# SUMMARY

Topic title: Building a fanpage-styled application for food and beverage providers.

Student: Nguyen Phuoc Hung

Student ID: 102140210 Class: 14TCLC2

The aim of this project was to create an application which supports food and beverage providers sale online on the facebook messenger platform. The food and beverage providers use a mobile application to create menus, promotions, answers of frequently asked questions which users access on the provider fanpage inbox. Through the provider fanpage inbox, users can choose some foods or drinks to request a delivery or get answers of some frequently asked questions without supporting from humans.

|  |  |
| --- | --- |
| DA NANG UNIVERSITY  **UNIVERSITY OF SICIENCE AND TECHNOLOGY**  INFORMATION TECHNOLOGY FALCUTY | **THE SOCIALIST REPUBLIC OF VIETNAM**  Independence - Freedom - Happiness |

# GRADUATION PROJECT REQUIREMENTS

Student Name: Nguyen Phuoc Hung Student ID : 102140210

Class: 14TCLC2 Faculty: Information Technology Major: Software Enginerring

1. *Topic title:*

Building a fanpage styled application for food and beverage providers.

1. *Project topic :*  *has signed intellectual property agreement for final result*
2. *Initial figure and data:*

Data generated from collection, searching the internetm, books ...

*Content of the explanations and calculations:*

* Manage and show menus.
* Manage and show promotions.
* Request delivery and handle orders.
* Book a table and handle reservations.
* Ask and manage answers of frequently asked questions.
* Switch to chat with admin status.

1. *Drawings, charts (specify the types and sizes of drawings):*

* Usercase diagram.
* Sequence diagram.
* Activity diagram.
* Class diagram.

1. *Supervisor (s):* Ph.D Huynh Huu Hung.
2. *Date of assignment : 03/09/2018.*
3. *Date of completion : 23/12/2018.*

|  |  |
| --- | --- |
|  | *Da Nang, 24 December 2018* |
| **Head of Division**…………………. | **Instructor** |

**PREFACE**

I am highly indebted to our project mentors, Mr. Huynh Huu Hung for their continuous support, supervision, motivation, and guidance throughout the tenure of my project in spite of their hectic schedule. They remained a driving spirit in my project and their experience gave me the understanding in handling research projects as well as helping me to clarify the abstruse concepts, requiring knowledge and perception, handling critical situations and in understanding the objective of my work.

I also want to thank my families and friends, who gave me the strength and confidence during my time of learning and during the implementation phase of this project. They have given a lot of love and encouragement for me which helped pass over the difficulties and fatigues.

I am also grateful to the directions of the management board and engineers of EM&AI company, who have facilitated my studying about technology, programming, workplace skill, enhance the best soft skills such as teamwork, attitude.

Without your generous help, my senior year would not have been successful.

Sincerely,

Nguyen Phuoc Hung

# ASSURRANCE

I assure:

1. The contents of this senior project are performed by myself following the guidance of lecturer Ph.D Huynh Huu Hung.

2. All references used in this senior project thesis, are quoted with the name of the author, project name, time and location to publish clearly and faithfully.

3. All invalid copies, educated statute violation or cheating will be borne the full responsibility by myself.

Student Performed

Nguyen Phuoc Hung

TABLE OF CONTENTS

[**SUMMARY**](#_Toc533084035)

[**GRADUATION PROJECT REQUIREMENTS**](#_Toc533084036)

[**PREFACE** i](#_Toc533084037)

[**ASSURRANCE** ii](#_Toc533084038)

[**TABLE OF CONTENTS** iii](#_Toc533084039)

[**LIST OF TABLES AND FIGURES** vi](#_Toc533084040)

[**LIST OF SYMBOLS AND ACRONYM** viii](#_Toc533084041)

[**INTRODUCTION** 1](#_Toc533084042)

[**1.** **Context and Purpose** 1](#_Toc533084043)

[**2. Scope** 2](#_Toc533084044)

[**Chapter 1: THEORIES AND TECHNOLOGIES** 4](#_Toc533084045)

[**1.1.** **Node js** 4](#_Toc533084046)

[1.1.1. Introduction 4](#_Toc533084047)

[1.1.2. Node js architecture 4](#_Toc533084048)

[1.1.3. Features 5](#_Toc533084049)

[**1.2.** **Messenger Platform** 6](#_Toc533084050)

[1.2.1. Introduction 6](#_Toc533084051)

[1.2.2. Platform Features 6](#_Toc533084052)

[**1.3.** **Ionic framework** 7](#_Toc533084053)

[1.3.1. Introduction 7](#_Toc533084054)

[1.3.2. Important concepts 7](#_Toc533084055)

[**1.4.** **MongoDB** 8](#_Toc533084056)

[1.4.1. Introduction 8](#_Toc533084057)

[1.4.2. Key Features 8](#_Toc533084058)

[**Chapter 2: ANALYSIS AND DESIGN** 10](#_Toc533084059)

[**2.1.** **Analysis** 10](#_Toc533084060)

[2.1.1. Manage and show menus 10](#_Toc533084061)

[2.1.2. Manage and show promotions 10](#_Toc533084062)

[2.1.3. Request delivery and handle orders 11](#_Toc533084063)

[2.1.4. Book a table and handle reservations 11](#_Toc533084064)

[2.1.5. Ask and manage frequently asked questions 12](#_Toc533084065)

[2.1.6. Chat with admin 12](#_Toc533084066)

[**2.2.** **Use case diagram** 12](#_Toc533084067)

[2.2.1. Overall diagram 12](#_Toc533084068)

[2.2.2. Manage menus 13](#_Toc533084069)

[2.2.3. View menus 14](#_Toc533084070)

[2.2.4. Manage promotions 14](#_Toc533084071)

[2.2.5. View promotions 15](#_Toc533084072)

[2.2.6. Request delivery 15](#_Toc533084073)

[2.2.7. Handle orders 16](#_Toc533084074)

[2.2.8. Book a table 16](#_Toc533084075)

[2.2.9. Handle reservations 17](#_Toc533084076)

[2.2.10. Ask FAQ 17](#_Toc533084077)

[2.2.11. Manage answer of FAQs 17](#_Toc533084078)

[2.2.12. Switch to chat with admin status 18](#_Toc533084079)

[**2.3.** **Activity diagram** 18](#_Toc533084080)

[**2.4.** **Sequence diagram** 19](#_Toc533084081)

[2.4.1. Request delivery 19](#_Toc533084082)

[2.4.2. Handle orders 20](#_Toc533084083)

[2.4.3. Manage menus 21](#_Toc533084084)

[2.4.4. View Menus 22](#_Toc533084085)

[**2.5. Class diagram** 23](#_Toc533084086)

[**Chapter 3: IMPLEMENTATION RESULTS** 25](#_Toc533084087)

[3.1. Manage and show menus 25](#_Toc533084088)

[3.2. Manage and show promotions 27](#_Toc533084089)

[3.3. Request delivery and handle orders 30](#_Toc533084090)

[3.4. Book a table and handle reservations 40](#_Toc533084091)

[3.5. Frequently asked questions 46](#_Toc533084092)

[3.6. Switch to chat with admin 47](#_Toc533084093)

[**CONCLUSION AND FUTURE WORK** 48](#_Toc533084094)

[**1. Achieve results** 48](#_Toc533084095)

[**2. Future works** 48](#_Toc533084096)

[**REFERENCES** 49](#_Toc533084097)

# LIST OF TABLES AND FIGURES

[Figure 0.1 Context 1](#_Toc533083924)

[Figure 0.2 Main flow 3](#_Toc533083925)

[Figure 1.1 Nodejs architecture 5](#_Toc533083926)

[Figure 2.1 Overall use case diagram 13](#_Toc533083927)

[Figure 2.2 Manage menus use case 14](#_Toc533083928)

[Figure 2.3 View menus use case 14](#_Toc533083929)

[Figure 2.4 Manage promotions use case 15](#_Toc533083930)

[Figure 2.5 View promotions use case 15](#_Toc533083931)

[Figure 2.6 Request delivery use case 16](#_Toc533083932)

[Figure 2.7 Handle orders use case 16](#_Toc533083933)

[Figure 2.8 Book a table use case 17](#_Toc533083934)

[Figure 2.9 Handle reservations use case 17](#_Toc533083935)

[Figure 2.10 Ask FAQs use case 17](#_Toc533083936)

[Figure 2.11 Manage answer of FAQs use case 18](#_Toc533083937)

[Figure 2.12 Switch to chat with admin use case 18](#_Toc533083938)

[Figure 2.13 Request a delivery activity diagram 19](#_Toc533083939)

[Figure 2.14 Request delivery sequence diagram 20](#_Toc533083940)

[Figure 2.15 Handle orders sequence diagram 21](#_Toc533083941)

[Figure 2.16 Manage menus sequence diagram 22](#_Toc533083942)

[Figure 2.17 View menus sequence diagram 23](#_Toc533083943)

[Figure 2.18 Class diagram 24](#_Toc533083944)

[Figure 3.1 “Thực đơn” screen 26](#_Toc533083945)

[Figure 3.2 Menu generate template 26](#_Toc533083946)

[Figure 3.3 Menu webview 27](#_Toc533083947)

[Figure 3.4 Promotion screen 28](#_Toc533083948)

[Figure 3.5 List promotion generate templates 29](#_Toc533083949)

[Figure 3.6 Promotion detail webview 30](#_Toc533083950)

[Figure 3.7 Delivery generate template 31](#_Toc533083951)

[Figure 3.8 Menu webview 32](#_Toc533083952)

[Figure 3.9 Order information webview 33](#_Toc533083953)

[Figure 3.10 Confirm Information webview 34](#_Toc533083954)

[Figure 3.11 Notify webview 35](#_Toc533083955)

[Figure 3.12 Delivery message 36](#_Toc533083956)

[Figure 3.13 “Danh sách yêu cầu” screen 37](#_Toc533083957)

[Figure 3.14 “Thông tin giao hàng” screen 38](#_Toc533083958)

[Figure 3.15 Order acceptance message 38](#_Toc533083959)

[Figure 3.16 “Từ chối giao hàng” screen 39](#_Toc533083960)

[Figure 3.17 Order refusal message 40](#_Toc533083961)

[Figure 3.18 Book generate template 40](#_Toc533083962)

[Figure 3.19 Book webview 41](#_Toc533083963)

[Figure 3.20 Reservation message 42](#_Toc533083964)

[Figure 3.21 “Danh sách yêu cầu” screen 43](#_Toc533083965)

[Figure 3.22 “Thông tin đặt bàn” screen 44](#_Toc533083966)

[Figure 3.23 Reservation acceptance message 45](#_Toc533083967)

[Figure 3.24 “Từ chối đặt bàn” screen 46](#_Toc533083968)

[Figure 3.25 Reservation refusal message 46](#_Toc533083969)

[Figure 3.26 Ask FAQs 46](#_Toc533083970)

[Figure 3.27 Switch to chat with admin 47](#_Toc533083971)

# LIST OF SYMBOLS AND ACRONYM

|  |  |  |
| --- | --- | --- |
| **No.** | **Items** | **Description** |
| 1 | SP | Service Provider |
| 2 | FAQ | Frequently Asked Questions |

# INTRODUCTION

1. **Context and Purpose**

Nowadays, the facebook has developed to become a one of populate social networks in the world. Follow facebook statistics, at the third quarter of 2018, the facebook had 2.27 monthly active users.

