香港浸會大學
HONG KONG BAPTIST UNIVERSITY

Computer Science Department

B.Sc. (Hons.)

In

Computing and Information Systems

Moving System

2015-2016

Ko Sing Chiu

14200163

Moving System

by

Ko Sing Chiu

Submitted in partial fulfillment of the requirements for the degree of

Bachelor of Science (Honours)
in Computing and Information Systems

Hong Kong Baptist University

April, 2016

Declaration

I hereby declare that all the work done in this Final Year Project is of my independent effort. I also certify that I have never submitted the idea and product of this Final Year Project for academic or employment credits.

_____

Ko Sing Chiu

Date: _____

Hong Kong Baptist University
Computer Science Department

We hereby recommend that the Final Year Project submitted by KO SING CHIU entitled “Moving System” be accepted in partial fulfillment of the requirement for the degree of Bachelor of Science (Honours) in Computing and Information Systems.

Dr. Joe C.K. YAU Supervisor	Dr. WONG, Kelvin Chi Kuen Observer
Date: _____	Date: _____

Acknowledgement

The Final Year project provides me a valuable opportunity to work on a system practically. I learn a lot from the Honours project and get help from my supervisor and observer. I would like to thank my supervisor, Dr. Joe C.K. Yau, for his valuable guideline and advice within the year. I would also like to thank my observer, Dr. WONG, Kelvin Chi Kuen for being my observer.

Moving System

by

Ko Sing Chiu

Computer Science Department
Hong Kong Baptist University

ABSTRACT

Moving System is a web and mobile application. By using Moving System, user can have a better management of orders and closer communication with customers.

Moving System provides the customers with online ordering services. At the same moment, it can help the admin staff to calculate order price and generate some documents related the order. In addition, it helps the admin staffs to manage and supervise the whole business flow. Besides the general order information, its mobile application can offer the drivers with accurate path between different delivery destinations. Consequently, the business processes can be simplified and become more efficient.

TABLE OF CONTENTS

Chapter 1

Introduction

CONTENTS OF CHAPTER 1

1.1 System Objectives

Moving system aims to serve the customers and the company staffs to simplify the ordering process including house moving, office moving and storage service. It is also a platform to supervise the whole ordering process so the manager can review their orders easily and manage their delivery team efficiently.

In this project, the moving system is designed to provide the customers and company staff including admin staffs and drivers with lots of convenience. It is Business-to-Customer (B2C) business and the company receives orders from the customers at the beginning of the orders. Therefore, the system helps to collect, process and save the orders.

1.2 Background

About the ordering process, it starts when receiving the orders from the customers. The company assigns some staffs to observe the house or office and quote the price for this moving. If the customer accepts the price, the staffs will pack everything which needs to be moved and start moving. The customer will pay for this order after checking all of the furniture and other things when the moving is finished.

On the other hand, for the admin staff, they do not only receive the order but also manage their own delivery team. They need to choose the suitable truck and manpower for the moving. Also, they need to supervise and review their orders for improving their service quality.

1.3 Present Situation

Nowadays, a lot of moving companies use Excel to store their staffs, customers and ordering information. As a result, they need to spend lots of time to find out the specific record and manage all of the information when the business is bigger and bigger. Also, it is difficult for them to supervise and review their orders because they need to spend lots of time to find, organize and present them on the report.

Before starting the moving, the admin staffs need to plan and schedule their drivers and assign suitable truck and manpower. At the same moment, they may need to order boxes or other things for the moving or storage. Also, they need to remind their delivery team about something they need to pay attention in the moving. In addition, they make reports for the supervising and reviewing the previous orders.

1.4 Problem / Improvement Areas

Inconvenient ordering process

The customer usually makes an order through the phone or email and it is not convenient enough for the ordering process. Also, the website only provide the general information about the moving and storage and it is wasted if the company does not use this tool to let the customers make orders.

Time consuming over quoting an order price

The admin staffs need to spend some time to assign another staff to following the order including visiting the location, calculating the time the order needed and counting the delivery items. These processes are time consuming and can be simplified.

Difficult to generate the report for supervising and reviewing orders

The admin staffs need to spend lots of time for finding out and organizing the previous to make the reports. In the process, it is easy to make the mistakes and wastes time. Also, the business may be affected if the managers make the wrong decision because of the wrong report.

1.5 Proposed System

Expanding to the online customers

The system provides the customers with online ordering service and let the customers know the approximate price for the moving or storage through the online ordering form. As a consequence, it is more convenient for the customers to make the orders.

Simplifying the ordering and driver scheduling process

The system can count the delivery items and calculate the duration between different delivery locations for quoting an order price. Also, it can generate the related documents including quotation, invoice and receipt at the same moment; therefore, the ordering process is simplified. Moreover, the system can list the drivers who are available and orders for them to plan and schedule the moving.

Easy to supervise and review the orders through the report

The system saves all of the information about the orders so it can organize them and make the report easily. As a result, the managers can supervise and review the orders in the quick and accurate manner. It also saves lots of time for the admin staffs and prevents the human mistakes

Chapter 2

Project Plan

CONTENTS OF CHAPTER 2

2.1 Resource Implications

In order to develop an efficient system, it is necessary to look for some resources and information about the specific industry and technical skills on the internet. The problems of the transporting industry can be found out by searching on different relevant companies websites. The source of customers and flow of services can be increased and improved after designing a solution, Moving System.

Observation (Website Visiting)

1. Star Moving Service Company

http://www.starmovinghk.com/service2.php?ID=3

Star Moving Service Company provides customers with services including house moving and office moving. Different types of service are described in its own company website. Also, it introduces the whole moving flow; therefore, the customers can clearly know the whole service provided. At the same moment, the system can take that as reference for designing the system to make the service smoother.

2. Aman Moving Services

http://www.aman01.com/add01/tc/05_02.asp

Aman Moving Services company also offer customers with different moving services. At the same time, it provides an online price list and calculates the approximate price for the reference. As a result, the system can regard its price list as reference for designing the online quoting and order making services.

3. Santa Fe Relocation Services

http://www.santaferelo.com/

Santa Fe Relocation Service Company is a global company and offers customers with different moving services and even immigration service. As you can see, this industry can be very large and that means the information management is more and more important for the company. This factor can be considered when developing and designing the system for improving the efficiency of data and letting the managers know the situation of the whole company in the report easily.

Software

In this Final Year project, I normally use the open-source software. For the back end part, Apache Server (XAMPP) and MySQL are used for accessing website and storing business information. For the front end part, NetBeans is used to develop HTML and PHP based website and Xcode is used to develop iOS mobile application. All of the software is free and open-source; therefore, the development cost can be minimized.

2.2 Development and Operating Costs

Project Annual Development Cost

Development cost

Quantity	Project Position	Working Hours	Hourly Rate ($)	Total Salary ($)
1	System Analyst	300	50	15000
1	System Designer	100	50	5000
1	Programmer	400	40	16000

New hardware and software

Quantity	Item	Unit Price ($)	Total Cost ($)
1	Web and Database Server	10000	10000
1	Microsoft Windows Server 2012	6835	6835
N/A	Apache Tomcat Server	0	0
N/A	Secure Socket Layer (SSL)	0	0
N/A	MySQL	0	0
N/A	NetBeans IDE	0	0

Total development cost: $52835

Project Annual Operating Cost

Personal

Quantity	Project Position	Hours / Month	Hourly Rate ($)	Total Salary ($)
1	System Administrator	30	50	18000
1	Programmer / Analyst	20	45	10800

Other expense

Item	Total Expense ($)
Hardware Maintenance Cost	2000
Software Maintenance Cost	2000

Total project annual operating cost: $32800

2.3 Tangible and Intangible Benefit

Tangible Benefits

- Improvement in productivity of process
- Expansion of customer source
- Increase of business income

Intangible Benefits

- 24 hours customer ordering service
- Easy to supervise productivity of drivers
- Easy to check the incoming delivery and the order records

2.4 Cost-benefits Analysis

Assumption

	Before using Moving System	**After using Moving System**
Average order income per order	5000	5000
Average number of order per month	30	40
Average total income per month	150000	200000
Increased sales ($)	**50000**	

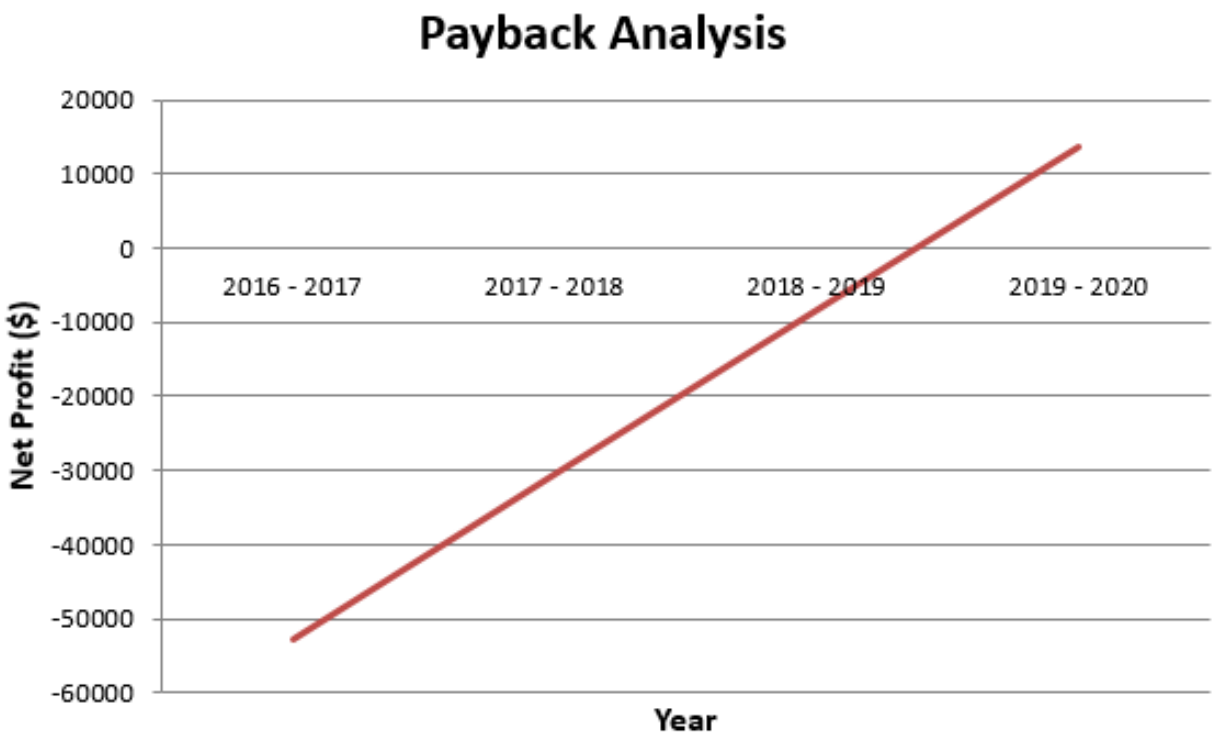
	Before using Moving System	**After using Moving System**
Average complaint per month	2	1
Reduced complaint income ($)	**5000**	

Total benefits: $50000 + $5000 = $55000

Based on above assumption, the payback analysis is in the following and the project can be break-even in 4 years.

Payback Analysis

	2016 - 2017	2017 - 2018	2018 - 2019	2019 - 2020	Total
Benefits					
Increased sales		50000	50000	50000	150000
Reduction in customer complaint calls		5000	5000	5000	15000
Total Benefits		55000	55000	55000	165000
Present Value Total Benefits		50000	45455	41323	136778
Development Costs					
Web and Database server	10000	0	0	0	10000
Server OS and Database	6835	0	0	0	6835
Development labor	36000	0	0	0	36000
Total Development Costs	52835	0	0	0	52835
Operational Costs					
Operational labor		28800	28800	28800	86400
Hardware maintenance		2000	2000	2000	6000
Software maintenance		2000	2000	2000	6000
Total Operational Costs		32800	32800	32800	98400
Total Costs	52835	32800	32800	32800	151235
Present Value Total Costs	52835	29819	27108	24644	134406
NPV (PV Total Benefits - PV Total Costs)					2372
Total Benefits - Total Costs	-52835	22200	22200	22200	13765
Cumulative Net Cash Flow	-52835	-30635	-8435	13765	
Return on Investment (ROI)	9.10 %				
Break-even Point	3.38 years				

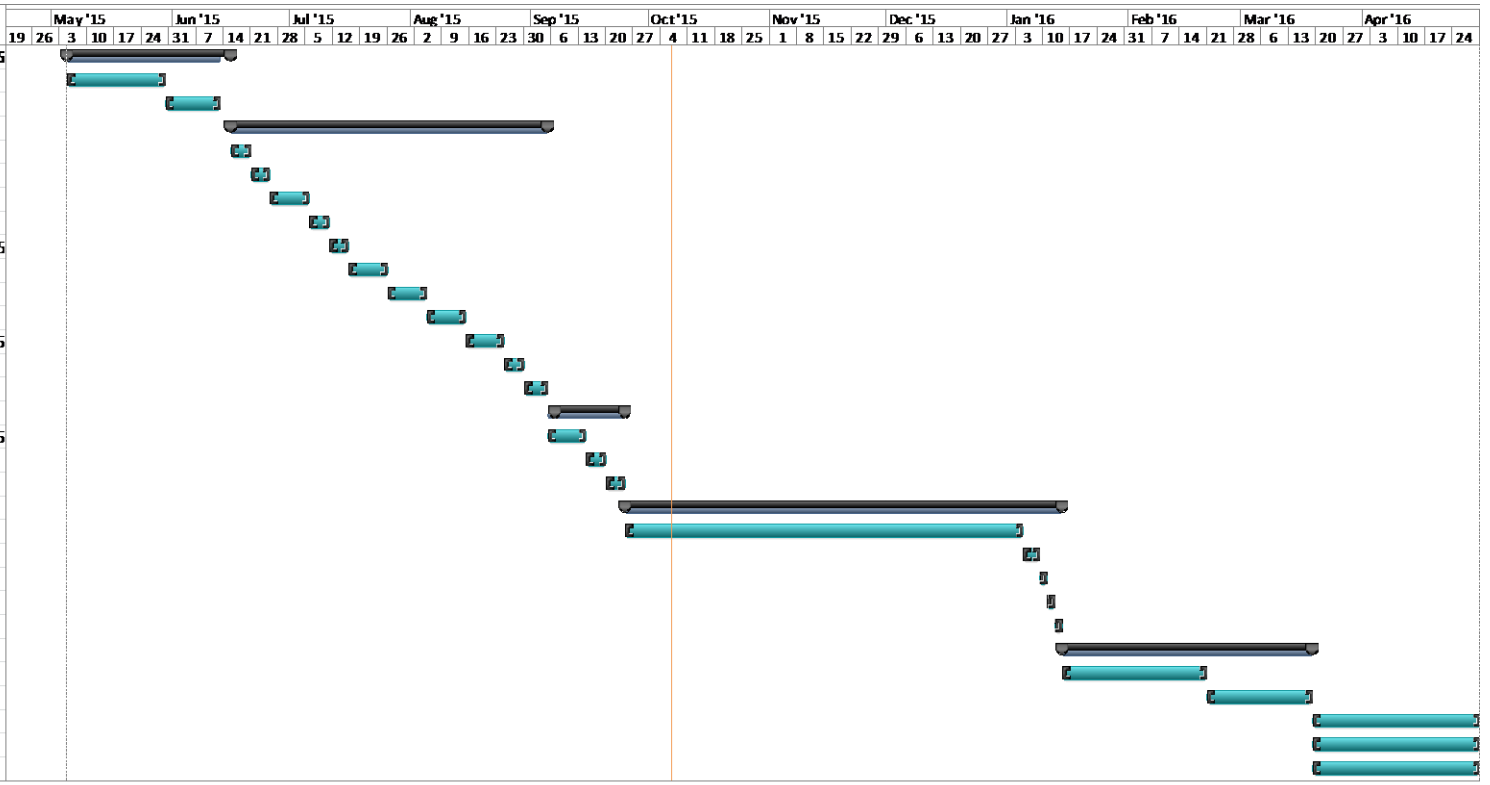


2.5 Development Schedule

Task	Start Date	End Date	Day Count
Project Proposal	5-May-15	15-Jun-15	30
Project Objectives	5-May-15	29-May-15	19
Problem Identification	5-May-15	15-May-15	9
Proposed Solution	16-May-15	29-May-15	10
Project Plan	29-May-15	15-Jun-15	11
Feasibility Studies	29-May-15	10-Jun-15	8
Project Development Schedule	11-Jun-15	11-Jun-15	1
Cost Estimation	12-Jun-15	15-Jun-15	2
First Progress Report	16-Jun-15	4-Sep-15	59
System Analysis	16-Jun-15	4-Sep-15	59
Source information	16-Jun-15	20-Jun-15	5
System Overview	21-Jun-15	25-Jun-15	5
Use Cases Diagram	26-Jun-15	5-Jul-15	7
Actor Descriptions	6-Jul-15	10-Jul-15	5
Use Cases Narrative	11-Jul-15	15-Jul-15	4
Entity Relationship Diagram	16-Jul-15	25-Jul-15	8
Class Diagram	26-Jul-15	4-Aug-15	8
Sequence Diagram	5-Aug-15	14-Aug-15	8
Activity Diagram	15-Aug-15	24-Aug-15	7
Process Specifications	25-Aug-15	29-Aug-15	5
Acceptance Criteria	30-Aug-15	4-Sep-15	6
System Design	5-Sep-15	24-Sep-15	14
Input & Output Design	5-Sep-15	14-Sep-15	7
Interface Design	15-Sep-15	19-Sep-15	5
Database Design	20-Sep-15	24-Sep-15	5
Midyear Demonstration	6-Jan-16	8-Jan-16	3
Second Progress Report	5-Sep-15	14-Jan-16	80
System Implementation	25-Sep-15	14-Jan-16	72
Program Coding	25-Sep-15	4-Jan-16	4
Test Plan	5-Jan-16	8-Jan-16	2
Black Box Testing	9-Jan-16	10-Jan-16	2
Stress Testing	11-Jan-16	12-Jan-16	2
Browser Test	13-Jan-16	14-Jan-16	2
Final Report	15-Jan-16	18-Mar-16	46
Reports Revise & Preparation	15-Jan-16	20-Feb-16	27
User Manual	21-Feb-16	18-Mar-16	21
Presentation and Demo Preparation	April-16		
Poster Session Preparation			
Final Submission Preparation			

Gantt chart

ID	Task Mode	Task Name	Duration	Start	Finish
1		**Prioject Proposal**	**30 days**	**Tue 5/5/15**	**Mon 6/15/15**
2		Project Objectives	19 days	Tue 5/5/15	Fri 5/29/15
3		Project Plan	11 days	Sat 5/30/15	Fri 6/12/15
4		**System Analysis**	**59 days**	**Tue 6/16/15**	**Fri 9/4/15**
5		Source Information	5 days	Tue 6/16/15	Sat 6/20/15
6		System Overview	5 days	Sun 6/21/15	Thu 6/25/15
7		Use Case Diagram	7 days	Fri 6/26/15	Sun 7/5/15
8		Actor Descriptions	5 days	Mon 7/6/15	Fri 7/10/15
9		Use Case Narrative	4 days	Sat 7/11/15	Wed 7/15/15
10		Entity Relationship Diagram	8 days	Thu 7/16/15	Sat 7/25/15
11		Class Diagram	8 days	Sun 7/26/15	Tue 8/4/15
12		Sequence Diagram	8 days	Wed 8/5/15	Fri 8/14/15
13		Activity Diagram	7 days	Sat 8/15/15	Mon 8/24/15
14		Process Specifications	5 days	Tue 8/25/15	Sat 8/29/15
15		Acceptance Criteria	6 days	Sun 8/30/15	Fri 9/4/15
16		**System Desgin**	**14 days**	**Mon 9/7/15**	**Thu 9/24/15**
17		Input & Output Design	7 days	Sat 9/5/15	Mon 9/14/15
18		Interface Design	5 days	Tue 9/15/15	Sat 9/19/15
19		Database Design	5 days	Sun 9/20/15	Thu 9/24/15
20		**System Implementation**	**80 days**	**Fri 9/25/15**	**Thu 1/14/16**
21		Program Coding	72 days	Fri 9/25/15	Mon 1/4/16
22		Test Plan	4 days	Tue 1/5/16	Fri 1/8/16
23		Black Box Testing	2 days	Sat 1/9/16	Sun 1/10/16
24		Stress Testing	2 days	Mon 1/11/16	Tue 1/12/16
25		Browser Test	2 days	Wed 1/13/16	Thu 1/14/16
26		**Final Report**	**46 days**	**Fri 1/15/16**	**Fri 3/18/16**
27		Reports Revise & Preparation	27 days	Fri 1/15/16	Sat 2/20/16
28		User Manual	21 days	Sun 2/21/16	Fri 3/18/16
29		**Presentation and Demo Preparati**	**32 days**	**Sat 3/19/16**	**Sat 4/30/16**
30		**Poster Session Preparation**	**32 days**	**Sat 3/19/16**	**Sat 4/30/16**
31		**Final Submission Preparation**	**32 days**	**Sat 3/19/16**	**Sat 4/30/16**



2.6 Recommendation

The highlights of the feasibility analysis are as follows:

Technical Feasibility

The Moving System is feasible technically, although there is some risk.
The risk regarding familiarity with the Moving System is middle.

- The users may have little experience with using web system to manage their business.
- The users may use Excel to save and manage order information only.
- The drivers may have no experience with using mobile application to finish their delivery jobs.

The project size is considered medium risk.

- The project team consists of 1 person.
- Web and mobile application are developed.

The compatibility with users' existing technical infrastructure should be good.

- The internet infrastructure is not highly required and the server of different platforms can be compatible.
- Desktop computer and mobile device are required to run the application.

Economic Feasibility

A cost-benefits analysis was performed. Conservative estimates show that the Moving System has a good chance of increasing the business income and reducing the order complaints.
ROI over 3 years: 42.0%
NPV over 3 years: $2372
Break-even occurs after 3.38 years
Intangible Costs and Benefits

- 24 hours customer ordering service
- Easy to supervise the business flow and their drivers.
- Easy to check the incoming delivery and the order records

Organizational Feasibility

From an organization perspective, this project has low risk. The business flow is not digital enough and it is difficult for the customers to chase their order records. Therefore, the Moving System can help to digitalize the ordering flow and manage the delivery orders for the drivers. Also, the system can help admin staff prevent from missing any moving order.

Additional Comments

- We need to train the admin staff and drivers to use the web and mobile application respectively.

Chapter 3

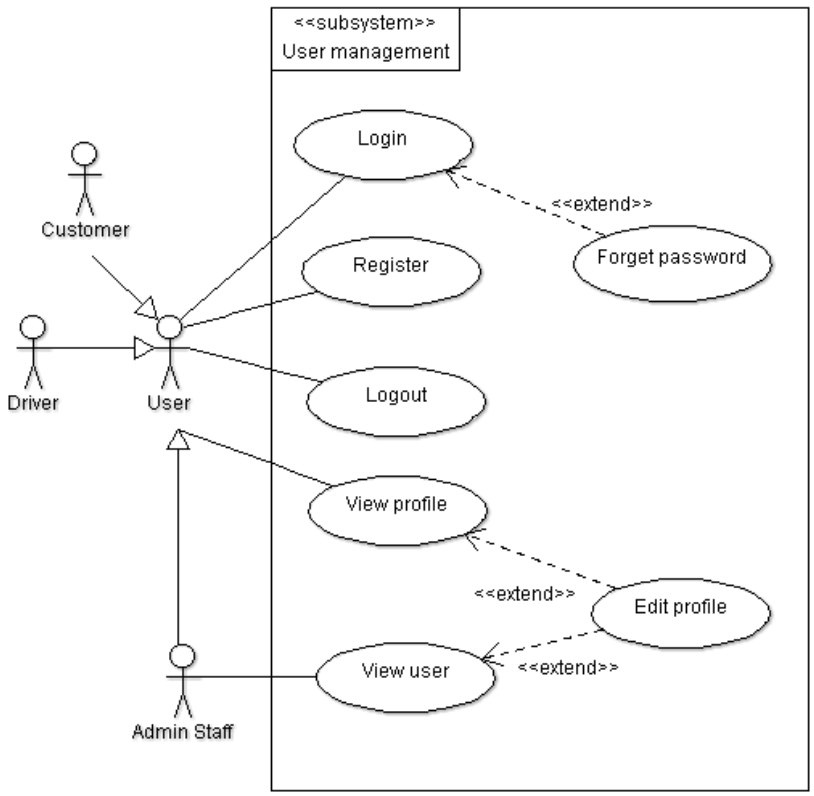
System Analysis

CONTENTS OF CHAPTER 3

3.1 Use Case Model

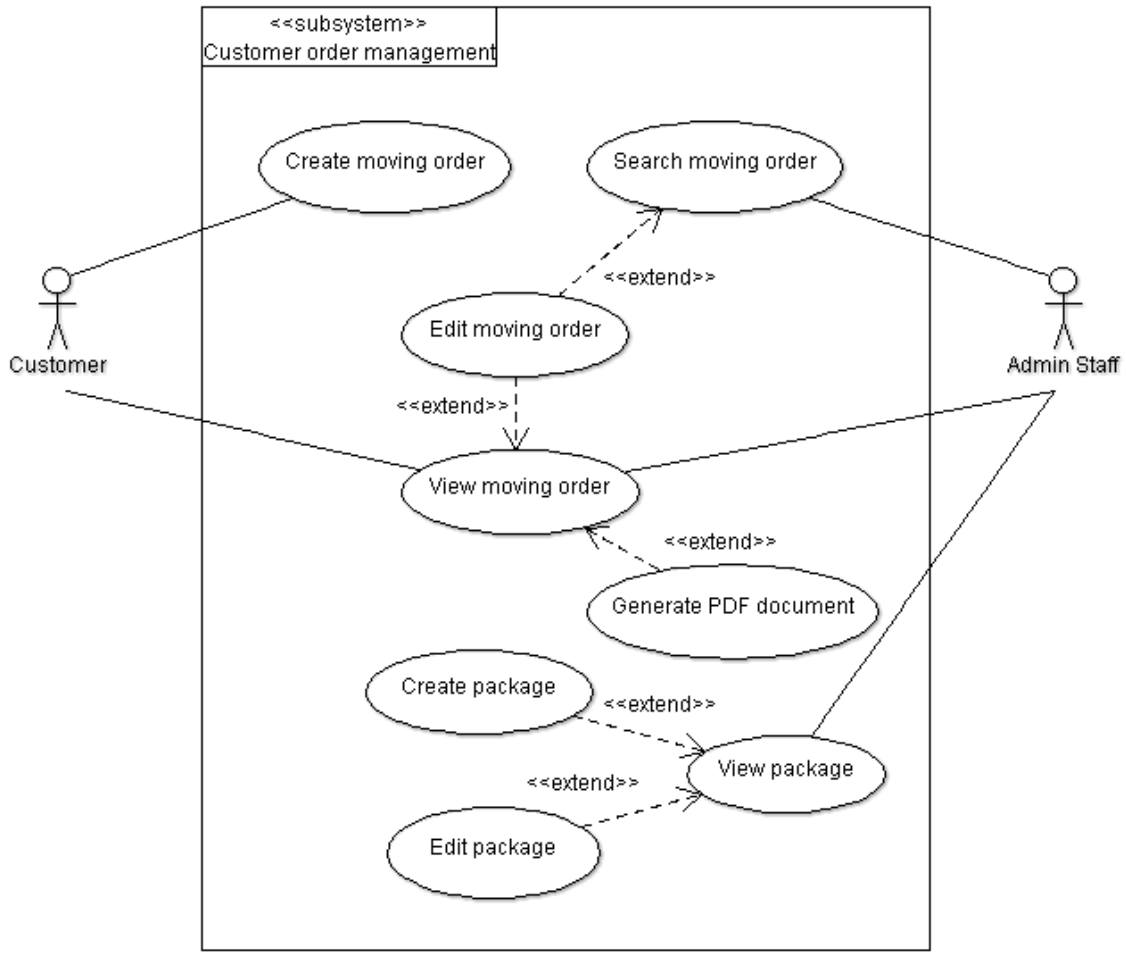
Use Cases Diagram

1. User management



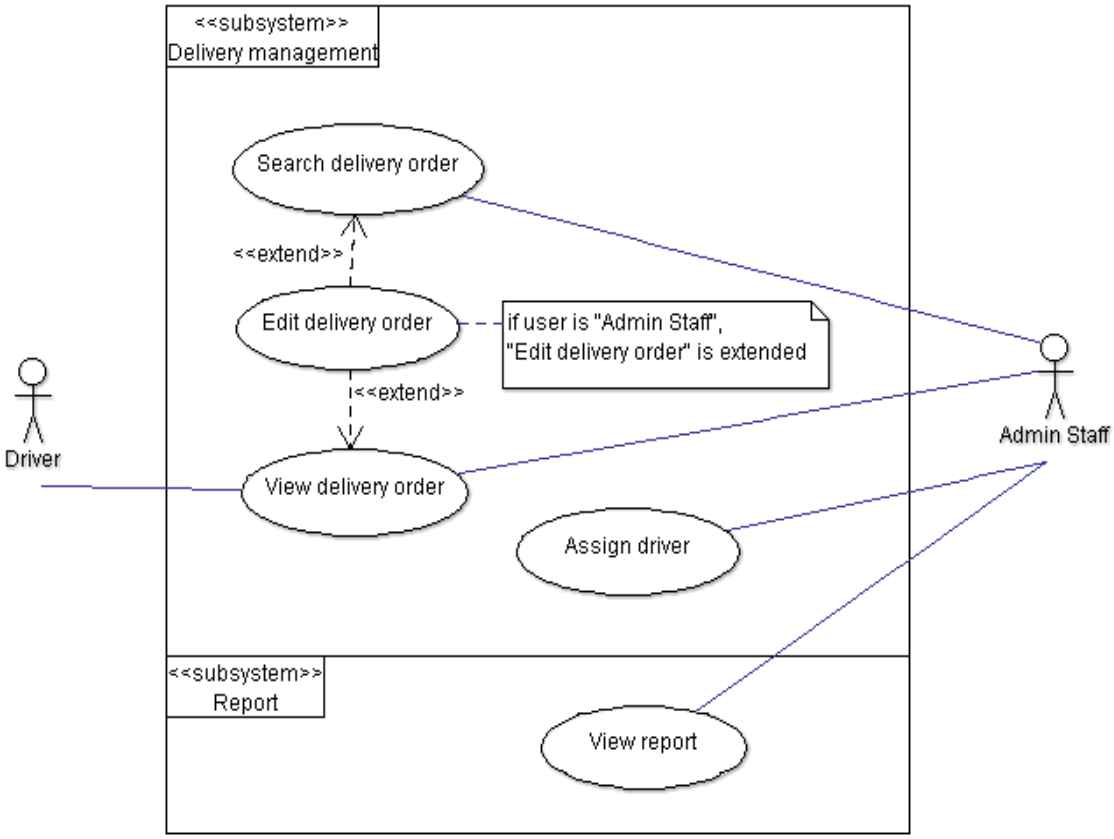
For user management module, customer, driver and admin staff inherit the user functions. Admin staff has more use cases to manage other two types of user.

2. Customer order management



For customer order management module, customer has use cases to create and view their order. In addition to that, the customer can print different PDF documents including quotation, invoice and receipt in different order status. Admin staff also has use case to manage all of the orders. In addition, admin staff can view all the packages, create package and edit package.

3. Delivery management & Report



For delivery management module, admin staff can schedule the delivery order and driver can view the delivery list through the mobile application. Other the other hand, for report module, admin staff including manager can view different reports for supervising the company.

Actor Description

There are four actors in Moving System including:

1. **User**

   User is inherited by the other three actors and has the basic functions of the system. As a result, user may specifically:

   1.1. login the system
   1.2. register a new user account
   1.3. logout the system
   1.4. get back the password if user forgets the login password
   1.5. view the personal information
   1.6. edit the personal information

2. **Admin Staff**

   Admin staff is the staff of the company office and manages the flow of service through the system. As a result, admin staff may specifically:

   2.1. view user
   2.2. edit user profile
   2.3. search moving order
   2.4. view moving order
   2.5. edit moving order
   2.6. assign driver
   2.7. search delivery order
   2.8. view delivery order
   2.9. edit delivery order
   2.10. view package
   2.11. create package
   2.12. edit package
   2.13. view report

3. Driver

Driver is the staff of the delivery team and responsible to complete the delivery. As a result, driver may specifically:

3.1. view delivery
3.2. accept delivery
3.3. complete delivery

4. Customer

Customer can surf the company website and make an order there. As a result, customer may specifically:

4.1. create moving order
4.2. view moving order
4.3. edit moving order
4.4. generate the PDF document of moving order

Use Cases Narrative

According to the use case diagram, there are twenty-four use cases, which are described in the following.

Use Case:	**Login**
Actor:	User
Actor Description:	1.1
Purpose:	Protect system data and identify user
Overview:	User information and all of the transactions data are stored in the system. Therefore, the system needs to identify users and different users have right to read or modify different kind of information.
Type:	Essential
Preconditions:	User must be registered before login.
Postconditions:	System functions are enabled.
Special Requirements:	No

Flow of Events	
ACTOR ACTION	**SYSTEM RESPONSE**
1. This use case begins when the user wants to login the Moving System and click “login” button.	2. Pop up a login box.
3. The user inputs username and password, and then click “sign in” button.	4. System verifies the input information and reloads the webpage. The login message is shown on top of it.
5. If successful, specific system functions are enabled. Otherwise, the user needs to input login information again.	

Alternative Flow of Events	
Step 4:	If the user is customer or driver, system reloads the webpage normally. If the user is admin staff, system redirects to another webpage and the admin functions are enabled.

Use Case:	**Register**
Actor:	User
Actor Description:	1.2
Purpose:	Record the user information
Overview:	The user inputs information including full name, username, contact information and password and all of them is stored in the system database. Therefore, system can verifies the user information when the user logins the system.
Type:	Essential
Preconditions:	The user does not have user account to login the system.
Postconditions:	The user can login the system and use its functions.
Special Requirements:	No

Flow of Events	
ACTOR ACTION	**SYSTEM RESPONSE**
1. This use case begins when the user wants to register in the Moving System and click “login” button.	2. Pop up a login box.
3. The user clicks “sign up” button.	4. Pop up a sign up form.
5. The user inputs first name, last name, username, email address, phone number and password, and then clicks “sign up” button.	6. System stores the user information in the database and reloads the webpage. The registration message is shown on top of it.
7. If successful, the user can click “login” button and input login information to login the system. Otherwise, the user needs to register again.	

Alternative Flow of Events

Use Case:	**Logout**
Actor:	User
Actor Description:	1.3
Purpose:	Leave system
Overview:	The user leaves the system and most of the system functions are disabled.
Type:	Essential
Preconditions:	The user already login the system.
Postconditions:	No
Special Requirements:	No

Flow of Events	
ACTOR ACTION	**SYSTEM RESPONSE**
1. This use case begins when the user wants to leave the system. The user clicks the user name on right top of the webpage and click “logout” button.	2. Leave the system and reload the webpage.
3. Another user can login the system.	

Alternative Flow of Events

Use Case:	**Forget password**
Actor:	User
Actor Description:	1.4
Purpose:	Get back the user password
Overview:	System verifies the user by their user name or email address and sets the new user password. The new password is sent to the user mail box.
Type:	Essential
Preconditions:	The user already registered in the system.
Postconditions:	No
Special Requirements:	No

Flow of Events	
ACTOR ACTION	**SYSTEM RESPONSE**
1. This use case begins when the user forget the password. The user clicks “login” button, then clicks “lost password” button.	2. Pop up “lost password” box.
3. The user inputs user name or email address, then clicks “reset password” button.	4. System verifies whether the user exists, then sets the new user password and reloads the webpage. The reset message is shown on top of it.
5. If successful, the new password will be sent to the user mail box and the user can login the system by the new password. Otherwise, the user needs to input the user name or email address again.	

