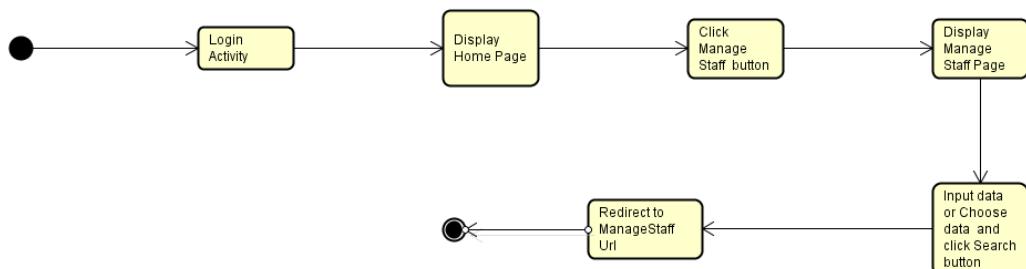
#### 4.2.5 Process view

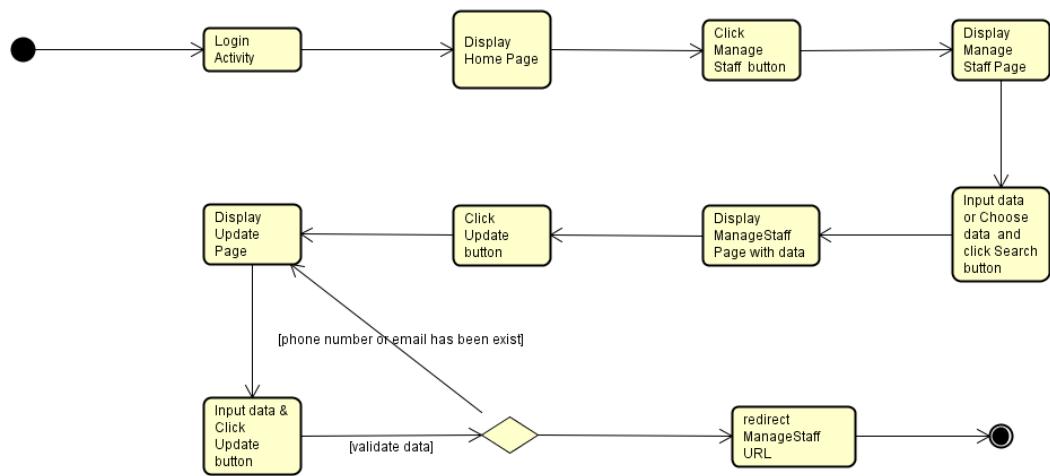


**Figure 4-4:** Check Out activity diagram



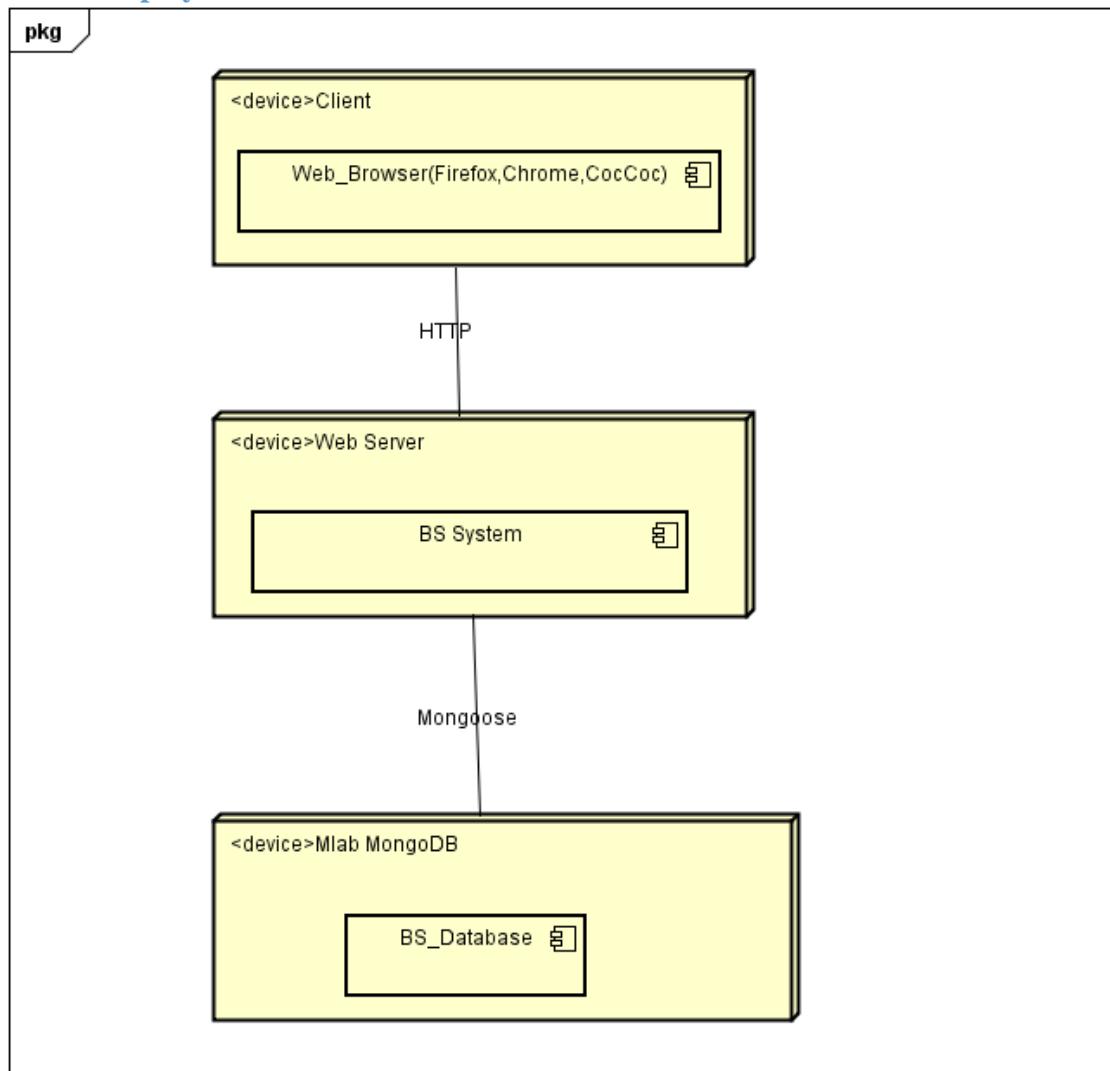
**Figure 4-5:** Insert staff activity diagram

**Figure 4-6:** Log In activity diagram**Figure 4-7:** Search Staff activity diagram



**Figure 4-8:** Update Staff activity diagram

#### 4.2.6 Deployment View



**Figure 4-9:** Deployment Diagram

No	Name	Description
1	MongoDB	SQL Server use to store system's data. Using SQL Server 2010 or higher.
2	Client	Client is web browser to use system. Firefox 30, Chrome 40 or higher.
3	Web Server	Web server is hosted by...

**Table 4-5:** Deployment Diagram Description

## 4.3 Detail Design

### 4.3.1 Package

#### 4.3.1.1 Package Diagram

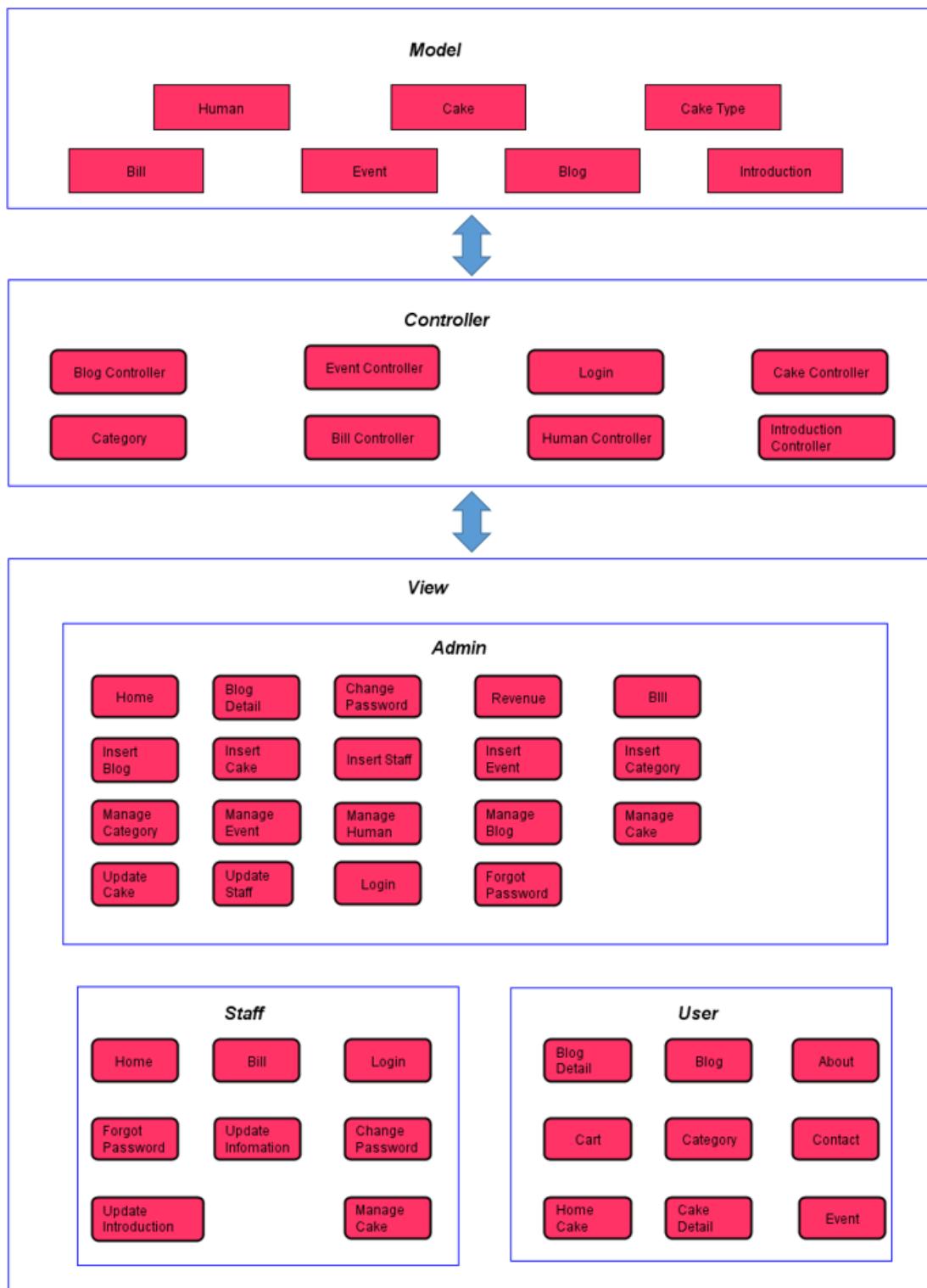


Figure 4-10: Package Diagram

#### 4.3.1.2 Package Description

##### 4.3.1.2.1 Model

###### 4.3.1.2.1.1 Bill

No	Field Name	Type	Allow Null	Description
1	_id	ObjectId	No	id of Bill
2	cake_id	ObjectId		id of Cake
3	name_id	ObjectId		id of Customer
4	name	String	No	name of Customer
5	phone	String	No	phone number of Customer
6	address	String	No	shipping address for Cusomer
7	email	String		email of Customer
8	confirm	Boolean		confirm has been shipped
9	time	Date	No	time order cake
10	Note	String		note for staff
11	cake	Array	No	information of order detail

**Table 4-6: Bill model**

###### 4.3.1.2.1.2 Blog

No	Field Name	Type	Allow Null	Description
1	_id	ObjectId	No	id of Blog
2	title	String	No	title of Blog
3	content	String	No	content of Blog
4	link	String	No	image link of content's Blog
5	date_create	Date	No	time create blog
6	ImagePanel	String	NO	imge link of panel's Blog

**Table 4-7: Blog model**

###### 4.3.1.2.1.3 Cake

No	Field Name	Type	Allow Null	Description
1	_id	ObjectId	No	id of Cake
2	cake_type_ids	ObjectId	No	id of Category
3	name_cake	String	No	name of Cake
4	price	String	No	price of Cake
5	name_size	String	No	size of Cake

6	linkImage	String	No	link Image of Cake
7	quantity	String	No	quantity of Cake exist
8	description	String	No	description about Cake
9	status	Boolean	No	status of Cake sell
10	date_create	Date	No	time create Cake
11	material	Array	No	material of Cake

**Table 4-8:** Cake model**4.3.1.2.1.4 CakeType**

No	Field Name	Type	Allow Null	Description
1	_id	ObjectId	No	id of Category
2	cake_type	String	No	name of Category
3	Blocked	Boolean	No	status of Category

**Table 4-9:** CakeTpye Model**4.3.1.2.1.5 Event**

No	Field Name	Type	Allow Null	Description
1	_id	ObjectId	No	id of event
2	title	String	No	title of event
3	content	String	No	content of event
4	link	String	No	link image of event
5	date_create	Date	No	time create event
6	lock	Boolean	No	status of event

**Table 4-10:** Event Model

#### 4.3.1.2.1.6 Human

No	Field Name	Type	Allow Null	Description
1	_id	ObjectId	No	id of User
2	name	String	No	name of User
3	phone	String	No	phone number of User
4	password	String	No	password of User
5	email	String	No	email of User
6	address	String	No	address of User
7	role	Int	No	role of User
8	note	String		note for Manage
9	linkImage	String	No	link Image of User
10	date_create	Date	No	time create Account
11	date_update	Date	No	time update Account
12	black_list	Boolean	No	blocked Account

**Table 4-11: Human Model**

#### 4.3.1.2.1.7 Introduction

No	Field Name	Type	Allow Null	Description
1	_id	ObjectId	No	id of store
2	address	String	No	name of store
3	phone	String	No	phone number of store
4	email	String	No	password of store
5	linkImage	String	No	linkImage map of store
10	date_update	Date	No	time update Introduction

**Table 4-12: Introduction Model**

#### 4.3.1.2.2 View

Contain many .ejs files to display user interface.

#### 4.3.1.2.3 Controller

##### 4.3.1.2.3.1 LoginController

No	Method	Parameter	Return	Description
1	Logout	Object	ActionResult	Log out Account
2	adminLocal	Object	ActionResult	Check email and password of Admin login to website and save to session
3	staffLocal	Object	ActionResult	Check email and password of Staff login to website and save to session
4	userLocal		ActionResult	Check email and password of User login to website and save to session
5	CheckLOginStaff		ActionResult	Check Status Login of Staff
6	CheckLoginUser		ActionResult	Check Status Login of User
7	CheckLoginAdmin		ActionResult	Check Status Login of Admin

**Table 4-13: Login Controller**

##### 4.3.1.2.3.2 HumanController

No	Method	Parameter	Return	Description
1	View		ActionResult	Show all details of account had exist for admin.
2	Search	Object	ActionResult	Search information about Staff or Customer by email or role
3	Insert	Object	ActionResult	User sign up or admin add new a Staff
4	Update	Object		update information for account
5	Block	String		Admin can lock account of customer and staff
6	unBlock	String		Admin can unlock account of customer and staff
7	changePassword	String		Change Password of Account
8	sendEMail	String		Email will send one new password with String Random to User

**Table 4-14:** Home Controller

## 4.3.1.2.3.3 CakeController

No	Method	Parameter	Return	Description
1	Insert	Object		Add new cake
2	View		ActionResult	View information of Cakes
3	Search	Object	ActionResult	Search cake by name or category
4	Update	Object		Update information cake
5	Lock	String		Update Status cake not sell
6	Unlock	String		Update status of cake (sell again )

**Table 4-15:** CakeController

## 4.3.1.2.3.4 categoryController

No	Method	Parameter	Return	Description
1	Insert	Object		Insert new category
2	View		ActionResult	View all category
3	Unlock	String		Update status of category
4	Lock	String		Update status of category

**Table 4-16:** categoryController

## 4.3.1.2.3.5 Event Controller

No	Method	Parameter	Return	Description
1	Insert	Object		Insert new event
2	View		ActionResult	View all event exist
3	Lock	String		Update status of event (close)
4	unLock	String		Update status of event (open)
5	Search	String	ActionResult	Find information an event

**Table 4-17:** Event Controller

## 4.3.1.2.3.6 BlogController

No	Method	Parameter	Return	Description
1	Create	Object		Insert new blog

2	View		ActionResult	View all blog
3	Search	String		Search blog by title
4	blogDetail	String	ActionResult	Show detail information of blog
5	Lock	String		Update status of blog(lock)
6	unLock	String		Update status of blog(open)

**Table 4-18:**  
**BlogController**

#### 4.3.1.2.3.7 BillController

No	Method	Parameter	Return	Description
1	newBill	Object		Insert new bill
2	Search	String	ActionResult	Find bill
3	View		ActionResult	View information of bill
4	confirmBill	String		Confirm bill (bill has been transported)
5	unConfirmBill	String		Unconfirm Bill
6	Note	String		Note information of bill

**Table 4-19:**  
**BillController**

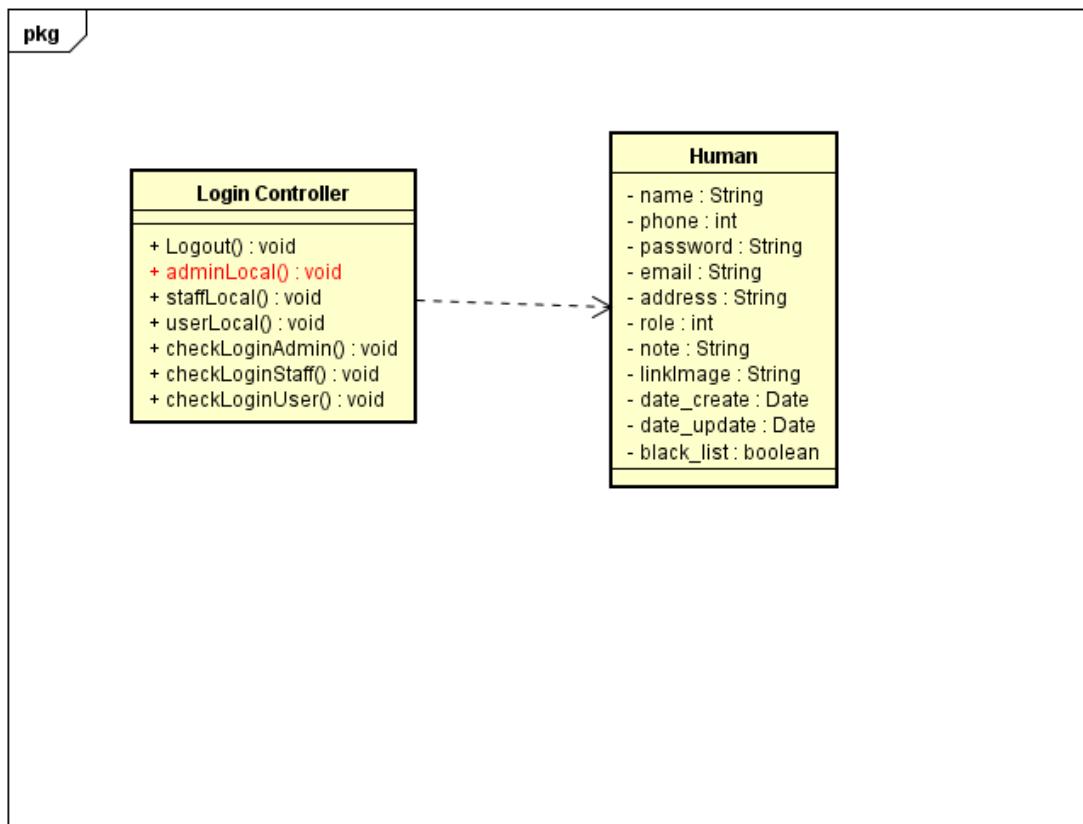
#### 4.3.1.2.3.8 IntroductionController

No	Method	Parameter	Return	Description
1	add	Object		Add new information about website
2	update	Object		Update information about website

**Table 4-20:**  
**IntroductionController**

#### 4.3.1.3 Class Diagram

##### 4.3.1.3.1 Admin Login

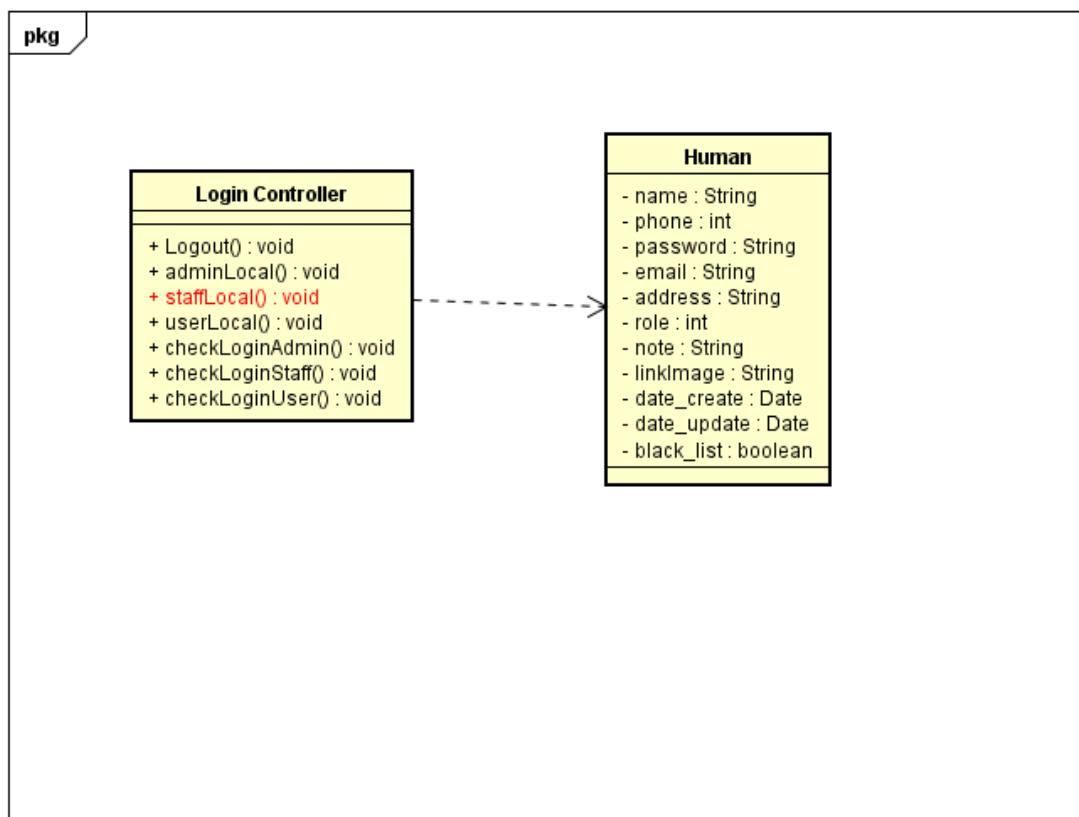


**Figure 4-11:** Admin Login

<b>Class</b>	LoginController	
<b>Description</b>	Admin Login	
<b>Base Class</b>	Controller	
<b>Source File</b>	LoginController.js	
<b>Namespace</b>	Controllers	
<b>Attributes</b>	N/A	
<b>Methods</b>	<b>Name</b>	<b>Description</b>
	adminLocal()	Admin login to website

**Table 4-21:** Admin login

#### 4.3.1.3.2 Staff Login

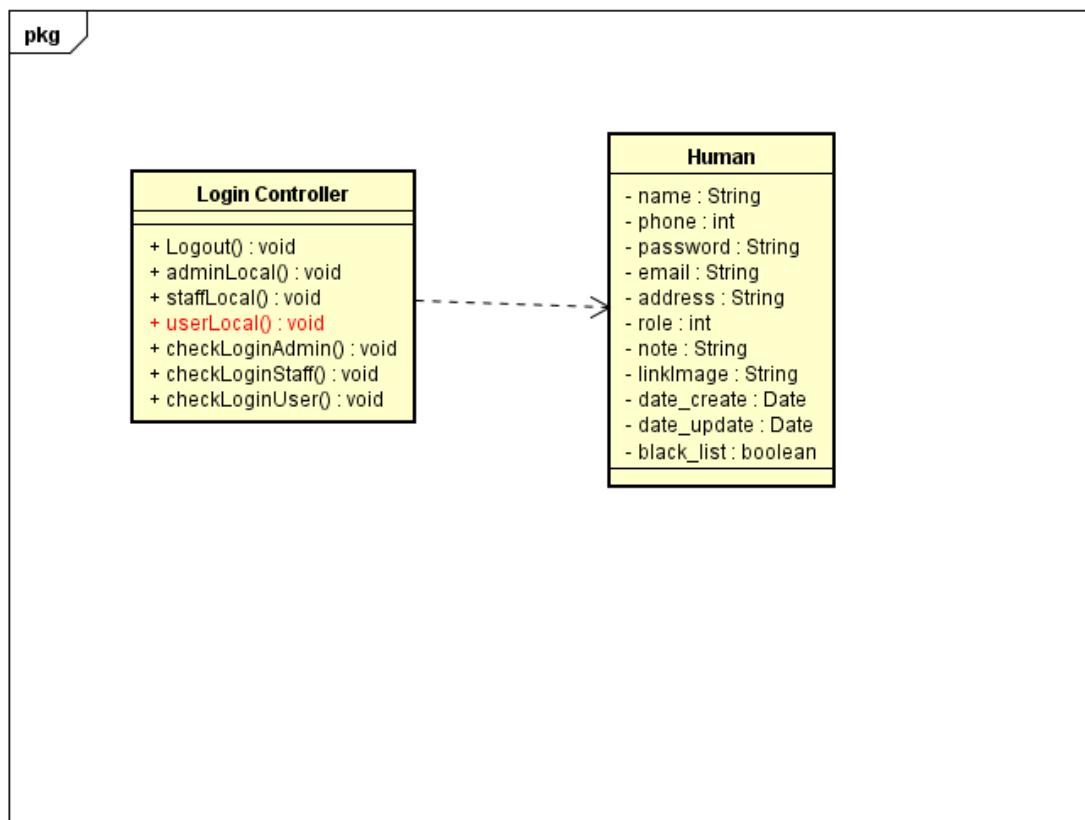


**Figure 4-12:** Staff Login

<b>Class</b>	LoginController	
<b>Description</b>	Staff Login	
<b>Base Class</b>	Controller	
<b>Source File</b>	LoginController.js	
<b>Namespace</b>	Controllers	
<b>Attributes</b>	N/A	
<b>Methods</b>	<b>Name</b>	<b>Description</b>
	staffLocal()	Staff login to website

**Table 4-22:** Staff Login

#### 4.3.1.3.3 User Login

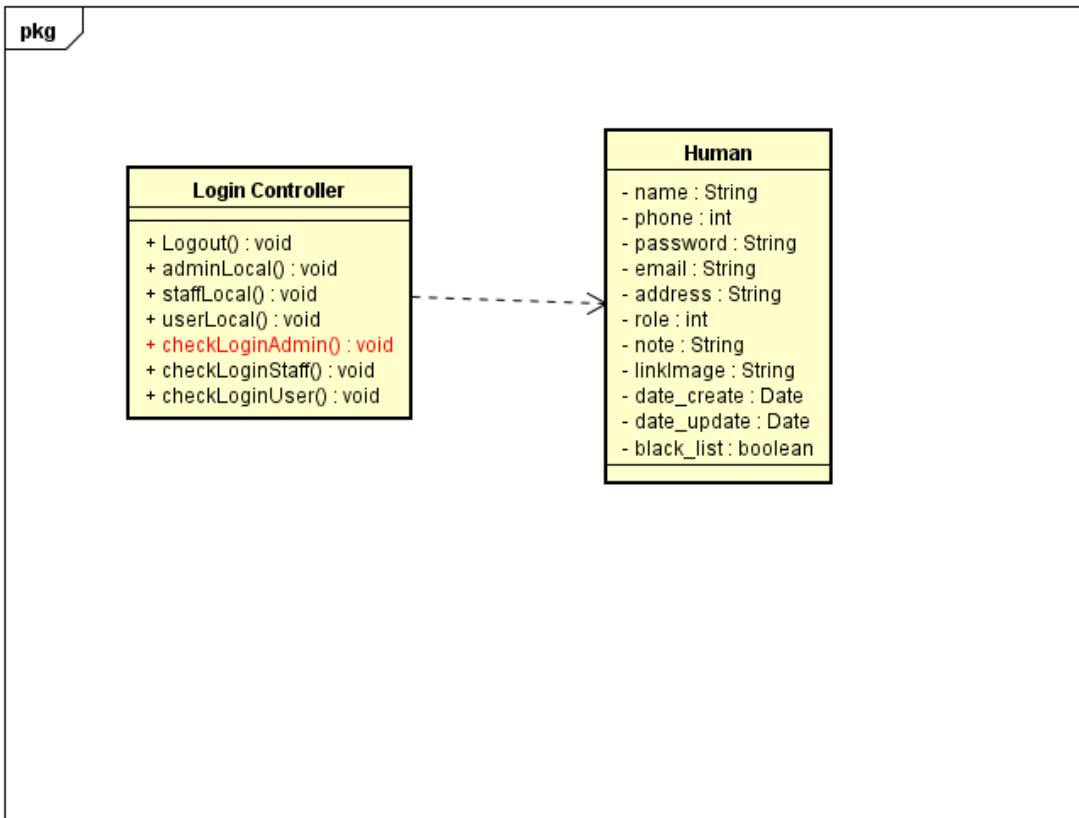


**Figure 4-13:** User Login

<b>Class</b>	LoginController	
<b>Description</b>	User Login	
<b>Base Class</b>	Controller	
<b>Source File</b>	LoginController.js	
<b>Namespace</b>	Controllers	
<b>Attributes</b>	N/A	
<b>Methods</b>	<b>Name</b>	<b>Description</b>
	<code>userLocal()</code>	User normal login to website

**Table 4-23:** User Login

#### 4.3.1.3.4 Check Login Admin



**Figure 4-14:** Check Login Admin

<b>Class</b>	LoginController	
<b>Description</b>	Check Status Login of Admin	
<b>Base Class</b>	Controller	
<b>Source File</b>	LoginController.js	
<b>Namespace</b>	Controllers	
<b>Attributes</b>	N/A	
<b>Methods</b>	<b>Name</b>	<b>Description</b>
	isLoggedAdmin()	Check Status Login of Admin

**Table 4-24:** Check Login Admin

#### 4.3.1.3.5 Check Login Staff

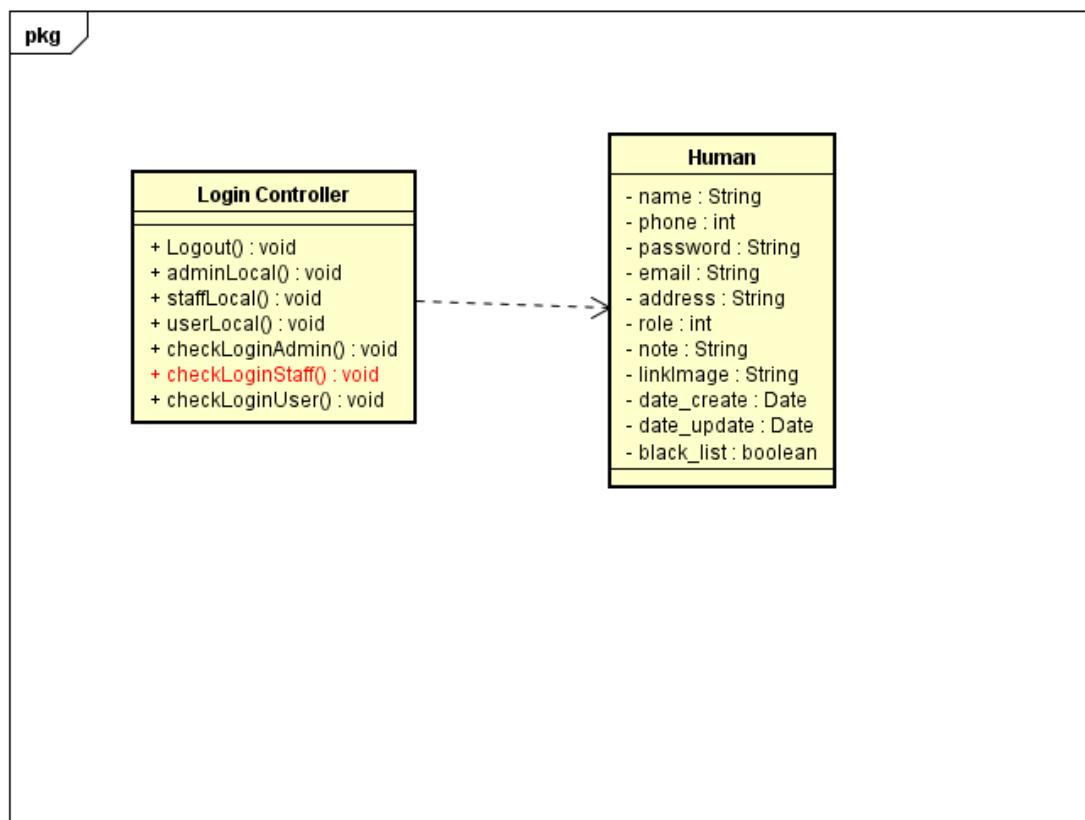


Figure 4-15: Check Login Staff

<b>Class</b>	LoginController	
<b>Description</b>	Check Status Login of Staff	
<b>Base Class</b>	Controller	
<b>Source File</b>	LoginController.js	
<b>Namespace</b>	Controllers	
<b>Attributes</b>	N/A	
<b>Methods</b>	<b>Name</b>	<b>Description</b>
	isLoggedStaff()	Check Status Login of Staff

Table 4-25: Check Login Staff

#### 4.3.1.3.6 Check Login User

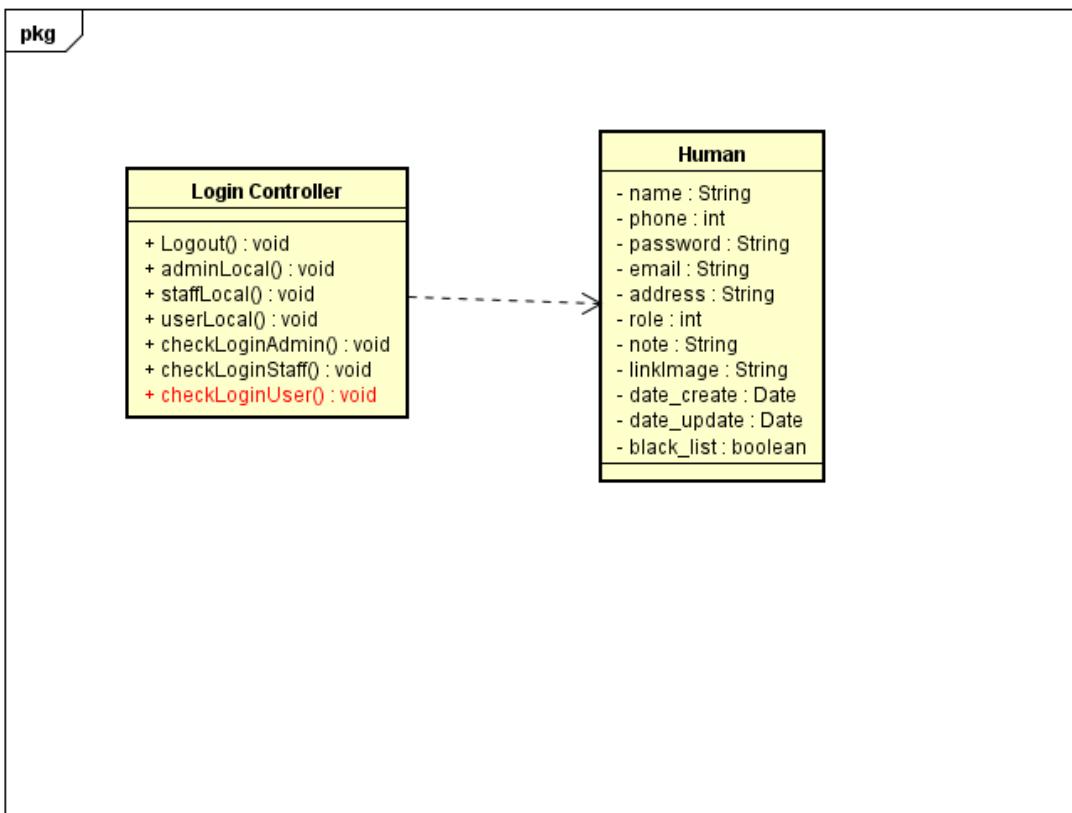
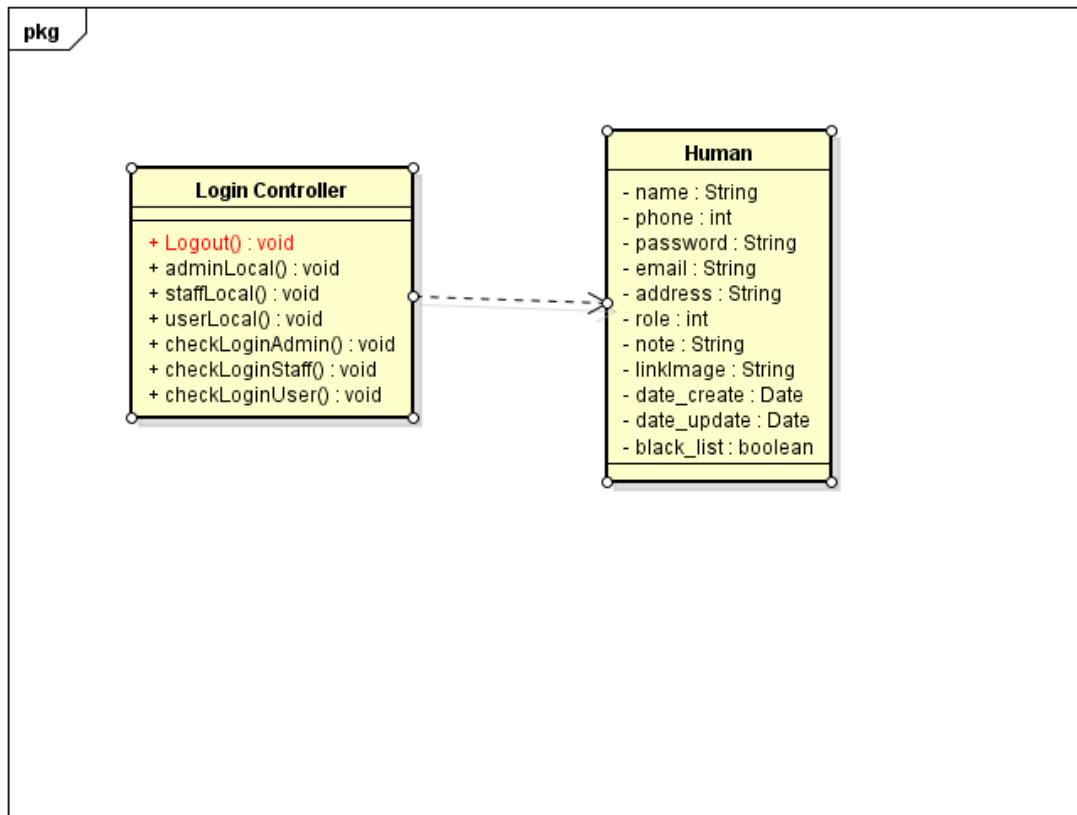


Figure 4-16: Check Login User

<b>Class</b>	LoginController	
<b>Description</b>	Check Status Login of User	
<b>Base Class</b>	Controller	
<b>Source File</b>	LoginController.js	
<b>Namespace</b>	Controllers	
<b>Attributes</b>	N/A	
<b>Methods</b>	<b>Name</b>	<b>Description</b>
	isLoggedUser()	Check Status Login of User

Table 4-26: Check Login User

#### 4.3.1.3.7 Logout

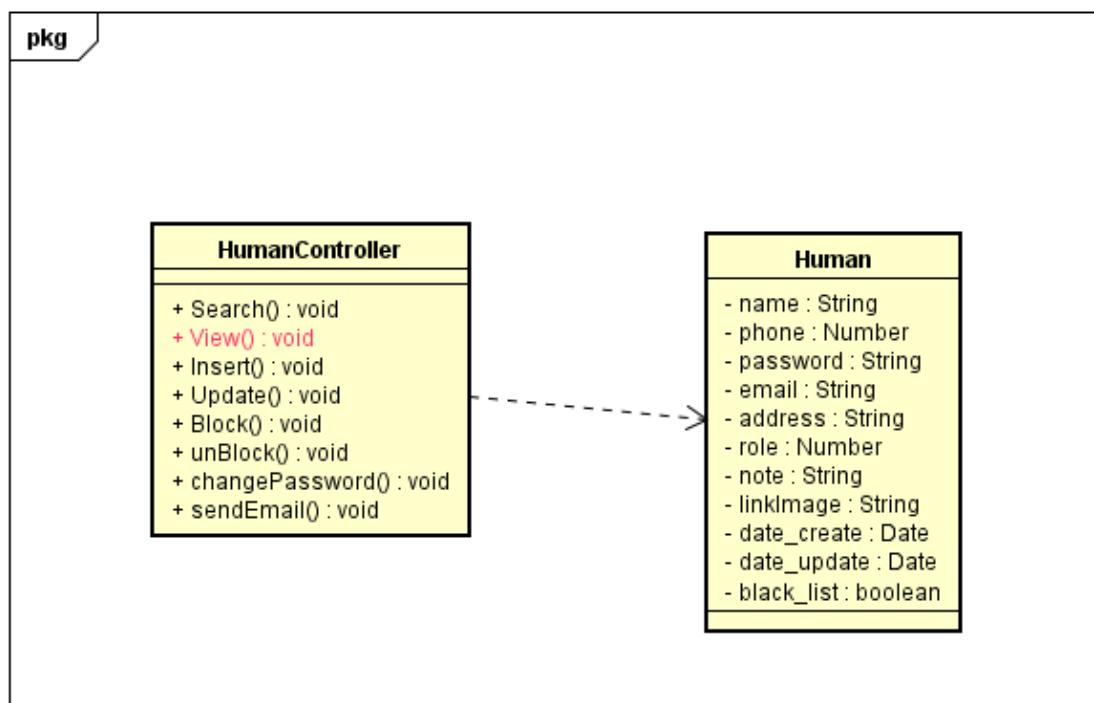


**Figure 4-17:** Logout

<b>Class</b>	LoginController	
<b>Description</b>	Log Out Account	
<b>Base Class</b>	Controller	
<b>Source File</b>	LoginController.js	
<b>Namespace</b>	Controllers	
<b>Attributes</b>	N/A	
<b>Methods</b>	<b>Name</b>	<b>Description</b>
	Logout()	Log out Account

**Table 4-27:** Logout

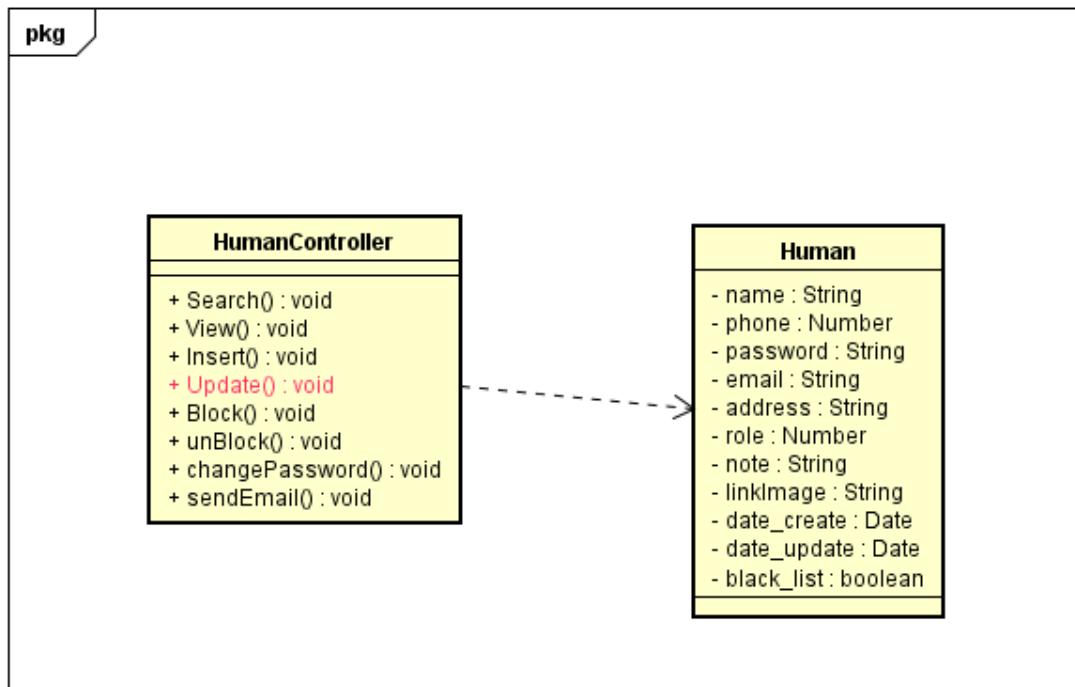
#### 4.3.1.3.8 View Account

**Figure 4-18:** View Account

<b>Class</b>	HumanController					
<b>Description</b>	Show all details of account had exist for admin					
<b>Base Class</b>	Controller					
<b>Source File</b>	HumanController.js					
<b>Namespace</b>	Controllers					
<b>Attributes</b>	N/A					
<b>Methods</b>	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>View()</td> <td>Show all details of account had exist for Admin</td> </tr> </tbody> </table>	Name	Description	View()	Show all details of account had exist for Admin	
Name	Description					
View()	Show all details of account had exist for Admin					

**Table 4-28:** View Account

#### 4.3.1.3.9 Update Account

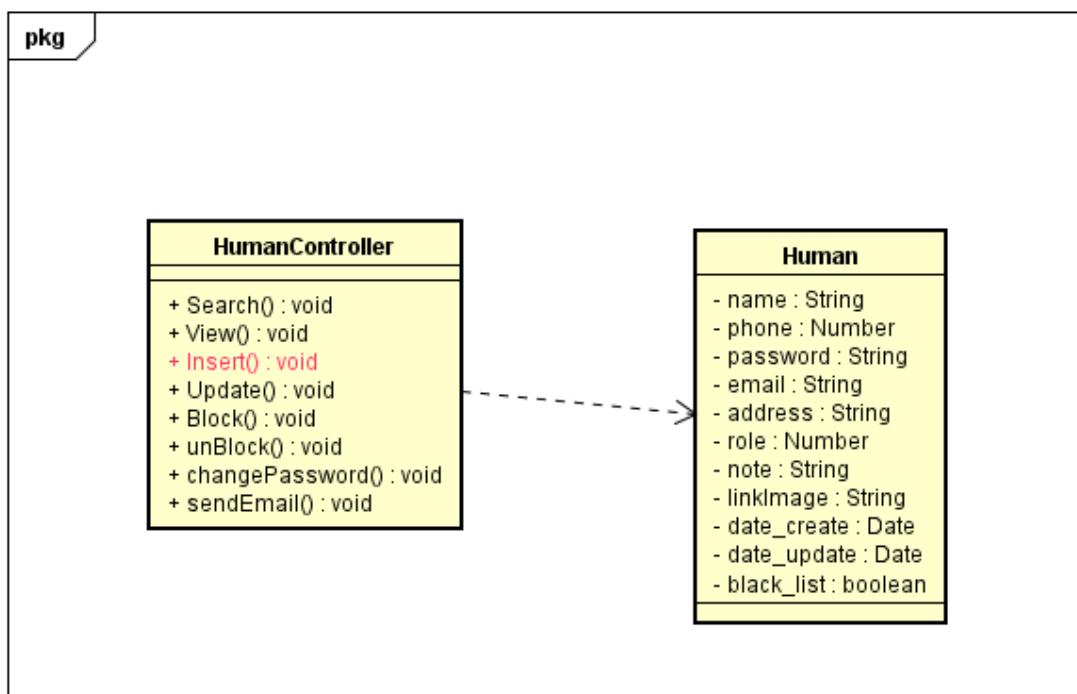


**Figure 4-19:** Update Account

<b>Class</b>	HumanController					
<b>Description</b>	update information for account					
<b>Base Class</b>	Controller					
<b>Source File</b>	HumanController.js					
<b>Namespace</b>	Controllers					
<b>Attributes</b>	N/A					
<b>Methods</b>	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>Update()</td> <td>Update information for account</td> </tr> </tbody> </table>	Name	Description	Update()	Update information for account	
Name	Description					
Update()	Update information for account					

**Table 4-29:** Update Account

#### 4.3.1.3.10 Insert Account



**Figure 4-20:** Insert Account

<b>Class</b>	HumanController	
<b>Description</b>	Insert new account	
<b>Base Class</b>	Controller	
<b>Source File</b>	HumanController.js	
<b>Namespace</b>	Controllers	
<b>Attributes</b>	N/A	
<b>Methods</b>	<b>Name</b>	<b>Description</b>
	Insert()	User sign up or admin add new a Staff

**Table 4-30:** Insert Account

#### 4.3.1.3.11 Search Human

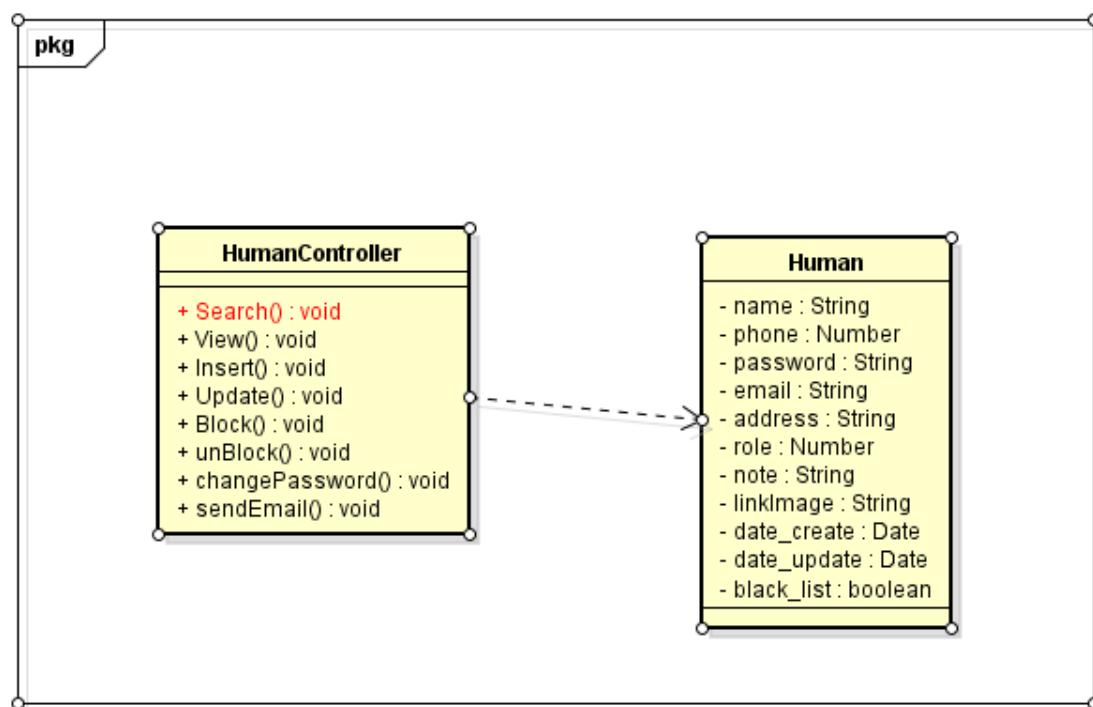


Figure 4-21: Search Human

<b>Class</b>	HumanController	
<b>Description</b>	Search information of Account	
<b>Base Class</b>	Controller	
<b>Source File</b>	HumanController.js	
<b>Namespace</b>	Controllers	
<b>Attributes</b>	N/A	
<b>Methods</b>	<b>Name</b>	<b>Description</b>
	Search()	Search information of Account

Table 4-31: Search Human

#### 4.3.1.3.12 Change Password

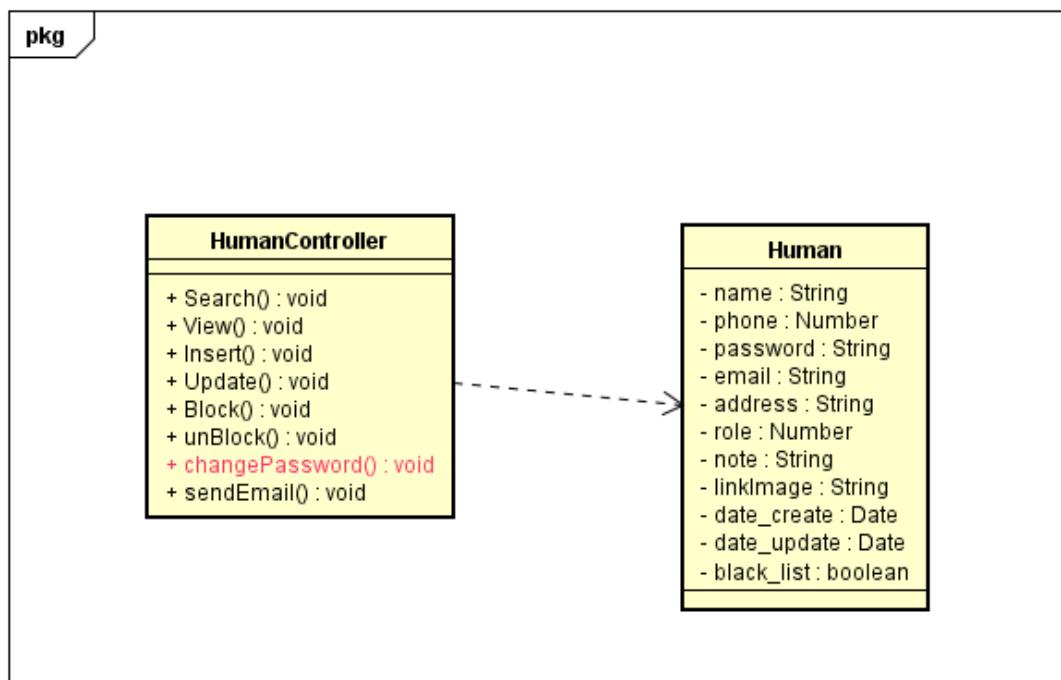
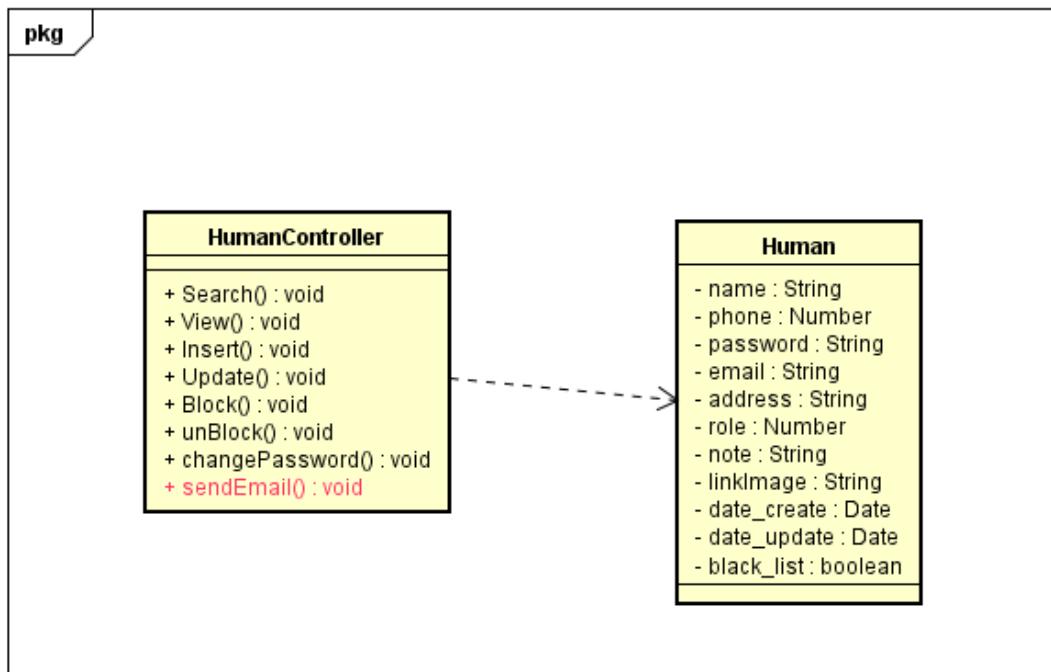