This has led to the development of the online sales, especially on the facebook platform has grown quickly. Usually, the milk tea, coffee places or restaurants always have a individual fanpage to marketing, sale, provide their information to customer.

Daily, the food and beverage provider have to spend expenses to hire a staff who onlines on the facebook to handle request orders and reply some repeated questions from user through their fanpage inbox.

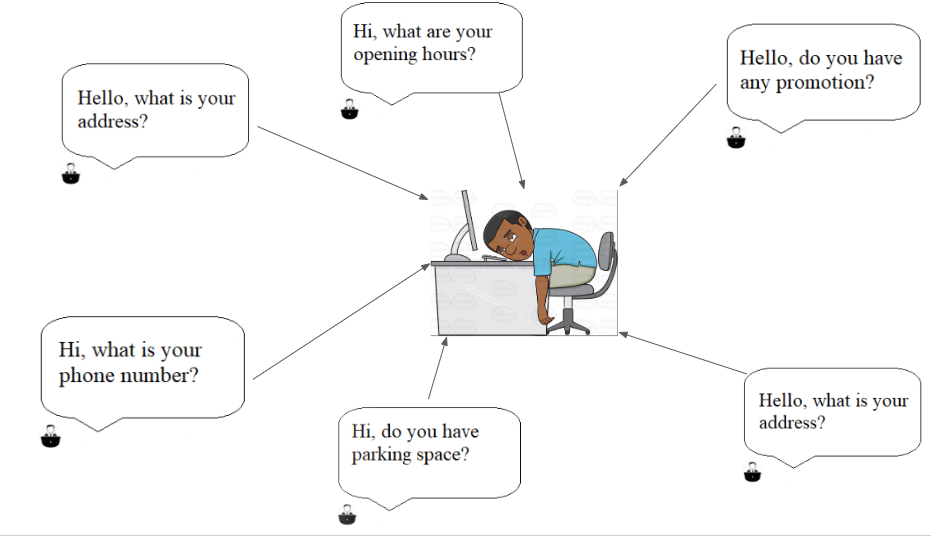


Figure 0.1 Context

So, I apply technology to build an application which replaces a staff response to customers.

This application will resolve some issues:

* Save expenses for the food and beverage providers.
* Response to users quickly.
* Avoid boring and repeated jobs.
* Users can easily request a delivery.

For that reason, I decided to do project named "***Building a fanpage styled application for food and beverage providers***".

1. **Scope**

Through fanpage inbox of providers, users can use some features:

* View menus
* View promotions
* Request delivery
* Book a table
* Ask some FAQs
* Switch to chat with admin

And to users can use that features above, I need a mobile application to food and beverage providers manage their information. Some features which food and beverage providers can use in this application:

* Manage menus
* Manage promotions
* Handle orders
* Handle reservation
* Manage answers of FAQs

Here is some main flows of system.

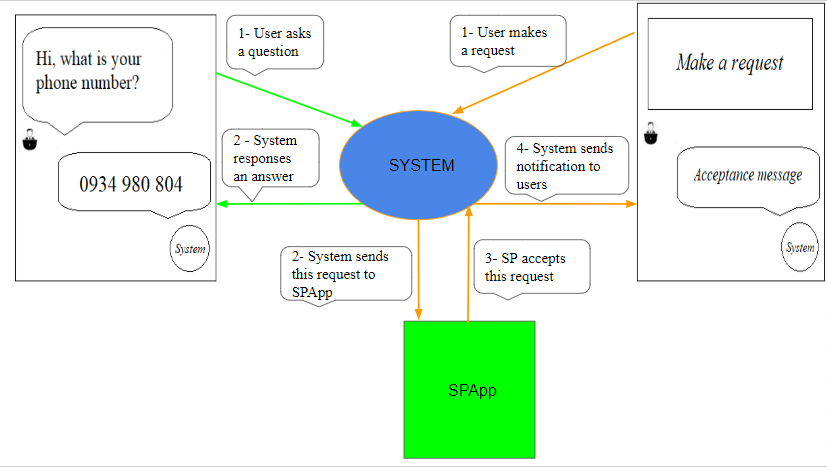


Figure 0.2 Main flow

# Chapter 1: THEORIES AND TECHNOLOGIES

Bellow is theories of technologies I did use in developing my project.

* 1. **Node js**

### Introduction

Node.js is a server-side platform built on Google Chrome's JavaScript Engine (V8 Engine). Node.js was developed by Ryan Dahl in 2009.

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

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

### Node js architecture

Node architecture includes components:

1. *V8*

The high-performance Javascript engine open sourced by Google and implement in C++. V8 takes the code writen in Javascript, compile it into machine code and excutes it.

1. *Libuv*

The C library that providers asynchronous features. It maintains an event loop, thread pool, file system events and child processes among other critical functionalities.

1. *Application/ Modules*

This is where all the javascript code lives: application code, nodejs core module, any modules installed from npm.

1. *Bindings*

Bindings basically are libraries that "bind" two different programming languages so that code written in one language can be used in code written in another library. In nodejs, bindings help javascript and C/C++ code can communication smoothly.

1. *C/C++ addons*

To can include a third-party or C/C++ library, we have to write the glue code for that libraries. That glue code are called addons. Bindings and addons as bridges between your JavaScript code and Node.js’ C/C++ code.

1. [*Other C/C++ Components/Dependencies*](https://nodejs.org/en/docs/meta/topics/dependencies/)

Such as [c-ares](http://c-ares.haxx.se/), [crypto (OpenSSL)](https://www.openssl.org/), [http-parser](https://github.com/nodejs/http-parser), and [zlib](http://zlib.net/). These dependencies provide low-level interactions with servers to establish important functionalities such as networking, compressing, encrypting, etc.

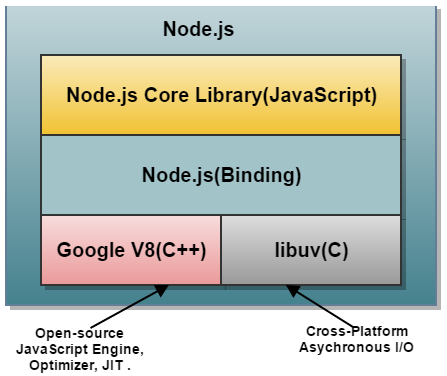


Figure 1.1 Nodejs architecture

### Features

1. *Modularity*

Major advantage of Node Js Platform is that it’s modularity. Each and every functionality is divided and implemented as a separate module or package.

1. *Non-blocking or Asynchronous IO*

Node JS supports Non-blocking IO i.e. it uses Asynchronous IO Model to interact with File system or to do Socket communication or network communication. Asynchronous IO Model means if IO processing is taking more time, then it permits other processing to continue before the transmission has finished.

1. *Single Threaded but Highly Scalable*

Node.js uses a single threaded model with event looping. Event mechanism helps the server to respond in a non-blocking way and makes the server highly scalable as opposed to traditional servers which create limited threads to handle requests. Node.js uses a single threaded program and the same program can provide service to a much larger number of requests than traditional servers like Apache HTTP Server.

1. *Better Socket API*

Node JS Platform provides very good Socket Module API to develop Real-time, Multi-User Chat and Multi-Player Gaming Applications very easily. It supports Unix Socket programming like pipe().

* 1. **Messenger Platform**

### Introduction

The messenger platform is a toolbox for buildings bots. Some core components is available to build a bot:

1. *Integration components*

The Messenger Platform is comprised of a set of features intended to give you everything you need to create awesome, interactive experiences in Messenger, including APIs, web plug-ins, and a full webview. This is intended to be an overview of the core platform components that are available to help you better understand the Platform.

1. *Conversation components*

Conversations are a lot more than simple text messages when you are building a bot on the Messenger Platform. In addition to text, the Platform allows you to send rich-media, like audio, video, and images, and provides a set of structured messaging options in the form of message templates, quick replies, buttons and more. This is intended to be an overview of the components that are available for you to create your Messenger experience in-conversation.

### Platform Features

1. *Messaging*

Some available message templates that allows we to send structured messages for a richer experience: generate template, list template, button template, open graph template, receipt template, airline template, media template.

The Messenger Platform supports sending many types of content in messages, include: text, audio, images, video, files.

1. *Webview*

The Messenger Platform allows open a standard webview, where developer can load webpages inside Messenger. This offers experiences and features that may be difficult to offer with message bubbles, such as picking products to buy, seats a book, or dates to reserve.