Alternative Flow of Events

Use Case:	**View profile**
Actor:	User
Actor Description:	1.5
Purpose:	View the user information
Overview:	The user can check their personal information and confirm it is correct.
Type:	Essential
Preconditions:	The user already registered and login the system.
Postconditions:	No
Special Requirements:	No

Flow of Events	
ACTOR ACTION	**SYSTEM RESPONSE**
1. This use case begins when the user click the user name on the top menu bar after login the system.	2. Pop up function box.
3. Click “profile” button.	4. System retrieves the user information and illustrates on the “profile” box.
5. The user can view their personal information including user name, first name, last name, phone number and email address.	

Alternative Flow of Events

Use Case:	**Edit profile**
Actor:	User
Actor Description:	1.6, 2.2
Purpose:	Change the user information
Overview:	The user can change their personal information when they need.
Type:	Essential
Preconditions:	The user is registered and login the system
Postconditions:	No
Special Requirements:	No

Flow of Events	
ACTOR ACTION	**SYSTEM RESPONSE**
1. This use case begins when the user clicks their user name and then clicks “profile” after login the system.	2. Pop up the user profile box.
3. The user changes the personal information and clicks “save change”.	4. System saves all of the changes and reloads the webpage, and the message will be shown on top of it.
5. If successful, all of the changes the user input will be applied. Otherwise, no change is applied. The user needs to edit their profile again.	

Alternative Flow of Events

Use Case:	**View user**
Actor:	User
Actor Description:	2.1
Purpose:	View the user
Overview:	The use can view the user and change their information.
Type:	Essential
Preconditions:	The user is login the system as admin staff
Postconditions:	No
Special Requirements:	No

Flow of Events	
ACTOR ACTION	**SYSTEM RESPONSE**
1. This use case begins when the user login as admin staff and click “User List” button.	2. Retrieve all of the user information and list on the table.
3. The user can click the user name to view and edit their information.	

Alternative Flow of Events

**Use case 2.2 is already descripted in use case 1.6*

Use Case:	**Search moving order**
Actor:	Admin staff
Actor Description:	2.3
Purpose:	Search the specific moving orders of the customer
Overview:	The admin staff can find out the specific moving order by the customer name.
Type:	Essential
Preconditions:	The user is login the system as admin staff.
Postconditions:	No
Special Requirements:	No

Flow of Events	
ACTOR ACTION	**SYSTEM RESPONSE**
1. This use case begins when the user logins as admin staff and click “Search Order” button in the left menu bar.	2. System redirects another page and there is text field for searching order.
3. The admin staff can input the customer name and click “Search” button.	4. System retrieves all of the orders which are created by specific customer and lists on the table.
5. The admin staff can select the specific order to view more order information and edit it.	

Alternative Flow of Events

Use Case:	**View moving order**
Actor:	Customer
Actor Description:	2.4, 4.2
Purpose:	View the moving orders of the customer
Overview:	The customer can view their orders and edit its information.
Type:	Essential
Preconditions:	The user is login the system as customer
Postconditions:	No
Special Requirements:	No

Flow of Events	
ACTOR ACTION	**SYSTEM RESPONSE**
1. This use case begins when the user click the user name on the top menu bar after login the system.	2. Pop up the user profile box.
3. The customer clicks “Order Records” button.	4. Redirect to another page and the system retrieves the order information of the customer and lists on the table.
5. The customer can click specific order to view more information.	

Alternative Flow of Events

Use Case:	**Edit moving order**
Actor:	Customer
Actor Description:	2.5, 4.3
Purpose:	Edit the moving orders of the customer
Overview:	The customer can update their orders before the order is confirmed.
Type:	Essential
Preconditions:	The user is login the system as customer.
Postconditions:	No
Special Requirements:	The customer made the moving order previously.

Flow of Events	
ACTOR ACTION	**SYSTEM RESPONSE**
1. This use case begins when the user click the user name on the top menu bar after login the system.	2. Pop up the user profile box.
3. The customer clicks “Order Records” button.	4. Redirect to another page and the system retrieves the order information of the customer and lists on the table.
5. The customer can click specific order to edit it and click “Save Change” button.	6. System updates the existing order information and redirects the home page. The message will be shown on the top of the website.
7. If successful, the order is updated. Otherwise, the customer needs to update the order again.	

Alternative Flow of Events

Use Case:	**Assign driver**
Actor:	Admin staff
Actor Description:	2.6
Purpose:	Assign delivery order to the driver
Overview:	The admin staff can assign the delivery order to their drivers.
Type:	Essential
Preconditions:	The user is login the system as admin staff.
Postconditions:	No
Special Requirements:	No

Flow of Events	
ACTOR ACTION	**SYSTEM RESPONSE**
1. This use case begins when the user logins as admin staff and click “Schedule Delivery” button.	2. System retrieves delivery orders which do not have drivers and lists on the table.
3. The admin staff can assign the driver and type of car to the available delivery order and click “Assign” button.	4. System updates the delivery order and reloads the page.
5. If successful, the delivery order is updated. Otherwise, the delivery order requires to be assigned again.	

Alternative Flow of Events

Use Case:	**Search delivery order**
Actor:	Admin staff
Actor Description:	2.7
Purpose:	Search delivery order
Overview:	The admin staff can search the existing delivery order.
Type:	Essential
Preconditions:	The user is login the system as admin staff.
Postconditions:	No
Special Requirements:	No

Flow of Events	
ACTOR ACTION	**SYSTEM RESPONSE**
1. This use case begins when the user logins as admin staff and click “Search Delivery” button.	2. System redirects to another page and there is a text field for inserting the delivery date or driver name.
3. The admin staff can insert the delivery date or driver name and click “Search” button.	4. System retrieves the relevant delivery order and lists on the table.
5. The admin staff can select specific delivery order to view order detail or edit it.	

Alternative Flow of Events

Use Case:	**View delivery order**
Actor:	Admin staff
Actor Description:	2.8
Purpose:	View all of the delivery orders
Overview:	The admin staff can view all of the delivery orders.
Type:	Essential
Preconditions:	The user is login the system as admin staff.
Postconditions:	No
Special Requirements:	No

Flow of Events	
ACTOR ACTION	**SYSTEM RESPONSE**
1. This use case begins when the user logins as admin staff and click “View Delivery” button.	2. System reloads the page, retrieves all of the delivery orders and lists on the table.
3. The admin staff can select specific delivery order to view the order details or edit it.	

Alternative Flow of Events

Use Case:	**Edit delivery order**
Actor:	Admin staff
Actor Description:	2.9
Purpose:	Edit delivery order
Overview:	The admin staff can edit the existing delivery order.
Type:	Essential
Preconditions:	The user is login the system as admin staff.
Postconditions:	No
Special Requirements:	No

Flow of Events	
ACTOR ACTION	**SYSTEM RESPONSE**
1. This use case begins when the user logins as admin staff and click “Search Delivery” or “View Delivery” button.	2. System retrieves specific delivery order details after the admin staff selects the delivery order.
3. The admin staff can update the delivery order form and click “Save Change” button.	4. System updates the delivery order details and reloads the page.
5. If successful, the delivery order is updated. Otherwise, the delivery order requires to be updated again.	

Alternative Flow of Events

Use Case:	**View package**
Actor:	Admin staff
Actor Description:	2.10
Purpose:	View all of the price packages
Overview:	The admin staff can review the existing order packages.
Type:	Essential
Preconditions:	The user is login the system as admin staff.
Postconditions:	No
Special Requirements:	No

Flow of Events	
ACTOR ACTION	**SYSTEM RESPONSE**
1. This use case begins when the user logins as admin staff and click “View Package” button.	2. System retrieves all of the packages and lists on the package table.
3. The admin staff can select specific package.	4. System retrieves specific package information and shows on the package form.
5. The admin staff can review the details of the package.	

Alternative Flow of Events

Use Case:	**Create package**
Actor:	Admin staff
Actor Description:	2.11
Purpose:	Create new package with different order price
Overview:	The admin staff can create new package with different order price for different types of customers.
Type:	Essential
Preconditions:	The user is login the system as admin staff.
Postconditions:	No
Special Requirements:	No

Flow of Events	
ACTOR ACTION	**SYSTEM RESPONSE**
1. This use case begins when the user logins as admin staff and click “View Package” button.	2. System retrieves all of the packages and lists on the package table.
3. The admin staff can click “here” button for creating new package.	4. System redirects to another page and loads a new package form.
5. The admin staff can fill in the new package form and then click “Create Package”.	6. System saves the new package information and refreshes package table.
7. If successful, the package is created. Otherwise, the package needs to be created again.	

Alternative Flow of Events

Use Case:	**Edit package**
Actor:	Admin staff
Actor Description:	2.12
Purpose:	Edit the existing package
Overview:	The admin staff can edit the existing package.
Type:	Essential
Preconditions:	The user is login the system as admin staff.
Postconditions:	No
Special Requirements:	No

Flow of Events	
ACTOR ACTION	**SYSTEM RESPONSE**
1. This use case begins when the user logins as admin staff and click “View Package” button.	2. System retrieves all of the existing packages and lists on the package table.
3. The admin staff can select specific package.	4. System retrieves specific package information and shows on the package form.
5. The admin staff can review and edit the package information and then click “Update Package” button.	6. System saves any change of the package details and refreshes the package table.
7. If successful, the package is updated. Otherwise, the package needs to be updated again.	

Alternative Flow of Events

Use Case:	**View report**
Actor:	Admin staff
Actor Description:	2.13
Purpose:	View reports
Overview:	The admin staff can view different kind of reports.
Type:	Essential
Preconditions:	The user is login the system as admin staff.
Postconditions:	No
Special Requirements:	No

Flow of Events	
ACTOR ACTION	**SYSTEM RESPONSE**
1. This use case begins when the user logins as admin staff and select the type of report they needed.	2. System retrieves the data the report needs and generates the report.
3. The admin staff can view the report.	

Alternative Flow of Events

Use Case:	**View delivery**
Actor:	Driver
Actor Description:	3.1
Purpose:	View the delivery order
Overview:	The driver can view their orders and information.
Type:	Essential
Preconditions:	The user is login the system as driver
Postconditions:	No
Special Requirements:	The delivery order is confirmed by both admin staff and customer.

Flow of Events	
ACTOR ACTION	**SYSTEM RESPONSE**
1. This use case begins when the user click the delivery list or waiting list on the bottom menu bar after login the system.	2. Reload the delivery list.
3. The driver can select delivery order.	4. Redirect to another page and the system retrieves the order information and shows the delivery path on the map.
5. The driver can view the delivery details.	

Alternative Flow of Events

Use Case:	**Accept delivery**
Actor:	Driver
Actor Description:	3.2
Purpose:	Edit the delivery order status
Overview:	The driver can accept the order if the order is not assigned driver.
Type:	Essential
Preconditions:	The user is login the system as driver.
Postconditions:	No
Special Requirements:	No

Flow of Events	
ACTOR ACTION	**SYSTEM RESPONSE**
1. This use case begins when the user click the delivery list or waiting list on the bottom menu bar after login the system.	2. Reload the delivery list.
3. The driver can select delivery order.	4. Redirect to another page and the system retrieves the order information and shows the delivery path on the map.
5. The driver can tap the “Accept” button.	6. Edit the driver ID of the order and pop up a message.
7. The driver reads the message and tap “OK” button.	8. Reload the delivery list.

Alternative Flow of Events

Use Case:	**Complete delivery**
Actor:	Driver
Actor Description:	3.3
Purpose:	Edit the delivery order status
Overview:	The driver can finish the order after finishing the delivery.
Type:	Essential
Preconditions:	The user is login the system as driver.
Postconditions:	No
Special Requirements:	No

Flow of Events	
ACTOR ACTION	**SYSTEM RESPONSE**
9. This use case begins when the user click the delivery list or waiting list on the bottom menu bar after login the system.	10. Reload the delivery list.
11. The driver can select delivery order.	12. Redirect to another page and the system retrieves the order information and shows the delivery path on the map.
13. The driver can tap the “Finish” button.	14. Change the order status to be “finish” and pop up a message.
15. The driver reads the message and tap “OK” button.	16. Reload the delivery list.

Alternative Flow of Events

Use Case:	**Create moving order**
Actor:	Customer
Actor Description:	4.1
Purpose:	Make a moving order
Overview:	The customer can make an order through the website.
Type:	Essential
Preconditions:	The user is registered and login the system as customer
Postconditions:	No
Special Requirements:	No

Flow of Events	
ACTOR ACTION	**SYSTEM RESPONSE**
1. The use case begins when user logins as customer, clicks “Price” on the top menu and select the type of the order the user needs.	2. Redirect to another page for creating new order.
3. The customer inserts the order information and click “Save Order”.	4. System saves the order information and redirect to home page. The order message will be shown on the top of the website.
5. If successful, the order is created. Otherwise, the order needs to be created again.	

Alternative Flow of Events

**Use case 4.2 is already descripted in use case 2.4*

**Use case 4.3 is already descripted in use case 2.5*

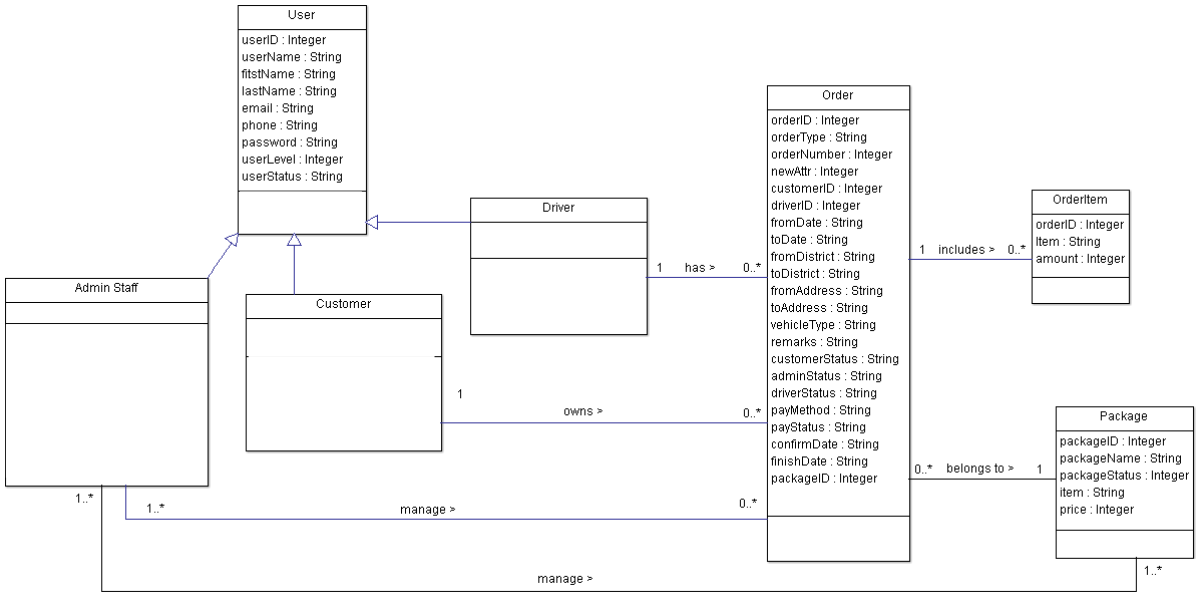
Use Case:	**Generate PDF document**
Actor:	Customer
Actor Description:	4.4
Purpose:	Generate the PDF document of the moving order
Overview:	The customer can generate the quotation, invoice and receipt in different order status.
Type:	Essential
Preconditions:	The user is login the system as customer and the order is created before.
Postconditions:	No
Special Requirements:	No

Flow of Events	
ACTOR ACTION	**SYSTEM RESPONSE**
1. This use case begins when the user click the user name on the top menu bar after login the system.	2. Pop up the user profile box.
3. The customer clicks “Order Records” button.	4. Redirect to another page and the system retrieves the customer orders.
5. The customer can select specific order to view more information and click “Generate Quotation” button.	6. System generates the order document and exports PDF file.
7. The customer can download the PDF file.	

Alternative Flow of Events	
Step 5:	If the order is delivered but not paid, “Generate Invoice” button is enabled. If the is delivered and paid, “Generate Receipt” button is enabled.

Alternative Flow of Events

3.2 Class Diagram (with attributes only)



Description

Mainly, there are four classes in the class diagram. However, the User class is super class in this situation, and Admin Staff, Customer and Driver class, sub class, inherit this super class. They are the different types of user in the system and all have the same attributes.

For handling the order, there are three classes including Order, OrderItem and Package, and they are responsible for any order operation in the system.

In the following, all the activity diagrams are categorized by class:

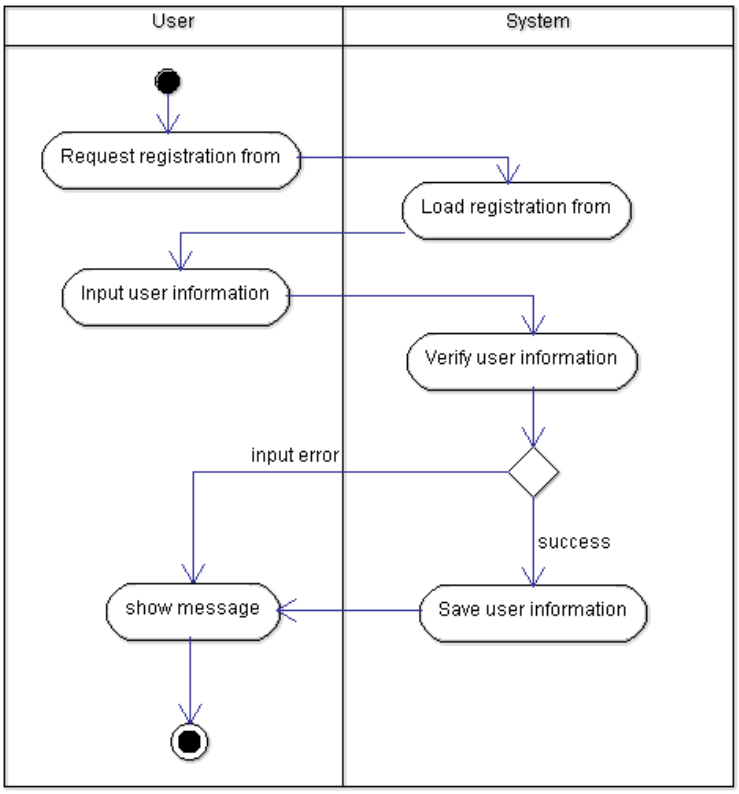
1. User
2. Admin staff
3. Customer
4. Driver

3.3 Activity Diagram

For the Moving System, there are 14 main activities and these activity diagrams are in the following:

1. Signing up activity

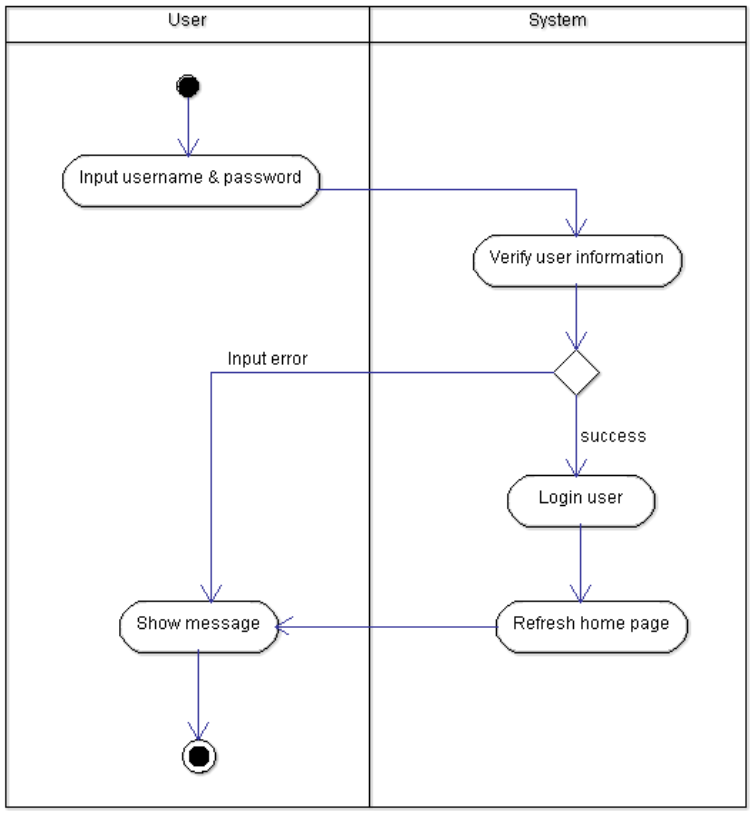
   Class: User



For the web application in the Moving System, the user needs to register first. For the customers, they can simply click “SIGN UP” button for requesting the registration form. For the admin staffs, they can click “User → New User” for requesting that form in admin page. Then the user can fill in the form and create new user account.

2. Signing in activity

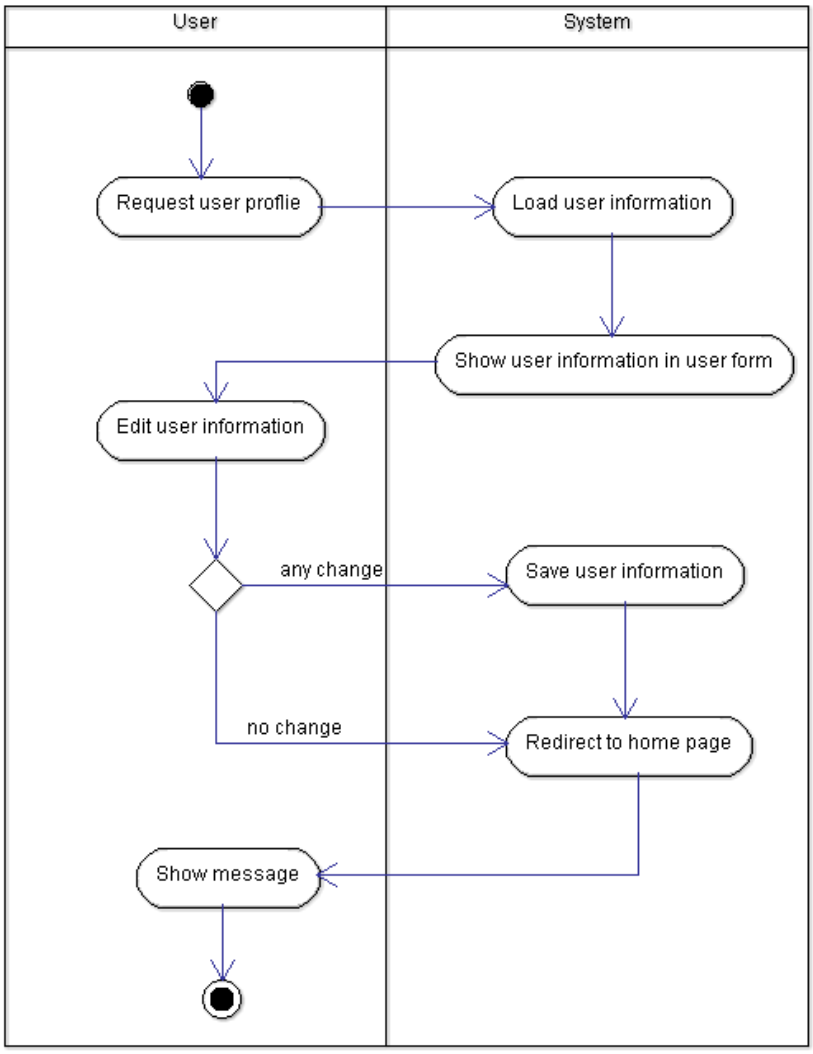
   Class: User



The visitor can input the user name and password in the sign in box. Then, the system verifies the user information. If successful, the visitor can login as the customer or admin staff based on their user level. Otherwise, the visitor needs to sign in again and the message is shown on the top of the website.

3. Viewing and editing user profile activity

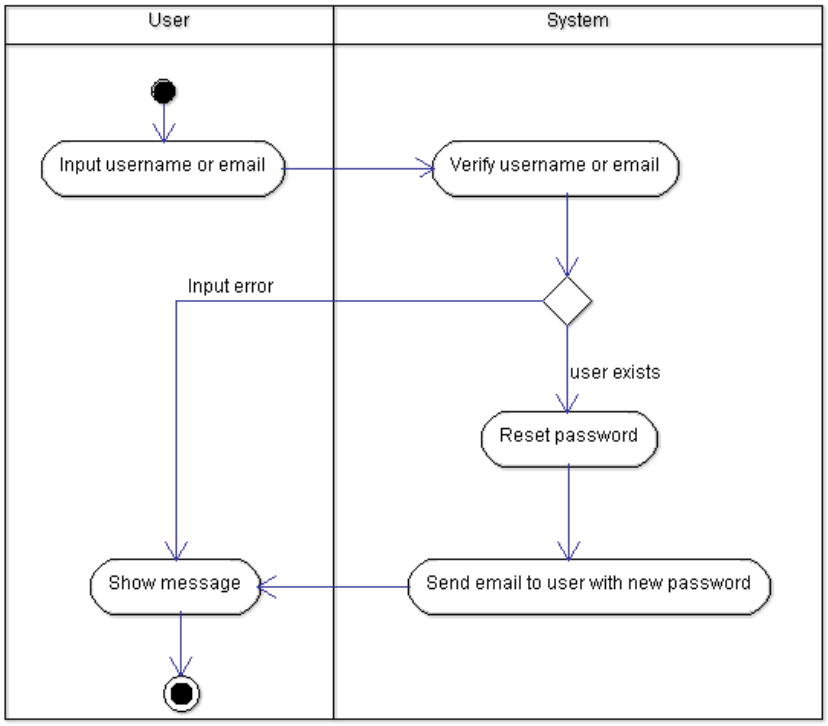
   Class: User



After the signing in process, the user can click “PROFILE” button for requesting the user information and it is illustrated in the form. The user can edit the user information if they need. If there is any change, the system will save the updated user information and the message is popped up.

4. Losing password activity

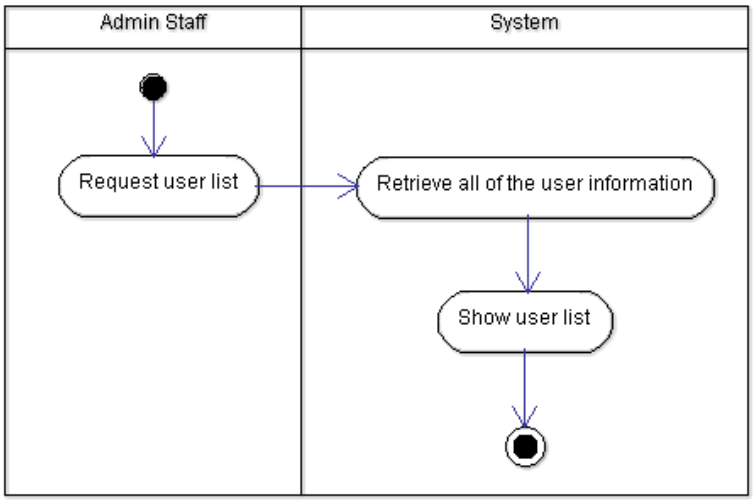
   Class: User



If the user loses their password, they can click “LOST PASSWORD?” button and “LOST PASSWORD” form will be popped up. Then, the user can input their user name or email address. If the user is existed, the system will reset the password and send the new password to the user email box. Finally, the message is shown on the top of the website.

5. Viewing user activity

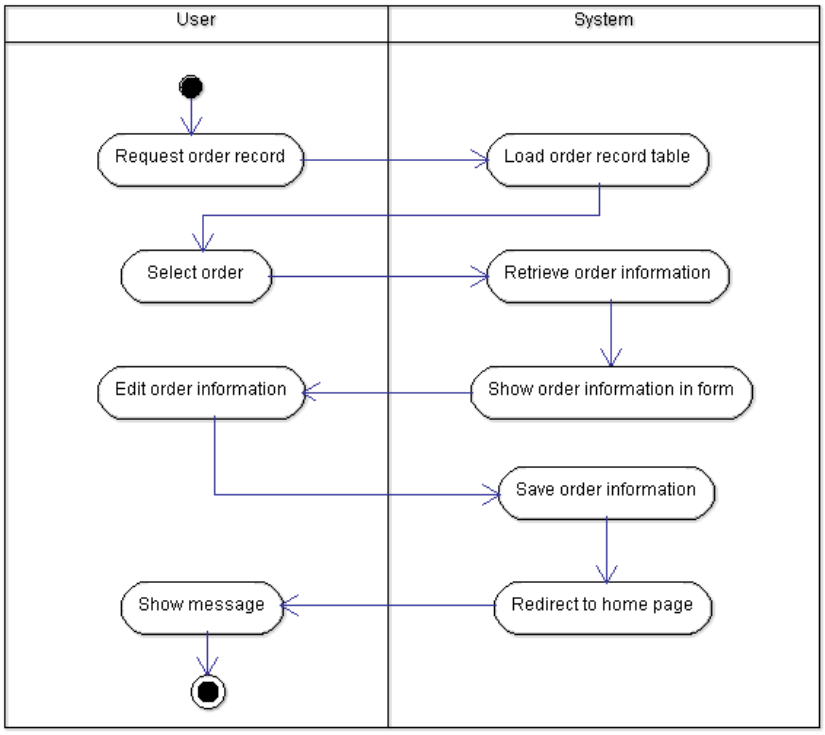
   Class: Admin staff



This activity is processed by the admin staff only and they can click “User → View User” button. The system will retrieve all of the general information of the user and illustrate on the user list.

6. Viewing and editing order activity

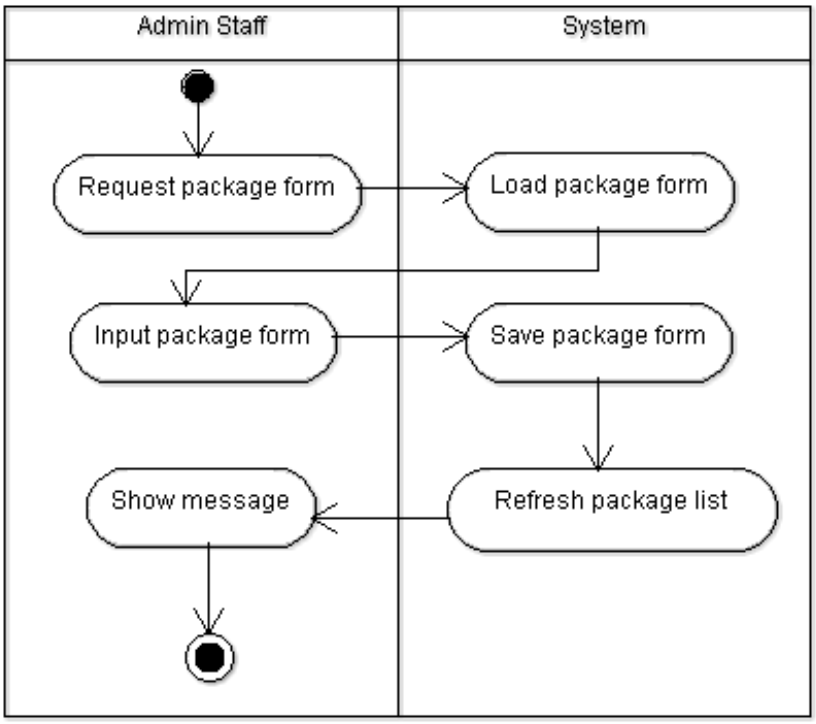
   Class: Admin staff, Customer



When the customer wants to check their order records, they can click “ORDER RECORDS” button. Then, the system can retrieve all of the orders of customer and show on the order list. The customer can select specific order for viewing the order details. If the order status is not confirmed, the customer can edit the order information any time. The system will save the updated order information and the message will be shown on the top of the website.

7. Create package activity

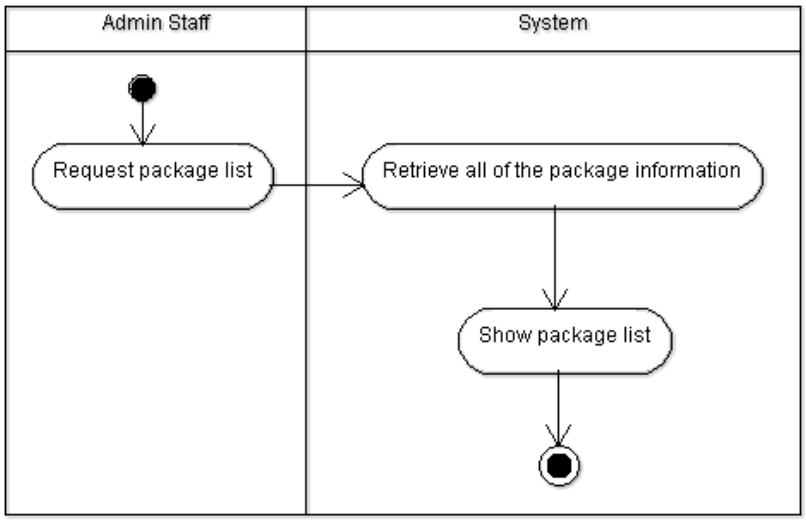
   Class: Admin staff



Some customers may have different price packages, so the admin staff can create package. Therefore, the item price of the order can be changed easily.

8. View package activity

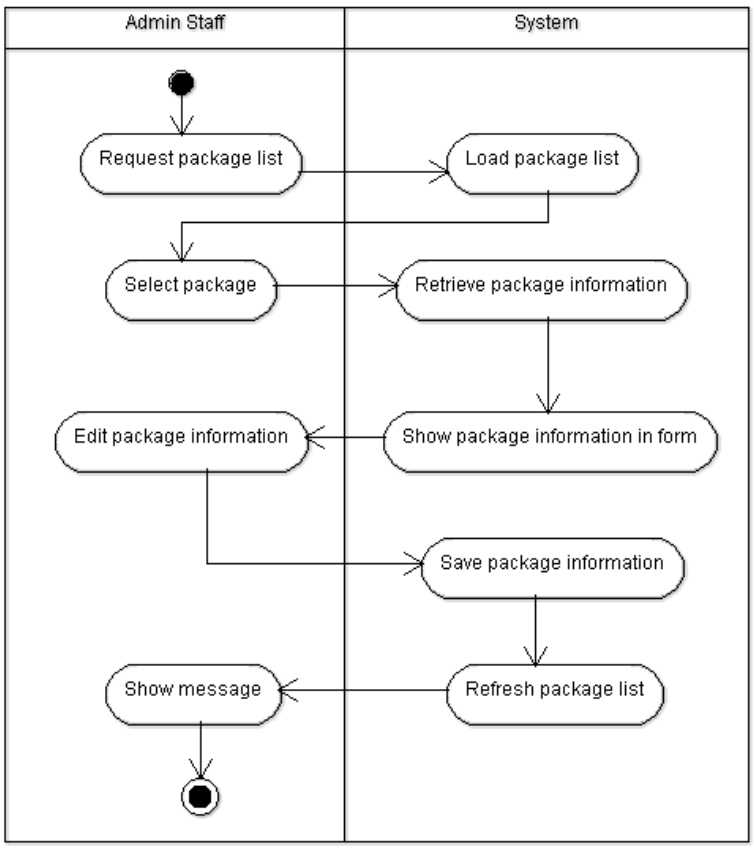
   Class: Admin staff



The admin staff can manage the existing packages and they are all listed on the package list in “View Package” page.

9. Edit package activity

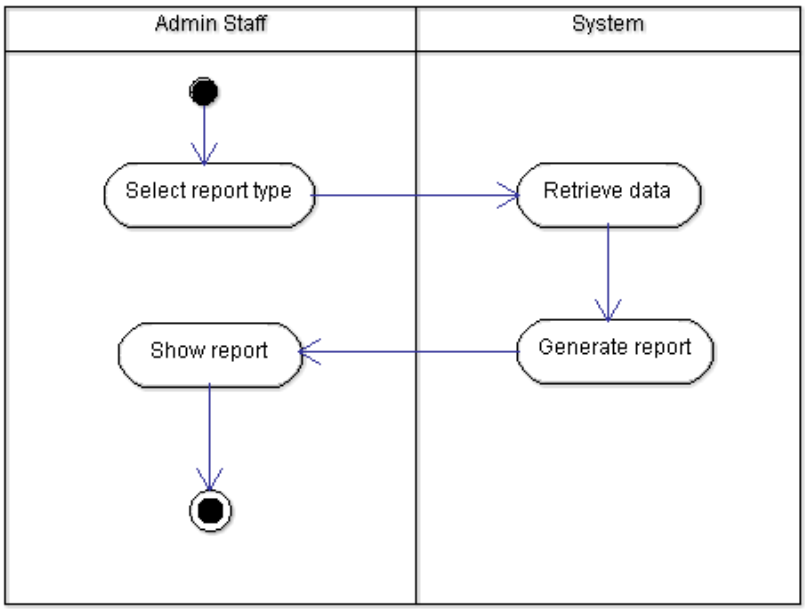
   Class: Admin staff



In “View Package” page, the admin staff can review the existing packages briefly. They can also select specific package and edit the package information in the package form.