Figure 4-22: Change Password

<b>Class</b>	HumanController	
<b>Description</b>	Change Password of Account	
<b>Base Class</b>	Controller	
<b>Source File</b>	HumanController.js	
<b>Namespace</b>	Controllers	
<b>Attributes</b>	N/A	
<b>Methods</b>	<b>Name</b>	<b>Description</b>
	changePassword()	Change Password of Account

Table 4-32: Change Password

#### 4.3.1.3.13 Forgot Password

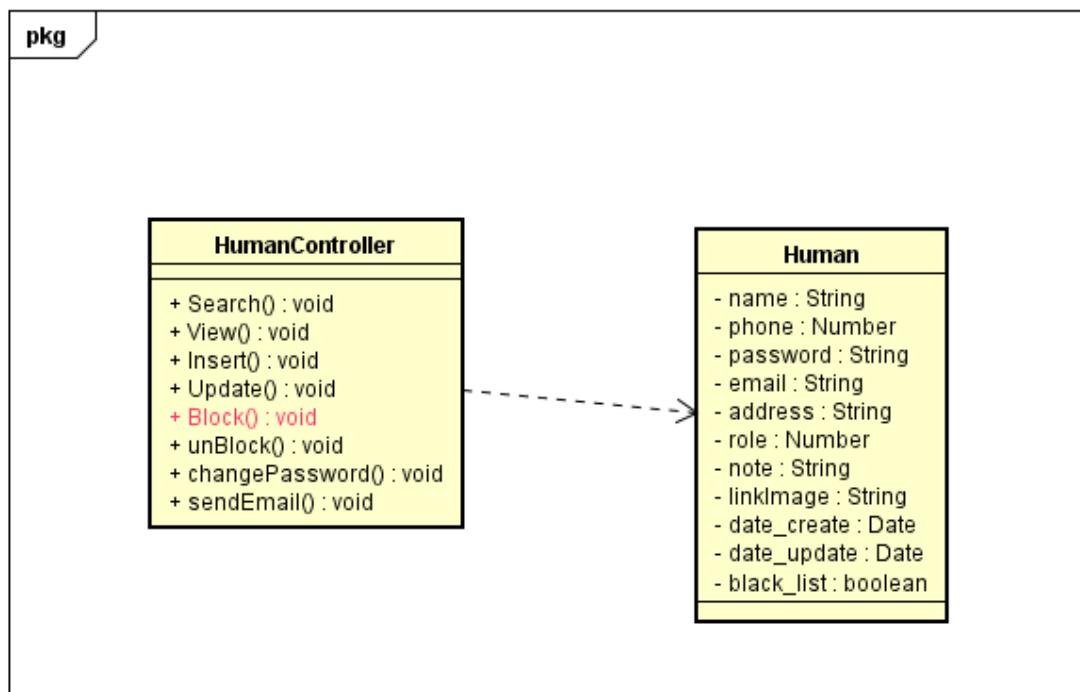


**Figure 4-23: Forgot Password**

<b>Class</b>	HumanController					
<b>Description</b>	Send new Password to Email					
<b>Base Class</b>	Controller					
<b>Source File</b>	HumanController.js					
<b>Namespace</b>	Controllers					
<b>Attributes</b>	N/A					
<b>Methods</b>	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>sendEmail()</td> <td>Send new password to email when forgot password</td> </tr> </tbody> </table>	Name	Description	sendEmail()	Send new password to email when forgot password	
Name	Description					
sendEmail()	Send new password to email when forgot password					

**Table 4-33: Forgot Password**

#### 4.3.1.3.14 Lock Account

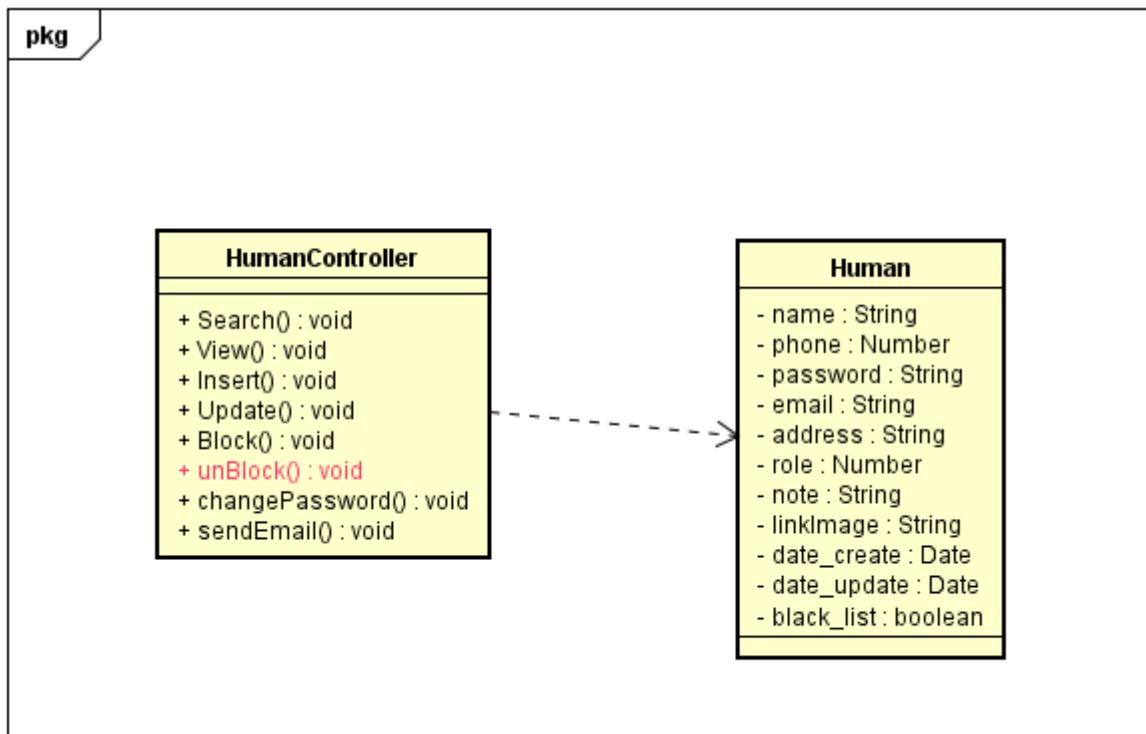


**Figure 4-24:** Lock Account

<b>Class</b>	HumanController					
<b>Description</b>	Admin can lock account of customer and staff					
<b>Base Class</b>	Controller					
<b>Source File</b>	HumanController.js					
<b>Namespace</b>	Controllers					
<b>Attributes</b>	N/A					
<b>Methods</b>	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>Block()</td> <td>Admin can lock account of customer and staff</td> </tr> </tbody> </table>	Name	Description	Block()	Admin can lock account of customer and staff	
Name	Description					
Block()	Admin can lock account of customer and staff					

**Table 4-34:** Lock Account

#### 4.3.1.3.15 Unlock Account

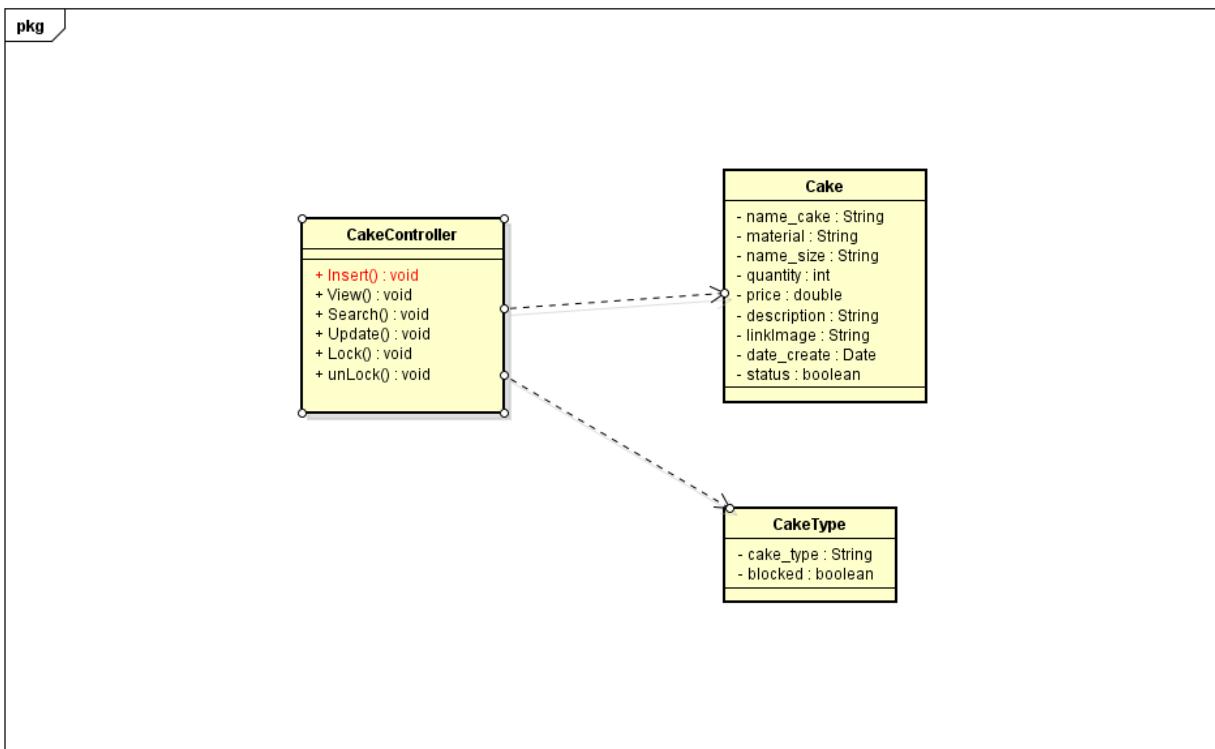


**Figure 4-25: Unlock Account**

<b>Class</b>	HumanController					
<b>Description</b>	Admin can unlock account of customer and staff					
<b>Base Class</b>	Controller					
<b>Source File</b>	HumanController.js					
<b>Namespace</b>	Controllers					
<b>Attributes</b>	N/A					
<b>Methods</b>	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>unBlock()</td> <td>Admin can lock account of customer and staff</td> </tr> </tbody> </table>	Name	Description	unBlock()	Admin can lock account of customer and staff	
Name	Description					
unBlock()	Admin can lock account of customer and staff					

**Table 4-35: Unlock Account**

#### 4.3.1.3.16 Add Cake

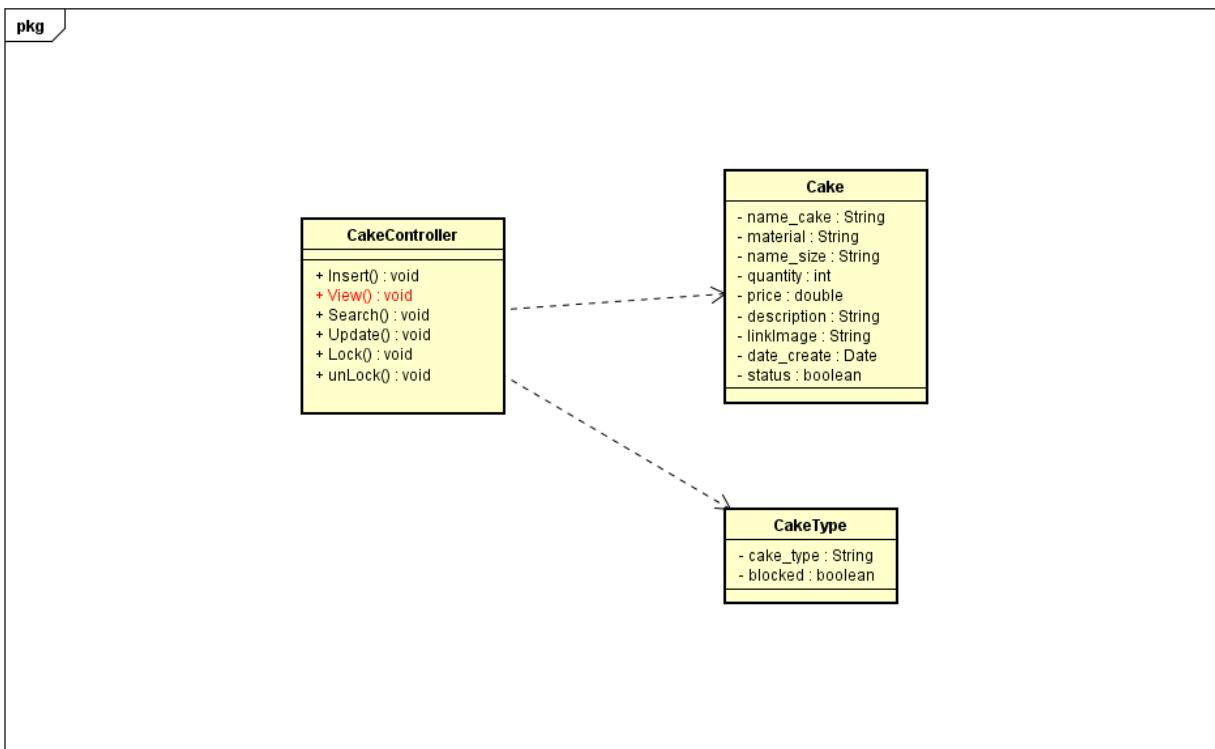


**Figure 4-26: Add Cake**

<b>Class</b>	CakeController					
<b>Description</b>	Add new Cake					
<b>Base Class</b>	Controller					
<b>Source File</b>	CakeController.js					
<b>Namespace</b>	Controllers					
<b>Attributes</b>	N/A					
<b>Methods</b>	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>Insert()</td> <td>Add new cake</td> </tr> </tbody> </table>	Name	Description	Insert()	Add new cake	
Name	Description					
Insert()	Add new cake					

**Table 4-36: Add Cake**

#### 4.3.1.3.17 View Cake



**Figure 4-27:** View Cake

<b>Class</b>	CakeController					
<b>Description</b>	View information of Cakes					
<b>Base Class</b>	Controller					
<b>Source File</b>	CakeController.js					
<b>Namespace</b>	Controllers					
<b>Attributes</b>	N/A					
<b>Methods</b>	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>Insert()</td> <td>View information of Cake</td> </tr> </tbody> </table>	Name	Description	Insert()	View information of Cake	
Name	Description					
Insert()	View information of Cake					

**Table 4-37:** View Cake

#### 4.3.1.3.18 Search Cake

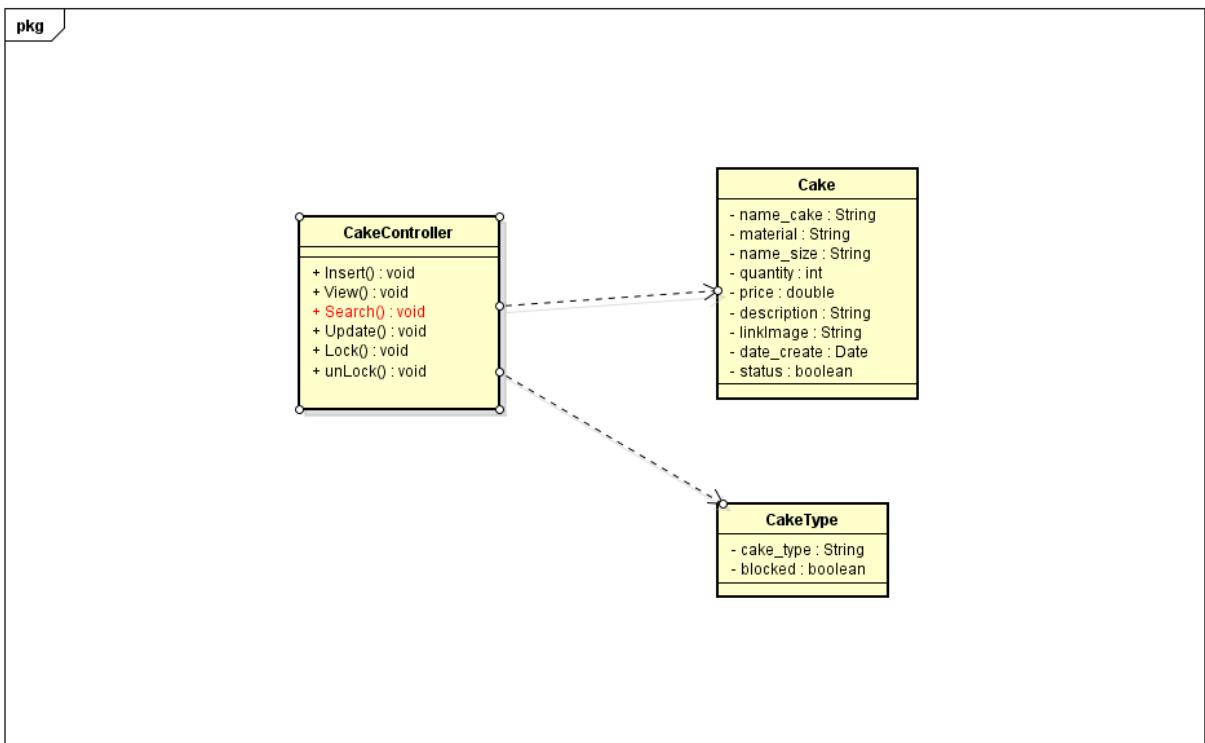
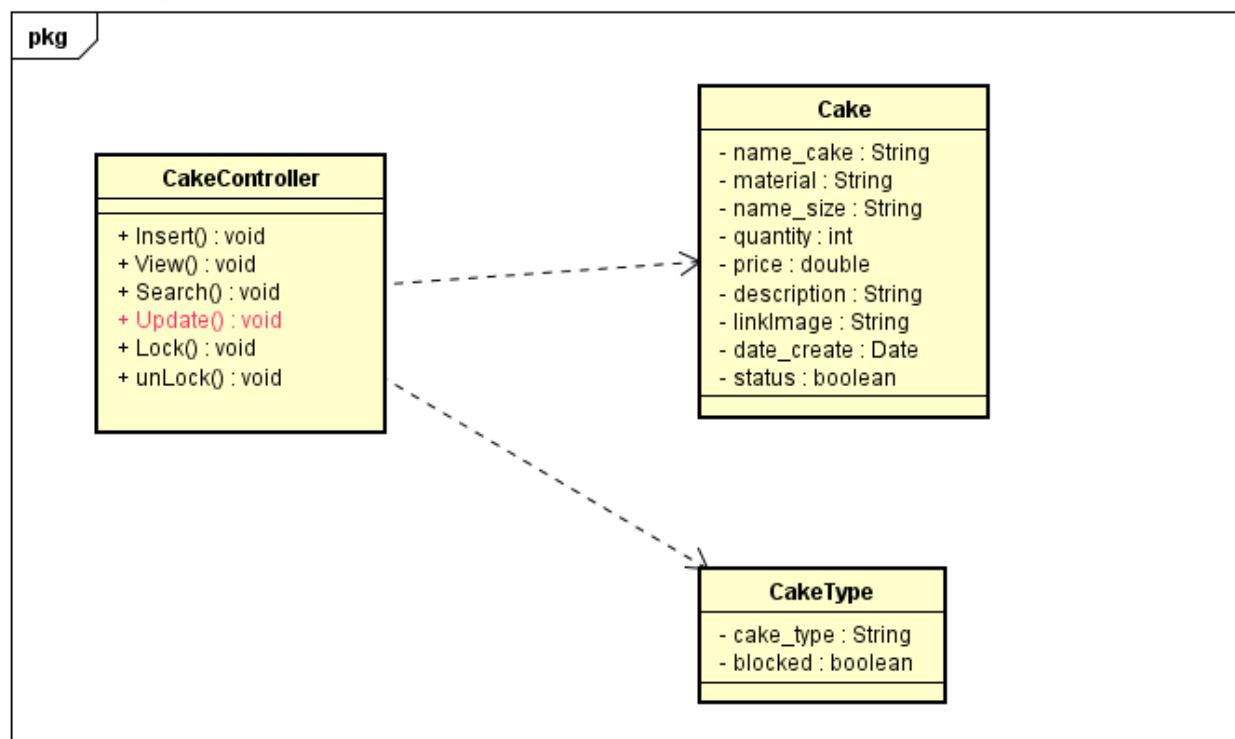


Figure 4-28: Search Cake

<b>Class</b>	CakeController					
<b>Description</b>	Search cake by name or category					
<b>Base Class</b>	Controller					
<b>Source File</b>	CakeController.js					
<b>Namespace</b>	Controllers					
<b>Attributes</b>	N/A					
<b>Methods</b>	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>Search()</td> <td>Search cake by name or category</td> </tr> </tbody> </table>	Name	Description	Search()	Search cake by name or category	
Name	Description					
Search()	Search cake by name or category					

Table 4-38: Search Cake

#### 4.3.1.3.19 Update Cake

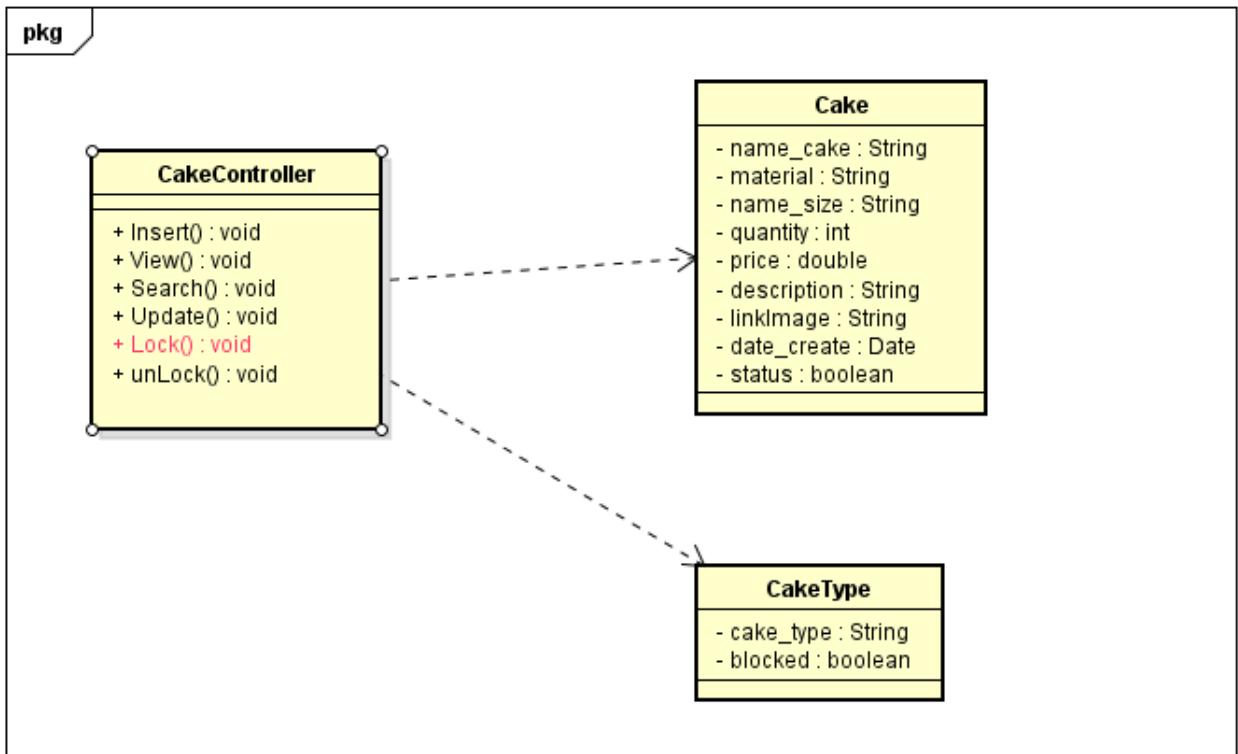


**Figure 4-29:** Update Cake

<b>Class</b>	CakeController	
<b>Description</b>	Update information cake	
<b>Base Class</b>	Controller	
<b>Source File</b>	CakeController.js	
<b>Namespace</b>	Controllers	
<b>Attributes</b>	N/A	
<b>Methods</b>	<b>Name</b>	<b>Description</b>
	Update()	Update information cake

**Table 4-39:** Update Cake

#### 4.3.1.3.20 Lock Cake



**Figure 4-30: Lock Cake**

<b>Class</b>	CakeController					
<b>Description</b>	Update Status of cake not sell					
<b>Base Class</b>	Controller					
<b>Source File</b>	CakeController.js					
<b>Namespace</b>	Controllers					
<b>Attributes</b>	N/A					
<b>Methods</b>	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>Lock()</td> <td>Update Status cake not sell</td> </tr> </tbody> </table>	Name	Description	Lock()	Update Status cake not sell	
Name	Description					
Lock()	Update Status cake not sell					

**Table 4-40: Lock Cake**

#### 4.3.1.3.21 Unlock cake

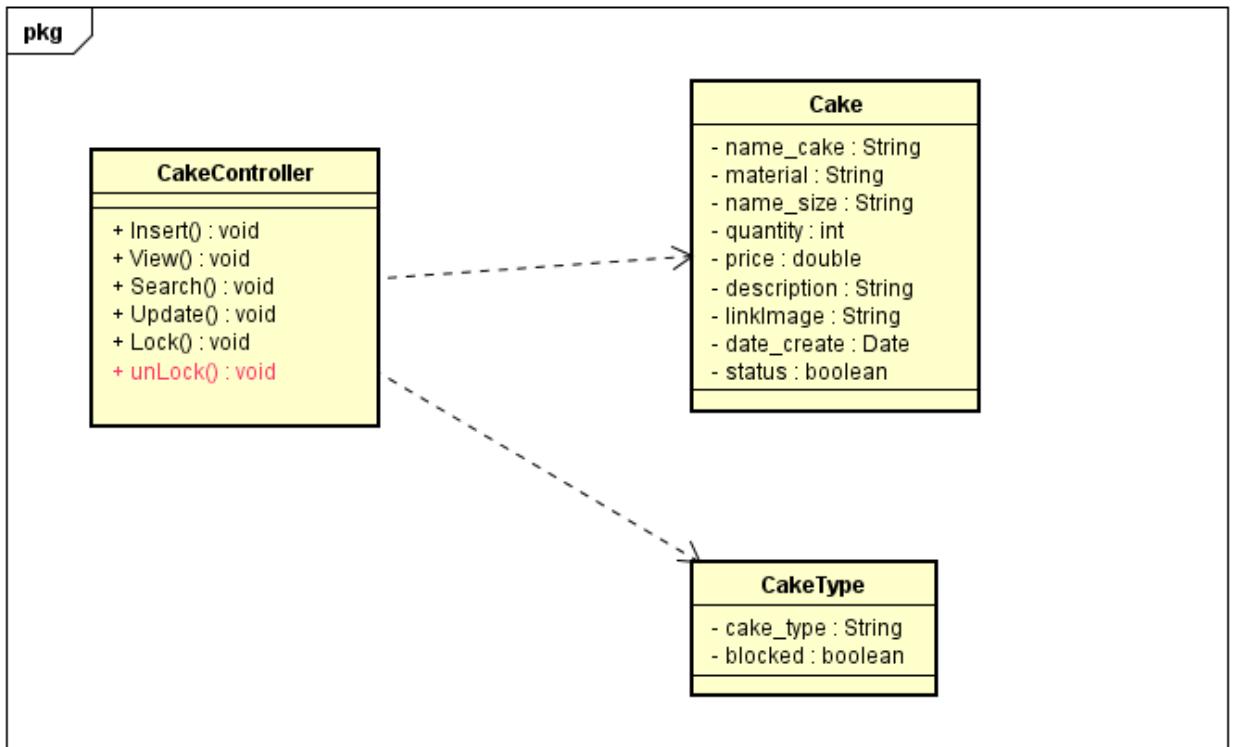
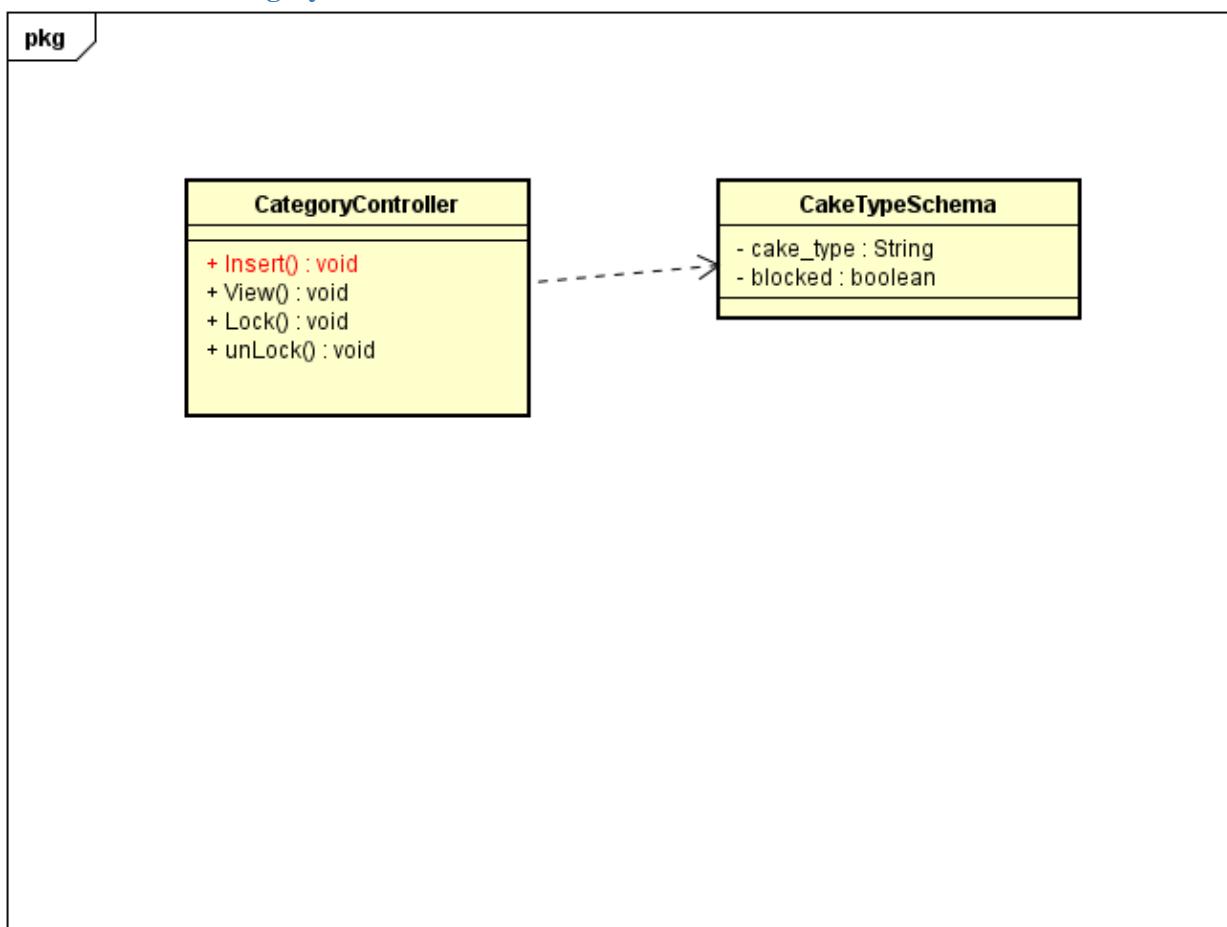


Figure 4-31: Unlock cake

<b>Class</b>	CakeController					
<b>Description</b>	Update status of cake (sell again )					
<b>Base Class</b>	Controller					
<b>Source File</b>	CakeController.js					
<b>Namespace</b>	Controllers					
<b>Attributes</b>	N/A					
<b>Methods</b>	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>unLock()</td> <td>Update status of cake (sell again )</td> </tr> </tbody> </table>	Name	Description	unLock()	Update status of cake (sell again )	
Name	Description					
unLock()	Update status of cake (sell again )					

Table 4-41: Unlock cake

#### 4.3.1.3.22 Insert category



**Figure 4-32:** Insert category

<b>Class</b>	CategoryController					
<b>Description</b>	Insert new category					
<b>Base Class</b>	Controller					
<b>Source File</b>	CategoryController.js					
<b>Namespace</b>	Controllers					
<b>Attributes</b>	N/A					
<b>Methods</b>	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>Insert()</td> <td>Insert new category</td> </tr> </tbody> </table>	Name	Description	Insert()	Insert new category	
Name	Description					
Insert()	Insert new category					

**Table 4-42:** Insert category

#### 4.3.1.3.23 View Category

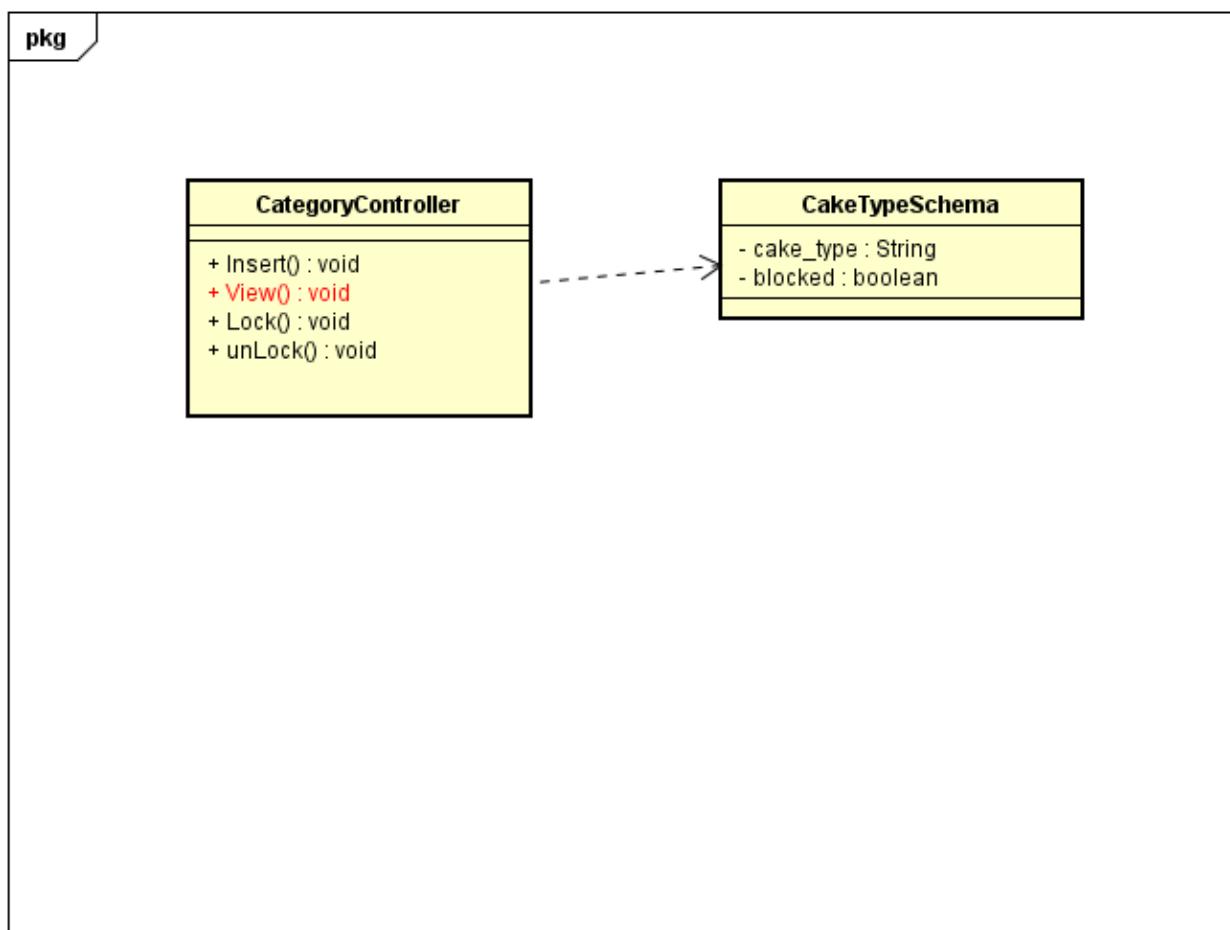
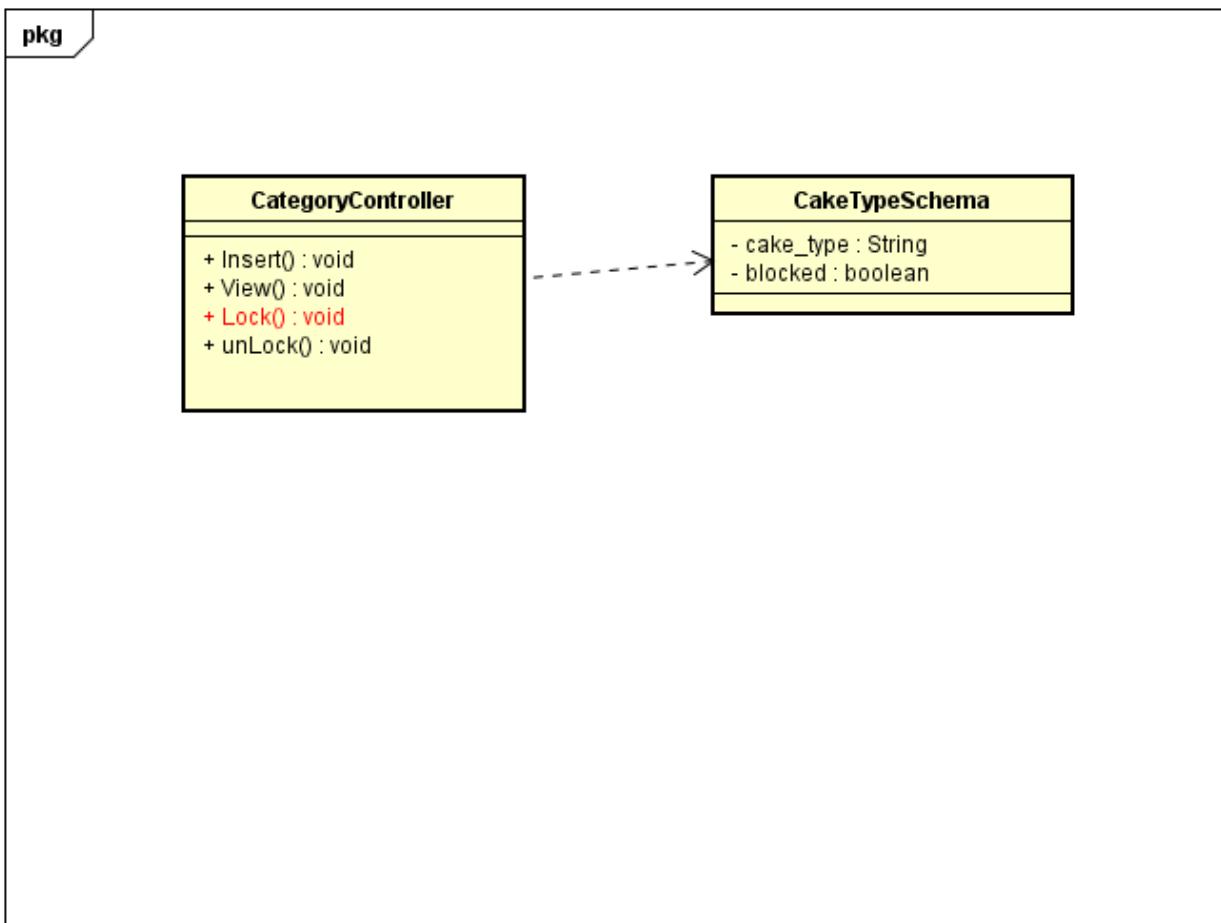


Figure 4-33: View Category

<b>Class</b>	CategoryController					
<b>Description</b>	View all category					
<b>Base Class</b>	Controller					
<b>Source File</b>	CategoryController.js					
<b>Namespace</b>	Controllers					
<b>Attributes</b>	N/A					
<b>Methods</b>	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>View()</td> <td>View all category</td> </tr> </tbody> </table>	Name	Description	View()	View all category	
Name	Description					
View()	View all category					

Table 4-43: View Category

#### 4.3.1.3.24 Lock Category

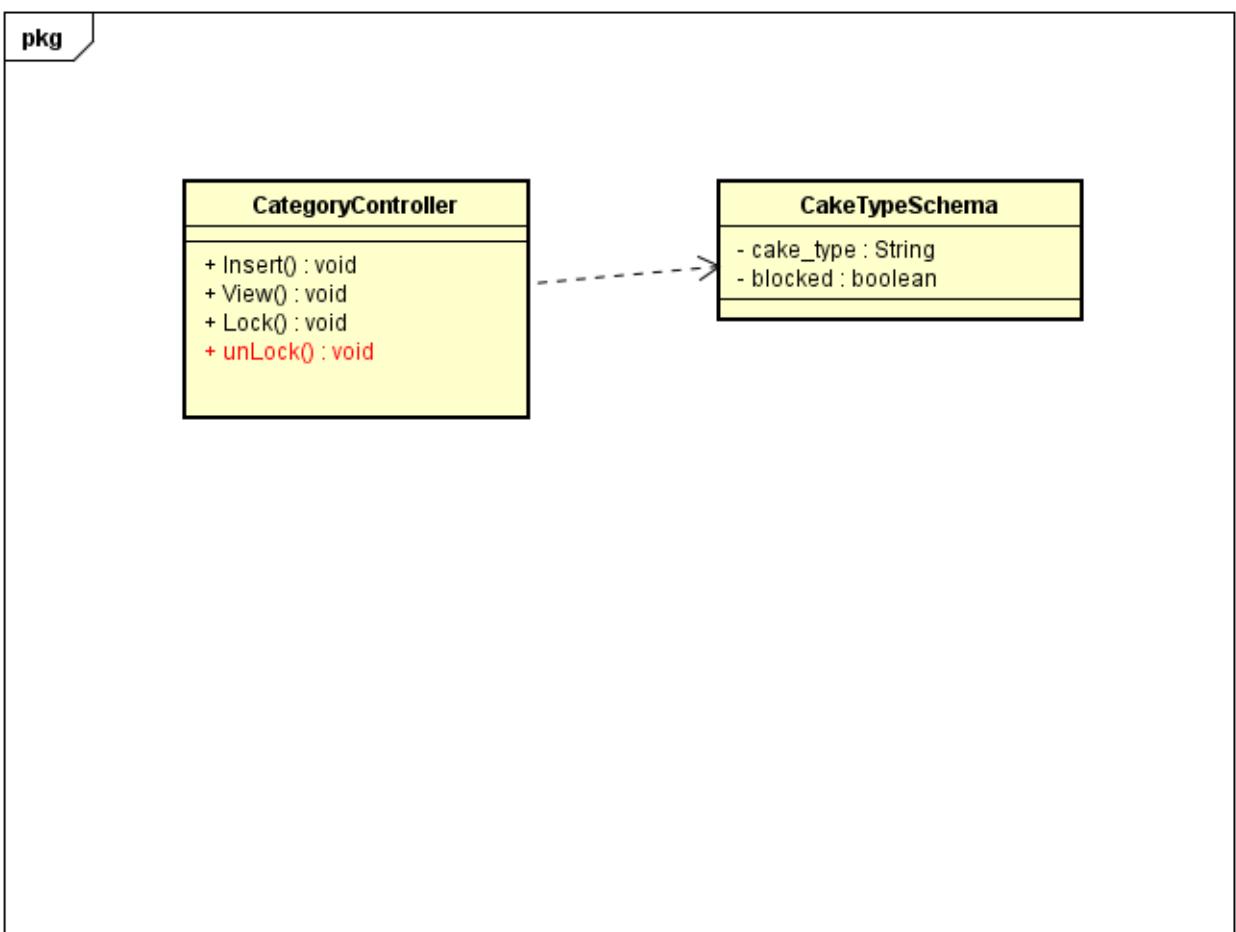


**Figure 4-34:** Lock Category

<b>Class</b>	CategoryController					
<b>Description</b>	Update status of category					
<b>Base Class</b>	Controller					
<b>Source File</b>	CategoryController.js					
<b>Namespace</b>	Controllers					
<b>Attributes</b>	N/A					
<b>Methods</b>	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>Lock()</td> <td>Update status of category</td> </tr> </tbody> </table>	Name	Description	Lock()	Update status of category	
Name	Description					
Lock()	Update status of category					

**Table 4-44:** Lock Category

#### 4.3.1.3.25 Unlock Category

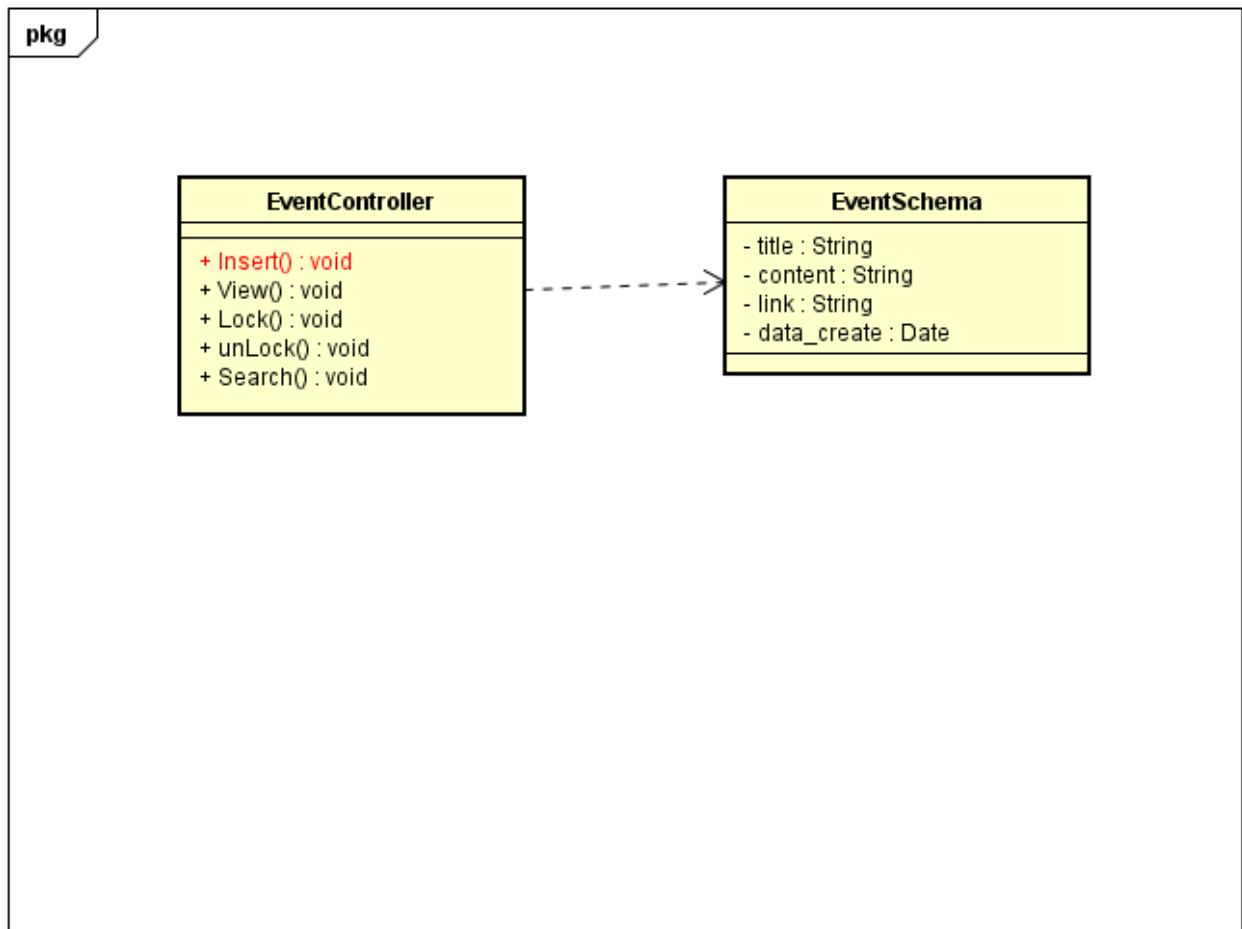


**Figure 4-35:** Unlock Category

<b>Class</b>	CategoryController	
<b>Description</b>	Update status of category	
<b>Base Class</b>	Controller	
<b>Source File</b>	CategoryController.js	
<b>Namespace</b>	Controllers	
<b>Attributes</b>	N/A	
<b>Methods</b>	<b>Name</b>	<b>Description</b>
	unLock()	Update status of category

**Table 4-45:** Unlock Category

#### 4.3.1.3.26 Insert Event



**Figure 4-36:** Insert Event

<b>Class</b>	EventController	
<b>Description</b>	Insert new event	
<b>Base Class</b>	Controller	
<b>Source File</b>	EventController.js	
<b>Namespace</b>	Controllers	
<b>Attributes</b>	N/A	
<b>Methods</b>	<b>Name</b>	<b>Description</b>
	Insert()	Insert new event

**Table 4-46:** Insert Event

#### 4.3.1.3.27 View Event

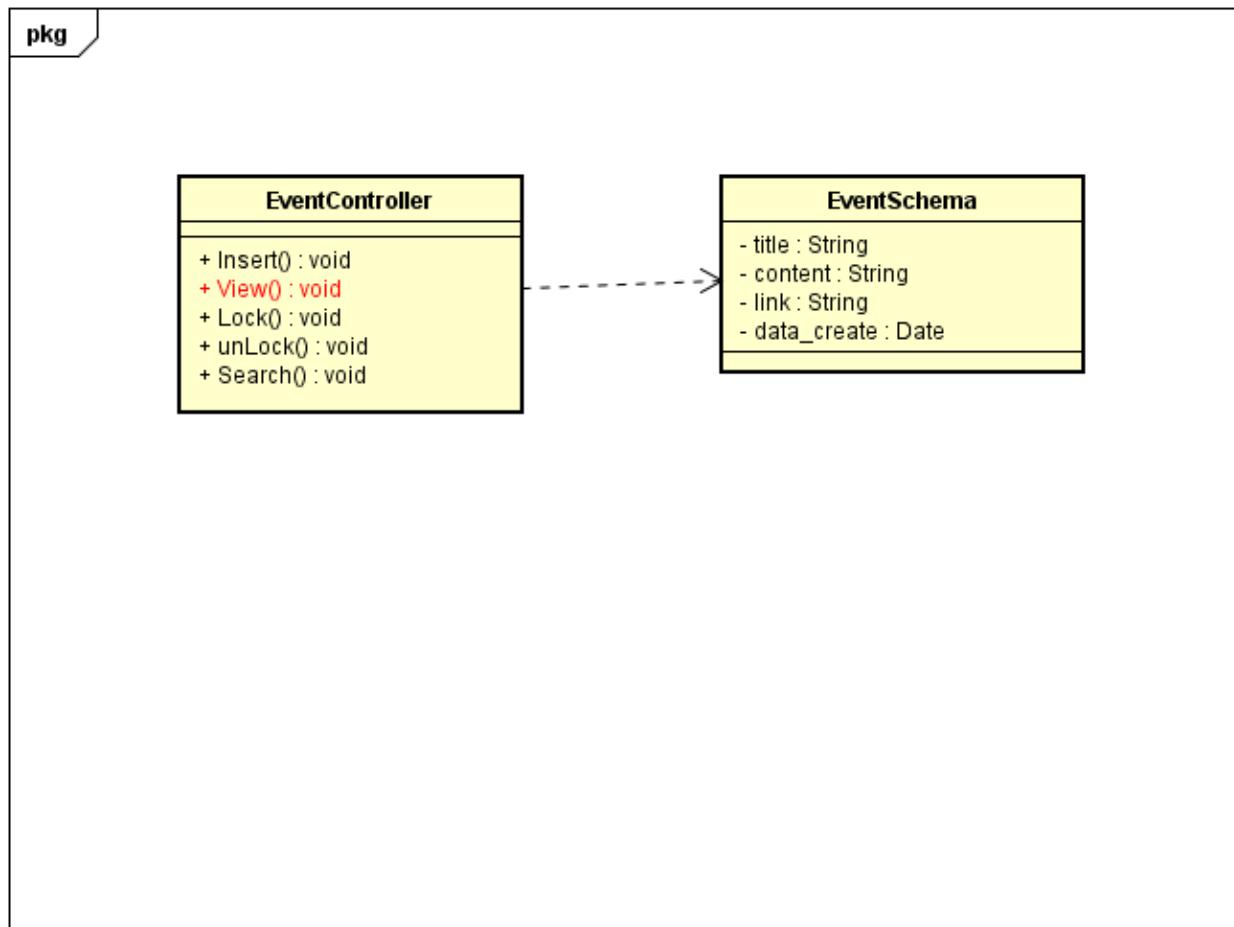
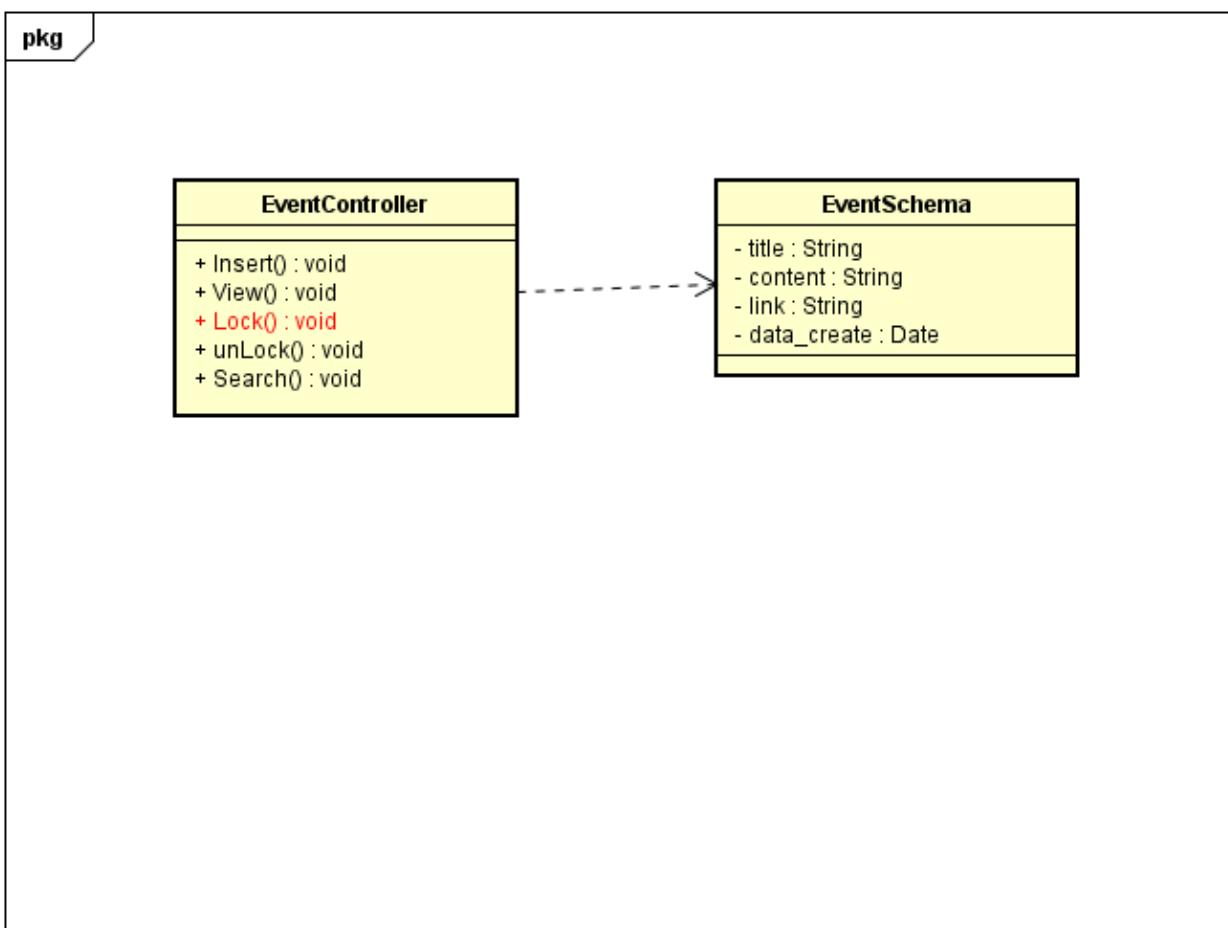