1. *Payments*

The Messenger Platform make it fast, easy, and seamless to accept payments from users of bots. Payments on the Messenger Platform can be implemented in conversation with the Buy Button, or in the Messenger webview with tokenized payments. This gives developer the flexibility to handle user payments in the way that makes the mose sense for bots.

* 1. **Ionic framework**

### Introduction

Ionic Framework is an open source SDK that enables developers to build performant, high-quality mobile apps using familiar web technologies (HTML, CSS, and JavaScript).

Ionic is focused mainly on the look and feel, or the UI interaction, of an app. This means that it’s not a replacement for Cordova or your favorite JavaScript framework. Instead, Ionic fits in well with these projects, in order to simplify one big part of your app development process: the front-end.

### Important concepts

1. *CLI*

The CLI or command line interface, is a tool that provides a number of helpful commands to Ionic developers. In addition to installing and updating Ionic, the CLI comes with a built-in development server, build and debugging tools, and much more.

1. *Components*

Components in Ionic are reusable UI elements that serve as the building blocks for mobile app. Components are made up of HTML, CSS, and sometimes JavaScript. Every Ionic component adapts to the platform on which your app is running.

1. *Theming*

Themes are sets of styles that get applied to an app. Ionic uses a light theme by default, but it also comes with a dark theme. In addition to theming, Ionic’s Platform Continuity enables components to have platform-specific styles. This means the app’s styles will change based on the platform (iOS, Android, etc.) on which it’s being run, offering users an experience with which they’re familiar.

* 1. **MongoDB**

### Introduction

MongoDB is a cross-platform, document oriented database that provides, high performance, high availability, and easy scalability. MongoDB works on concept of collection and document.

Collection is a group of MongoDB documents. It is the equivalent of an relational database management system table. A collection exists within a single database. Collections do not enforce a schema. Documents within a collection can have different fields. Typically, all documents in a collection are of similar or related purpose.

A document is a set of key-value pairs. Documents have dynamic schema. Dynamic schema means that documents in the same collection do not need to have the same set of fields or structure, and common fields in a collection's documents may hold different types of data.

### Key Features

1. *Uses BSON format*

BSON is a JSON-like storage format. BSON stands for Binary JSON which is a binary-encoded serialization of JSON-like documents that MongoDB uses when storing documents in collections. It adds support for data types like Date and binary that aren’t supported in JSON. BSON format is the use of the \_id field as primary key. The value of \_id field will usually be a unique identifier type, named ObjectId, that is either generated by the application driver or by the mongodb service. Another advantage of using BSON format is that it enables MongoDB to internally index and map document properties and even nested documents. It is designed to be more efficient in size and speed, allowing MongoDB’s high read/write throughput.

1. *MongoDB sharding*

The major and very common problem with a growing web application is scaling. To overcome this, MongoDB has come up with Sharding feature. It is one of the greatest key features of MongoDB. Sharding is a method for distributing data across multiple machines. MongoDB uses sharding to support deployments with very large data sets and high throughput operations.

1. *MongoDB Ad hoc queries*

MongoDB supports field, range queries, regular expression searches. Queries can return specific fields of documents and also include user-defined JavaScript functions. MongoDB is able to support ad hoc queries by indexing BSON documents and using a unique query language.

1. *MongoDB is Schema–Less*

MongoDB is a schema-less database because of which is much more flexible than traditional database tables. The benefit is the lack of setup and the reduced friction with OOP. So, in order to save an object, I just have to serialize it to JSON and send it to MongoDB. There is no need for type mapping which removes an additional burden.

1. *MongoDB Indexing*

Indexes are created to improve the performance of searches. The good thing is that any field in a MongoDB document can be indexed with primary and secondary indices. It enables the database engine to efficiently resolve queries which make it one of the best key features of MongoDB. The database engine can use a predefined index, which maps documents fields and can tell the engine which documents are compatible with this query statement, hence improves performance.

1. *File Storage*

MongoDB can be used as a file system with load balancing and data replication features over multiple machines for storing files. This function, called Grid File System, is included with MongoDB drivers which stores files. MongoDB exposes functions for file manipulation and content to developers.

1. *Replication*

MongoDB provides replication feature by distributing data across different machines. It can have one primary node and one or more secondary nodes. This typology is known as replica set. Replica set is like master-slave replication. A master can perform Reads and Writes and a Slave copies data from the master and can only be used for reads or back up (not writes).

# Chapter 2: ANALYSIS AND DESIGN

* 1. **Analysis**

Follow the name of the project, the application must build through facebook fanpage. It replaces a staff to response some frequently asked questions and requests from users.

To clarify this application, we have these features as following.

* + 1. ***Manage and show menus***

A food and beverage provider alway has a list of dishs or drinks. So I need a SPApp to Service Provider can manage their menus and a webview to user can view menus of providers.

In SPApp’s “Trang chủ” screen, Service Provider taps on “Thực đơn” section to open “Thực đơn” screen. “Thực đơn” screen will show list menus of provider. To create a menu, Service Provider taps on “+” button in bottom right corner of “Thực đơn” screen. Taps on a item of list menus in “Thực đơn” screen to open “Chỉnh sửa” screen. In “Chỉnh sửa” screen, Service Provider taps on “LƯU” button to save edited information or taps on “XÓA” button to delete a menu.

In Fanpage Inbox, User click on “Thực đơn” quick reply to open “Thực đơn” generic template. Then, User click on “Thực đơn” button bellow “Thực đơn” generic template to open webview. Webview display all menus which provider created in SPApp.

* + 1. ***Manage and show promotions***

In SPApp’s “Trang chủ” Screen, Service Provider taps on “Khuyến mãi” section to open “Khuyến mãi” screen. “Khuyến mãi” screen will show list promotions of provider. To create a promotion, Service Provider taps on “+” button in bottom right corner of “Khuyến mãi” screen. Tap on a item of list promotions in “Khuyến mãi” screen to open “Chỉnh sửa” screen. In “Chỉnh sửa” screen, Service Provider taps on “LƯU” button to save edited information or taps on “XÓA” button to delete a promotion.

In Fanpage Inbox, when user clicks on “Khuyến mãi” quick reply, a list of promotions generate templates will be displayed. To view detail of a promotion, user can click on this promotion generate template.

* + 1. ***Request delivery and handle orders***

This feature is the main feature of this application. This feature will include two sub-features: request a delivery and handle order.

In Fanpage Inbox, Users can click on “Giao hàng” call to action in Persistent Menu or click on “Giao hàng” quick reply to open Delivery generate template. Then user clicks on “Giao hàng” button bellow this generate template to open “Thực đơn” webview and carry out four steps to request a delivery:

* Select items in a list of menus which user wants to buy and click on “TIẾP TỤC” button to finish step order.
* In the second step, user have to fill in their address and phone number and clicks on “TIẾP TỤC” button to finish this step.
* In the third step, user view ordered items again and click on “Xác nhận” button to confirm their order.
* Final step is a successful orderd notification form system. User clicks on “Đóng” button to close this page.

After that, System will send a message to user through facebook messenger which includes “Hủy” button to user can cancel their.

In SPApp’s “Trang chủ” Screen, Service Provider taps on “Giao hàng/ Đặt bàn” section to open “Danh sách yêu cầu” screen and taps on “Giao hàng” tab to view a list of orders. Tap on a item of a list of orders to open “Thông tin giao hàng” screen. In “Thông tin giao hàng” screen, Service Provider taps on “CHẤP NHẬN” button to accept this order or taps on “TỪ CHỐI” button and write reasons to refuse this order.

After that, System will send a message to user about status of their order through facebook messenger.

* + 1. ***Book a table and handle reservations***

This feature will include two sub-features: book a table and handle reservations.

In Fanpage Inbox, Users can click on “Đặt bàn” call to action in Persistent Menu or click on “Đặt bàn” quick reply to open Book generate template. Then user clicks on “Đặt bàn” button bellow this generate template to open “Đặt bàn” webview and fill in date, time and number of people and clicks on “Đặt bàn” to book a table.

After that, System will send a message to user through facebook messenger which includes “Hủy” button to user can cancel their reservation.

In SPApp’s “Trang chủ” screen, Service Provider taps on “Giao hàng/ Đặt bàn” section to open “Danh sách yêu cầu” screen and taps on “Đặt bàn” tab to view a list of reservations. Tap on a item of a list of reservations to open “Thông tin đặt bàn” screen. In “Thông tin đặt bàn” screen, Service Provider taps on “CHẤP NHẬN” button to accept this order or taps on “TỪ CHỐI” button and write reasons to refuse this reservation.

After that, System will send a message to user about status of their reservation through facebook messenger.

* + 1. ***Ask and manage frequently asked questions***

This feature will include two sub-features: ask FAQ and manage answer of FAQs.

In Fanpage Inbox, Users can input some FAQs to request service provider’s information. For example FAQs: where is your address?, what is your phone number? ...

After that, System will send a request to API of Dialogflow to get intent of question and response a answer of question to users.

In SPApp’s “Trang chủ” Screen, Service Provider taps on “FAQs” section to open “FAQs” screen. “FAQs” screen will show list of answers of FAQs. To update answer for a FAQ, Service Provider taps on this item to open “Chỉnh sửa” screen and taps on “LƯU” button to save edited information.

* + 1. ***Chat with admin***

In Fanpage Inbox, Users can click on “Chat với Admin” call to action in Persistent Menu or click on “Chat với Admin” quick reply to switch to chat with admin of fanpage.

When users want to stop chat with admin and begin chat with system, users can input “#STOP” or click on “Dừng Chat với Admin” call to action in Persistent Menu.

* 1. **Use case diagram**
     1. ***Overall diagram***

The application includes these features:

* Manage and show menus
* Manage and show promotions
* Request delivery and handle orders
* Book a table and handle reservation
* Anwer some frequently asked questions
* Switch to chat with admin

With these features above (the details of the features will be explained above), the application has two actors (the users and the service provider) interact with the system. Here is overall use-case.