10. Viewing report activity

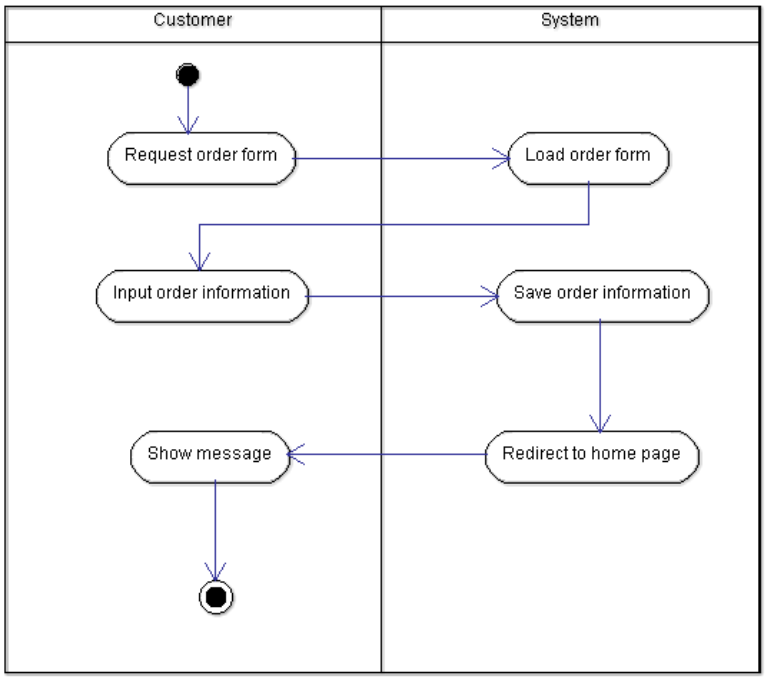
    Class: Admin staff



The admin staff can select different report types when they need to prepare the report for the manager. After selecting specific report type, the system will retrieve relevant data for generating the report. Finally, it is shown on the website.

11. Creating order activity

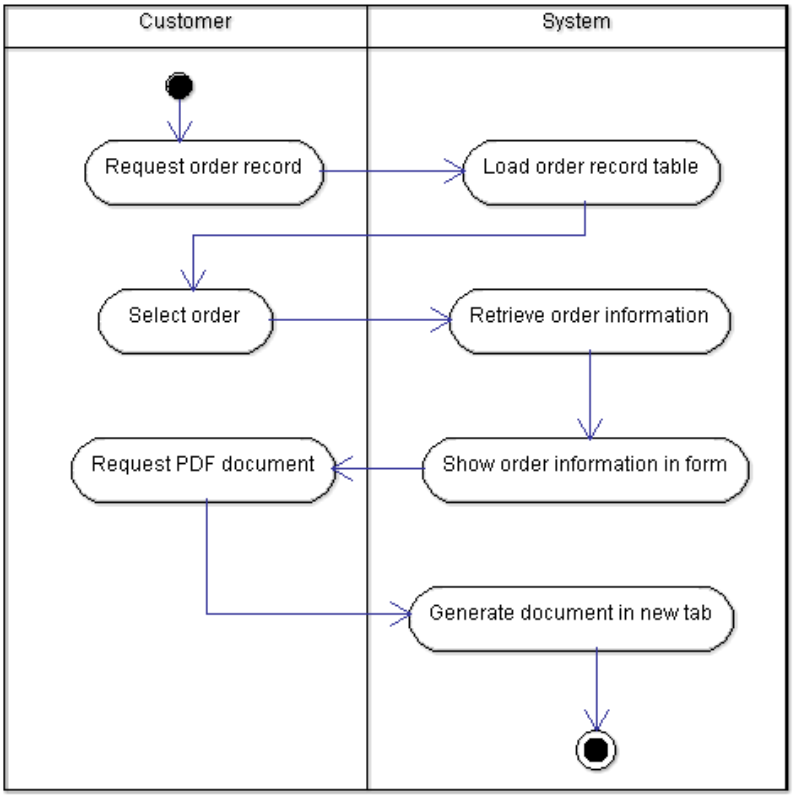
    Class: Customer



The customer can create the order by clicking the “Order” button. Then, the system pops up an order form. Therefore, the customer can fill in the form and save the order information. Finally, the message is shown on the top of the website.

12. Generating PDF document activity

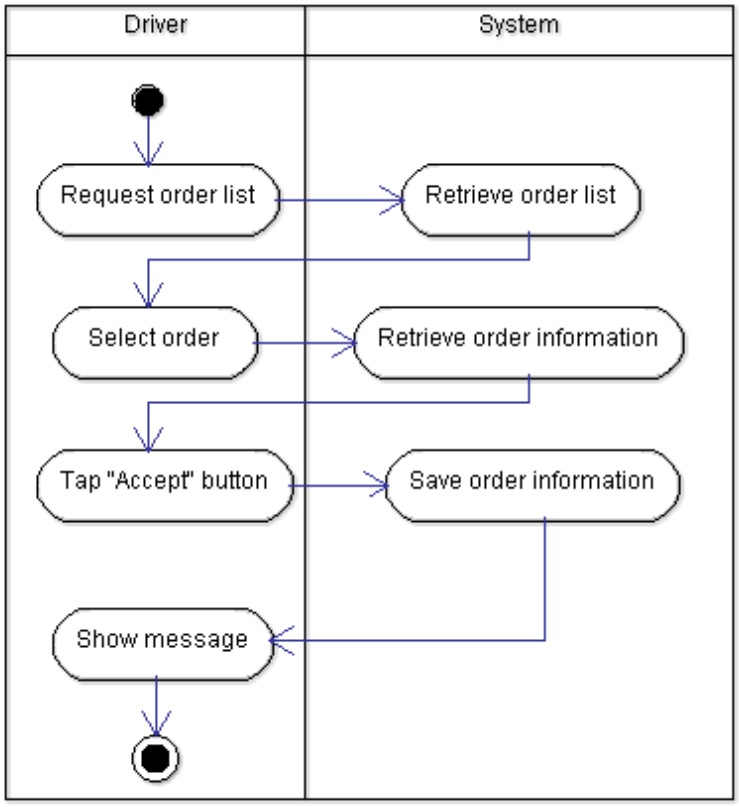
    Class: Customer



If the customer wants to generate the order receipt, they can click “ORDER RECORDS” button and select the specific order. Then, the order information is shown on the order form. The customer can click “Generate Quotation/Invoice/Receipt” button according to the order status. Finally, the PDF document is generated in the new website tab.

13. Accepting order activity

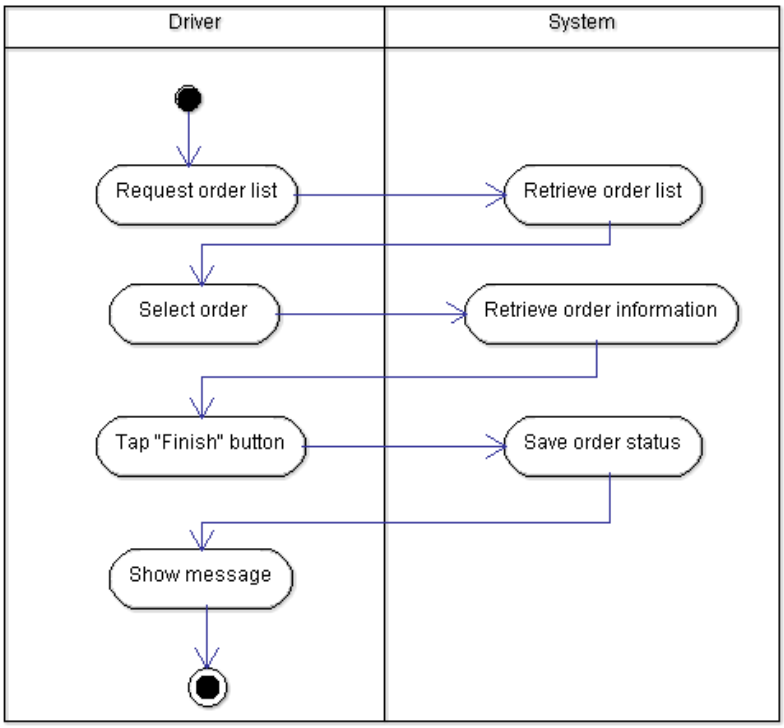
    Class: Driver



For the drivers, they can accept the orders in the waiting list of the mobile application. They can tap the “Waiting list” tab and select the orders, which are not assigned driver. After they read the order information, they can click “Accept” button for taking that order and the message will pop up.

14. Finishing order activity

    Class: Driver



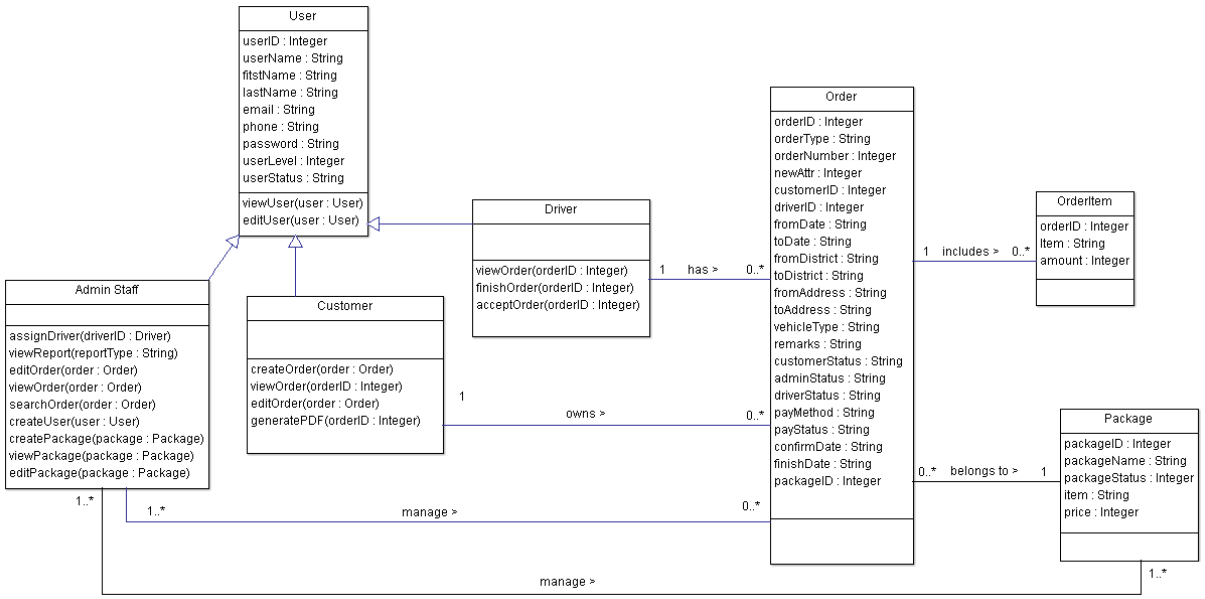
After the delivery, the driver can select the order in the delivery list of mobile application and check whether the order information is correct. Then, the driver can tap “Finish” button and the system will update the order status. Finally, the message is popped up.

Chapter 4

System Design

CONTENTS OF CHAPTER 4

4.1 Class Diagram (with attributes and methods)



Description

In the Moving System, there are various classes and different responsibilities are assigned to them. Based on the above diagram, there are two types of classes including user and order components.

In order to process different use cases, the following classes are required to be created.

1. Class User
2. Class Admin Staff
3. Class Customer
4. Class Driver
5. Class Order
6. Class OrderItem
7. Class Package

Class Responsibility

1. **Class User**

   Responsibility

   1.1 Store the general user variables

   1.2 View user profile

   1.3 Edit user profile

2. **Class Admin Staff**

   Responsibility

   2.1 Create different types of users

   2.2 View user

   2.3 View order details

   2.4 Edit order details

   2.5 Assign driver to specific order

   2.6 Search order by order number

   2.7 Create package

   2.8 View existing packages and their details

   2.9 Edit package details

   2.10 View different kinds of reports

3. **Class Customer**

   Responsibility

   3.1 Create order

   3.2 View orders the customer made before

   3.3 Edit orders if the orders are not confirmed

   3.4 Generate PDF document in different order status

4. **Class Driver**

   Responsibility

   4.1 View order information

   4.2 Accept order if the order is on the waiting list

   4.3 Finish order after the delivery is completed

5. **Class Order**

   Responsibility

   5.1 Store order information

6. **Class OrderItem**

   Responsibility

   6.1 Store order items which involve in the delivery

7. **Class Package**

   Responsibility

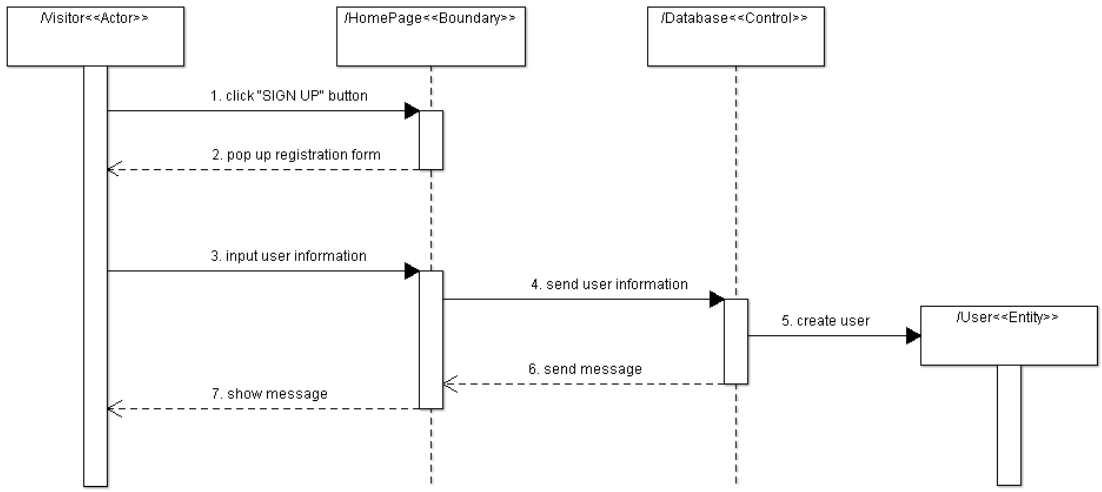
   7.1 Store package information

4.2 Sequence Diagram

For the Moving System, the following 14 sequence diagrams are descripting the general system processes.

1. Signing up

   Class Responsibility: 1.1

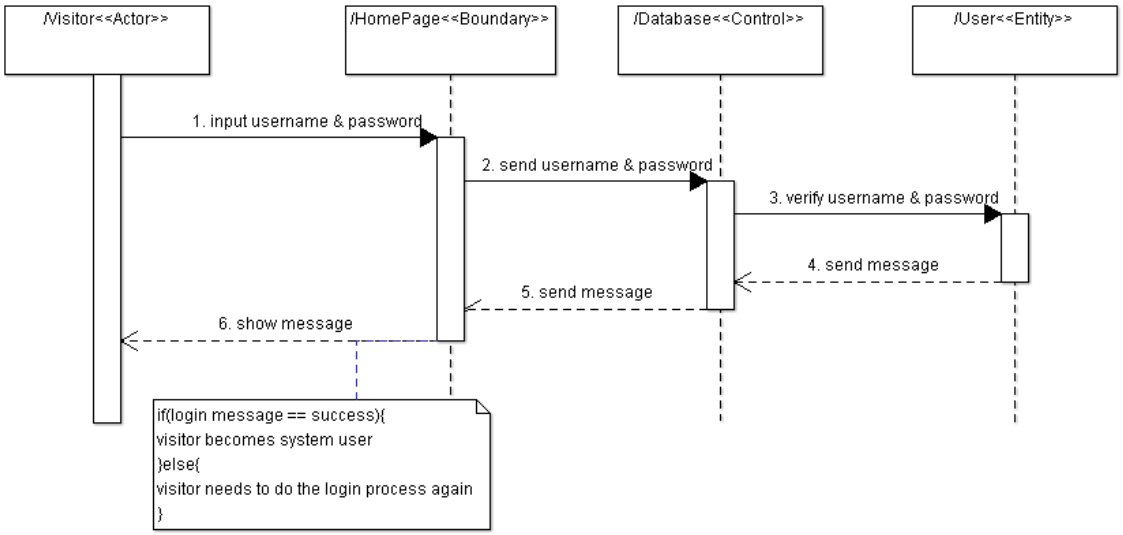


Flow of events

1. Visitor clicks “SIGN UP” button.
2. Home Page pops up a registration form.
3. Visitor inputs the user information and clicks “SIGN UP” button.
4. Home Page gets the user information and sends to Database.
5. Database creates new User.
6. Database sends a message to Home Page.
7. Home Page shows the message whether the user is created successfully.

2. Signing in

   Class responsibility: 1.1

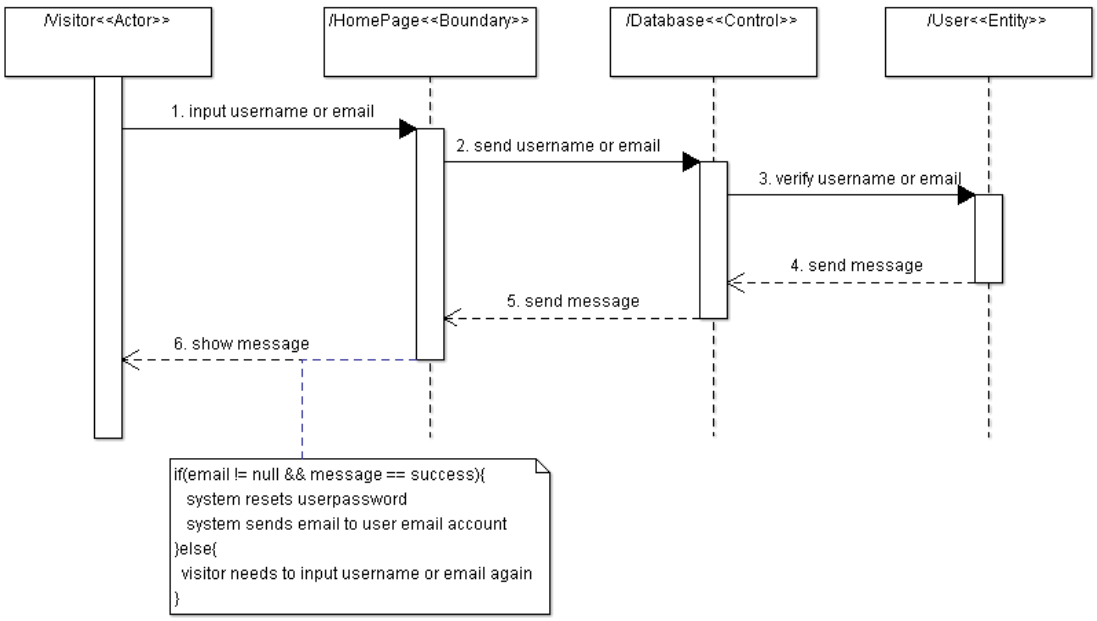


Flow of event

1. Visitor inputs the user name and password.
2. Home Page gets the user name and password and sends them to Database.
3. Database verifies the user name and password whether they are matched.
4. User entity returns a message.
5. Database sends the message to Home Page.
6. Home Page shows the message whether the visitor login successfully.

3. Losing password

   Class Responsibility: 1.1

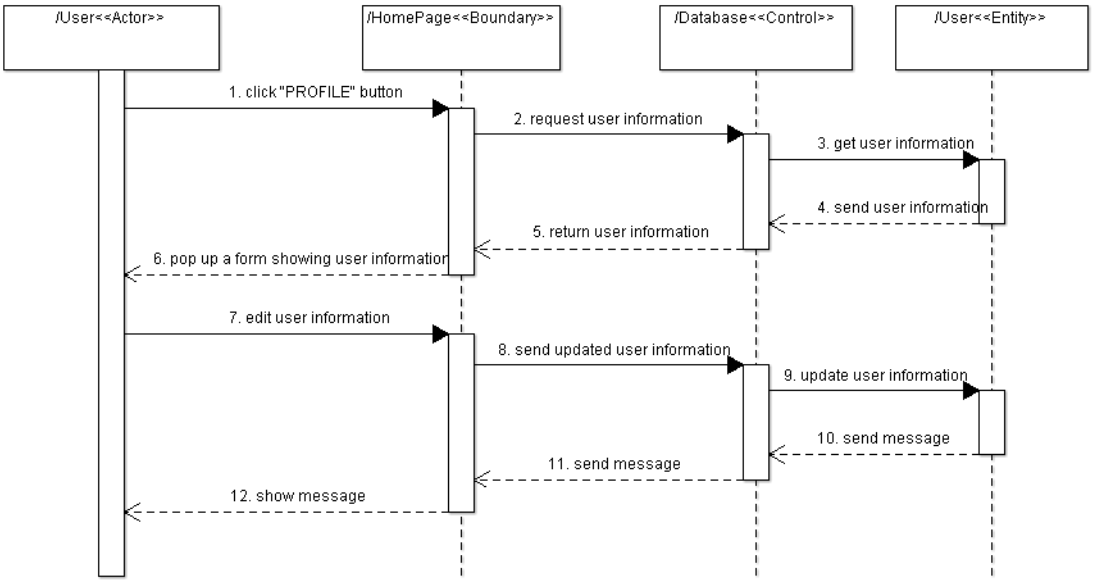


Flow of event

1. Visitor can input their user name or email address if they lose their password.
2. Home Page gets the user name or email address and sends to Database.
3. Database verifies the user name or email address.
4. User entity returns a message.
5. Database sends the message to Home Page.
6. Home Page shows the message whether the password is reset and sent to the user email account successfully.

4. Viewing and editing user profile

   Class responsibility: 1.2, 1.3

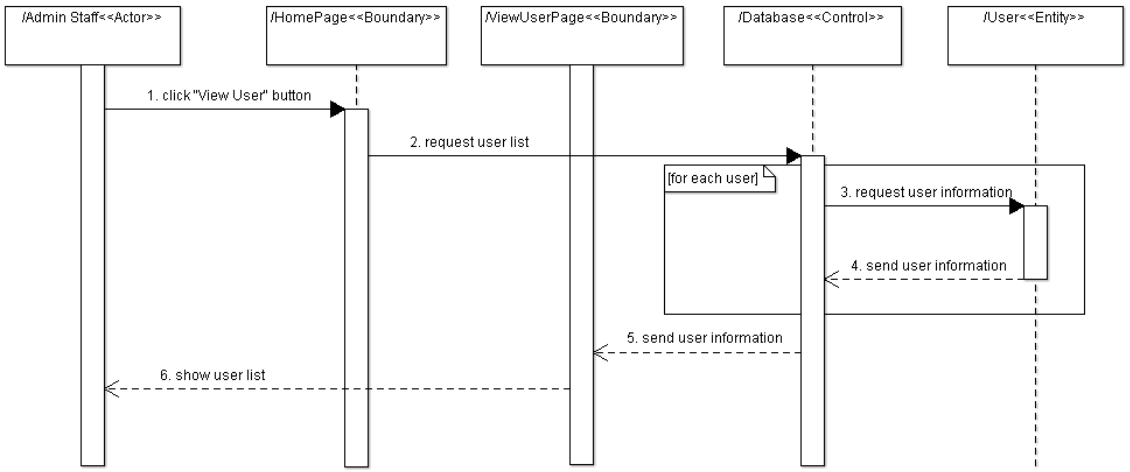


Flow of event

1. User, who may be admin staff, customer and driver, clicks “PROFILE” button.
2. Home page asks Database for user information.
3. Database retrieves user information.
4. User entity returns user information.
5. Database returns user information to Home Page.
6. Home Page shows the user information in a form.
7. User can view and edit the user information.
8. Home Page gets the updated user information and sends to Database.
9. Database updates the user information.
10. User entity returns a message.
11. Database sends the message to Home Page.
12. Home Page show the message whether the user updates their user information successfully.

5. Viewing user

   Class responsibility: 2.2

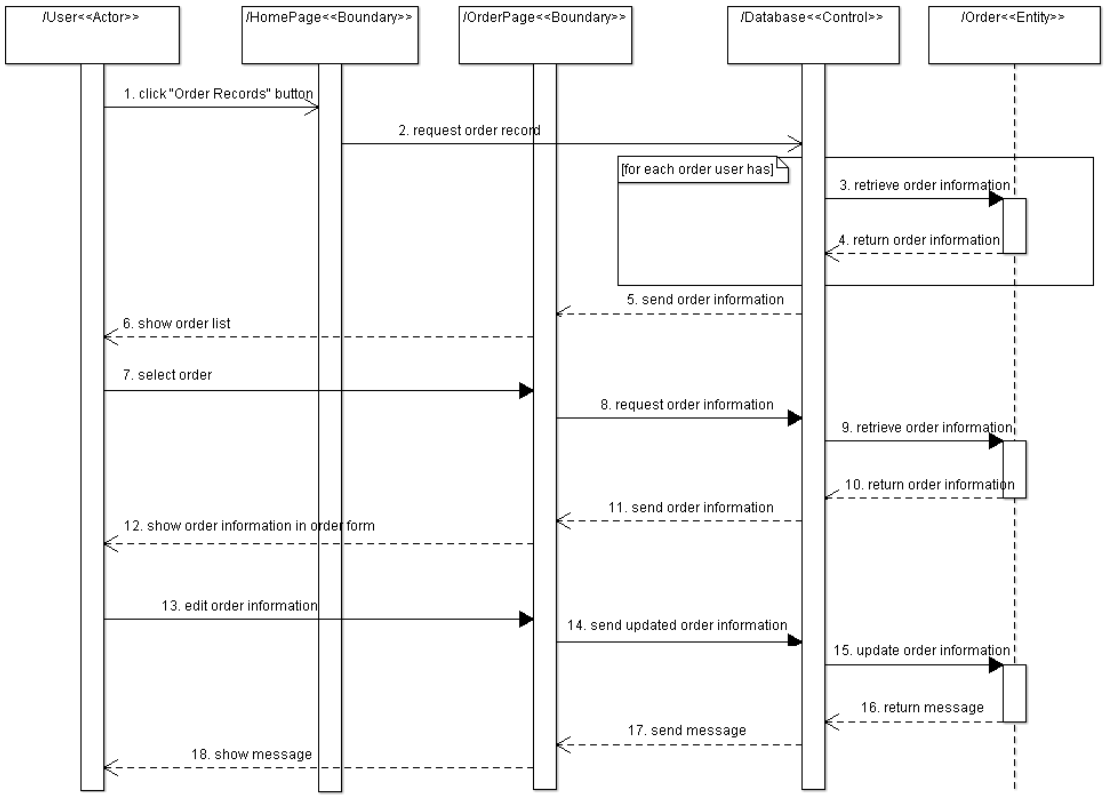


Flow of event

1. Admin staff clicks “View User” button.
2. Home Page request user list.
3. Database retrieves all of the user information.
4. User entity returns user information.
5. Database sends the user information to View User Page.
6. View User Page shows the user list, which contains all of the general user information.

6. Viewing and editing order

   Class responsibility: 2.3, 2.4, 2.5, 3.2, 3.3, 4.1

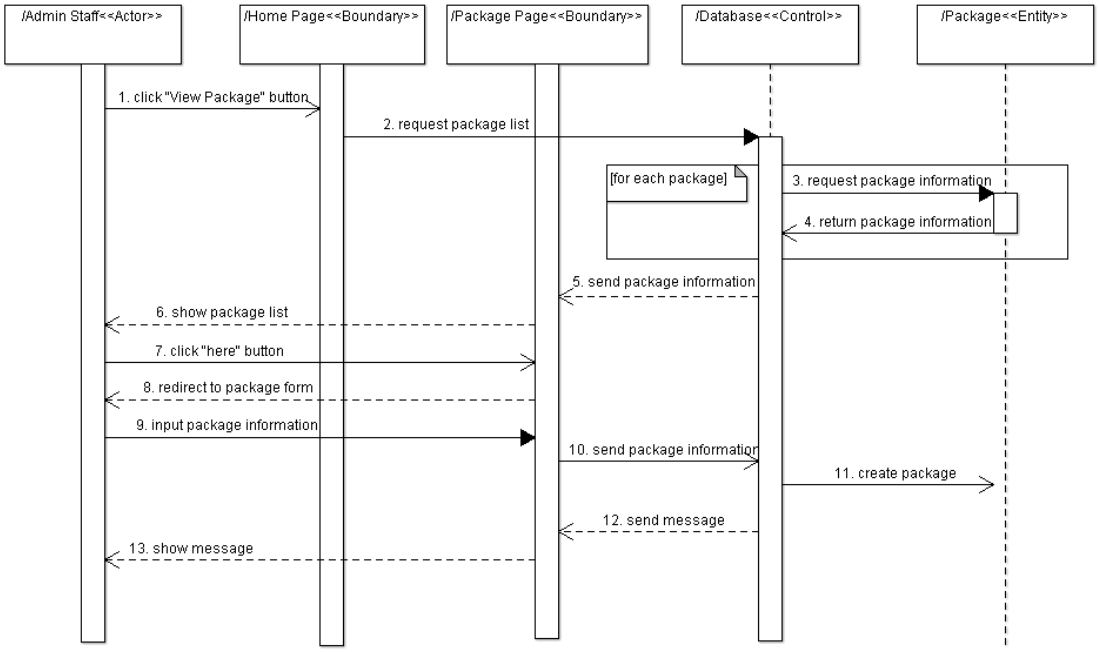


Flow of event

1. User clicks “Order Records” button for requesting the order records.
2. Home Page requests order records.
3. Database retrieves order information. If the user is customer, Database retrieves order customer has. If the user is admin staff, Database retrieves all of the order.
4. Order entity returns order information.
5. Database sends order information to Order Page.
6. Order Page shows order list.
7. User selects order from the order list.
8. Order Page requests specific order information.
9. Database retrieves order information.
10. Order entity returns order information.
11. Database sends order information to Order Page.
12. Order Page shows order information in the order form.
13. User can view and edit order information.
14. Order Page sends updated order information to Database.
15. Database updates order information.
16. Order entity returns a message.
17. Database sends the message to Order Page.
18. Order Page shows the message whether the order is edited successfully.

7. Creating package

   Class responsibility: 2.7

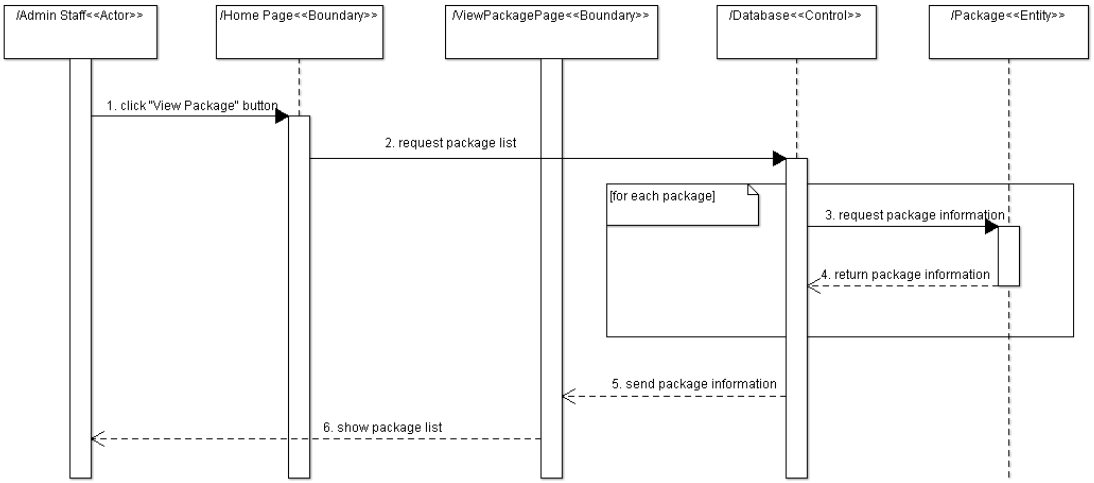


Flow of event

1. Admin staff clicks “View Package” button for requesting the package list.
2. Home page requests the package list.
3. Database retrieves all of the existing package information.
4. Each existing package returns its information.
5. Database sends the package information to Package page.
6. Package page shows the general package information in the package list.
7. After reviewing the package list, admin staff can click “here” button for creating new package if needed.
8. Package page shows a package form.
9. Admin staff inputs and saves package information.
10. Package page sends package information to database.
11. Database creates new package.
12. Database sends message after creating the package.
13. Package page shows message whether the message is created successfully.

8. Viewing package

   Class responsibility: 2.8

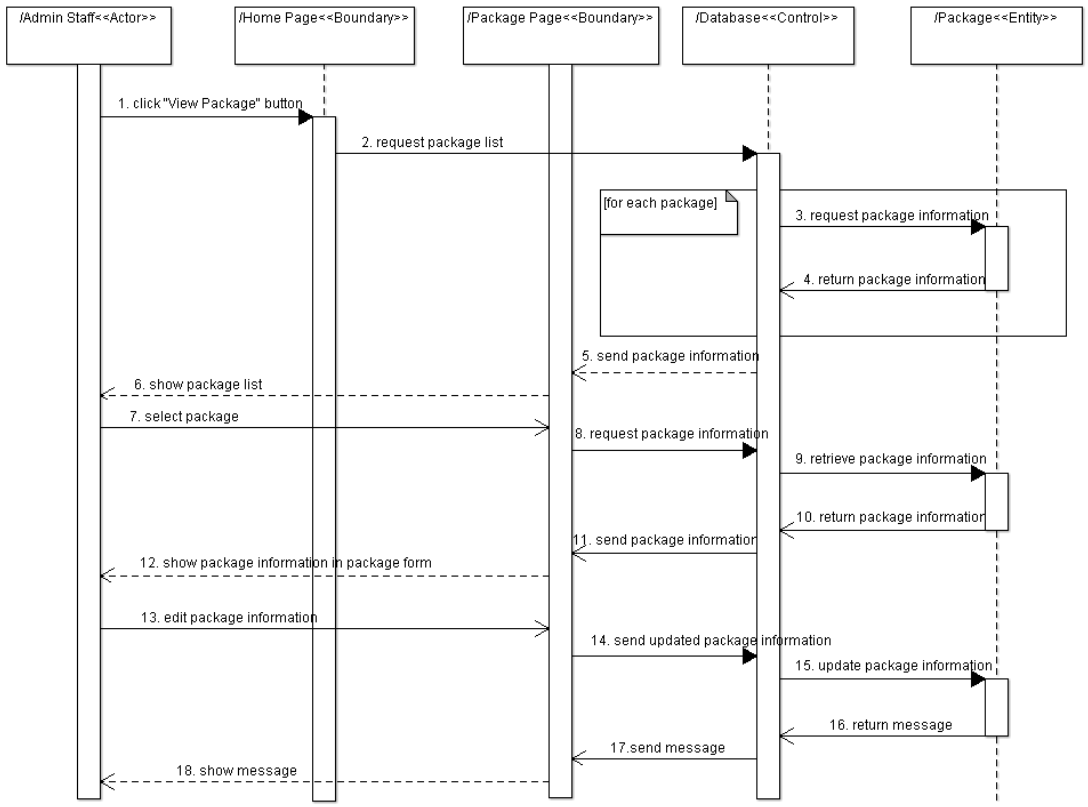


Flow of event

1. Admin staff clicks “View Package” button for requesting the package list.
2. Home page requests the package list.
3. Database retrieves all of the existing package information.
4. Each existing package returns its information.
5. Database sends the package information to Package page.
6. Package page shows the general package information in the package list.

