



**UNIVERSITY OF
PLYMOUTH**



**PENINSULA
COLLEGE**

School of Engineering,
Computing, and Mathematics

MAL3018

Computing Project

2024/2025

SHIPMATE

Students Peer-to-Peer Service System

Ng Man Yew

BSSE2309754

BSc (Hons) Computer Science (Software Engineering)

DECLARATION

I hereby declare that this project report entitled "Shipmate: Student Peer-to-Peer Service System" is the result of my own work except as cited in the references. It has not been submitted in whole or in part for any degree or diploma to any other institution.

Ng Man Yew

BSC (Hons) Computer Science (Software Engineering)

Peninsula College Georgetown

Date: April 21, 2025

ACKNOWLEDGEMENT

First and foremost, I would like to express my sincere gratitude to my project supervisor, Miss Nadhrah Abdul Hadi, for the valuable guidance, encouragement, and continuous support throughout the development of this project. Their insights and feedback have been instrumental in shaping the Shipmate system from idea to reality.

I also extend my appreciation to my lecturers and peers at The Ship Campus for their motivation and contributions during various phases of this journey. Special thanks to my family for their unwavering patience, understanding, and support during times of personal and technical challenges.

Lastly, I would like to thank everyone who participated in the development and testing process, as well as those who helped indirectly through discussions, resources, and shared experiences. This accomplishment would not have been possible without your involvement.

ABSTRACT

This report presents Shipmate, a web-based peer-to-peer service platform designed specifically for students of The Ship Campus. The system allows users to offer and request services within a secure and student-verified environment. Key features include real-student verification, service listing, service application with status tracking, a messaging system, and an anonymous rating and review system. The project follows a structured Web Development Lifecycle (WDLC), encompassing research, design, implementation, and testing phases. It adopts client-server architecture using Next.js (React) on the frontend and Node.js with Express on the backend, with MongoDB as the database and Auth0 for secure authentication. Manual and automated testing were conducted to assess system reliability. Out of 21 manual test cases, 20 passed and 1 failed (non-real-time messaging). Additionally, all 31 unit tests for backend controllers passed successfully. While the system successfully delivers its core functions, some limitations such as the absence of real-time chat and admin role management were identified. Future enhancements are proposed to address these. The development of Shipmate has not only achieved its technical objectives but also contributed significantly to the developer's growth in full-stack development, architectural planning, and resilience in overcoming personal and technical challenges.

Table of Contents

DECLARATION	1
ACKNOWLEDGEMENT	2
ABSTRACT	3
LIST OF FIGURES	6
LIST OF TABLES	8
LIST OF ABBREVIATIONS	9
CHAPTER 1: INTRODUCTION	10
1.1 Project idea and background	10
1.2 Problem Statement	11
1.3 Objectives	13
1.4 Project Scope	14
1.5 Significance of the Project	16
1.6 Limitation of the Project	17
CHAPTER 2: LITERATURE REVIEW	19
2.1 Introduction	19
2.2 Peer-to-peer(P2P) Services	19
2.3 Digital Business Model and Marketplaces	22
2.4 Research Area.....	24
2.5 Conclusion	35
CHAPTER 3: LEGAL, SOCIAL, ETHICAL, PROFESSIONAL CONSIDERATIONS	36
3.1 Introduction	36
3.2 Legal Considerations	37
3.3 Social Considerations	38
3.4 Ethical Considerations	39
3.5 Professional Considerations	40
3.6 Conclusion	41
CHAPTER 4: SYSTEM DESIGN AND ARCHITECTURE	42
4.1 Introduction	42
4.2 System Architecture	42
4.3 Technology	46
4.4 Methodology	47

4.5	Database Design	51
4.6	System Process Flowcharts	55
4.7	Conclusion	58
CHAPTER 5: IMPLEMENTATION		59
5.1	Introduction	59
5.2	Development Environment	60
5.3	Server-side Implementation	63
5.4	Client-Side Implementation	72
5.5	Security Considerations	84
5.6	Conclusion	86
CHAPTER 6: TESTING		87
6.1	Introduction	87
6.2	Manual Testing	87
6.3	Automated Testing	98
6.4	Conclusion	102
CHAPTER 7: CONCLUSION		103
7.1	Project Summary	103
7.2	Future Enhancements	104
7.3	Personal Growth and Learning Experience	104
7.4	Special Acknowledgements	105
7.5	Final Thoughts	105
REFERENCES		106