Figure 4-37: View Event

<b>Class</b>	EventController	
<b>Description</b>	View all Event	
<b>Base Class</b>	Controller	
<b>Source File</b>	EventController.js	
<b>Namespace</b>	Controllers	
<b>Attributes</b>	N/A	
<b>Methods</b>	<b>Name</b>	<b>Description</b>
	View()	View all event exist

Table 4-47: View Event

#### 4.3.1.3.28 Lock Event

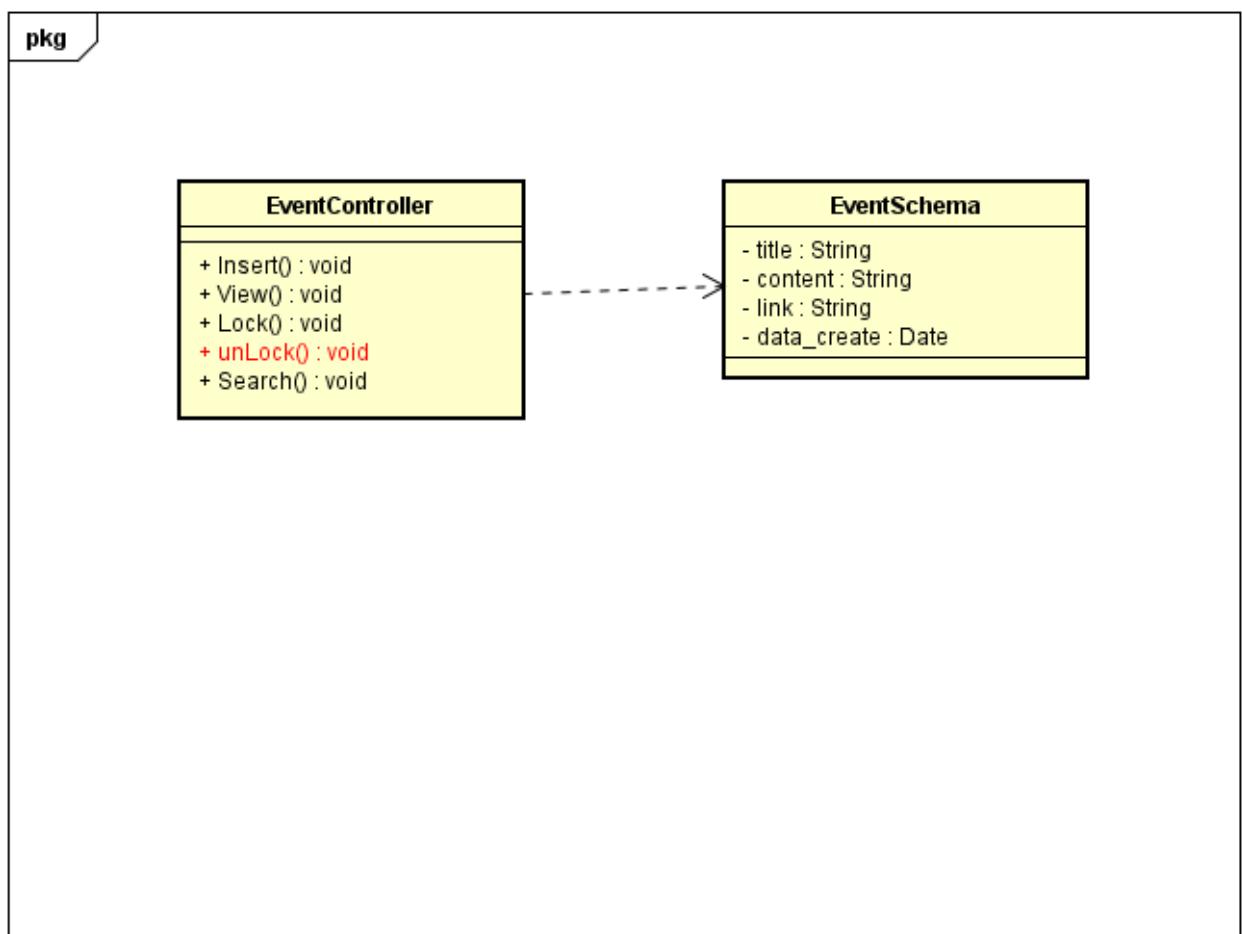


**Figure 4-38: Lock Event**

<b>Class</b>	EventController	
<b>Description</b>	Update status of event (close)	
<b>Base Class</b>	Controller	
<b>Source File</b>	EventController.js	
<b>Namespace</b>	Controllers	
<b>Attributes</b>	N/A	
<b>Methods</b>	<b>Name</b>	<b>Description</b>
	Lock()	Update status of event (close)

**Table 4-48: Lock Event**

#### 4.3.1.3.29 Unlock Event

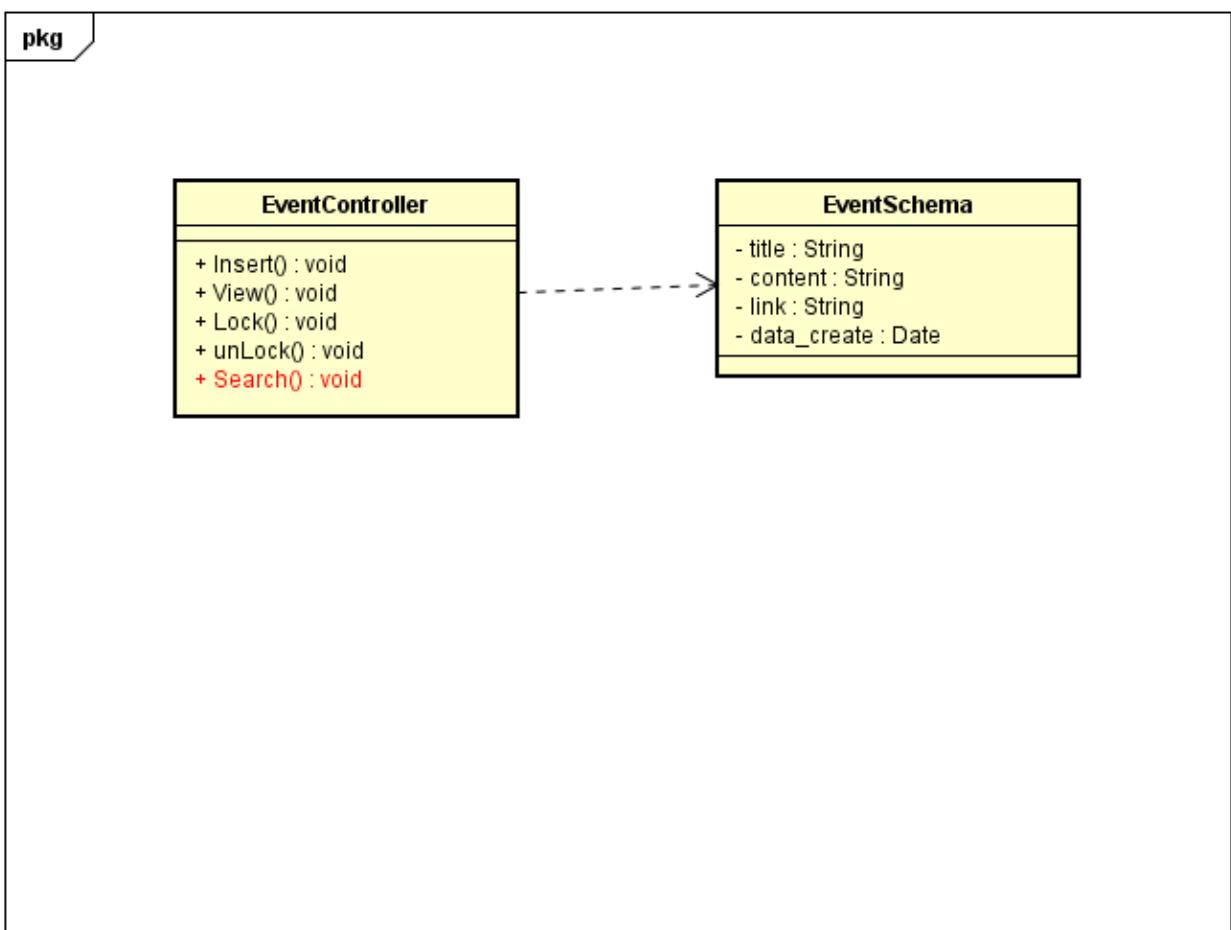


**Figure 4-39:** Unlock Event

<b>Class</b>	EventController	
<b>Description</b>	Update status of event (open)	
<b>Base Class</b>	Controller	
<b>Source File</b>	EventController.js	
<b>Namespace</b>	Controllers	
<b>Attributes</b>	N/A	
<b>Methods</b>	<b>Name</b> Name Unlock()	<b>Description</b> Update status of event (open)

**Table 4-49:** Unlock Event

#### 4.3.1.3.30 Search Event



**Figure 4-40:** Search Event

<b>Class</b>	EventController					
<b>Description</b>	Find information an event					
<b>Base Class</b>	Controller					
<b>Source File</b>	EventController.js					
<b>Namespace</b>	Controllers					
<b>Attributes</b>	N/A					
<b>Methods</b>	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>Search()</td> <td>Find information an event</td> </tr> </tbody> </table>	Name	Description	Search()	Find information an event	
Name	Description					
Search()	Find information an event					

**Table 4-50:** Search Event

#### 4.3.1.3.31 Insert Blog

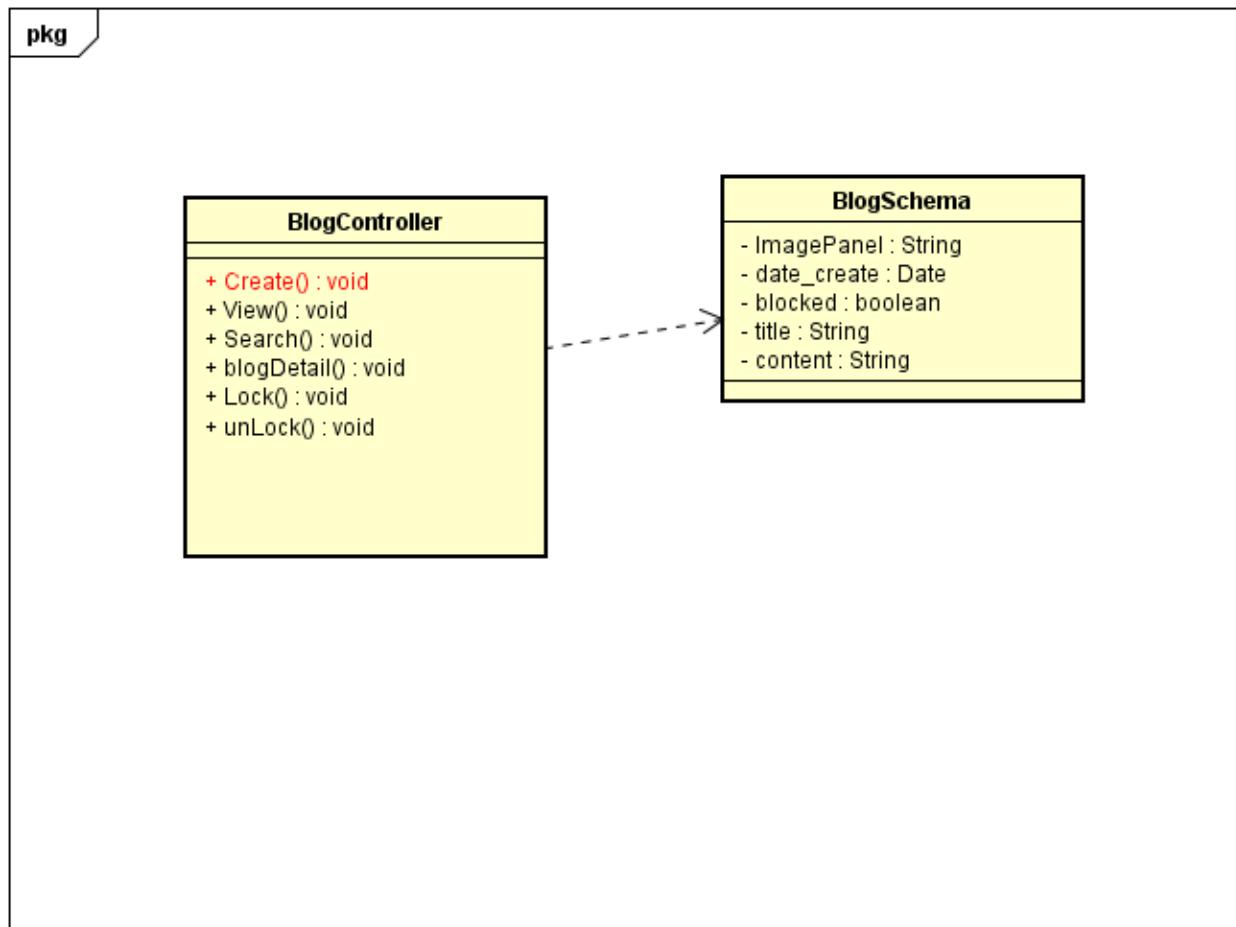
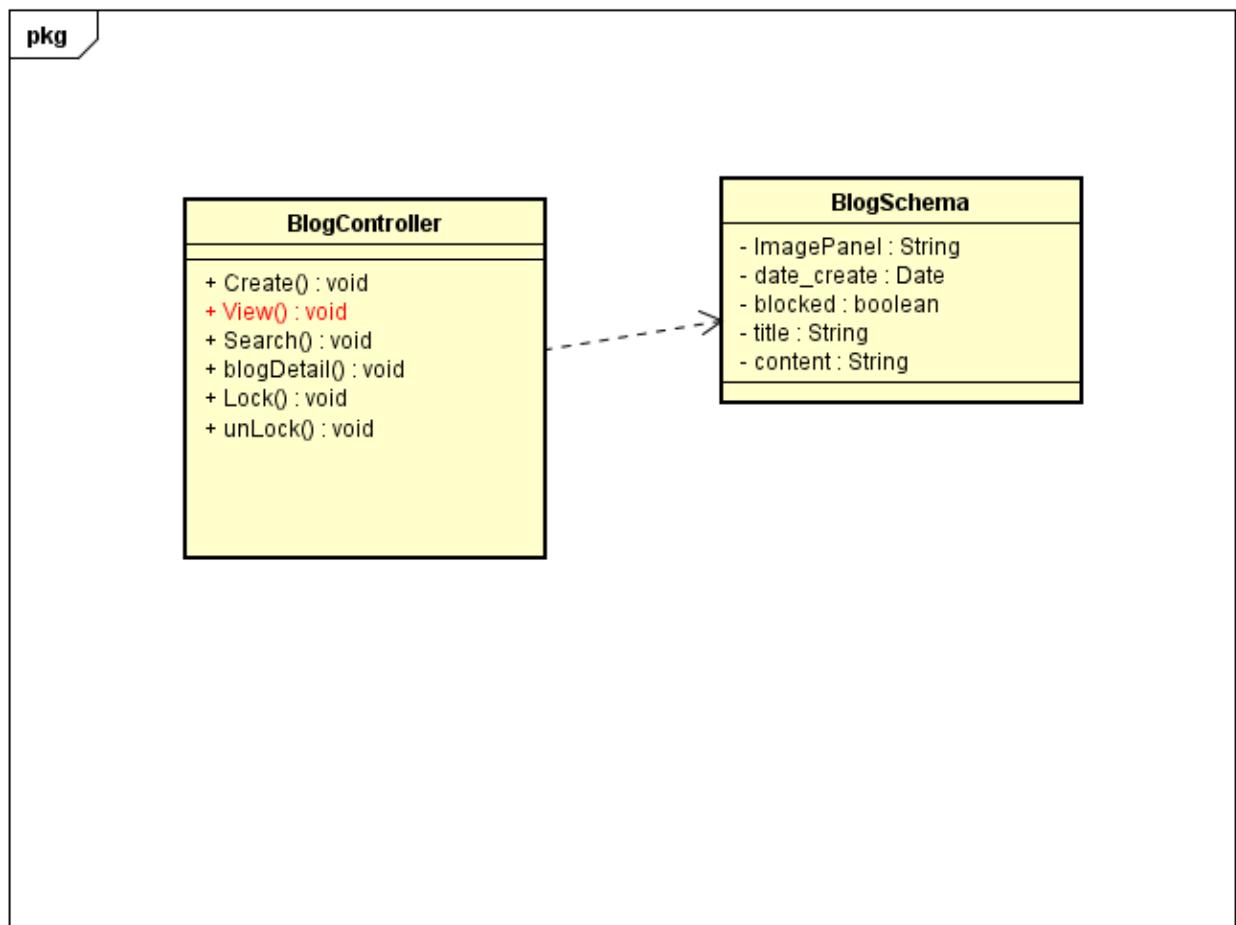


Figure 4-41: Insert Blog

<b>Class</b>	BlogController	
<b>Description</b>	Insert new blog	
<b>Base Class</b>	Controller	
<b>Source File</b>	BlogController.js	
<b>Namespace</b>	Controllers	
<b>Attributes</b>	N/A	
<b>Methods</b>	<b>Name</b> Create()	<b>Description</b> Insert new blog

Table 4-51: Insert Blog

#### 4.3.1.3.32 View Blog



**Figure 4-42:** View Blog

<b>Class</b>	BlogController	
<b>Description</b>	View all blog	
<b>Base Class</b>	Controller	
<b>Source File</b>	BlogController.js	
<b>Namespace</b>	Controllers	
<b>Attributes</b>	N/A	
<b>Methods</b>	<b>Name</b>	<b>Description</b>
	View()	View all blog

**Table 4-52:** View Blog

### 4.3.1.3.33 Search Blog

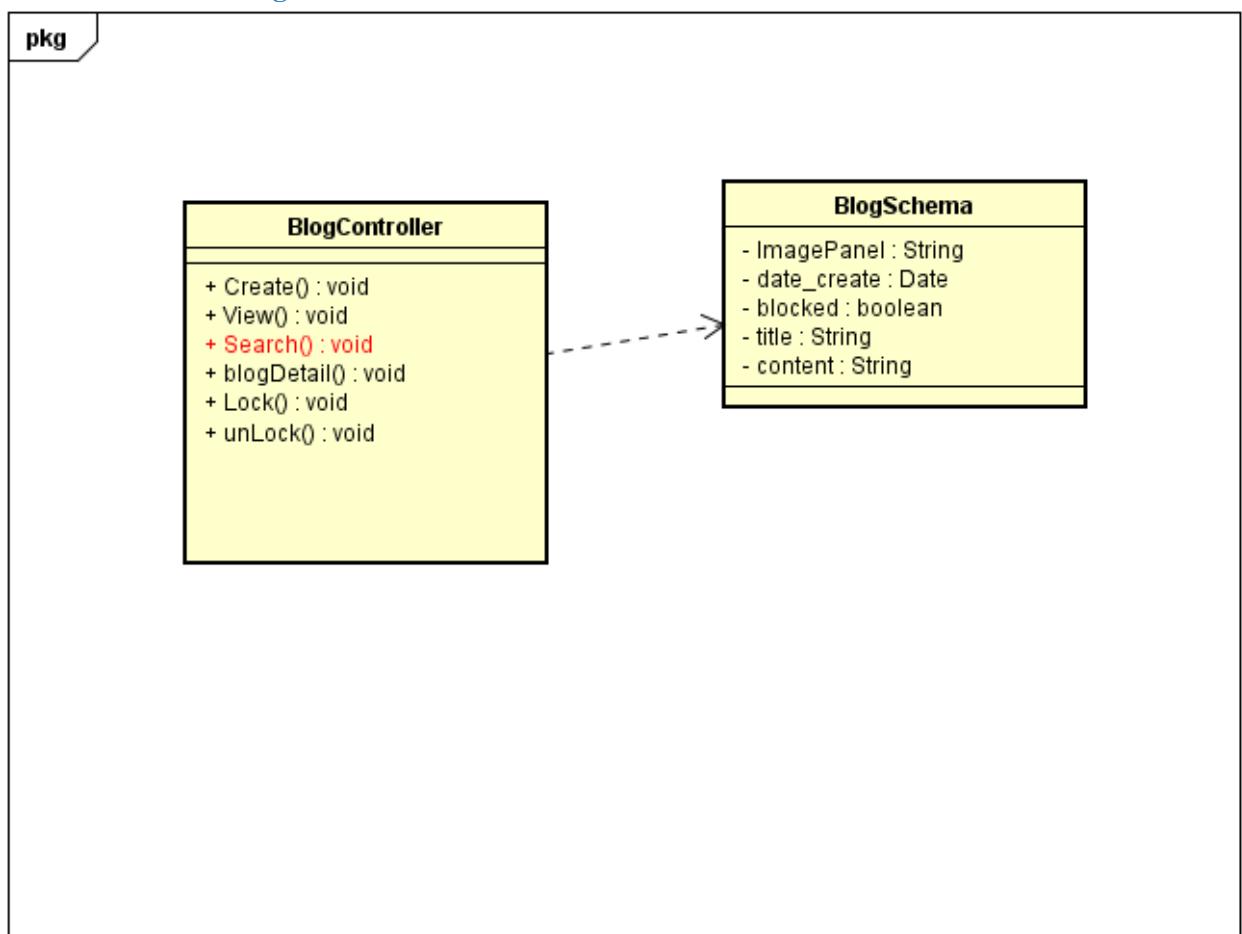
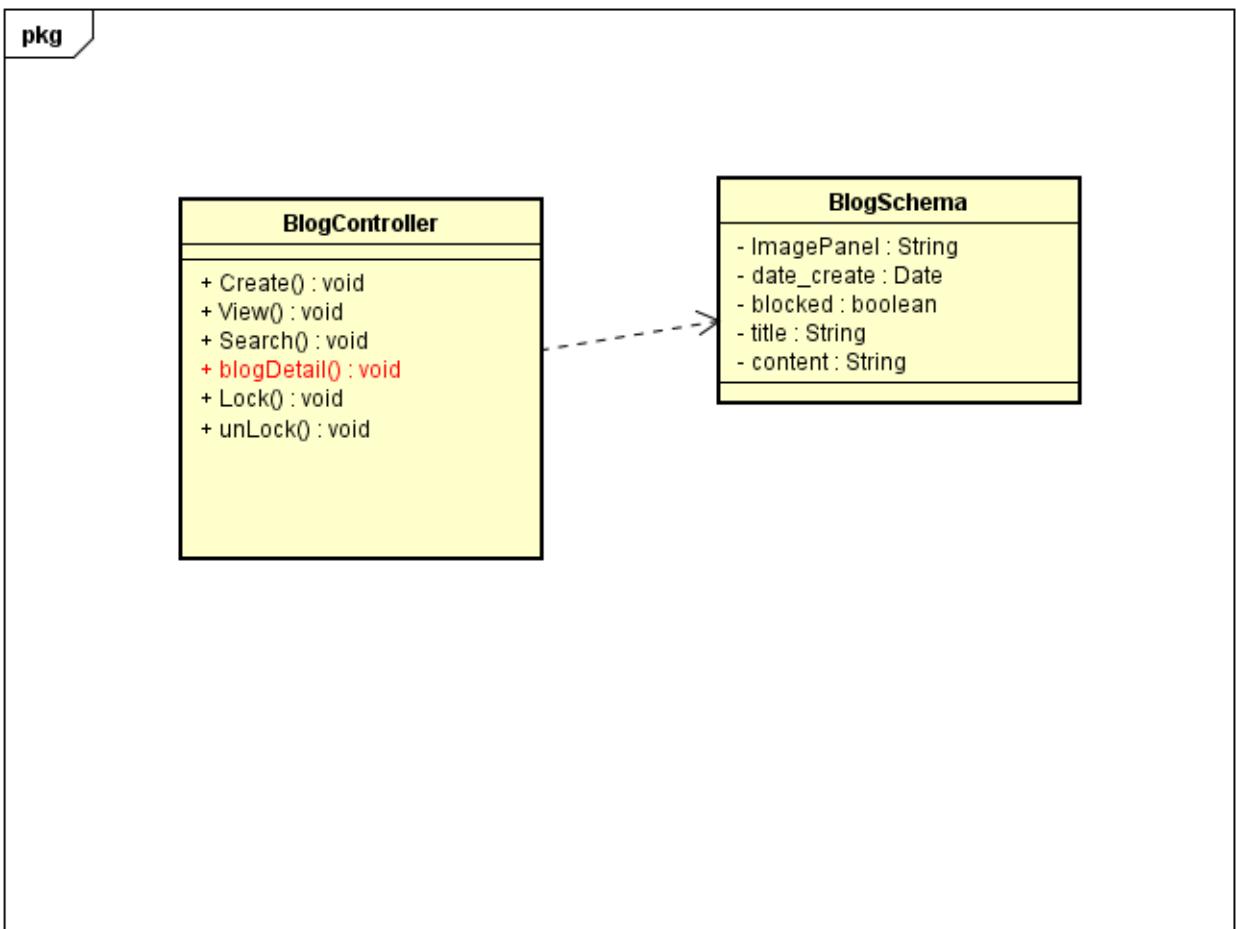


Figure 4-43: Search Blog

<b>Class</b>	BlogController	
<b>Description</b>	Seach blog by title	
<b>Base Class</b>	Controller	
<b>Source File</b>	BlogController.js	
<b>Namespace</b>	Controllers	
<b>Attributes</b>	N/A	
<b>Methods</b>	<b>Name</b>	<b>Description</b>
	Search()	Search blog by title

Table 4-53: Search  
Blog

#### 4.3.1.3.34 Blog Detail



**Figure 4-44: Blog Detail**

<b>Class</b>	BlogController	
<b>Description</b>	Show detail information of blog	
<b>Base Class</b>	Controller	
<b>Source File</b>	BlogController.js	
<b>Namespace</b>	Controllers	
<b>Attributes</b>	N/A	
<b>Methods</b>	<b>Name</b>	<b>Description</b>
	blogDetail()	Show detail information of blog

**Table 4-54: Blog Detail**

#### 4.3.1.3.35 Lock Blog

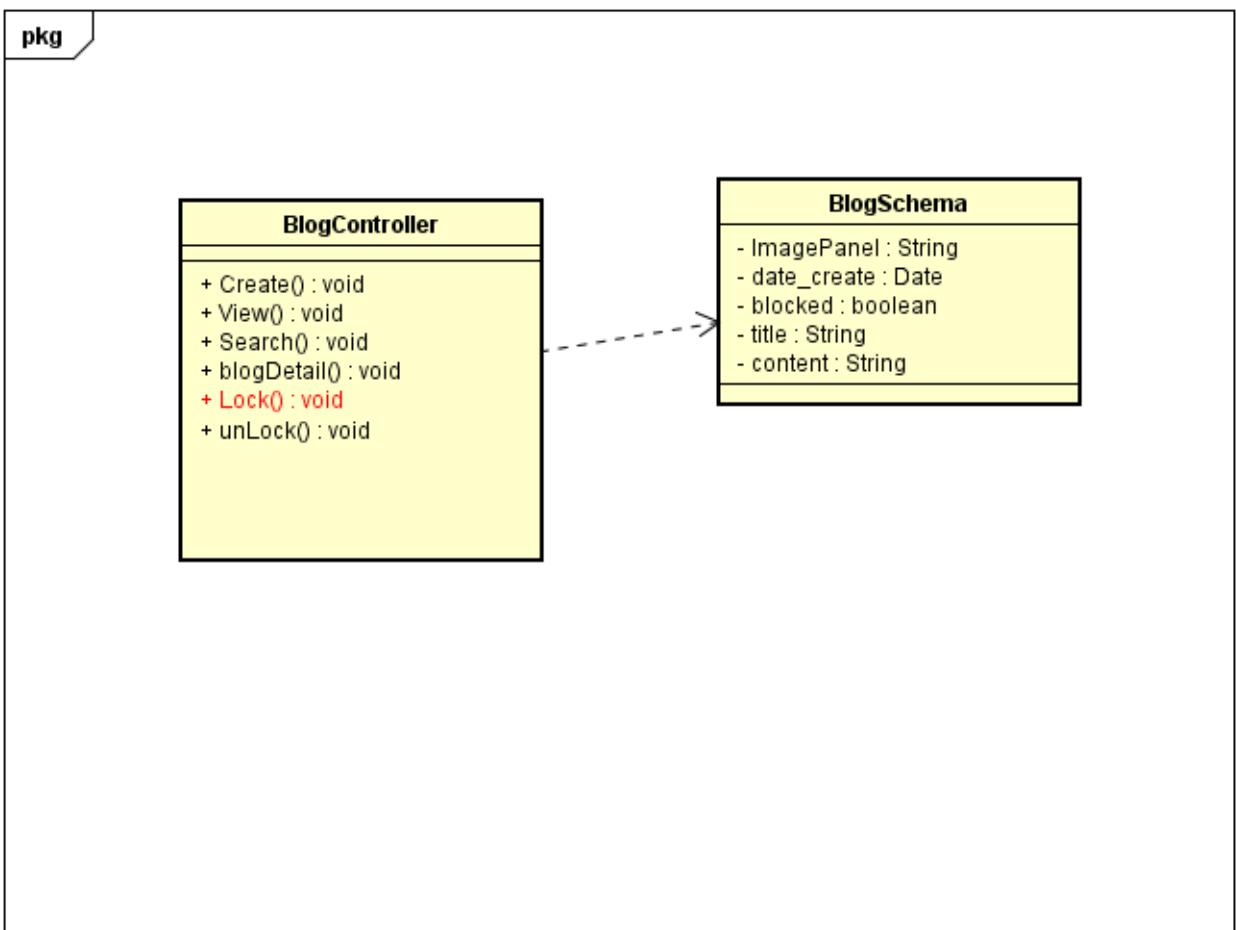
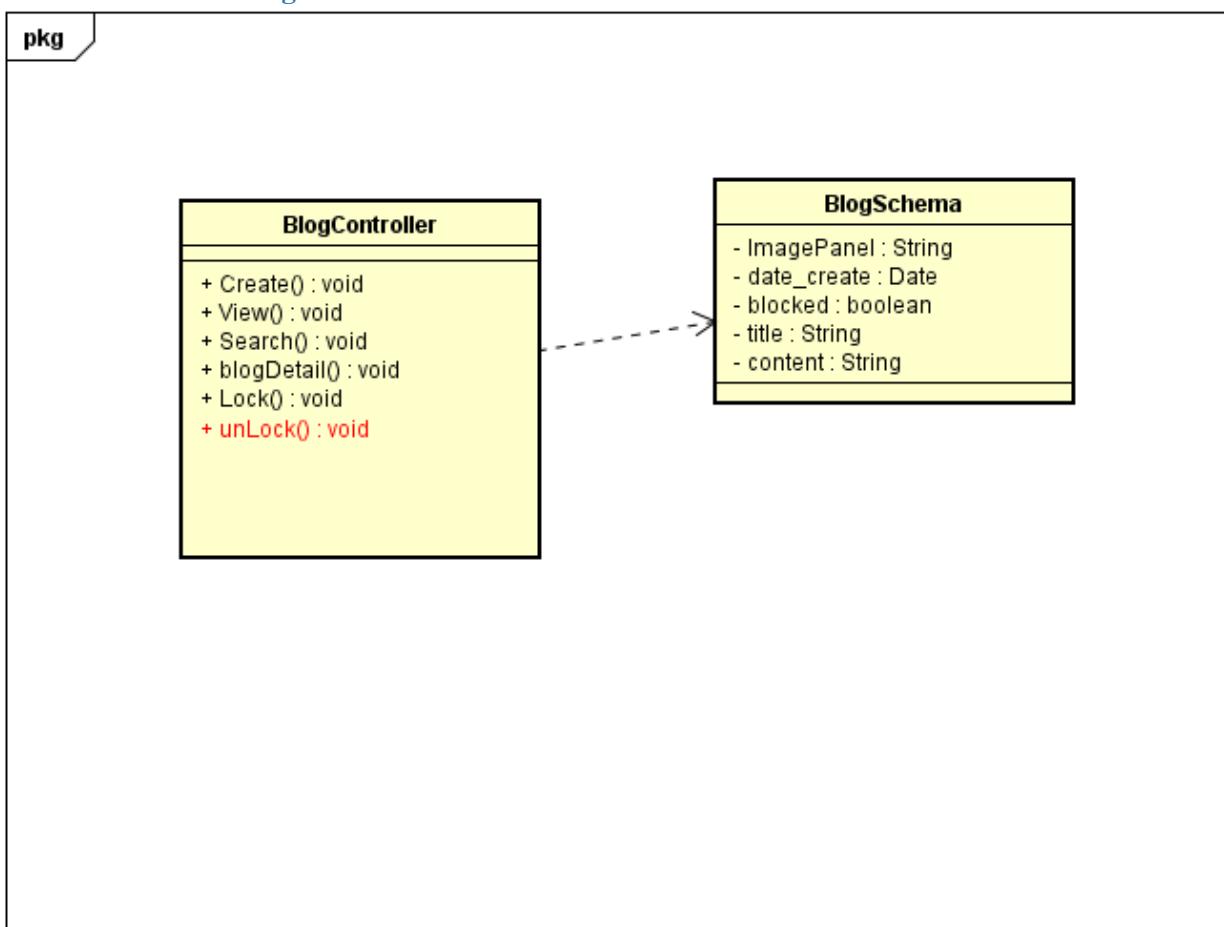


Figure 4-45: Lock Blog

<b>Class</b>	BlogController	
<b>Description</b>	Update status of blog (lock)	
<b>Base Class</b>	Controller	
<b>Source File</b>	BlogController.js	
<b>Namespace</b>	Controllers	
<b>Attributes</b>	N/A	
<b>Methods</b>	<b>Name</b>	<b>Description</b>
	Lock()	Update status of blog(lock)

Table 4-55: Lock Blog

#### 4.3.1.3.36 UnLock Blog



**Figure 4-46: UnLock Blog**

<b>Class</b>	BlogController					
<b>Description</b>	Update status of blog(open)					
<b>Base Class</b>	Controller					
<b>Source File</b>	BlogController.js					
<b>Namespace</b>	Controllers					
<b>Attributes</b>	N/A					
<b>Methods</b>	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>unLock()</td> <td>Update status of blog(open)</td> </tr> </tbody> </table>	Name	Description	unLock()	Update status of blog(open)	
Name	Description					
unLock()	Update status of blog(open)					

**Table 4-56: UnLock Blog**

#### 4.3.1.3.37 Insert Bill

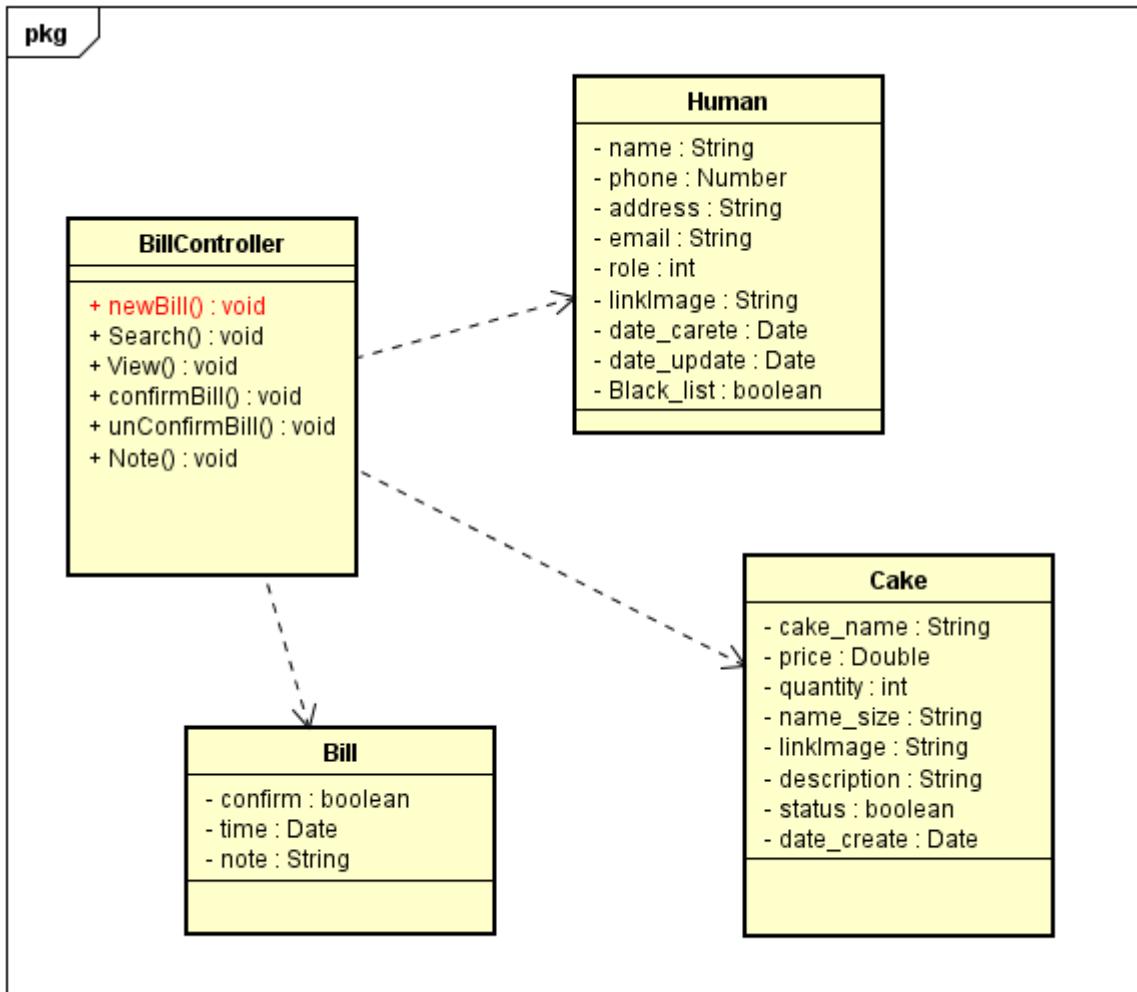


Figure 4-47: Insert Bill

<b>Class</b>	BillController	
<b>Description</b>	Insert new bill	
<b>Base Class</b>	Controller	
<b>Source File</b>	BillController.js	
<b>Namespace</b>	Controllers	
<b>Attributes</b>	N/A	
<b>Methods</b>	<b>Name</b>	<b>Description</b>
	newBill()	Insert new bill

Table 4-57: Insert Bill

#### 4.3.1.3.38 Search Bill

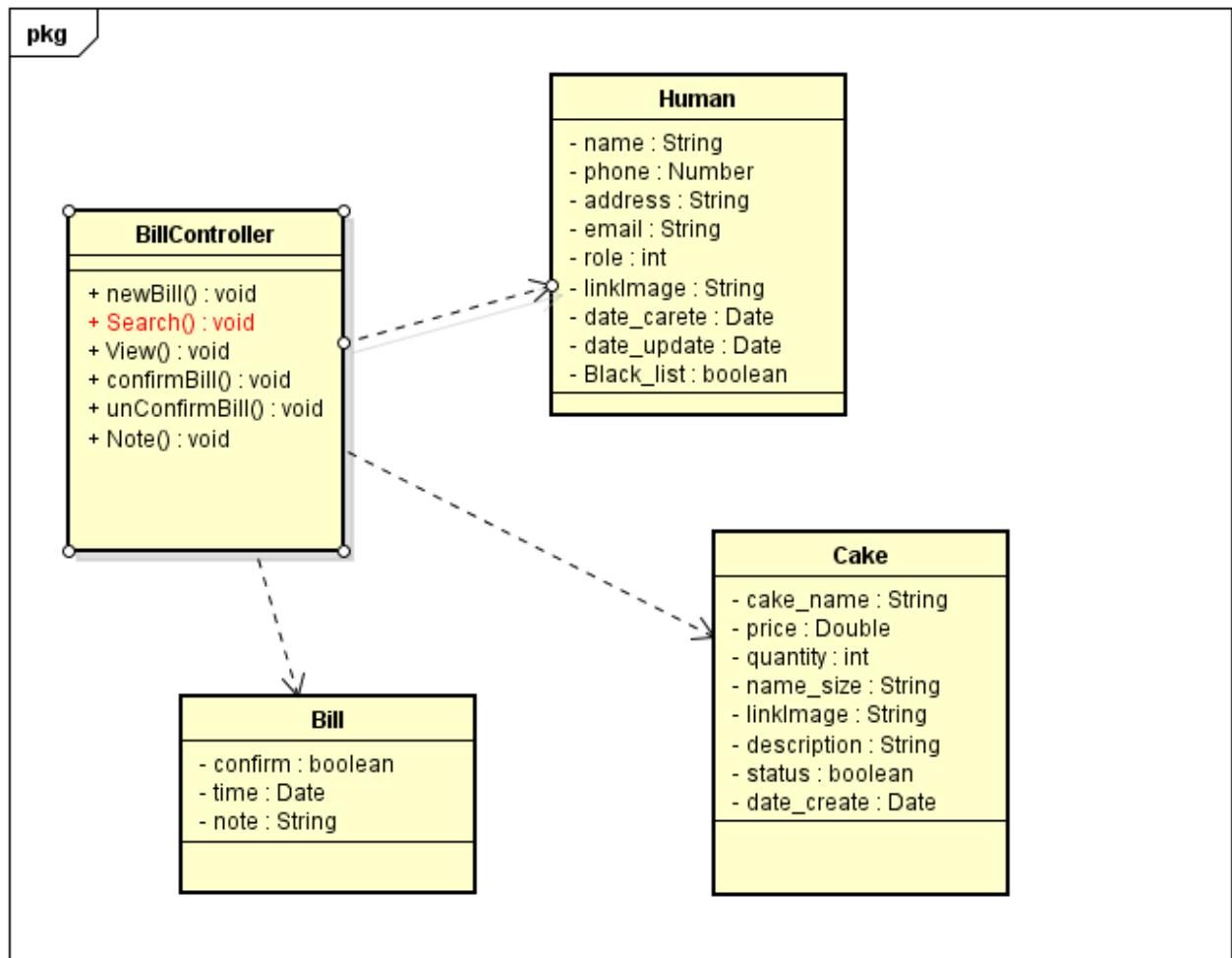


Figure 4-48: Search Bill

<b>Class</b>	BillController	
<b>Description</b>	Find bill	
<b>Base Class</b>	Controller	
<b>Source File</b>	BillController.js	
<b>Namespace</b>	Controllers	
<b>Attributes</b>	N/A	
<b>Methods</b>	<b>Name</b>	<b>Description</b>
	Search()	Find bill

Table 4-58: Search Bill

#### 4.3.1.3.39 View Bill

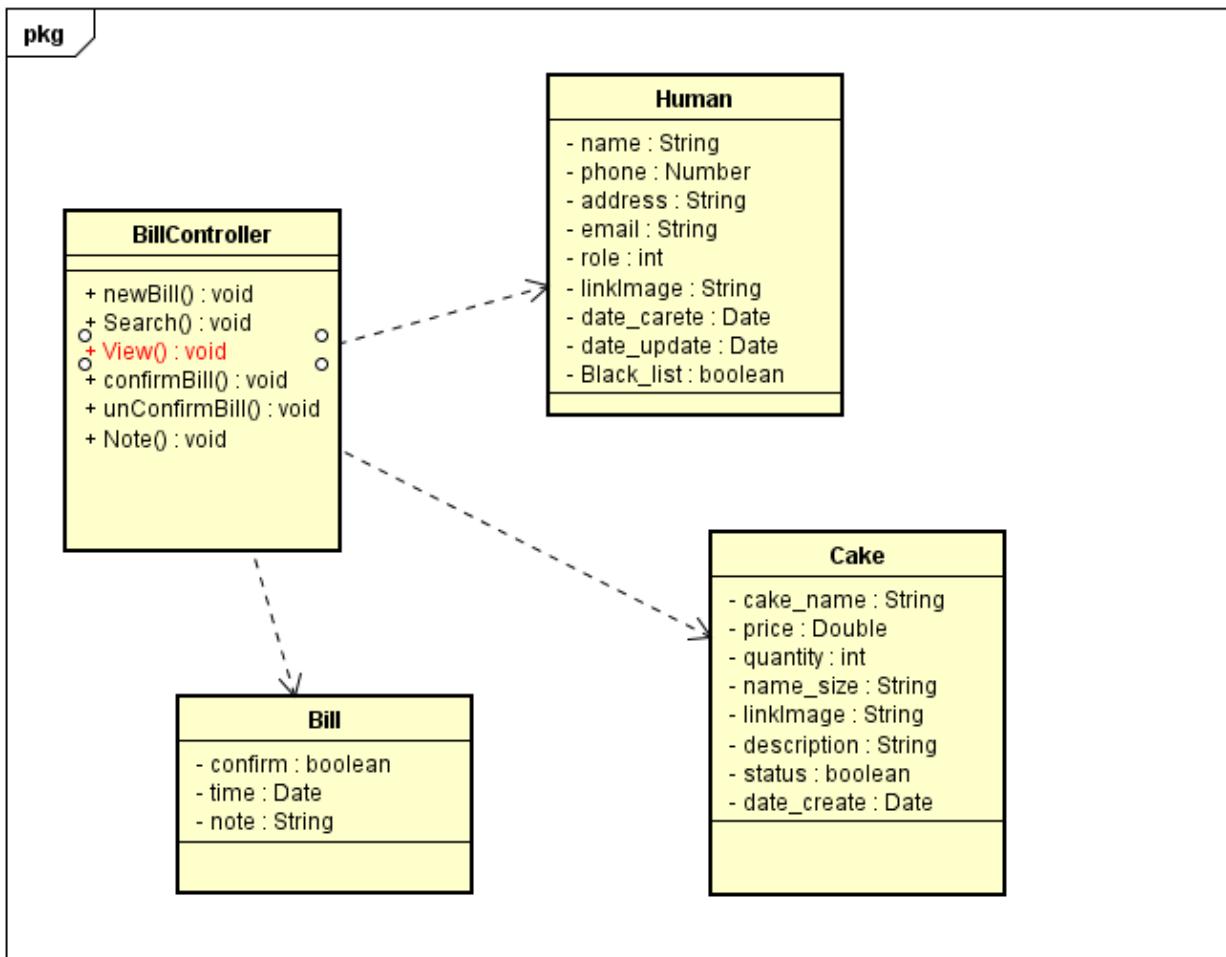


Figure 4-49: View Bill

<b>Class</b>	BillController	
<b>Description</b>	View information of bill	
<b>Base Class</b>	Controller	
<b>Source File</b>	BillController.js	
<b>Namespace</b>	Controllers	
<b>Attributes</b>	N/A	
<b>Methods</b>	<b>Name</b>	<b>Description</b>
	View()	View information of bill

Table 4-59: View Bill

#### 4.3.1.3.40 Confirm Bill

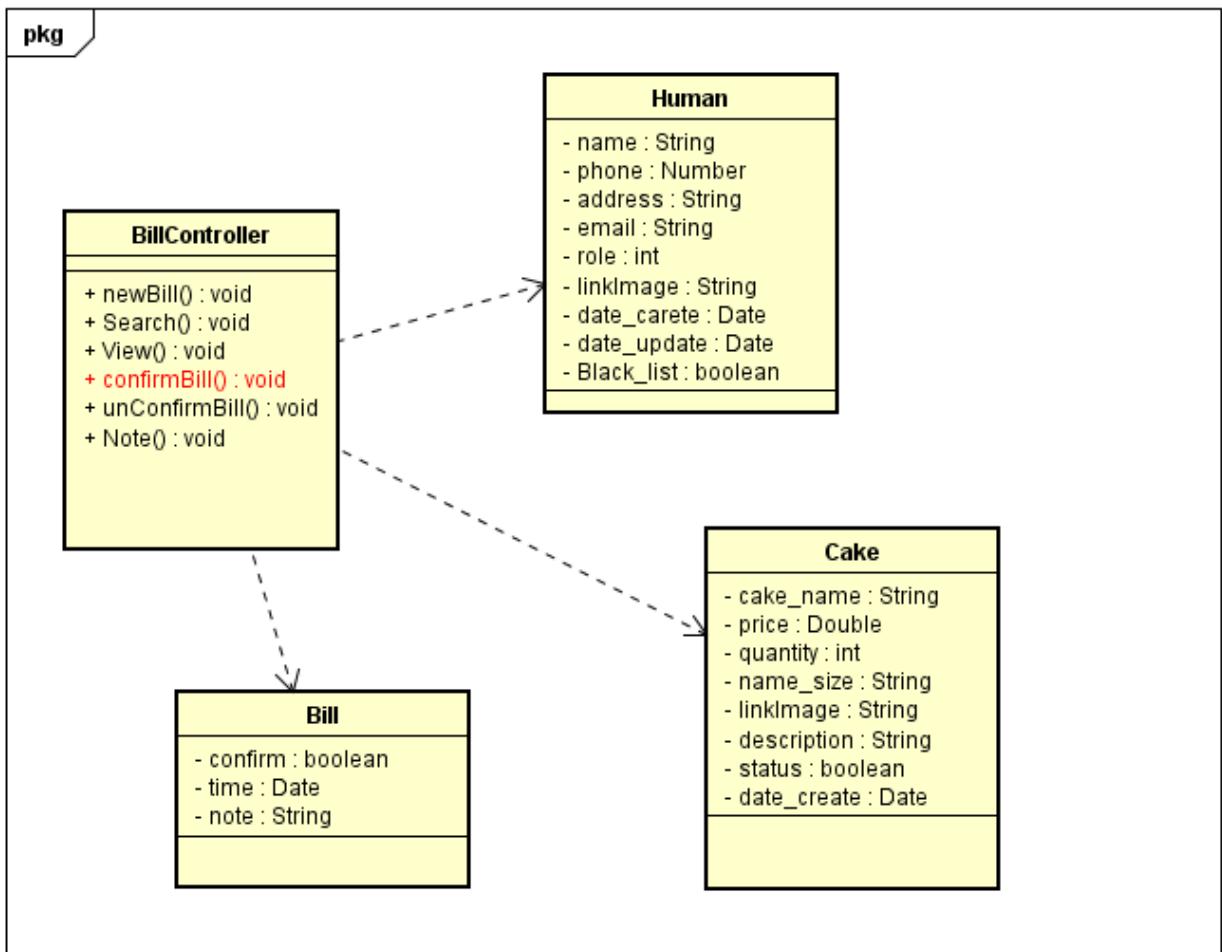


Figure 4-50: Confirm Bill

<b>Class</b>	BillController	
<b>Description</b>	Confirm bill	
<b>Base Class</b>	Controller	
<b>Source File</b>	BillController.js	
<b>Namespace</b>	Controllers	
<b>Attributes</b>	N/A	
<b>Methods</b>	<b>Name</b>	<b>Description</b>
	confirmBill()	Confirm bill

Figure 4-51: Confirm Bill

#### 4.3.1.3.41 Unconfirm Bill

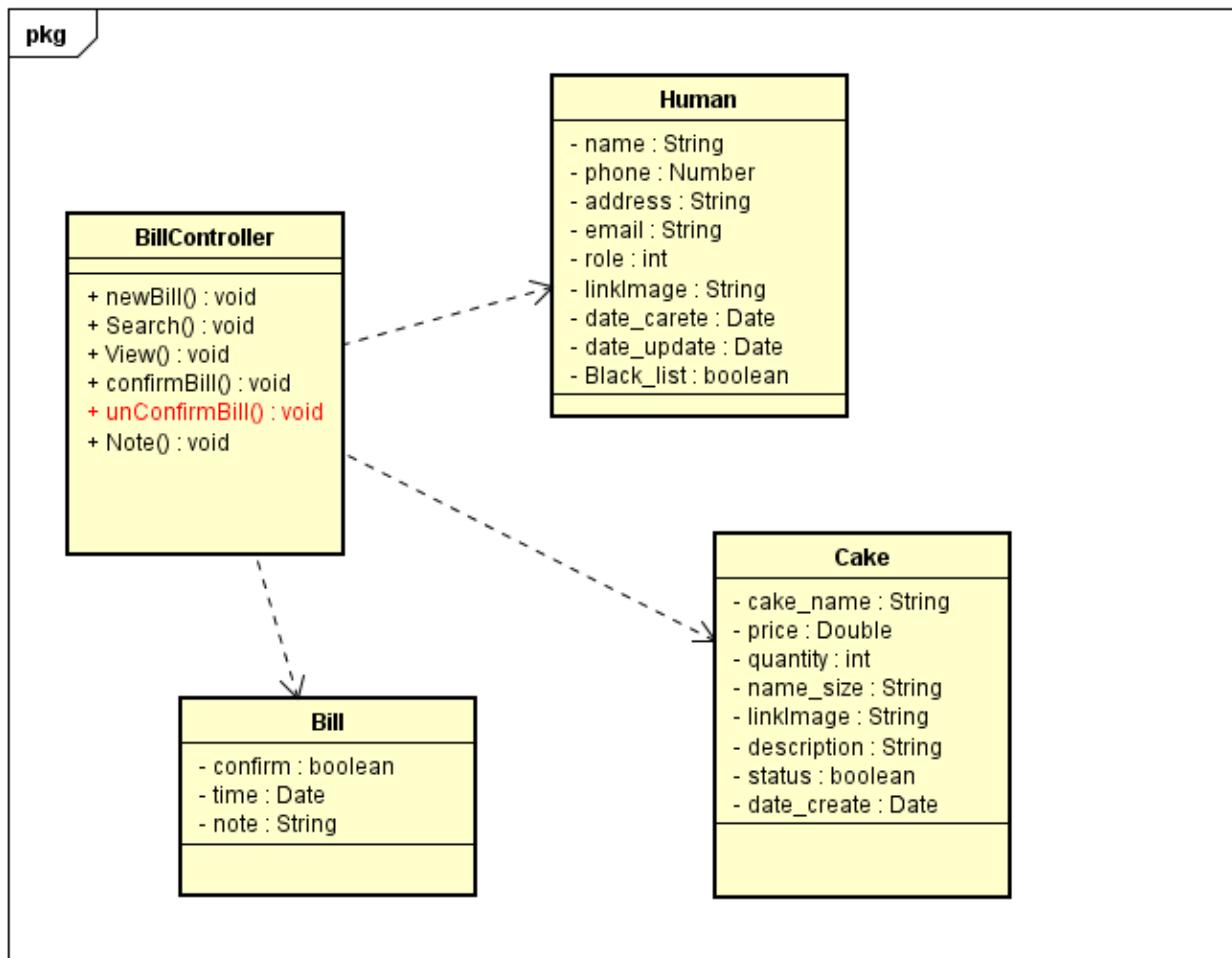
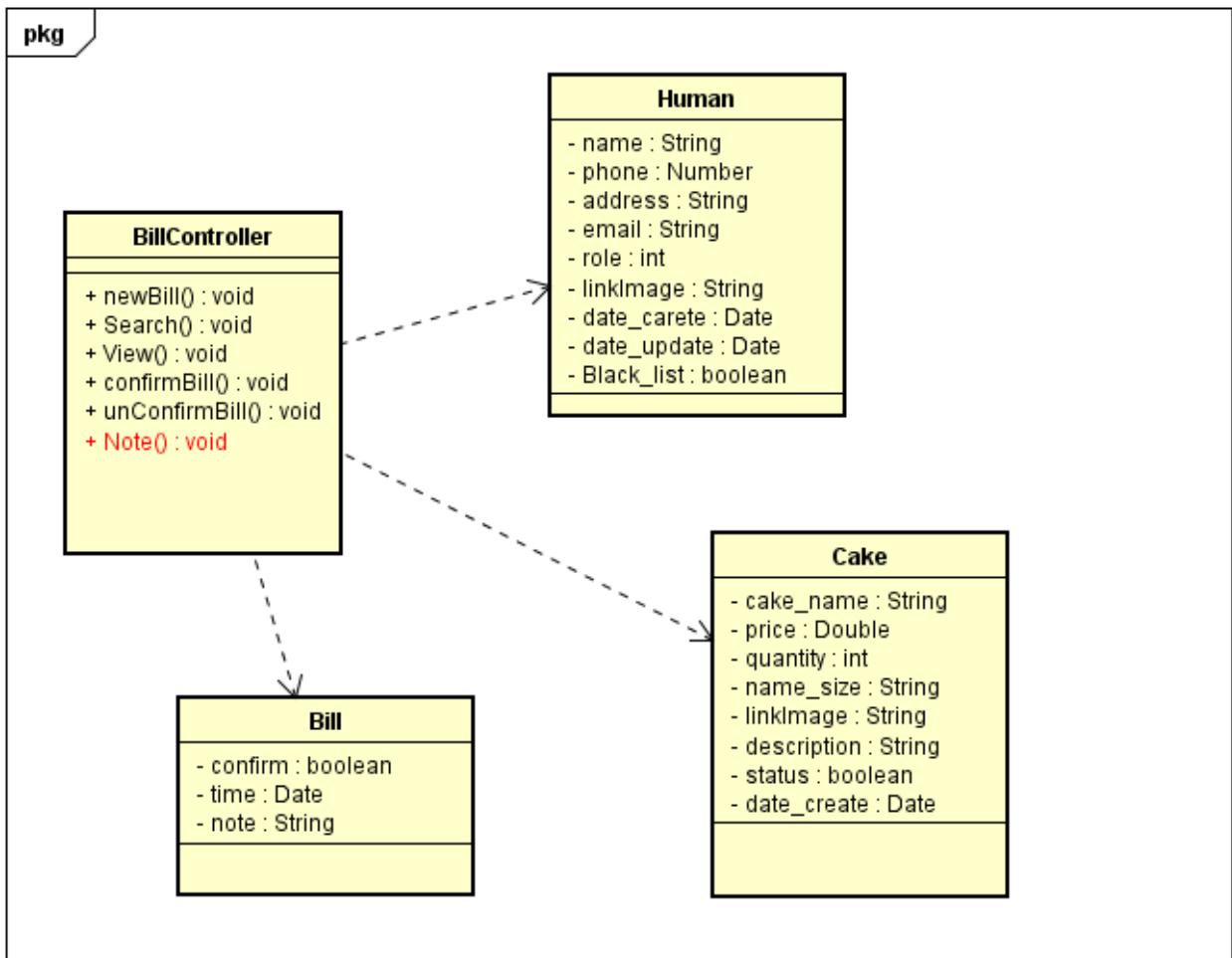