9. Editing package

   Class responsibility: 2.9

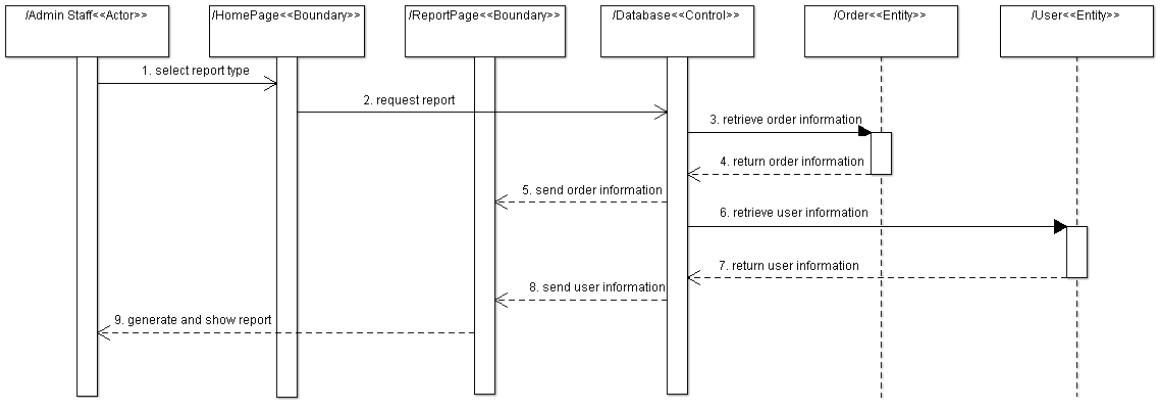


Flow of event

1. Admin staff clicks “View Package” button for requesting the package list.
2. Home page requests the package list.
3. Database retrieves all of the existing package information.
4. Each existing package returns its information.
5. Database sends the package information to Package page.
6. Package page shows the general package information in the package list.
7. Admin staff selects specific package.
8. Package page request specific package information.
9. Database retrieves the package information.
10. Selected package returns its information.
11. Database sends the package information to Package page.
12. Package page shows the package information in the package form.
13. Admin staff can edit and save the updated package information.
14. Package page sends the updated package information to database.
15. Database updated the package.
16. Package returns a message after updating.
17. Database sends the message to Package page.
18. Package page shows the message whether the package is edited successfully.

10. Viewing report

Class responsibility: 2.10

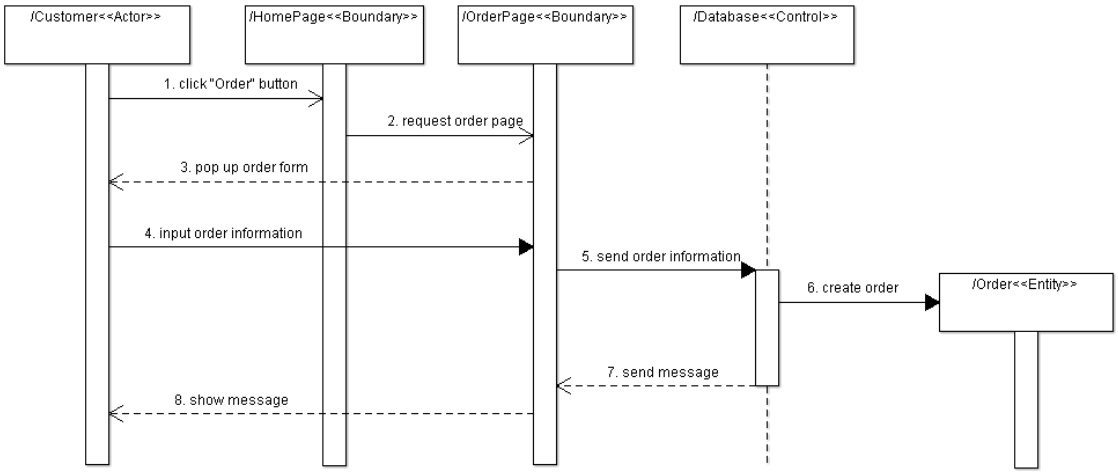


Flow of event

1. Admins staff selects report type.
2. Home Page gets the report type and request the report.
3. Database retrieves order information.
4. Order entity returns order information.
5. Database sends order information to Report Page.
6. Database retrieves user information.
7. User entity returns user information.
8. Database sends user information to Report Page.
9. Report Page generates and shows the report.

11. Creating order

Class responsibility: 3.1

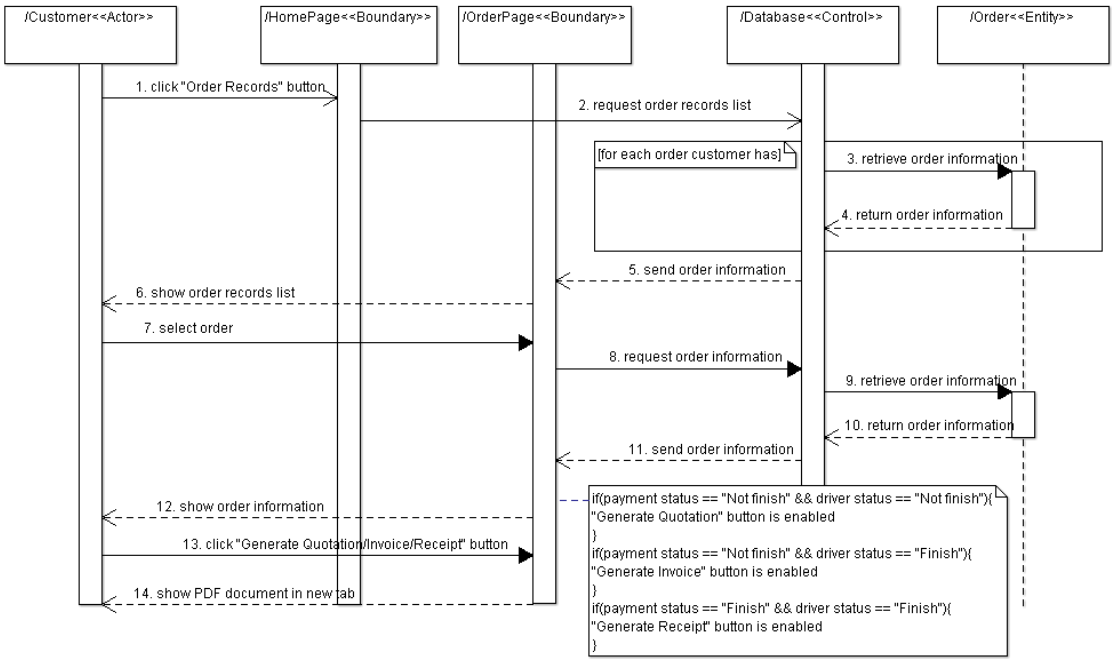


Flow of event

1. Customer clicks “Order” button after selecting specific order type.
2. Home Page requests Order Page.
3. Order Page pops up the order form.
4. Customer inputs order information.
5. Order Page gets the order information and sends it to Database.
6. Database creates new order.
7. Database sends a message to Order Page.
8. Order Page shows the message whether the order is created successfully.

12. Generating PDF document

Class responsibility: 3.4

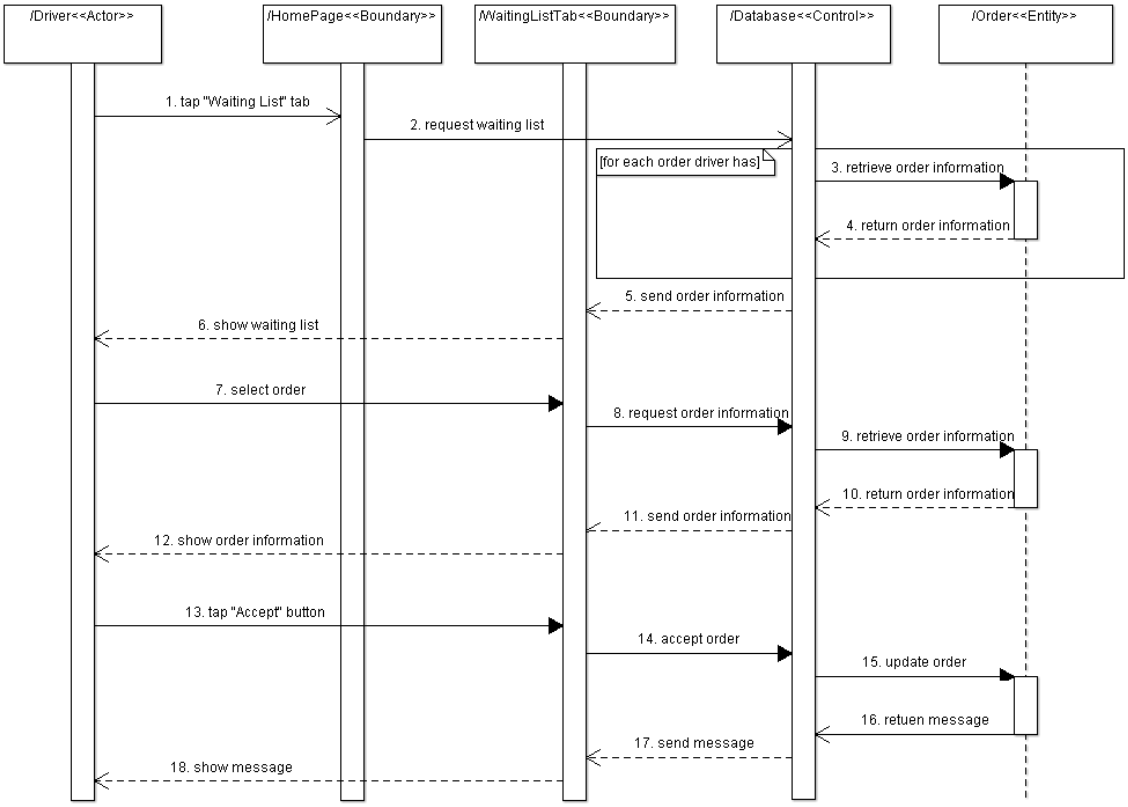


Flow of event

1. Customer clicks “Order Records” button for requesting the order records list.
2. Home Page requests order records list.
3. Database retrieves order information the customer has.
4. Order entity returns order information.
5. Database sends order information to Order Page.
6. Order Page shows order records list.
7. Customer selects order form the order records list.
8. Order Page requests specific order information.
9. Database retrieves order information.
10. Order entity returns order information.
11. Database sends order information to Order Page.
12. Order Page shows order information in the order form and the “Generate Quotation/Invoice/Receipt” button based on different order status including payment and driver status.
13. Customer click “Generate Quotation/Invoice/Receipt” button.
14. Order Page shows the PDF document in a new tab.

13. Accepting order

Class responsibility: 4.2

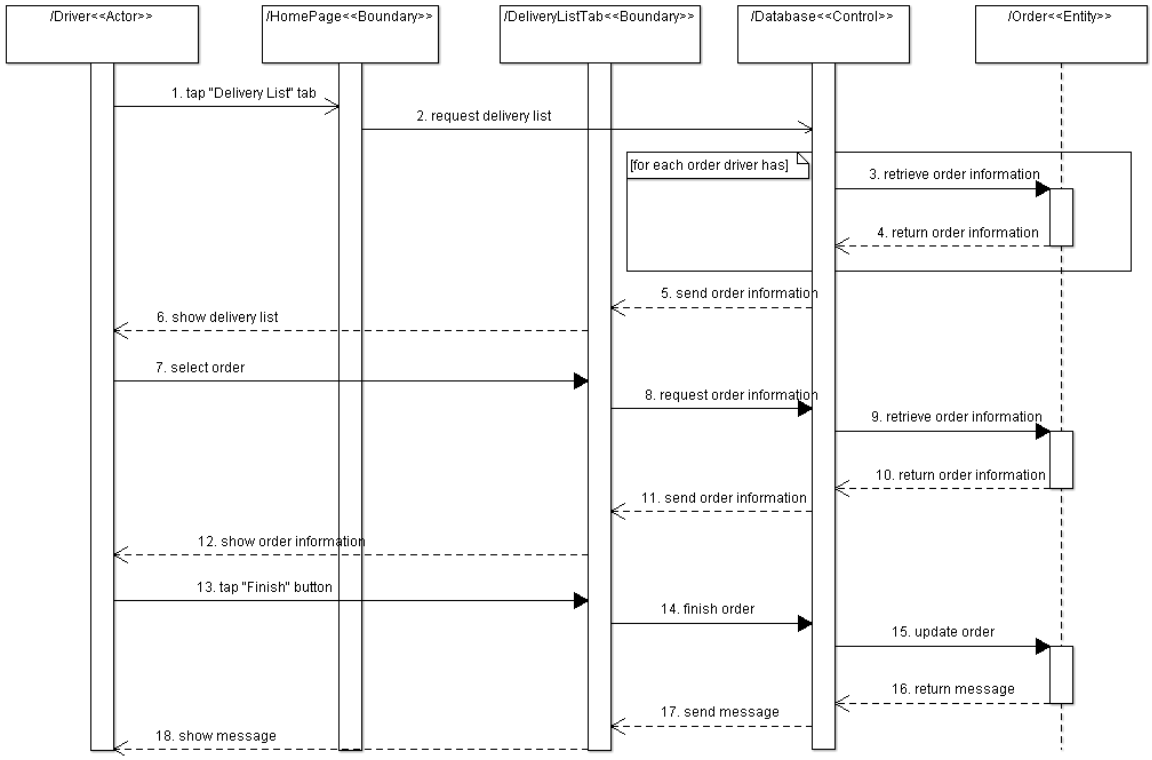


Flow of event

1. Driver taps “Waiting List” tab for requesting the order list in mobile application.
2. Home Page requests waiting list.
3. Database retrieves the orders, which is not assigned driver.
4. Order entity returns order information.
5. Database sends order information to Waiting List Tab.
6. Waiting List Tab shows the waiting list.
7. Driver selects order.
8. Waiting List Tab requests order information.
9. Database retrieves specific order information.
10. Order entity return order information.
11. Database sends order information to Waiting List Tab.
12. Waiting List Tab shows the order information.
13. Driver taps “Accept” button.
14. Waiting List Tab accepts the order the driver selects.
15. Database updates the order.
16. Order entity returns a message.
17. Database sends the message to Waiting List Tab.
18. Waiting List Tab shows the message whether the driver accept the order successfully.

14. Finishing order

Class responsibility: 4.3

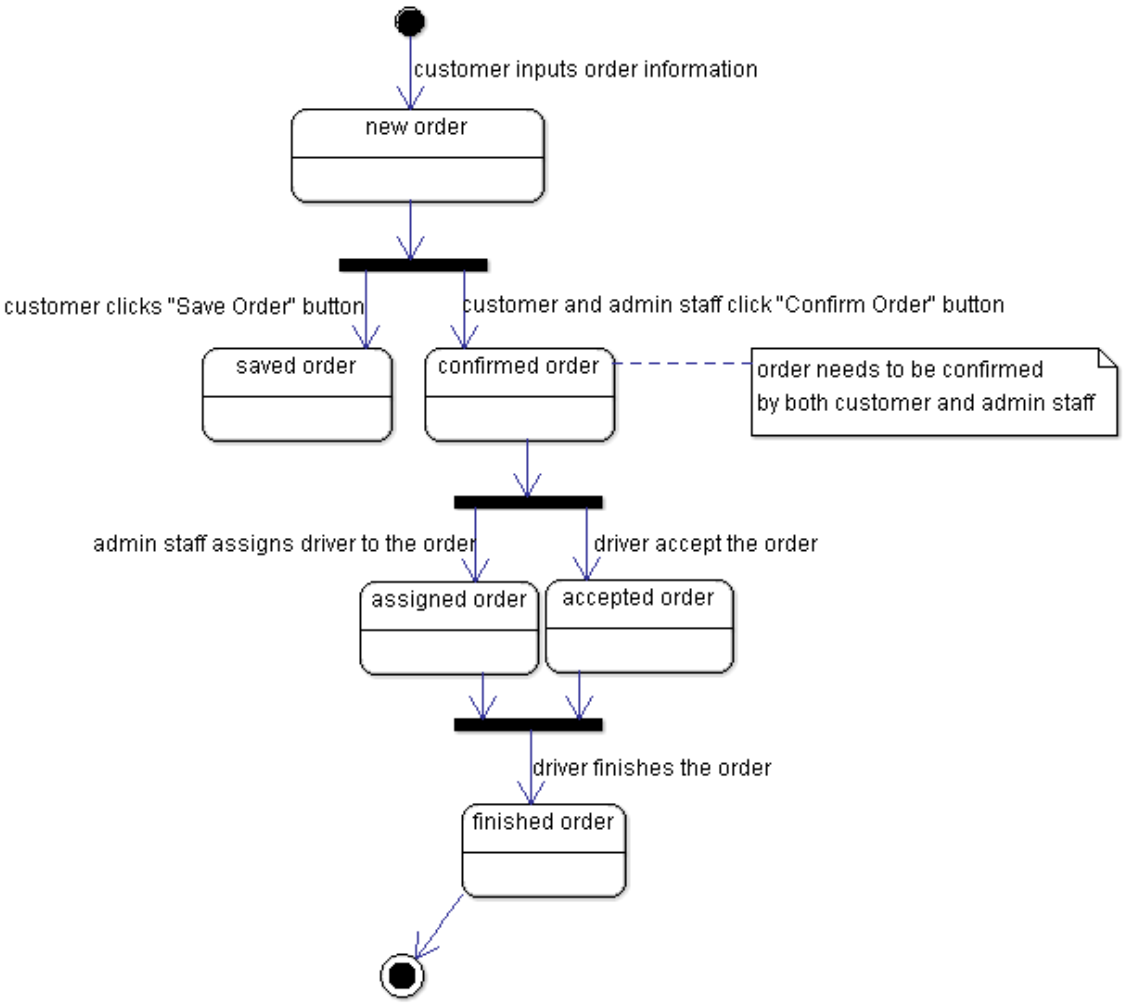


Flow of event

1. Driver requests order list in “Delivery List” tab in mobile application.
2. Home Page requests delivery list.
3. Database retrieves the orders the driver has.
4. Order entity returns order information.
5. Database sends order information to Delivery List Tab.
6. Delivery List Tab shows the delivery list.
7. Driver selects order.
8. Delivery List Tab requests order information.
9. Database retrieves specific order information.
10. Order entity return order information.
11. Database sends order information to Delivery List Tab.
12. Delivery List Tab shows the order information.
13. Driver taps “Finish” button.
14. Delivery List Tab accepts the order the driver selects.
15. Database updates the order.
16. Order entity returns a message.
17. Database sends the message to Delivery List Tab.
18. Delivery List Tab shows the message whether the driver finish the order successfully.

4.3 State Chart

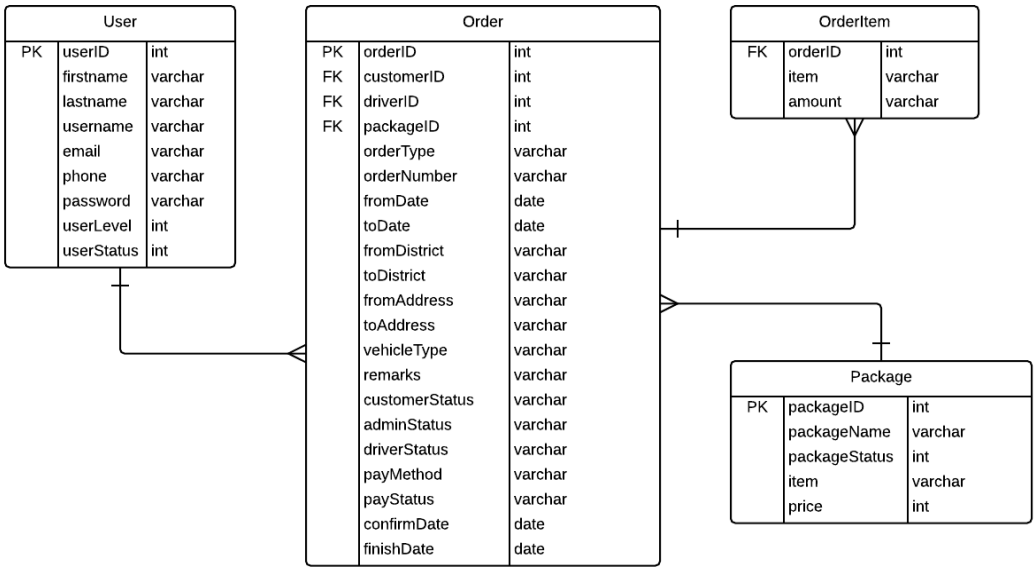
In the Moving System, the most complicated process is ordering process and there are different states for the order in the whole process. Therefore, the following state chart is about the ordering process.



The process is starting from creating the order. The customer inputs order information such as delivery date, address and items. Then, the customer can save the order or confirm the order. After both customer and admin staff confirm the order, admin staff can assign driver to the order or driver can accept the order in the waiting list of mobile application. After the delivery, driver can finish the order and the whole process can be completed.

4.4 Database Design & Physical Data Model

About the database for the Moving System, MySQL database is used and the database design is presented by the following Entity Relationship (ER) diagram.



The system has four tables for storing the user and transaction data including User, Order, OrderItem and Package table. The relationship between User and Order table and the relationship between Order and OrderItem are one-to-many. One user can have zero or many orders and one order can have zero or many order items. Also, because of different types of customer having price package, the relationship between Order and Package table is many-to-one.

Chapter 5

System Implementation

CONTENTS OF CHAPTER 5

5.1 Use of Software/Library

In this honours project, different open source software and libraries are used including:

1. **XAMPP**
   It can help to establish an Apache server for storing the web application file. Also, it can build MySQL database in the server for storing all of the system data.

2. **NetBeans**
   It is a development tool for developing the web application such as designing the web page and writing different programming language. In this honours project, HTML and PHP are mainly used for web design and database communication.

3. **Xcode**
   It is a development tool for developing the mobile application such as designing the user interface. In this project, Swift is used for designing the mobile application.

4. **ArgoUML**
   It is a UML tool for drawing UML diagram including use case diagram, activity diagram, sequence diagram and so forth.

5. **Google Map API**
   In this honours project, some functions need to calculate the distance and show the path on the map. Therefore, Google Map API is used in both web and mobile applications.

5.2 System Constraints

There are some constraints for the Moving System and they are listed on the following:

1. **System can only be applied in small or middle companies**
   The scale of company is limited because of the server and the database. They are both open source and have lower response time if many users access and retrieve data concurrently.

2. **System can only be available for local transactions**
   System cannot handle the transactions from other countries since the Google Map API is set to calculate order delivered by trucks. If the order needs to be delivered by plane, the whole price calculation process will have errors.

3. **Mobile application can run in iOS platform only**
   On account of time limitation, the mobile application is developed in one system platform, iOS. Therefore, only iOS mobile user can use this mobile application to connect with the Moving System and process the orders.

5.3 Test Strategies

About the testing approaches, there are different techniques, which are used in the following including:

1. **Black-box test**

   This testing is to describe the Actual Result and Expected Result in different system functions and outstanding the abnormal response. The test cases results are listed in appendices (Black-box test).

2. **Stress test**

   loadimpact.com generates a number of users to test the system load performance. They make the website requests concurrently and the tool analysis the load time.

3. **Browser test**

   BrowserStack simulates different devices including Desktop PC and mobile phone and test the website in various web browsers.

Through above tests, the following rules and processes are identified:

1. The result of different system functions
2. The network performance
3. The user interface in different platforms

5.4 List of items and features to be tested

The following list is about the aspects, which are concentrated on during the web and mobile application testing processes in black-box test and browser test.

1. **User management process**
    a. Registration
    b. Authentication
    c. Resetting password
    d. Editing profile

2. **Customer order management process**
    a. Creating order
    b. Editing order
    c. Generate PDF document (Quotation, Invoice & Receipt)

3. **Delivery management process**
    a. Assigning driver
    b. Accepting order
    c. Finishing order

4. **Report generation process**
    a. Order report

5.5 Passing or failing criteria for items / features

About the completion criteria for this plan, there are some goals to be achieved.

1. All test cases are completed.
2. The actual results are as same as the expected results in black box test.
3. The load time should no more than 1 second under 10 users in stress test.
4. The web interface should be clearly shown in all of the selected browsers in browser test.

5.6 Test deliverables

There are several items, which are delivered in this test plan.

1. Test plan document
2. Test cases
3. Processes and their results
4. Testing tools and their outputs

5.7 Stress test

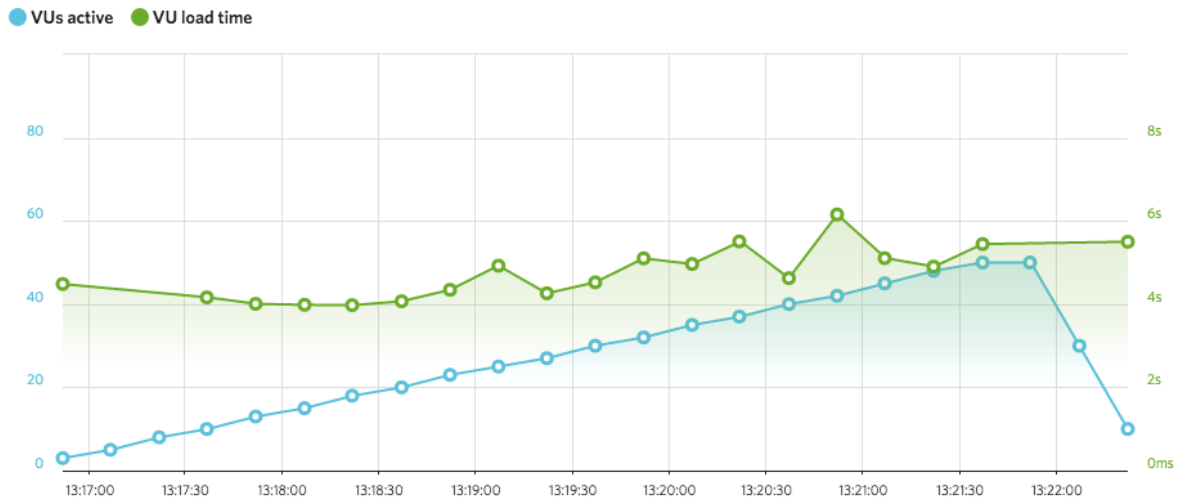


Figure 1: User load time diagram

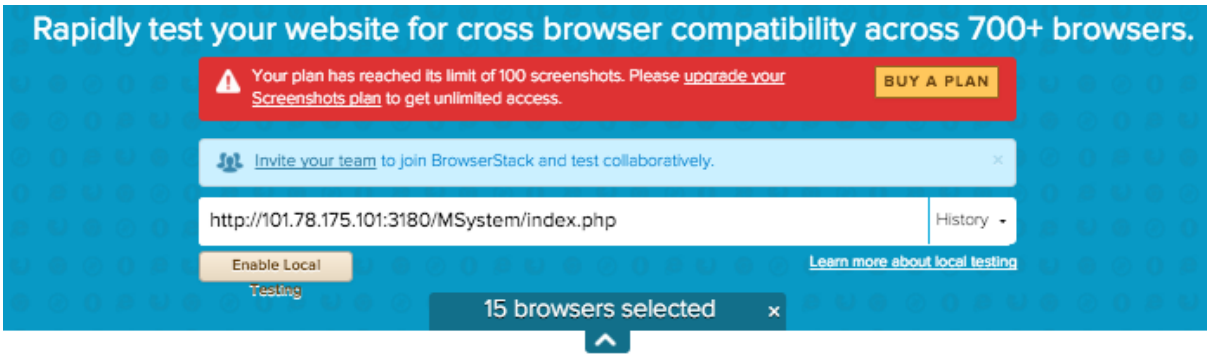
From figure 1, the number of user is gradually increased and reaches to the maximum 50 while the user load time remains not more than 6 seconds at most of time.

Conclusion of stress test

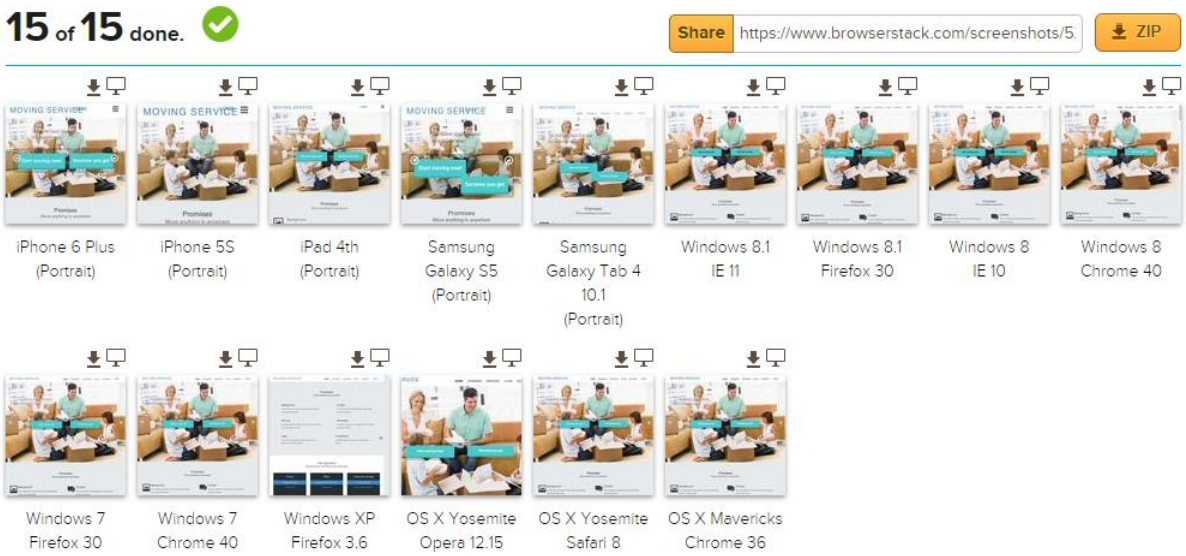
The load time is limited by the bandwidth of Hong Kong Baptist University since the whole system is uploaded onto the virtual machine in Computer Science Faculty. Of course, there is lots of room for improvement on different aspects such as database design and hardware specification.

5.8 Browser test

In the following table, test case for system rendering between the fifteen browsers will be compared. Generally, there are 5 types of browser including Internet Explorer, Chrome, Firefox, Opera and Safari. Also, they are run in different operating system platform such as Windows, Mac OS, iOS and Android.



iOS	v8.3 iPad Mini 2 iPad Air iPhone 6 iPhone 6 Plus	v7 iPhone 5S iPad 4th iPad Mini	v6 iPhone 4S iPhone 5 iPad 3	v5.1 iPhone 4S iPad 3	v5 iPad 2	
Android	Samsung Galaxy S5 Galaxy S4 Galaxy S3 Galaxy S2 Galaxy Tab 2 Galaxy S5 Mini Galaxy Tab 4 Galaxy Note 3 Galaxy Note 2 Galaxy Note 10.1 Galaxy Note	Amazon Kindle Fire 2 Kindle Fire HD 8.9 Kindle Fire HDX	Motorola Razr Droid Razr Razr Maxx HD	Google Nexus 9 Nexus 7 Nexus 6 Nexus 5 Nexus 4 Nexus	HTC One M8 One X Wildfire	Sony Xperia Tipo
Windows 8.1	11	37 36 35 34 33 32 31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16	42 41 40 39 38 37 36 35 34 33 32 31 30 29 28 27 26 25 24 23 22	12.16 12.15 12.14 12.10 12	5.1*	
Windows 8	10	37 36 35 34 33 32 31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16	42 41 40 39 38 37 36 35 34 33 32 31 30 29 28 27 26 25 24 23 22	12.16 12.15 12.14 12.10 12	5.1*	
Windows 7	11 10 9 8	37 36 35 34 33 32 31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3.6	42 41 40 39 38 37 36 35 34 33 32 31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14	12.16 12.15 12.14 12.10 11.6	5.1*	
Windows XP	7 6	37 36 35 34 33 32 31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3.6 3	42 41 40 39 38 37 36 35 34 33 32 31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14	12.16 12.15 12.14 12.10 11.6	5.1*	
Mac OS X Yosemite		37 36 35 34 33 32 31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3.6	42 41 40 39 38 37 36 35 34 33 32 31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 14	12.15 12.14 12.12	8	
Mac OS X Mavericks		37 36 35 34 33 32 31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15	42 41 40 39 38 37 36 35 34 33 32 31 30 29 28 27 26 25 24 23	12.15 12.14 12.12 12 11.6	7.1	



Browser	Internet Explorer (IE)	Chrome	Firefox	Opera	Safari
Platform	Windows	Windows, Mac OS	Windows	Mac OS	Mac OS
Version	8 or above	7 or above for Windows and Mac OS X Mavericks	XP or above	Mac OS X Yosemite	Mac OS X Yosemite
CSS	✓	✓	✓	✓	✓
HTML	✓	✓	✓	✓	✓
Page Validation	✓	✓	✓	✓	✓
Font Size Validation	✓	✓	✓	✓	✓
All Image Alignment	✓	✓	✓	✓	✓
Header and Footer Alignment	✓	✓	✓	✓	✓
Page Alignment	✓	✓	✓	✓	✓
Verification of information submitted of database	✓	✓	✓	✓	✓

Text Alignment	✓	✓	✓	✓	✓
Cookies	✓	✓	✓	✓	✓
Ajax	✓	✓	✓	✓	✓
JQuery	✓	✓	✓	✓	✓

In the following table, test case for system function between 5 types of browser will be compared.

Browser	Internet Explorer (IE)	Chrome	Firefox	Opera	Safari
Platform	Windows	Windows, Mac OS	Windows	Mac OS	Mac OS
Version	8 or above	7 or above for Windows and Mac OS X Mavericks	XP or above	Mac OS X Yosemite	Mac OS X Yosemite
1. User management subsystem					
Create user	✓	✓	✓	✓	✓
Validate login	✓	✓	✓	✓	✓
Update user profile (by customer)	✓	✓	✓	✓	✓
Update user profile (by admin staff)	✓	✓	✓	✓	✓
Update user profile (by driver)	×	×	×	×	×
Forget Password	✓	✓	✓	✓	✓
2. Customer order management subsystem					
Create order	✓	✓	✓	✓	✓
Edit order (by customer)	✓	✓	✓	✓	✓
Edit order (by admin staff)	✓	✓	✓	✓	✓
Generate PDF document (Quotation, Invoice & Receipt)	✓	✓	✓	✓	✓
3. Delivery management subsystem					
Assign driver	✓	✓	✓	✓	✓
Accept order	×	×	×	×	×
Finish order	×	×	×	×	×

View order history	✗	✗	✗	✗	✗
4. Report subsystem					
View order report	✓	✓	✓	✓	✓

Conclusion of browser test

Some of the testing items are failed because these items are run in mobile application platform instead of web application platform. Therefore, the results of these items are failed but they are fully functional in mobile application.

Most of the system functions work very well but there is still room for improvement of system layout in different screen resolution because some HTML elements such as button are located incorrectly when the screen size is very small.

Chapter 6

Discussion & Conclusion

CONTENTS OF CHAPTER 6

6.1 Discussion

Difficulties

During this honours project, there are some difficulties I need to deal with. First, it is mobile application development. Since I just learn Java for Android application, but I want to learn a new programming language in this project, Swift for iOS application is chosen finally. In the summer holiday, lots of web tutorials are taken including designing user interface and transferring data between mobile phone and web database.

Moreover, transferring my ideas to the logical application is another barrier. Take simplifying order process as an example. My idea is that the customer can easily get the price of the delivery order and time of involving the admin staff can be minimized. Therefore, I spend some time for searching the API for getting the distance between different delivery locations and designing algorithms for calculating the delivery items, staff salary and car rental at the same time. Also, after the calculation, the price can be illustrated in the PDF documents such as quotation, invoice and receipt.

Personal Gain

Overcoming above obstacles, new programming skills and system design techniques are gained in this honours project. This valuable experience can help me for the future career. For instance, the system can be developed in both Android and iOS platform. Even though there are some new programming language, I can learn by myself quickly. As a result, skill for adapting myself to new environment is obtained as well.

Some project management and presentation skills are gained except above technical skills. In this honours project, well project development schedule is needed as everything is addressed by myself only. In addition, I learn how to present my system design better in the report such as adding the gantt chart for the project schedule and adding number in front of the subtitles, so the readers can follow the report easily.

6.2 Conclusion

Moving System is a commercial system and help the moving company to simplify their business processes. For the web application, it is mainly for the customer and the admin staff. The customer can easily make a delivery order any time. More importantly, the system can help the admin staff to calculate the order price quickly and accurately. The distance and duration of different delivery locations, staff salary and number of delivery items are calculated; hence, different order documents with the order price can be generated including quotation, invoice and receipt. As you can see, the workload of the admin staff can be reduced and the customer can also review and get the related order documents conveniently.

