

**myPesananLIVE: THE DEVELOPMENT OF
FACEBOOK LIVE ORDER MANAGEMENT SYSTEM
USING NATURAL LANGUAGE PROCESSING
APPROACH**

MUHAMMAD DANIAL AIMAN BIN MOHD HANIF

FACULTY OF COMPUTING AND INFORMATICS

UNIVERSITI MALAYSIA SABAH

2022



UMS
UNIVERSITI MALAYSIA SABAH

**myPesananLIVE: THE DEVELOPMENT OF
FACEBOOK LIVE ORDER MANAGEMENT SYSTEM
USING NATURAL LANGUAGE PROCESSING
APPROACH**

MUHAMMAD DANIAL AIMAN BIN MOHD HANIF

**THESIS SUBMITTED IN PARTIAL FULFILLMENT
FOR THE DEGREE OF BACHELOR OF COMPUTER
SCIENCE WITH HONOURS
(NETWORK ENGINEERING)**

FACULTY OF COMPUTING AND INFORMATICS

UNIVERSITI MALAYSIA SABAH

2022



UMS
UNIVERSITI MALAYSIA SABAH

DECLARATION

I hereby declare that the material in this thesis is my own except for quotations, equations, summaries and references, which have been duly acknowledged.

22 February 2022



MUHAMMAD DANIAL AIMAN BIN MOHD HANIF

BI18110242



UMS
UNIVERSITI MALAYSIA SABAH

NAME : MUHAMMAD DANIAL AIMAN BIN MOHD HANIF
MATRIC NO. : BI18110242
TITLE : STUDENT
DEGREE : BACHELOR OF COMPUTER SCIENCE WITH HONOURS
NETWORK ENGINEERING
VIVA DATE : 22 February 2022

CERTIFIED BY,

Pei Yee

SUPERVISOR: DR. CHIN PEI YEE



UMS
UNIVERSITI MALAYSIA SABAH

ACKNOWLEDGEMENT

I would like to express my gratitude towards my lecturers and mentors that gave their best to guide and support me throughout this journey. The pandemic had greatly affected everyone's social norms that leads to severe mental health problems that are still a stigma in our society.

I am grateful for the resources and support were given by my supervisor, Dr Chin Pei Yee, whether it is related to this project or otherwise. I am also grateful for Prof. Jordan Peterson's lectures and his books have empowered me to strive to become a better version of myself every day through finding meaning in what I do. I am also grateful to Dr Tan Soo Fun and Dr Lai Po Hung for sacrificing their energy and time to give me another chance to pass this subject. I cannot express my gratitude enough for being able to be surrounded by supportive people. Thank you.



ABSTRACT

A natural language processing (NLP) integrated Order Management System (OMS) for live streaming merchants are not widely developed, especially for small and medium businesses. The commonly used method for collecting orders from customers is screenshotting the Customer's buying action comments (BACs) along with the snapshot of the live stream for the order context. This method will become much difficult when the number of orders reaches hundreds or thousands, requiring a significant number of human resources. Another problem is that orders through comments can be missed due to typos and incorrect format by customers when commenting, leading to orders not being detected by existing OMS systems. Not many existing OMS were developed with NLP integrated to identify BACs to automate the process of capturing orders. Therefore, a Facebook live order management system with natural language processing was proposed to resolve these issues. Text tokenisation with rule-based approach was utilised to capture orders from Facebook live stream comments. The objective of this project is: (i) To investigate the process of capturing order commands using a combination of rule-based algorithm and regular expression from live video comments and convert them into orders. (ii) To develop a Facebook Live order management system integrated with natural language processing techniques. (iii) To evaluate the system's accuracy in detecting buying action comments from live streams. The expected outcome of this project is a fully functional Facebook live order management system.