Figure 4-52: Unconfirm Bill

<b>Class</b>	BillController	
<b>Description</b>	Unconfirm bill	
<b>Base Class</b>	Controller	
<b>Source File</b>	BillController.js	
<b>Namespace</b>	Controllers	
<b>Attributes</b>	N/A	
<b>Methods</b>	<b>Name</b>	<b>Description</b>
	unconfirmBill()	Unconfirm bill

**Table 4-60:** Unconfirm Bill

#### 4.3.1.3.42 Note



**Figure 4-53:** Note

<b>Class</b>	BillController	
<b>Description</b>	Note information of bill	
<b>Base Class</b>	Controller	
<b>Source File</b>	BillController.js	
<b>Namespace</b>	Controllers	
<b>Attributes</b>	N/A	
<b>Methods</b>	Name	Description

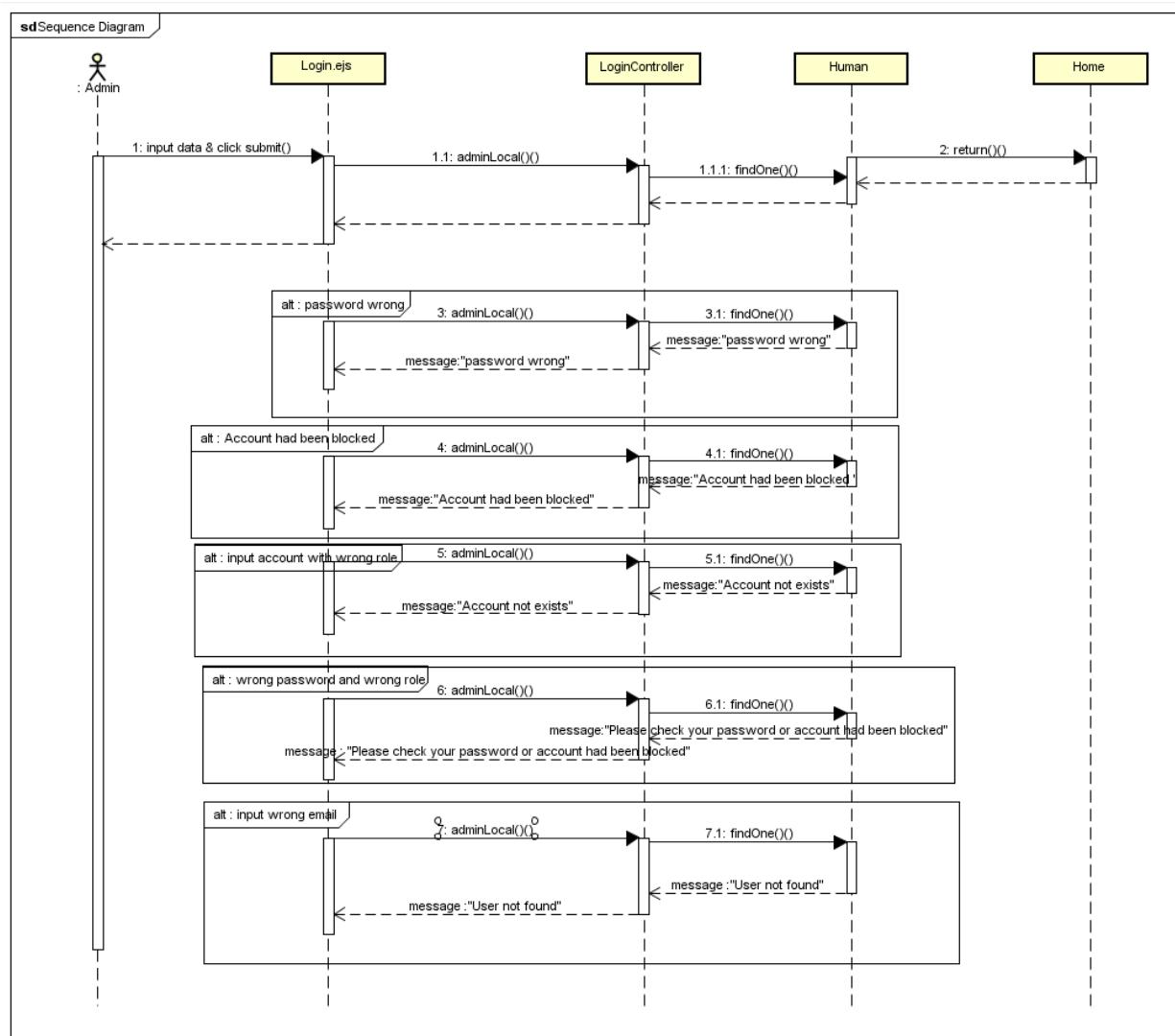
	Note()	Note information of bill
--	--------	--------------------------

**Table 4-61:** Note

#### 4.3.1.4 Sequence diagram

##### 4.3.1.4.1 Login

###### 4.3.1.4.1.1 Login Admin

**Figure 4-54:** Login Admin

#### 4.3.1.4.1.2 Login Staff

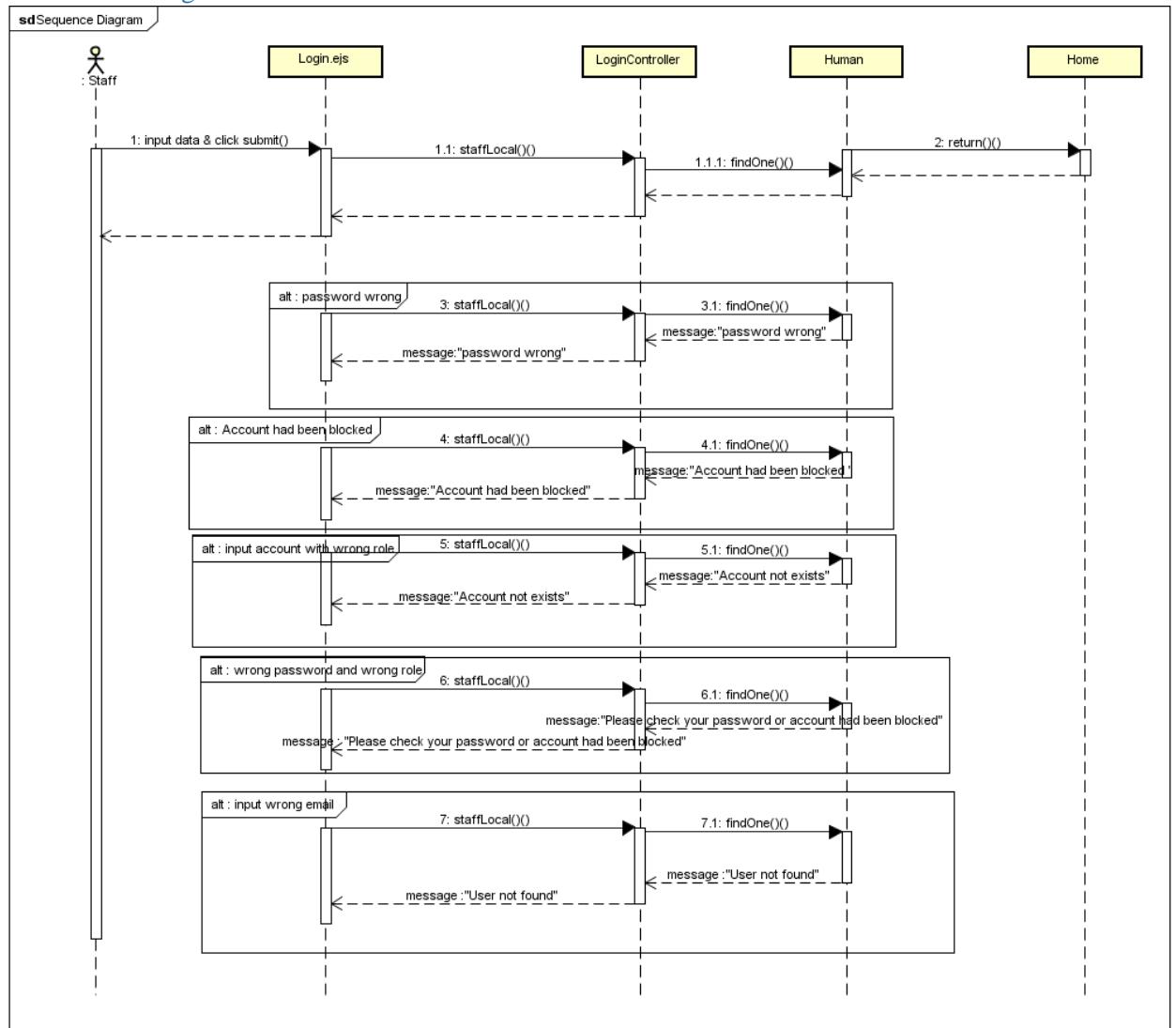


Figure 4-55: Login Staff

#### 4.3.1.4.1.3 Login User

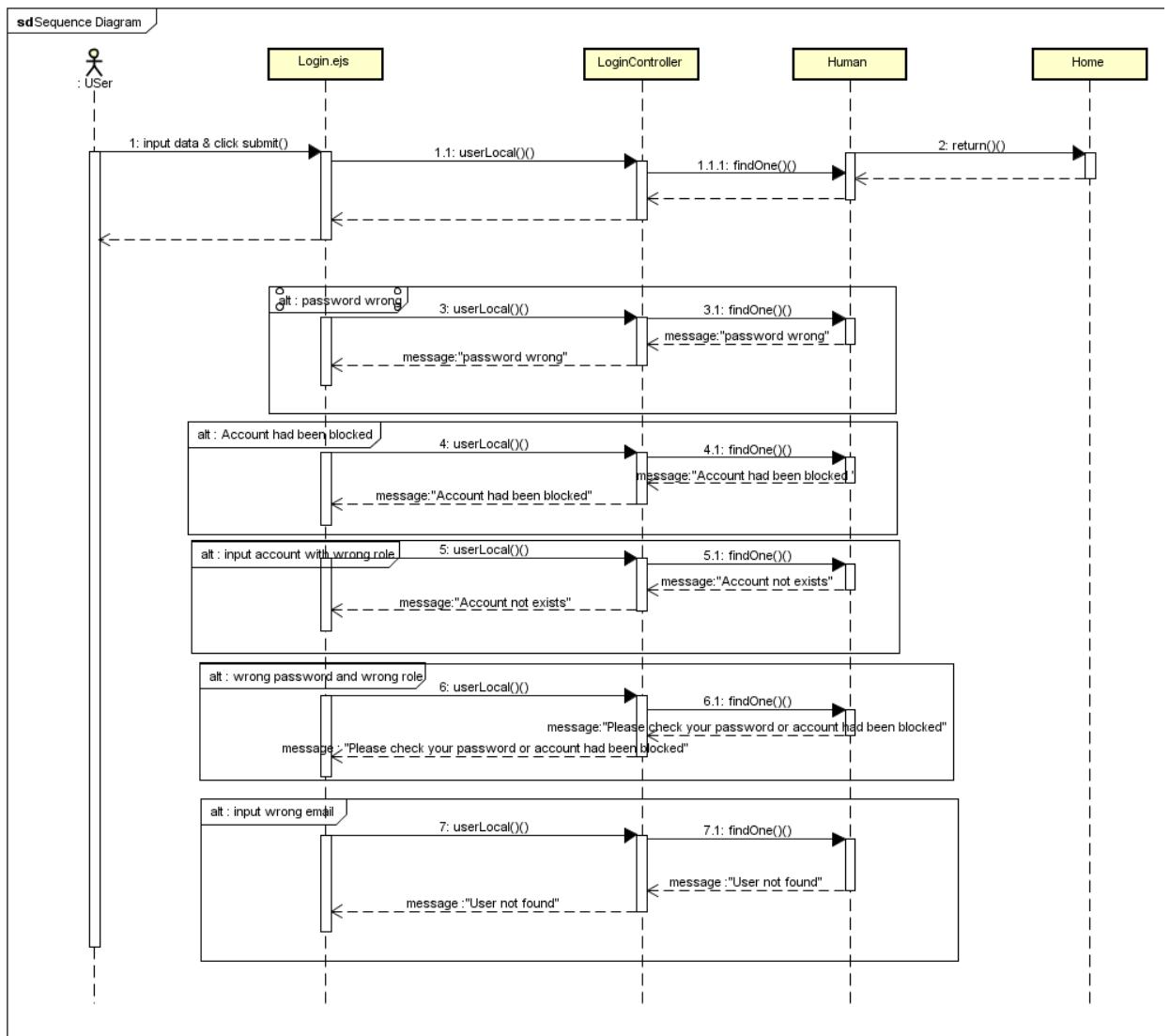


Figure 4-56: Login User

#### 4.3.1.4.1.4 Logout

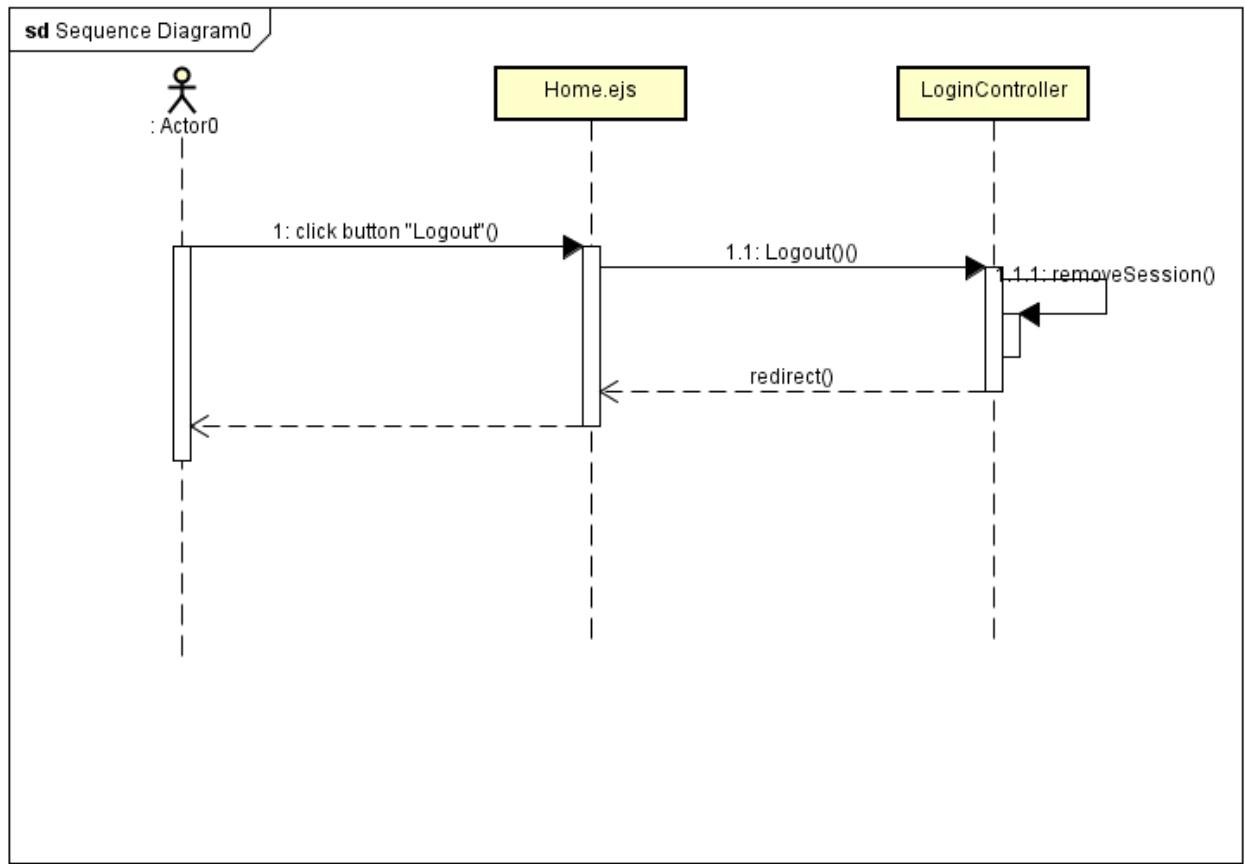


Figure 4-57: Logout

#### 4.3.1.4.1.5 Login Register

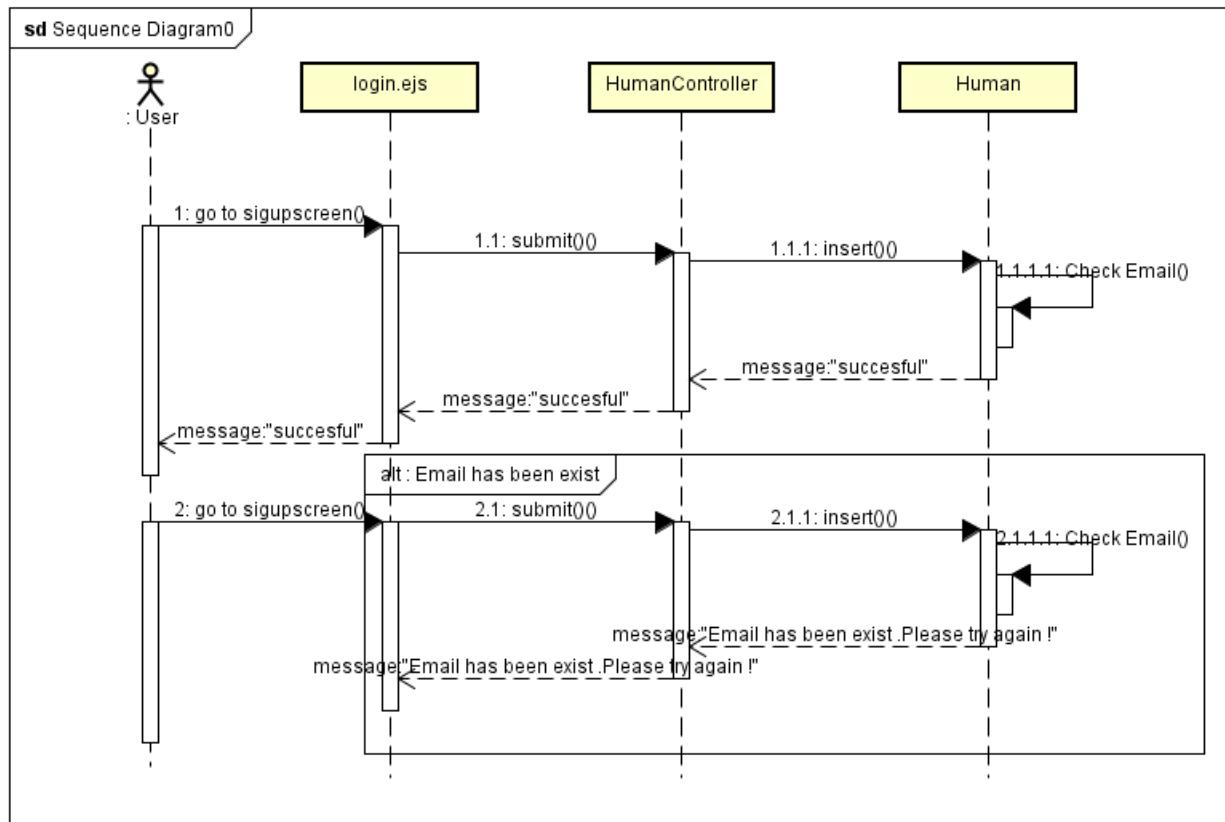
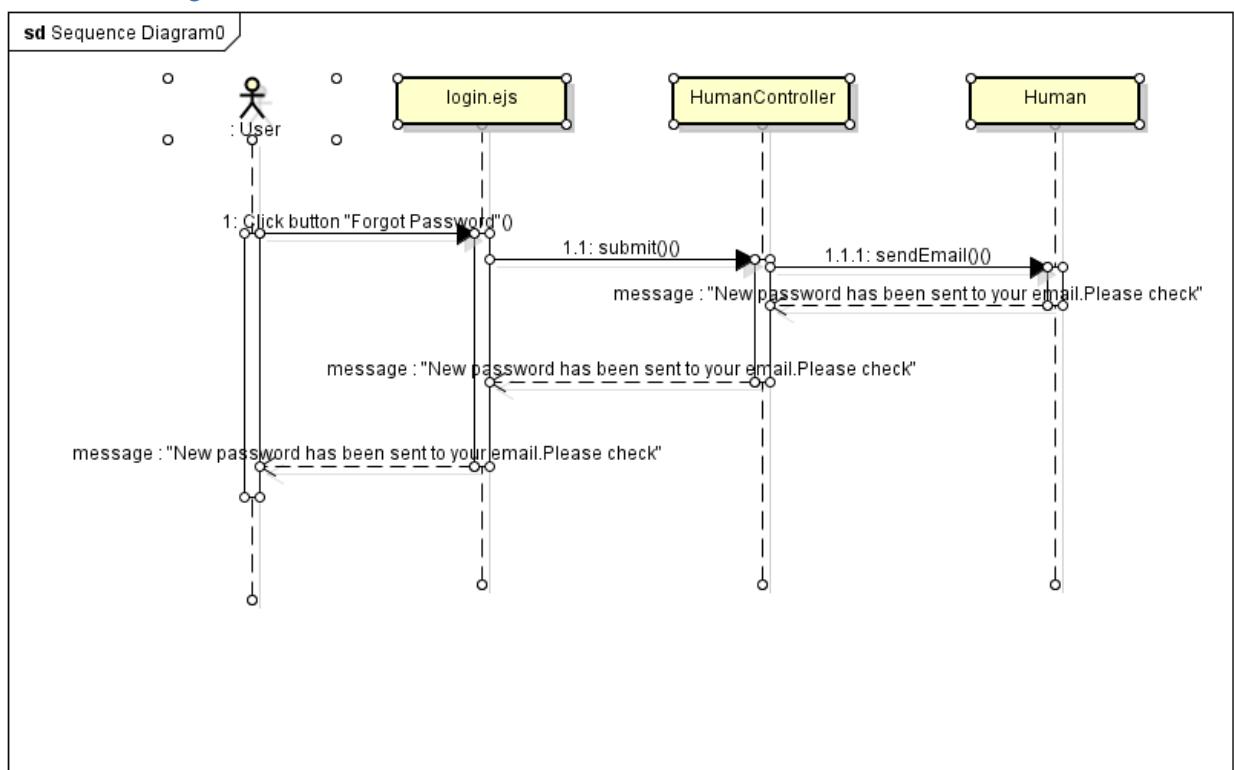


Figure 4-58: Login Register

#### 4.3.1.4.1.6 Forgot Password



**Figure 4-59:** Forgot Password

#### 4.3.1.4.2 Manage Account

#### 4.3.1.4.2.1 Insert Staff

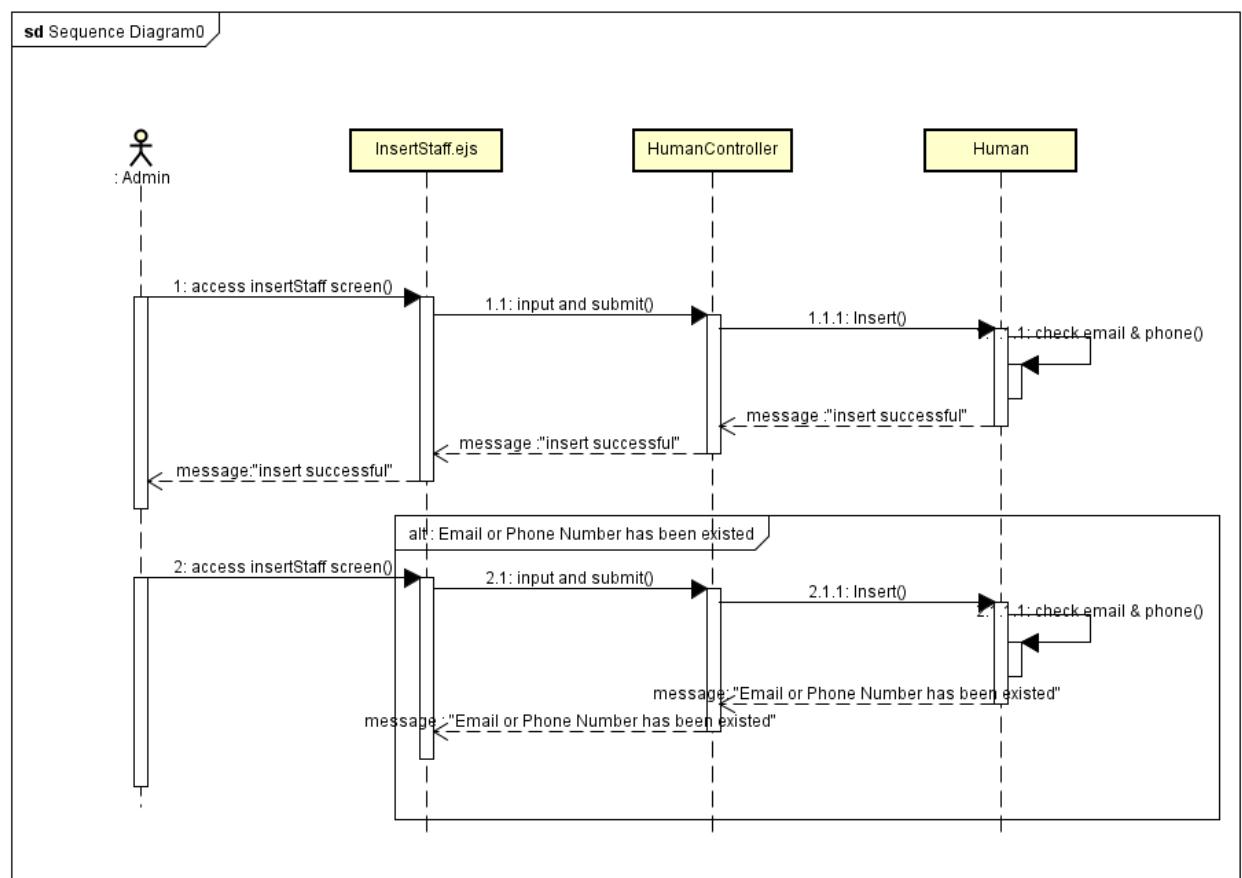


Figure 4-60: Insert Staff

#### 4.3.1.4.2.2 Update Staff

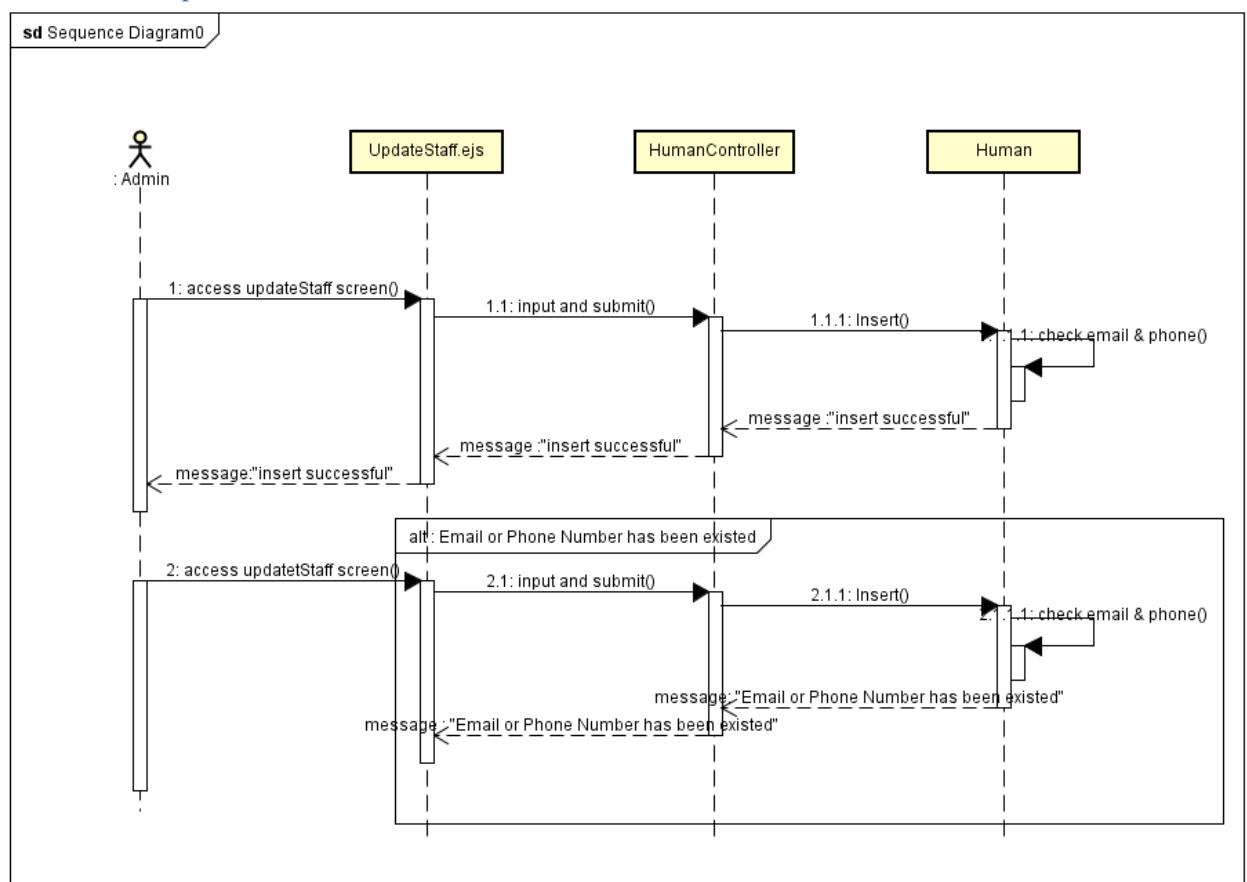
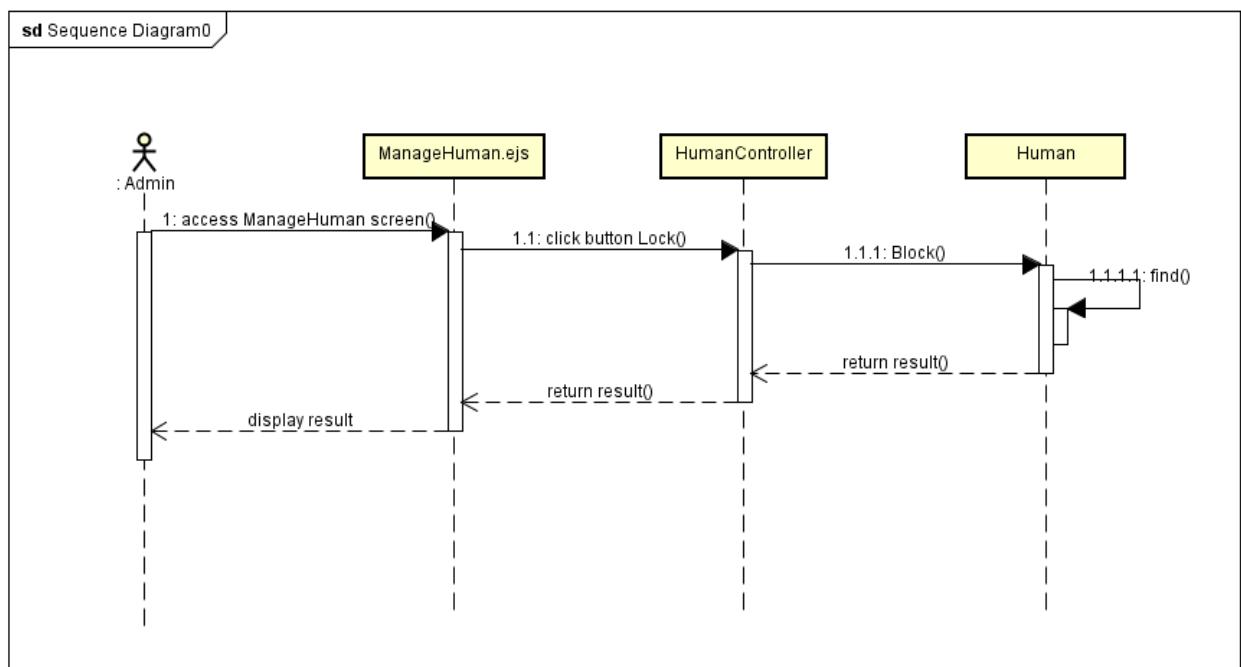


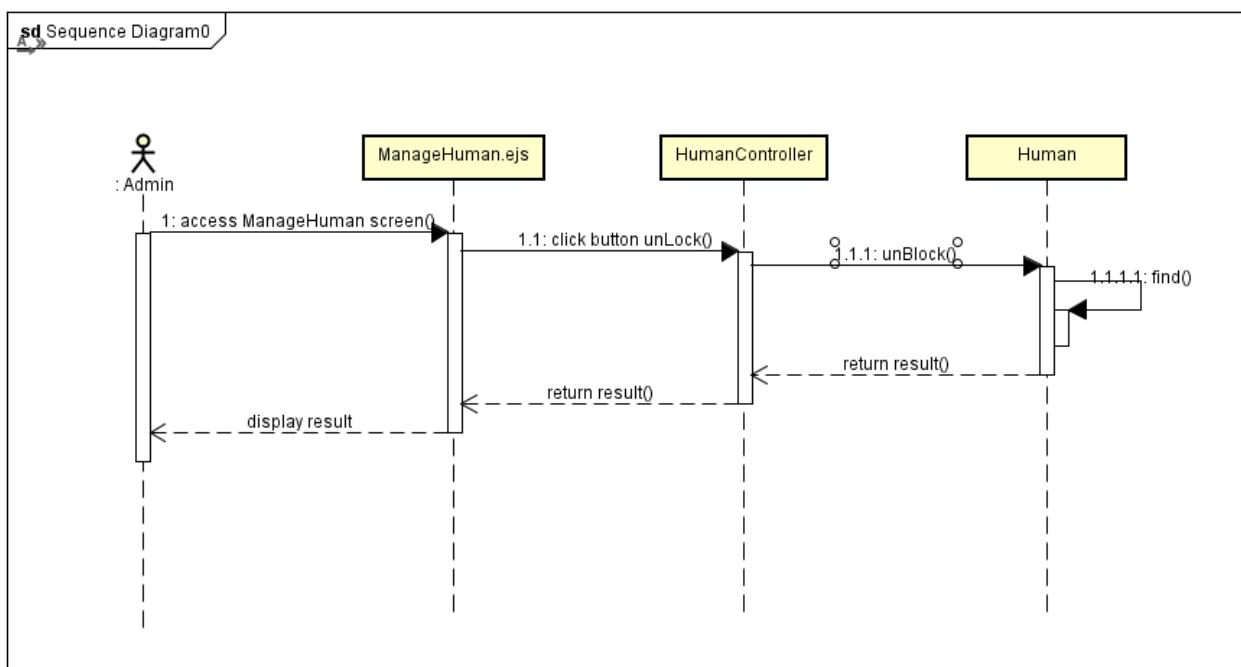
Figure 4-61: Update Staff

#### 4.3.1.4.2.3 Lock Account



**Figure 4-62:** Lock Account

#### 4.3.1.4.2.4 UnLock Account



**Figure 4-63:** UnLock Account

#### 4.3.1.4.3 Manage Cake

##### 4.3.1.4.3.1 Insert Cake

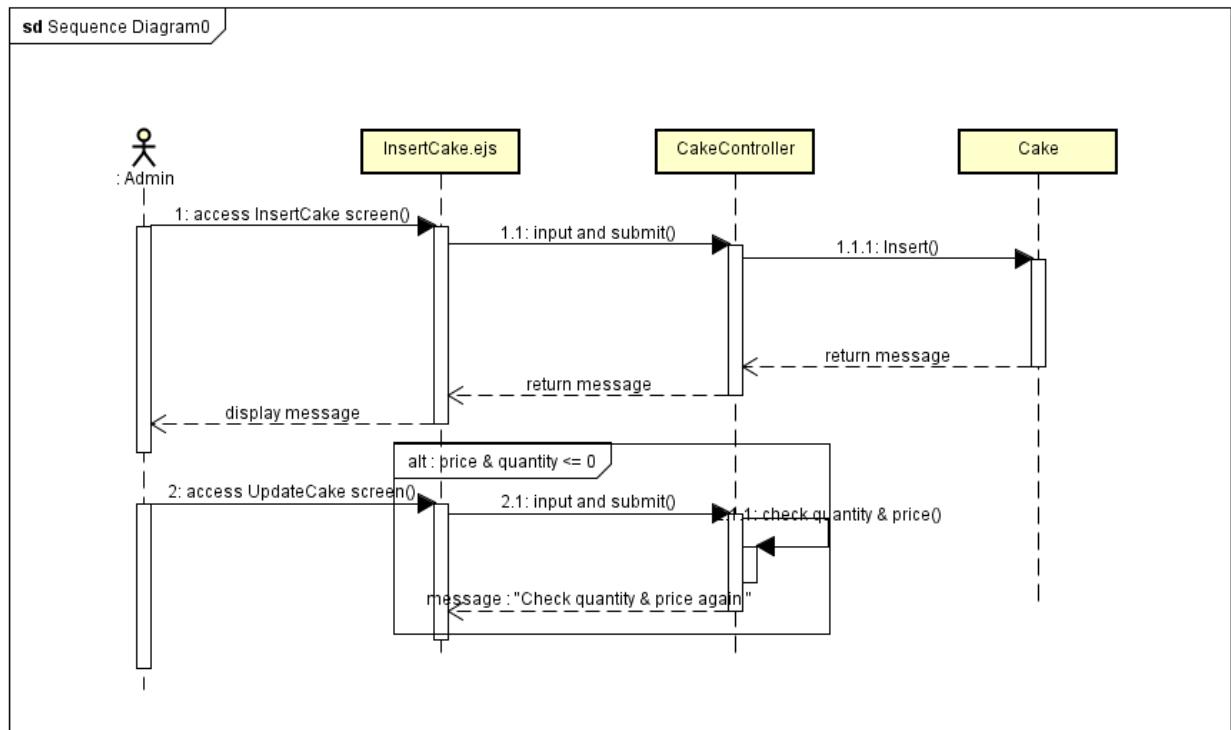
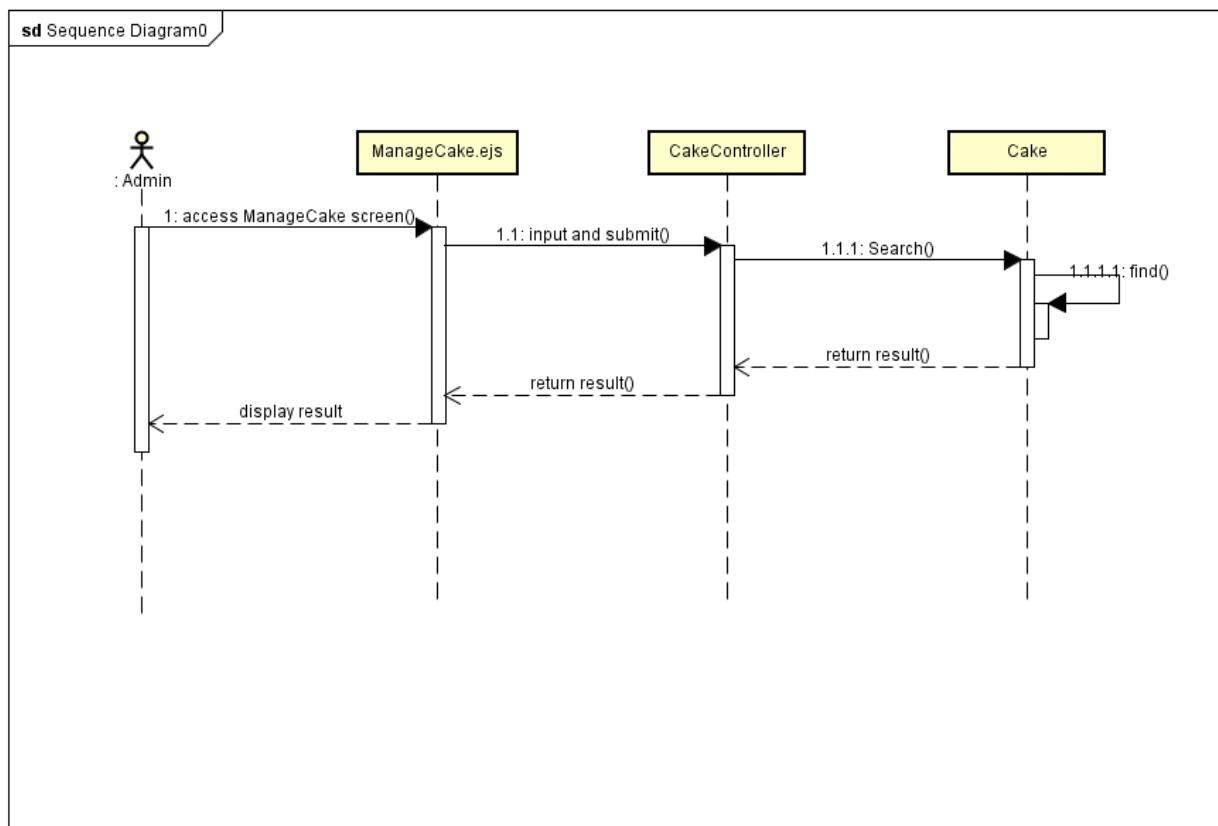