For the mobile application, it can help the admin staffs and the drivers to communicate with each other. The drivers can simply take the delivery order the admin staffs assign and have clear order information and delivery path on the map. Therefore, the time of each delivery order can be reduced and the company resources can be used efficiently.

6.3 Foreseeable Modification and Enhancement

In this honours project, there still is room for improvement and they are listed in the following:

1. **Provide more different types of report**
   The web application provides the admin staff with one type of report only, so more kinds of report should be generated such as key performance indicator (KPI) of the divers. Also, the mobile application should offer the drivers with the payroll report for showing their salary records and number of order they finish every month.

2. **Update the map information automatically**
   The mobile application can show one path with two destinations only. It is inconvenient if the delivery order includes multiple destinations. As a consequence, the mobile application should auto-detect the current location and change the delivery path.

3. **Provide more opportunities for the driver to communicate with the admin staff**
   After the delivery order, the mobile application should form or message box to send some additional information of the order. Thus, the communication between the drivers and the admin staffs can be improved and closer.

4. **Develop mobile application in both main system platform, iOS and Android**
   The mobile application can support iOS platform only in this honours project. Therefore, Android version should be developed, so the mobile application can support different kinds of the mobile users.

Reference

STAR MOVING SERVICE COMPANY (2014) *Service Flow* Retrieved June 15, 2015 from the World Wide Web: http://www.starmovinghk.com/service2.php?ID=3

AMAN MOVING SERVICES (2016) *Price List Retrieve* June 18, 2015 from the World Wide Web: http://www.aman01.com/add01/tc/05_01.asp

SANTA FE RELOCATION SERVICES (2016) *Moving Services* Retrieved June 21, 2015 from the World Wide Web: http://www.santaferelo.com/hk/relocating-myself/moving-services

PHP (2001) *Php Manual* Retrieved August, 2015 from the World Wide Web: http://php.net/manual/en/

Bootstrap *Components* Retrieved August, 2015 from the World Wide Web: http://getbootstrap.com/components/

W3Schools *JavaScript Tutorial* Retrieved September, 2015 from the World Wide Web: http://www.w3schools.com/js/

Apple (2016) *The Swift Programming Language (Swift 2.1)* Retrieved September, 2015 from the World Wide Web: https://developer.apple.com/library/ios/documentation/Swift/Conceptual/Swift_Programming_Language/index.html#//apple_ref/doc/uid/TP40014097-CH3-ID0

Google Developers *Google Map SDK for iOS* Retrieved October, 2015 from the World Wide Web: https://developers.google.com/maps/documentation/ios-sdk/

Highcharts (2016) *Highcharts Demos* Retrieved November, 2015 from the World Wide Web: http://www.highcharts.com/demo

Appendices

CONTENTS OF APPENDICES

A1. Black-box Test Cases

Event 1 User management subsystem

Event 1.1 Create user

Test case	Test Condition	Expected result	Actual result
1	Visitor does not input required fields including first name, last name, user name, email, phone number and password	Refresh the home page and display error message – "All of the fields should be filled in"	Same as expected output
2	Visitor inputs existing user name	Refresh the home page and display error message – "User name already exists"	Same as expected output
3	Visitor inputs mismatch password	Refresh the home page and display error message – "Password does not match"	Same as expected output
4	Visitor inputs valid user information	Create new user account and refresh the home page Display message – "User is created successfully. Please try to login"	Same as expected output

Event 1.2 Validate login

Test case	Test Condition	Expected result	Actual result
1	Visitor does not input user name and password	Refresh the home page and display error message – "Null user name and password input"	Same as expected output
2	Visitor inputs invalid user name or password	Refresh the home page and display error message – "Wrong user name OR password"	Same as expected output
3	Visitor inputs valid user name and password	Login the system and refresh the home page	Same as expected output

Event 1.3 Update user profile (by customer in web application)

Test case	Test Condition	Expected result	Actual result

1	Customer inputs mismatch password	Refresh the home page and display error message – "Password does not match"	Same as expected output
2	Customer inputs valid password	Refresh the home page and display message – "User profile is updated"	Same as expected output
3	Customer edits first name, last name, email and phone number	Refresh the home page and display message – "User profile is updated"	Same as expected output

Event 1.4 Update user profile (by admin staff in web application)

Test case	Test Condition	Expected result	Actual result
1	Admin staff inputs mismatch password	Refresh the profile page and display error message – "Password does not match"	Same as expected output
2	Admin staff inputs valid password	Refresh the profile page and display message – "User is updated successfully"	Same as expected output
3	Admin staff edits first name, last name, email, phone number, user level and status	Refresh the profile page and display message – "User is updated successfully"	Same as expected output

Event 1.5 Update user profile (by driver in mobile application)

Test case	Test Condition	Expected result	Actual result
1	Driver inputs mismatch password	Refresh the profile tab and display error message – "New passwords do not match"	Same as expected output
2	Driver inputs valid password	Refresh the profile tab and display message – "Password is changed"	Same as expected output
3	Driver edits first name, last name, email and phone number	Refresh the profile tab and display message – "User Profile is updated"	Same as expected output

Event 1.6 Forget password

Test case	Test Condition	Expected result	Actual result
1	Visitor does not input user name or email	Refresh the home page and display error message – "Either one of the fields should be filled in"	Same as expected output
2	Visitor inputs wrong user name or email	Refresh the home page and display error message – "Please try again"	Same as expected output
3	Visitor inputs valid user name or email	Reset user password, send the email to user email account with new password and refresh the home page Display message – "User password is reset! Please check your email"	Same as expected output

Event 2 Customer order management subsystem

Event 2.1 Create order

Test case	Test Condition	Expected result	Actual result
1	Customer does not input delivery from date or to date	Pop up a warning prompt – "Please fill out this field"	Same as expected output
2	Customer fills in valid order information	Redirect to the home page and display message – "The order is saved. Thank you very much!"	Same as expected output

Event 2.2 Edit order (by customer)

Test case	Test Condition	Expected result	Actual result
1	Customer changes delivery from date or to date to null value	Pop up a warning prompt – "Please fill out this field."	Same as expected output
2	Customer changes other order information including delivery address, remarks	Redirect to the home page and display message – "The order is saved. Thank you	Same as expected output

	and item list.	very much!”	

Event 2.3 Edit order (by admin staff)

Test case	Test Condition	Expected result	Actual result
1	Admin staff changes the package of the order	Pop up a warning prompt – “Package is updated successfully and the order price is re-calculated.”	Same as expected output
2	Admin staff changes other order information including basic order information, item list, price adjustment and advanced order information.	Redirect to the delivery list page and display message – “Order is updated successfully”	Same as expected output

Event 2.5 Generate PDF document (Quotation, Invoice & Receipt)

Test case	Test Condition	Expected result	Actual result
1	Customer creates new order and reviews the order in order records.	Display “Generate Quotation” button and generate the PDF quotation in the new tab after clicking the button	Same as expected output
2	Customer reviews the order record after the delivery is finished (Driver finished the order in mobile application).	Display “Generate Invoice” button and generate the PDF invoice in the new tab after clicking the button	Same as expected output
3	Customer reviews the order record after making its payment (Admin staff edits the payment of the order).	Display “Generate Receipt” button and generate PDF receipt in the new tab after click the button	Same as expected output

Event 3 Delivery management subsystem

Event 3.1 Assign driver

Test case	Test Condition	Expected result	Actual result
1	Admin staff selects a driver in the advanced order	Redirect to the delivery list page and display message –	Same as expected output

	information of the order, and saves or confirms the order.	"Order is updated successfully"	

Event 3.2 Accept order

Test case	Test Condition	Expected result	Actual result
1	Driver accepts the order of the Waiting List in the mobile application.	Refresh the Waiting List and display message – "Order is accepted and listed in the delivery list"	Same as expected output

Event 3.3 Finish order

Test case	Test Condition	Expected result	Actual result
1	Driver accepts the order of the Delivery List in the mobile application.	Refresh the Delivery List and display message – "Order is finished"	Same as expected output

Event 3.4 View order history

Test case	Test Condition	Expected result	Actual result
1	Driver selects the order of the History in the mobile application.	Display the order information	Same as expected output

Event 4 Report subsystem

Event 4.1 View order report

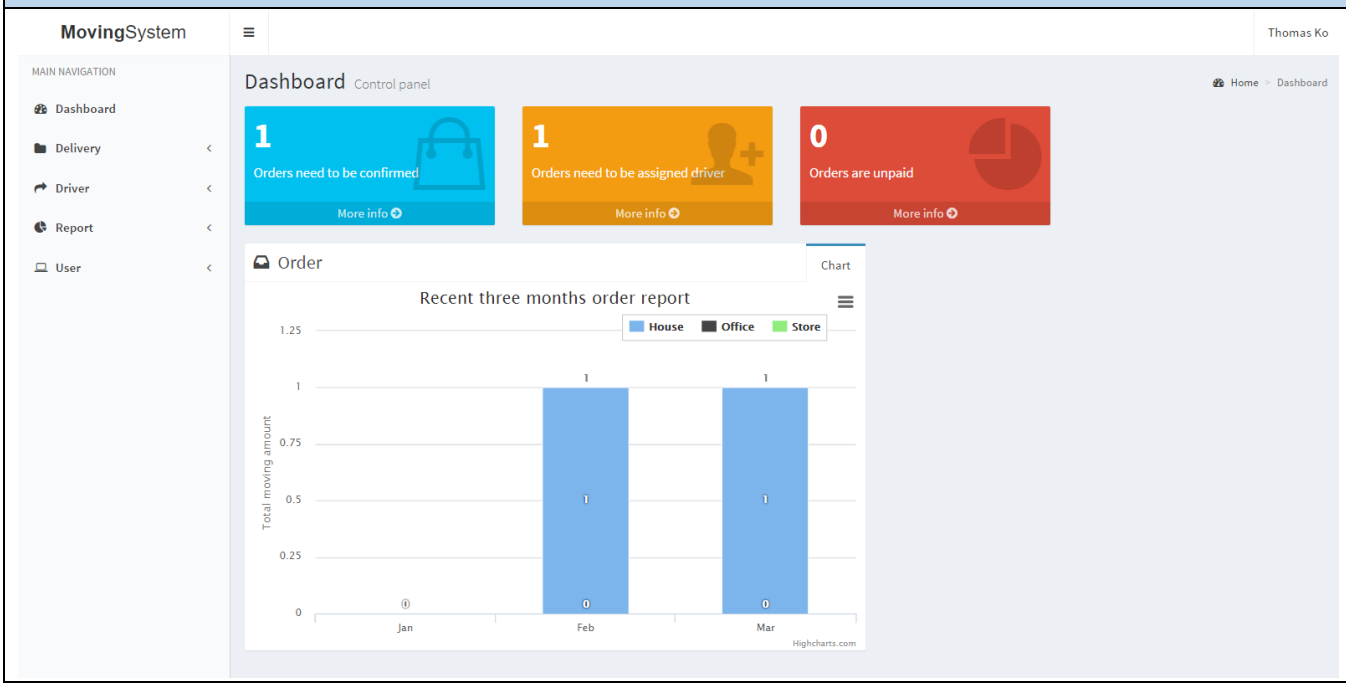
Test case	Test Condition	Expected result	Actual result
1	Admin staff selects Order Report of Report in the menu	Display the order report about current year	Same as expected output
2	Admin staff selects different order year	Change the report which is about the specific year admin staff selects	Same as expected output

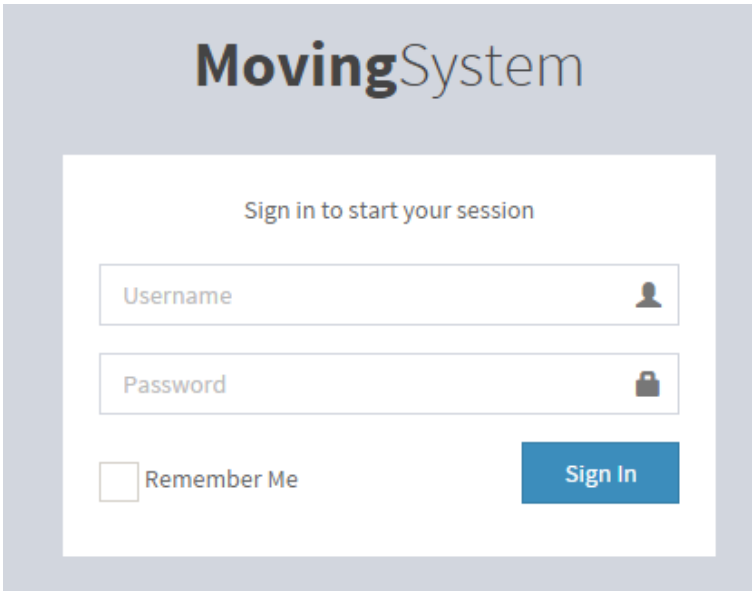
Conclusion of black-box test

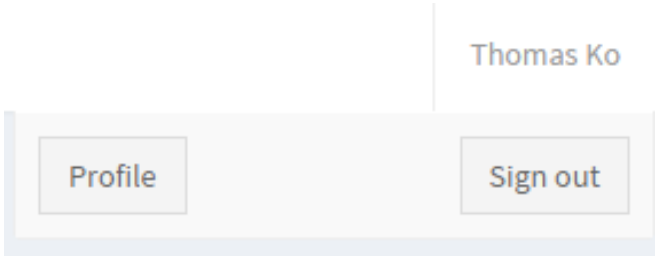
From the test result, all input processes work very well and be able to generate the expected result.

A2. User Manual – Administrator

A2.1 About homepage

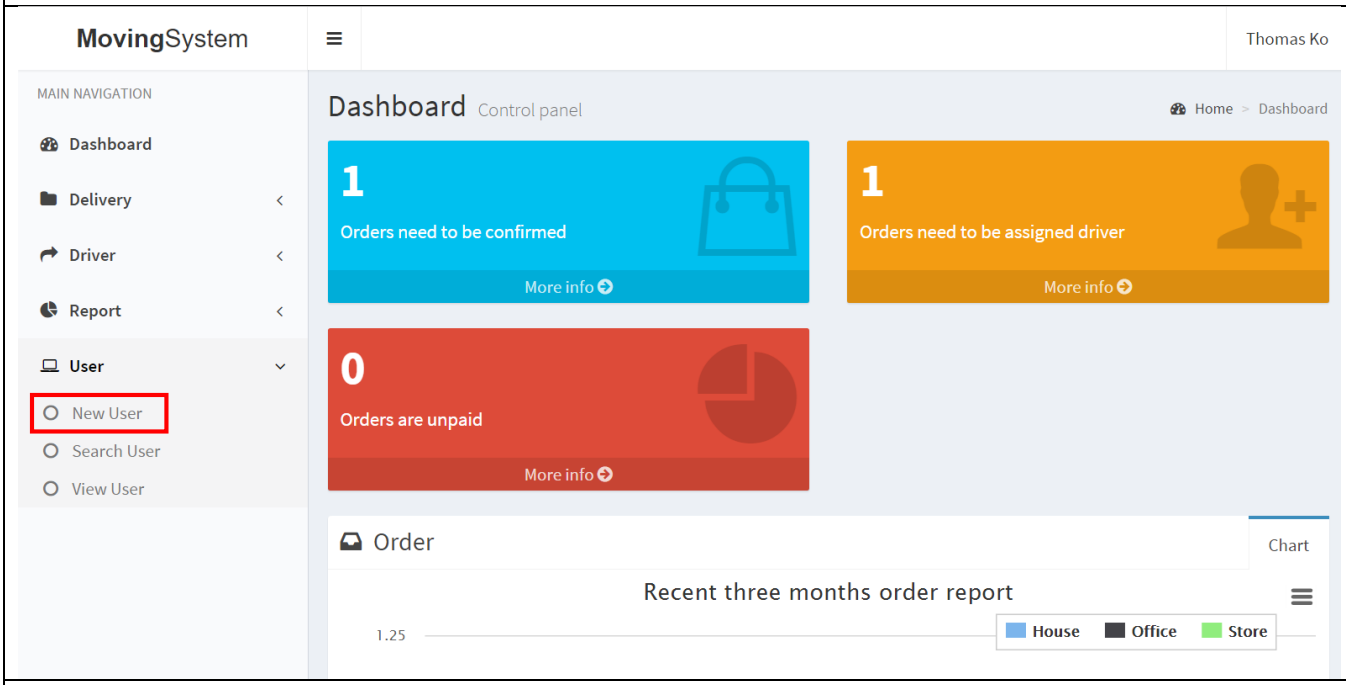


A2.2 Signing in the system as administrator
Step 1: In homepage, input the username and password. Step 2: The home page will be refreshed after login successfully.


A2.3 Signing out the system
Step 1: In homepage, Click “Username” in the top menu bar. Step 2: Click “Sign out”.


A2.4 Creating different user accounts

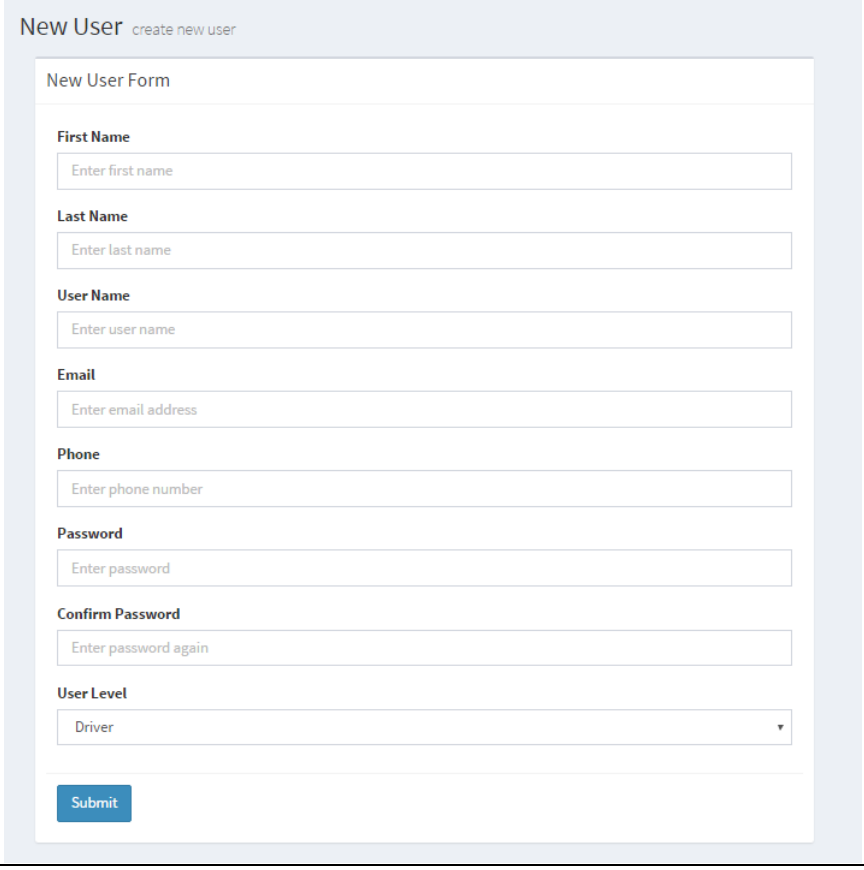
Step 1: Click “New User” in menu bar.



Step 2: Input user information.

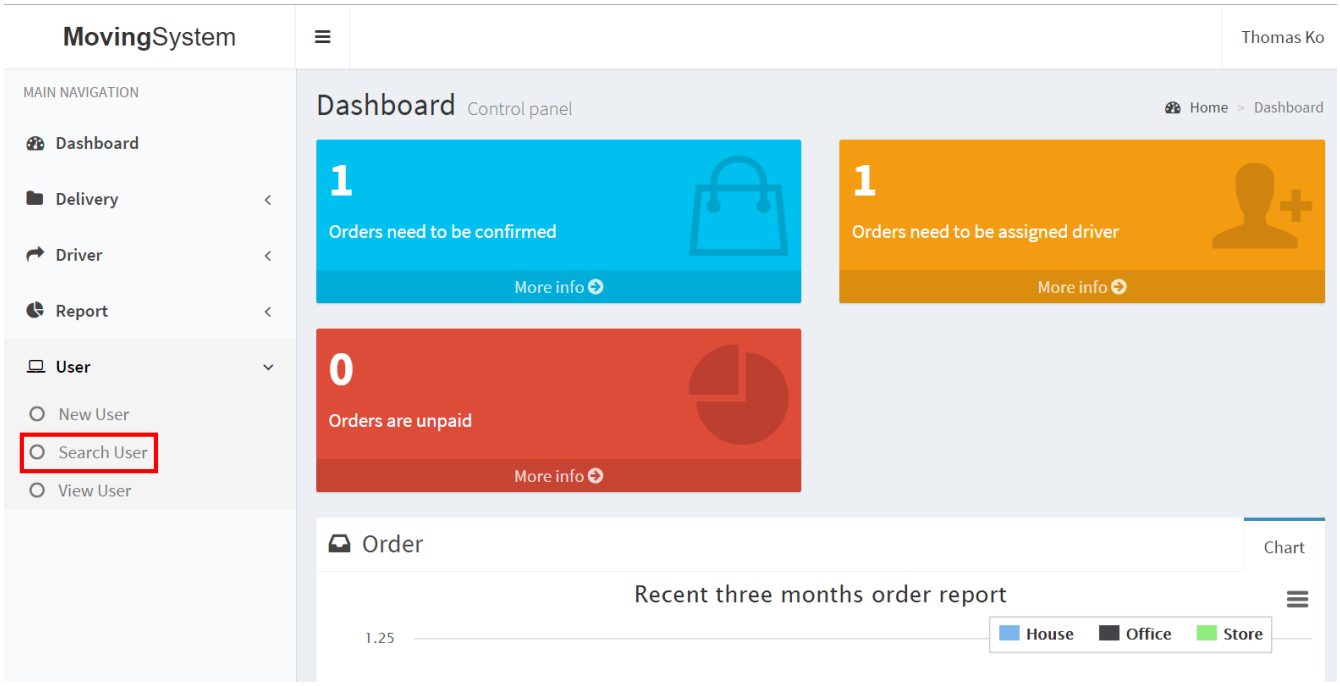
Step 3: Click “Submit”.

Step 4: Message: “User is created successfully” will be displayed if the user is created.

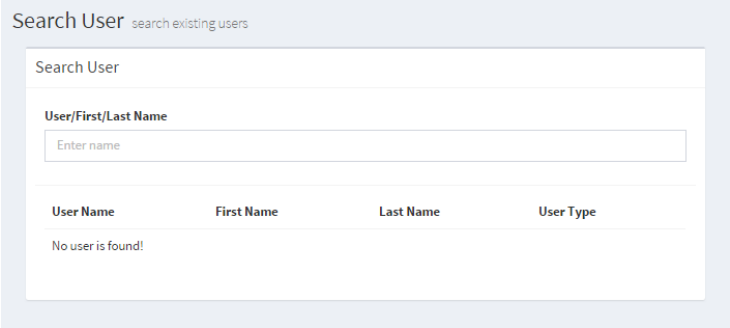


A2.5 Searching users by username, first name and last name

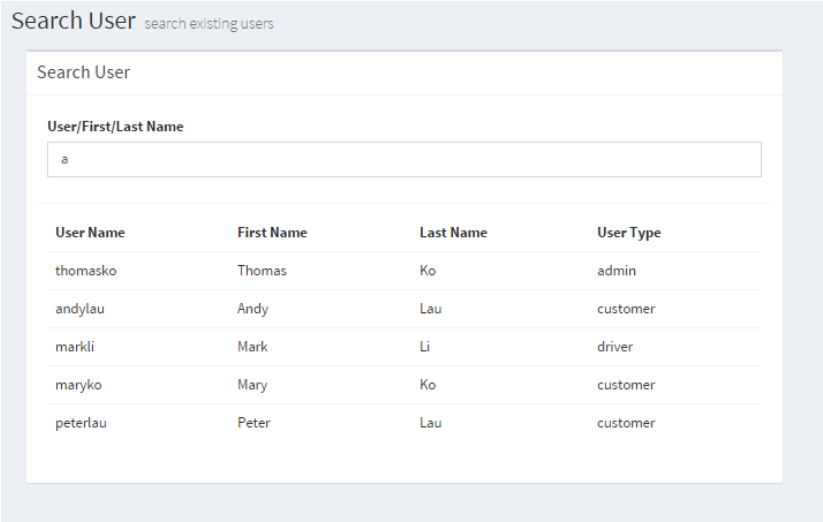
Step 1: Click “Search User” in menu bar.



Step 2: Input any user name, first name or last name for searching related users.

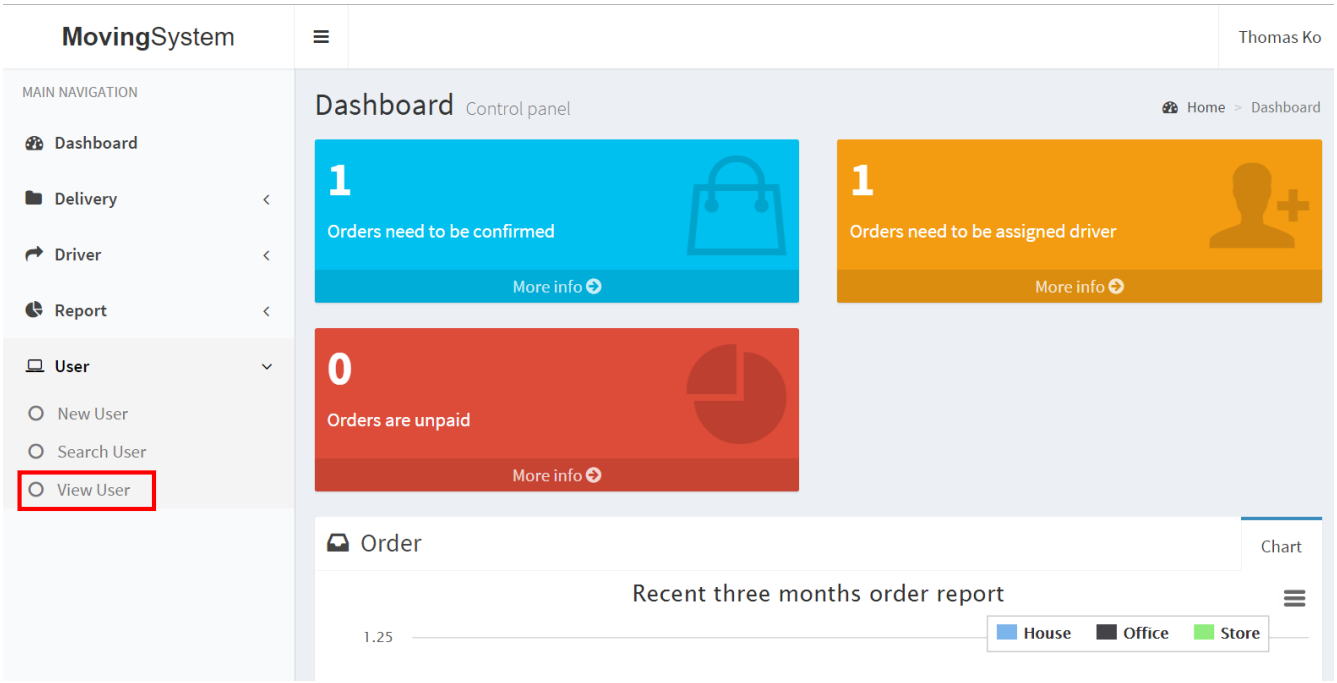


Step 3: Select specific user for reviewing or editing the profile.

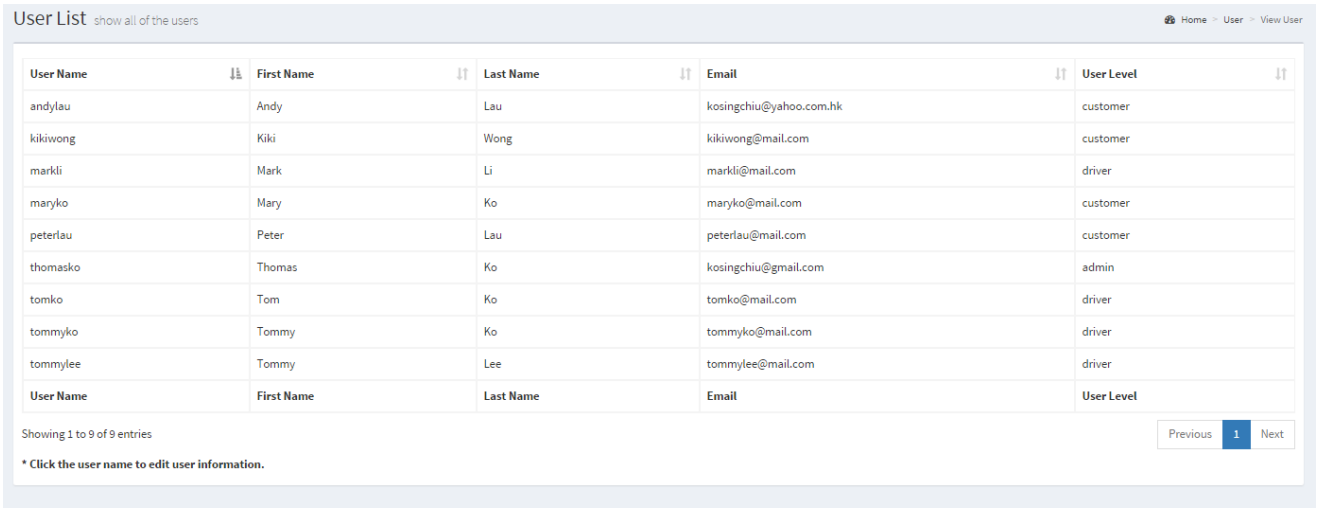


A2.6 Viewing user list

Step 1: Click “View User” in menu bar.



Step 2: Select specific user for reviewing or editing the profile.



User List show all of the users

Home > User > View User

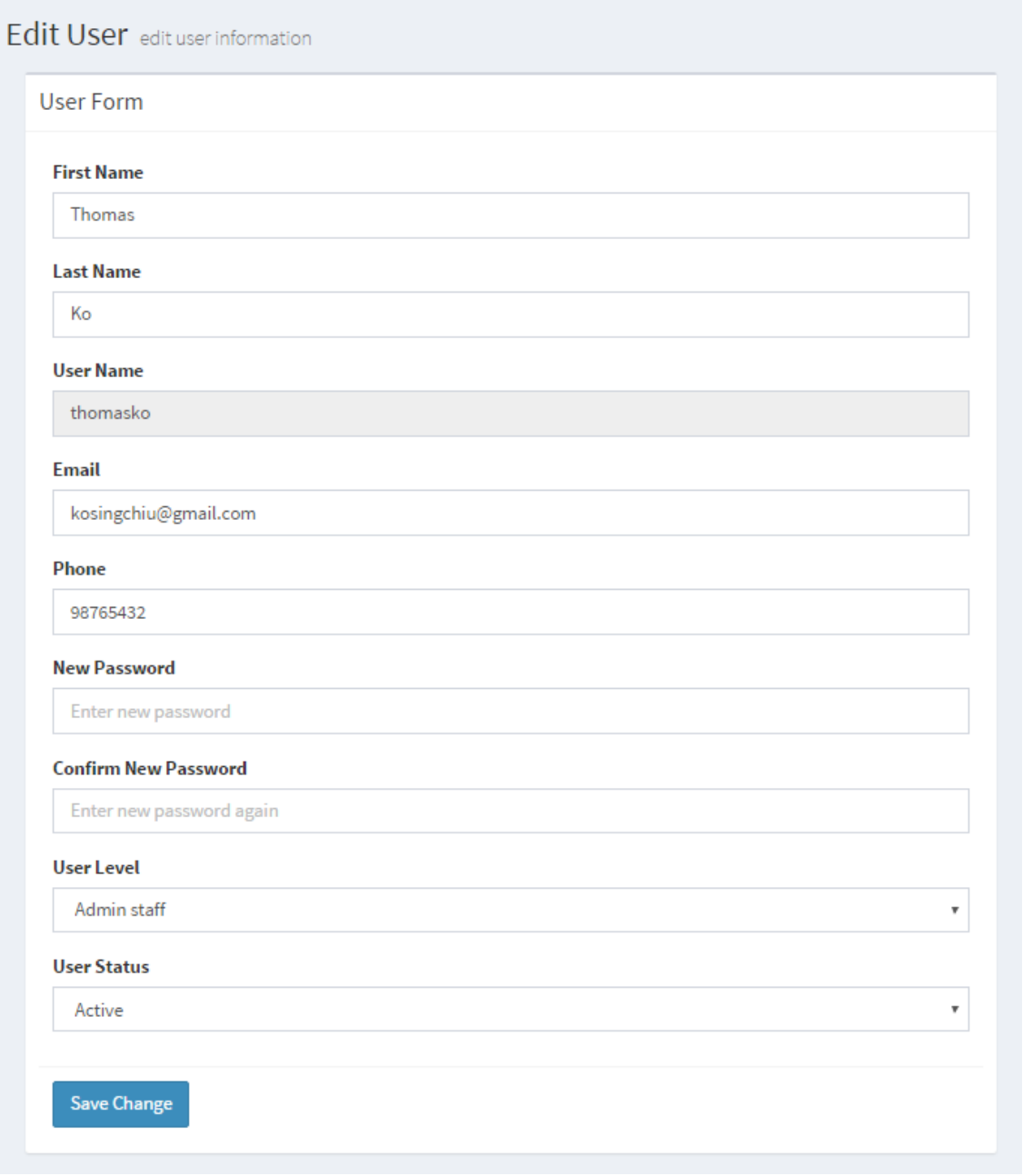
User Name	First Name	Last Name	Email	User Level
andylau	Andy	Lau	kosingchiu@yahoo.com.hk	customer
kikiwong	Kiki	Wong	kikiwong@mail.com	customer
markli	Mark	Li	markli@mail.com	driver
maryko	Mary	Ko	maryko@mail.com	customer
peterlau	Peter	Lau	peterlau@mail.com	customer
thomasko	Thomas	Ko	kosingchiu@gmail.com	admin
tomko	Tom	Ko	tomko@mail.com	driver
tommyko	Tommy	Ko	tommyko@mail.com	driver
tommylee	Tommy	Lee	tommylee@mail.com	driver
User Name	First Name	Last Name	Email	User Level

Showing 1 to 9 of 9 entries

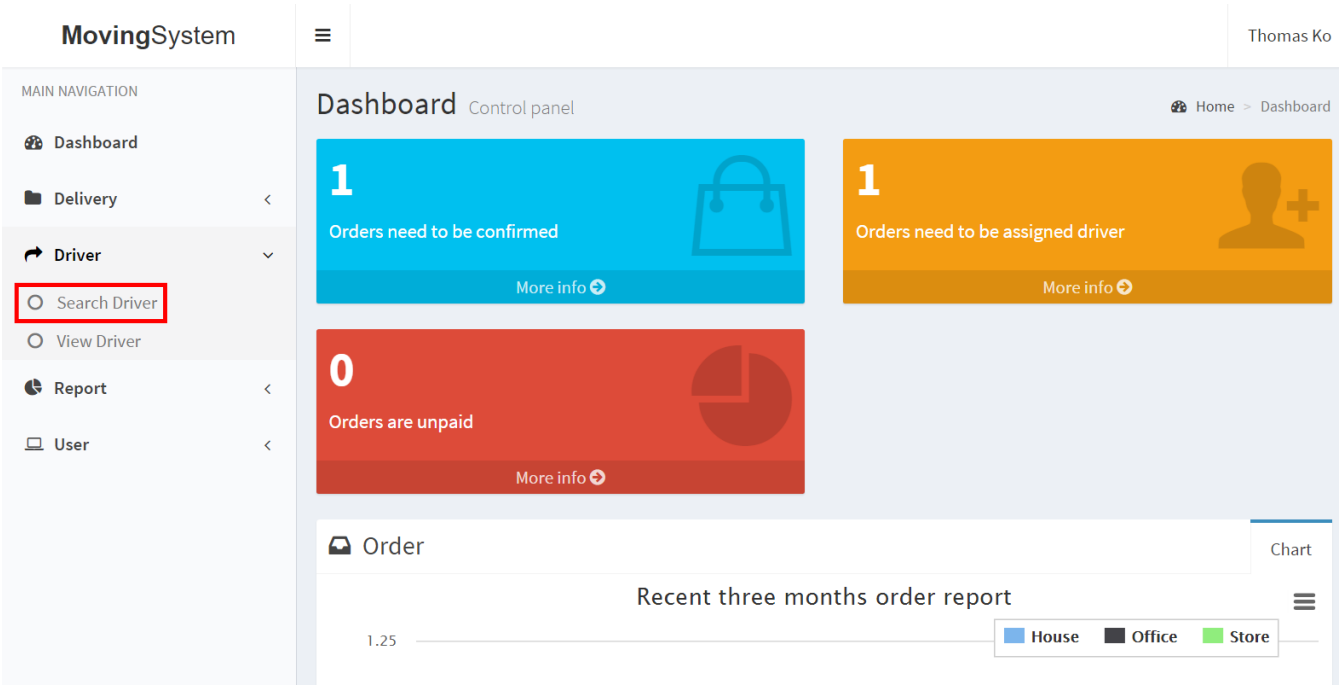
Previous 1 Next

* Click the user name to edit user information.