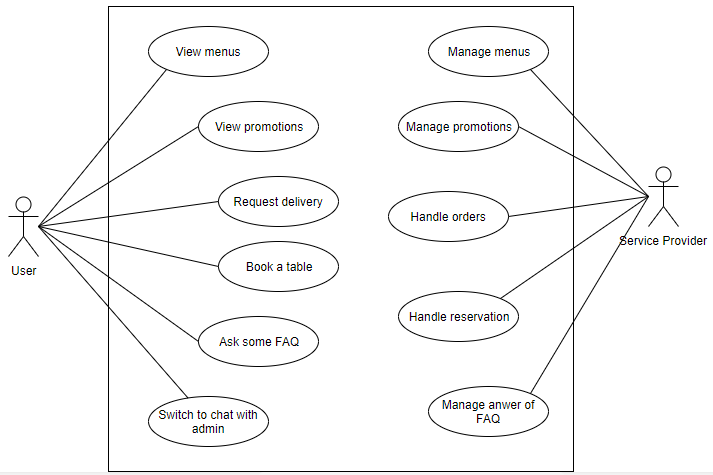


Figure 2.1 Overall use case diagram

* + 1. ***Manage menus***

Manage menus includes there feature: View, Add, View Detail, Update, Delete.

After Service Provider opens the SPApp, login and taps on “Thực đơn” section at “Trang chủ” screen, “Thực đơn” screen will be opened. And Service Provider can add, view detail, update or delete them.

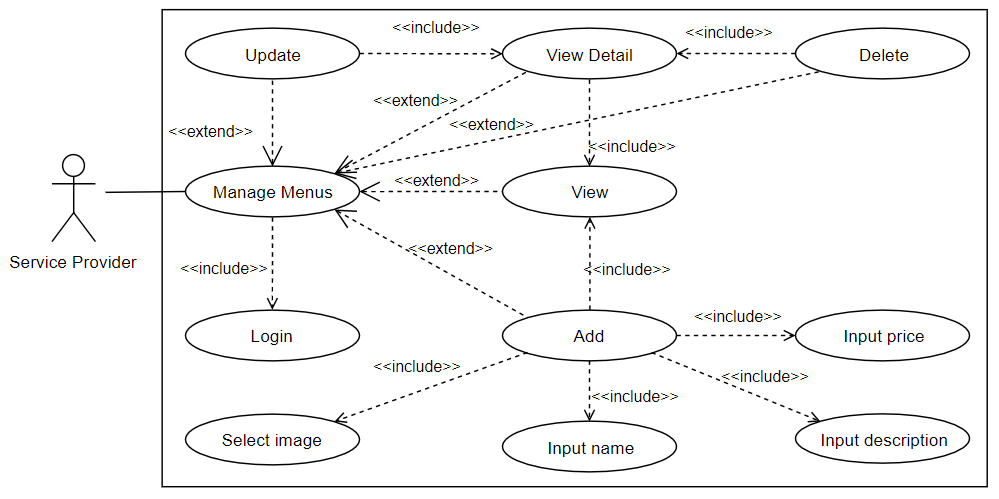


Figure 2.2 Manage menus use case

* + 1. ***View menus***

After the users open the fanpage inbox of Service Provider, to view a list of menus of Service Provider, the users must open “Thực đơn” webview. But before that, they have to open “Thực đơn” generate template.

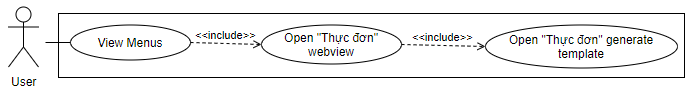


Figure 2.3 View menus use case

* + 1. ***Manage promotions***

Manage promotions includes there feature: View, Add, View Detail, Update, Delete.

After Service Provider opens the SPApp, login and taps on “Khuyến mãi” section at “Trang chủ” screen, “Khuyến mãi” screen will be opened. And Service Provider can add, view detail, update or delete them.

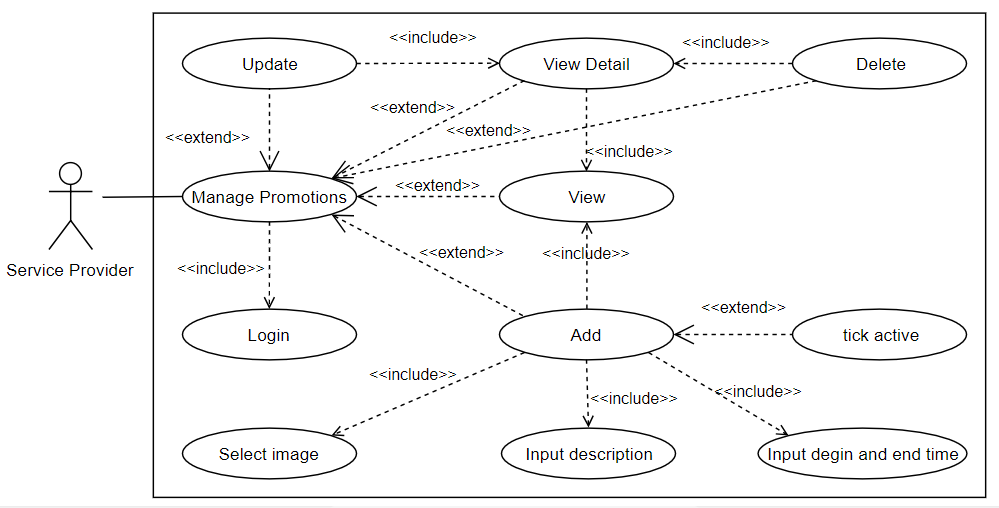


Figure 2.4 Manage promotions use case

* + 1. ***View promotions***

After the users open the fanpage inbox of Service Provider, to view a list of promotions of Service Provider, the users have to click on “Khuyến mãi” quick reply.

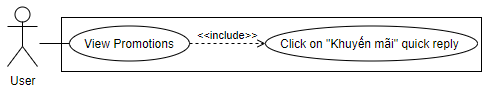


Figure 2.5 View promotions use case

* + 1. ***Request delivery***

After the users open the fanpage inbox of Service Provider, to request delivery the users have to open “Thực đơn” webview, choose foods or drinks, input order information and confirm the order. But before that, they must open “Giao hàng” generate template by clicking “Giao hàng” quick reply or “Giao hàng” call to action in Persistent Menu.

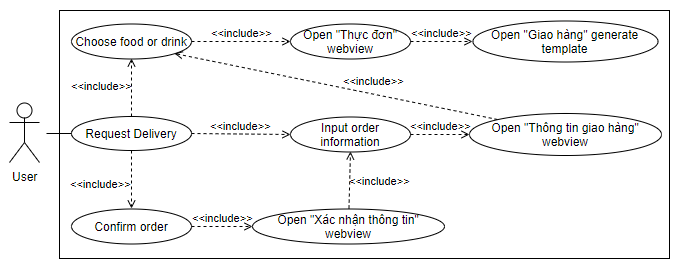


Figure 2.6 Request delivery use case

* + 1. ***Handle orders***

Handle orders includes there feature: Accept and Refuse.

After Service Provider opens the SPApp, login and taps on “Giao hàng/ Đặt bàn” section at “Trang chủ” screen, “Danh sách yêu cần” screen will be opened. And Service Provider can view detail and accept or refuse them.

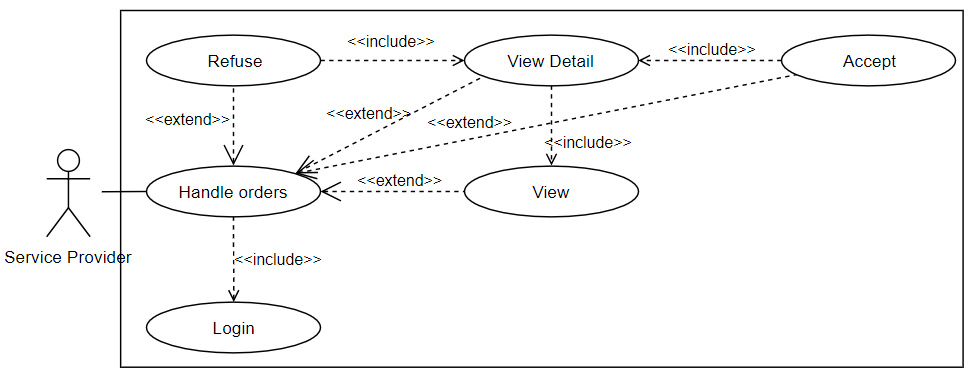


Figure 2.7 Handle orders use case

* + 1. ***Book a table***

After the users open the fanpage inbox of Service Provider, to book a table the users have to open Book webview, input reservation information and click on “Đặt bàn” button. But before that, they must open Book generate template by clicking “Đặt bàn” quick reply or “Đặt bàn” call to action in Persistent Menu.

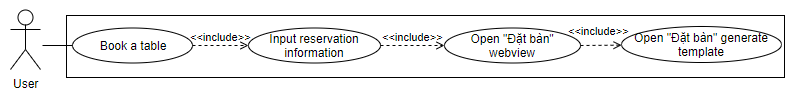


Figure 2.8 Book a table use case

* + 1. ***Handle reservations***

Handle orders includes there feature: Accept and Refuse.

After Service Provider opens the SPApp, login, taps on “Giao hàng/ Đặt bàn” section at “Trang chủ” screen and taps on “Đặt bàn” tab, “Danh sách yêu cầu” screen will display list of reservations. And Service Provider can view detail and accept or refuse them.