LIST OF FIGURES

Figure 1.1 Likelihood of using the Shipmate platform for service exchange	12
Figure 1.2 Importance of Identity Verification for Users on the Shipmate Platform	12
Figure 1.3 Significance of Rating and Review Systems for Service Providers and Seekers on the Shipmate Platform	13
Figure 2.1 Stocket website main pages (Stocket.com, 2024)	24
Figure 2.2 Stocket mobile responsive UI (Stocket.com, 2024)	25
Figure 2.3 Seller Verification (Stocket.com, 2024)	26
Figure 2.4 TaskRabbit website main page (Taskrabbit.com, 2024)	27
Figure 2.5 TaskRabbit project quick browse (Taskrabbit.com, 2024)	28
Figure 2.6 Fiverr website main page (Fiverr.com, 2024)	30
Figure 2.7 Fiverr item listing from category Graphic & Design (Fiverr.com, 2024)	31
Figure 2.8 Fiverr massive competitors on single categories (Fiverr.com, 2024)	32
Figure 4.1 System Architecture Diagram	43
Figure 4.2 Web Development Lifecycle methodology overview	47
Figure 4.3 Shipmate Entity Relationship Diagram	53
Figure 4.4 Service Posting Flowchart.....	55
Figure 4.5 Service Application Flowchart	56
Figure 4.6 Real Student Verification Process Flow	57
Figure 5.1 Shipmate coding organized structure	61
Figure 5.2 GitHub version control evidence	62
Figure 5.3.1 Service Controller coding snippet	63
Figure 5.3.2 User Controller coding snippet	64
Figure 5.3.3 Verification Controller coding snippet	64
Figure 5.3.4 Chat Controller coding snippet	65
Figure 5.4.1 userRoutes.js coding snippet	67
Figure 5.4.2 serviceRoutes.js coding snippet	67
Figure 5.4.3 chatRoutes.js coding snippet	68
Figure 5.4.4 verificationRoutes.js coding snippet	68
Figure 5.5.1 user schema example snippet	69
Figure 5.5.2 service schema example snippet	70
Figure 5.5.3 verification schema example snippet	70
Figure 5.5.4 message schema example snippet	71
Figure 5.6.1 find service page snippet	73
Figure 5.6.2 find service with filter snippet	74
Figure 5.6.3 My Service Post tab snippet	75
Figure 5.6.4 My Applicant tab with no applicant snippet	76
Figure 5.6.5 My Applicant tab with new applicant snippet	76
Figure 5.6.6 Requested Service tab snippet	77

Figure 5.6.7 Saved Services tab snippet	78
Figure 5.6.8 custom select dropdown for mobile's tab navigation snippet	78
Figure 5.6.9 Service Posting Page for unverified user snippet	79
Figure 5.6.10 Service Posting Page for verified user snippet	79
Figure 5.6.11 Service Rating Page snippet	80
Figure 5.6.12 Service Details Page snippet	81
Figure 5.6.13 Student Verification Page snippet	82
Figure 5.6.14 Messaging Page example snippet	83
Figure 5.7.1 auth0 login page snippet.....	84
Figure 5.7.2 Anonymous reviewer snippet	86
Figure 6.1 db-setup for testing snippet	98
Figure 6.2 mock-object setup for testing snippet	99
Figure 6.3 Service Controller Testing snippet	100
Figure 6.4 Verification Controller Testing snippet	100
Figure 6.5 Verification Controller Unit Testing Results	101
Figure 6.6 Service Controller Unit Testing Results	102

LIST OF TABLES

Table 2.1 Comparison of existing products	33
Table 4.1 Shipmate System Architecture’s layer overview	43
Table 4.2 Client Layer Modular Components	44
Table 4.4 Server Layer Components	45
Table 4.4 Shipmate Database Main Collections Summary	51
Table 4.5 Main Collections Relationship Summary	52
Table 4.6 ERD justifications	54
Table 5.1 Software and tools used during development	60
Table 5.2 routes functionalities	65
Table 5.3 find page functional layout	74
Table 6.1.1: Browse Service as unauthorized user	88
Table 6.1.2: Bookmark service as unauthorized user	88
Table 6.1.3: Apply for a service as unauthorized user	89
Table 6.1.4: Attempt to access “My Services” or “Post Service” as unauthorized user ..	89
Table 6.2.1: Browse Service	90
Table 6.2.2: Bookmark Service	90
Table 6.2.3: Unsave a Service	91
Table 6.2.4: Apply for a Service	91
Table 6.2.5: Search Services by Keyword	91
Table 6.2.6: Search Services by Filter	92
Table 6.2.7: View Requested Services as Authorized User	92
Table 6.2.8: Message provider from a accepted service	93
Table 6.2.9: Mark an Accepted Service as Completed	93
Table 6.2.10: Rating after service completion – input validation	94
Table 6.2.11: Rating after service completion – input validation	94
Table 6.2.12: Accessing Post Service page as Authorized but Unverified User	95
Table 6.2.13: Verification with invalid Credentials	95
Table 6.3.1: Attempt to Post a Service with Empty Fields	96
Table 6.3.2: Attempt to Post a Valid Service	96
Table 6.3.3: Attempt to Apply to Own Service	97
Table 6.3.4: View and Manage Applicants for Own Service	97
Table 6.4 Unit Testing Components.....	99