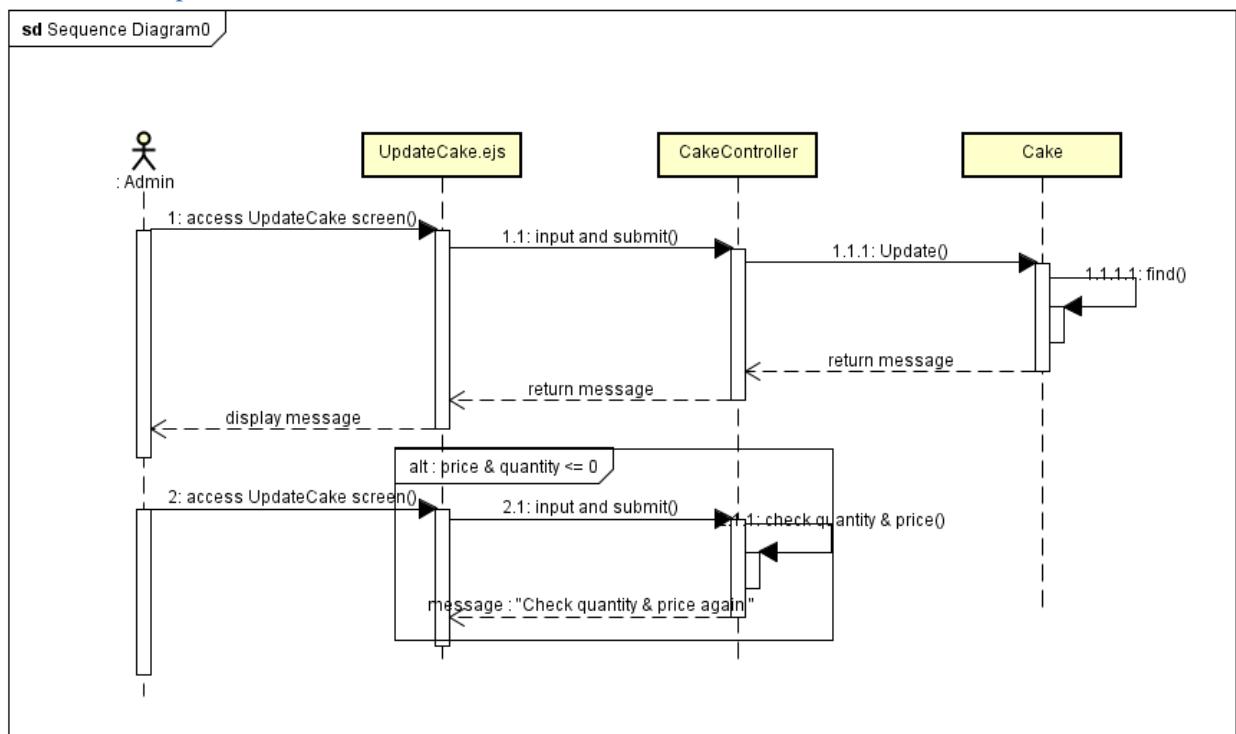
Figure 4-64: Insert Cake

#### 4.3.1.4.3.2 Search Cake



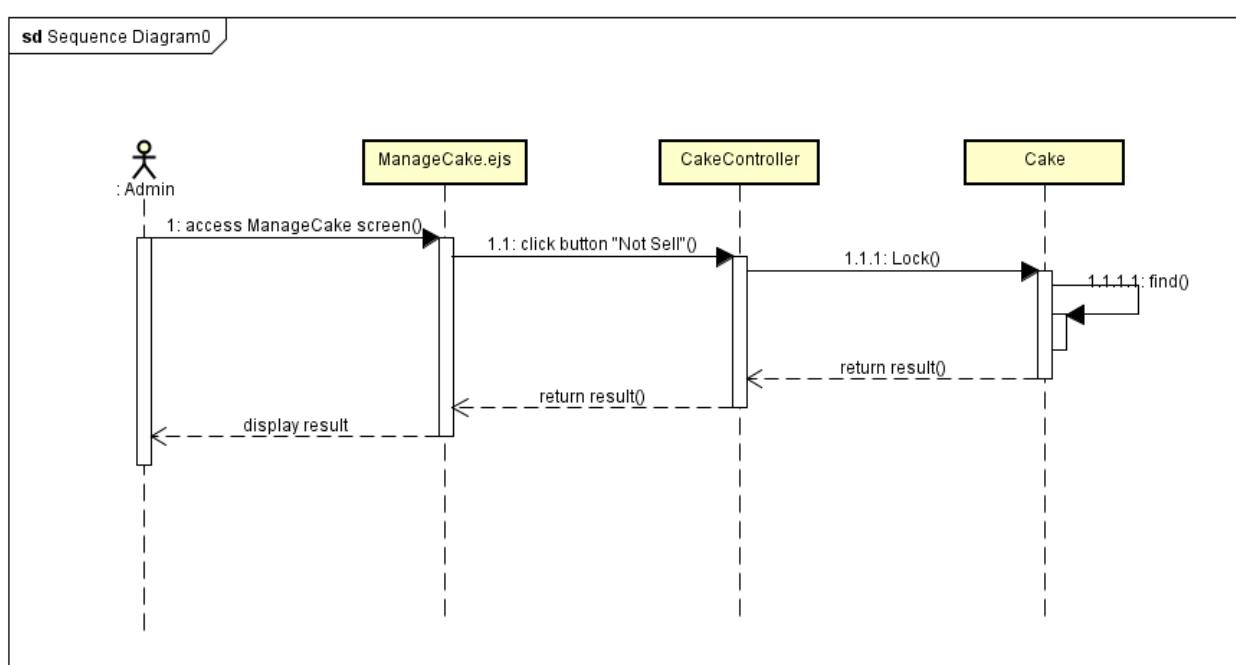
**Figure 4-65: Search Cake**

#### 4.3.1.4.3.3 Update Cake



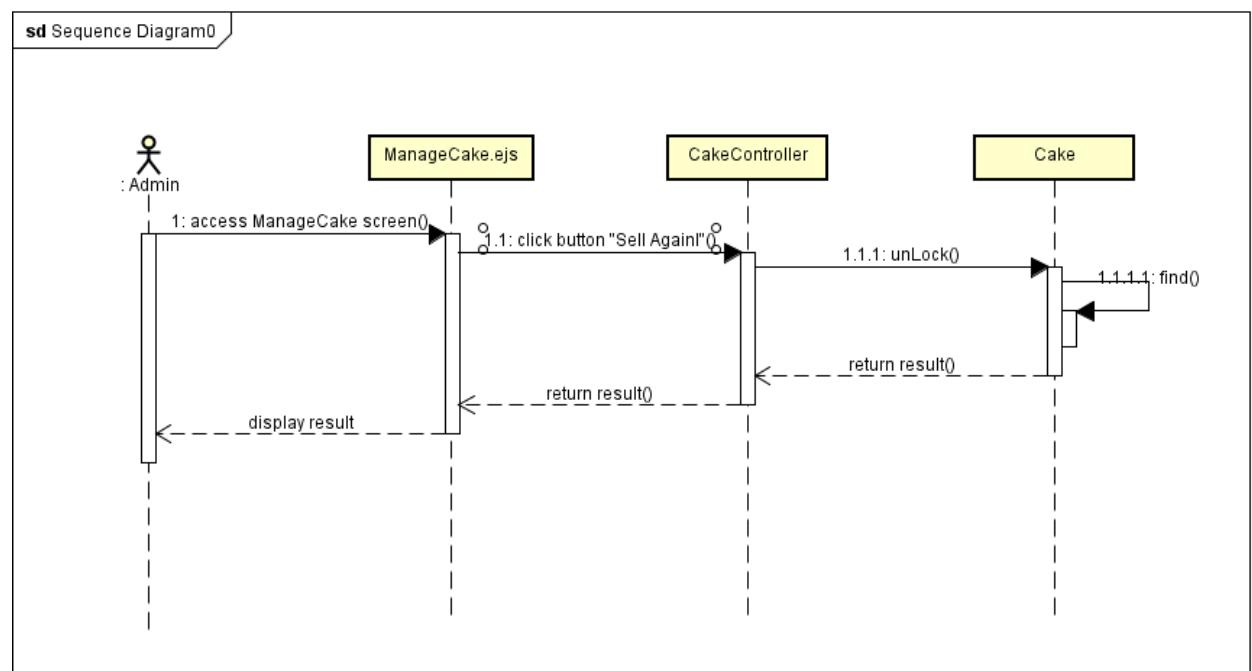
**Figure 4-66: Update Cake**

#### 4.3.1.4.3.4 Lock Cake



**Figure 4-67: Lock Cake**

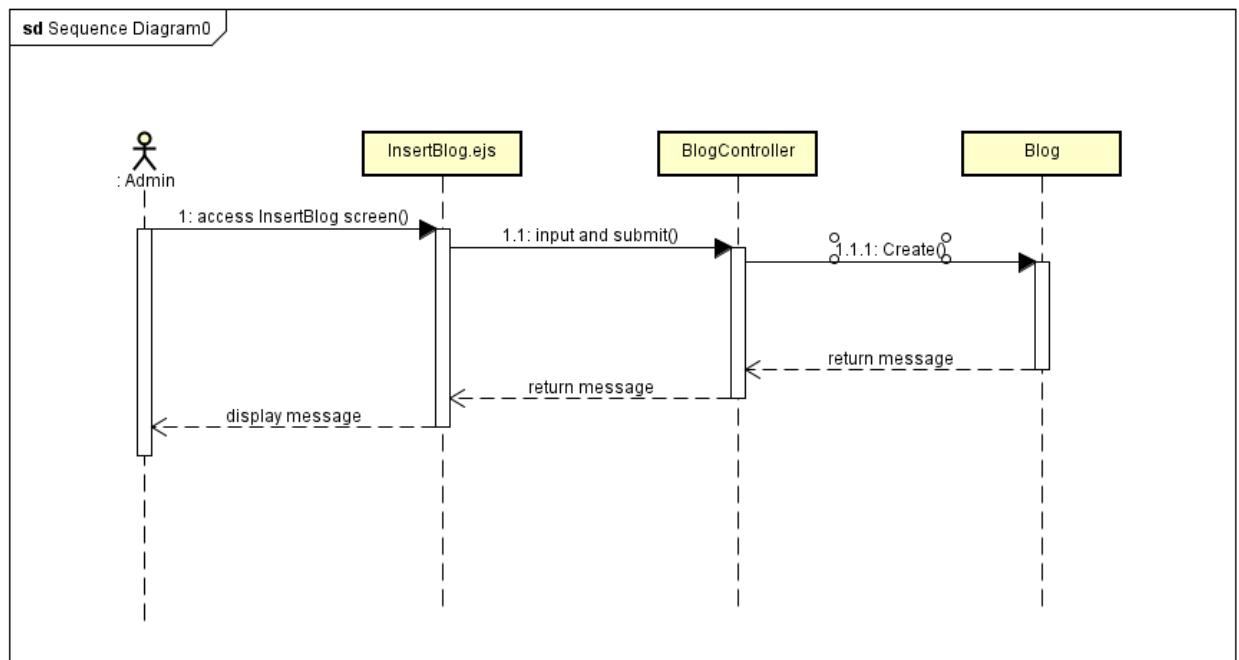
#### 4.3.1.4.3.5 UnLock Cake



**Figure 4-68:** UnLock Cake

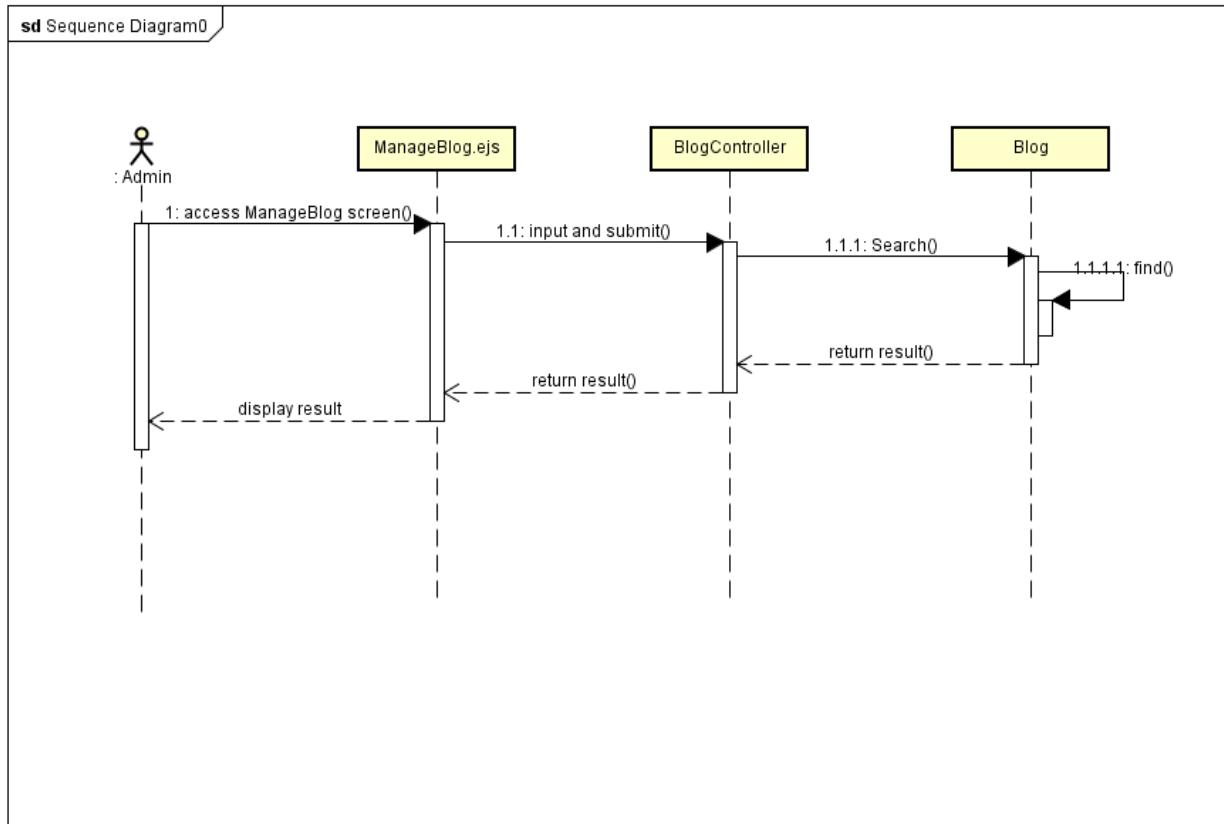
#### 4.3.1.4.4 Manage Blog

##### 4.3.1.4.4.1 Insert Blog



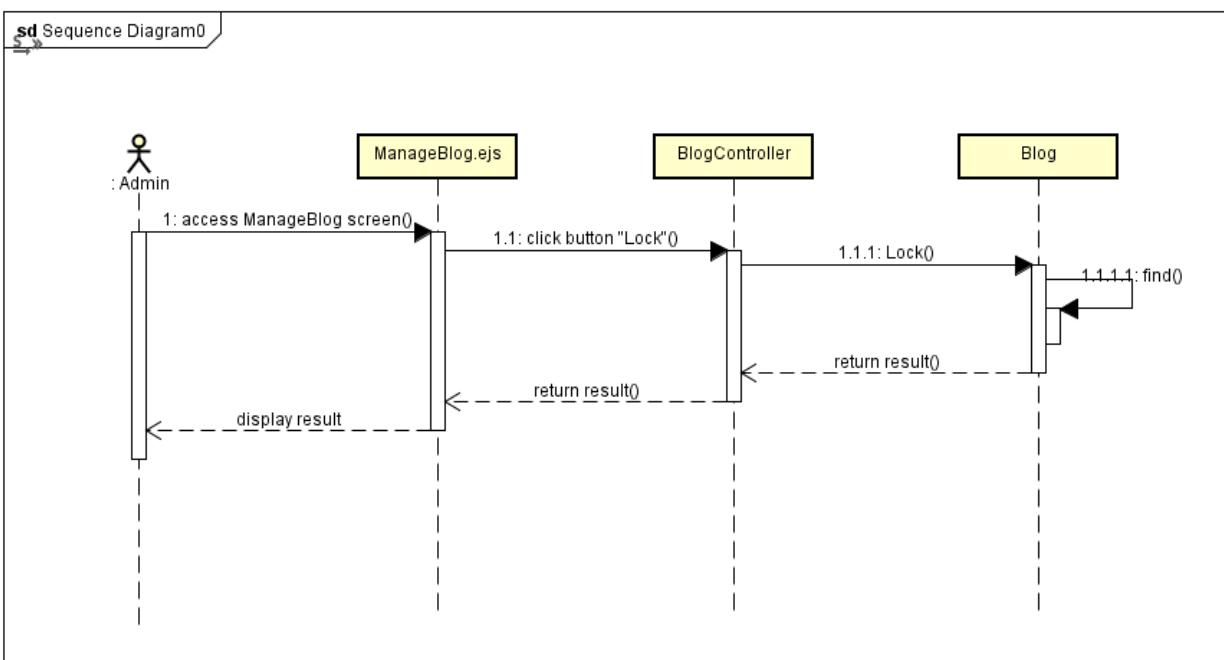
**Figure 4-69:** Insert Blog

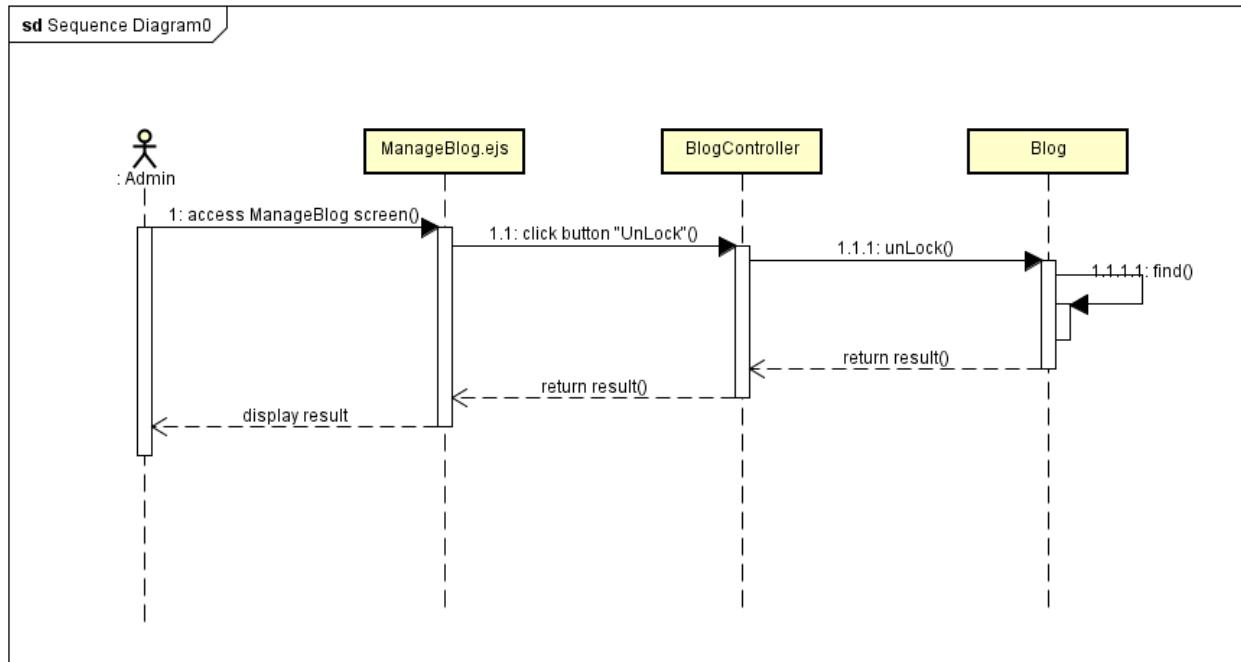
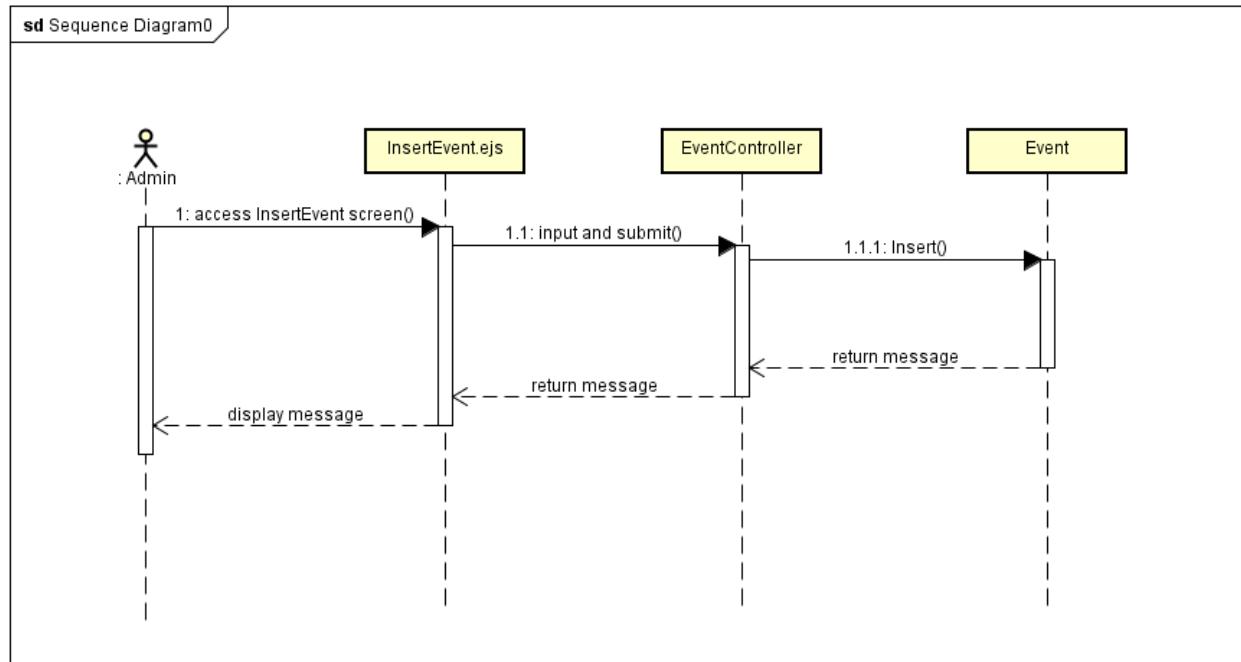
#### 4.3.1.4.4.2 Search Blog

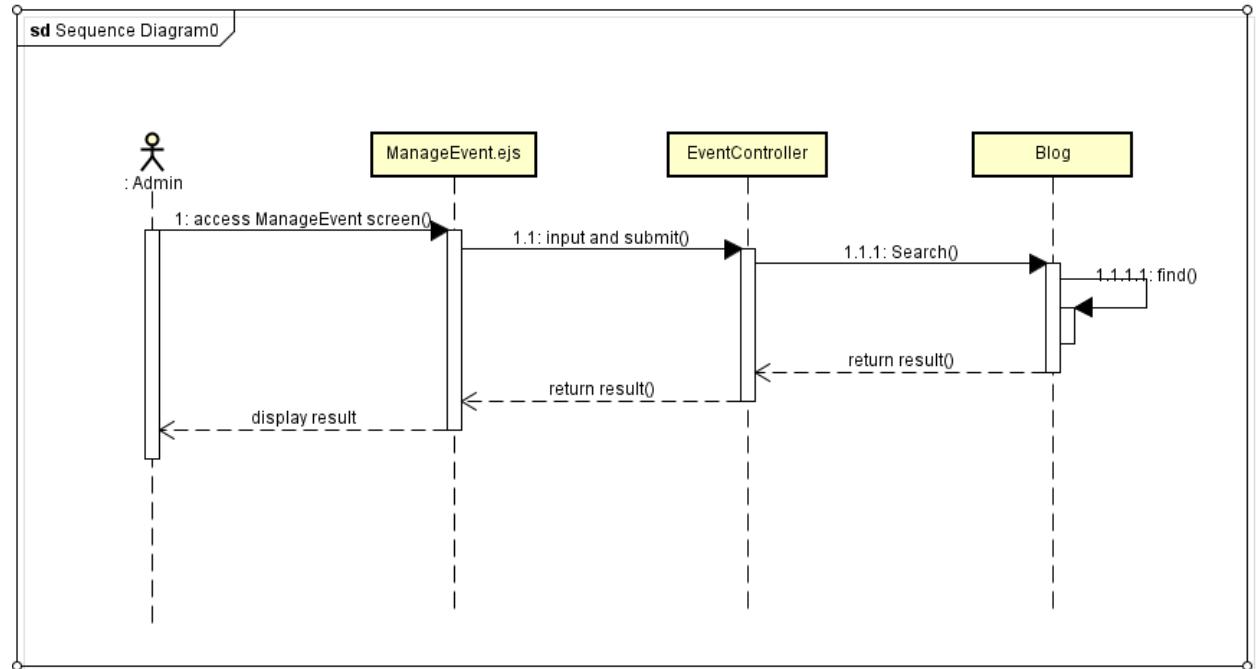
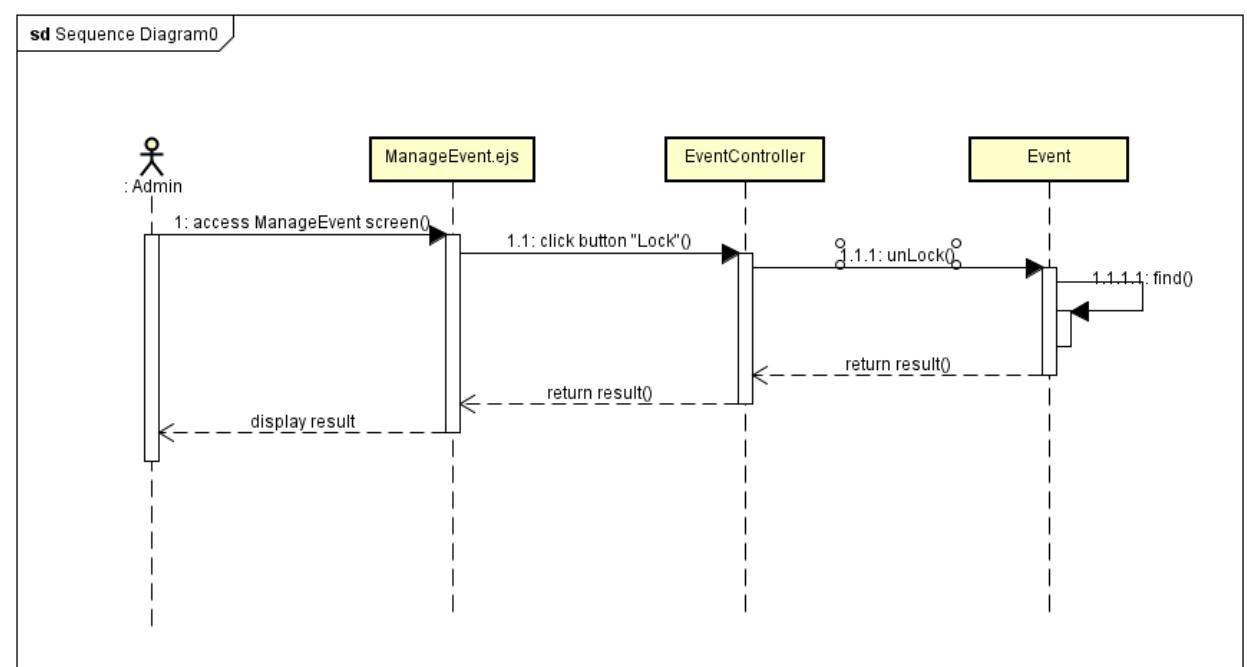


**Figure 4-70: Search Blog**

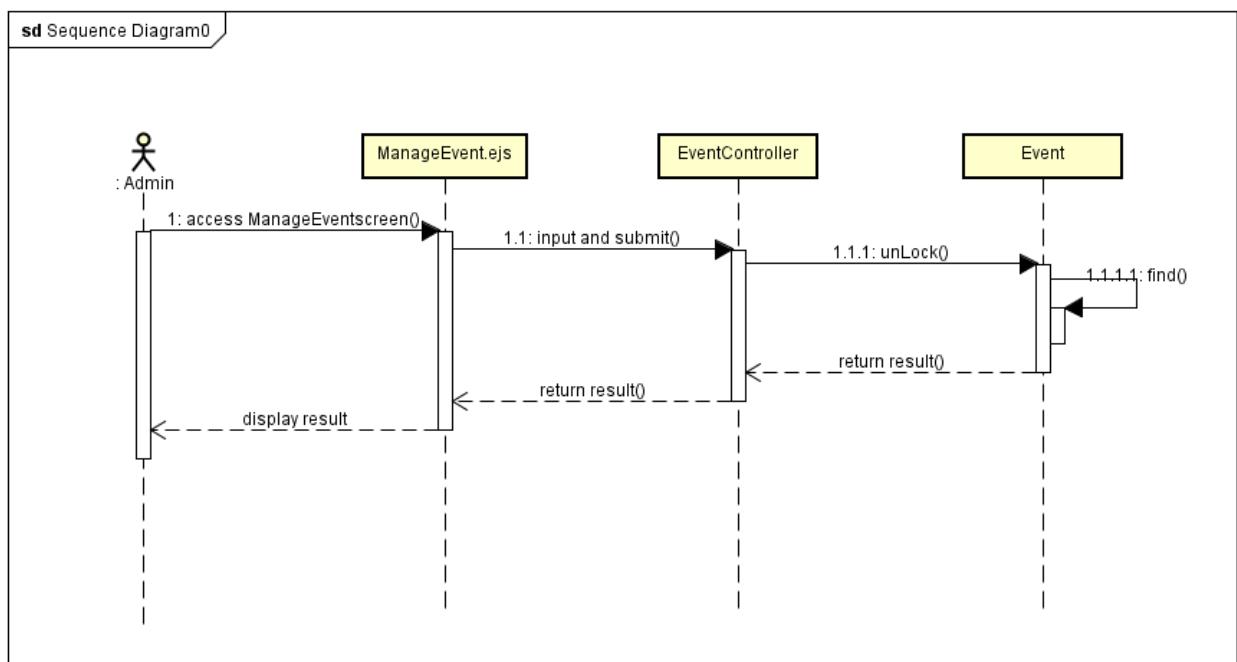
#### 4.3.1.4.4.3 Lock Blog



**Figure 4-71: Lock Blog****4.3.1.4.4.4 UnLock Blog****Figure 4-72: UnLock Blog****4.3.1.4.5 Manage Event****4.3.1.4.5.1 Insert Event**

**Figure 4-73: Insert Event****4.3.1.4.5.2 Search Event****Figure 4-74: Search Event****4.3.1.4.5.3 Lock Event****Figure 4-75: Lock Event**

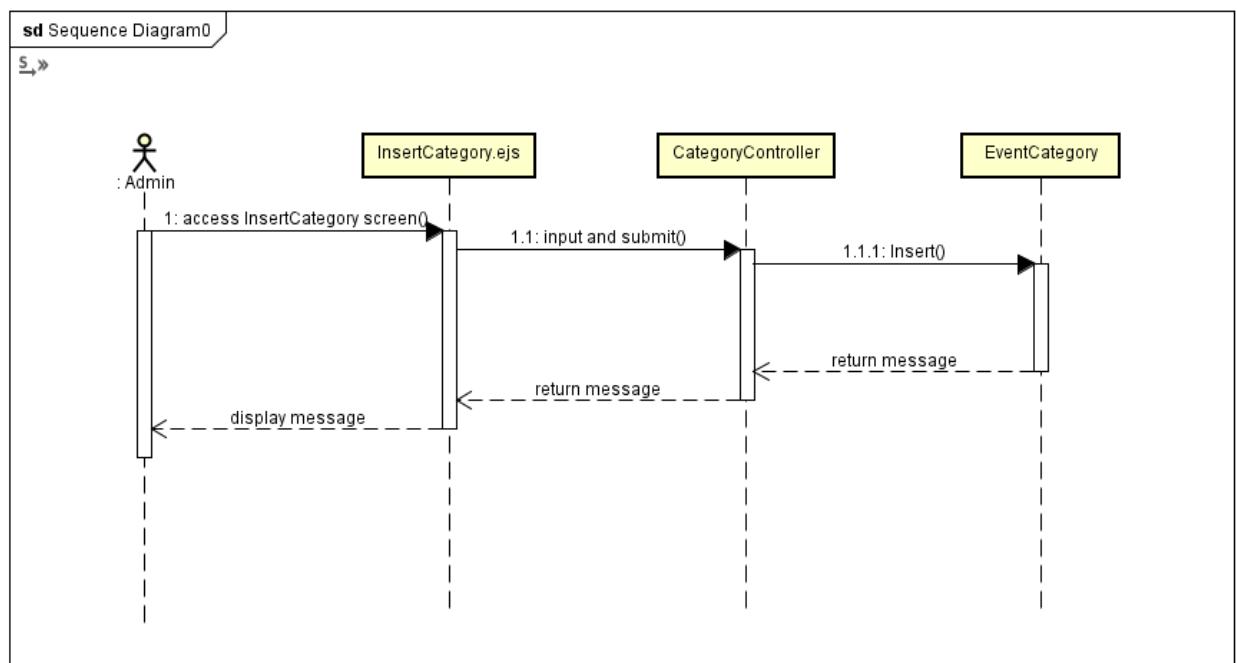
#### 4.3.1.4.5.4 Unlock Event

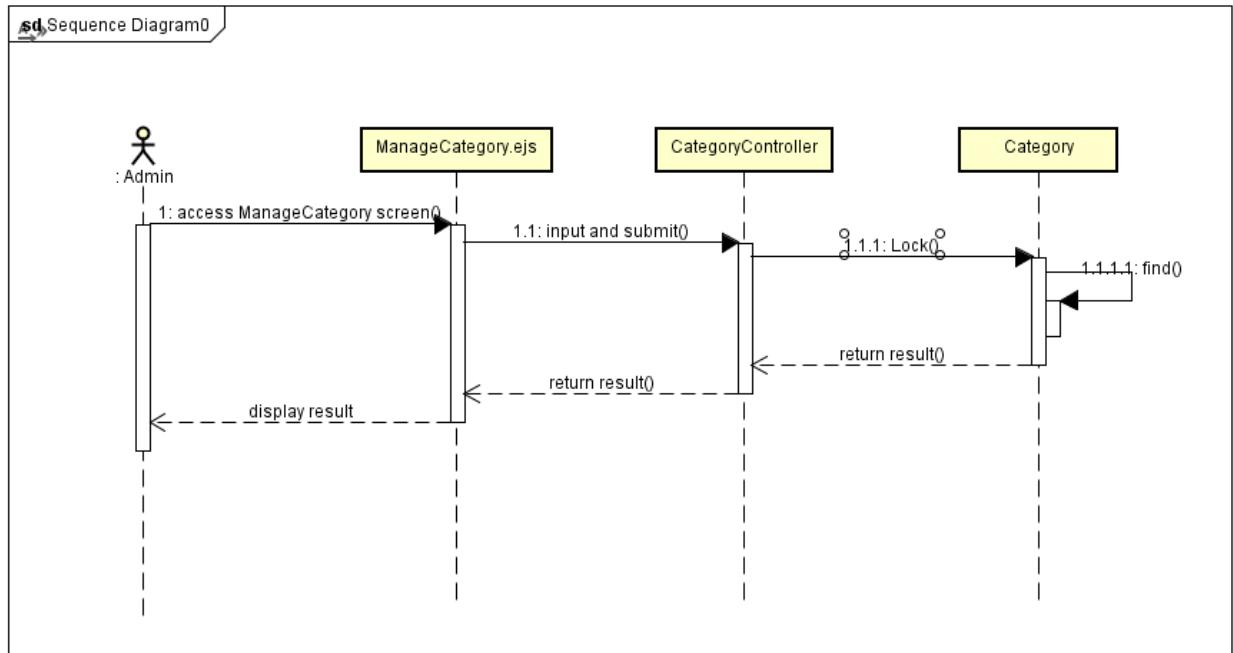
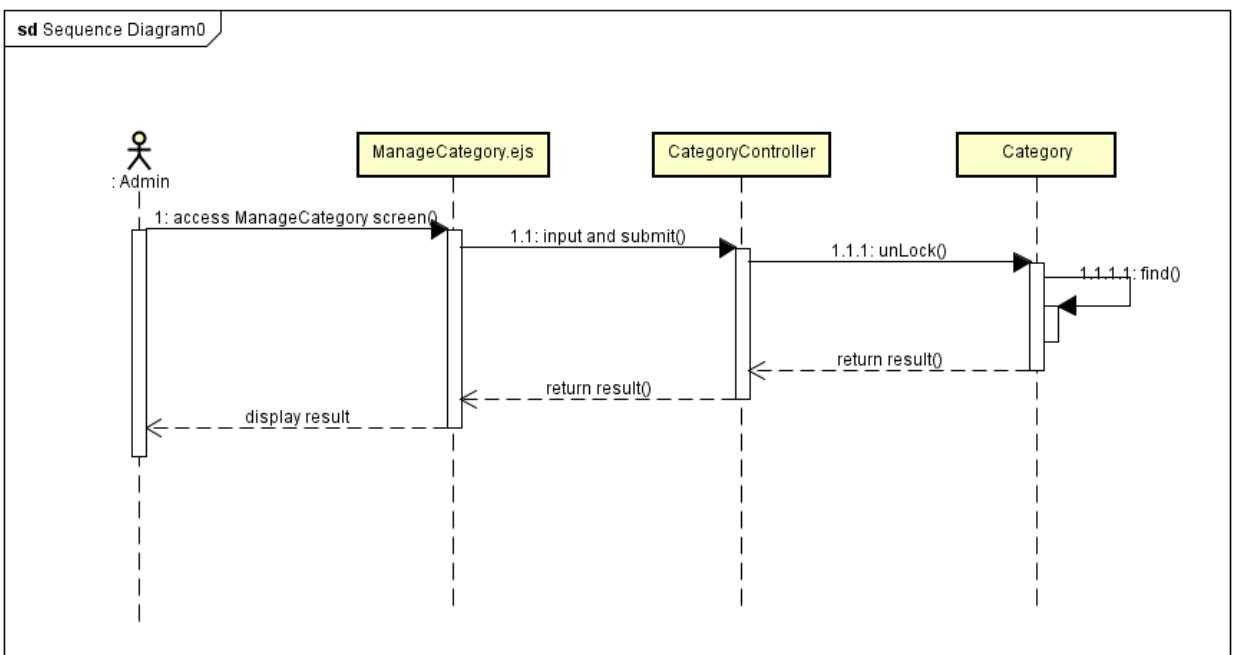


**Figure 4-76: Unlock Event**

#### 4.3.1.4.6 Manage Category

##### 4.3.1.4.6.1 Insert Category



**Figure 4-77: Insert Category****4.3.1.4.6.2 Lock Category****Figure 4-78: Lock Category****4.3.1.4.6.3 UnLock Category****Figure 4-79: UnLock Category**

#### 4.3.1.4.7 Check Out

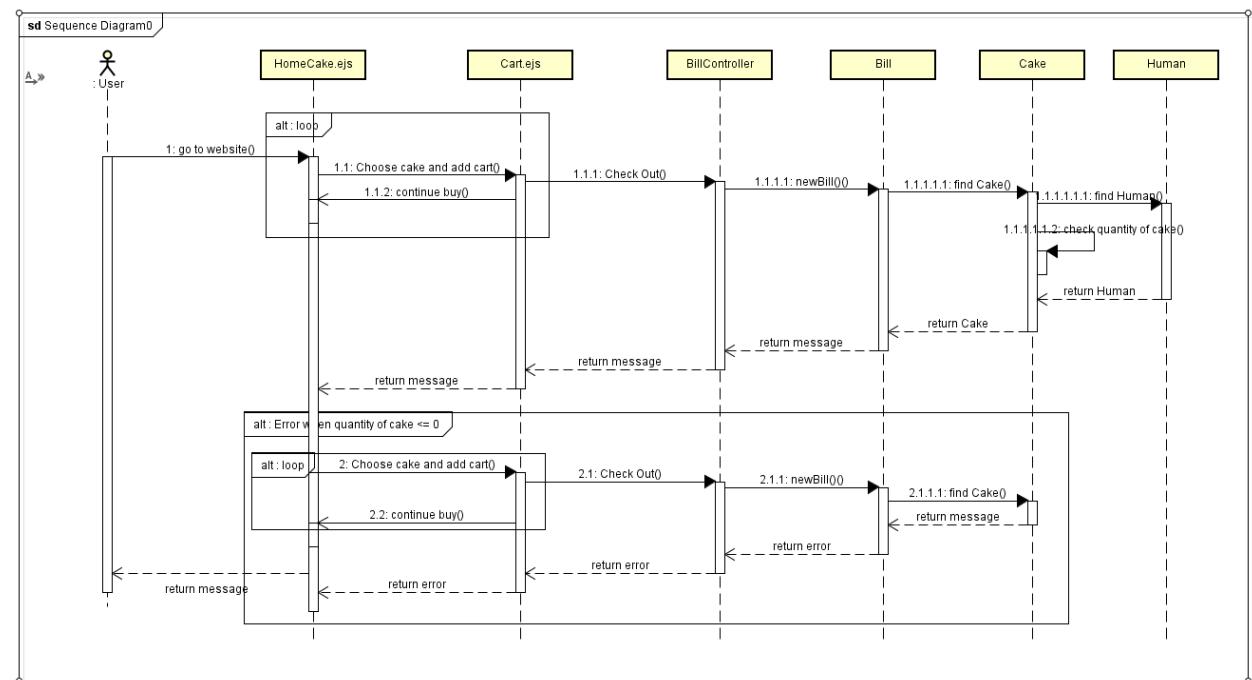
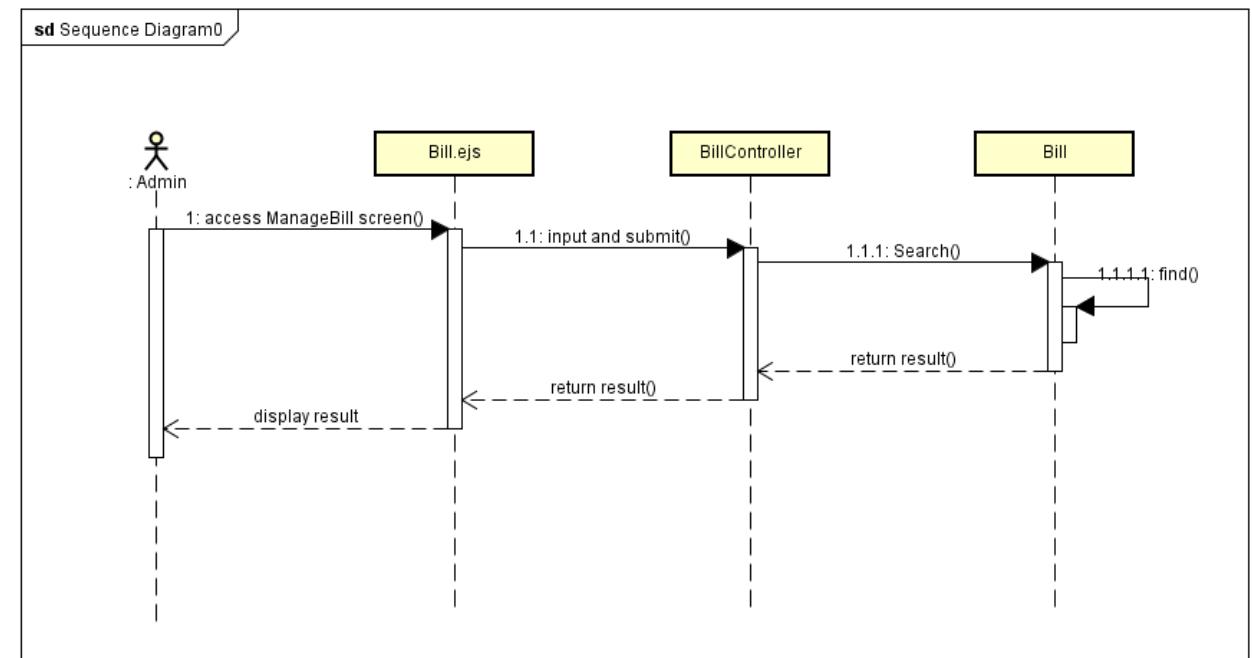
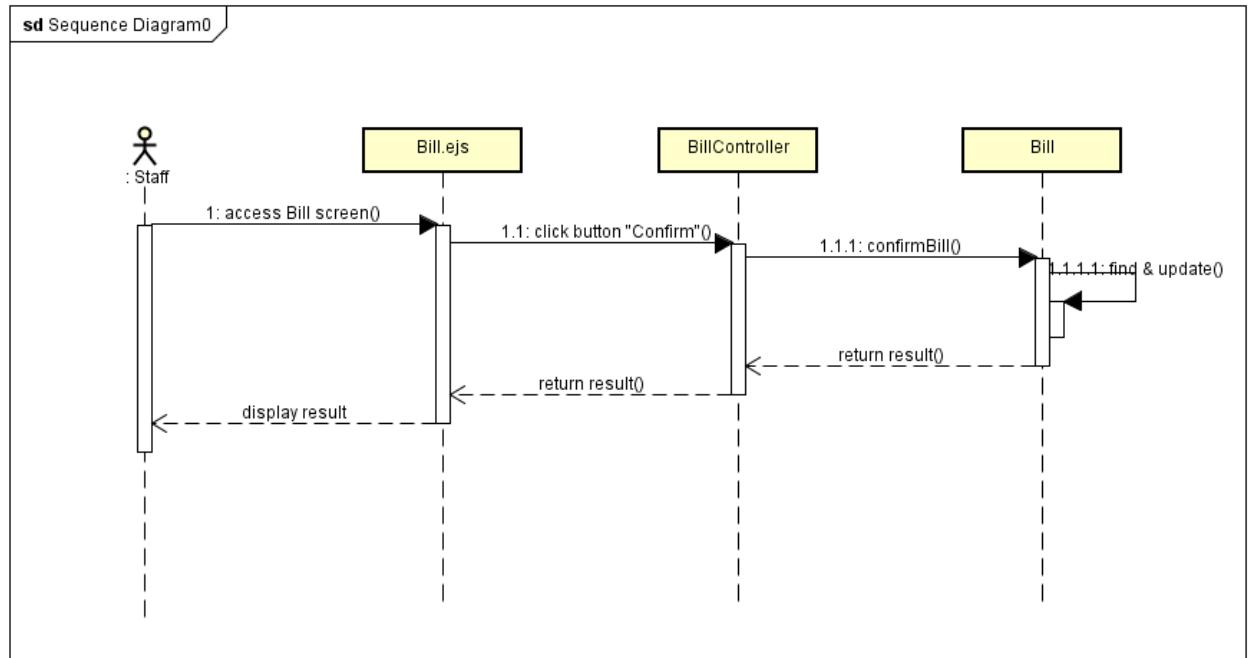
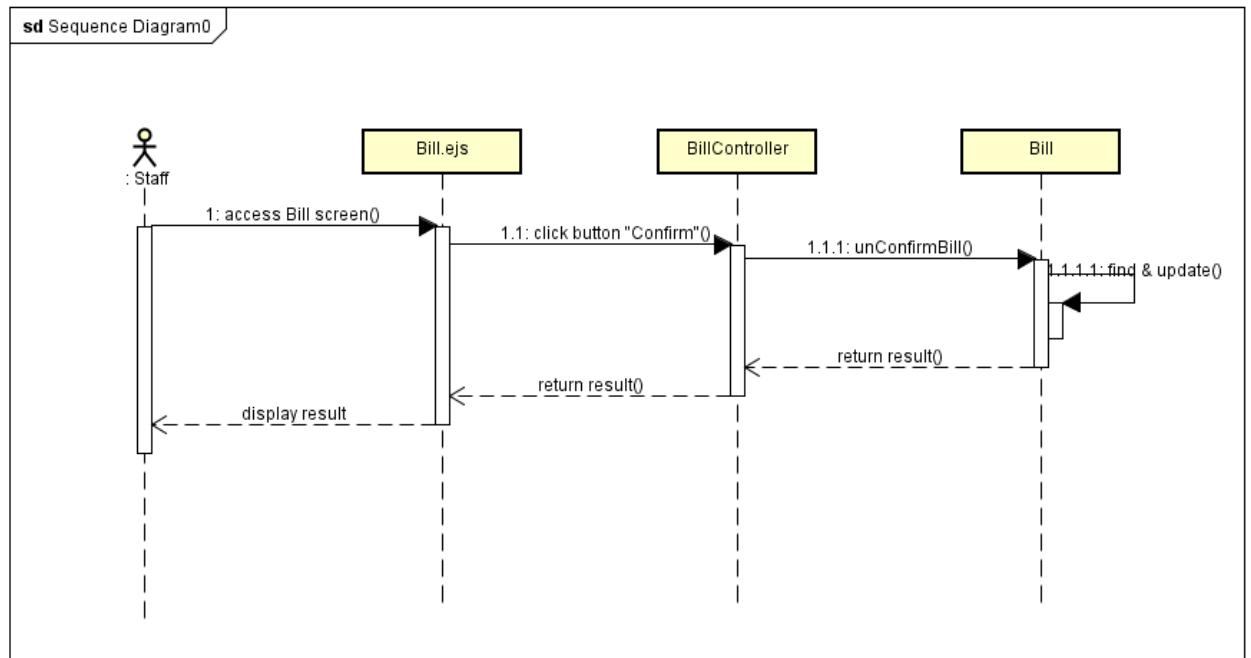


Figure 4-80: Check Out

#### 4.3.1.4.8 Manage Bill

##### 4.3.1.4.8.1 Search Bill



**Figure 4-81: Search Bill****4.3.1.4.8.2 Confirm Bill****Figure 4-82: Confirm Bill****4.3.1.4.8.3 Unconfirm Bill****Figure 4-83: Unconfirm Bill**

#### 4.3.1.4.9 Search revenue

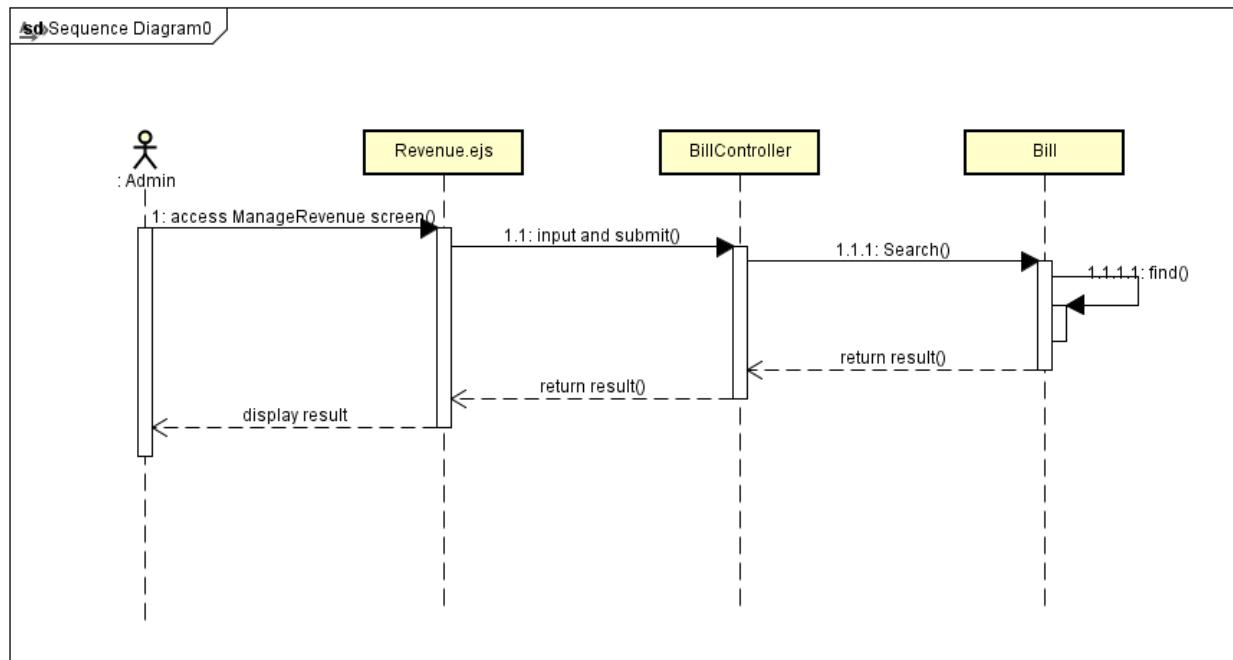


Figure 4-84: Search revenue

#### 4.3.1.4.10 Change Password

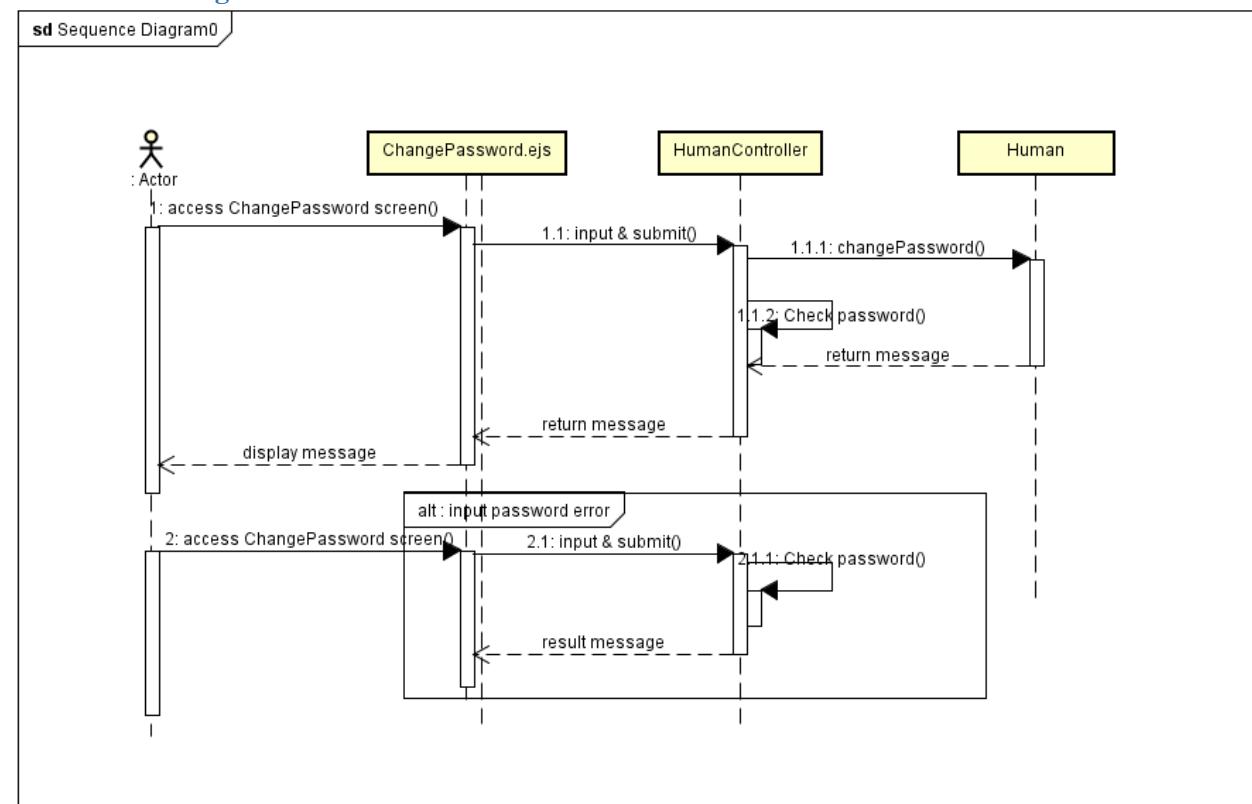


Figure 4-85: Change Password

#### 4.3.1.4.11 Update Information

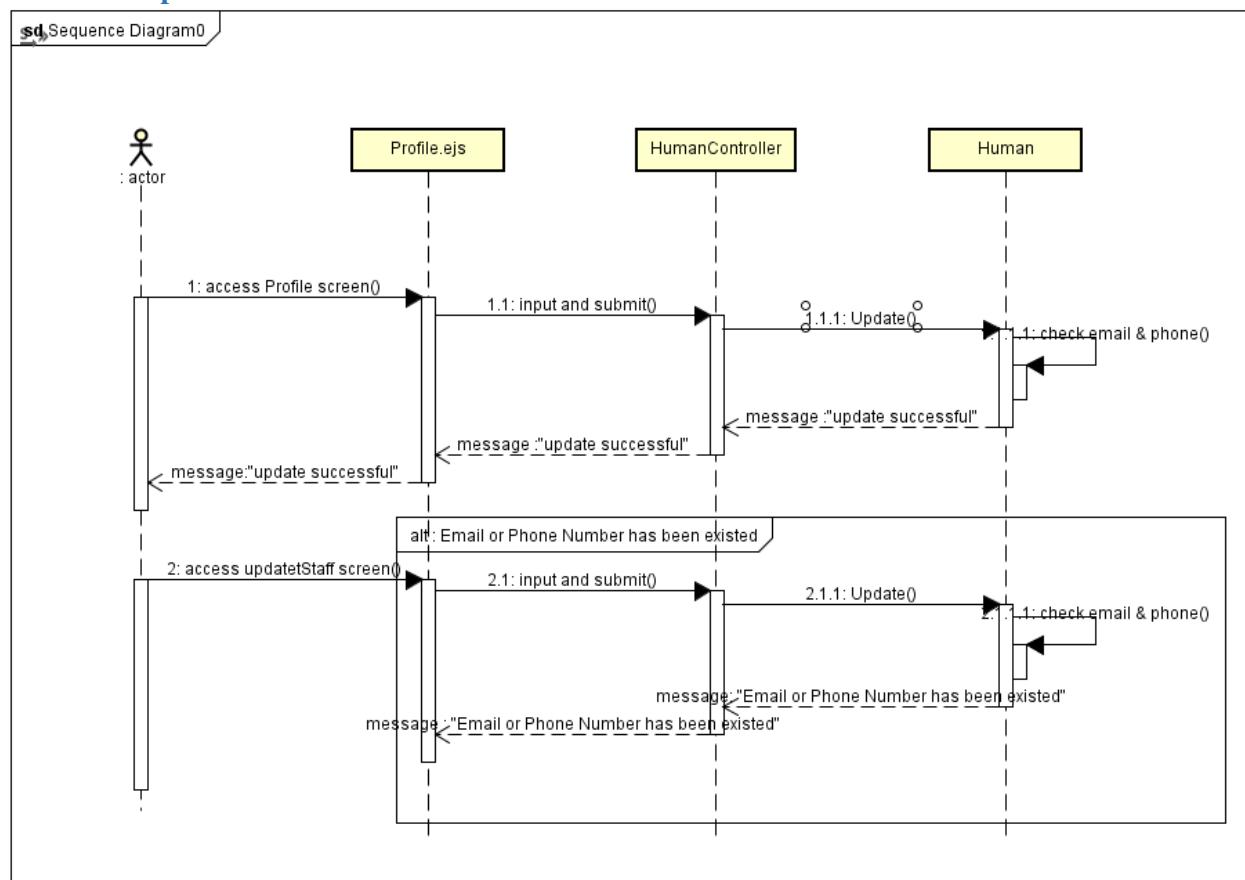


Figure 4-86: Update Information

## 4.4 Data design

### 4.4.1 Entity Relationship Diagram - ER 図

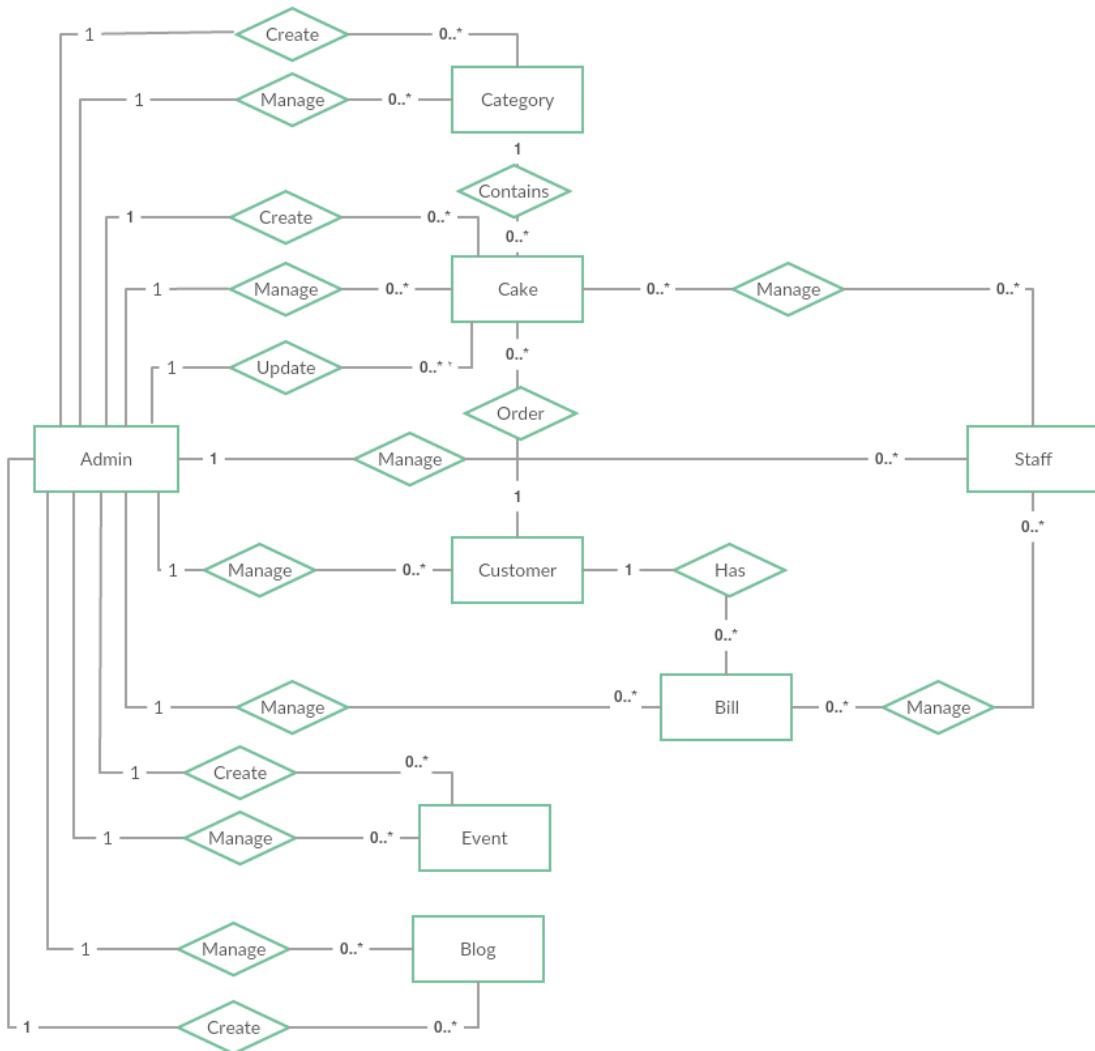


Figure 4-87: Entity Relationship Diagram

#### 4.4.1.1 Entity description

No	Entity	Description
1	Admin	Manager of the store
2	Staff	Shop staff
3	Customer	Customer is all people who use BSO
4	Event	An event created by the manager includes image content to describe
5	Cake	Cake is a kind of item
6	Blog	An blog created by the manager includes image content to describe
7	Category	A category which customer uses to filter cake

8	Bill	Managers and Customer use to review transaction history
---	------	---

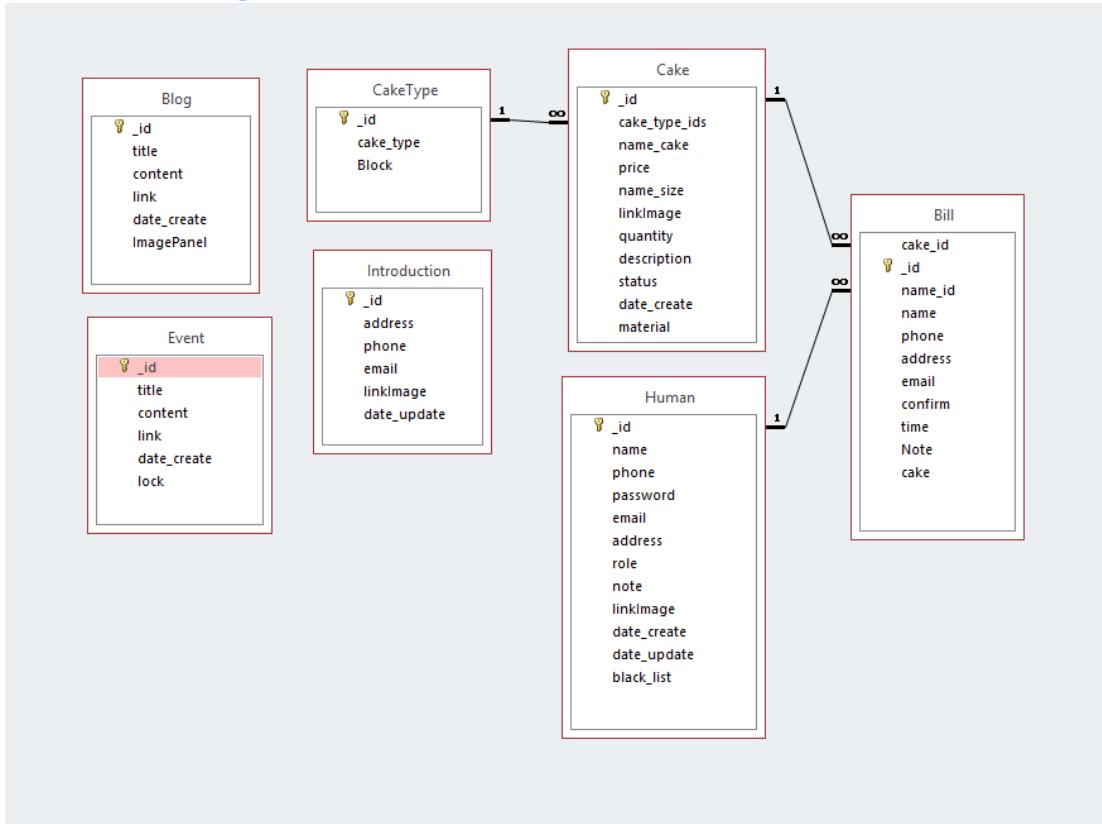
**Table 4-62: Entity description**

#### 4.4.1.2 Relationship

No	Relation	Properties	From	To	Description
1	CONTAINS	None	Category	Cake	A category contains many cakes
2	MANAGE	None	Staff	Cake	The staff used to check information about the cake
3	MANAGE	None	Staff	Bill	Staff to check their order
4	MANAGE	None	Admin	Category	The manager used to manage information about the category
5	MANAGE	None	Admin	Staff	The manager used to manage information about the staff
6	MANAGE	None	Admin	Cake	The manager used to manage information about the cake
7	MANAGE	None	Admin	Customer	The manager used to check information about the user
8	MANAGE	None	Admin	Bill	The manager used to manage information about the bill
9	MANAGE	None	Admin	Event	The manager used to manage information about the event
10	MANAGE	None	Admin	Blog	The manager used to manage information about the blog
11	CREATE		Admin	Category	Manager adds new category
12	CREATE		Admin	Cake	Manager adds new cake
13	CREATE		Admin	Event	Manager adds new event
14	CREATE		Admin	Blog	Manager adds new blog
15	UPDATE		Admin	Cake	Manager update information of cake
16	ORDER		Customer	Cake	Customer use to order cake
17	HAS		Customer	Bill	Customer to view transaction history

**Table 4-63:**  
**Relationship description**

#### 4.4.2 Table Diagram



**Figure 4-88: Table Diagram**

##### 4.4.2.1 Bill table

No	Field Name	Type	Allow Null	Description
1	<code>_id</code>	ObjectId	No	id of Bill
2	<code>cake_id</code>	ObjectId		id of Cake
3	<code>name_id</code>	ObjectId		id of Customer
4	<code>name</code>	String	No	name of Customer
5	<code>phone</code>	String	No	phone number of Customer
6	<code>address</code>	String	No	shipping address for Cusomer
7	<code>email</code>	String		email of Customer
8	<code>confirm</code>	Boolean		confirm has been shipped
9	<code>time</code>	Date	No	time order cake
10	<code>Note</code>	String		note for staff
11	<code>cake</code>	Array	No	information of order detail