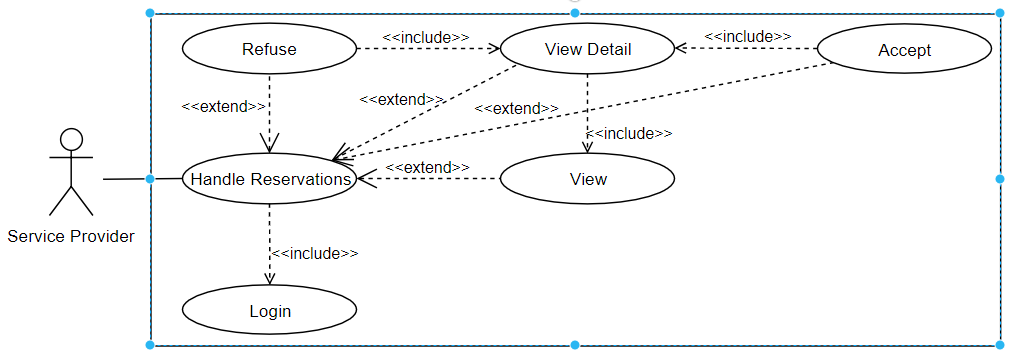


Figure 2.9 Handle reservations use case

* + 1. ***Ask FAQ***

After the users open the fanpage inbox of Service Provider, to ask some FAQs about information of Service Provider, the users have to input question.

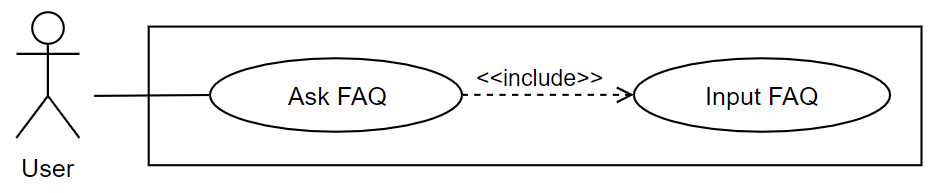


Figure 2.10 Ask FAQs use case

* + 1. ***Manage answer of FAQs***

Manage answer of FAQs includes there feature: View, View Detail, Update.

After Service Provider opens the SPApp, login and taps on “FAQs” section at “Trang chủ” screen, “FAQs” screen will be opened. And Service Provider can view detail or update.

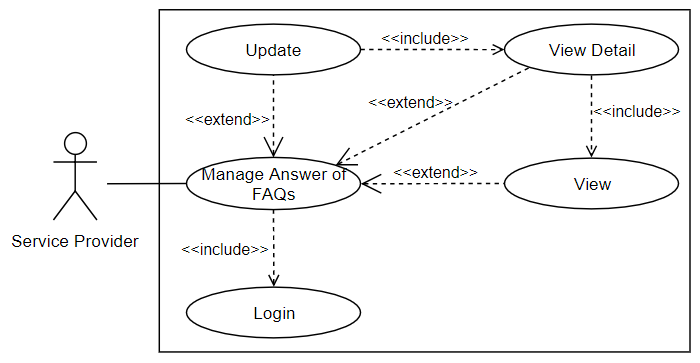


Figure 2.11 Manage answer of FAQs use case

* + 1. ***Switch to chat with admin status***

After the users open the fanpage inbox of Service Provider, to switch to chat with admin, the users must click on “Chat với Admin” quick reply.

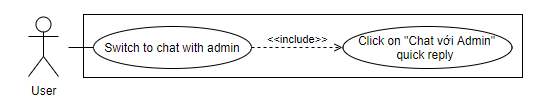


Figure 2.12 Switch to chat with admin use case

* 1. **Activity diagram**

The main feature is delivery requests so here is the activity of that feature:

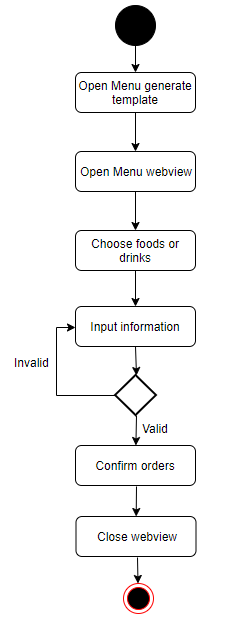


Figure 2.13 Request a delivery activity diagram

* 1. **Sequence diagram**
     1. ***Request delivery***

In request delivery sequence diagram, ShipController is responsible for handling the request when users interact with Fanpage Inbox. Depend on request types, ShipController will call MenuService to query menus or call ShipService to create a order.

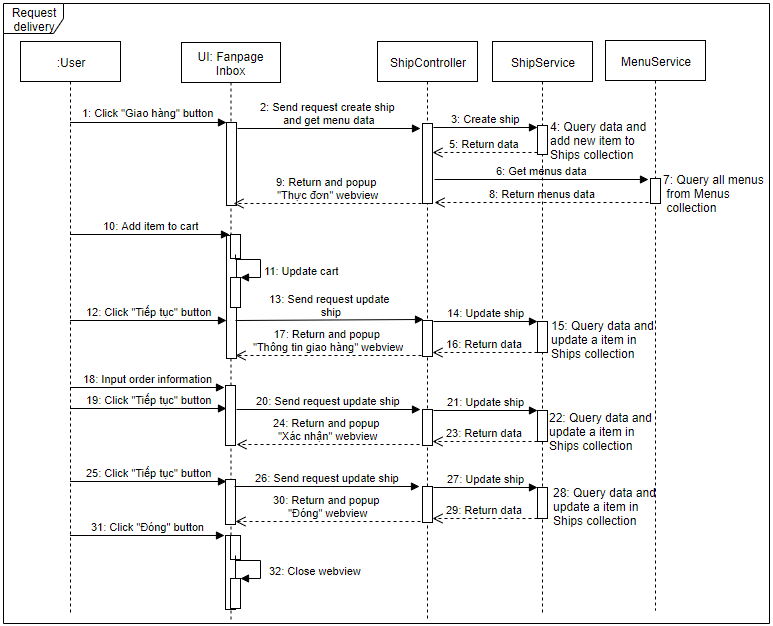


Figure 2.14 Request delivery sequence diagram

* + 1. ***Handle orders***

In handle orders sequence diagram, ShipRouter is responsible for defining routes and forwarding datas from request to ShipController. ShipController calls ShipService to get or update datas to Database and responses to the request.

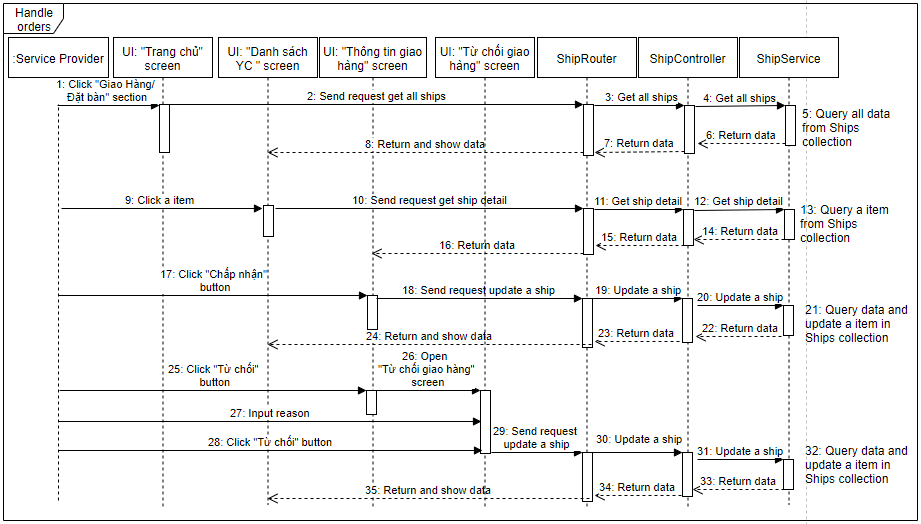


Figure 2.15 Handle orders sequence diagram

* + 1. ***Manage menus***

In manage menus sequence diagram, MenuRouter is responsible for defining routes and forwarding datas from request to MenuController. MenuController calls MenuService to get or update datas to Database and responses to the request.

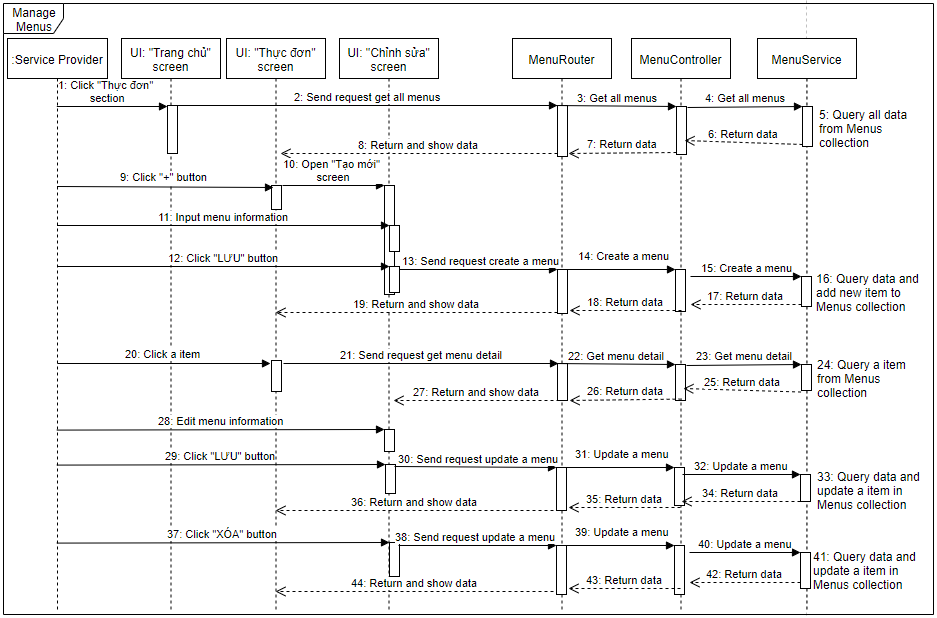


Figure 2.16 Manage menus sequence diagram

* + 1. ***View Menus***

In view menus sequence diagram, MenuController is responsible for handling the request when users interact with Fanpage Inbox, call MenuService to get or update datas to Database and response to the request.