UMS
UNIVERSITI MALAYSIA SABAH

ABSTRAK

myPesananLIVE: PEMBANGUNAN SISTEM PENGURUSAN PESANAN LANGSUNG FACEBOOK MENGGUNAKAN PENDEKATAN PEMROSESAN BAHASA ASLI

Sistem Pengurusan Pesanan (OMS) bersepadu pemprosesan bahasa semula jadi (NLP) untuk pedagang penstriman langsung tidak dibangunkan secara meluas, terutamanya untuk perniagaan kecil dan sederhana. Kaedah yang biasa digunakan untuk mengumpul pesanan daripada pelanggan ialah menangkap skrin ulasan tindakan membeli (BAC) Pelanggan bersama-sama dengan petikan strim langsung untuk konteks pesanan. Kaedah ini akan menjadi lebih sukar apabila bilangan pesanan mencecah ratusan atau ribuan, memerlukan sejumlah besar sumber manusia. Masalah lain ialah pesanan melalui ulasan boleh terlepas kerana kesilapan menaip dan format yang salah oleh pelanggan semasa mengulas, menyebabkan pesanan tidak dapat dikesan oleh sistem OMS sedia ada. Tidak banyak OMS sedia ada dibangunkan dengan NLP bersepadu untuk mengenal pasti BAC untuk mengautomatiskan proses menangkap pesanan. Oleh itu, sistem pengurusan pesanan langsung Facebook dengan pemprosesan bahasa semula jadi telah dicadangkan untuk menyelesaikan isu ini. Tokenisasi teks dengan pendekatan berasaskan peraturan digunakan untuk menangkap pesanan daripada ulasan strim langsung Facebook. Objektif projek ini ialah: (i) Untuk menyiasat proses menangkap perintah pesanan menggunakan gabungan algoritma berasaskan peraturan dan ungkapan biasa daripada ulasan video langsung dan menukarnya kepada pesanan. (ii) Untuk membangunkan sistem pengurusan pesanan Facebook Live yang disepadukan dengan teknik pemprosesan bahasa semula jadi. (iii) Untuk menilai ketepatan sistem dalam mengesan ulasan tindakan pembelian daripada strim langsung. Hasil yang diharapkan daripada projek ini ialah sistem pengurusan pesanan langsung Facebook yang berfungsi sepenuhnya.



TABLE OF CONTENTS

TITLE	i
ACKNOWLEDGEMENT	ii
ABSTRACT	v
ABSTRAK	vi
TABLE OF CONTENTS	vii
LIST OF FIGURES	11
LIST OF TABLES	15
CHAPTER 1 : INTRODUCTION	16
1.1 Introduction	17
1.2 Background Study	18
1.3 Problem Statement	19
1.4 Project Objectives	20
1.5 Project Scope	20
1.6 Report Organisation	23
1.7 Conclusion	23
CHAPTER 2 : LITERATURE REVIEW	24
2.1 Introduction	24
2.2 Order Management System (OMS)	24
2.3 Natural Language Processing (NLP)	26
2.4 Text Classification	26
2.4.1 Rule-based	27
2.4.2 Machine learning-based	27
2.4.3 NLP Pipeline	28
2.4.4 Naïve Bayes algorithm	29
2.4.5 Support Vector Machines (SVM)	30
2.4.6 Regular Expression	32



2.5	Application Review	34
2.5.1	Shoplus	34
2.5.2	CommentSold	48
2.5.3	Boxify's FB Live System	61
2.5.4	Comparison of Existing Order Management Systems	70
2.6	Conclusion	74
CHAPTER 3 : METHODOLOGY		75
3.1	Introduction	75
3.2	An Overview of Waterfall model of SDLC	76
	Phase 01: Requirements Gathering and documentation	76
	Phase 02: System Design	77
	Phase 03: Implementation	77
	Phase 04: Testing	77
	Phase 05: Delivery/Deployment	77
	Phase 06: Maintenance	77
3.3	Software and Hardware Requirements	78
3.4	Conclusion	79
CHAPTER 4 : SYSTEM DESIGN		80
4.1	Introduction	80
4.2	Client-server architecture	81
4.3	Server architecture	82
4.4	Use Case Diagram	83
4.5	Entity Relationship Diagram	84
4.5.1	Data Dictionary	85
4.6	Sequence Diagram	93
CHAPTER 5 : RESEARCH IMPLEMENTATION		95
5.1	Introduction	95