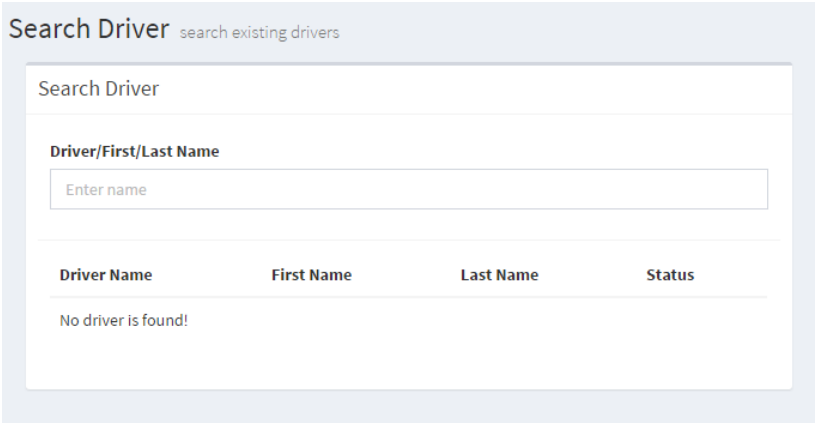
A2.7 Editing user profile
Step 1: Select specific user in the user list generated by “View User” or “Search User”. Step 2: Edit the user information. Step 3: Click “Save Change”. Step 4: Message: “User is updated successfully” will be displayed if the user is updated.



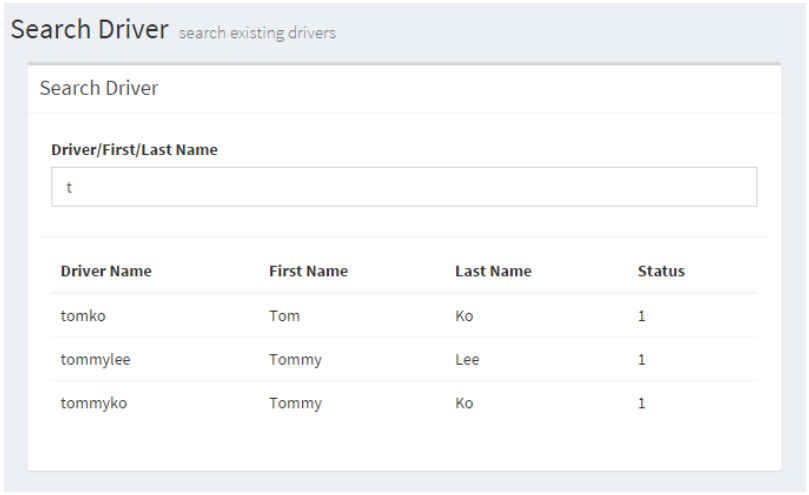
A2.8 Searching driver by username, first name and last name
Step 1: Select “Search Driver” in menu bar.



Step 2: Input driver user name, first name or last name for searching related drivers.

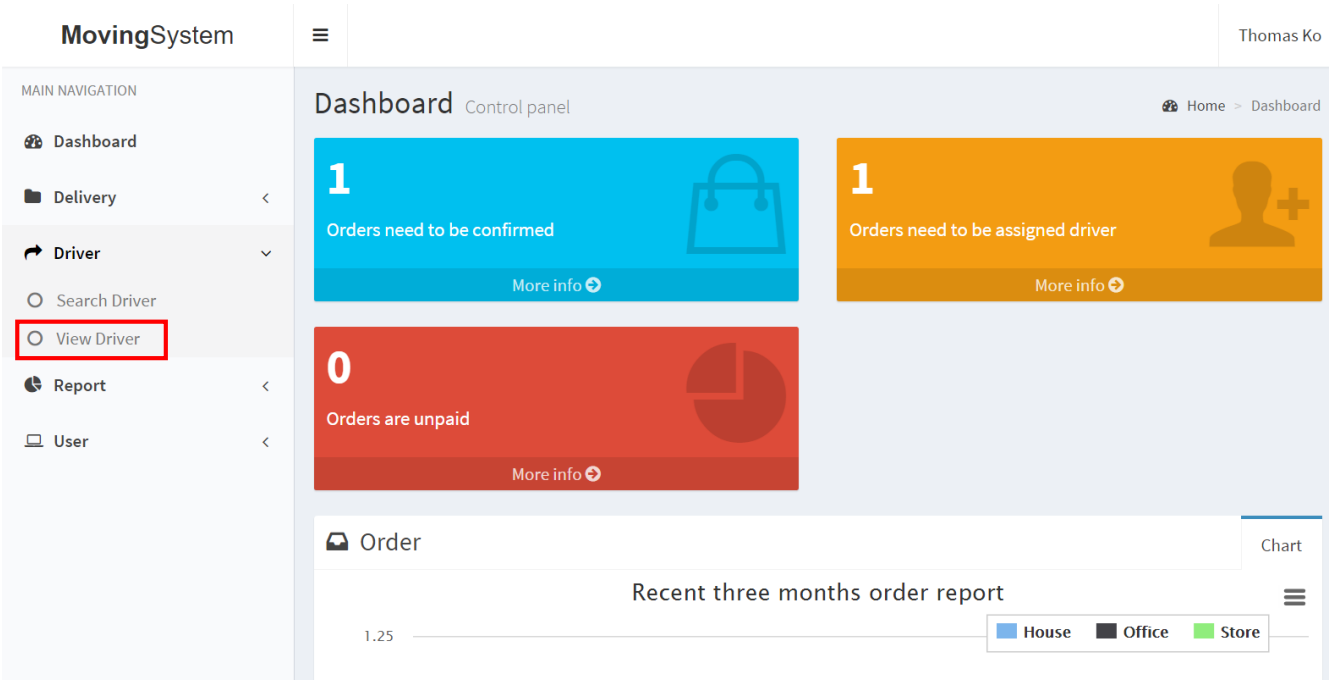


Step 3: Select specific driver for reviewing or editing the profile.

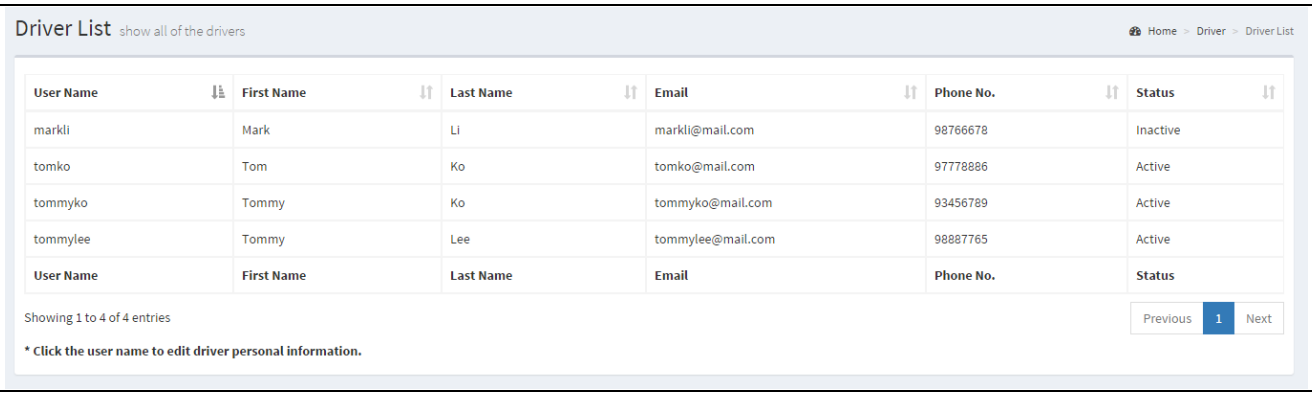

Search Driver
search existing drivers
Search Driver
Driver/First/Last Name
t
Driver Name
First Name
Last Name
Status
tomko
Tom
Ko
1
tommylee
Tommy
Lee
1
tommyko
Tommy
Ko
1

A2.9 Viewing driver list

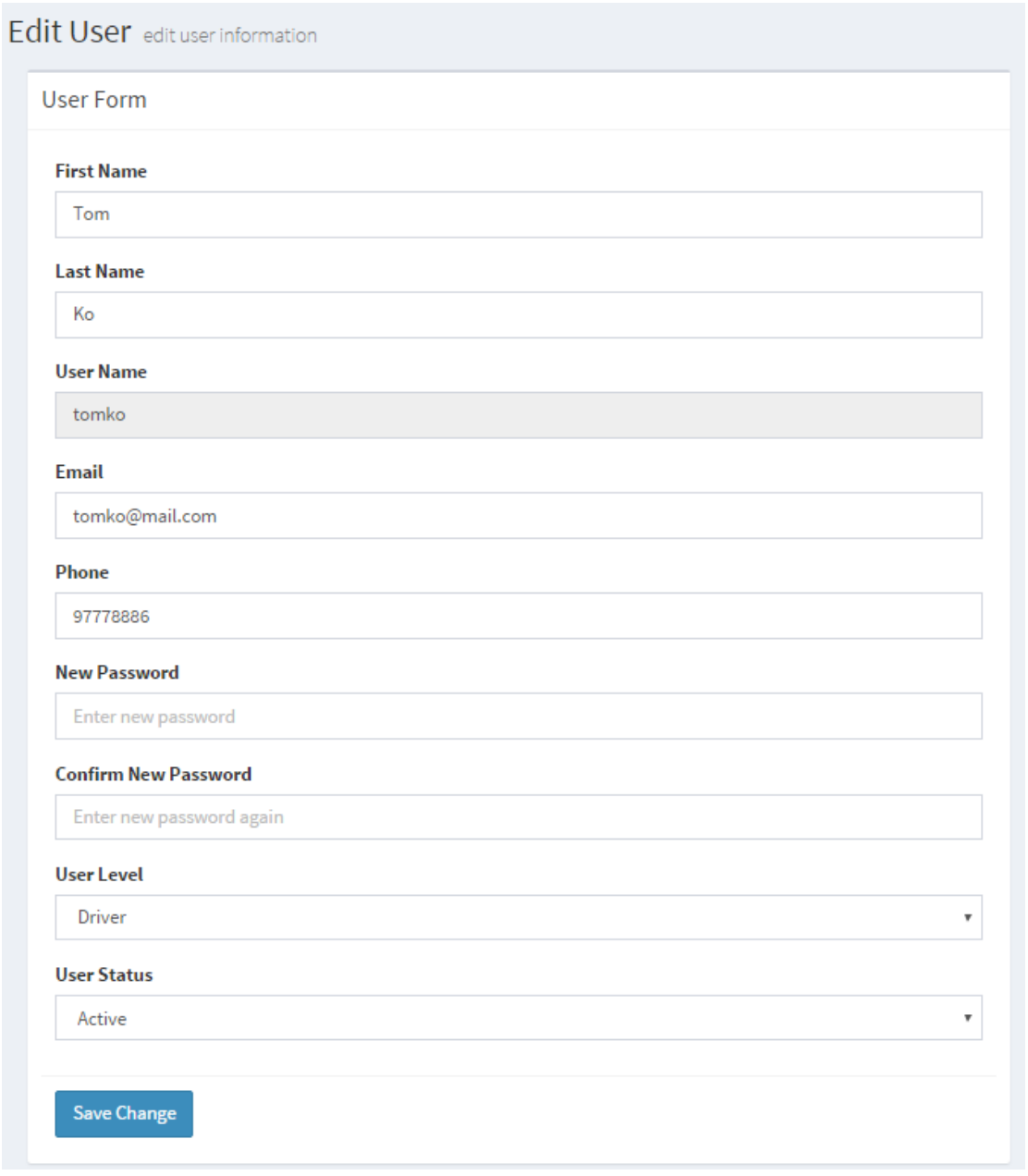
Step 1: Select “View Driver” in menu bar.



Step 2: Select specific driver for reviewing or editing the profile.

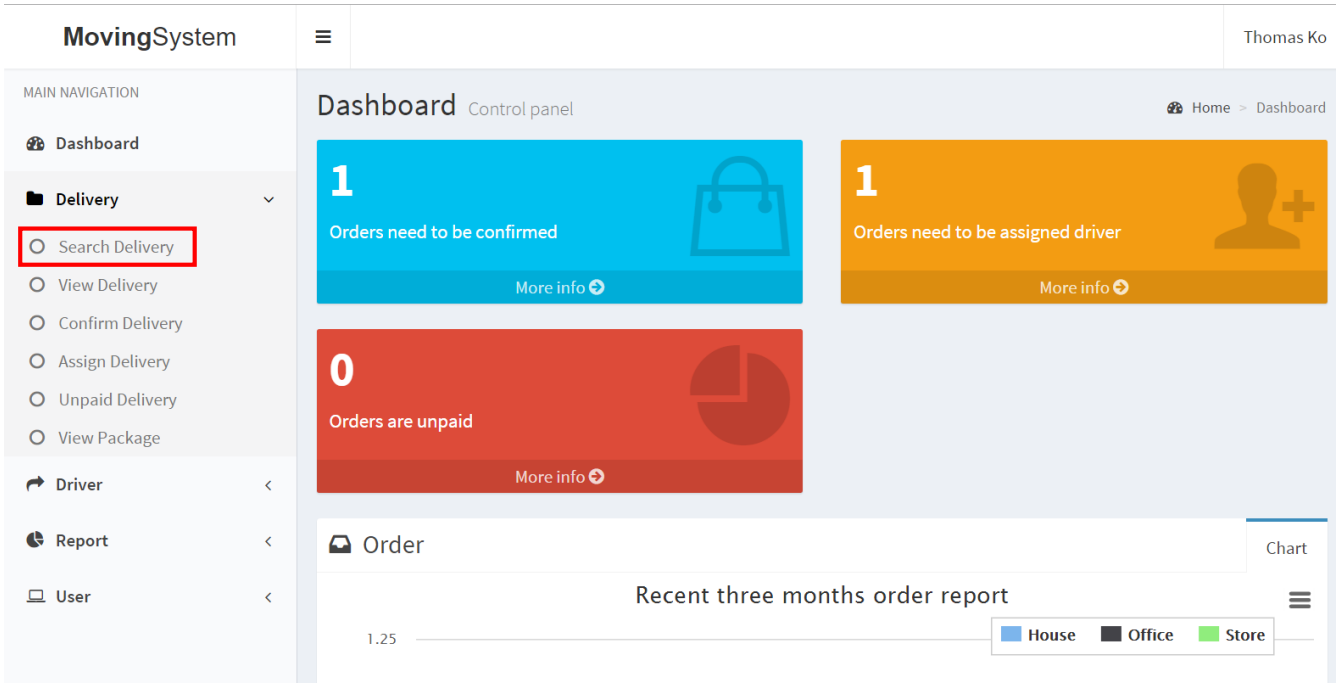


A2.10 Editing driver profile
Step 1: Select specific driver in the driver list generated by “View Driver” or “Search Driver”. Step 2: Edit the driver information. Step 3: Click “Save Change”. Step 4: Message: “User is updated successfully” will be displayed if the driver is updated.



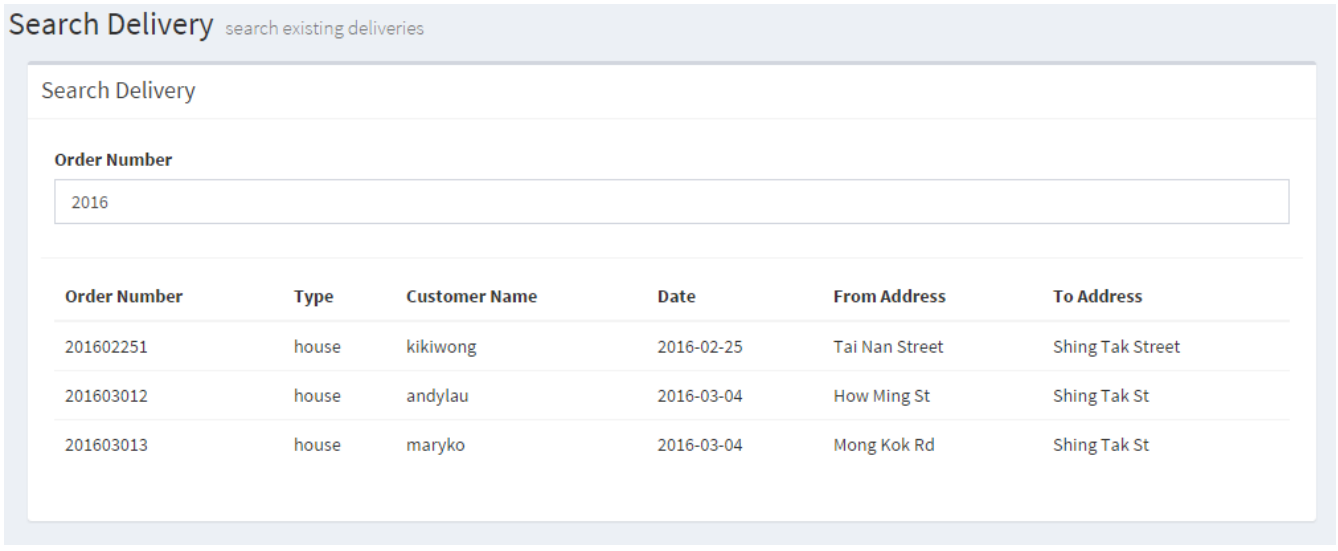
A2.11 Searching delivery by order number

Step 1: Click “Search Delivery” in menu bar.



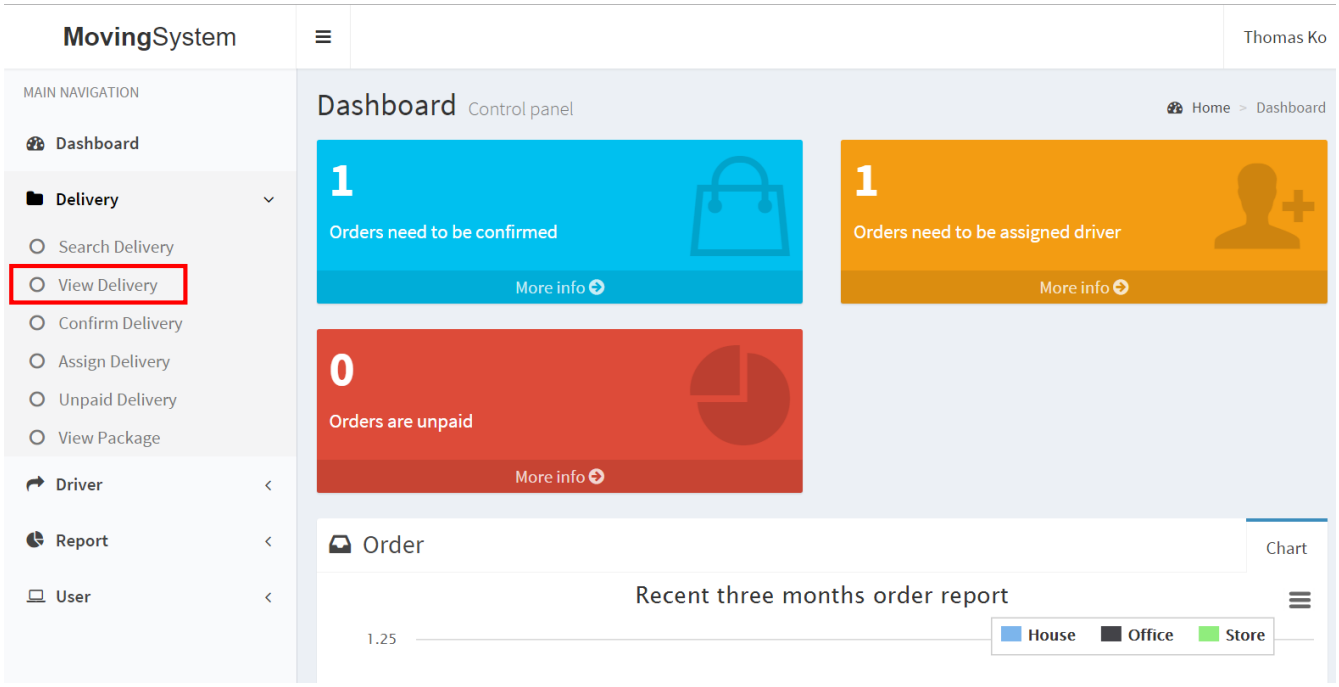
Step 2: Input order number for searching related orders.

Step 3: Select specific order for reviewing or edit the order details.

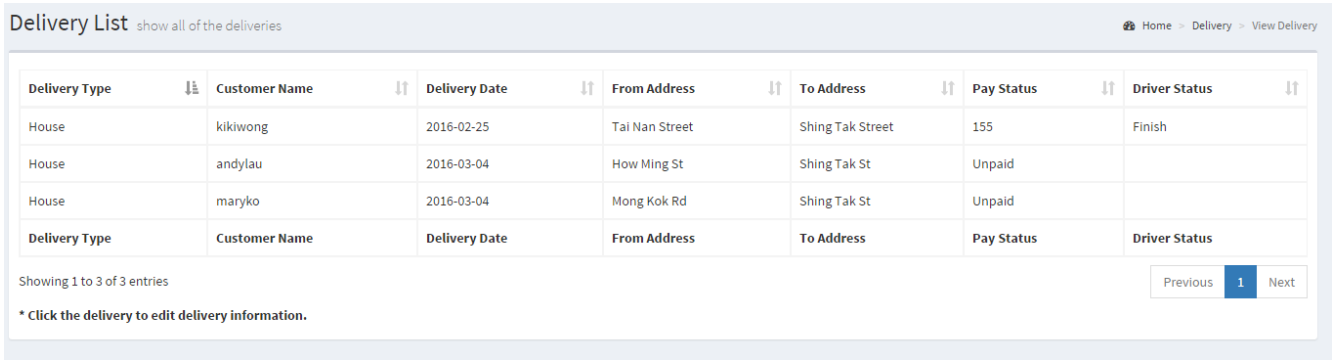


A2.12 Viewing general delivery list

Step 1: Click “View Delivery” in menu bar.

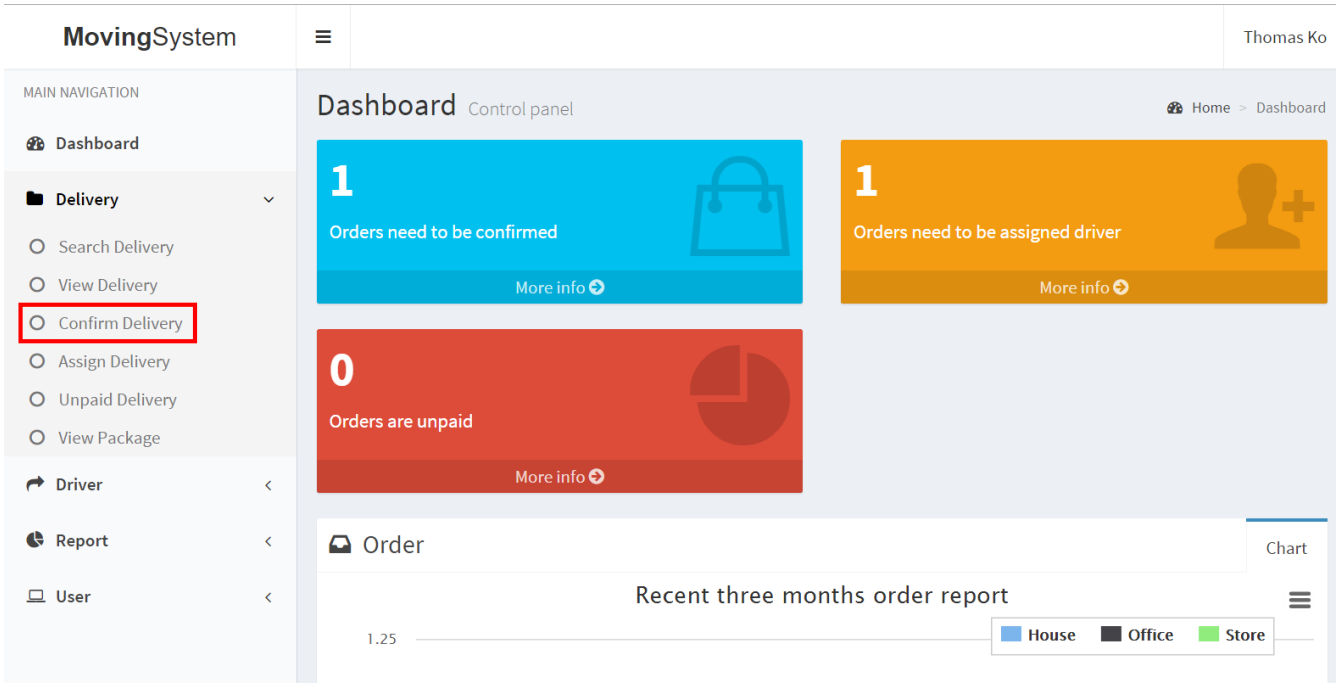


Step 2: Select specific order for reviewing or edit the order details.

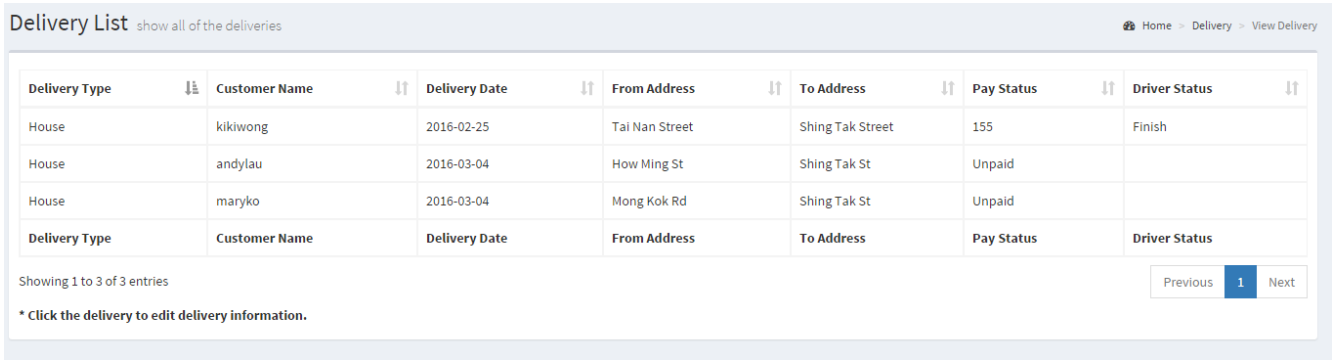


A2.13 Confirming delivery

Step 1: Click “Confirm Delivery” in menu bar.

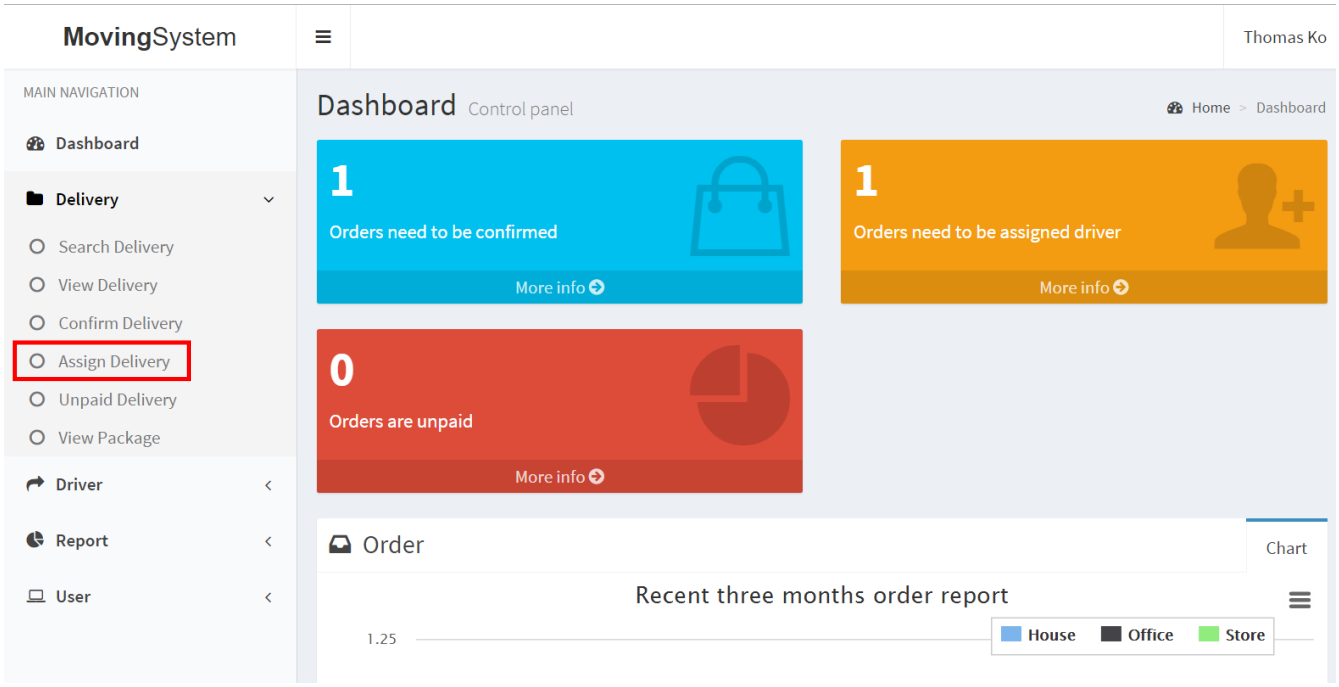


Step 2: Select specific order for confirming the order details.

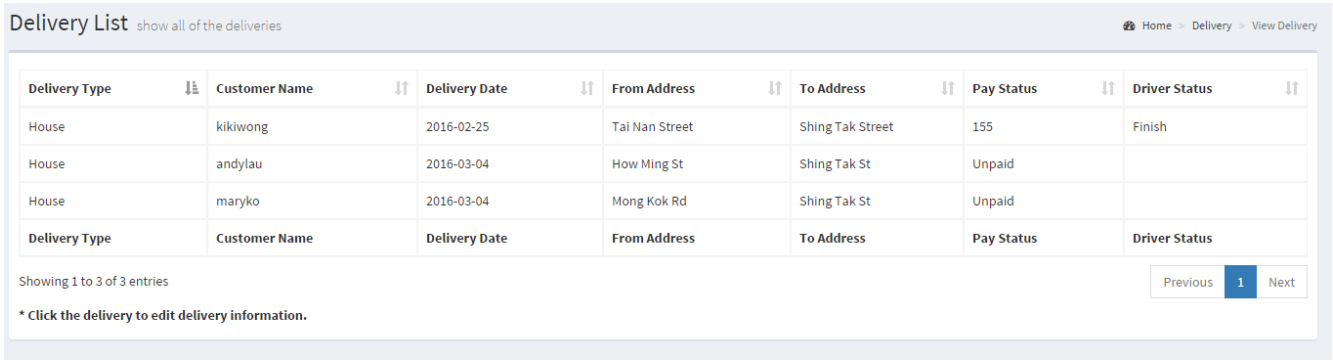


A2.14 Assigning driver to delivery

Step 1: Click “Assign Delivery” in menu bar.

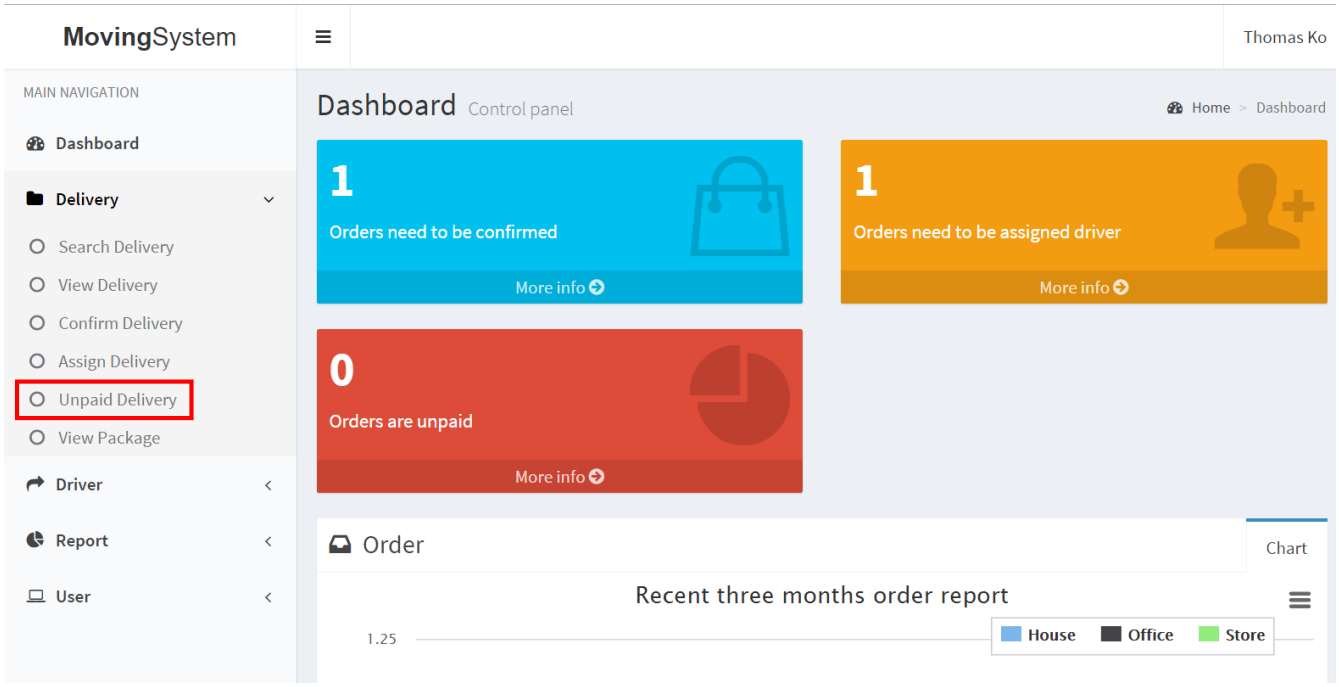


Step 2: Select specific order for assigning drivers.

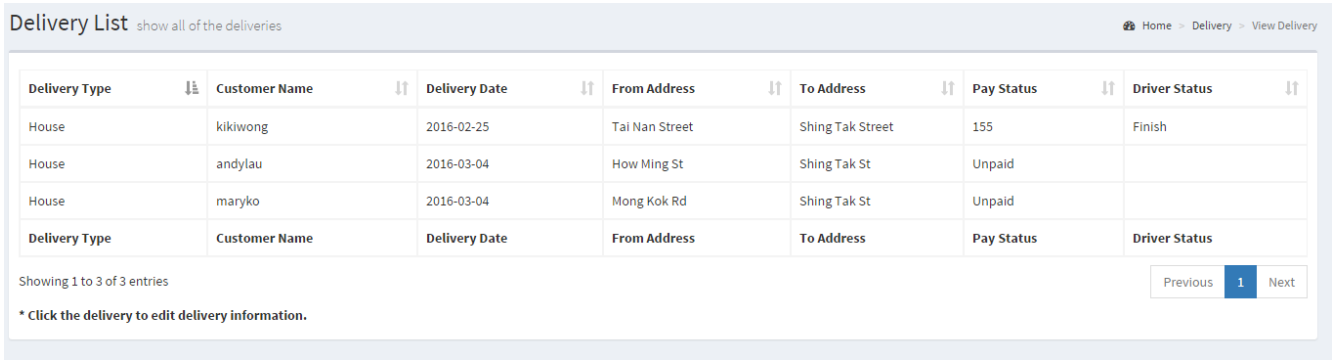


A2.15 Viewing unpaid delivery list

Step 1: Click “Unpaid Delivery” in menu bar.