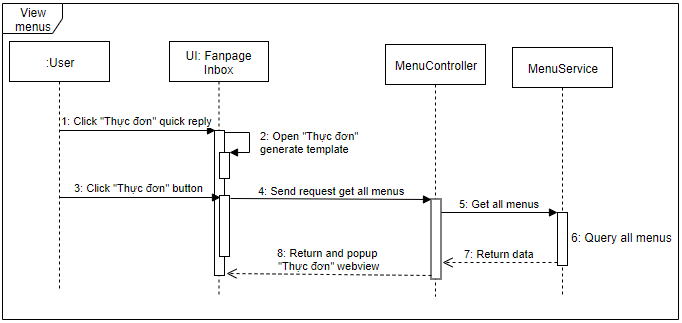


Figure 2.17 View menus sequence diagram

* 1. **Class diagram**

My application has nine classes. In that, Provider class is main class. Relationship between Provider class with Promotion, Menu, AnswerFAQ, Book and Ship class is one to many, because one provider can has many promotions, menus, answers of FAQs, books and orders. Relationship between Customer class with Book and Ship class is also one to many, because the customer can request many delivery times.

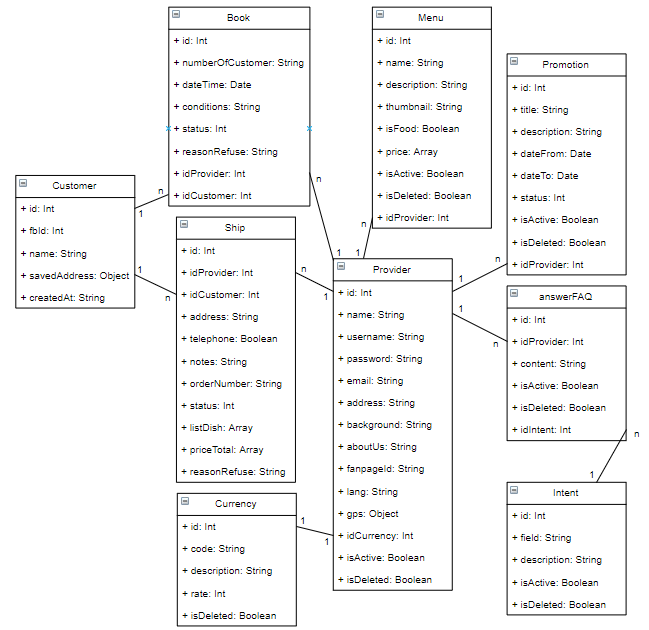


Figure 2.18 Class diagram

# Chapter 3: IMPLEMENTATION RESULTS

This chapter, I will talk about the implementation and results.

* 1. **Manage and show menus**

When Service Provider uses the SPApp to create list of menus. “Thực đơn” screen in SPApp will show list of menus of them.

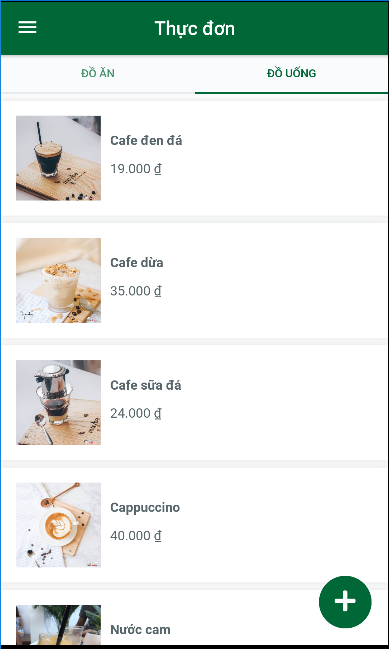


Figure 3.1 “Thực đơn” screen

When user clicks on “Thực đơn” quick reply in fanpage inbox, system will response Menu generic template.

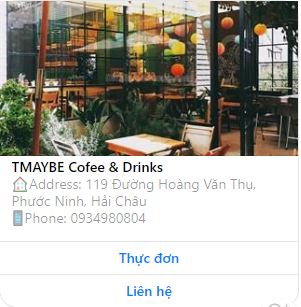


Figure 3.2 Menu generate template

Then user clicks on “Thực đơn” button to open menu webview. Menu webview will show list of menus of provider.

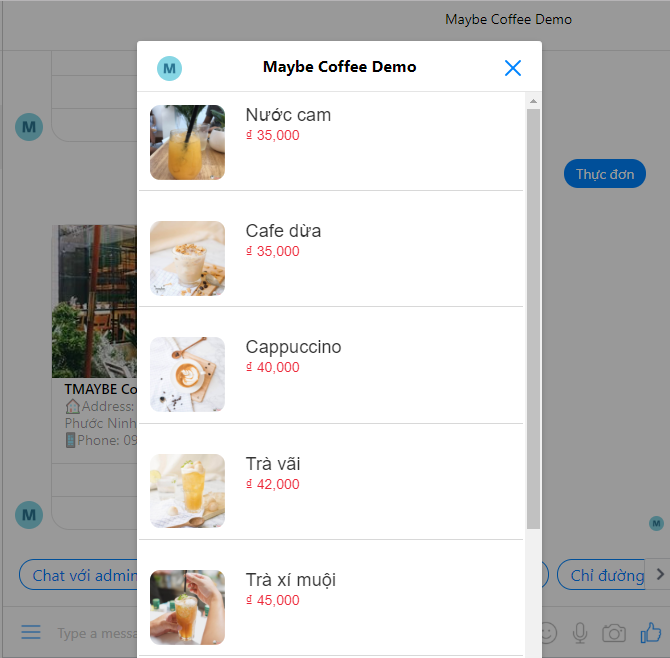


Figure 3.3 Menu webview

* 1. **Manage and show promotions**

When Service Provider uses SPApp create promotions. “Khuyến mãi” screen will show list of promotions.

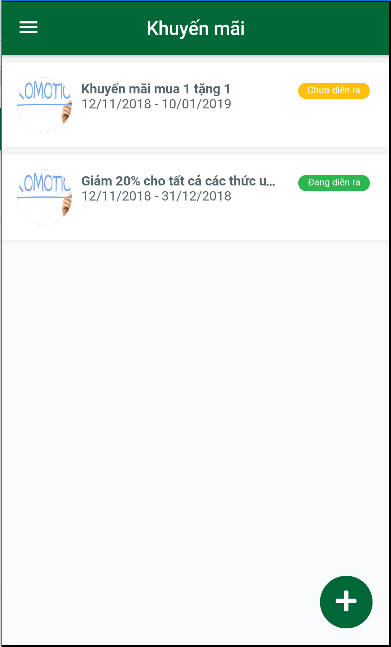


Figure 3.4 Promotion screen

When users click on “Khuyến mãi” quick reply in fanpage inbox, system will response a list of promotion generate templates.

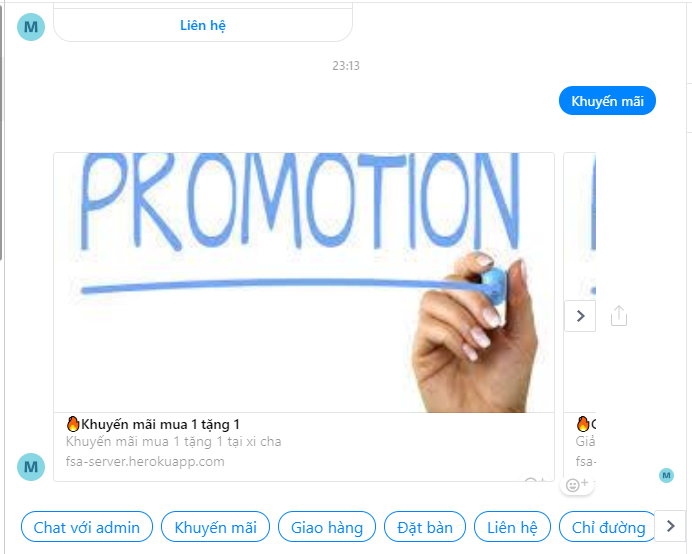


Figure 3.5 List promotion generate templates

A promotion detail webview will show when uses click on a promotion generate template.

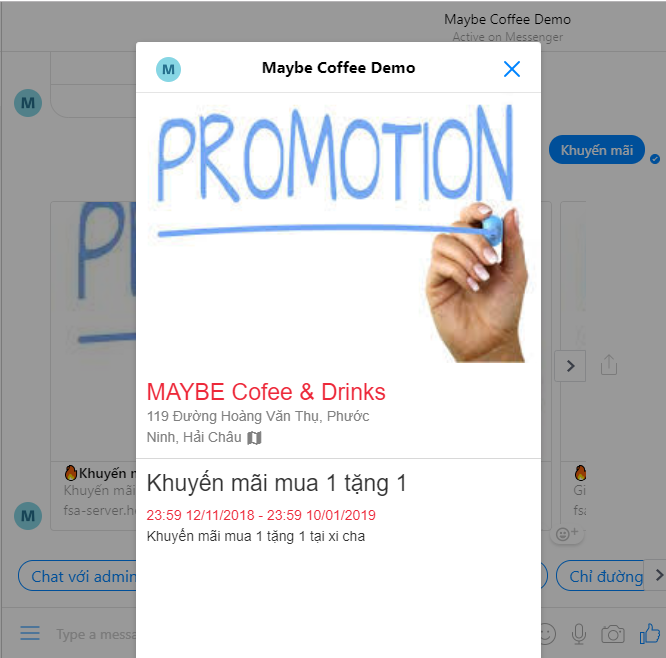


Figure 3.6 Promotion detail webview

* 1. **Request delivery and handle orders**

When users click on “Giao hàng” call to action in persistent menu or click on “Giao hàng” quick reply, system will response Delivery generate template.

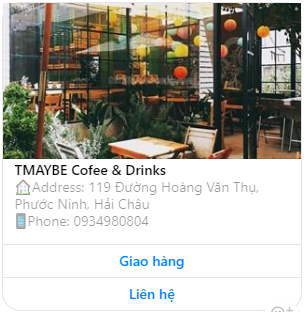


Figure 3.7 Delivery generate template

Then users click on “Giao hàng” button to open Menu webview.