5.2	Regular expression	96
5.3	Facebook Graph API	99
5.4	Facebook Messaging API	101
CHAPTER 6 : SYSTEM IMPLEMENTATION		103
6.1	Introduction	103
6.2	Backend (Django)	104
6.2.1	Implementing the entity-relationship diagram into code	104
6.2.2	Creating endpoints for creating, retrieving, updating and deleting data	116
6.2.3	Creating nested url endpoints	125
6.2.4	Creating Order success and cancelled template	126
6.2.5	Implementing webhook to receive updates from Facebook Graph API	127
6.3	Frontend (Flutter)	128
6.3.1	Defining objects in models.dart	128
6.3.2	Defining methods for API client integration	129
6.3.3	Defining routes and bindings	130
6.3.4	Defining pages and controllers	132
CHAPTER 7 : PRELIMINARY TESTING		134
7.1	Introduction	134
7.2	Login	134
7.3	Adding Products	137
7.4	Starting live selling session	138
7.5	Adding keyword	141
7.6	Ordering Product and Checkout Order	142
7.7	Updating Order status	146
CHAPTER 8 : CONCLUSION		148
REFERENCES		149





x

UMS
UNIVERSITI MALAYSIA SABAH

LIST OF FIGURES

Figure 2.2.1	: IBM Sterling Order Management Suite	25
Figure 2.4.1	: Basic Natural Language Processing Pipeline	28
Figure 2.4.2	: Support Vector Machine Hyperplane	31
Figure 2.5.1	: Login Screen	34
Figure 2.5.2	: Shoplus First time Dashboard	35
Figure 2.5.3	: Shoplus Dashboard	36
Figure 2.5.4	: Shoplus Adding Product Form	36
Figure 2.5.5	: Shoplus Inventory Page	37
Figure 2.5.6	: Shoplus Shipping/Payment Settings	38
Figure 2.5.7	: Shoplus Live List Page	39
Figure 2.5.8	: Shoplus Live Comments Page (auto-detect disabled)	40
Figure 2.5.9	: Shoplus Live Comments Page (auto detect enabled)	41
Figure 2.5.10	: Shoplus Automated Message	42
Figure 2.5.11	: Shoplus Order list	43
Figure 2.5.12	: Shoplus Customer Relationship Management	44
Figure 2.5.13	: Shoplus Order details	45
Figure 2.5.14	: Shoplus Order receipt	46
Figure 2.5.15	: Shoplus Order receipt for a customer	47
Figure 2.5.16	: CommentSold Login Page	48
Figure 2.5.17	: CommentSold First time Dashboard	49
Figure 2.5.18	: CommentSold Shop Settings	49
Figure 2.5.19	: CommentSold Payment Gateway Setup	50
Figure 2.5.20	: CommentSold Facebook Account Linking	51
Figure 2.5.21	: CommentSold Add Product Form	52
Figure 2.5.22	: CommentSold Product Variants Edit	53
Figure 2.5.23	: CommentSold Product List Page	54
Figure 2.5.24	: CommentSold Live Overview Page	55
Figure 2.5.25	: CommentSold Broadcast iOS App	56
Figure 2.5.26	: CommentSold Live Overview When During Live Streaming	56
Figure 2.5.27	: CommentSold Automated Message	57
Figure 2.5.28	: CommentSold Webstore Page With Cart	58

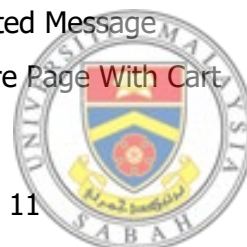


Figure 2.5.29	: CommentSold Orders/Fulfillment page	59
Figure 2.5.30	: CommentSold customers list page	60
Figure 2.5.31	: Boxify's FB Live System login page	61
Figure 2.5.32	: Boxify's FB Live System Dashboard	62
Figure 2.5.33	: Boxify's FB Live System Create Keyword Form	63
Figure 2.5.34	: Boxify's FB Live System Keyword List Page	64
Figure 2.5.35	: Boxify's FB Live System Live Window Page	65
Figure 2.5.36	: Boxify's FB Live System Automated Message	66
Figure 2.5.37	: Boxify's FB Live System Payment Gateway	67
Figure 2.5.38	: Boxify's FB Live System Order Receipt	68
Figure 2.5.39	: Boxify's FB Live System Order details	68
Figure 4.2.1	: Client-Server architecture of myPesananLIVE	81
Figure 4.3.1	: Server architecture of myPesananLIVE	82
Figure 4.4.1	: Use Case Diagram of myPesananLIVE	83
Figure 4.5.1	: Entity Relationship Diagram of myPesananLIVE	84
Figure 4.6.1	: Ordering Sequence Diagram of myPesananLIVE	93
Figure 5.2.1	: Regular expression pattern for BAC	96
Figure 5.2.2	: Regular expression pattern for detecting a keyword	97
Figure 5.2.3	: Regular expression pattern for detecting BAC	98
Figure 5.3.1	: Retrieving a list of livestream using Facebook Graph API	99
Figure 5.3.2	: Retrieving real-time comments from a Facebook live stream	100
Figure 5.4.1	: Replying to Customer's comment using Facebook Messaging API	101
Figure 5.4.2	: The reply received in a Customer's inbox	102
Figure 5.4.3	: Facebook message with button template	102
Figure 6.2.1	: Django app boilerplate files	104
Figure 6.2.2	: Implementation of Comment object in models.py	105
Figure 6.2.3	: Implementation of Customer object in models.py	106
Figure 6.2.4	: Implementation of Livestream and Livestream product object in models.py	107
Figure 6.2.5	: Order object in models.py	108
Figure 6.2.6	: Implementation or Order item object in models.py	109
Figure 6.2.7	: Implementation of Payment object in models.py	109
Figure 6.2.8	: Implementation of Product object in models.py	110