Step 2: Select specific order for reviewing the details of the upaid order.



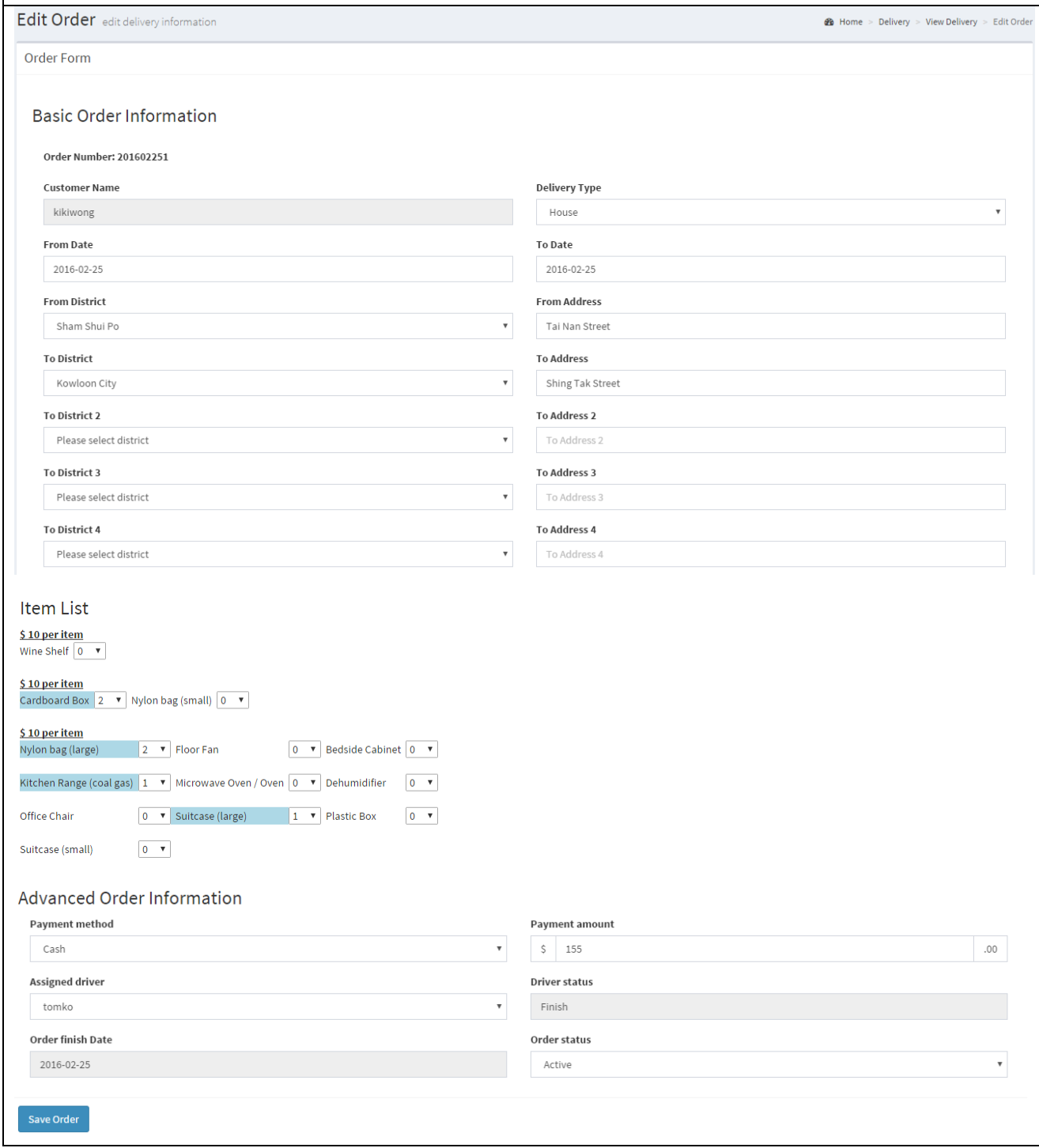
A2.16 Editing delivery details

Step 1: Select specific order in the delivery list generated by “Search Delivery”, “View Delivery”, “Confirm Delivery”, “Assign Delivery” or “Unpaid Order”.

Step 2: Edit the order information.

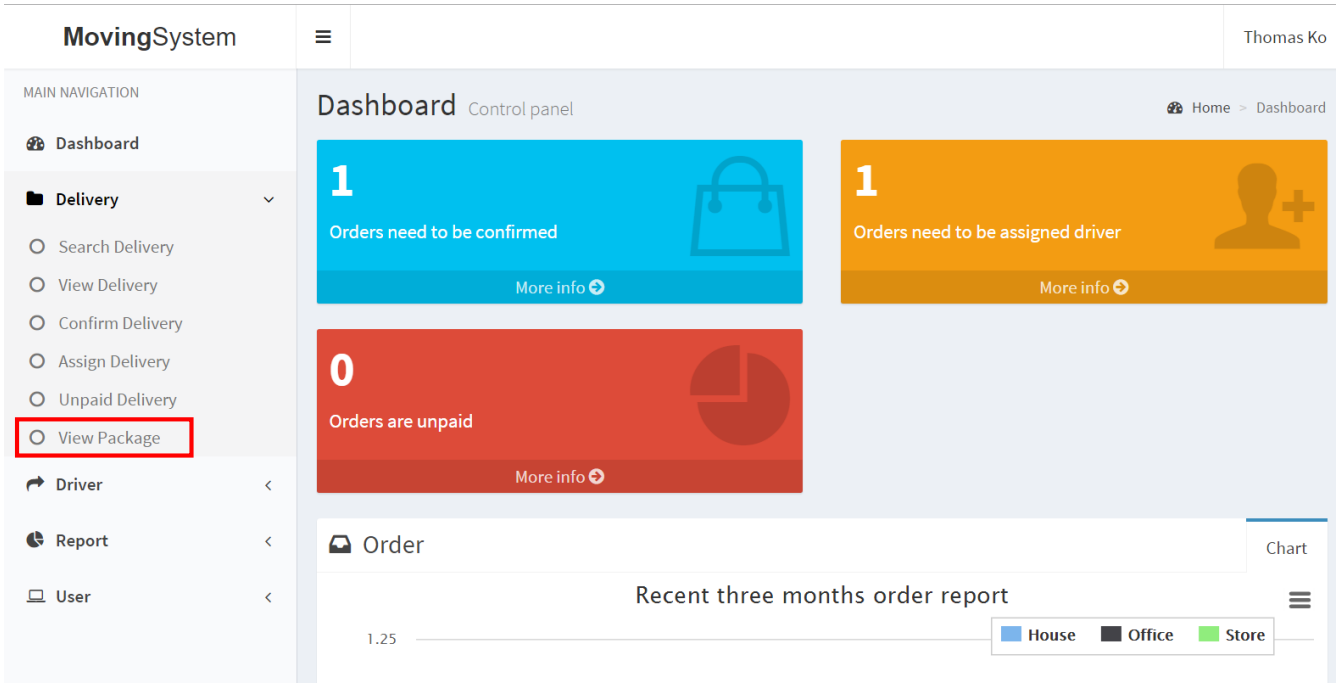
Step 3: Click “Save Order”.

Step 4: Message: “Order is updated successfully” will be displayed if the order is updated.

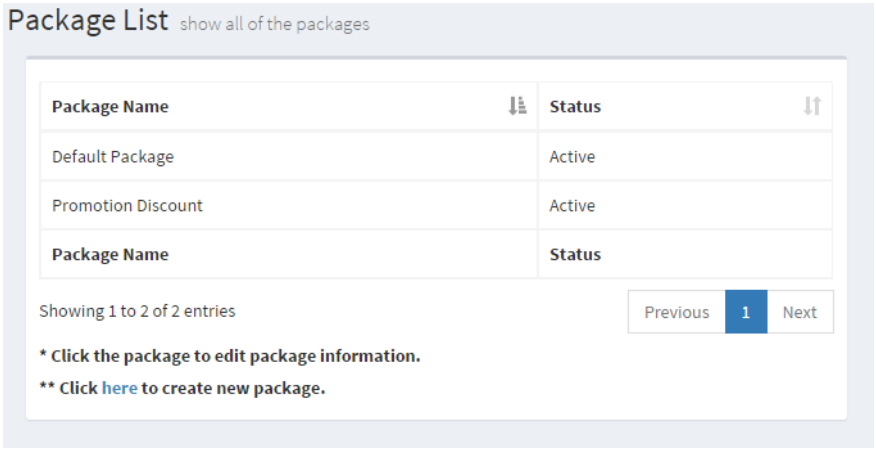


A2.17 Viewing order package list

Step 1: Select “View package” in menu bar.



Step 2: Select specific package for reviewing or editing package details.



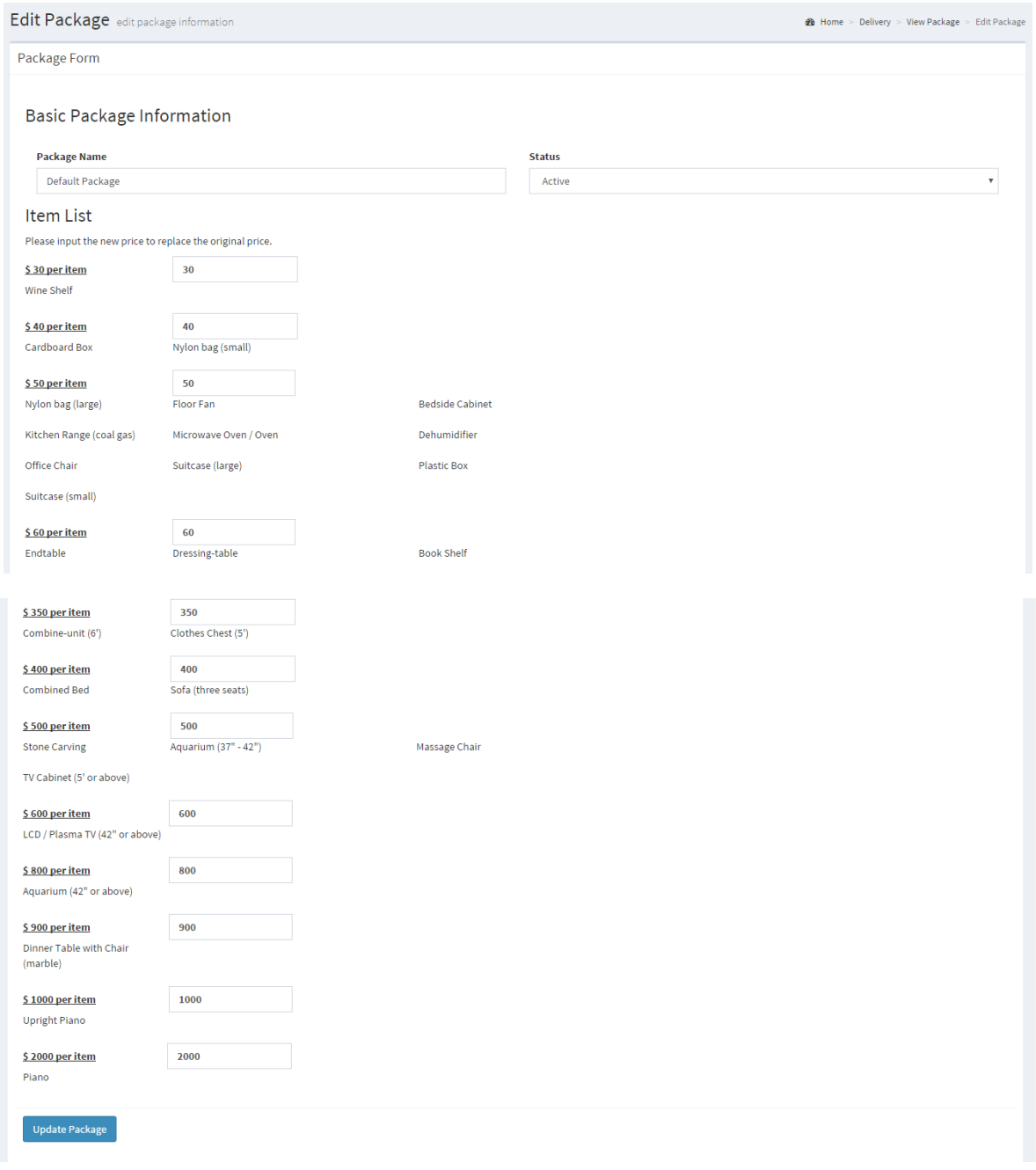
A2.18 Editing order package details

Step 1: Select specific package in the package list generated by “View Package”.

Step 2: Edit package details.

Step 3: Click “Update Package”.

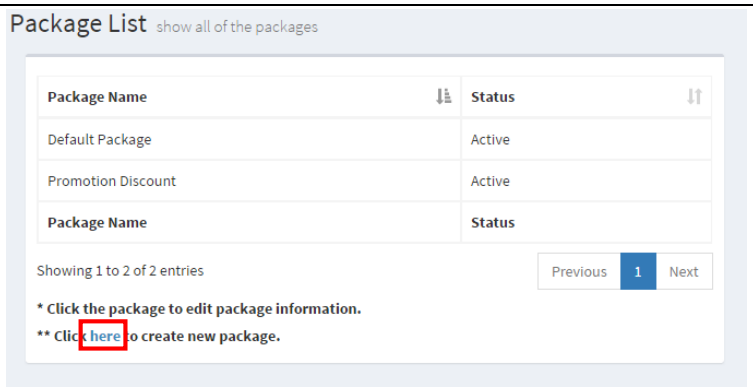
Step 4: Message: “Package is updated successfully” will be displayed if the package is updated.



A2.19 Creating order package with new price

Step 1: Click “View Package” in menu bar for getting the package list.

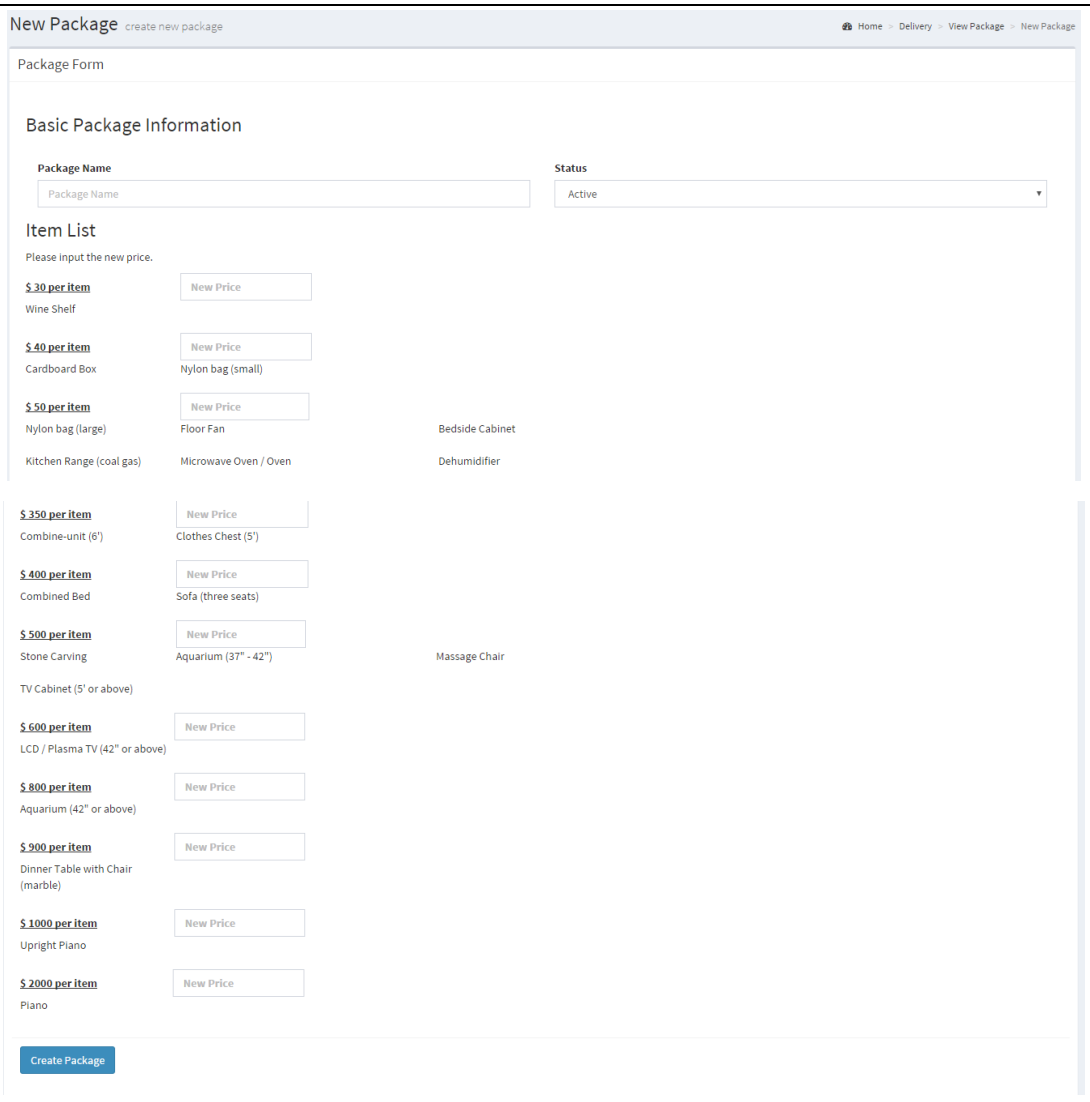
Step 1: Click “here” for redirecting to the package form.



Step 3: Input package information.

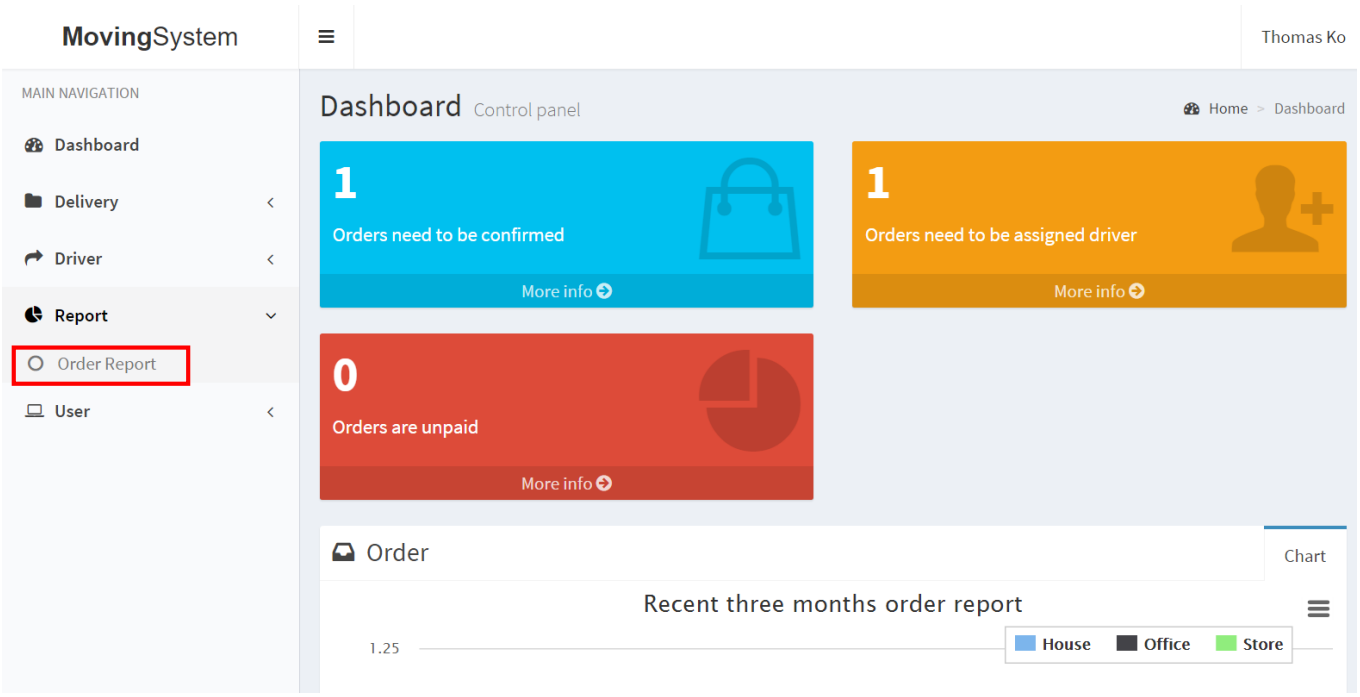
Step 4: Click “Create Package”

Step 5: Message: “Package is created successfully” will be displayed if the package is created.

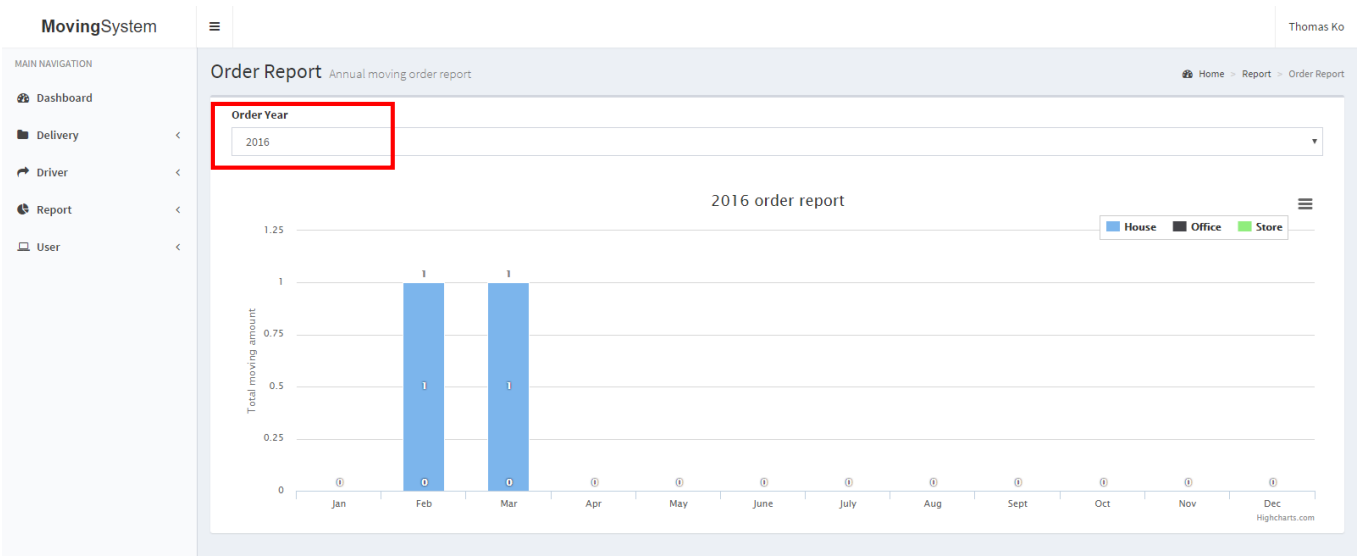


A2.20 Viewing transaction report

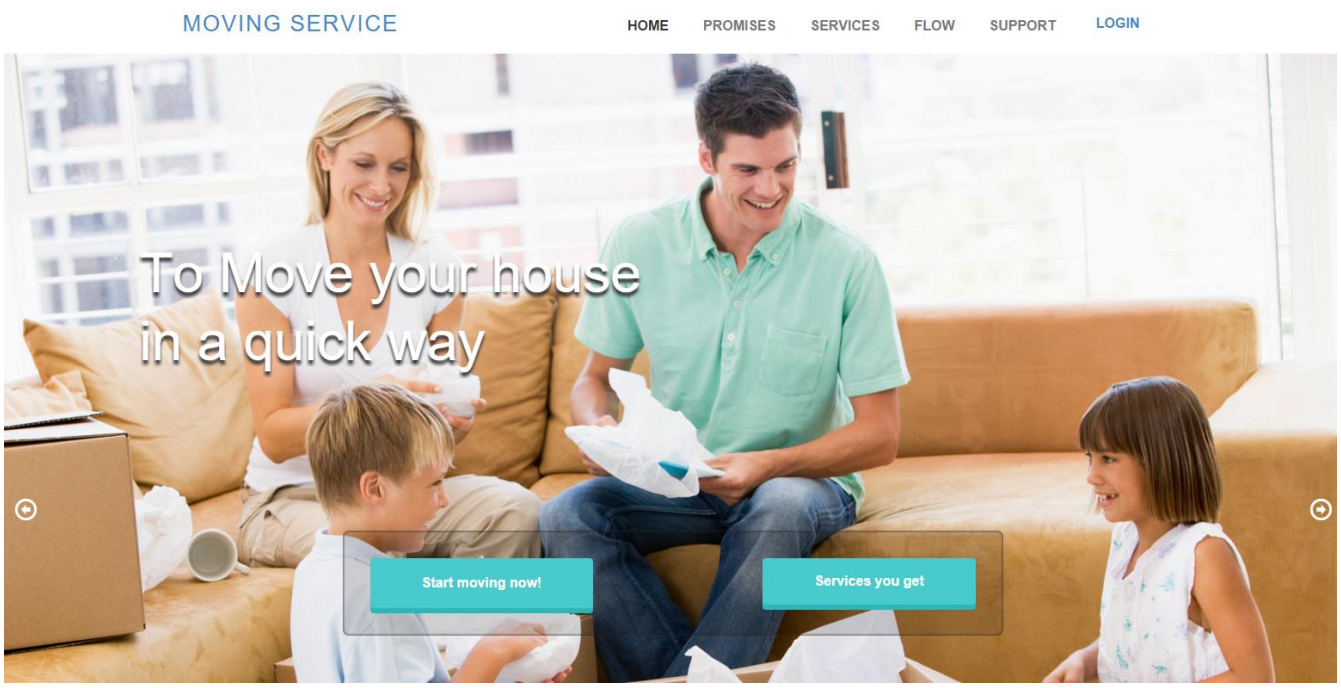
Step 1: Click “View Report” in menu bar.

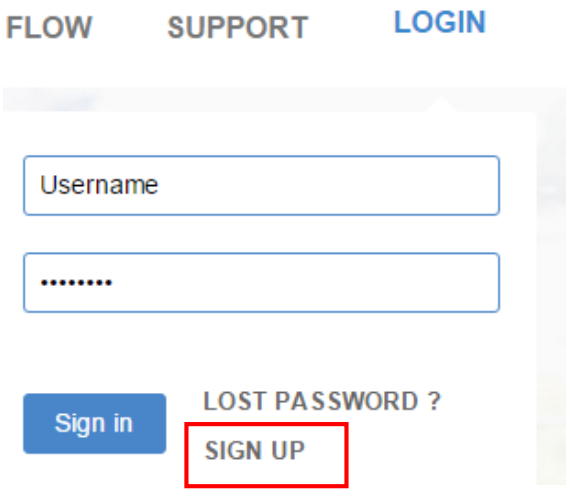
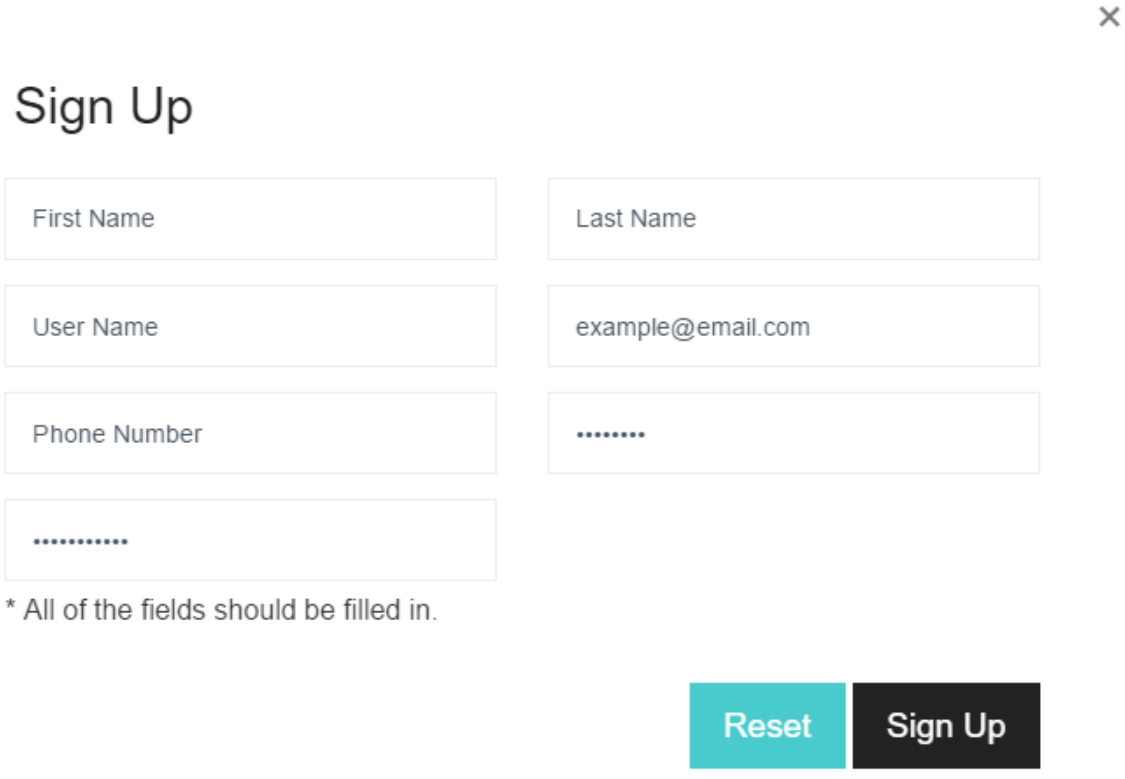


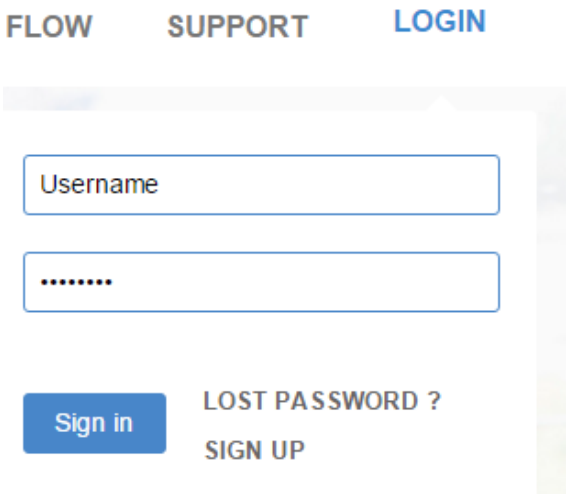
Step 2: Select suitable Order Year for generating corresponding report.

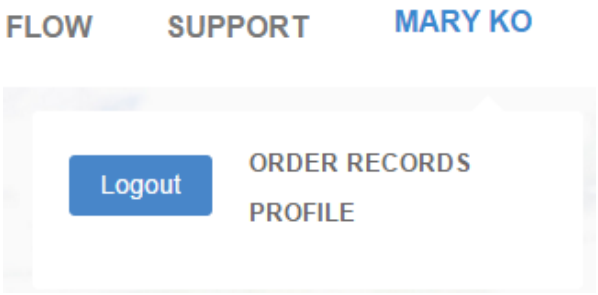


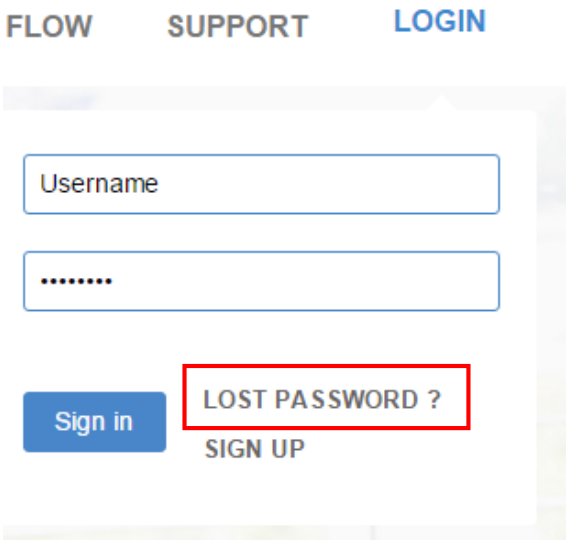
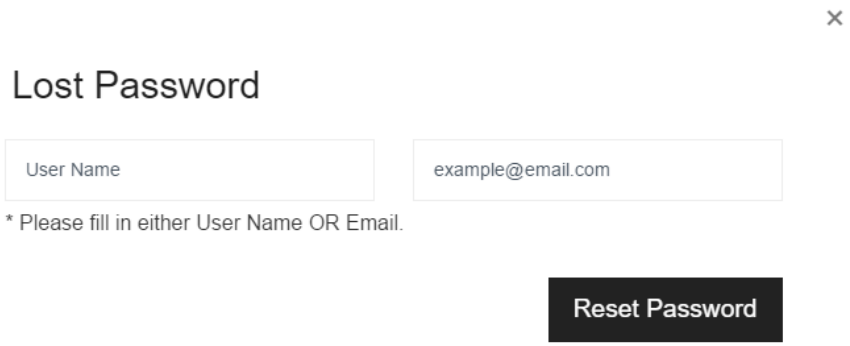
A3. User Manual – Customer

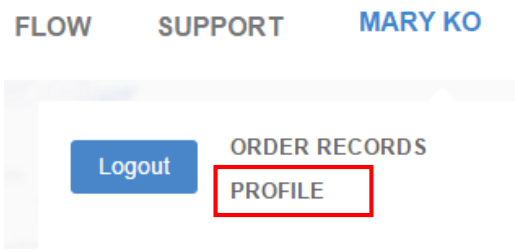
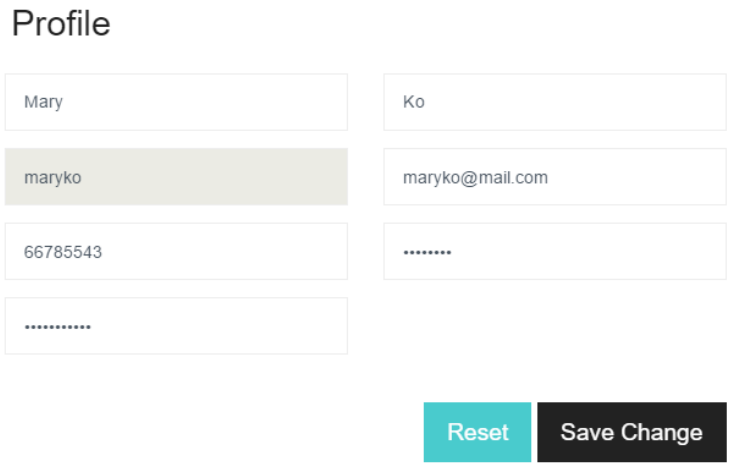
A3.1 About homepage


A3.2 Signing up a customer
Step 1: Click “LOGIN” in menu bar. Step 2: Click “SIGN UP”.

Step 3: Input user information. Step 4: Click “Sign Up”.