**Table 4-64:** Bill table**4.4.2.2 Blog table**

No	Field Name	Type	Allow Null	Description
1	_id	ObjectId	No	id of Blog
2	title	String	No	title of Blog
3	content	String	No	content of Blog
4	link	String	No	image link of content's Blog
5	date_create	Date	No	time create blog
6	ImagePanel	String	NO	imge link of panel's Blog

**Table 4-65:** Blog table**4.4.2.3 Cake table**

No	Field Name	Type	Allow Null	Description
1	_id	ObjectId	No	id of Cake
2	cake_type_ids	ObjectId	No	id of Category
3	name_cake	String	No	name of Cake
4	price	String	No	price of Cake
5	name_size	String	No	size of Cake
6	linkImage	String	No	link Image of Cake
7	quantity	String	No	quantity of Cake exist
8	description	String	No	description about Cake
9	status	Boolean	No	status of Cake sell
10	date_create	Date	No	time create Cake
11	material	Array	No	material of Cake

**Table 4-66:** Cake table**4.4.2.4 CakeType table**

No	Field Name	Type	Allow Null	Description
1	_id	ObjectId	No	id of Category
2	cake_type	String	No	name of Category
3	Blocked	Boolean	No	status of Category

**Table 4-67:** CakeType table**4.4.2.5 Event table**

No	Field Name	Type	Allow Null	Description
1	_id	ObjectId	No	id of event
2	title	String	No	title of event
3	content	String	No	content of event
4	link	String	No	link image of event
5	date_create	Date	No	time create event
6	lock	Boolean	No	status of event

**Table 4-68:** Event table**4.4.2.6 Human table**

No	Field Name	Type	Allow Null	Description
1	_id	ObjectId	No	id of User
2	name	String	No	name of User
3	phone	String	No	phone number of User
4	password	String	No	password of User
5	email	String	No	email of User
6	address	String	No	address of User
7	role	Int	No	role of User
8	note	String		note for Manage
9	linkImage	String	No	link Image of User
10	date_create	Date	No	time create Account
11	date_update	Date	No	time update Account
12	black_list	Boolean	No	blocked Account

**Table 4-69:** Human table**4.4.2.7 Introduction table**

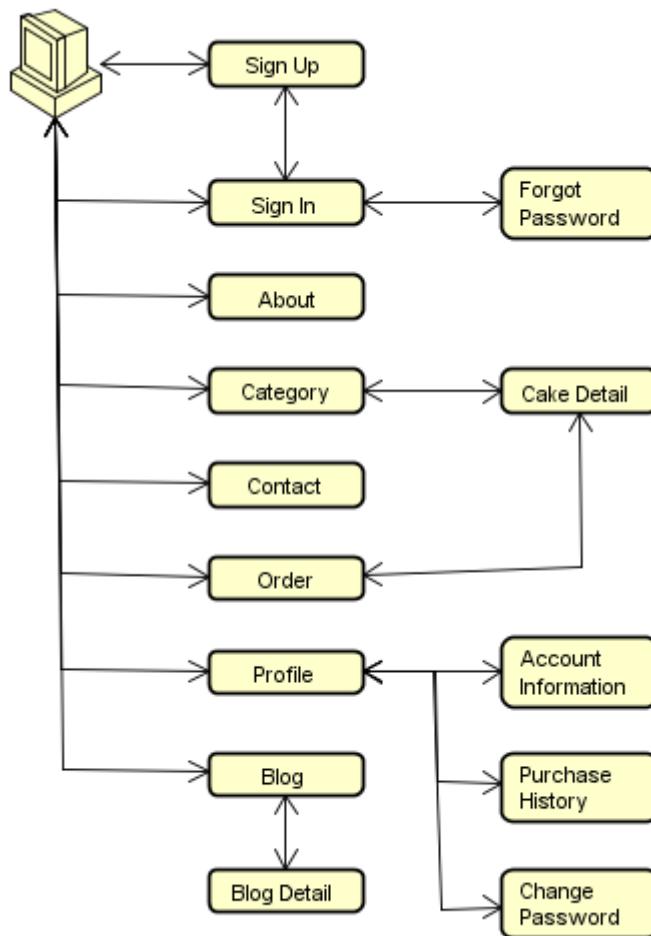
No	Field Name	Type	Allow Null	Description
1	_id	ObjectId	No	id of store
2	address	String	No	name of store

3	phone	String	No	phone number of store
4	email	String	No	password of store
5	linkImage	String	No	linkImage map of store
10	date_update	Date	No	time update Introduction

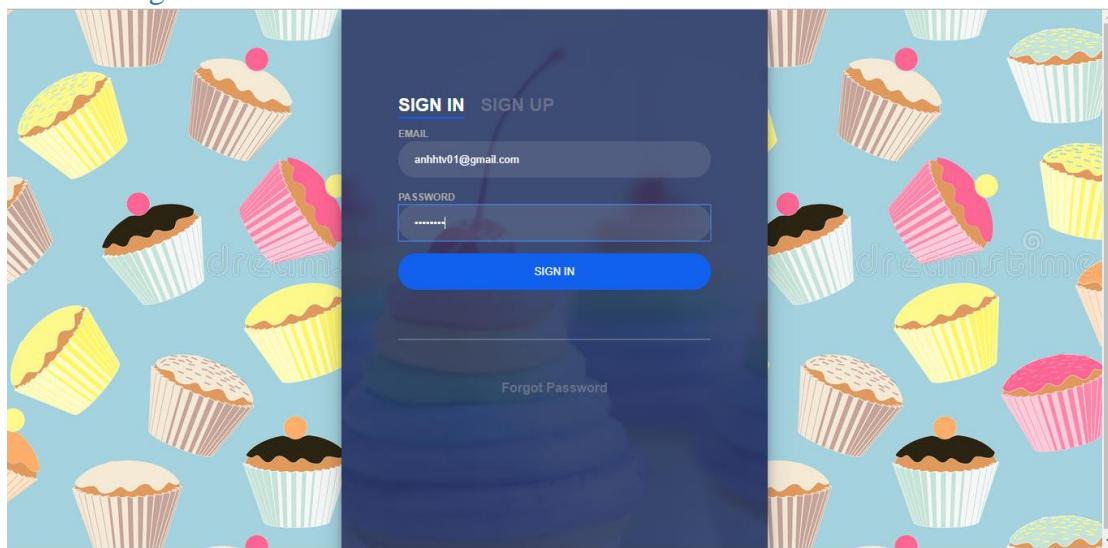
**Table 4-1: Introduction table**

## 4.5 Screen Design

### 4.5.1 User

**Figure 4-89: User screen flow**

#### 4.5.1.1 Sign In

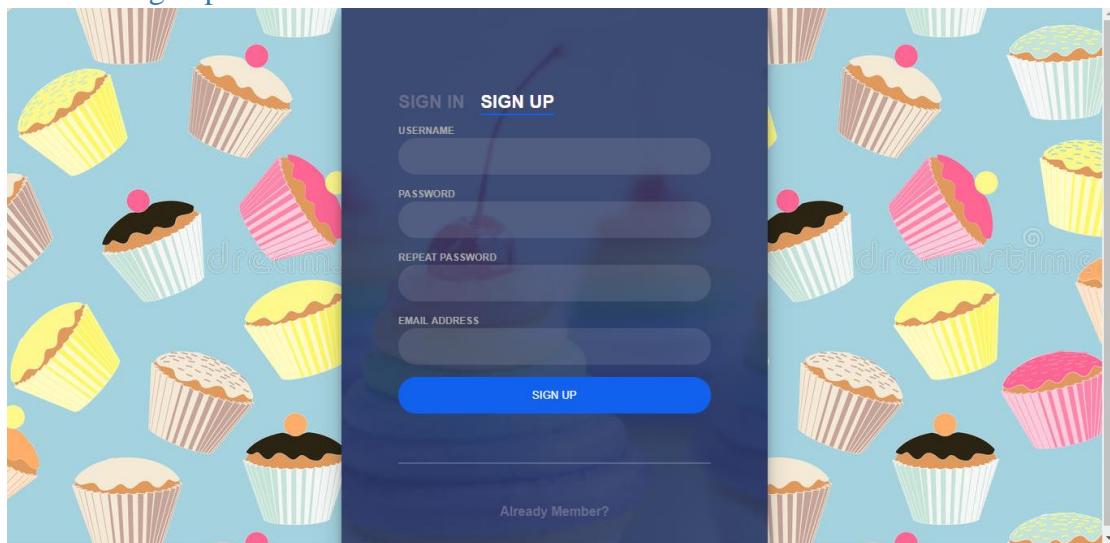


**Figure 4-90:** Login screen

No	Field name	Type	Require	Mandator	Max-Lengt	Description
1	Sign in	Label	Click	Y		From Sign in
2	Sign up	Label	Click	Y		From Sign up
3	Email	Text box	Email	Y		Email of User
4	Password	Text box	Type	Y		Password
5	Sign in	Button	Click	Y		Login
6	Forgot Password	Link	Click	Y		Forgot Password

**Table 4-70:** Register screen

#### 4.5.1.2 Sign up

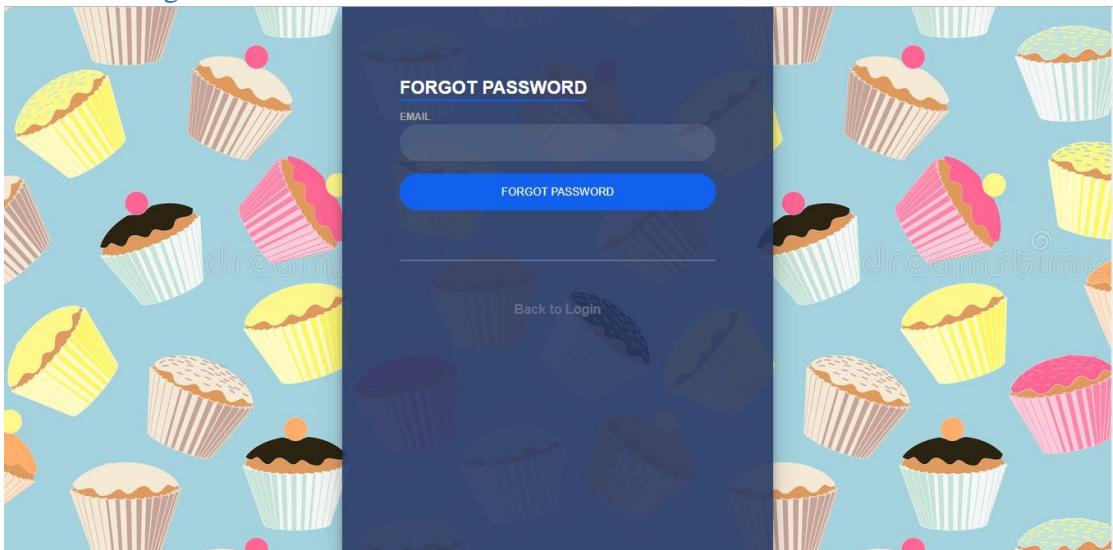


**Figure 4-91:** Sign up Screen

No	Field name	Type	Require	Mandatory	Max-Length	Description
1	Sign in	Label	Click	Y		From Sign in
2	Sign up	Label	Click	Y		From Sign up
3	UserNa me	Text		Y		User Name
4	Passwor d	Text	Password	Y		Password
5	Repeat Passwor d	Text	Password			Repeat Password
6	Email	Text	Email	Y		Email of User
7	Sign up	Button	Click	Y		Sign up
8	Already Member	Link	Click	Y		Back to Sign In

**Table 4-71:** Sign Up Screen

#### 4.5.1.3 Forgot Password

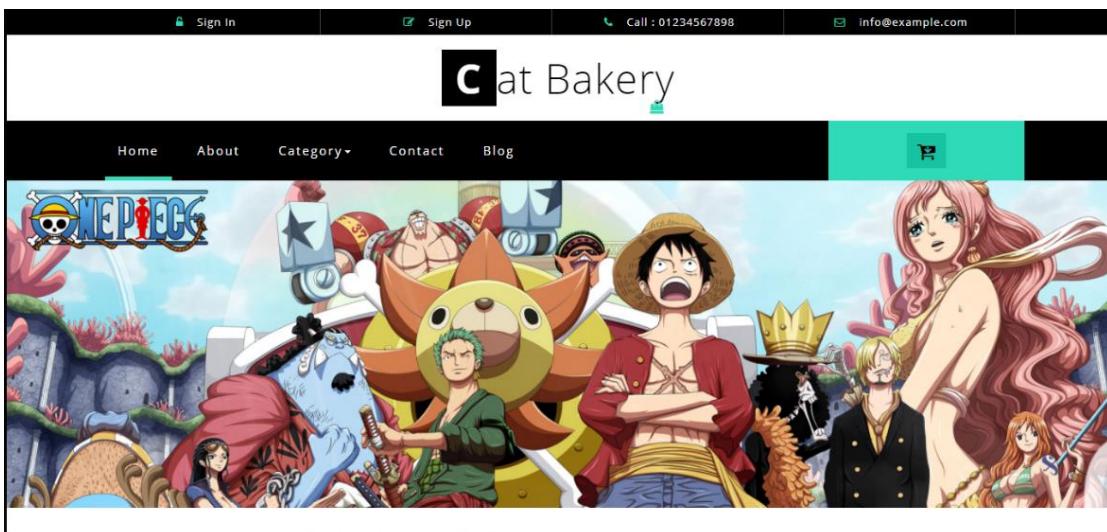


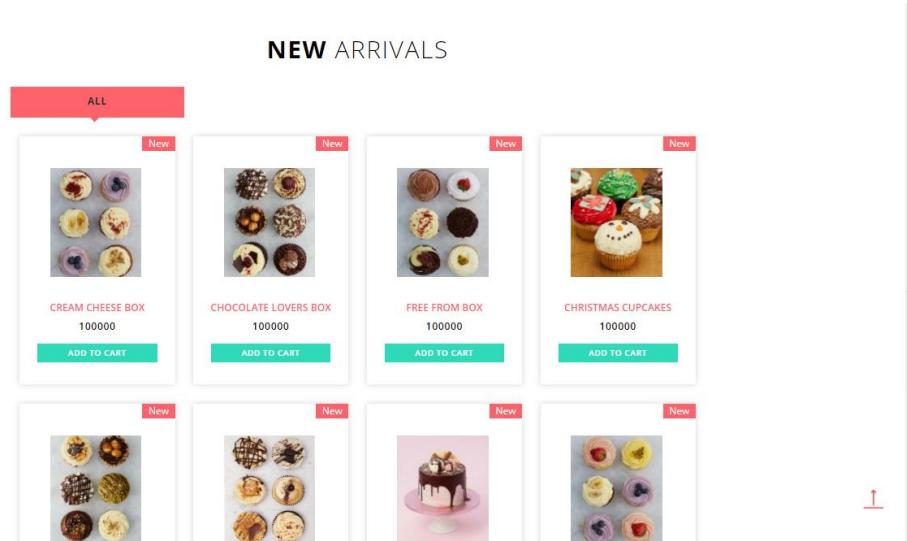
**Figure 4-92:** Forgot Password

No	Field name	Type	Require	Mandatory	Max-Length	Description
1	Email	Text box	Email	Y		Email
2	Forgot Password	Button	Click	Y		Forgot Password
3	Back to login	Link	Click	Y		Back to login

**Table 4-72:** Forgot Password

#### 4.5.1.4 Home

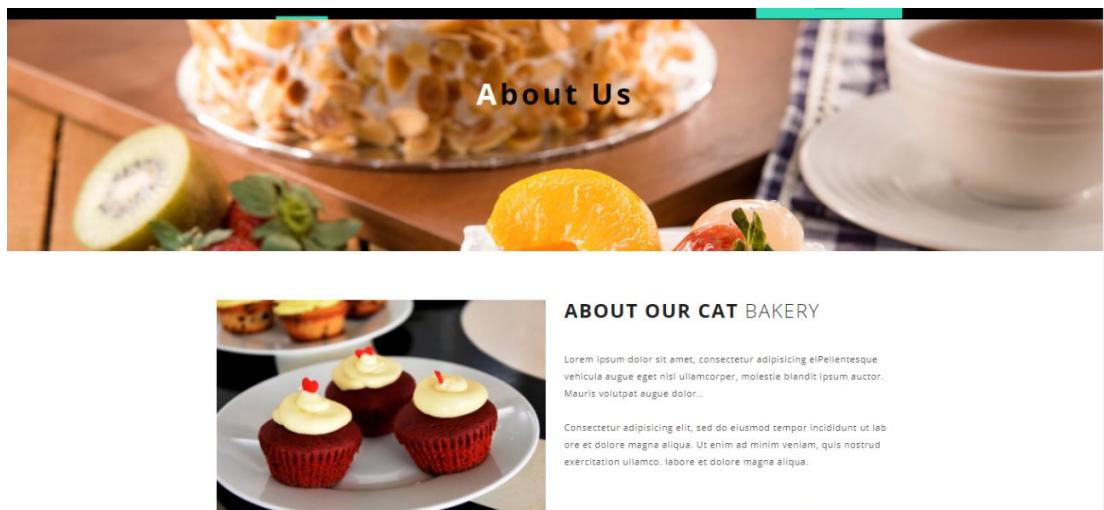


**Figure 4-93: Home screen 1****Figure 4-94: Home screen 2**

No	Field name	Type	Require	Mandator	Max-Lengt	Description
1	Home	Tab	Click	Y		Home Page
2	About	Tab	Click	Y		About Shop
3	Category	Drowd ownlist	Click	Y		Display category of cake
4	Contact	Tab	Click	Y		Contact of shop
5	Icon shopping cart	Button	Click	Y		Display shopping cart
6	Event	Panel	Click	Y		Display event detail
7	List new arrival		Click	Y		Display cake detail
8	Add to cart	Button	Click	Y		Add to shopping cart

**Table 4-73: Home Screen 2**

#### 4.5.1.5 About



**Figure 4-95:** About Us screen

No	Field name	Type	Require	Mandatory	Max-Length	Description
1	Home	Tab	Click	Y		Home Page
2	About	Tab	Click	Y		About Shop
3	Category	Drowd ownlist	Click	Y		Display category of cake
4	Contact	Tab	Click	Y		Contact of shop
5	Icon shopping cart	Button	Click	Y		Display shopping cart
6	About our cat bakery	Label		Y		About us

**Table 4-74:** About Us

#### 4.5.1.6 Category

BIRTHDAY CAKES

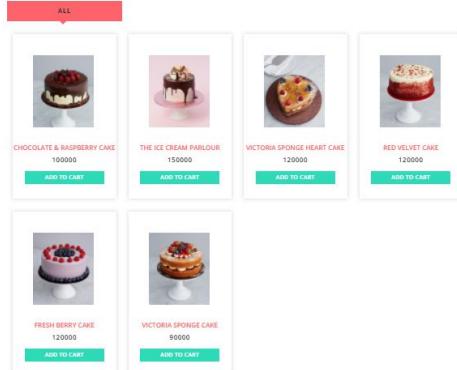
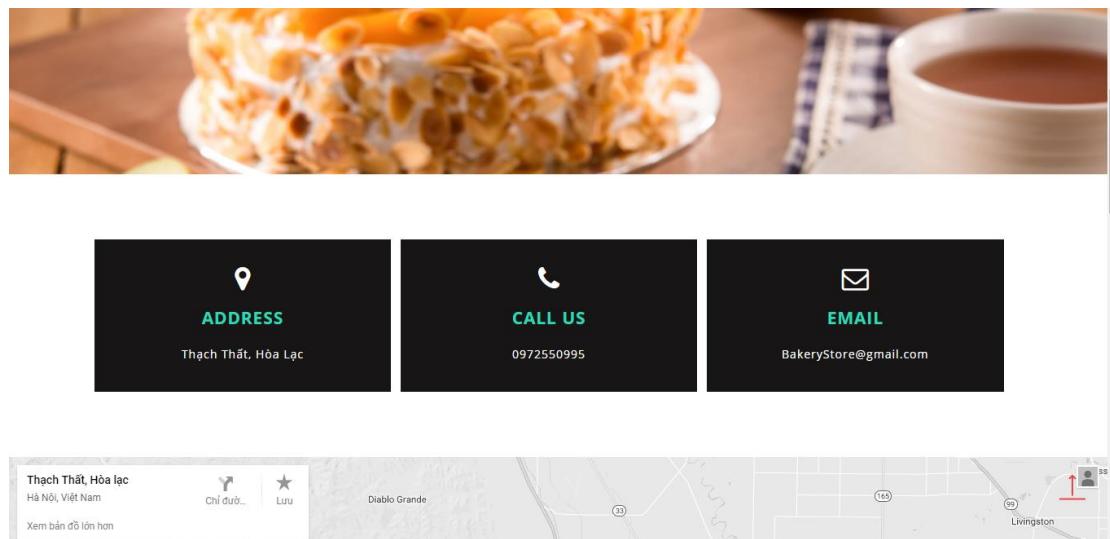


Figure 4-96: Category screen

No	Field name	Type	Requ ire	Mandator y	Max- Length	Description
1	Home	Tab	Click	Y		Home Page
2	About	Tab	Click	Y		About Shop
3	Category	Drowd ownlist	Click	Y		Display category of cake
4	Contact	Tab	Click	Y		Contact of shop
5	Icon shopping card	Button	Click	Y		Display shopping card
6	Event	Panel	Click	Y		Display event detail
7	List new arrival		Click	Y		Display cake detail
8	Add to cart	Button	Click	Y		Add to shopping cart

Table 4-75: Category

#### 4.5.1.7 Contact



**Figure 4-97: Contact Screen**

No	Field name	Type	Require	Mandatory	Max-Length	Description
1	Home	Tab	Click	Y		Home Page
2	About	Tab	Click	Y		About Shop
3	Category	Drowd ownlist	Click	Y		Display category of cake
4	Contact	Tab	Click	Y		Contact of shop
5	Icon shopping cart	Button	Click	Y		Display shopping cart
6	Contact Us	Label		Y		Contact Us

**Table 4-76: Contact**

## 4.5.1.8 Order

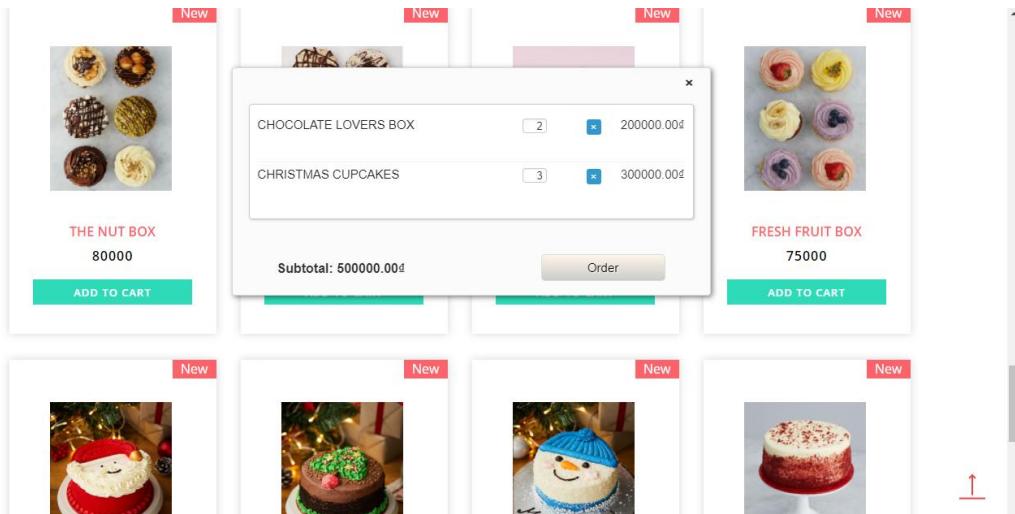


Figure 4-98: Shopping Cart Screen

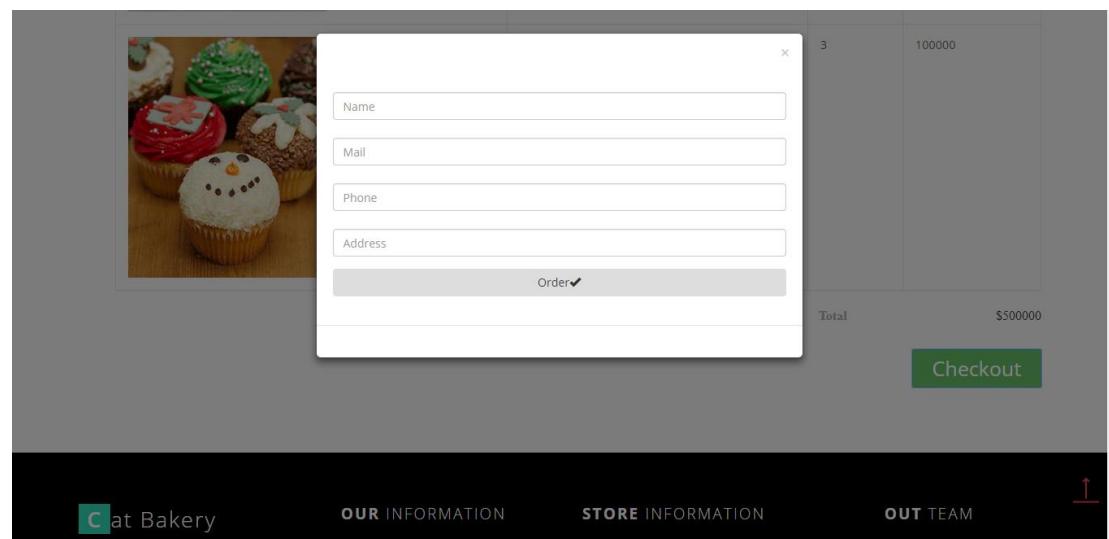
No	Field name	Type	Require	Mandatory	Max-Length	Description
1	List cake	Table		Y		List cake
2	Text box	Text		Y		Input quantity of cake
3	Icon (x)	Button	Click	Y		Remove Cake
4	Subtotal	Text		Y		Price total
5	Order	Button	Click	Y		Order

Table 4-77: Shopping Cart

The screenshot shows a detailed shopping cart view. The title 'SHOPPING CART' is at the top. Below it is a table with columns: Image, Product, Price, and Quantity. The first row shows 'CHOCOLATE LOVERS BOX' with an image of six chocolate cupcakes, a price of 2, and a quantity of 100000. The second row shows 'CHRISTMAS CUPCAKES' with an image of three decorated cupcakes, a price of 3, and a quantity of 100000. At the bottom of the table, it says 'Total \$500000'. At the very bottom right is a green 'Checkout' button.

**Figure 4-99:** Shopping Cart Detail Screen

No	Field name	Type	Require	Mandatory	Max-Length	Description
1	Home	Tab	Click	Y		Home Page
2	About	Tab	Click	Y		About Shop
3	Category	Drowd ownlist	Click	Y		Display category of cake
4	Contact	Tab	Click	Y		Contact of shop
5	Icon shopping cart	Button	Click	Y		Display shopping cart
6	List Cake	Table		Y		List Cake
7	Total	Text		Y		Price total
8	Checkout	Button	Click	Y		Check out

**Table 4-78:** Shopping Cart Detail**Figure 4-100:** Checkout Screen

No	Field name	Type	Require	Mandatory	Max-Length	Description
1	Name	Text		Y		Name of Customer
2	Mail	Text		Y		Mail of Customer
3	Phone	Text		Y		Phone of Customer
4	Address	Text		Y		Address of Customer
5	Order	Button	Click			Order

**Table 4-79:** Check out**4.5.1.9 Profile****4.5.1.9.1 Account information**

The screenshot shows a user profile page. At the top, there are two tabs: "ACCOUNT INFORMATION" (selected) and "CHANGE PASSWORD". Below the tabs, the user's name is displayed as "Hoang Van Anh" next to a circular profile picture. There is a button labeled "Edit profile picture" with a camera icon. Below the name, there are four input fields containing the user's name, email, phone number, and address. At the bottom of the form is a blue "Update" button.

**Figure 4-101: Account Information screen**

No	Field name	Type	Require	Mandatory	Max-Length	Description
1	Account Information	Tab	Click	Y		Tab Information of account
2	Purchase history	Tab	Click	Y		Tab purchase history
3	Change Password	Tab	Click	Y		Tab Change Password
4	Choose File	Button	Click	Y		Select Photo
5	Name Customer	Text		Y		Name of Customer

<b>6</b>	Email	Text		Y		Email of Customer
<b>7</b>	Phone Number	Text		Y		Phone of Customer
<b>8</b>	Address	Text		Y		Address of Customer
<b>9</b>	Update	Button	Click	Y		Update

**Table 4-80:** Account Information

#### 4.5.1.9.2 Purchase history



The screenshot shows a user profile for 'Hoang Van Anh' with a profile picture. Below the profile, there are three tabs: 'ACCOUNT INFORAMTION', 'CHANGE PASSWORD', and 'PURCHASE HISTORY'. The 'PURCHASE HISTORY' tab is selected, displaying a table of purchases. The table has columns for Cake, Price, Quantity, Date Buy, and Total. The data shows four items: THE NUT BOX (80000), VEGAN RED VELVET (50000), THE ICE CREAM PARLOUR (150000), and FRESH FRUIT BOX (75000). The total amount is 555000.

Cake	Price	Quantity	Date Buy	Total
+THE NUT BOX +VEGAN RED VELVET +THE ICE CREAM PARLOUR +FRESH FRUIT BOX	80000 50000 150000 75000	1 2 1 3	Sun Dec 10 2017	555000

**Figure 4-102:** Purchase History Screen

No	Field name	Type	Requ ire	Mandator y	Max-Lengt h	Description
<b>1</b>	Account Informati on	Tab	Click	Y		Information of account
<b>2</b>	Purchase History	Tab	Click	Y		Purchase History
<b>3</b>	Change Passwor d	Tab	Click	Y		Change Password
<b>4</b>	List Purchase History	Table		Y		List Purchase History

**Table 4-81:** Purchase History

#### 4.5.1.9.3 Change Password

The screenshot shows a user interface for changing a password. At the top, there are two tabs: 'ACCOUNT INFORMATION' and 'CHANGE PASSWORD'. The 'CHANGE PASSWORD' tab is active. On the left, there's a circular profile picture of a person and the name 'Hoang Van Anh'. To the right, there are three input fields: 'Current Password', 'New Password', and 'Confirm Password'. Below these fields is a blue 'Change' button.

**Figure 4-103: Change Password screen**

No	Field name	Type	Require	Mandatory	Max-Length	Description
1	Old Password	Text		Y		Old Password
2	New Password	Text		Y		New Password
3	Repeat Password	Text		Y		Repeat Password
4	Change Password	Button	Click	Y		Change Password

**Table 4-82: Change Password**

#### 4.5.1.10 Blog

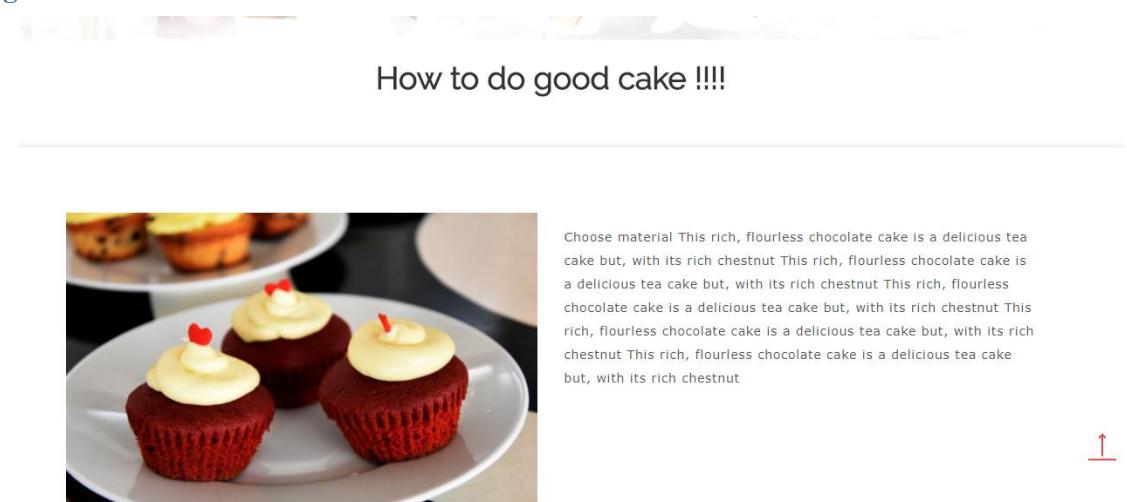


**Figure 4-104: Blog Screen**

No	Field name	Type	Require	Mandatory	Max-Length	Description
1	List Blog	List		Y		List blog
2	Read more	Button	Click	Y		View Detail

**Table 4-83: Blog**

#### 4.5.1.10.1 Blog Detail

**Figure 4-105: Blog Detail**

#### 4.5.2 ADMIN

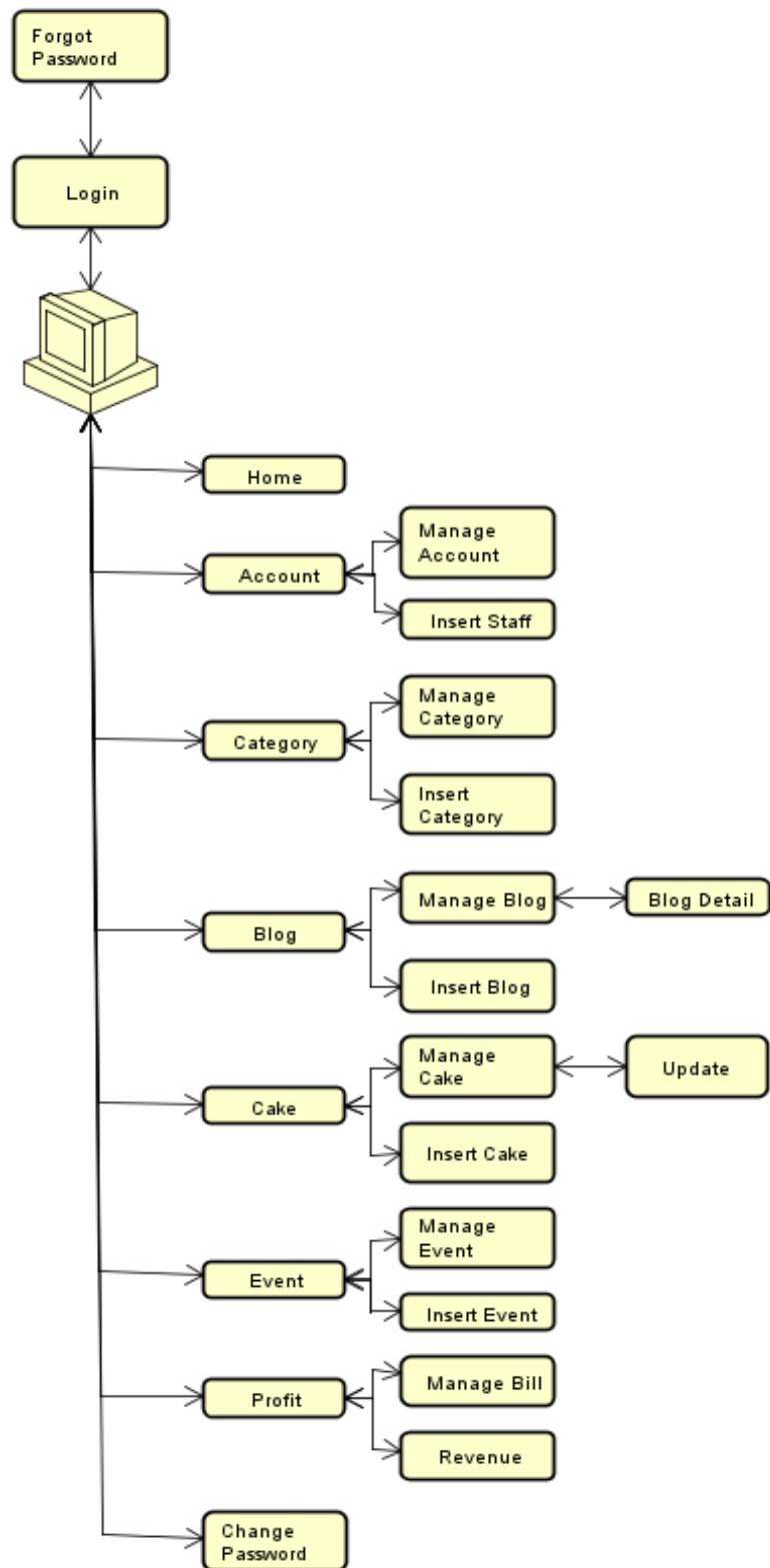
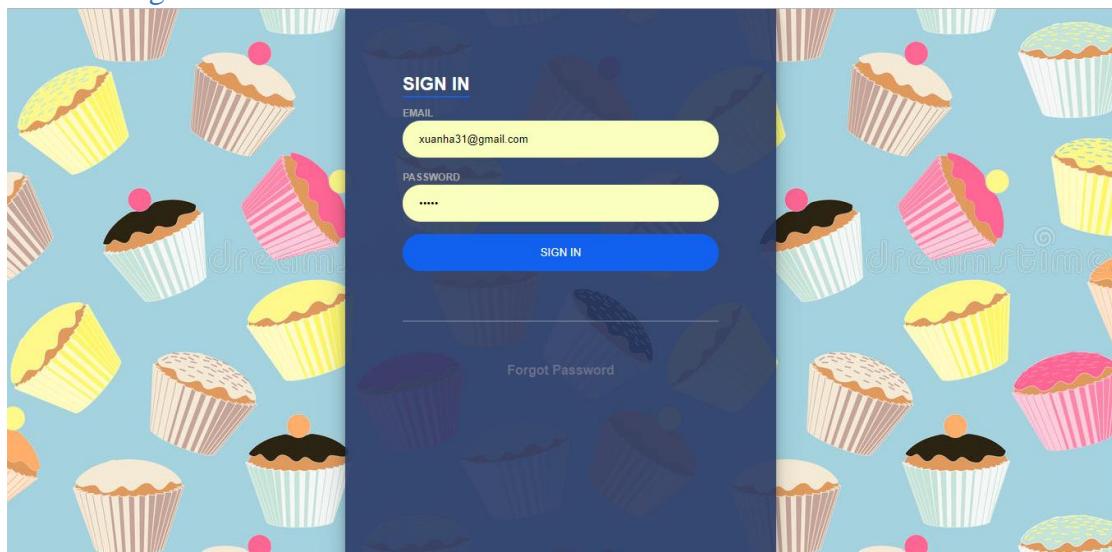


Figure 4-106: Admin screen flow

#### 4.5.2.1 Sign In

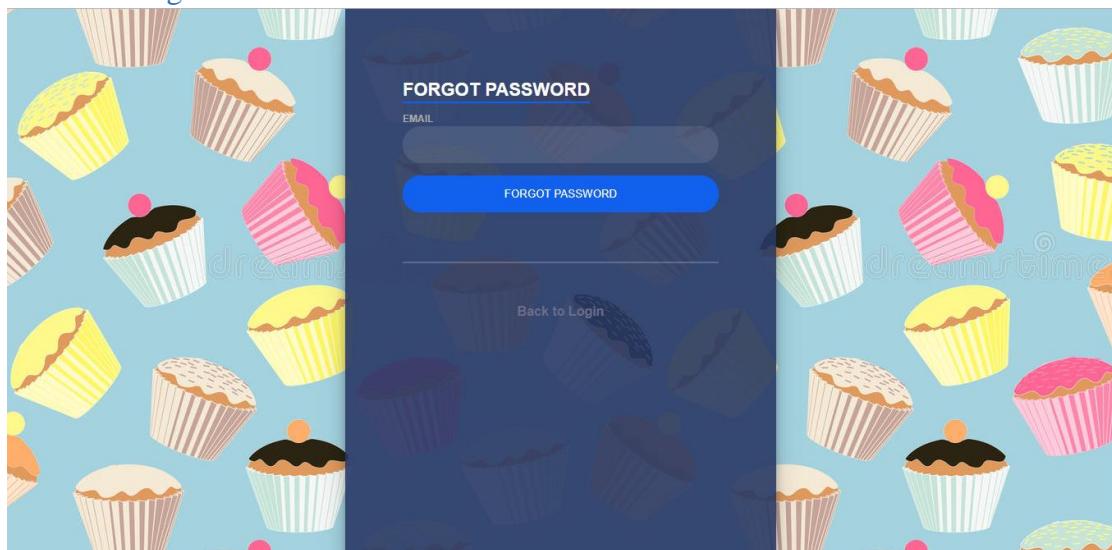


**Figure 4-107:** Login screen

No	Field name	Type	Require	Mandator y	Max-Lengt h	Description
1	Email	Text box	Email	Y		Email of Admin
2	Password	Text box	Type	Y		Password of Admin
3	Sign in	Button	Click	Y		Login
4	Forgot Password	Link	Click	Y		Forgot Password

**Table 4-84:** Login

#### 4.5.2.2 Forgot Password



**Figure 4-108:** Forgot Password screen

No	Field name	Type	Require	Mandatory	Max-Lengt h	Description
1	Email	Text box	Email	Y		Email
2	Forgot Password	Button	Click	Y		Forgot Password
3	Back to login	Link	Click	Y		Back to login

**Table 4-85:** Forgot Password

#### 4.5.2.3 Account

##### 4.5.2.3.1 Manage Account

Email	Name Customer	Address	Phone Number	Date Create	Date Update	Role	Block
kiennt@gmail.com	Kien	Ha Noi	01234556781	Mon Nov 13 2017	Sat Dec 09 2017	Customer <input checked="" type="checkbox"/>	<button>Lock</button> <button>UnLock</button>
anhhtv01@gmail.com	Tuan	Phu Tho	0987455644	Mon Nov 13 2017	Sat Dec 09 2017	Customer <input checked="" type="checkbox"/>	<button>Lock</button> <button>UnLock</button>
duc@gmail.com	Duc12	Hai Duong	098555512	Mon Nov 13 2017	Sat Dec 09 2017	Staff <input type="checkbox"/>	<button>Lock</button> <button>UnLock</button> <button>Update</button>
tuyenla@gmail.com	TuyenLA	Phu Tho	0855456323	Mon Nov 13 2017	Sat Dec 09 2017	Staff <input type="checkbox"/>	<button>Lock</button> <button>UnLock</button> <button>Update</button>
xiatian2902@gmail.com	Toan	Muong La - Son La	0868774057	Tue Nov 21 2017	Sat Dec 09 2017	Staff <input type="checkbox"/>	<button>Lock</button> <button>UnLock</button> <button>Update</button>
tuanca@gmail.com	Cao Anh Tuan	Thach That - Hà Nội	01668658546	Wed Nov 22 2017	Wed Nov 22 2017	Staff <input type="checkbox"/>	<button>Lock</button> <button>UnLock</button> <button>Update</button>

Figure 4-109: Manage Account screen

No	Field name	Type	Require	Mandatory	Max-Length	Description
1	Search	Input Text	Input Phone number	Y		Search by phone number of account
2	Icon Search	Button	Click	Y		Icon Search
3	Choose role you want	Drop Downlist	Click	Y		Search by role
4	List Account	Table		Y		List Account
5	Block	Check box		Y		Block account
6	Lock	Button	Click	Y		Button Lock Account
7	Unlock	Button	Click	Y		Button Unlock Account

**Table 4-86: Table  
2.13.1: Manage Account**

#### 4.5.2.3.2 Insert Staff

The screenshot shows the AdminLTE administrative interface. On the left, there is a dark sidebar with a user profile picture and the word 'Admin'. Below it, the 'MAIN NAVIGATION' includes links for Home, Account (which is currently selected), Category, Blog, Cake, Event, and Profit. The 'Account' link has two sub-options: 'Manage Account' and 'Insert Staff', with 'Insert Staff' being the active one. The main content area has a light blue header with the title 'Insert Staff'. Below the header, there are several input fields: 'Staff Name' (with a red asterisk indicating it's required), 'Image' (with a 'Choose File' button and a message 'No file chosen'), 'Password' (with a red asterisk), 'Address' (with a red asterisk), 'Phone Number' (with a red asterisk), 'Email' (with a red asterisk), and 'Note'. At the bottom of the form are two buttons: 'Cancel' (in red) and 'Add' (in blue).

**Figure 4-110: Insert Staff screen**

No	Field name	Type	Require	Mandatory	Max-Length	Description
1	Staff name	Text		Y		Staff name
2	Password	Text		Y		Password of staff
3	Address	Text		Y		Address of staff
4	Phone Number	Text		Y		Phone Number of Staff
5	Email	Text		Y		Email of staff
6	Note	Text		N		Other information
7	Choose File	Button	Click	Y		Select photo
8	Cancel	Button	Click	Y		Cancel and back manage account page
9	Add	Button	Click	Y		Add Staff

**Table 4-87: Insert Staff**

#### 4.5.2.4 Category

##### 4.5.2.4.1 Manage Category

Category Name	Status	Lock	UnLock
Bánh trung Thu	Status : UnLock	<button>Lock</button>	<button>UnLock</button>
Bánh Truyền Thống	Status : UnLock	<button>Lock</button>	<button>UnLock</button>
Adc	Status : UnLock	<button>Lock</button>	<button>UnLock</button>
123	Status : Lock	<button>Lock</button>	<button>UnLock</button>
asdad	Status : UnLock	<button>Lock</button>	<button>UnLock</button>

Figure 4-111: Manage Category screen

No	Field name	Type	Require	Mandatory	Max-Length	Description
1	List Category	Table		Y		List Category
2	Lock	Button		Y		Lock Category
3	UnLock	Button		Y		UnLock Category

Table 4-88: Manage Category

##### 4.5.2.4.2 Insert Category

**Figure 4-112: Insert Category screen**

No	Field name	Type	Require	Mandatory	Max-Length	Description
1	Category Name	Text		Y		List projects
2	Cancle	Button	Click	Y		Cancle and back to manage category
3	Post	Button	Click	Y		Post Category

**Table 4-89: Insert Catgeory**

#### 4.5.2.5 Blog

##### 4.5.2.5.1 Manage Blog

Title	Time Post	Status	Lock	Unlock
Cách làm bánh trung thu tuyệt vị trà xanh	Sun, 10 Dec 2017 13:41:26 GMT	UnLock	<button>Lock</button>	<button>Unlock</button>
Cách làm bánh trung thu rau câu vị sầu riêng cực độc	Sat, 09 Dec 2017 08:14:39 GMT	UnLock	<button>Lock</button>	<button>Unlock</button>
Cách làm bánh trung thu thạch đậu nành sương sáo đặc đáo	Fri, 08 Dec 2017 20:59:08 GMT	UnLock	<button>Lock</button>	<button>Unlock</button>
Cách làm bánh trung thu hình con vật ngộ nghĩnh dành tặng bé	Fri, 08 Dec 2017 20:58:40 GMT	UnLock	<button>Lock</button>	<button>Unlock</button>
Cách làm sữa gạo Hàn Quốc ngon đúng chuẩn	Fri, 08 Dec 2017 16:52:44 GMT	UnLock	<button>Lock</button>	<button>Unlock</button>

Figure 4-113: Manage Blog screen

No	Field name	Type	Require	Mandatory	Max-Length	Description
1	Search	Text		Y		Search by name blog
2	Search (icon)	Button	Click	Y		Icon search
3	List Blog	List	Click	Y		List blog (click to view detail blog)
4	Lock	Button	Click	Y		Lock blog
5	UnLock	Button	Click	Y		Unlock blog

Table 4-90: Manage Blog

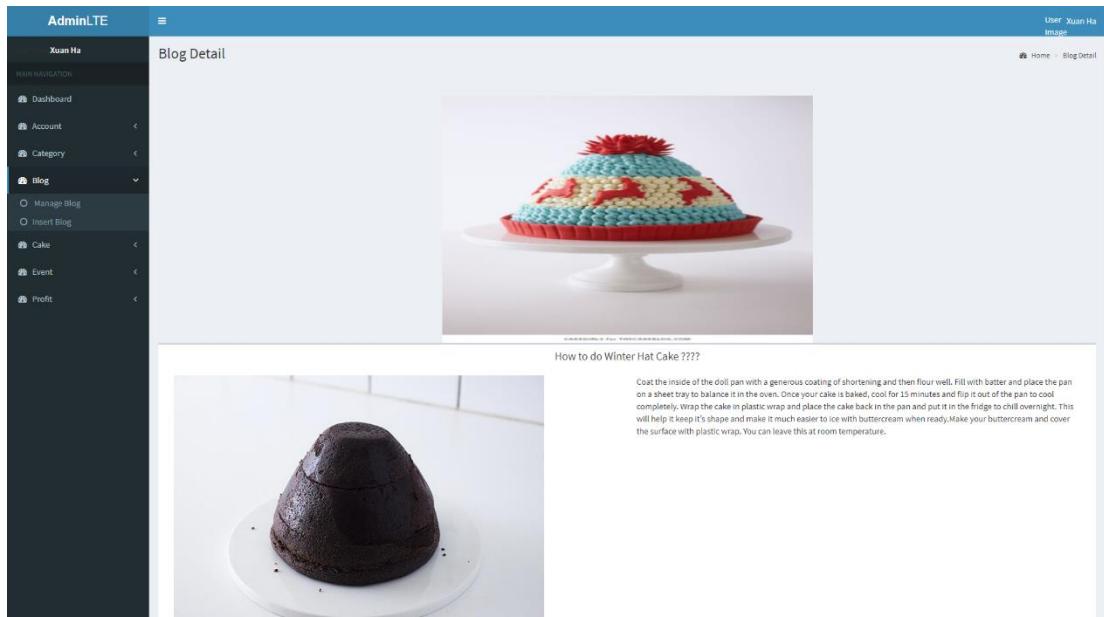


Figure 4-114: Blog Detail Screen

#### 4.5.2.5.2 Insert Blog

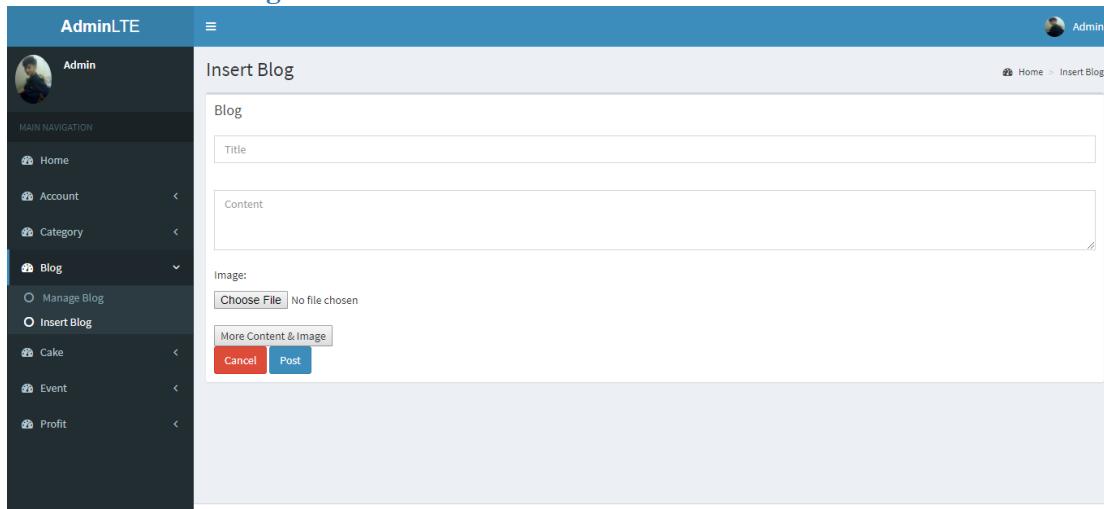


Figure 4-115: Insert Blog Screen

No	Field name	Type	Require	Mandatory	Max-Length	Description
1	Title	Text		Y		Title blog
2	Content	Text		Y		Content blog
3	Choose File	Button	Click	Y		Select Photo
4	More Content & Image	Button	Click			More Content and Image
5	Cancle	Button	Click	Y		Cancle and back to manage blog
6	Post	Button	Click	Y		Post Blog

Table 4-91: Insert Blog

#### 4.5.2.6 Cake

##### 4.5.2.6.1 Manage Cake

Figure 4-116: Manage Cake screen

No	Field name	Type	Require	Mandatory	Max-Length	Description
1	Search	Input Text		Y		Search by name
2	Icon Search	Button	Click	Y		Icon Search
3	Choose role you want	Drop Downlist	Click	Y		Search by role

<b>4</b>	List Cake	Table		Y		List Cake
<b>5</b>	Sell Again	Button	Click	Y		Updates Status
<b>6</b>	Not Sell	Button	Click	Y		Updates Status
<b>7</b>	Update	Button	Click	Y		Update cake

**Table 4-92:** Cake management

The screenshot shows the 'Manage Cake' update screen. On the left is a dark sidebar with a user profile picture and the word 'Admin'. Below it is a 'MAIN NAVIGATION' section with links: Dashboard, Account, Category, Blog, Cake (which is currently selected and highlighted in blue), Event, and Profit. The main content area has a title 'Manage Cake' and a sub-section 'Update Cake'. It contains several input fields with validation stars: 'Cake Name \*' (Bánh Nướng), 'Description \*' (Sản phẩm làm thủ công không có chất bảo quản), 'Size \*' (100g), 'Price \*' (100000), and 'Quantity \*' (678). There is also a 'Image \*' field with a preview of three cakes and a 'Choose File' button. At the bottom are two buttons: 'Cancel' (red) and 'Update' (blue).

**Figure 4-117:** Updates Cake Screen

No	Field name	Type	Require	Mandatory	Max-Length	Description
<b>1</b>	Cake Name	Text		Y		Cake Name
<b>2</b>	Decsription	Text		Y		Decsription Cake
<b>3</b>	Size	Text		Y		Size Cake
<b>4</b>	Price	Text		Y		Price of cake
<b>5</b>	Quantity	Text		Y		Quantity
<b>6</b>	Choose File	Button	Click	Y		Select photo
<b>7</b>	Cancel	Button	Click	Y		Cancel and back manage Cake
<b>8</b>	Update	Button	Click	Y		Update cake

**Table 4-93:** Update cake

#### 4.5.2.6.2 Insert Cake

The screenshot shows the 'Insert Cakes' page within an AdminLTE interface. The left sidebar has a dark theme with categories like Home, Account, Category, Blog, Cake (with Manage Cake and Insert Cake options), Event, and Profit. The main content area is titled 'Insert Cakes' and contains a form for adding a new cake. The form fields include:

- Cake Type: Radio buttons for 'Banh trung Thu', 'Banh Truyền Thống', 'Adc', and 'assdad'.
- Image: A file input field labeled 'Choose File' with the message 'No file chosen'.
- Cake Name: A text input field.
- Material: A text input field.
- More Material: A button labeled 'More Material'.
- Description: A text input field.
- Size: A text input field.
- Price: A text input field.
- Quantity: A text input field.
- Buttons: 'Cancel' and 'Add'.

**Figure 4-118:** Insert Cake Screen

No	Field name	Type	Require	Mandatory	Max-Length	Description
1	Cake Type	Check box	Click	Y		Cake Type
2	Cake Name	Text		Y		Cake Name
3	Material	Text		Y		Material
4	More Material	Button	Click	Y		More Material
5	Description	Text		Y		Description cake
6	Size	Text		Y		Size cake
7	Price	Text		Y		Price of cake
8	Quantity	Text		Y		Quantity
9	Choose File	Button	Click	Y		Select photo
10	Cancel	Button	Click	Y		Cancel and back manage Cake
11	Add	Button	Click	Y		Add cake

**Table 4-94:**  
Insert Cake

#### 4.5.2.7 Event

##### 4.5.2.7.1 Manage Event

Mỗi dịp lễ là khoảng thời gian để nhìn lại và trân trọng những con người đã làm nên một năm đặc biệt với chúng ta. Nhân dịp năm mới, chúng tôi xin gửi lời cảm ơn chân thành đến tất cả khách hàng đã luôn ủng hộ Fresh Garden. Chúc quý khách và gia đình một năm thành công, vui tươi và ngập tràn hạnh phúc.