Figure 6.2.9	: Implementation of Shop object in models.py	111
Figure 6.2.10	: Serializer for Comment object in serializers.py	112
Figure 6.2.11	: Serializer for Customer object in serializers.py	113
Figure 6.2.12	: Serializer for Livestream and Livestream Product object in serializers.py	113
Figure 6.2.13	: Serializer for Order and Order Item object in serializers.py	114
Figure 6.2.14	: Serializer for Payment object in serializers.py	114
Figure 6.2.15	: Serializer for Product object in serializers.py	115
Figure 6.2.16	: Serializer for the Shop object in serializers.py	115
Figure 6.2.17	: CommentViewSet class defined in views.py	116
Figure 6.2.18	: CustomerViewSet class defined in views.py	117
Figure 6.2.19	: LiveStreamViewSet class defined in views.py	118
Figure 6.2.20	: LiveStreamProductViewSet class defined in views.py	119
Figure 6.2.21	: OrderViewSet class defined in views.py	119
Figure 6.2.22	: update_item method in OrderViewSet class	120
Figure 6.2.23	: checkout method in OrderViewSet class	121
Figure 6.2.24	: success method in OrderViewSet class	122
Figure 6.2.25	: cancel method in OrderViewSet class	123
Figure 6.2.26	: OrderItemViewSet class defined in views.py	124
Figure 6.2.27	: PaymentViewSet class defined in views.py	124
Figure 6.2.28	: ShopViewSet class defined in views.py	125
Figure 6.2.29	: URL endpoints defined in urls.py file	125
Figure 6.2.30	: Success.html and cancel.html template file for order success or cancelled	126
Figure 6.2.31	: Webhook implementation for Facebook API	127
Figure 6.3.1	: Shop object defined in models.dart	128
Figure 6.3.2	: MyPesananLIVEApiClient class defined in api.dart	129
Figure 6.3.3	: Routes defined in Routes class in routes.dart file	130
Figure 6.3.4	: AppPages class defined in pages.dart file	131
Figure 6.3.5	: Bindings for Setup page in bindings.dart file	131
Figure 6.3.6	: SignIn class defined in page.dart	132
Figure 6.3.7	: AuthController defined in controller.dart	133
Figure 7.2.1	: Login screen of myPesananLIVE	134
Figure 7.2.2	: Facebook Page (Shop) selection to integrate myPesananLIVE	135



Figure 7.2.3	: myPesananLIVE Dashboard	136
Figure 7.3.1	: Product page of myPesananLIVE	137
Figure 7.3.2	: Add product form of myPesananLIVE	138
Figure 7.4.1	: Livestream page of myPesananLIVE	139
Figure 7.4.2	: Livestream Detail page of myPesananLIVE	140
Figure 7.5.1	: Add keyword page of myPesananLIVE	141
Figure 7.6.1	: Customer buying action comment on Facebook LIVE stream	142
Figure 7.6.2	: Order message of myPesananLIVE send to Customer	143
Figure 7.6.3	: Confirming order via Facebook Messaging	144
Figure 7.6.4	: Stripe checkout page	144
Figure 7.6.5	: Order success page	145
Figure 7.6.6	: Product stock deducted from the order	145
Figure 7.7.1	: Order page of myPesananLIVE	146
Figure 7.7.2	: Update status and tracking number popup	147
Figure 7.7.3	: Status update via Facebook Messaging to Customer	147