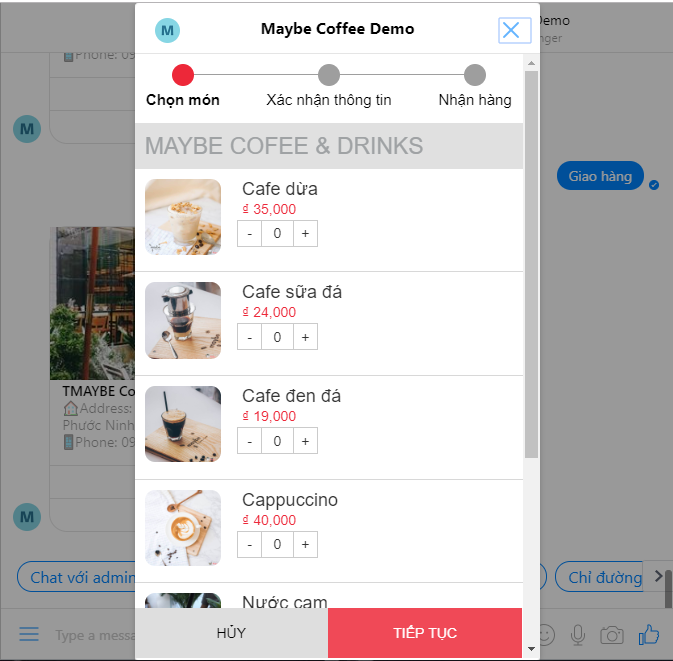


Figure 3.8 Menu webview

After users select item in a list of menus to buy and click “TIẾP TỤC” button, Order Information webview opened to user fill in their address and phone number.

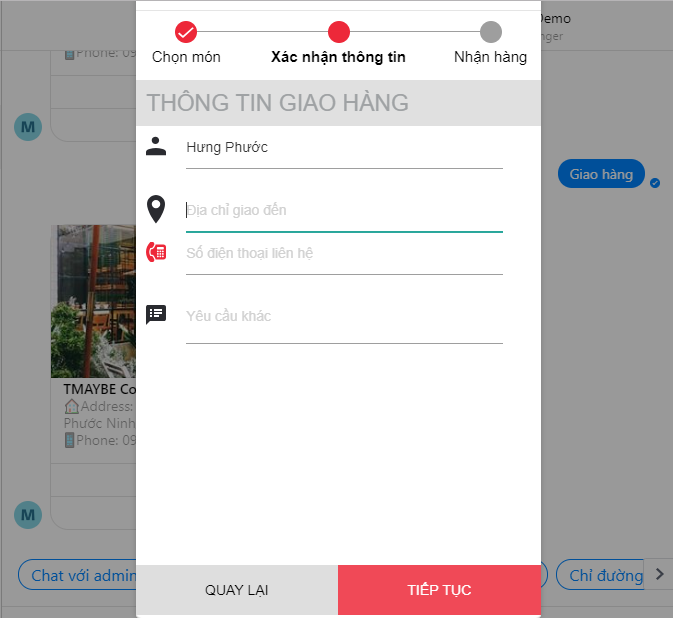


Figure 3.9 Order information webview

When users fill their information and click on “TIẾP TỤC” button, Confirm Information webview will be opened.

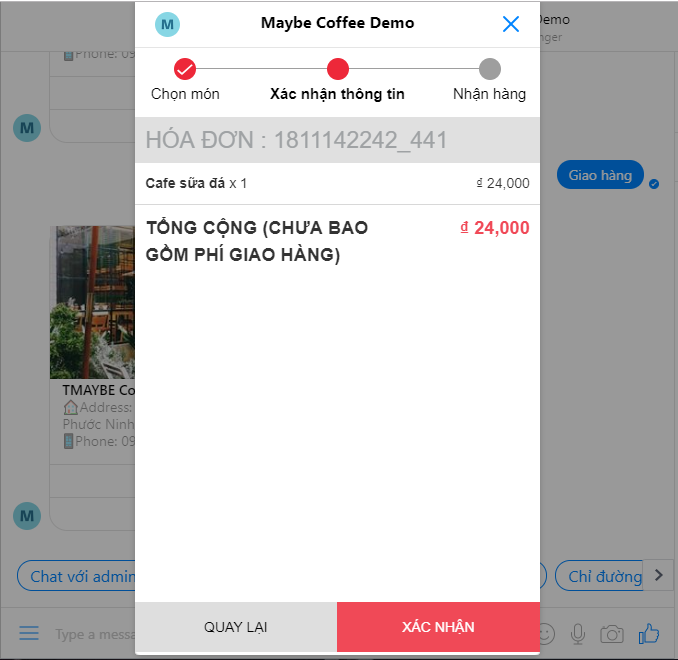


Figure 3.10 Confirm Information webview

After user verify their order again and click “Xác nhận” button to confirm their order.

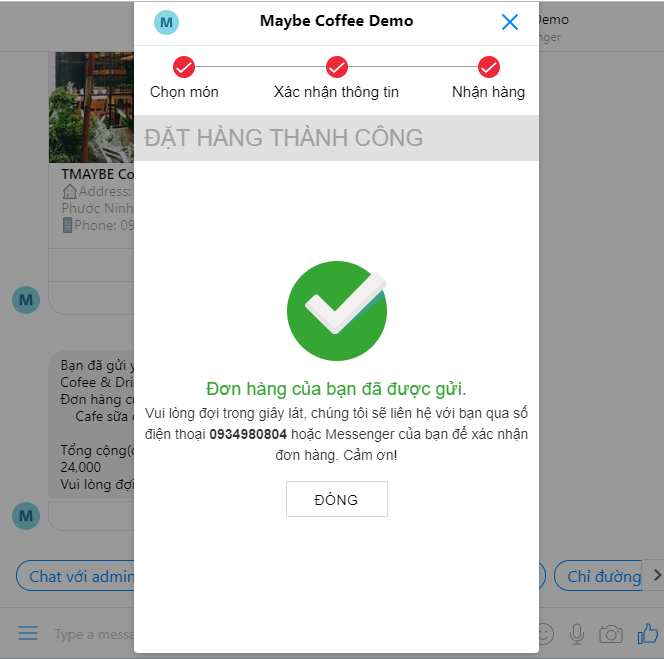


Figure 3.11 Notify webview

In this final step, users click on “Đóng” button to finish their order. After that, a message was sent to user in fanpage inbox.

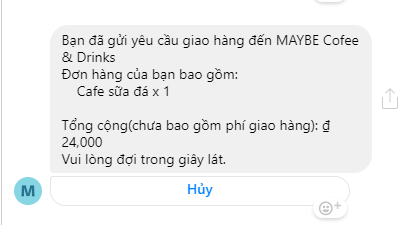


Figure 3.12 Delivery message

If users want to cancel their order, they can click on “Hủy” button.

In SPApp, a list of orders displayed in “Giao hàng” tab in “Danh sách yêu cầu” screen.

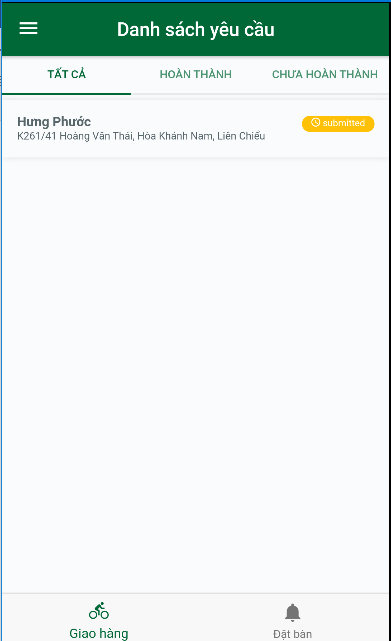


Figure 3.13 “Danh sách yêu cầu” screen

When Service Provider taps on a item in a list of orders, “Thông tin giao hàng” screen will open.

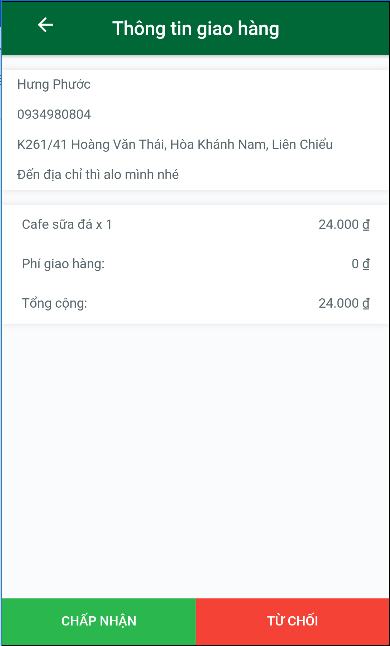


Figure 3.14 “Thông tin giao hàng” screen

In this screen, if Service Provider taps on “CHẤP NHẬN” button, a message will be sent to user.



Figure 3.15 Order acceptance message

If Service Provider taps on “TỪ CHỐI” button, “Từ chối giao hàng” screen will be opened.

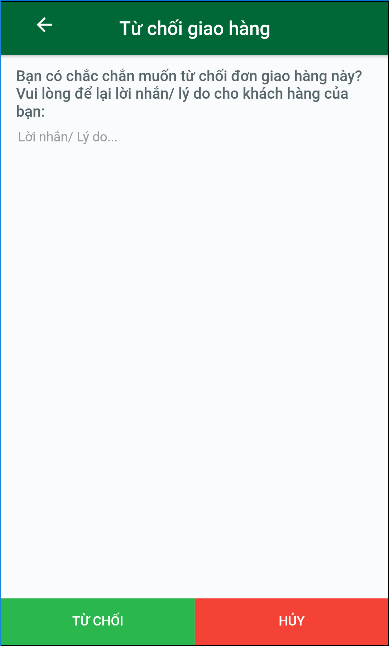


Figure 3.16 “Từ chối giao hàng” screen

In this screen, when Service Provider tap on “TỪ CHỐI” button, a message will be sent to users.