Chào đón năm mới Bính Thân và để tri ân khách hàng, Fresh Garden ưu đãi 50% tất cả sản phẩm, duy nhất trong ngày 06/02/2016. Hãy nhanh chân ghé các cửa hàng Fresh Garden để tận hưởng khuyến mại hấp dẫn này nhé. Bạn sẽ có cơ hội thường xuyên được thử các sản phẩm bánh của Fresh Garden với mức giá tuyệt vời không thể bỏ lỡ. Chương trình áp dụng cho tất cả sản phẩm bánh tại các cửa hàng trong hệ thống của bánh tươi Fresh Garden. Chương trình không áp dụng tại Fresh Garden Vinmec và Fresh Garden sân bay Nội Bài. Chương trình không áp dụng cho các sản phẩm bánh đặt ngoài. Chương trình không áp dụng

**Figure 4-119: Manage Event screen**

No	Field name	Type	Require	Mandatory	Max-Length	Description
1	Search	Text		Y		Search by name Event
2	Search (icon)	Button	Click	Y		Icon search
3	List Event	List		Y		List Event
4	Lock	Button	Click	Y		Lock Event
5	UnLock	Button	Click	Y		Unlock Event

**Table 4-95: Manage Event**

##### 4.5.2.7.2 Insert Event

**Figure 4-120:** Insert Event screen

No	Field name	Type	Require	Mandatory	Max-Length	Description
<b>1</b>	Title	Text		Y		Title event
<b>2</b>	Content	Text		Y		Content event
<b>3</b>	Choose File	Button	Click	Y		Select Photo
<b>4</b>	Cancle	Button	Click	Y		Cancle and back to manage Event
<b>5</b>	Post	Button	Click	Y		Post Event

**Table 4-96:** Insert Event

#### 4.5.2.8 Profit

##### 4.5.2.8.1 Manage Bill

Email	Name Customer	Phone Number	Bill	Quantity	Price	Time Order	Delivery Location	Note	Status
kiennt@gmail.com	Kien	01234556781	Bánh Nướng Bánh Déo Nhân Đậu Xanh	3 2	740000	Tue, 14 Nov 2017 05:47:11 GMT	Ha Noi		Confirm UnConfirm
anhhtv01@gmail.com	Tuan	0987455644	Bánh Tết	3	105000	Thu, 16 Nov 2017 02:23:28 GMT	Phu Tho		Confirm UnConfirm

Figure 4-121: Manage Bill Screen

No	Field name	Type	Requi re	Ma nda tory	Max- Length	Description
1	Search	Date		Y		Search by date
2	Search (icon)	Button	Click	Y		Icon search
3	List Bill	Table		Y		List Bill
4	Confirm	Button		Y		Confirm bill
5	UnConfirm	Button		Y		UnConfirm bill

Table 4-97: Manage bill

#### 4.5.2.8.2 Revenue

Figure 4-122: Revenue Screen

No	Field name	Type	Require	Manda tory	Max- Length	Description
1	Search	Date		Y		Search by date
2	Search (icon)	Button	Click	Y		Icon search
3	From Date	Date		Y		From Date
4	To Date	Date		Y		To Date
5	Search (icon)	Button	Click	Y		Icon search
6	List Bill	Table		Y		List Bill

Table 4-98: Revenue

#### 4.5.2.8.3 Change Password

**Figure 4-123: Change Password Screen**

No	Field name	Type	Require	Mandatory	Max-Length	Description
<b>1</b>	Old Password	Text		Y		Old Password
<b>2</b>	New Password	Text		Y		New Password
<b>3</b>	Repeat Password	Text		Y		Repeat Password
<b>4</b>	Cancel	Button	Click	Y		Cancel
<b>5</b>	Change Password	Button	Click	Y		Change Password

**Table 4-99: Change Password**

#### 4.5.3 Staff

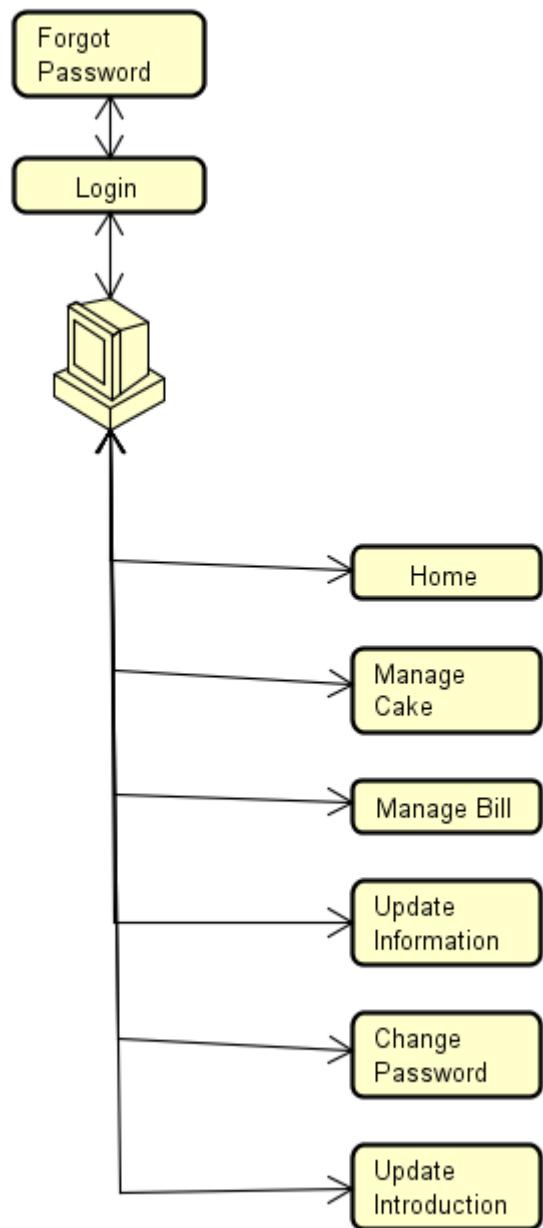
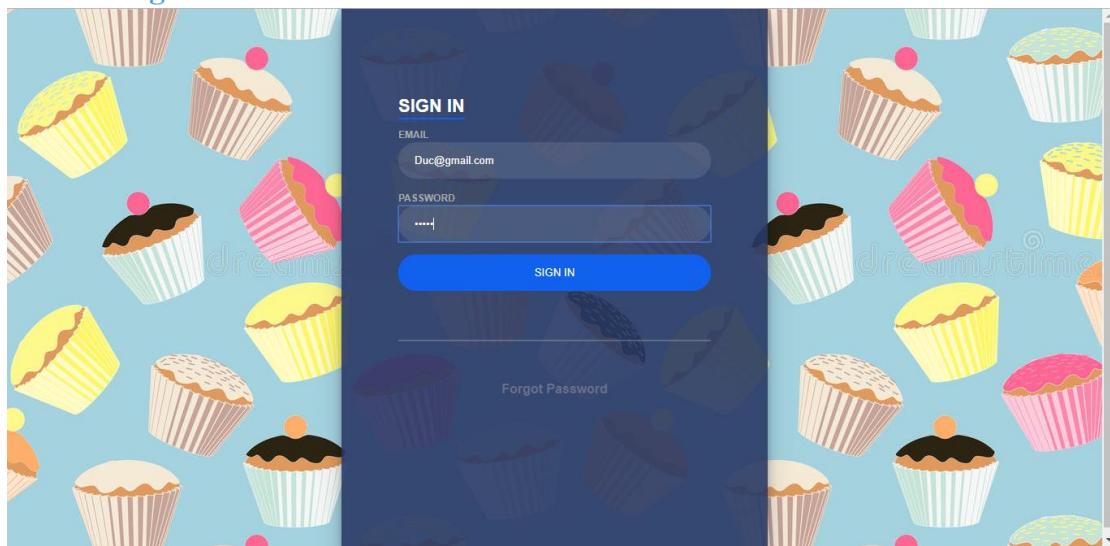


Figure 4-124: Staff screen flow

#### 4.5.3.1 Sign In

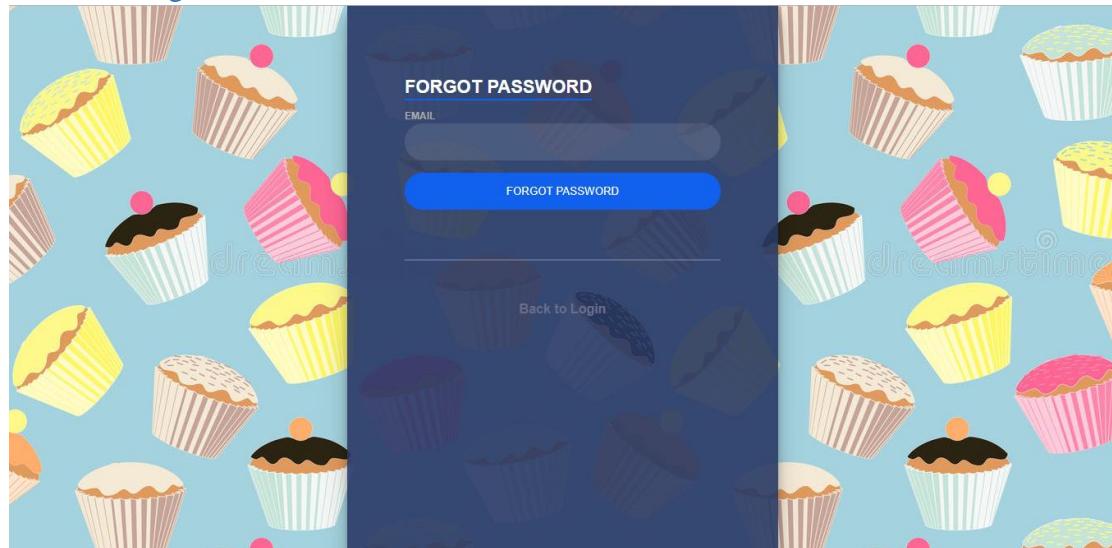


**Figure 4-125: LoginStaff Screen**

No	Field name	Type	Require	Mandator	Max-Lengt	Description
1	Email	Text box	Email	Y		Email of Staff
2	Password	Text box	Type	Y		Password of Staff
3	Sign in	Button	Click	Y		Login
4	Forgot Password	Link	Click	Y		Forgot Password

**Table 4-100:** LoginStaff

## 4.5.3.2 Forgot Password

**Figure 4-126:** Forgot Password screen

No	Field name	Type	Require	Mandatory	Max-Length	Description
1	Email	Text box	Email	Y		Email
2	Forgot Password	Button	Click	Y		Forgot password
3	Back to login	Link	Click	Y		Back to login

**Table 4-101:** Forgot Password

#### 4.5.3.3 Manage Cake

Cake Name	Description	Material	Size	Price	Quantity	Status
Bánh Nướng	Sản phẩm làm thủ công không có chất bảo quản	Đường hoa mai Chanh Mạch Nha Lòng đỏ trứng gà	50g	70001	12	Selling
Bánh Nướng	Sản phẩm làm thủ công không có chất bảo quản	Đường hoa mai Chanh Mạch Nha Lòng đỏ trứng gà	100g	100000	678	Selling
Bánh Nướng	Sản phẩm làm thủ công không có chất bảo quản	Đường hoa mai Chanh Mạch Nha Lòng đỏ trứng gà	150g	180000	10	Selling
Bánh Dẻo Nhân Đầu Xanh	Bánh làm thủ công	Đậu xanh không vỏ Mạch Nha	180g	100000	15	Not Sell

**Figure 4-127: Manage Cake screen**

No	Field name	Type	Require	Mandatory	Max-Length	Description
1	Search	Input Text		Y		Search by name
2	Icon Search	Button	Click	Y		Icon Search
3	Choose role you want	Drop Downlist	Click	Y		Search by role
4	List Cake	Table		Y		List Cake

**Table 4-102: Manage Cake screen**

#### 4.5.3.4 Manage Bill

Email	Name Customer	Phone Number	Bill	Price	Time Order	Delivery Location	Status
kiennt@gmail.com	Kien	01234556781	Bánh Nướng x 3 Bánh Đéo Nhân Đậu Xanh x 2	740000	Tue, 14 Nov 2017 05:47:11 GMT	Hà Nội	Confirm <span style="background-color: #007bff; color: white; padding: 2px 10px;">Confirm</span> UnConfirm
anhhtv01@gmail.com	Tuan	0987455644	Bánh Tết x 3	105000	Thu, 16 Nov 2017 02:23:28 GMT	Phú Thọ	Confirm <span style="background-color: #007bff; color: white; padding: 2px 10px;">Confirm</span> UnConfirm

**Figure 4-128: Manage Bill**

No	Field name	Type	Require	Mandatory	Max-Length	Description
1	Search	Date		Y		Search by date
2	Search (icon)	Button	Click	Y		Icon search
3	List Bill	Table		Y		List Bill
4	Confirm	Button		Y		Confirm bill
5	UnConfirm	Button		Y		UnConfirm bill

**Table 4-103: Manage Bill**

#### 4.5.3.5 Update Information

**Update Information**

Staff

Staff Name \*

Address \*

Phone Number \*

Email \*

Note

  No file chosen

Cancel Save

**Figure 4-129: Update Information screen**

No	Field name	Type	Require	Mandatory	Max-Length	Description
1	Staff Name	Text		Y		Staff Name
2	Addrees	Text		Y		Address Staff
3	Phone Number	Text		Y		Phone Number
4	Email	Text		Y		Email
5	Note	Text		Y		Other information
6	Choose File	Button	Click	Y		Select photo
7	Cancel	Button	Click	Y		Cancel
8	Save	Button	Click	Y		Save

**Table 4-104: Update Information**

#### 4.5.3.6 Change Password

The screenshot shows the 'Change Password' screen. On the left is a dark sidebar with a user icon and the name 'Duc12'. Below it is a 'MAIN NAVIGATION' section with links to Home, Manage Cake, Manage Bill, Update Information, Change Password, and Update Introduction. The main content area has a light blue header 'Change Password'. It contains three input fields: 'Old Password \*' with a placeholder 'Old Password', 'New Password \*' with a placeholder 'New Password', and 'Repeat Password \*' with a placeholder 'Repeat Password'. At the bottom are two buttons: a red 'Cancel' button and a blue 'Change Password' button.

**Figure 4-130: Change Password screen**

No	Field name	Type	Require	Mandatory	Max-Length	Description
1	Old Password	Text		Y		Old Password

<b>2</b>	New Password	Text		Y		New Password
<b>3</b>	Repeat Password	Text		Y		Repeat Password
<b>4</b>	Cancel	Button	Click	Y		Cancel
<b>5</b>	Change Password	Button	Click	Y		Change Password

**Table 4-105: Change Password****4.5.3.7 Update Introduction**

The screenshot shows the 'UpdateIntroduction' page. On the left is a dark sidebar with a user icon and the name 'Duc12'. Below it are navigation links: Home, Manage Cake, Manage Bill, Update Information, Change Password, and Update Introduction. The main area has a blue header 'UpdateIntroduction'. It contains four input fields: 'Address \*' with 'Ha Noi 2', 'Phone Number \*' with '0972550995', 'Email \*' with 'kient12012@gmail.com', and a 'Choose File' button with a placeholder 'No file chosen'. At the bottom are 'Cancel' and 'Lưu' buttons.

**Figure 4-131: Update Introduction screen**

No	Field name	Type	Require	Mandatory	Max-Length	Description
1	Address	Text		Y		Address your store
2	Phone Number	Text		Y		Phone Number
3	Email	Text		Y		Email
4	Choose File	Button	Click	Y		Map Photo
5	Cancel	Button	Click	Y		Cancel
6	Update	Button	Click	Y		Update

**Table 4-106: Update Introduction**

## 5 SOFTWARE TESTING

### 5.1 Introduction

#### 5.1.1 Purpose

This is the comprehensive test plan of the BSO project. The purpose of this chapter describes scopes of test and activities which need to be taken during test process of project. It addresses the following items: Scopes of Testing, Requirements for Testing, Test Strategy, Test Resources, Test Environment, Test Milestones and Deliverables.

#### 5.1.2 Definitions and Acronyms

This section describes the definitions, terms, and acronyms that are used in software requirements specification.

Acronym	Definition	Note
BSO	Bakery Store Online	
GUI	Graphic User Interface	
IT	Integration Test	
KLOC	1000 line of code	
PM	Project Manager	
QA	Quality Assurance	
SRS	Software Requirement Specification	
ST	System Test	
TC	Test Case	
TL	Test Leader	
TP	Test Plan	
TR	Test Report	
UT	Unit Test	

**Table 5-1: Definitions and Acronyms**

#### 5.1.3 Scope of testing

BSO will be tested by 3phases:

### **Phase 1: Integration testing**

- After finishing component testing, integration testing will be performed by testers.
- Material are integration test cases, high- level design and test tools.
- Integration test focuses on specific areas of use cases when all requirements are completed.
- Integration test should be performed to ensure all components incorporate well.
- When executing integration testing, if any bugs are found, testers have to log on “Defect Log Management” file and assign to developer fix it and redo this process until it is correct.

*Rule for filling test result:*

Test result pass	Pass
Test result fail	Fail
Do not test	Untested
Cannot test	N/A (Not available)

**Table 5-2: Rule integration test**

### **Phase 2: System testing**

- After finishing integration testing and developers collect all functions and items, testers will be performed system testing, it means doing test whole system.
- Material area system test case, SRS
- If any bugs are found, developers have to fix and testers will verify them. System test is ended only when test cases are passed and no bug is found.

*Rule for filling test result:*

Test result pass	Pass
Test result fail	Fail
Do not test	Untested
Cannot test	N/A (Not available)

**Table 5-3: Rule system test**

### **Phase 3: Acceptance testing**

- Base on customer/user requirement specification, system is tested again, for ensure there is not lacking or mistake any requirement.
- If there is any problem, developers have to fix/update and tester will verify them.

- Acceptance testing is ended only when whole system met requirement specification.

#### 5.1.4 Constraints

- Deadline for testing only can be met if development progress is on time.
- Test execution can be performed when system passes Unit Test Inspection.
- At least one round of testing must be performed for requirements.
- Have more environments should be tested: Window 10, Window 8... and more browsers: Firefox 30 and Google Chrome 40...

### 5.2 Test plan

#### 5.2.1 Test type

##### 5.2.1.1 Function Testing

- Functional testing is a type of software testing whereby the system is tested against the functional requirements/specifications.
- Functions are tested by feeding them input and examining the output. Functional testing ensure that the requirements are properly satisfied by the website. This type of testing is not concerned with how processing occurs, but rather, with the results of processing.
- During functional testing, Black Box Testing technique is used in which the internal logic of the system being tested is not known to the tester.

<b>Test Objective:</b>	The type of this test is to ensure proper target-of-test functionality, including user interaction, all function defined in specification document implemented correctly.
<b>Technique:</b>	Executing each use case, use-case flow, or function, using valid and invalid data, to verify the following: <ul style="list-style-type: none"> <li>- The expected results occur when valid data is used.</li> <li>- The appropriate error or warning messages are displayed when invalid data is used.</li> <li>- Each business rule is properly applied.</li> </ul>
<b>Completion Criteria:</b>	<ul style="list-style-type: none"> <li>- All planned tests have been executed.</li> <li>- All identified defects have been addressed and closed.</li> </ul>
<b>Special Considerations:</b>	Testing may be stopped when Time runs out A certain number of defects found

	Test coverage > 97% Stop when testing becomes unproductive
--	---

**Table 5-4: Function Testing**

### 5.2.1.2 User Interface Testing

GUI testing is the process of ensuring proper functionality of the GUI for a given web and making sure it conforms to its written specifications.

GUI testing evaluates design elements such as layout, colors, fonts, font sizes, labels, text boxes, text formatting, captions, buttons, lists, icons, links, content and more.

<b>Test Objective:</b>	Verify the following: <ul style="list-style-type: none"><li>- Navigation through the target-of-test properly reflects business functions and requirements, including window-to-window, field-to-field, and use of access methods (tab keys, mouse movements, accelerator keys)</li><li>- Window objects and characteristics, such as menus, size, position, state, and focus conform to standards.</li></ul>
<b>Technique:</b>	Create or modify tests for each window to verify proper navigation and object states for each application window and objects.
<b>Completion Criteria:</b>	Each window successfully verified to remain consistent with benchmark version or within acceptable standard
<b>Special Considerations:</b>	Not all properties for custom and third party objects can be accessed.

**Table 5-5: GUI Testing**

#### 5.2.1.2.1 Data and Database Integrity Testing

The databases and the database processes should be tested as a subsystem within the Project. These subsystems should be tested without the target-of-test's User Interface as the interface to the data. Additional research into the Database Management System (DBMS) needs to be performed to identify the tools and techniques that may exist to support the testing identified below.

<b>Test Objective:</b>	Ensure database access methods and processes function properly and without data corruption.
<b>Technique:</b>	<ul style="list-style-type: none"><li>- Invoke each database access method and process, seeding each with valid and invalid data or requests for data.</li><li>- Inspect the database to ensure the data has been populated as intended, all database events occurred properly, or review the returned data to ensure that the correct data was retrieved for the correct reasons.</li></ul>

<b>Completion Criteria:</b>	All database access methods and processes function as designed and without any data corruption.
<b>Special Considerations:</b>	<ul style="list-style-type: none"> <li>- Testing may require a DBMS development environment or drivers to enter or modify data directly in the databases.</li> <li>- Processes should be invoked manually.</li> <li>- Small or minimally sized databases (limited number of records) should be used to increase the visibility of any non-acceptable events.</li> </ul>

**Table 5-6: Data and Data Integrity Testing**

### 5.2.2 Test stages

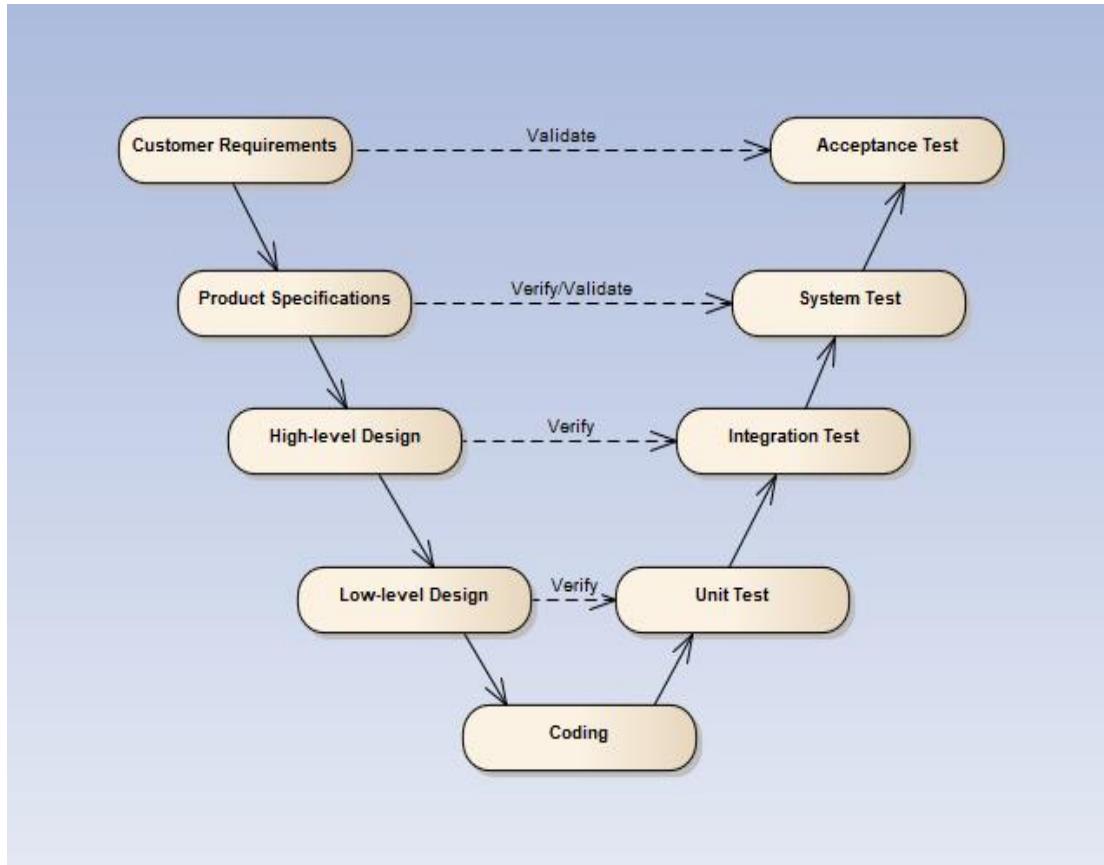
Clearly state the stage in which the test will be executed. Identified below are the stages in which common test are executed

<b>Type of Tests</b>	<b>Stage of Test</b>	
	<b>Integration</b>	<b>System</b>
Function Testing	X	X
User Interface Testing		X

**Table 5-7: Test stages**

### 5.2.3 Test model

BSO follows V-Model process:

**Table 5-8: V-model process**

Testing progress is divided to 5 phases include: Unit test, Component test, Integration test, System test and Acceptance test

- Unit test:
  - Unit testing is used to verify a single minimal unit of source code. The purpose of unit testing is to isolate the smallest testable parts of BSO and verify that they function properly in isolation.
  - Unit testing is the first level of testing and is performed prior to component testing
  - Unit testing will be done by developer.
- Integration test:
  - Integration testing is a level of the software testing process where individual units or component are combined and tested as a group.
  - The purpose is to expose faults in the interaction between integrated units.
  - Integration testing is performed after component testing
  - Integration testing will be done by tester

- There are two methods of doing integration testing: Bottom-up Integration testing and Top Down Integration testing:

No	Integration Testing Method
1	<p>Bottom-up integration</p> <p>This testing begins with unit testing, followed by tests of progressively higher-level combinations of units called modules.</p>
2	<p>Top- Down integration</p> <p>This testing, the highest-level modules are tested first and progressively lower-level modules are tested after that</p>

**Figure 5-1: Integration test**

- System test:
  - System Testing is a level of the software testing process where a complete, integrated system is tested
  - The purpose is to evaluate the system's compliance with the specified requirements
  - System testing is performed after integration testing
  - System testing will be done by tester
- Acceptance test:
  - Acceptance testing is performed after system testing
  - Acceptance testing will be performed by the test leader and team leader.
  - The acceptance test will be done for a period of 1 weeks after completion of the system test process

#### 5.2.4 Acceptance test criteria.

- Criteria for Integration test:
  - Number of TC/KLOC: 30 TC/KLOC
  - Number defects/KLOC: 2-3 defects/KLOC
- Criteria for System test:
  - Number of TC/KLOC: 20 TC/KLOC
  - Number defects/KLOC: 4-6 defects/KLOC
- Criteria for Acceptance test:
  - Number defects/KLOC: 1-2 defects/KLOC

#### 5.2.5 Futures to be tested

Refer [3.2.1.2 Product features](#) for more detail.

### 5.2.6 Feature not to be tested

- The stable of website when do not connect internet.
- Over than 5000 users connect to system at the same time.

### 5.2.7 Resources

#### 5.2.7.1 Human resource

Worker/Doer	Role	Specific Responsibilities/Comments
PhuongHA	Test Leader	Manage Test Resource and assign test tasks. Create and review Test Plan. Create and review Test Case. Execute test. Create and review Test Report
PhuongPTM	Tester	Create and review Test Case. Execute test. Create Test view points Create and review Test Report

**Table 5-9: Human resource**

#### 5.2.7.2 Test Environment

##### 5.2.7.2.1 Hardware

Name	Purpose	Detail
Laptop HP	Device for create and execute test	Window 8.1 Pro Core i3
Laptop Asus	Device for create and execute test	Window 8.1 Pro Core i5

**Table 5-10: Hardware**

##### 5.2.7.2.2 Software

Name	Purpose	Detail
Test Plan	Managing test	Microsoft Word 2010, 2013
Test case	Executing test	Microsoft Excel 2010, 2013
Test report, Test checklist	Tracking test	Microsoft Excel 2010, 2013
Chrome	Executing test	Chrome 40.0
Firefox	Executing test	Firefox

**Table 5-11: Software**

### 5.2.7.2.3 Infrastructure

Name	Purpose	Detail	Vendor/In-house	Version
BSO_Defect Log Management_v1.0_EN	Tracking bug during testing time	Microsoft Excel 2013, 2010	FPT-University	v1.0
Test Effort	Effort execute test	Microsoft Excel 2013, 2010	FPT-University	v1.0

**Table 5-12:Infrastructure**

### 5.2.8 Test milestones

Milestone Task	Effort (pd)	Start Date	End Date
Create Test Plan	5	21/10/2017	23/10/2017
Create Integration Test case	2	06/11/2017	10/11/2017
Review & Update Integration Test Case	2	11/11/2017	12/11/2017
Create System Test case	5	30/10/2017	05/11/2017
Review & Update System Test Case	2	04/11/2017	05/11/2017
Execute Integration test phase 1	2	13/11/2017	17/11/2017
Execute Integration test phase 2	2	24/11/2017	26/11/2017
Execute Integration test phase 3	2	03/12/2017	05/12/2017
Execute System test phase 1	4	18/11/2017	20/11/2017
Execute System test phase 2	4	27/11/2017	01/12/2017
Execute System test phase 3	4	06/12/2017	09/12/2017
Create Test Plan	5	21/10/2017	23/10/2017
Review & Update Test Plan	2	16/12/2017	17/12/2017

**Table 5-13: Test milestones**

### 5.2.9 Deliverables

No	Deliverables	Responsibilities	Delivered Date
1	Test Plan	Tester	
2	Integration Test case	Tester	
3	System Test case	Tester	
4	Defect Log Management	All members	

5	Test report	Test Leader	
---	-------------	-------------	--

**Table 5-14:Deliverables**

### 5.3 Test case

Refer to:

- BSO\_Integration Test Case\_v1.0\_EN.xlsx
- BSO\_System Test Case\_v1.0\_EN.xlsx
- BSO\_Common Test Case\_v1.0\_EN.xlsx

### 5.4 Test Report

#### 5.4.1 System test report

No	Module Code	Pass	Fail	N/A	Number of Test Case
1	Display Personal Page	16	0	0	16
2	Admin Login	64	0	0	64
3	Account Management	58	0	0	58
4	Category Management	16	0	0	16
5	Blog Management	16	0	0	16
6	Cake Management	38	0	0	38
7	Event Management	66	0	0	66
8	Profit Management	30	0	0	30
9	Staff Login	66	0	0	66
10	Message	52	0	0	52
11	Admin Module	160	0	0	160
12	User Login	104	0	0	104
	<b>Total</b>	<b>634</b>	<b>0</b>	<b>0</b>	<b>634</b>

**Table 5-15: System test report**

#### 5.4.2 Integration test report

No	Module Code	Pass	Fail	N/A	Number of Test Case
1	Registered_User_function	128	0	0	128
2	Admin_Function	244	0	0	244
3	Staff_Function	126	0	0	126
	<b>Total</b>	<b>498</b>	<b>0</b>	<b>0</b>	<b>498</b>

**Table 5-16: Integration test report**

#### 5.4.3 Test report

No	Module Code	Round 1	Round 2	Round 3	Final
----	-------------	---------	---------	---------	-------

		Pass	Fail	Pass	Fail	Pass	Fail	
1	Integration Test	498	0	498	0	498	0	498
2	System Test	583	51	602	32	634	0	634
	<b>Total</b>	1081	51	1100	32	1131	0	1131

**Table 5-17: Test Report****5.4.4 Defect report**

Defects	Logged	%Closed	%Invalid	%Open
<b>Critical</b>	0	0%	0%	0%
<b>High</b>	4	0%	0%	0%
<b>Medium</b>	24	100%	0%	0%
<b>Low</b>	38	100%	0%	0%
<b>Critical</b>	0	0%	0%	0%

## 6 USER MANUAL

### 6.1 Introduction

#### 6.1.1 Purpose

This document contains guide-lines step by step to deploy BSO App to Azure website and This user's manual has been developed to help users better understand the requirements that may apply to use this website by providing them simple step by step tutorials. This manual is intended to help make website's functions more accessible and their associated requirements more understandable to users.

#### 6.1.2 Environment

Following are the software required to start the BSO system:

- Operating System: Windows 7, Windows 8, Window 10
- Browsers: Firefox 40, Chrome 44 or higher.
- Database: MongoDB
- NodeJS Framework
  - Express
  - Multer
  - BodyParser
  - Passport
  - LocalStrategy
  - Session
  - CookieParser
  - Flash
  - Morgan
  - Nodemailer

### 6.2 Installation Guideline

#### 6.2.1 Connecting Mlab

##### 6.2.1.1 Creating the Mongo Database

The first thing to do is creat your database on Mlab:

- Step 1: Go to your mlab, and click the Create new button in the right corner.

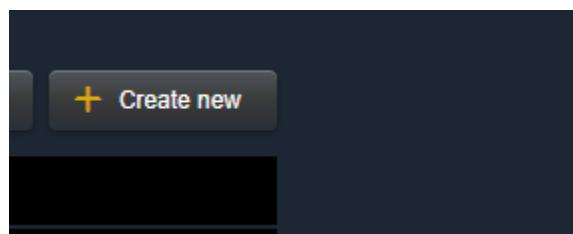
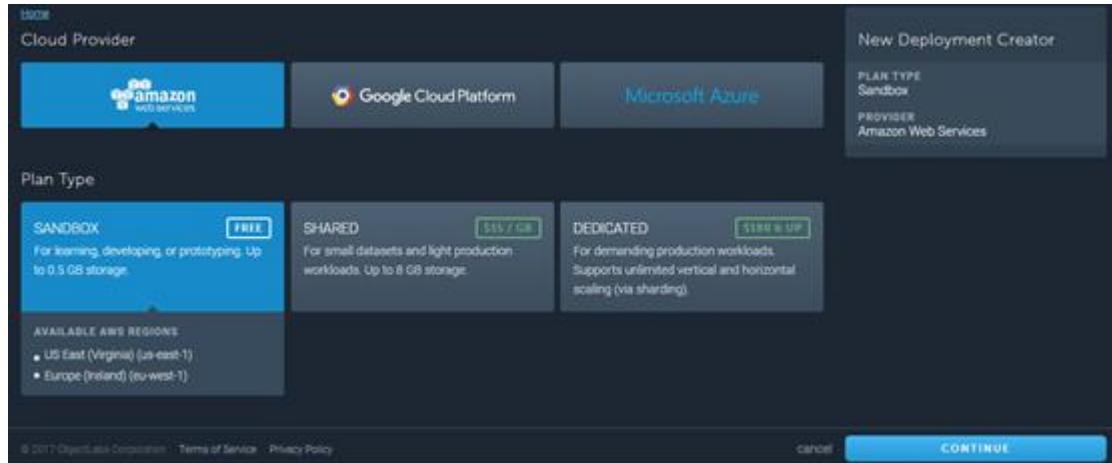


Figure 6-1: New button

- Step 2: Choose one in three Clooud Provider and PlanType then click continue button.



- Step 3: Select Region then click continue button.



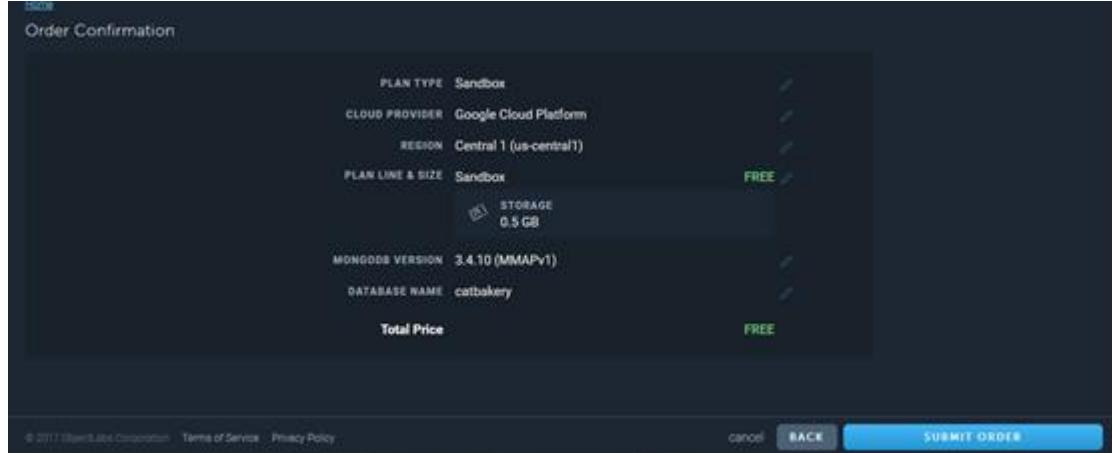
- Step 4: Write your database name
  - Database Name: **catbakery**

Then click continue button.



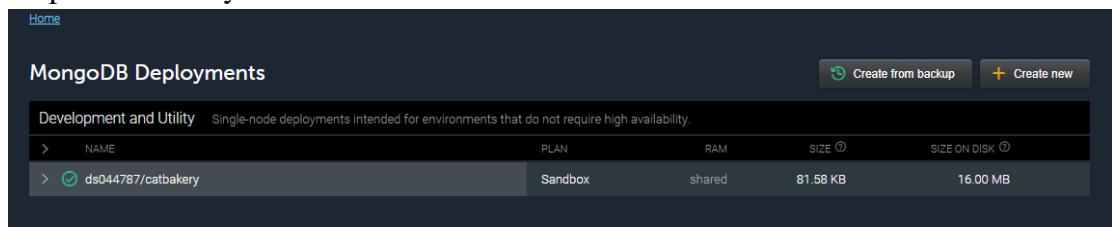
**Figure 6-2: List SQL Database**

- Step 5: Confirm all the things you have done then click Submit order button

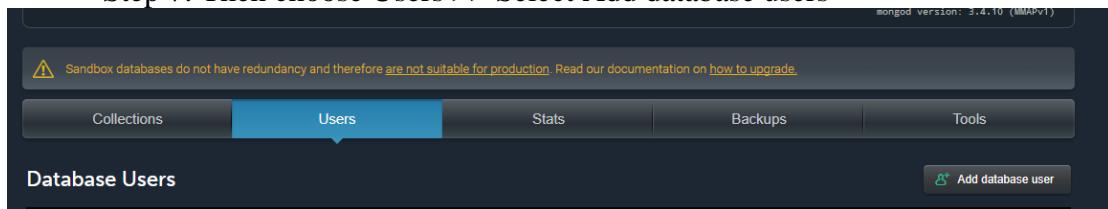
**Figure 6-3:**

After waiting a few seconds your database should be created.

- Step 6: Click on your Database

**Figure 6-6:**

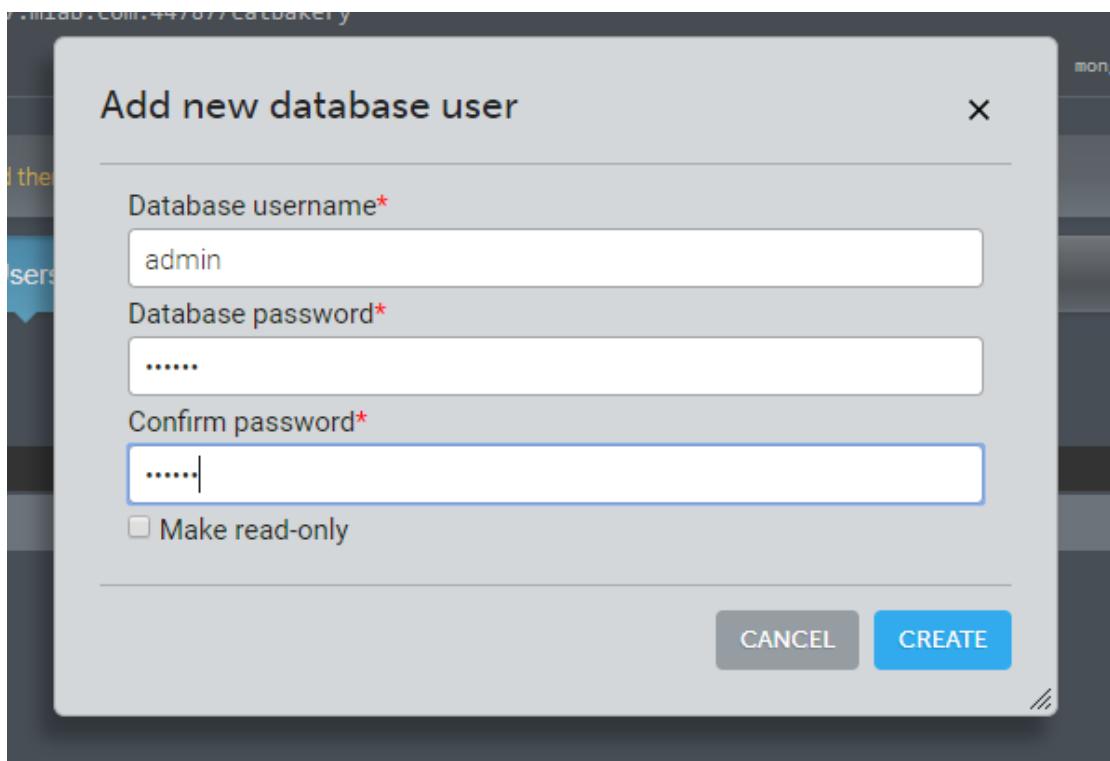
- Step 7: Then choose Users >> Select Add database users

**Figure 6-7:**

- Step 8 : Fill in the form. My settings were:

- Database username: **admin**
- Database password: **123456**
- Confirm password: **123456**

Then click Create button

**Figure 6-8:**

After waiting a few seconds your database user should be created.

NAME	READ ONLY?
admin	false

- Step 9 : Go back to Homepage

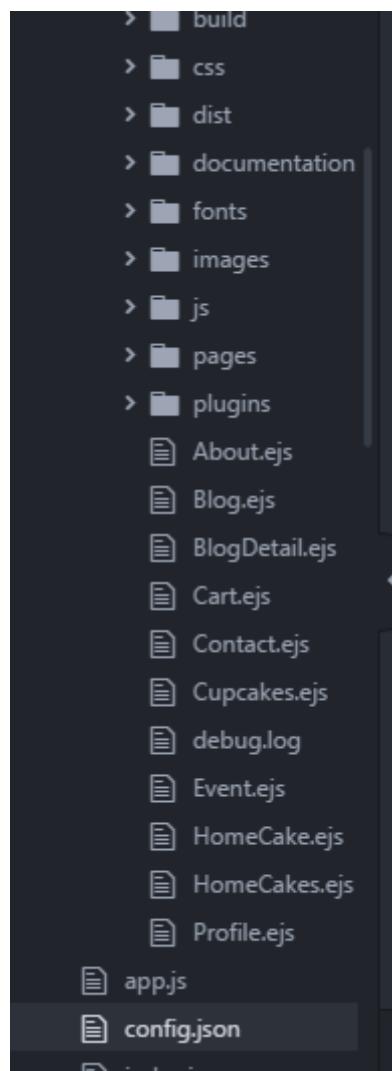
**Figure 6-10:**

Your database is now set-up and ready for you to configure in your web app.

#### 6.2.1.2 Updating Connection Details in Your Web App

The next step is to update your MVC application to point at the newly created mongo database.

Step 1: Within your atom, find the file called config.json and open it



**Figure 6-11:**

- Step 2: Add the following code into the file directly.

```
1 {  
2   "connectionString": "mongodb://<dbuser>:<dbpassword>@ds044787.mlab.com:44787/catbakery"  
3 }  
4
```

The screenshot shows a code editor window with four lines of JSON code. Line 1 starts with a '{' character. Line 2 contains the key 'connectionString' followed by a value enclosed in quotes. The value is a MongoDB connection string. Line 3 ends with a '}' character. Line 4 is an empty line.

**Figure 6-12:**

- Step 3: Enter you dbuser and db password that you create on mlab (Figure 6-8).

```

1  {
2    "connectionString": "mongodb://admin:123456@ds044787.mlab.com:44787/catbakery"
3  }
4

```

- Step 5: Save the file

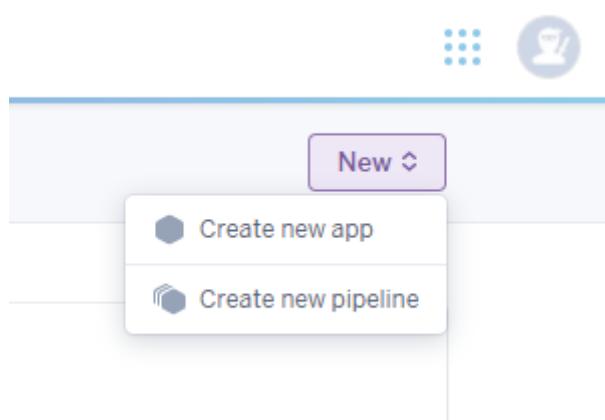
## 6.2.2 Create & Deploying the Heroku Website

### 6.2.2.1 Create Heroku Website

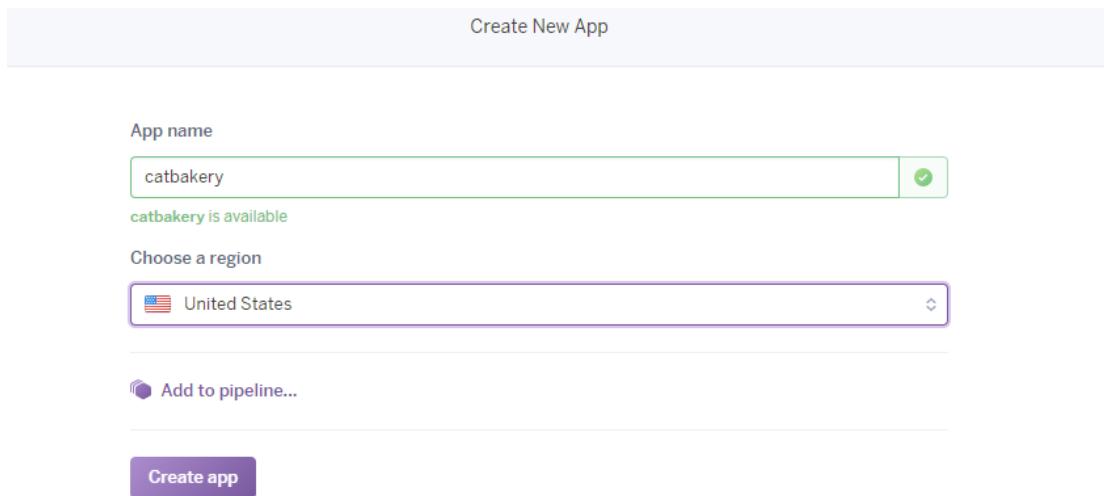
In order to deploy your website to Heroku you're going to need to set-up an account. At [Heroku Website](#) you can create an account for free and have a free trial. Furthermore, at this point in time you can set-up up to 5 heroku websites for free – you cannot use custom domain names, ssl or several other features, but they are perfect for experimentation.

Once you have created your account head to <https://dashboard.heroku.com/apps> where you can set-up your first website:

- Step 1: From the main screen click the new button in the right corner  
Then select Create new app



- Step 2: Enter the name for your website, I used “[catbakery](#)” (note that you have to choose a name that's available).
- Step 3: Select a region where you want your site to be hosted, I suggest choose United States
- Step 4: Press create website

**Figure 6-4: Create Web Site**

You'll have to wait a few seconds for heroku to spin up the website

**Figure 6-5: List Website**

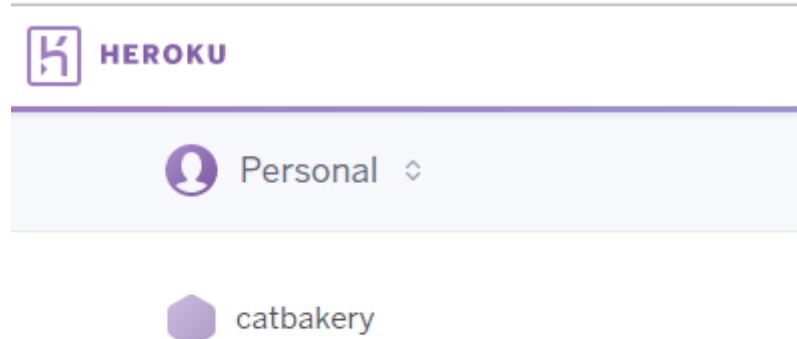
Now you've got an heroku website ready, we can get to publishing your mvc site to the heroku service.

- Step 6 : Click Open app in the right corner to see your website looklike.

**Figure 6-16:**

### 6.2.2.2 Deploying to the Heroku Website

- Step 1: From heroku dashboard, right click to your app in the left corner.



**Figure 6-17:**

- Step 2: Then select Deploy

This screenshot shows the Heroku dashboard with the 'Deploy' tab selected. The top navigation bar includes links for Overview, Resources, Deploy, Metrics, Activity, Access, and Settings. The 'Deploy' link is highlighted in blue. The main content area contains sections for adding the app to a pipeline and choosing a deployment method. It also lists various deployment options like Heroku Git, GitHub, Dropbox, and Container Registry.

Deployment method	Description
Heroku Git	Use Heroku CLI
Github	Connect to GitHub
Dropbox	Connect to Dropbox
Container Registry	Use Heroku CLI

You will see the Deployment method. Select one. I choose Heroku Git.

- Step 3: Install the Heroku CLI

Install the Heroku CLI

Download and install the [Heroku CLI](#).

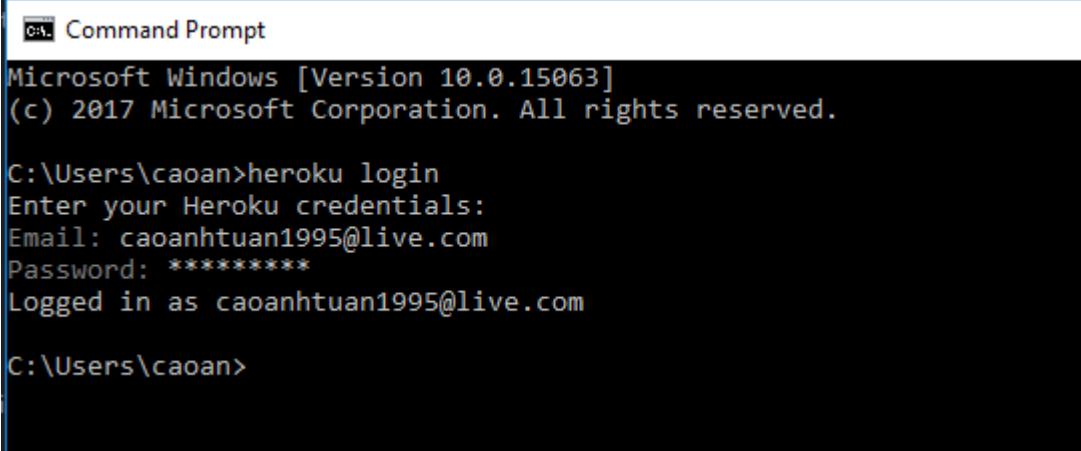
If you haven't already, log in to your Heroku account and follow the prompts to create a new SSH public key.

```
$ heroku login
```

**Figure 6-18:**

- Step 4: Open the Command Prompt type: heroku login
  - You need to enter your email and your password  
(email and password of your heroku account)

---



```
cmd Command Prompt
Microsoft Windows [Version 10.0.15063]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\Users\caoan>heroku login
Enter your Heroku credentials:
Email: caoanthuan1995@live.com
Password: *****
Logged in as caoanthuan1995@live.com

C:\Users\caoan>
```

**Figure 6-18:**

- Step 5: Create a new git repository

Follow this step.

Create a new Git repository

Initialize a git repository in a new or existing directory

```
$ cd my-project/
$ git init
$ heroku git:remote -a catbakery
```

The screenshot shows a Microsoft Windows Command Prompt window titled 'Command Prompt'. The window displays the following terminal session:

```
Microsoft Windows [Version 10.0.15063]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\Users\caoan>heroku login
Enter your Heroku credentials:
Email: caoanthuan1995@live.com
Password: *****
Logged in as caoanthuan1995@live.com

C:\Users\caoan>cd C:\Users\caoan\Desktop\JPD322\do_an\Project Back End\Project Back End\DoAn

C:\Users\caoan\Desktop\JPD322\do_an\Project Back End\Project Back End\DoAn>git init
Reinitialized existing Git repository in C:/Users/caoan/Desktop/JPD322/do_an/Project Back End/Project Back End/DoAn/.git/

C:\Users\caoan\Desktop\JPD322\do_an\Project Back End\Project Back End\DoAn>heroku git:remote -a catbakery
set git remote heroku to https://git.heroku.com/catbakery.git

C:\Users\caoan\Desktop\JPD322\do_an\Project Back End\Project Back End\DoAn>
```

**Figure 6-19:**

- Step 6: Deploy you application

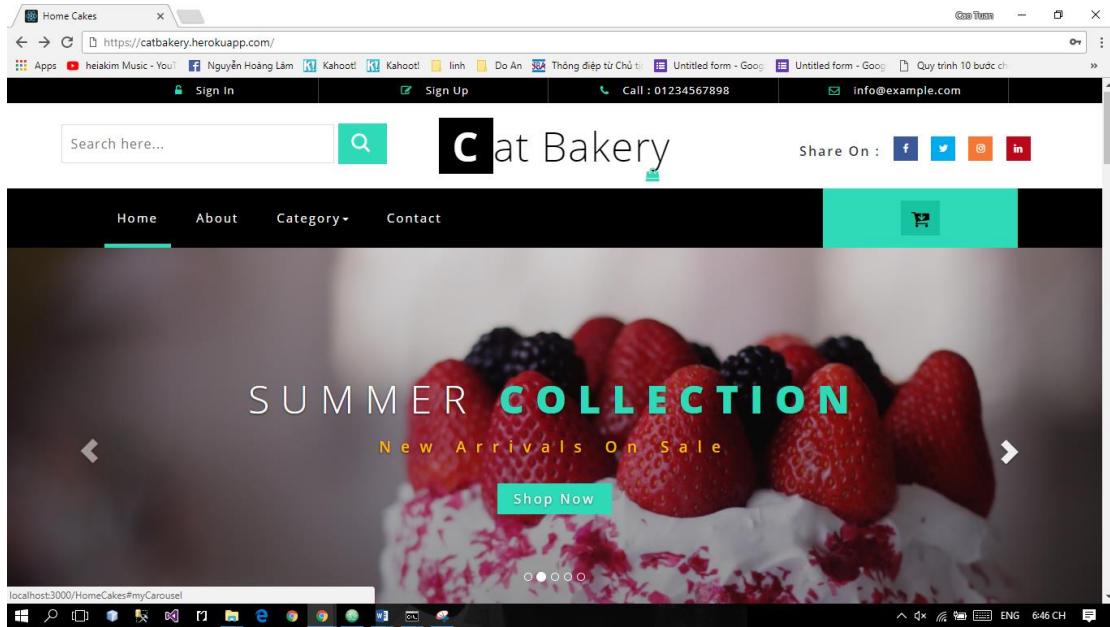
Follow this step.

Deploy your application

Commit your code to the repository and deploy it to Heroku using Git.

```
$ git add .
$ git commit -am "make it better"
$ git push heroku master
```

Once your website has finished deploying, open your browser type  
[yourwebsitename.herokuapp.com](http://yourwebsitename.herokuapp.com)



**Figure 6-6: Save button**

## 6.3 User Guideline

### 6.3.1 User Client

#### 6.3.1.1 Sign up

This is the first page which user will see. From this page the user can go to every page they want.