LIST OF TABLES

Table 1.5.1 An Overview of myPesananLIVE Modules	22
Table 2.4.6.1 Regular expression examples	32
Table 2.5.4.1 Comparison of Shoplus, CommentSold, and Boxify's FB Live System	70
Table 2.5.4.2 Comparison of Shoplus, CommentSold, Boxify's FB Live System and myPesananLIVE	72
Table 3.3.1 Hardware specification	78
Table 3.3.2 Software specification	78
Table 4.5.1.1 Customer Table	85
Table 4.5.1.2 Order Table	86
Table 4.5.1.3 Order Item Table	87
Table 4.5.1.4 Payment Table	88
Table 4.5.1.5 Comment Table	89
Table 4.5.1.6 Livestream Table	90
Table 4.5.1.7 Livestream Product Table	90
Table 4.5.1.8 Shop Table	91
Table 4.5.1.9 Livestream Table	92



LIST OF ABBREVIATIONS

- OMS** - Order Management System
BAC - Buying Action Comment



CHAPTER 1

INTRODUCTION

1.1 Introduction

Social media platforms such as Youtube, Facebook, Instagram and Twitter are some of the spaces that allow users to get information about products and services. These platforms are suitable to develop social commerce that will enable consumers to experience the online shopping, purchase products and share information (Liang et al., 2011; Nakayama & Wan, 2019). Wham (2018) discovered that 76% of consumers are likely to purchase products or services on social media. In that regard, businesses can adopt social commerce to improve their sales and services.

Live streaming selling has also shown a significant increase since the coronavirus disease 2019 (COVID-19) pandemic. (iiMedia Research, 2020) reported that live streams increased by 120% in China during the pandemic. This increase has yielded more than approximately RM 69.9 million in revenue through live stream selling by Luo Yonghao, the smartphone maker Smartisan. Taobangdan (2019) also reported that in 2018, live streaming selling had helped online vendors to achieve over 100 billion Yuan in sales. The increasing trend of live stream selling has transformed social commerce in various ways. For example, traditionally, customers rely on text and pictures to evaluate products online. However, live streaming allows consumers to know more about the product in real-time and allows them to know more in detail about the product's information such as by demonstrating how the product is used, show the product from different perspectives, answer customer questions and

entertain customers to encourage them to buy immediately (Lu et al., 2018; Wongkitrungrueng & Assarut, 2020).

1.2 Background Study

Facebook allows live streamers to conduct live selling that requires the streamer to register the product by uploading its photo and details such as title, price and description. When the streamer goes live, the registered products can be chosen to be featured on stream so that the viewers can click on it to know more in detail and purchase it. Instagram also has this feature but is only limited to eligible businesses in the US (Warren, 2020). Other platforms such as Youtube lacks this feature. An alternative method would be customers commenting in the live comments to show buying the product shown in the live stream. This way, merchants would save time by not registering the products they want to pitch before the session, especially when the number of catalogues is vast. However, this alternative method has a significant number of cumbersome procedures for the Merchant. Firstly, the Merchant or assistant need to screenshot the Customer's buying action comment (BACs) that contains the Customer's account name, phone number (if given), product name, quantity and details (if applicable) along with the snapshot of the product from the live stream. The assistant then notes the Customer's name and phone number given in the comments to be further contacted.

With the increased usage of live streams, not many systems are being developed to help businesses manage their orders smoothly. The Merchant (and with the help of their assistants) had to manually manage the orders by hand that requires a considerable number of human resources, especially when the session has high number of customers and the quantity of orders reaches thousands or even the rate of comments appearing on the Merchant's view is too fast. This situation will also increase the chance of missing taking note of orders from the comments. Therefore, automation is required to minimise the work needed to manage orders through live stream selling. Not many order



management systems (OMS) offer automation developed. The only notable commercial ones are Shoplus which claimed to be using Artificial Intelligence (AI) to process comments to detect BACs (Nu, 2019); CommentSold that, detects any comments that contain the word 'Sold' as buying action (Melendy, 2018); and Boxify FB Live System. These solutions can only be integrated with Facebook and Instagram and only supports English as the primary language to detect BACs. Not much is known about how the systems work as they are proprietary and not open-source. As of now, there are no articles that explain in detail how these systems utilise AI to determine BACs. We suspect that they use natural language processing (NLP) to process these texts, but what pipeline and types of algorithms they use is unknown.