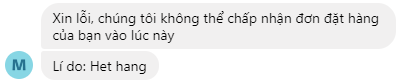


Figure 3.17 Order refusal message

* 1. **Book a table and handle reservations**

When users click on “Đặt bàn” call to action in persistent menu or click on “Đặt bàn” quick reply, system will response Book generate template.

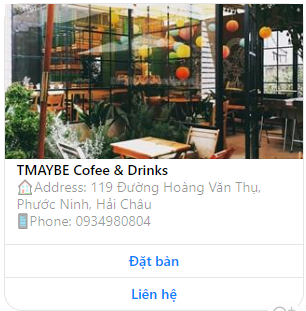


Figure 3.18 Book generate template

Then users click on “Đặt bàn” button to open Book webview.

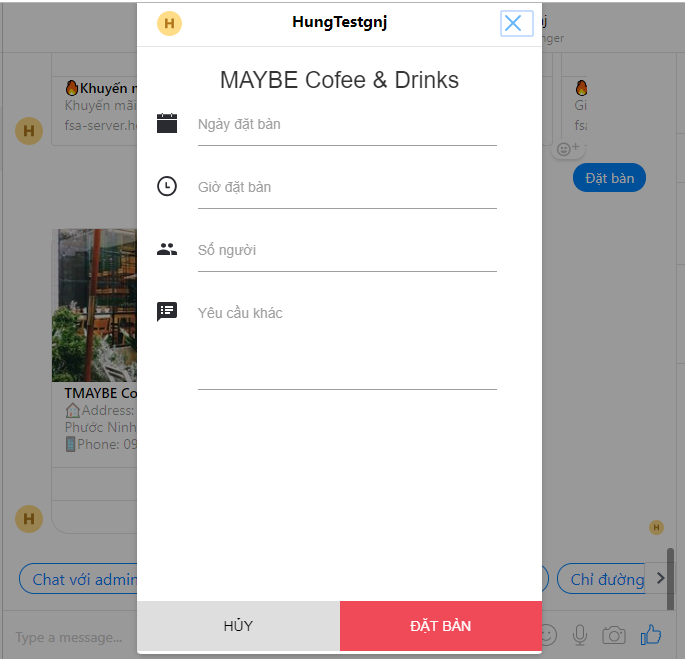


Figure 3.19 Book webview

When users fill reservation information and click on “Đặt bàn” button, system will send a message to user in fanpage inbox.

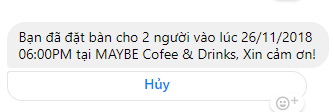


Figure 3.20 Reservation message

If users want to cancel their order, they can click on “Hủy” button.

In SPApp, a list of reservations displayed in “Giao hàng” tab in “Danh sách yêu cầu” Screen.

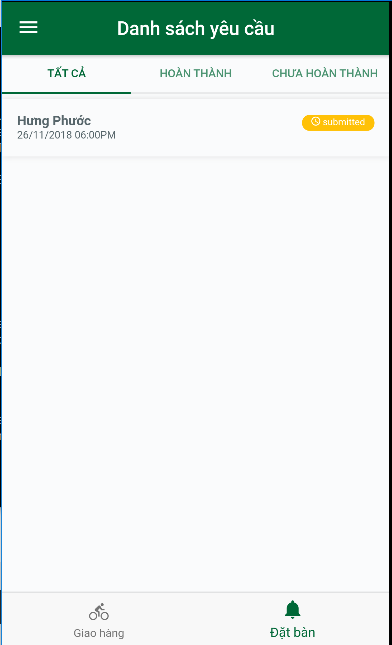


Figure 3.21 “Danh sách yêu cầu” screen

When Service Provider taps on a item in a list of reservations, “Thông tin đặt bàn” screen will be opened.

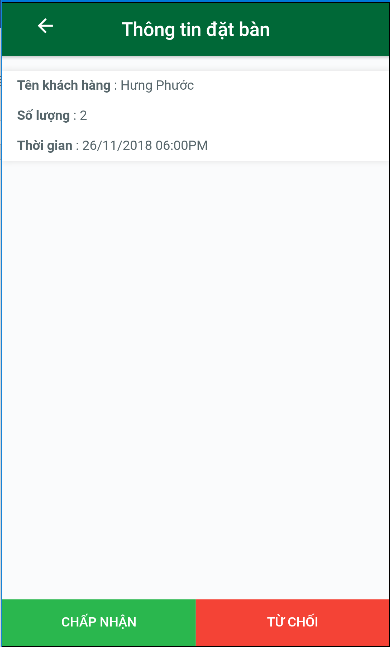


Figure 3.22 “Thông tin đặt bàn” screen

In this screen, if Service Provider taps on “CHẤP NHẬN” button, a message will be sent to user.

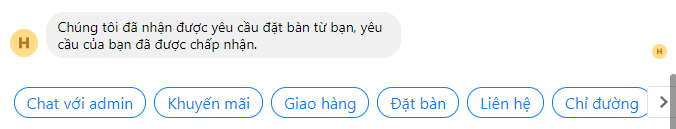


Figure 3.23 Reservation acceptance message

If Service Provider taps on “TỪ CHỐI” button, “Từ chối đặt bàn” screen will be opened.

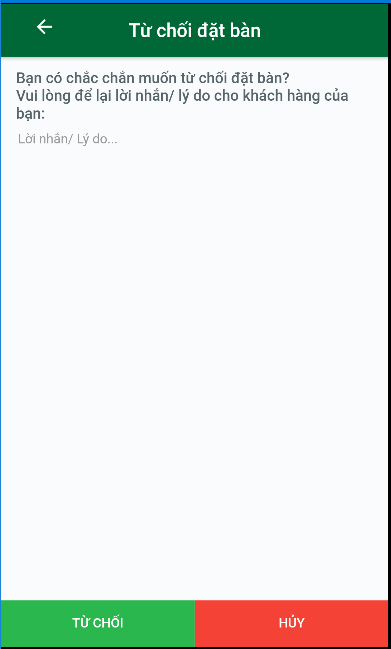


Figure 3.24 “Từ chối đặt bàn” screen

In this screen, when Service Provider tap on “TỪ CHỐI” button, a message will be sent to users.

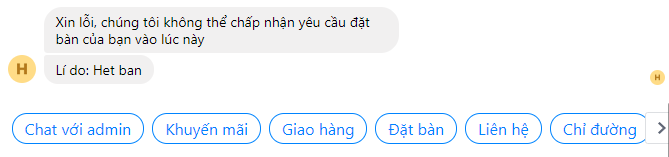


Figure 3.25 Reservation refusal message

* 1. **Frequently asked questions**

When the users input question in fanpage inbox, system will response individual information of SP.

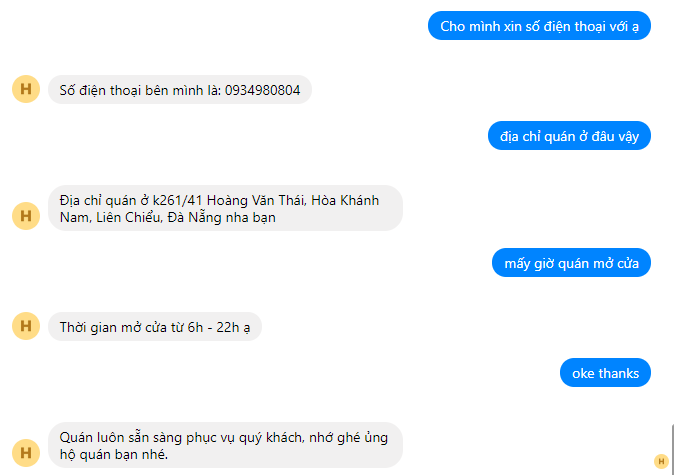


Figure 3.26 Ask FAQs

* 1. **Switch to chat with admin**

When the users click on “Chat với admin” quick reply, sysem send a message to user about chat with admin status is actived.

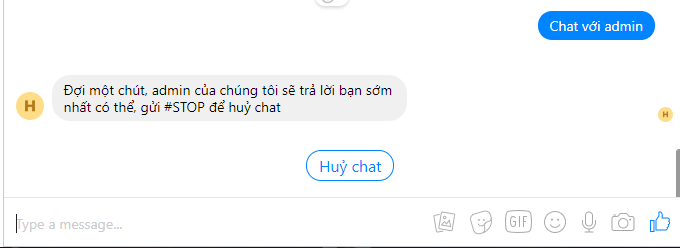


Figure 3.27 Switch to chat with admin

# CONCLUSION AND FUTURE WORK

## Achieve results

In this project, I did learn Javascript language and I could understand how to work with Nodejs platform.

I learned how to program asynchronous with nodejs.

I could use Ionic framwork to build a mobie application.

I also improved researching skill, technical skill, presentation skill, English and other soft skills.

With gained bits of knowledge, I built the project meet initial requirements with achievements as below:

* The application can understand some simple inputs from user and response the right answer.
* The application provides the order system on facebook messenger for food and beverage providers.

However, with advantages, this application still have foibles:

* I did not send a notification to SPApp which is installed mobile phone of service provider when the user requests a delivery or switch to chat with admin status.
* With complex questions from users, system can’t response the right answer.

## Future works

With disadvantages which are mentioned above. In the future, if continued, the project will be developed as below:

* I will send to SPApp a notification when system receives orders from user or users switch to chat with admin status.
* I will write update information function in SPApp to user can update their name, address, phone number …

# REFERENCES

[1] Nodejs documentation. Available on website: <https://nodejs.org/>

[2] Ionic framework documentation. Available on website: https://ionicframework.com

[3] Messenger platform. Available on website: https://developers.facebook.com

[4] MongoDB documentation. Available on website: https://docs.mongodb.com