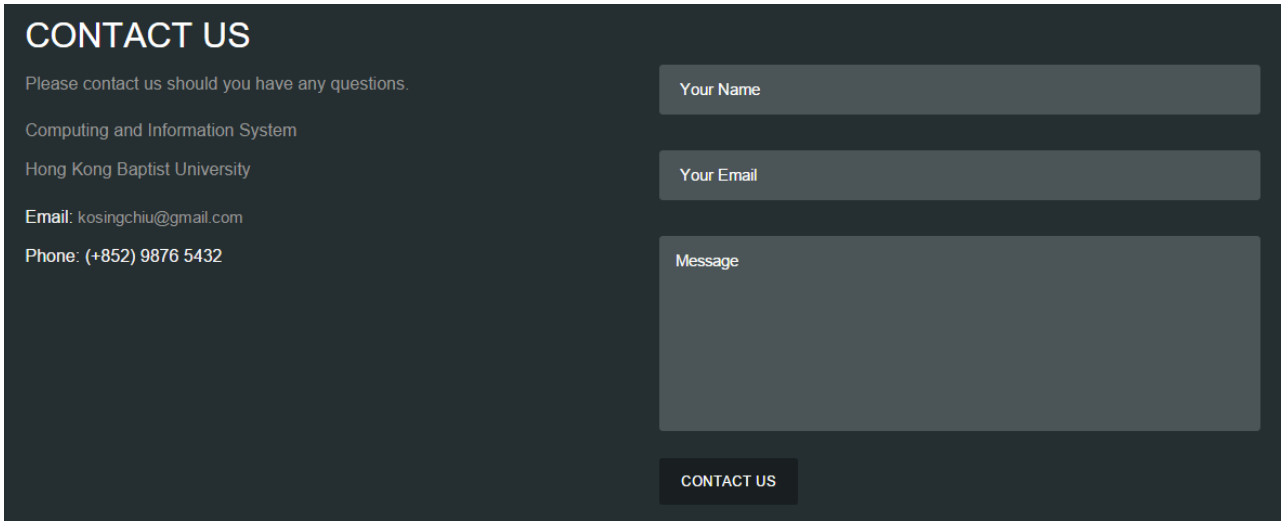
A3.3 Signing in the system as customer
Step 1: Click “LOGIN” in menu bar. Step 2: Input username and password. Step 3: Click “Sign in”. Step 4: Message: “Welcome” will be displayed if both username and password are valid.


A3.4 Signing out the system
Step 1: Click “Username” in menu bar. Step 2: Click “Logout”.


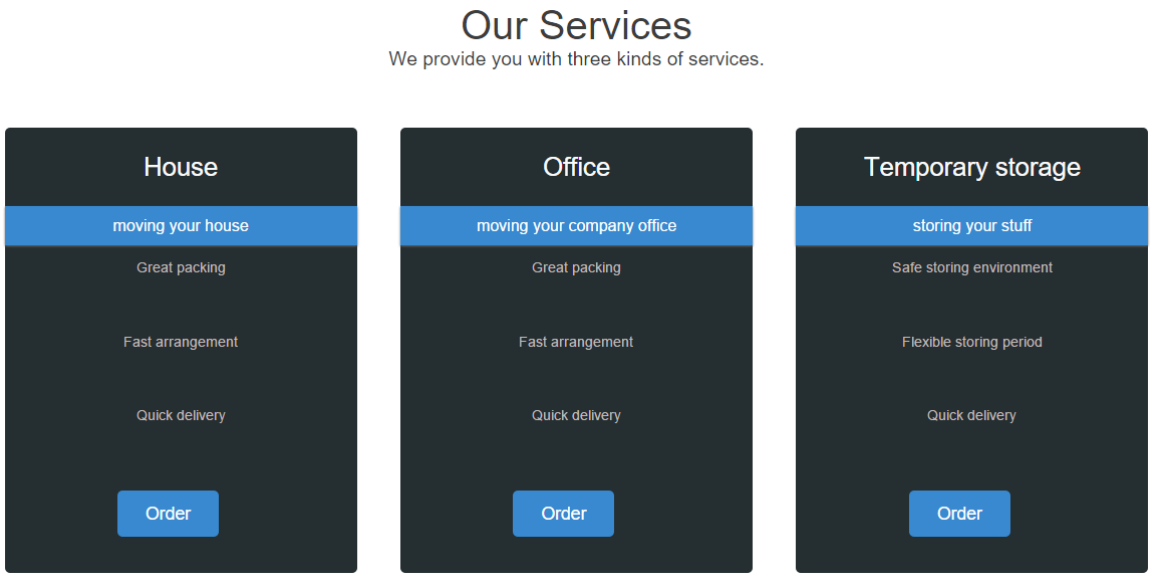
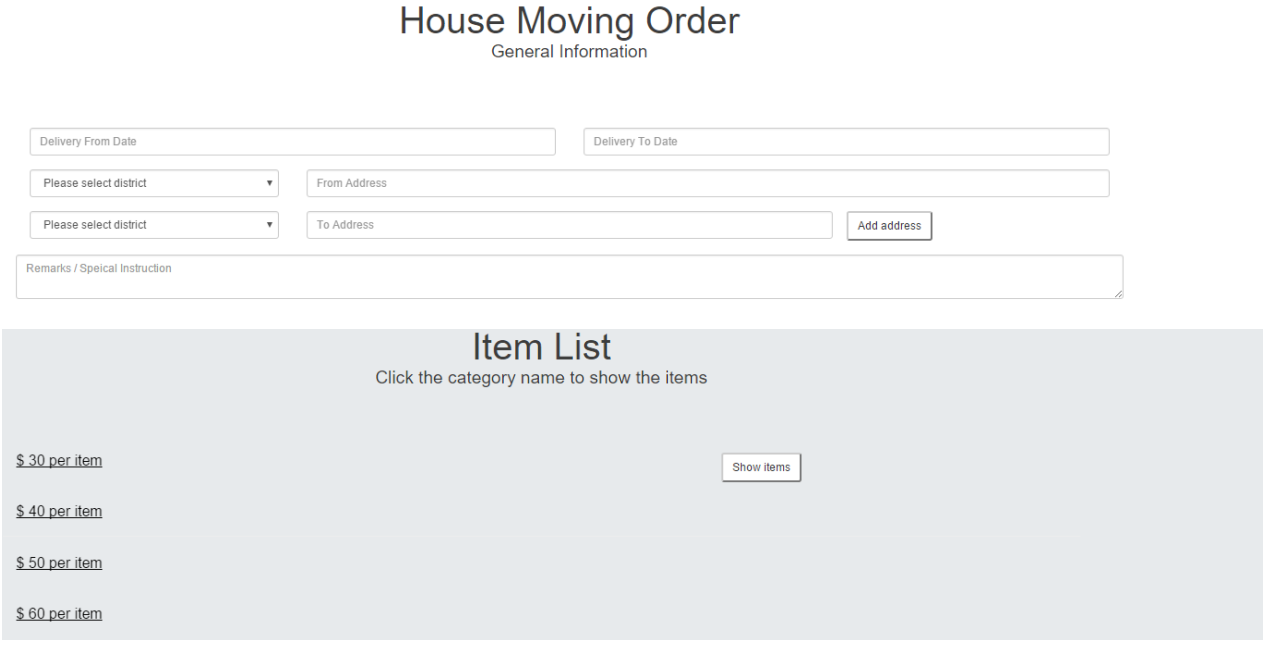
A3.5 Resetting password when losing it
Step 1: Click “LOGIN” in menu bar. Step 2: Click “LOST PASSWORD?”.

Step 3: Input either username or email address. Step 4: Click “Reset Password”. Step 5: Message: “Password is reset” will be displayed if the password is reset. Step 6: Check user mail box for getting the new password.


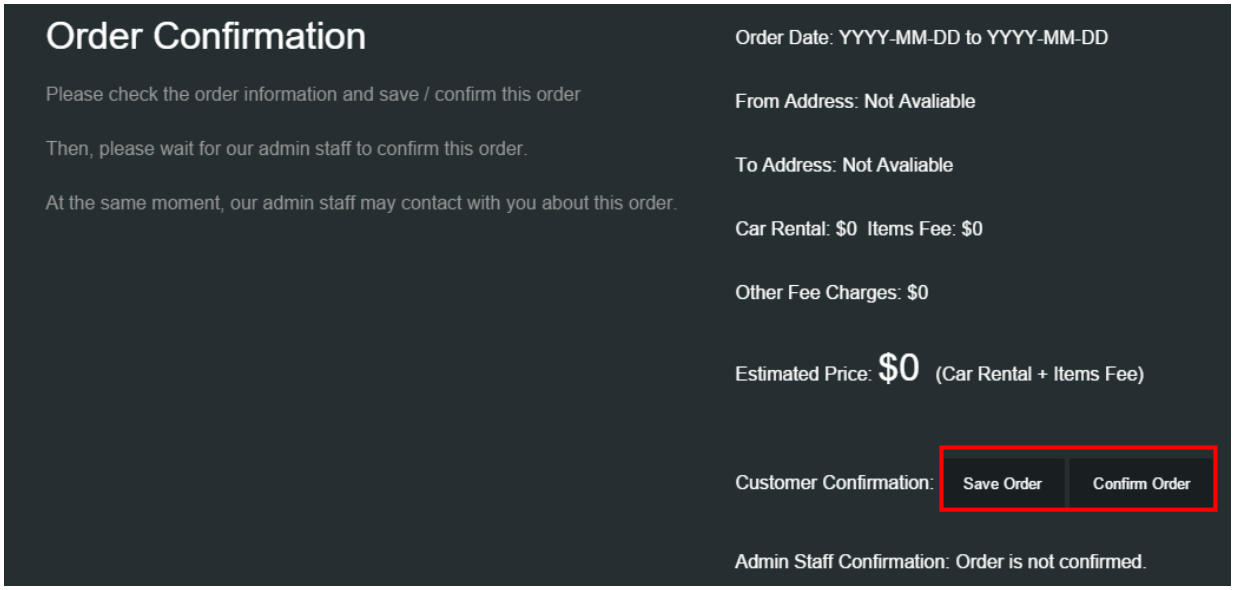
A3.6 Editing customer profile
Step 1: Click “Username” in menu bar. Step 2: Click “PROFILE”.

Step 3: Edit user profile. Step 4: Click “Save Change”. Step 5: Message: “Profile is updated successfully” will be displayed if the profile is updated.


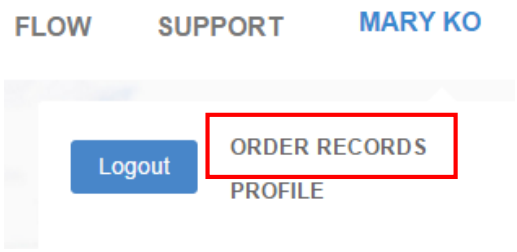
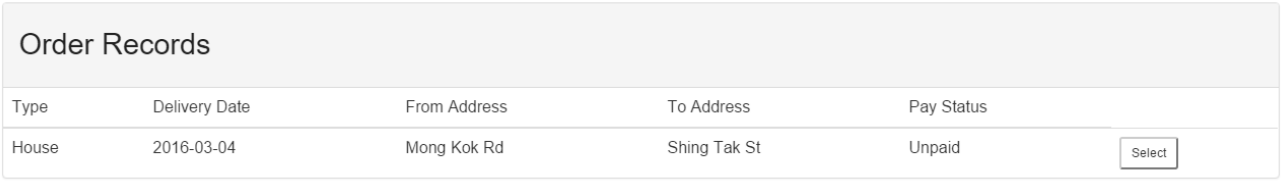
A3.7 Sending email to admin staff
Step 1: Click “SUPPORT” in menu bar.

Step 2: Input customer name, email address and message. Step 3: Click “CONTACT US” for sending the email to the admin staff. Step 4: Message: “Email is sent successfully” will be displayed if the email is sent.


A3.8 Creating different types of order
Step 1: Click “SERVICES” in menu bar.

Step 2: Select order type and click “Order”.

Step 3: Input order information. Step 4: Click “Save Order” or “Confirm Order” Step 5: Message: “Order is created successfully” will be displayed if the order is created.



Order Confirmation
Please check the order information and save / confirm this order
Then, please wait for our admin staff to confirm this order.
At the same moment, our admin staff may contact with you about this order.
Order Date: YYYY-MM-DD to YYYY-MM-DD
From Address: Not Avaliable
To Address: Not Avaliable
Car Rental: $0 Items Fee: $0
Other Fee Charges: $0
Estimated Price: $0 (Car Rental + Items Fee)
Customer Confirmation:
Save Order
Confirm Order
Admin Staff Confirmation: Order is not confirmed.

A3.9 Viewing customer order records
Step 1: Click “Username” in menu bar. Step 2: Click “ORDER RECORDS”.

Step 3: Select specific order for reviewing or editing the order details.


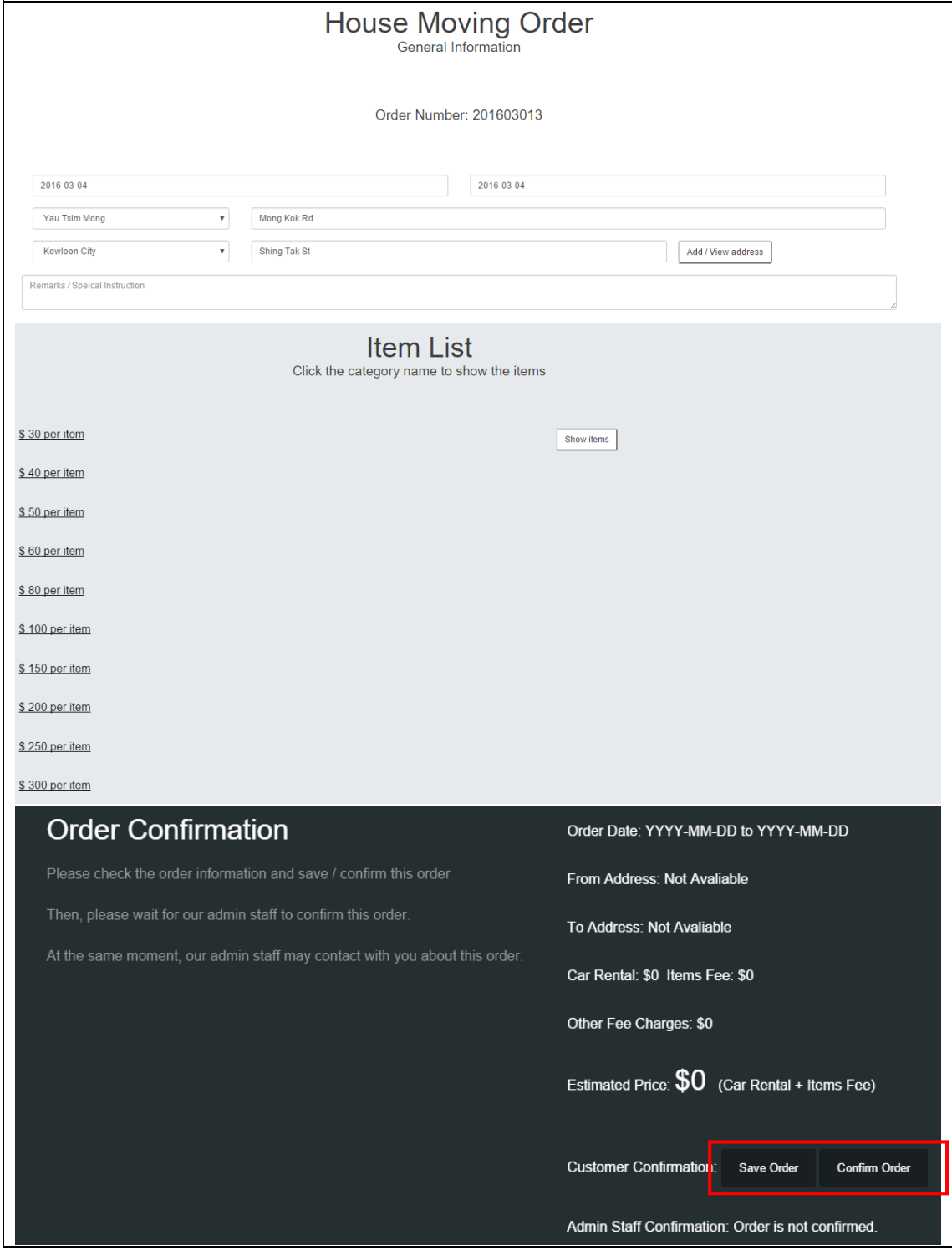
A3.10 Editing order customer made

Step 1: Select specific order in order records list generated by “ORDER RECORDS”.

Step 2: Edit order details.

Step 3: Click “Save Order” or “Confirm Order”.

Step 4: Message: “Order is saved successfully” will be displayed if the order is updated.

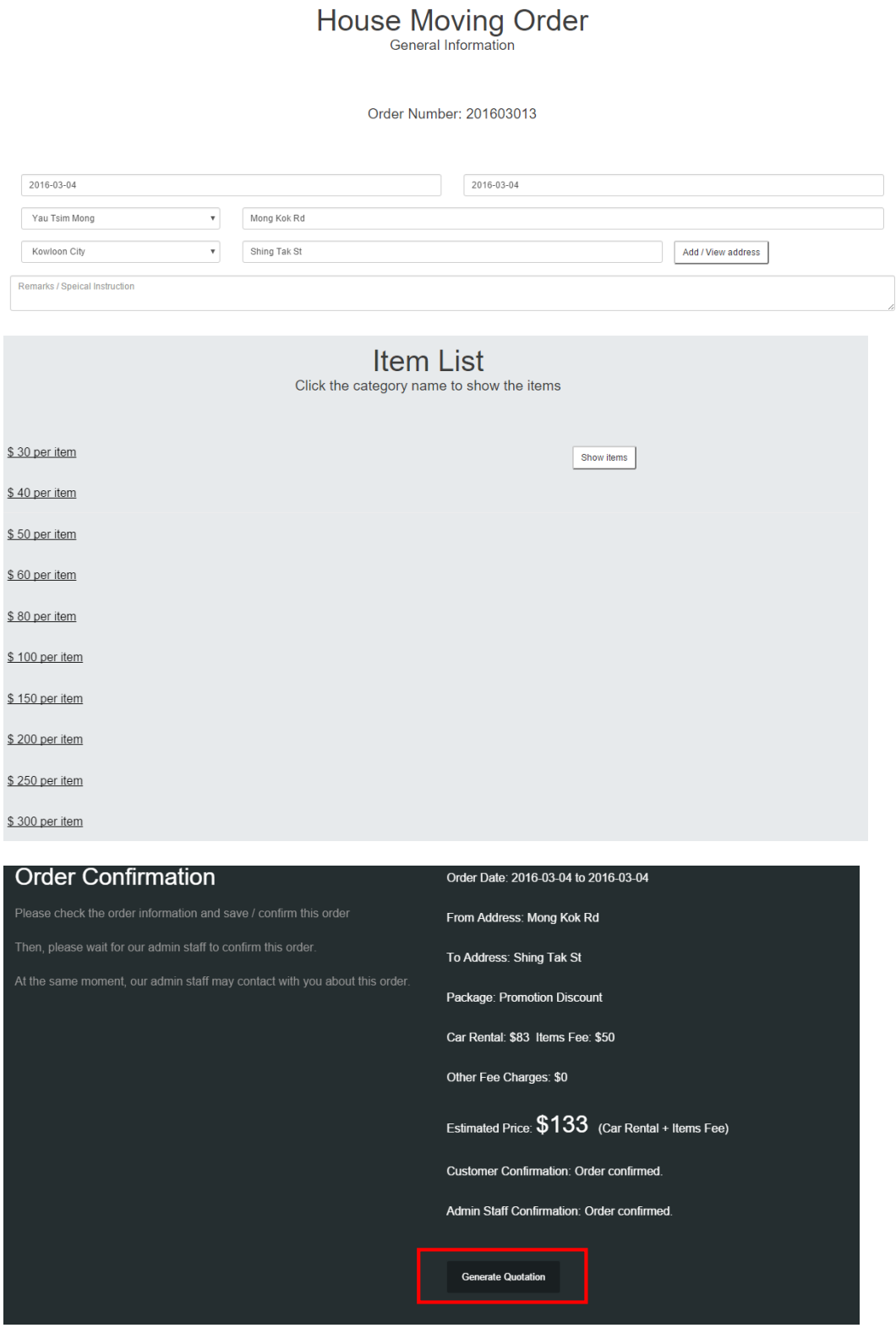


A3.11 Generating PDF documents including quotation, invoice and receipt

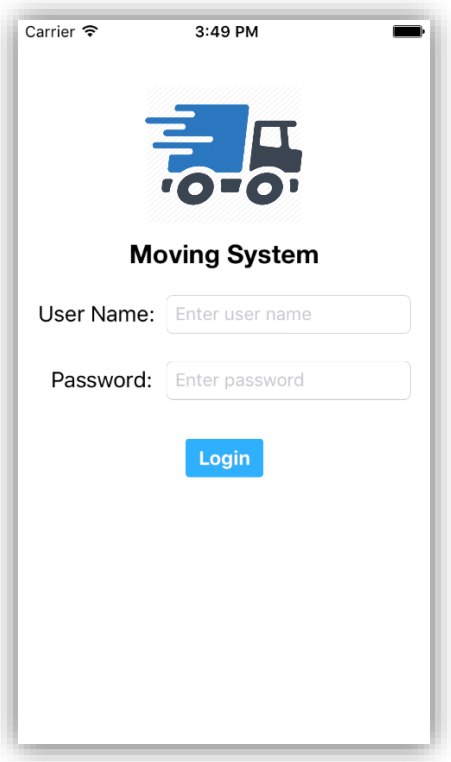
Step 1: Select specific order in order records list generated by “ORDER RECORDS”.

Step 2: Click “Generate Quotation”, “Generate Invoice” or “Generate Receipt” according to the order status.

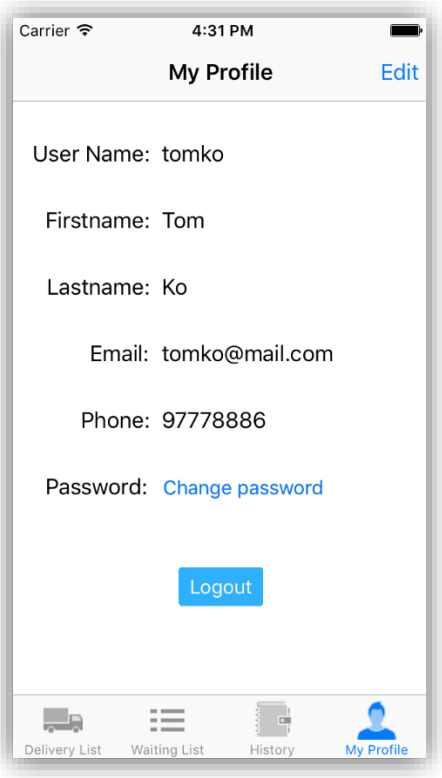
Step 3: Review the PDF documents (Quotation/Invoice/Receipt) in new tab.



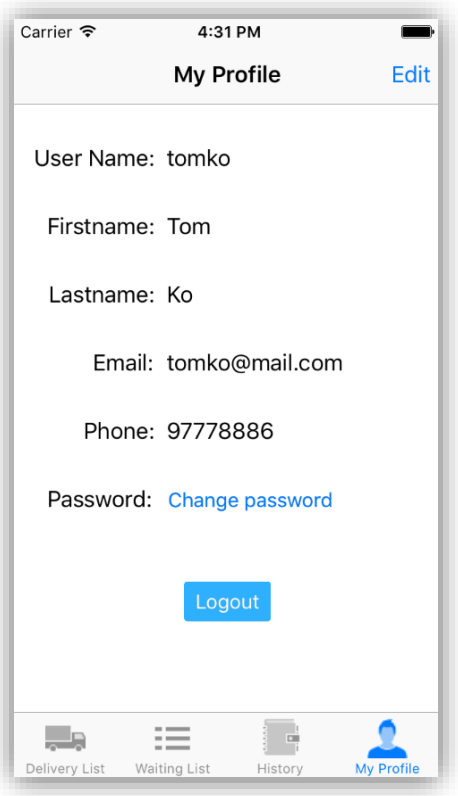
A4. User Manual – Driver

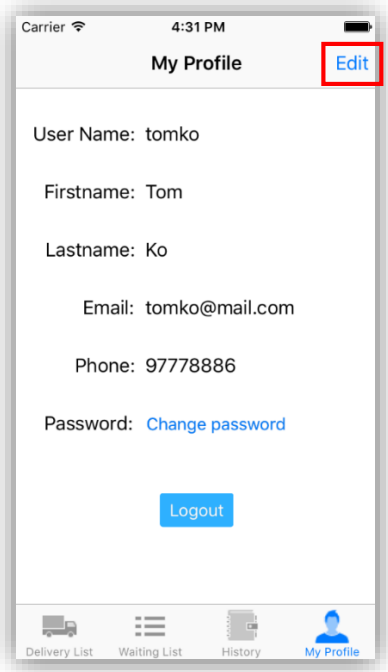
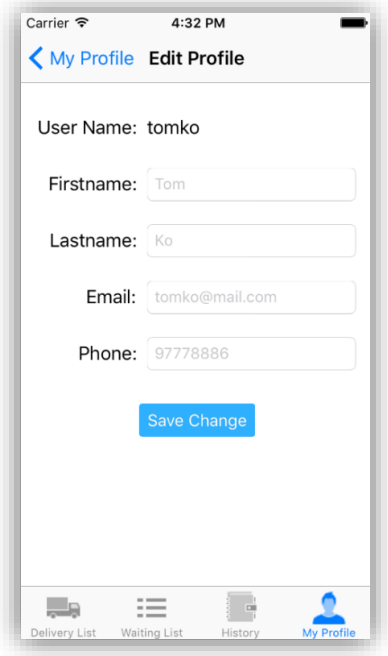
A4.1 Signing in the system as driver
Step 1: Open the mobile application in iOS platform “Moving System”. Step 2: Input username and password. Step 3: Tap “Login” Step 4: Redirect to the main page if the driver logins successfully.


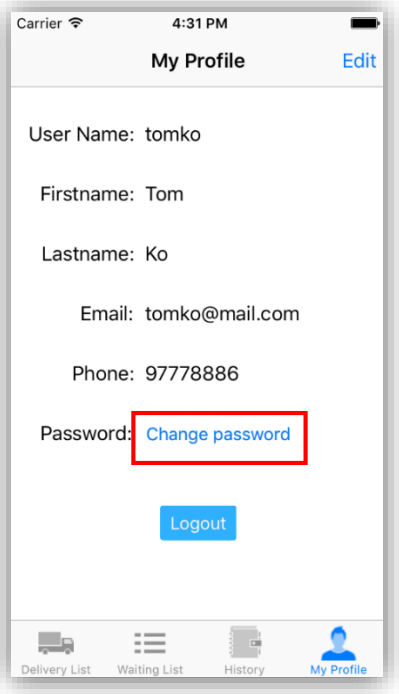
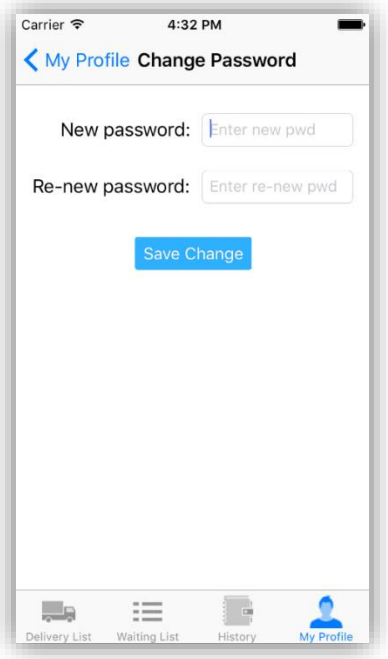
A4.2 Signing out the system
Step 1: Tap “My Profile” in bottom menu bar. Step 2: Tap “Logout”.



A4.3 Viewing driver profile
Step 1: Tap “My Profile” in bottom menu bar. Step 2: Review the user profile.

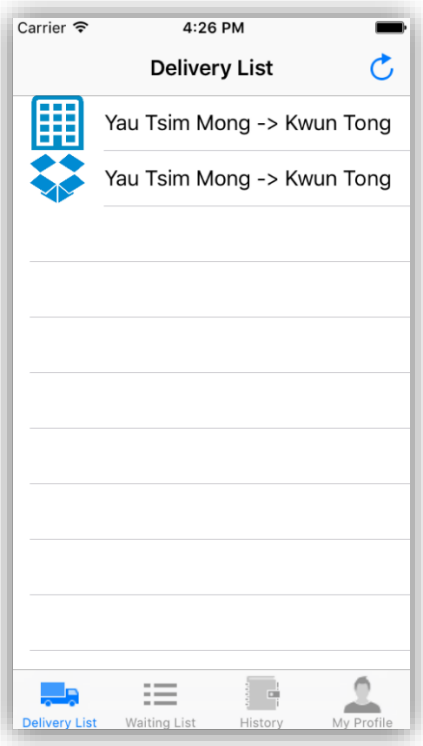


A4.4 Editing driver profile
Step 1: Tap “My Profile” in bottom menu bar. Step 2: Tap “Edit” in top right concern.

Step 3: Input updated user information. Step 4: Tap “Save Change”. Step 5: Redirect to previous page “My Profile” if the user profile is updated.


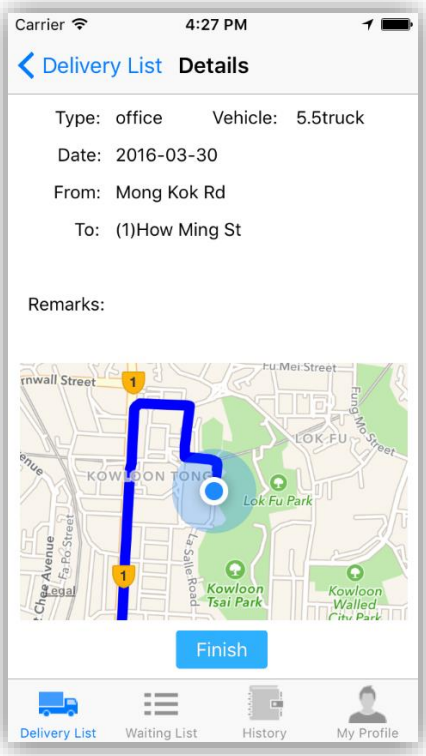
A4.5 Changing driver account password
Step 1: Tap “My Profile” in bottom menu bar. Step 2: Tap “Change password”.

Step 3: Input new password. Step 4: Tap “Save Change”. Step 5: Redirect to previous page “My Profile” if the user password is updated.


A4.6 Viewing delivery list and order details

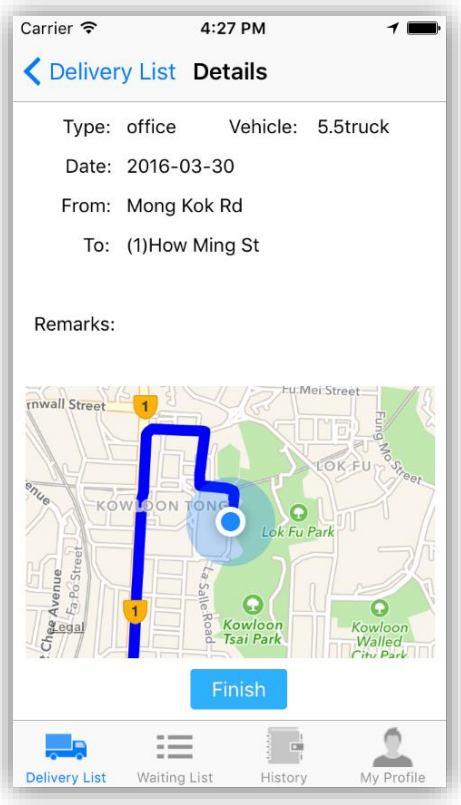
Step 1: Tap “Delivery List” in bottom menu bar.



Step 2: Select specific order for reviewing or finishing the order.

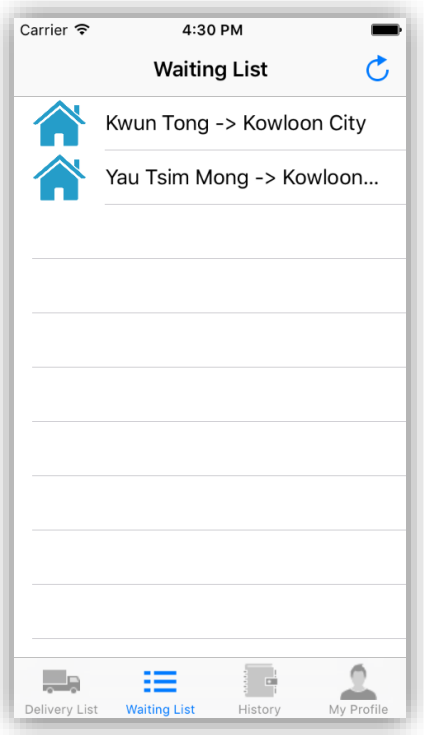


A4.7 Finishing delivery order of delivery list
Step 1: Select specific order in “Delivery List” for reviewing or finishing the order. Step 2: Tap “Finish”. Step 3: Message: “Order is finished” will be displayed if the order is finished.

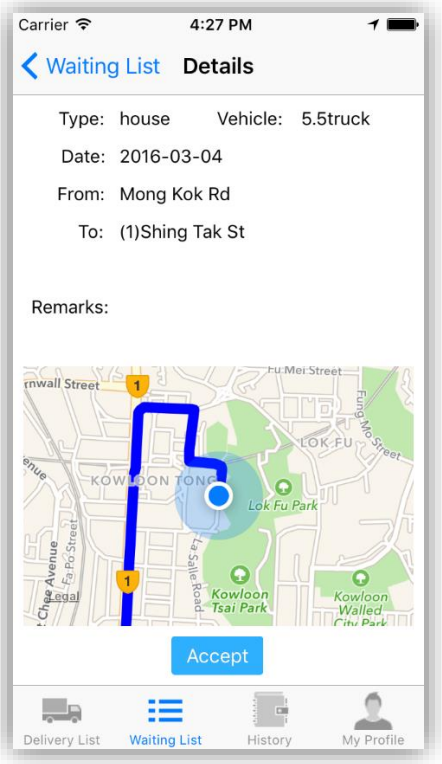


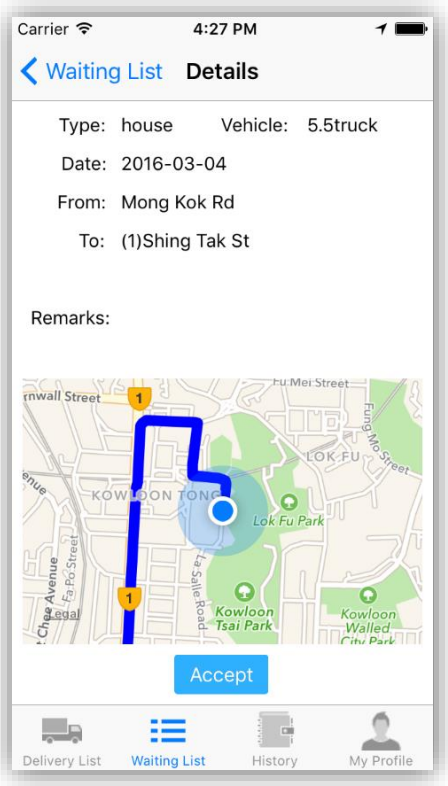
A4.8 Viewing waiting list and order details

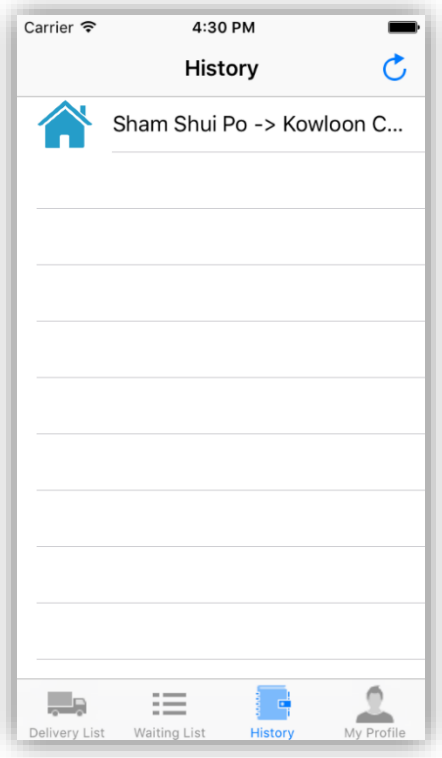
Step 1: Tap “Waiting List’ in bottom menu bar.



Step 2: Select specific order for reviewing or accepting the order.



A4.9 Accepting delivery order of waiting list
Step 1: Select specific order in “Waiting List” for reviewing or accepting the order. Step 2: Tap “Accept”. Step 3: Message: “Order is accepted and listed in the delivery list” if the order is accept.


A4.10 Viewing delivery history and order details
Step 1: Tap “History” in bottom menu bar.

Step 2: Select specific order for reviewing the order details.