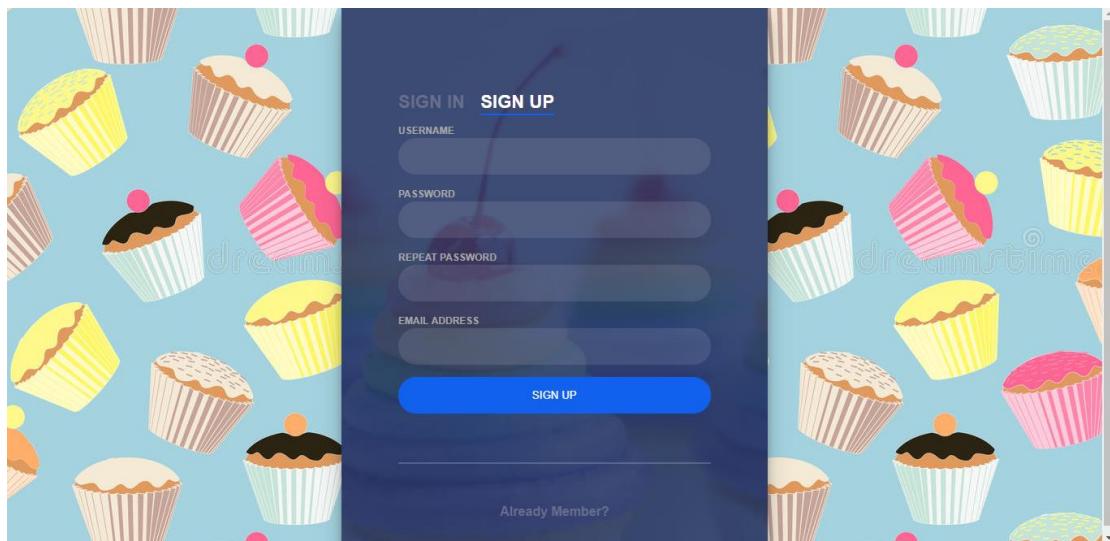
To register account on website, the users do the steps below:

Step 1: Open browser and enter to the address bar: <http://catbakery.herokuapp.com/>. Homepage is displayed.

Step 2: Click on Sign in button in header. Users will go to Sign in page.

Step 3: Click Sign up on Sign in page. Users will move to Sign up page.

Step 4: Enter information and click on Sign up to register.



**Figure 2-1: Sign up screen**

### 6.3.1.2 Login

To login account on website, the users do the steps below:

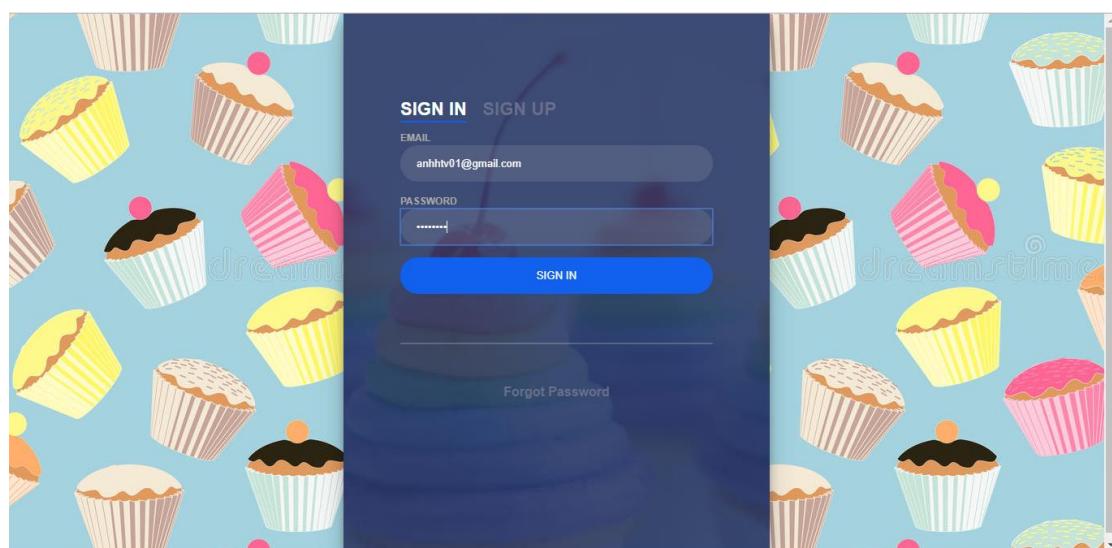
Step 1: Open browser and enter to the address bar: <http://catbakery.herokuapp.com/>.  
Homepage is displayed.

Step 2: Click on Sign in button in header. User will go to Login page.

Step 3: User can Login with BSO'account:

Step 3.1: User enter username and password

Step 3.2: Click on Login button

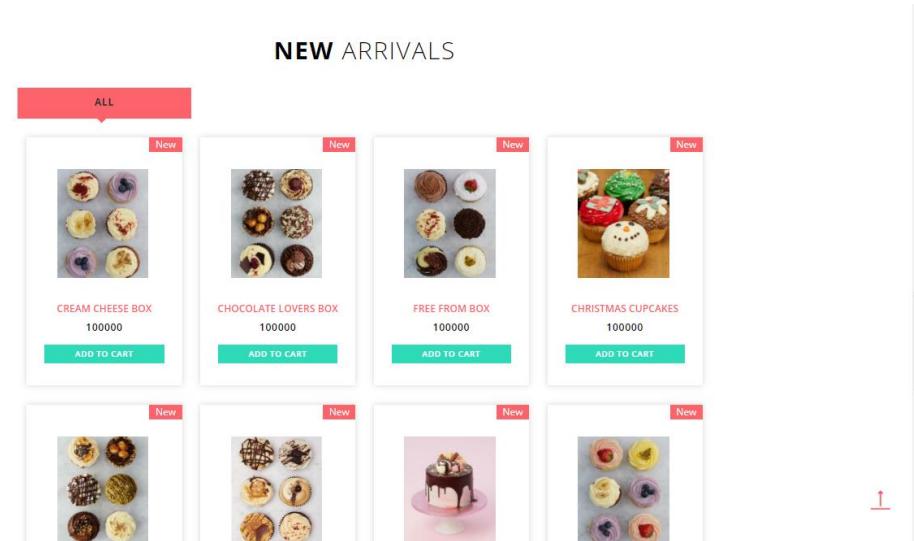


**Figure 2-1: Login page**

### 6.3.1.3 Buy Cake

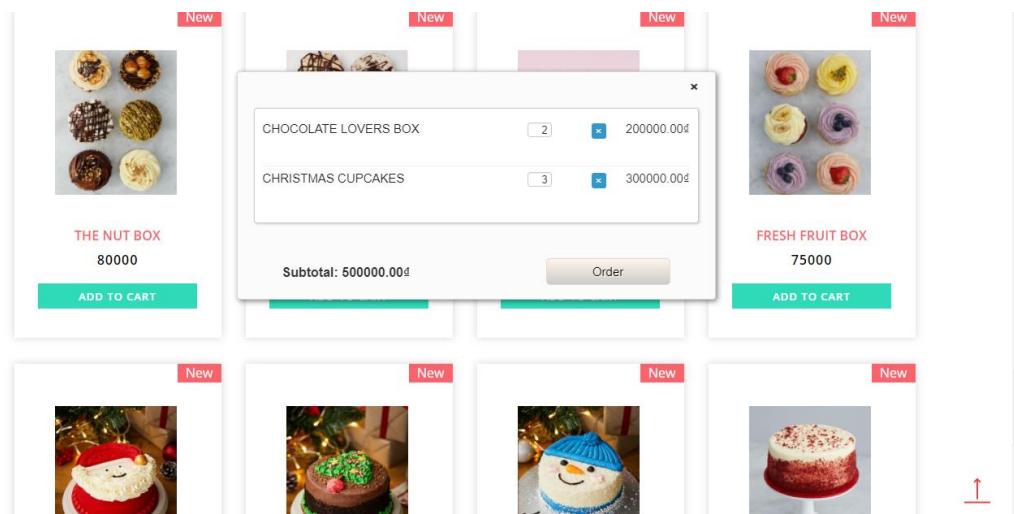
To buy a cake on website, the users do the steps below:

Step 1: In Home page user will see list of cake.



**Figure 2-1: List cake**

Step 2: Click on add to cart to add cake to your cart.



**Figure 2-2: Order cake**

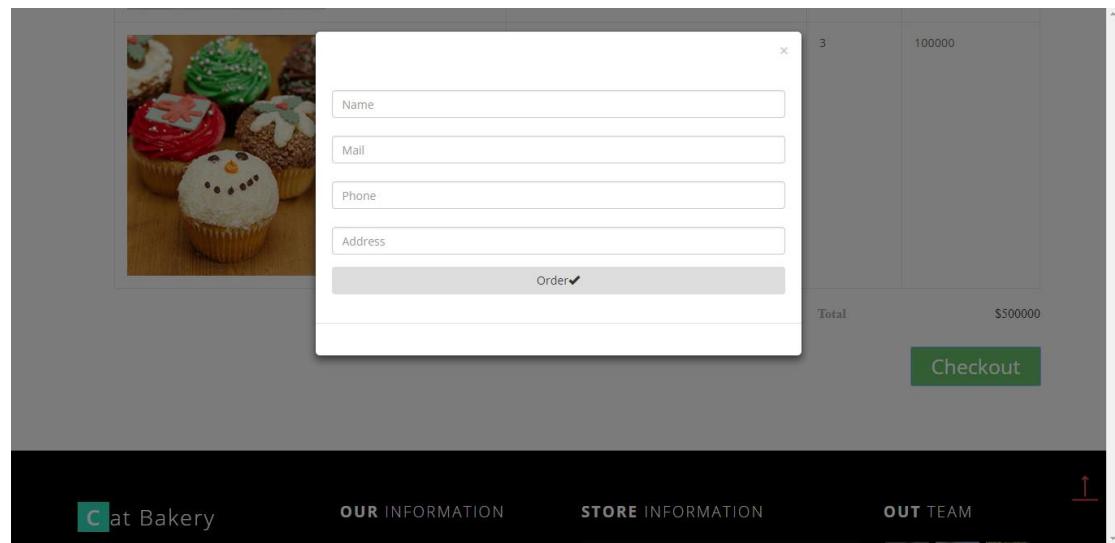
Step3: Click order to view your order

SHOPPING CART			
Image	Product	Price	Quantity
	CHOCOLATE LOVERS BOX Good for health	2	100000
	CHRISTMAS CUPCAKES Good for health	3	100000
		Total	\$500000
		<a href="#">Checkout</a>	

**Figure 2-3: Check out**

Step4: Click check out to finish your order

If you don't have an account you need to fill this form to finish your order



The screenshot shows a shopping cart summary at the top with two items: a 'CHOCOLATE LOVERS BOX' and 'CHRISTMAS CUPCAKES'. The total price is \$500000. Below the cart is a green 'Checkout' button. A modal window is overlaid on the page, containing fields for Name, Mail, Phone, and Address, along with a 'Order✓' button.

**Figure 2-4: Finish order**

If you have an account your order will be success.

#### 6.3.1.4 Profile

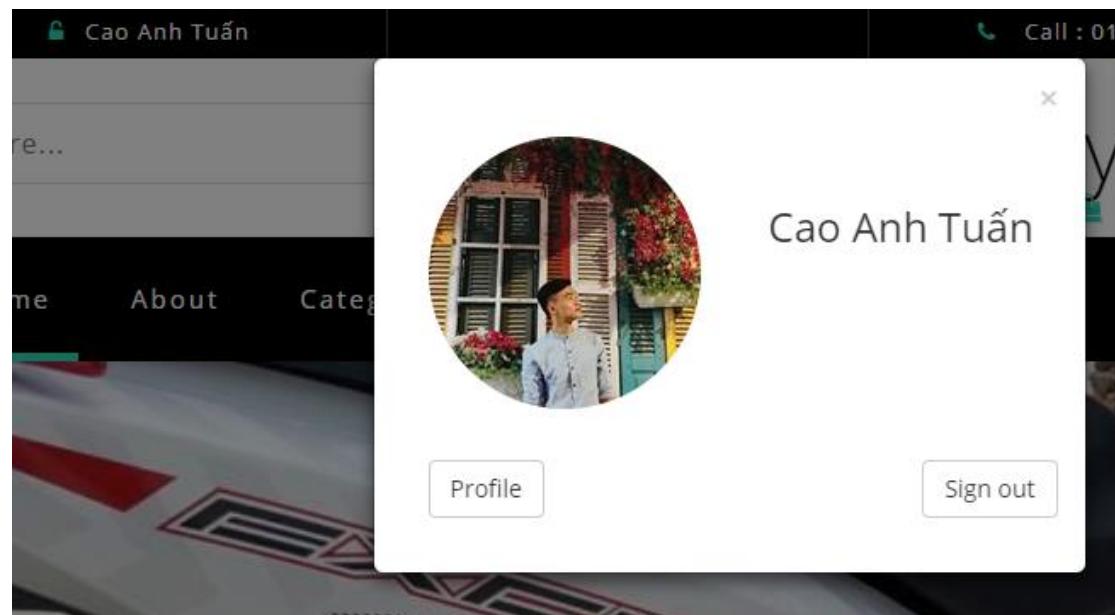
When you loggin to the page with BOS'account.

In this page, user can view, edit profile, change password, view purchase history, the user do the steps below:

Step 1: Go to Homepage.

Step 2: Click on your name in the top left corner

Step 3: You will see the popup then click to the Profile button. User will go to Profile page



**Figure 2-5: Go to Profile Page**

#### 6.3.1.4.1 View and Edit Profile

In this page, user can view and edit profile, the user do the steps below:

Step 1: Click to Account Information.

Step 2: User can change any item in account information.

Step 3: User click update to update profile

Hoang Van Anh	Edit profile picture Choose File   No file chosen
Hoang Van Anh	
anhhtv01@gmail.com	
016364045056876	
Dai Hoc FPT - Thach Hoa - Thach That - Ha N	
<b>Update</b>	

**Figure 2-6: Account Infomation**

#### 6.3.1.4.2 Change Password

In this page, user can change account password , the user do the steps below:

Step 1: Click to Change Password.

Step 2: User type current password then type new password và confirm new password

Step 3: Click on Change to change your password.

**Figure 2-7: Change Password**

#### 6.3.1.4.3 Purchase History

In this page, user can view purchase history, the user do the steps below:

Step 1: Click to Purchase History.

Step 2: User can view all the has bought.

Cake	Price	Quantity	Date Buy	Total
+THE NUT BOX	80000	1	Sun Dec 10 2017	555000
+VEGAN RED VELVET	50000	2		100000
+THE ICE CREAM PARLOUR	150000	1		150000
+FRESH FRUIT BOX	75000	3		225000

**Figure 2-10: Purchase History**

#### 6.3.1.5 Event

In Homepage User can see slide event.

User can view detail event, the user do the steps below:

Step 1: Click on the image of the event in slide on screen

**Figure 2-11: Slide event**

Step 2: Users will go to Event detail page.



SALE 50-50

A bake sale is a fundraising activity where baked goods such as doughnuts, cupcakes and cookies, sometimes along with ethnic foods, are sold. Bake sales are usually held by small, non-profit organizations, such as clubs, school groups and charitable organizations.<sup>[1]</sup> Bake sales are often set up around an area of pedestrian traffic, such as outside a grocery store or at a busy intersection near a mall. Bake sales are also a popular fund raising activities within corporations.

**Figure 2-12: Detail Event**

#### 6.3.1.6 Blog

User can view blog, the user do the steps below:

Step 1: Click on the blog button on menu bar



**Figure 2-13: Menu Bar**

Step 2: User will go to the Blog page

**Figure 2-14: Blog page**

Step 3: User can view detail of blog by click to the button Read more

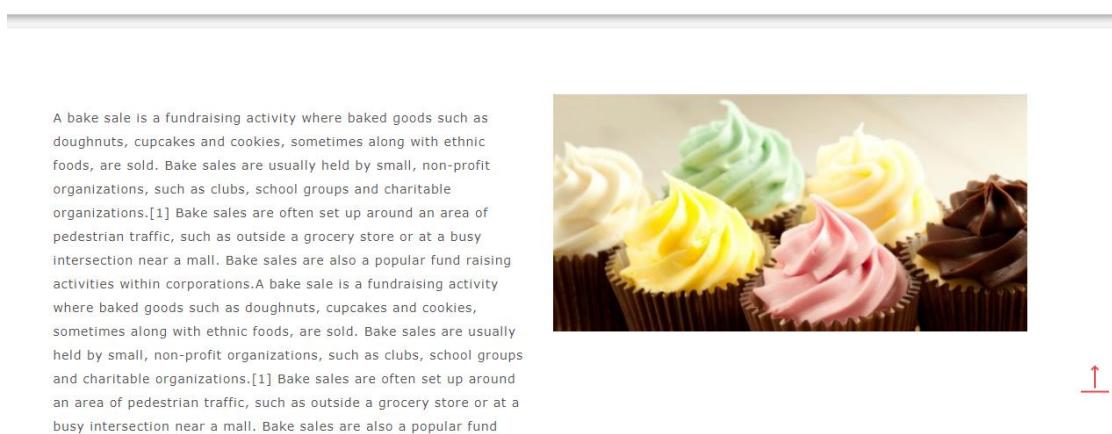
Step 4: User will go to Detail Blog page

### How to do good cake !!!!



A bake sale is a fundraising activity where baked goods such as doughnuts, cupcakes and cookies, sometimes along with ethnic foods, are sold. Bake sales are usually held by small, non-profit organizations, such as clubs, school groups and charitable organizations.<sup>[1]</sup> Bake sales are often set up around an area of pedestrian traffic, such as outside a grocery store or at a busy intersection near a mall. Bake sales are also a popular fund raising activities within corporations. A bake sale is a fundraising activity where baked goods such as doughnuts, cupcakes and cookies, sometimes along with ethnic foods, are sold. Bake sales are usually held by small, non-profit organizations, such as clubs, school groups and charitable organizations.<sup>[1]</sup> Bake sales are often set up around

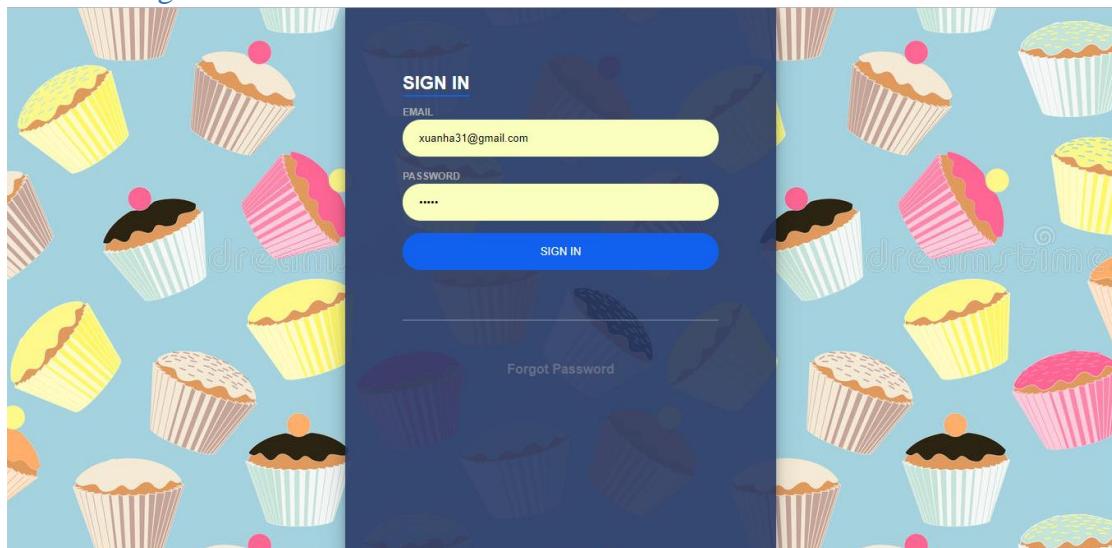
**Figure 2-15: Blog detail page**



**Figure 2-16: Blog detail page**

### 6.3.2 Admin

#### 6.3.2.1 Login



**Figure 2-17: Sign in Screen**

To login admin account, the users do the steps below:

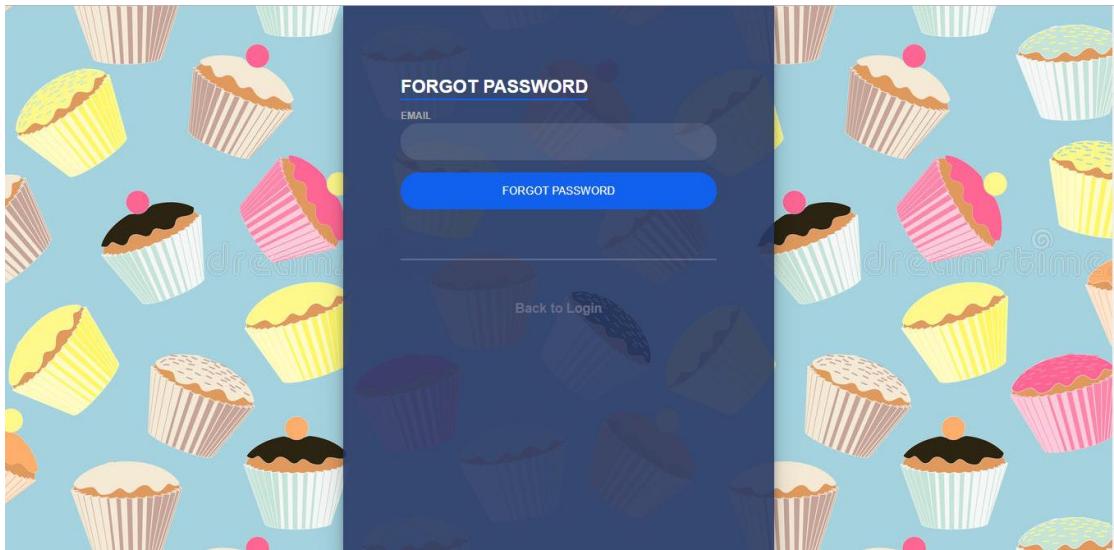
Step 1: Open browser and enter to the address bar:

<http://catbakery.herokuapp.com/loginAD/>.

Step 2: User enter username and password

Step 3: Click on Sign in button

### 6.3.2.2 Forgot Password



**Figure 2-18: Forgot password Screen**

If forgot password , the users do the steps below:

Step 1: Open browser and enter to the address bar:

<http://catbakery.herokuapp.com/loginAD/>.

Step 2: Click to the Forgot password button below sign in button.

Step 3: Enter the email of admin

Step 4: System will sent to your mail a new password with random character

### 6.3.2.3 Change password

To change admin account password, the users do the steps below:

Step 1: Click admin on the top right corner then select change password. User will go to change password page

Step 2: User enter old password then enter new password and confirm password

Step 3: Click on change to change password or cancel to cancel.

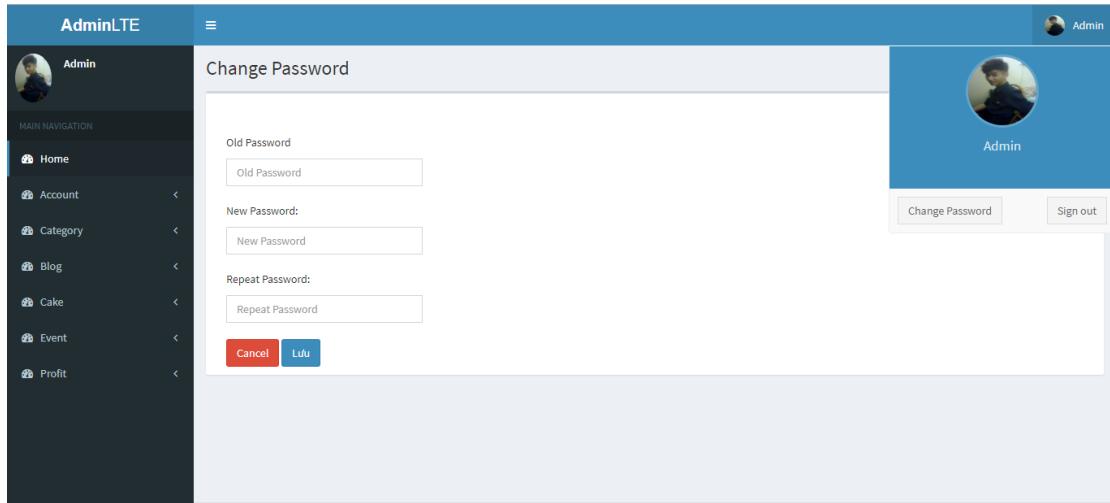


Figure 2-19: Change password Screen

#### 6.3.2.4 Account

##### 6.3.2.4.1 Manage Account

To lock/unlock a account. Admin do the step below:

Step 1: Go to Admin page, Click on “Account”. Then click on manage account. System return list account.

Step 2: Choose account and click on lock/unlock icon button to lock/unlock account.

Manage Account							
Account							
Email	Name Customer	Address	Phone Number	Date Create	Date Update	Role	Block
kiennt@gmail.com	Kien	Ha Noi	01234556781	Mon Nov 13 2017	Sat Dec 09 2017	Customer <input checked="" type="checkbox"/>	<button>Lock</button> <button>UnLock</button>
anhhtv01@gmail.com	Tuan	Phu Tho	0987455644	Mon Nov 13 2017	Sat Dec 09 2017	Customer <input checked="" type="checkbox"/>	<button>Lock</button> <button>UnLock</button>
duc@gmail.com	Duc12	Hai Duong	098555512	Mon Nov 13 2017	Sat Dec 09 2017	Staff <input type="checkbox"/>	<button>Lock</button> <button>UnLock</button> <button>Update</button>
tuyenla@gmail.com	TuyenLA	Phu Tho	0855456323	Mon Nov 13 2017	Sat Dec 09 2017	Staff <input type="checkbox"/>	<button>Lock</button> <button>UnLock</button> <button>Update</button>
xiatian2902@gmail.com	Toan	Muong La - Sơn La	0868774057	Tue Nov 21 2017	Sat Dec 09 2017	Staff <input type="checkbox"/>	<button>Lock</button> <button>UnLock</button> <button>Update</button>
tuanca@gmail.com	Cao Anh Tuan	Thach Thát - Hà Nội	01668658546	Wed Nov 22 2017	Wed Nov 22 2017	Staff <input type="checkbox"/>	<button>Lock</button> <button>UnLock</button> <button>Update</button>

Figure 2-20: Manage Account Screen

To update a staff profile. Admin do the step below:

Step 1: Go to Admin page, Click on “Account”. Then click on manage account. System return list account.

Step 2: Choose staff account and click update icon button to update account. Admin will go to update staff profile page.

Step 3: Fill the form and click save to update staff profile or click cancel to cancel the update

The screenshot shows a user interface for updating a staff account. On the left is a dark sidebar menu with icons and labels for Dashboard, Account (selected), Category, Blog, Cake, Event, and Profit. Under the Account section, 'Manage Account' is also listed. The main area is titled 'Update Account'. It contains fields for Staff Name (Duc), Password (staff), Address (Hai Duong), Phone Number (01668658546), Email (xiatian2902@gmail.com), and a Note field (213). There are 'Cancel' and 'Update' buttons at the bottom. The 'Image' field has a placeholder 'Choose File' and 'No file chosen'.

**Figure 2-21: Update Account Staff Screen**

#### 6.3.2.4.2 Insert Staff

Admin can add a new staff. Admin do the step below:

Step 1: Go to Admin page, Click on “Account”. Then click on insert staff. System return a form.

Step 2: Fill the form and click add to add to add new staff account or click cancel to cancel.

The screenshot shows the 'Insert Staff' form within the 'Account' section of the application. The left sidebar displays a navigation menu with items like Dashboard, Account (selected), Manage Account, Insert Staff (selected), Category, Blog, Cake, Event, and Profit. The main content area is titled 'Insert Staff' and contains fields for 'Staff Name', 'Image', 'Password', 'Address', 'Phone Number', 'Email', and 'Note'. Each field has a red asterisk indicating it is required. Below the fields are 'Cancel' and 'Add' buttons.

**Figure 2-22: Add Account Staff Screen**

### 6.3.2.5 Category

#### 6.3.2.5.1 Manage Category

To lock/unlock a category. Admin do the step below:

Step 1: Go to Admin page, Click on “Category”. Then click on manage category. System return list category.

Step 2: Choose account and click on lock/unlock icon button to lock/unlock category.

The screenshot shows the 'Manage Category' page. On the left is a dark sidebar with a user profile picture and the name 'Admin'. Below it is a 'MAIN NAVIGATION' section with links: Home, Account, Category (which is expanded to show 'Manage Category' and 'Insert Category'), Blog, Cake, Event, and Profit. The main content area has a header 'Manage Category' and a breadcrumb 'Home > Manage Category'. It contains a table with the following data:

Category Name	Status	Lock	UnLock
Bánh trung Thu	Status : UnLock	<button>Lock</button>	<button>UnLock</button>
Bánh Truyền Thống	Status : UnLock	<button>Lock</button>	<button>UnLock</button>
Adc	Status : UnLock	<button>Lock</button>	<button>UnLock</button>
123	Status : Lock	<button>Lock</button>	<button>UnLock</button>
asdad	Status : UnLock	<button>Lock</button>	<button>UnLock</button>

**Figure 2-23: Manage Category Screen**

### 6.3.2.5.2 Insert Category

To insert category. Admin do the step below:

Step 1: Go to Admin page, Click on “Category”. Then click on insert category. System return a form.

Step 2: Fill the form and click add to add to add new category or click cancel to cancel.

The screenshot shows the 'Insert Category' page. On the left is a dark sidebar with a user profile picture and the name 'Admin'. Below it is a 'MAIN NAVIGATION' section with links: Home, Account, Category (which is expanded to show 'Manage Category' and 'Insert Category'), Blog, Cake, Event, and Profit. The main content area has a header 'Insert Category' and a breadcrumb 'Home > Insert Category'. It contains a form with the following fields:

- Category
- Category Name
- 

**Figure 2-24: Add Category Screen**

### 6.3.2.6 Blog

#### 6.3.2.6.1 Manage Blog

To lock/unlock a blog. Admin do the step below:

Step 1: Go to Admin page, Click on “Blog”. Then click on manage blog. System return list blog.

Step 2: Choose blog and click on lock/unlock icon button to lock/unlock blog.

Title	Time Post	Status	Action
Cách làm bánh trung thu tuyết vị trà xanh	Sun, 10 Dec 2017 13:41:26 GMT	UnLock	<a href="#">Lock</a> <a href="#">Unlock</a>
Cách làm bánh trung thu rau câu vị sầu riêng cực độc	Sat, 09 Dec 2017 08:14:39 GMT	UnLock	<a href="#">Lock</a> <a href="#">Unlock</a>
Cách làm bánh trung thu thạch đậu nành sương sáo đặc đáo	Fri, 08 Dec 2017 20:59:08 GMT	UnLock	<a href="#">Lock</a> <a href="#">Unlock</a>
Cách làm bánh trung thu hình con vật ngộ nghĩnh dành tặng bé	Fri, 08 Dec 2017 20:58:40 GMT	UnLock	<a href="#">Lock</a> <a href="#">Unlock</a>
Cách làm sữa gạo Hàn Quốc ngon đúng chuẩn	Fri, 08 Dec 2017 16:52:44 GMT	UnLock	<a href="#">Lock</a> <a href="#">Unlock</a>

**Figure 2-25: Manage Blog Screen**

Admin can view detail of a blog by click on them to view detail

How to do Winter Hat Cake ????

Coat the inside of the cake pan with a generous coating of shortening and then flour well. Fill with batter and place the pan on a sheet tray to balance it in the oven. Once your cake is baked, cool for 15 minutes and flip it out of the pan to cool completely. Wrap the cake in plastic wrap and place the cake back in the pan and put it in the fridge to chill overnight. This will help it keep its shape and make it much easier to ice with buttercream when ready. Make your buttercream and cover the surface with plastic wrap. You can leave this at room temperature.

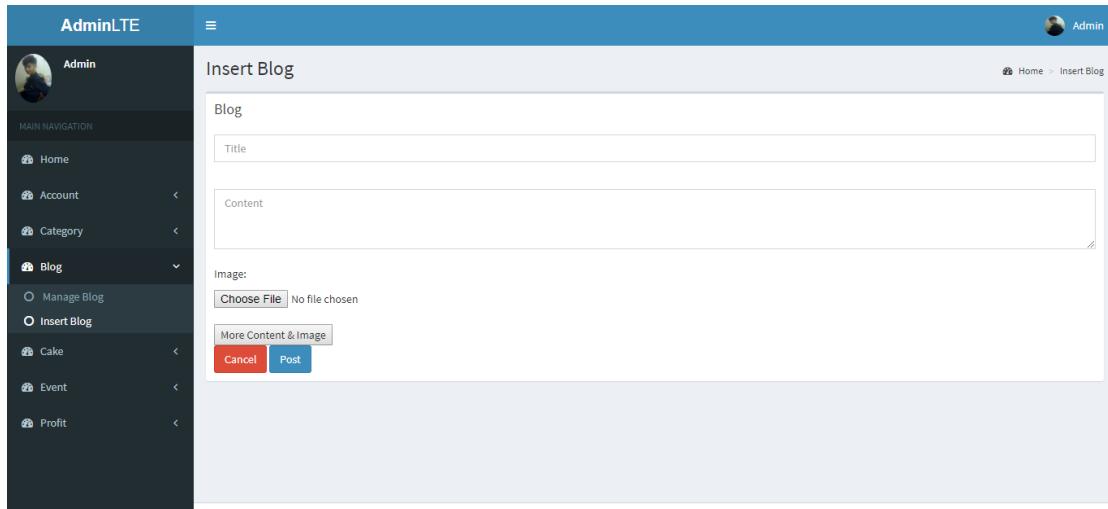
**Figure 2-26: Blog detail Screen**

### 6.3.2.6.2 Insert Blog

To insert blog. Admin do the step below:

Step 1: Go to Admin page, Click on “Blog”. Then click on insert blog. System return a form.

Step 2: Fill the form and click add to add to add new blog or click cancel to cancel.



The screenshot shows the 'Insert Blog' form within the AdminLTE interface. On the left is a dark sidebar with a user profile picture and the name 'Admin'. Below it is a 'MAIN NAVIGATION' section with links: Home, Account, Category, Blog (which is currently selected and highlighted in blue), Cake, Event, and Profit. Under the 'Blog' section, there are two options: 'Manage Blog' and 'Insert Blog', with 'Insert Blog' being the active one. The main content area has a title 'Insert Blog' at the top. It contains fields for 'Title' (a text input box) and 'Content' (a large text area). Below these is a 'Image:' section with a 'Choose File' button and a note 'No file chosen'. There is also a 'More Content & Image' link. At the bottom of the form are two buttons: 'Cancel' (red) and 'Post' (blue).

**Figure 2-27: Inserrt Blog Screen**

### 6.3.2.7 Cake

#### 6.3.2.7.1 Manage Cake

To lock/unlock a cake. Admin do the step below:

Step 1: Go to Admin page, Click on “Cake”. Then click on manage cake. System return list cake.

Step 2: Choose cake and click on lock/unlock icon button to lock/unlock cake.

Cake Name	Description	Material	Size	Price	Quantity	Status	Sell Again	Not Sell	Update
Bánh Nướng	Sản phẩm làm thủ công không có chất bảo quản	Đường hoa mai Chanh Mạch Nha Lòng đỏ trứng gà	50g	70001	12	Selling	<span>Sell Again</span>	<span>Not Sell</span>	<span>Update</span>
Bánh Nướng	Sản phẩm làm thủ công không có chất bảo quản	Đường hoa mai Chanh Mạch Nha Lòng đỏ trứng gà	100g	100000	678	Selling	<span>Sell Again</span>	<span>Not Sell</span>	<span>Update</span>
Bánh Nướng	Sản phẩm làm thủ công không có chất bảo quản	Đường hoa mai Chanh Mạch Nha Lòng đỏ trứng gà	150g	180000	10	Selling	<span>Sell Again</span>	<span>Not Sell</span>	<span>Update</span>
Bánh Dẻo Nhân Đậu Vịn	Bánh làm thủ công	Đậu xanh không vỏ Mạch nha	180g	100000	15	Not Selling	<span>Sell Again</span>	<span>Not Sell</span>	<span>Update</span>

**Figure 2-28: Manage Cake Screen**

To update a cake. Admin do the step below:

Step 1: Go to Admin page, Click on “Cake”. Then click on manage cake. System return list cake.

Step 2: Choose cake and click update icon button to update cake. Admin will go to update Cake page.

Step 3: Fill the form and click save to update ake or click cancel to cancel the update

Manage Cake

Update Cake

Cake Name \*

Description \*

Size \*

Price \*

Quantity \*

Image \*

Choose File | No file chosen

Cancel Update

**Figure 2-29: Update Cake Screen**

### 6.3.2.7.2 Insert Cake

To insert cake. Admin do the step below:

Step 1: Go to Admin page, Click on “Cake”. Then click on insert cake. System return a form.

Step 2: Fill the form and click add to add to add new cake or click cancel to cancel.

The screenshot shows the 'Insert Cakes' form within the AdminLTE application. The left sidebar has a dark theme with a user icon and the word 'Admin'. The main navigation under 'MAIN NAVIGATION' includes 'Home', 'Account', 'Category', 'Blog', 'Cake' (with 'Manage Cake' and 'Insert Cake' options), 'Event', and 'Profit'. The 'Insert Cake' option is selected. The right panel title is 'Insert Cakes' and shows a breadcrumb path 'Home > Insert Cakes'. The form itself is titled 'Cakes' and contains the following fields:

- Cake Type \***: Radio buttons for 'Bánh trung Thu', 'Bánh Truyền Thống', 'Adc', and 'ascd'.
- Image \***: A file input field with a 'Choose File' button and a note 'No file chosen'.
- Cake Name \***: A text input field.
- Material**: A text input field with a 'More Material' link.
- Description \***: A text input field.
- Size \***: A text input field.
- Price \***: A text input field.
- Quantity \***: A text input field.
- Buttons**: 'Cancel' and 'Add' buttons at the bottom.

**Figure 2-30: Insert Cake Screen**

### 6.3.2.8 Event

#### 6.3.2.8.1 Manage Event

To lock/unlock a event. Admin do the step below:

Step 1: Go to Admin page, Click on “Event”. Then click on manage event. System return list event.

Step 2: Choose event and click on lock/unlock icon button to lock/unlock event.

Mỗi dịp lễ là khoảng thời gian để nhìn lại và trân trọng những con người đã làm nên một năm đặc biệt với chúng ta. Nhân dịp năm mới, chúng tôi xin gửi lời cảm ơn chân thành đến tất cả khách hàng đã luôn ủng hộ Fresh Garden. Chúc quý khách và gia đình một năm thành công, vui tươi và ngập tràn hạnh phúc. Chào đón năm mới Bình Thành và để tri ân khách hàng, Fresh Garden ưu đãi 50% tất cả sản phẩm, duy nhất trong ngày 06/02/2016. Hãy nhanh chân ghé các cửa hàng Fresh Garden để tận hưởng khuyến mại hấp dẫn này nhé. Bạn sẽ có cơ hội thường thức các sản phẩm bánh của Fresh Garden với mức giá tuyệt vời không thể bỏ lỡ. Chương trình áp dụng cho tất cả sản phẩm bánh tại các cửa hàng trong hệ thống của bánh tươi Fresh Garden. Chương trình không áp dụng tại Fresh Garden Vinmec và Fresh Garden sân bay Nội Bài. Chương trình không áp dụng cho các sản phẩm bánh đặt ngoài. Chương trình không áp dụng

**Figure 2-31: Manage Event Screen**

### 6.3.2.8.2 Insert Event

To insert event. Admin do the step below:

Step 1: Go to Admin page, Click on “Event”. Then click on insert event. System return a form.

Step 2: Fill the form and click add to add to add new event or click cancel to cancel.

**Figure 2-32: Insert Event Screen**

### 6.3.2.9 Profit

#### 6.3.2.9.1 Manage Bill

To Confirm/Unconfirm a bill. Admin do the step below:

Step 1: Go to Admin page, Click on “Profit”. Then click on manage bill. System return list Bill.

Step 2: Choose bill and click on Confirm/Unconfirm icon button to Confirm/Unconfirm bill.

Admin can search Bill by date with search bar

Email	Name Customer	Phone Number	Bill	Quantity	Price	Time Order	Delivery Location	Note	Status
kiennt@gmail.com	Kien	01234556781	Bánh Nướng Bánh Déo Nhân Đậu Xanh	3 2	740000	Tue, 14 Nov 2017 05:47:11 GMT	Ha Noi		Confirm UnConfirm
anhhtv01@gmail.com	Tuan	0987455644	Bánh Tết	3	105000	Thu, 16 Nov 2017 02:23:28 GMT	Phu Tho		Confirm UnConfirm

**Figure 2-33: Manage Bill Screen**

#### 6.3.2.9.2 Revenue

Admin can view revenue of the store by date, month, quarter, years.

Bill Date	Details	Price
Tue, 14 Nov 2017 05:47:11 GMT	Bánh Nướng x 3 Bánh Déo Nhân Đậu Xanh x 2	740000
Thu, 16 Nov 2017 02:23:28 GMT	Bánh Tết x 3	105000
		Total Profit : 845000

**Figure 2-34: Revenue Screen**

## 6.4 Staff

### 6.4.1 Sign in

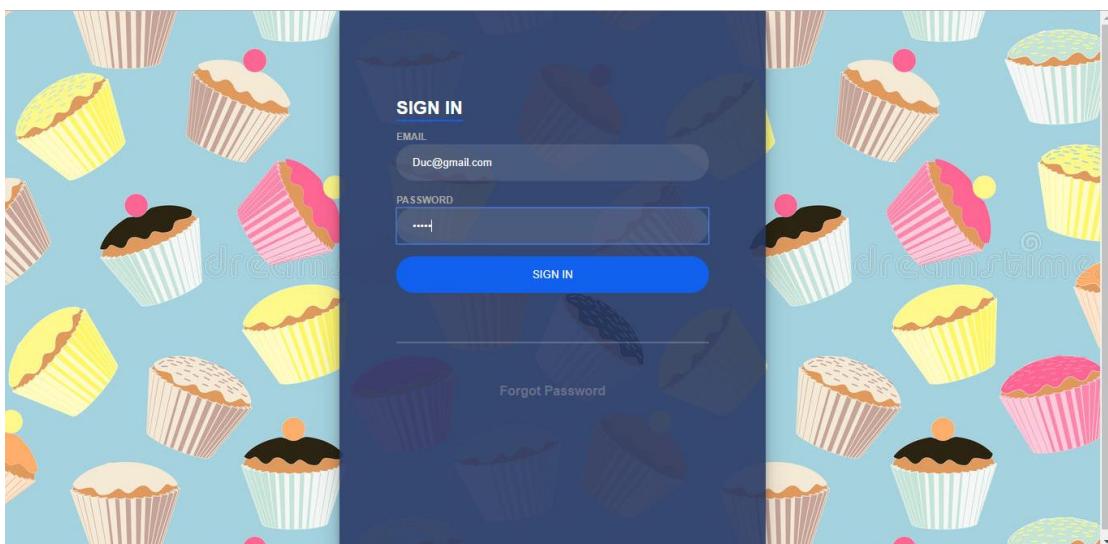
To login staff account, the users do the steps below:

Step 1: Open browser and enter to the address bar:

[http://catbakery.herokuapp.com/loginStaff/.](http://catbakery.herokuapp.com/loginStaff/)

Step 2: User enter username and password

Step 3: Click on Sign in button

**Figure 2-35: Sign in Screen**

### 6.4.2 Forgot Password

If forgot password , the users do the steps below:

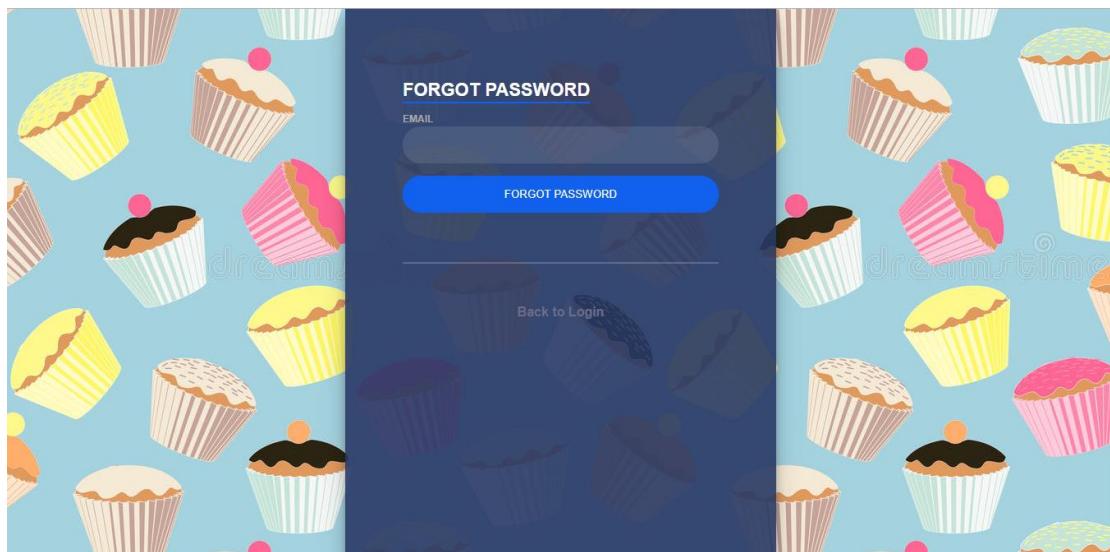
Step 1: Open browser and enter to the address bar:

[http://catbakery.herokuapp.com/loginStaff/.](http://catbakery.herokuapp.com/loginStaff/)

Step 2: Click to the Forgot password button below sign in button.

Step 3: Enter the email of staff

Step 4: System will sent to your mail a new password with random character



**Figure 2-36: Forgot password Screen**

#### 6.4.3 Change Password

To change staff account password, the users do the steps below:

Step 1: Go to Staff page, Click on “Change Password”.

Step 2: User enter old password then enter new password and confirm password

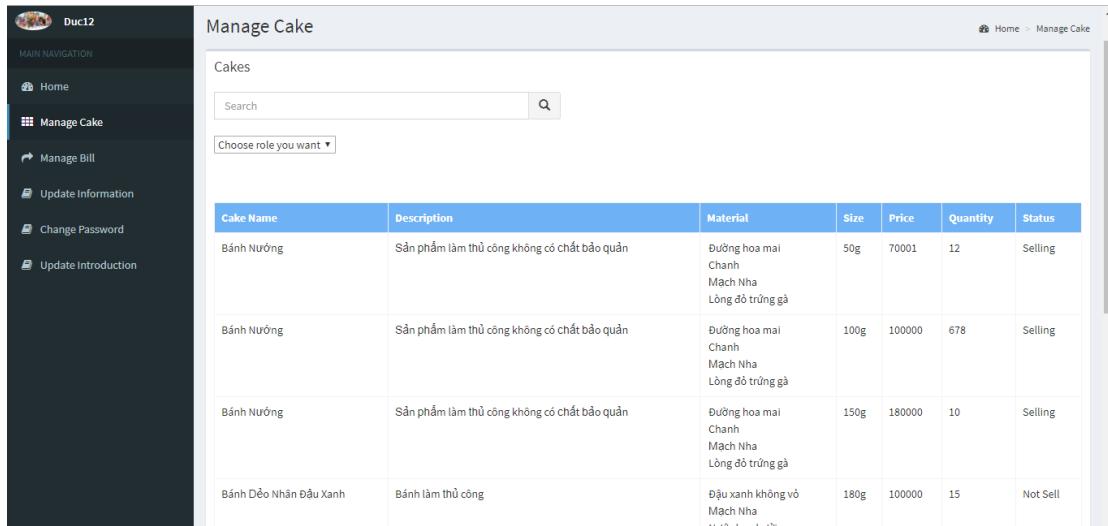
Step 3: Click on change to change password or cancel to cancel.

 A screenshot of the 'Change Password' screen from the Bakery Store Online application. The interface includes a dark sidebar on the left with a user profile icon and the name 'Duc12'. The sidebar also contains a 'MAIN NAVIGATION' section with links: Home, Manage Cake, Manage Bill, Update Information, Change Password (which is highlighted in blue), and Update Introduction. The main content area has a light gray background. At the top right of the content area, there are links for 'Home' and 'Change Password'. The central part of the screen is titled 'Change Password'. It contains three input fields with validation requirements: 'Old Password \*' (with a placeholder 'Old Password'), 'New Password \*' (with a placeholder 'New Password'), and 'Repeat Password \*' (with a placeholder 'Repeat Password'). Below these fields are two buttons: a red 'Cancel' button and a blue 'Change Password' button. The overall layout is organized and user-friendly, designed for password management tasks.

**Figure 2-37: Change password Screen**

#### 6.4.4 Manage Cake

Staff can view all the cake in store and can search cake by category and name



Cake Name	Description	Material	Size	Price	Quantity	Status
Bánh Nướng	Sản phẩm làm thủ công không có chất bảo quản	Đường hoa mai Chanh Mạch Nha Lòng đỏ trứng gà	50g	70001	12	Selling
Bánh Nướng	Sản phẩm làm thủ công không có chất bảo quản	Đường hoa mai Chanh Mạch Nha Lòng đỏ trứng gà	100g	100000	678	Selling
Bánh Nướng	Sản phẩm làm thủ công không có chất bảo quản	Đường hoa mai Chanh Mạch Nha Lòng đỏ trứng gà	150g	180000	10	Selling
Bánh Dẻo Nhân Đậu Xanh	Bánh làm thủ công	Đậu xanh không vỏ Mạch Nha	180g	100000	15	Not Sell

**Figure 2-38: Manage Cake Screen**

#### 6.4.5 Manage Bill

To Confirm/Unconfirm a bill. Staff do the step below:

Step 1: Go to Staff page, Click on “Manage Bill”. System return list Bill.

Step 2: Choose bill and click on Confirm/Unconfirm icon button to Confirm/Unconfirm bill.

Staff can search Bill by date with search bar

Email	Name Customer	Phone Number	Bill	Price	Time Order	Delivery Location	Status		
kiennt@gmail.com	Kien	01234556781	Bánh Nướng x 3 Bánh Déo Nhân Đậu Xanh x 2	740000	Tue, 14 Nov 2017 05:47:11 GMT	Ha Noi	Confirm	<button>Confirm</button>	<button>UnConfirm</button>
anhhtv01@gmail.com	Tuan	0987455644	Bánh Tết x 3	105000	Thu, 16 Nov 2017 02:23:28 GMT	Phu Tho	Confirm	<button>Confirm</button>	<button>UnConfirm</button>

**Figure 2-39: Manage Bill Screen**

#### 6.4.6 Update information

To update profile. Staff do the step below:

Step 1: Go to Staff page, Click on “Update Infomation”.

Step 3: Fill the form and click save to update profile or click cancel to cancel the update

**Figure 2-40: Update Staff Information Screen**

### 6.4.7 Update Introduction

Staff can update the information of the store (Change location, email, phone number)

The screenshot shows the 'UpdateIntroduction' page. On the left is a sidebar with a user profile picture and the name 'Duc12'. Below it is a 'MAIN NAVIGATION' menu with links: Home, Manage Cake, Manage Bill, Update Information, Change Password, and Update Introduction (which is highlighted in blue). The main content area has a title 'UpdateIntroduction'. It contains three input fields: 'Address \*' with 'Ha Noi 2', 'Phone Number \*' with '0972550995', and 'Email \*' with 'kienti2012@gmail.com'. Below these is a 'Choose File' button with the placeholder 'No file chosen'. At the bottom are two buttons: a red 'Cancel' button and a blue 'Lưu' (Save) button.

**Figure 2-41: Update Introduction Screen**

## 7 PROJECT RESULT REPORT

### 7.1 Product Judging

#### 7.1.1 Advantage of Bakery Store Online system

The idea of Bakery Store Online project is to develop a website where can look for cakes with many different flavors. As the goals that we have set before started project, the Bakery Store Online system has been developed in right direction. And what it provides to users is listed below:

- Customer: customer can view cake, refer to the formula in the blog, order cake, view purchase history
- Staff: staff can manage quantity of cake, manage contact, manage order (check confirm or unconfirm)
- Admin: Admin can manage customer, staff, benefit...

#### 7.1.2 Current Limitations

- Some screen can't display well on Mobile or screen which has resolution lower than 1024x768.
- Website not support for English language.
- Not support payment methods online.
- Not support payment by mobile cards..

### 7.1.3 Expectation in Future

- Support multi-language and other payment method such as visa and master card to get funding from people in other countries.
- Support payment by mobile cards such as Viettel, Mobi, Vina, ...
- App for Android and IOS which supports members manage their projects and admin can manage system anywhere by smart phone.
- Expand server to increase number of users.

## 7.2 Project Summary

<b>Project Title</b>	<b>Bakery Store Online</b>		
<b>Project Supervisor</b>	Nguyễn Văn Sang		
<b>Committers (6)</b>			
<b>1</b>	<b>Project Manager</b>	Phạm Thị Minh Phương	<a href="mailto:phuongptmse03773@fpt.edu.vn">phuongptmse03773@fpt.edu.vn</a>
<b>2</b>	<b>Team member</b>	Nguyễn Xuân Hà	<a href="mailto:hanxse03484@fpt.edu.vn">hanxse03484@fpt.edu.vn</a>
<b>3</b>	<b>Team member</b>	Hoàng Anh Phương	<a href="mailto:phuonghase03715@fpt.edu.vn">phuonghase03715@fpt.edu.vn</a>
<b>4</b>	<b>Team member</b>	Cao Anh Tuấn	<a href="mailto:tuancase03737@fpt.edu.vn">tuancase03737@fpt.edu.vn</a>
<b>5</b>	<b>Team member</b>	Ngô Trung Kiên	<a href="mailto:se03736@fpt.edu.vn">se03736@fpt.edu.vn</a>
<b>Main objective</b>		Develop a system which will provide to users functionality to collect funding for their project and back to get the amazing products from creative projects.	
<b>Purpose</b>		Provide a Bakery Store Online online on website.	
		<b>Plan</b>	<b>Actual</b>
<b>Start Date</b>		07/09/2017	07/09/2015
<b>End Date</b>		25/12/2017	23/12/2015
<b>Releases version</b>		1.0	1.0
<b>Duration</b>		81	70
<b>Effort (person-day)</b>		405	350
<b>Test cases</b>		1000	1192
<b>Defects</b>		200-300	66
<b>Document pages</b>		~ 3000 pages	~ 3500 pages
<b>Lines of code (Nodejs)</b>		~ 10000	~ 15000
<b>Lines of code (HTML)</b>		~ 5000	~ 10000
<b>LOC (Total)</b>		~ 25000	~ 38000

Table 7-1: Project summary

**Knowledge:**

- Putting the learned knowledge into practice.
- Improve skills: teamwork, leadership, research and self-study, plan and manage time.
- Technical: Node.JS, MongoDB
- Tools: Atom, Robo3T, SmartGit, Office tools.

### 7.3 Lesson Learnt

- The success of a project is largely dependent on the skills and strengths of the people involved. Therefore, a project needs to have a dedicated, talented set of individuals working towards a common goal.
- When things go wrong, know who you can turn to for help.
- Always disclose information to those, who will need it. This is not the time or place for obtaining an edge over another by keeping crucial data close to your chest. People, who know what is expected of them and have the means of doing so, will play a pivotal role in making the project a success.
- Know what resources are available. Not just those under your purview but those which are at the discretion of other teams. Sometimes, others may be happy to help.
- Be open to change. Sometimes, you may find that the things you knew along may not be correct at this given time, under these specific conditions.
- Good communication is that which will stop mistakes from becoming failures.
- Please think carefully before making a decision because if the wrong decisions you'll have to redo it from scratch.

## 8 REFERENCES

[1]. Paris Gateaux

<http://parisgateaux.vn/>

[2]. Madame Huong

<https://madamehuong.vn/>

[3]. Wikipedia, 2012, IBM Rational Unified Process, viewed 30 September 2013

[http://en.wikipedia.org/wiki/IBM\\_Rational\\_Unified\\_Process](http://en.wikipedia.org/wiki/IBM_Rational_Unified_Process)

[4]. Capstone Project Report Teamplate – FPT University, Ha Noi, Viet Nam.

[5]. Report document guidelines (2013) – FPT University, Ha Noi, Viet Nam.

[6]. Template\_Project\_Plan, (2010), FPT University, Hanoi, Vietnam

[7]. Template\_Software\_Requirements\_Specification, (2010), FPT Software, Hanoi, Vietnam

[8]. Template\_Software\_Design\_Template, (2010), FPT Software, Hanoi, Vietnam

[9]. Templates for Test case & Defect log, FPT Software, Hanoi, Vietnam

[10]. The Unified Modeling Language diagram

<http://www.uml-diagrams.org/>