Russell et al. (2016) summarise that AI systems should have the capability of natural language processing (NLP) to communicate in a natural language, knowledge representation to store information, automated reasoning to use the stored information to answer questions and to draw conclusions, and machine learning for adaptation of new environments and to identify and extrapolate patterns (Huang & Rust, 2018; Russell et al., 2016). Loureiro et al. (2021) cited that applying AI to businesses will maximise the efficiency of the working environment by taking over repetitive tasks (Huang et al., 2019). Thus, using AI, particularly NLP, in live streaming selling could potentially reduce the workload by merchants so that any buying intents and orders are immediately consolidated and prepared for the Merchant and logistics. Unfortunately, there are not many papers that study the use case of NLP in live stream selling or OMS to study its effects on the customer experience and the order management efficiency.

1.3 Problem Statement

Order management systems are necessary to manage orders, especially when orders reach thousands when conducting social e-commerce or live streaming selling. Not many existing OMS integrated NLP to convert BACs from live



streaming selling to orders and had to rely on a manual solution that requires a significant number of resources. This lack of integration will delay the shipment of the items to the Customer due to the amount of time needed to consolidate the orders manually. Some orders might mistakenly not be taken. Another problem is that orders through comments can be missed due to typos and incorrect format by customers when commenting, leading to charges not being detected by existing OMS systems.

1.4 Project Objectives

Three objectives need to be achieved in this study which are:

- i. To investigate the process of capturing order commands using a combination of rule-based algorithm and regular expression from live video comments and convert them into orders.
- ii. To develop a Facebook Live order management system integrated with natural language processing techniques.
- iii. To evaluate the system's accuracy in detecting buying action comments from live streams.

1.5 Project Scope

- myPesananLIVE will have an authentication module that integrates Facebook Login API to retrieve the access token required to send requests to Facebook Graph API.
- myPesananLIVE will include the following models: User, Customer, Live Stream, Product, Order, Invoice and Payment, which are essential for the core functionality of the OMS.
- Merchants will be the primary user of this project and can start a live selling session from a live stream and add a product.
- myPesananLIVE frontend will include the dashboard, orders, products, live stream and customers page.
- Customers will only interact with myPesananLIVE indirectly through the Merchant's Facebook Live Stream to create orders and confirm or cancel an order through Facebook messaging.



- Facebook Live comments will be processed by myPesananLIVE to identify ordering comments.
- Stripe API is integrated to act as the payment gateway for the Merchant.
- myPesananLIVE runs primarily on the web browser and will support cross-platform usage due to its framework.



Table 1.5.1 An Overview of myPesananLIVE Modules

Module	Description	Targeted Users
Login	Login endpoint for merchants and admins	Merchants, Admin
Dashboard	Shows the basic information of the shop such as last stream performance, activities and latest orders.	Merchants
Product	Defines the details of the product, views the list of products, and do CRUD operations on products.	Merchants
Order	Defines the order details, view the list of orders, and update an order's status.	Merchants
Customer	Record the details of customers after an order has been created and do CRUD operations on customers.	Merchants
Live stream	Starts the live selling session from an ongoing Facebook Live session, views live comments, detect BACs from live comments, and store past live stream details.	Merchants
Facebook Messaging	To confirm, checkout or cancel orders	Customer
Admin dashboard	Strictly for the administrator of myPesananLIVE for CRUD operations of all models	Admin

1.6 Report Organisation

Chapter 1 briefly introduces live eCommerce and the study's background.

Chapter 2 introduces some reviews on the characteristics of OMSs, natural language processing techniques, and several existing Facebook Live OMSs.

Chapter 3 describes the methodology of the project in detail.

Chapter 4 explains the system design of the project.

Chapter 5 presents the research experimentation and implementation of the project.

Chapter 6 presents the system implementation of the project.

Chapter 7 presents project testing.

Chapter 8 presents future works and the conclusion of the project.

1.7 Conclusion

A Facebook live order management system with natural language processing can solve the high number of human resources needed and prevent orders from being missed. It can also solve undetected orders by existing OMS. This project aims to solve these issues by choosing a suitable NLP library to create a better and more robust OMS.

