

Group members

N1051811

T0321003

T0315059

T0293214

Nuno Rodrigues Amalio

The following PLAGIARISM DECLARATION must be entered into the report and the name and ID completed:

“This report and the software it documents are the result of my own work. Any contributions to the work by third parties, other than tutors, are stated clearly below this declaration. Should this statement prove to be untrue I recognise the right and duty of the Board of Examiners to take appropriate action in line with the university’s regulations on assessment.

Name: Al Khattab Saif Abdullah AL-Sadi ID No: T0293214

Name: Iruobe Akpatason ID No: N1051811

Name: Rick Richard Harith ID No: T0321003

Name: Ayesheh Bkeirat ID No: T0315059

Git Repository: <https://olympuss.ntu.ac.uk/N1051811/SDIuserInterface>

Abstract

This project for System Analysis and Design (SOFT20091) was to design and develop an e-transport marketplace for shipping packages. The chosen platform and language used for the development of this application are QT creator, QT designer and C++ as well as Docker for development environment.

For this project we followed the software development lifecycle of Planning, Analysis, Design and implementation. The basic idea of this application is to allow the users to create accounts with the role of a customer, driver, or a company. Customers can place orders for delivery, providing the weight, dimensions, pickup and drop off destination of their package. This user also receives an estimated cost for the delivery as well as a receipt and a notification confirming delivery. The company can then receive the customer's order and forward them to the driver to be accepted or rejected. The driver can then accept the order and give the user an update on the delivery status of the package.

Plagiarism Declarations

I, Iruobe Akpatason, declare that this report and software documented are a result of our own work. All contributions made by third parties are stated clearly below and reference. I take full responsibility for any errors, inaccuracies, and omissions.

Name: Iruobe Akpatason. NTU ID: N1051811

I, AL Khattab, declare that this report and software documented are a result of our own work. All contributions made by third parties are stated clearly below and reference. I take full responsibility for any errors, inaccuracies, and omissions.

Name: AL Khattab. NTU ID: T0293214

I Rick Richard Harith, declare that this report and software documented are a result of our own work. All contributions made by third parties are stated clearly below and reference. I take full responsibility for any errors, inaccuracies, and omissions.

Name: Rick Richard Harith. NTU ID: T0321003

Revision History

Version	Issue Date	Stage	Changes	Author
5f21d4010aac8 7e16a49201e2 80aaae5fb8b97 30	Wed Apr 5 12:35:27 2023	in progress	Created login UI	Iruobe Akpatason <N1051811@ntu.ac.uk>
6b16212bcc56d 94574869a2ccb bc731ce342605 5	Thu Apr 6 12:24:49 2023	in progress	Created newUser window UI	Iruobe Akpatason <N1051811@ntu.ac.uk>
71b4354c742c6 ad72d6a21ac3e feb3b65b9b113 3	Fri Apr 7 15:14:34 2023	in progress	Created userregistration UI	Iruobe Akpatason <N1051811@ntu.ac.uk>
93e3c058b72f4 8012b986b771 5bc5a3b1fb233 79	Fri Apr 7 15:14:45 2023	in progress	Created userregistration UI	Iruobe Akpatason <N1051811@ntu.ac.uk>
6aa2b78830bc6 437172719f2ea f3439156d7eec 5	Fri Apr 7 16:20:15 2023	in progress	Update userregistration. ui	N1051811 <iruobe.akpatason2021@my.ntu.ac.uk>
558687597664 b1403a5a6e35 b5b877603976 861c	Fri Apr 7 15:57:34 2023	in progress	connected the user registration button	Iruobe Akpatason <N1051811@ntu.ac.uk>

2ecbe5c1c412d 09f90b0c3841a 36b5229da671 e3	Fri Apr 7 17:29:32 2023	in progress	Created registration page for driver	Iruobe Akpatason <N1051811@ntu.ac.uk>
d78c874fafbc59 c8909d4500aff4 ff8a399e8ef0	Fri Apr 7 18:12:17 2023	in progress	Created registration page for driver	Iruobe Akpatason <N1051811@ntu.ac.uk>
553a6e9c35369 5027fa74b1634 ac045f6fa0f4da	Fri Apr 7 18:21:27 2023	in progress	Created registration page for driver	Iruobe Akpatason <N1051811@ntu.ac.uk>
80f2635a99226 c2fa3b2a83f598 a0a847038e6ac	Sat Apr 8 09:11:32 2023	in progress	Added lorry type to driver registration	Iruobe Akpatason <N1051811@ntu.ac.uk>
e79224c53df82 1f4eddbc50e6a 750c0069ace98 8	Sat Apr 8 09:11:42 2023	in progress	Added lorry type to driver registration	Iruobe Akpatason <N1051811@ntu.ac.uk>
c9cafb3d63d36 7edb67aa43c2b d7fcc99e6bdcb e	Sat Apr 8 11:09:26 2023	in progress	Added created company registration page	Iruobe Akpatason <N1051811@ntu.ac.uk>
a98ce445afa58 6d5b4271f099c	Sat Apr 8 11:24:17 2023	in progress	Added label to driver	Iruobe Akpatason

97cdd3f39b83aa			registration for lorry type	<N1051811@ntu.ac.uk>
16347e11970161bdddc3102662b5d1b148bb8ee8	Sat Apr 8 13:02:27 2023	in progress	Added functionality to logout and edit details button for customer	Iruobe Akpatason <N1051811@ntu.ac.uk>
2e4ea6413bf66cb70c10c39b2d0de1d4c3703392	Sat Apr 8 13:46:30 2023	in progress	Added functionality to logout and edit details button for customer	Iruobe Akpatason <N1051811@ntu.ac.uk>
8bd114eca4c186df723ec1645423f647cc6ff441	Sat Apr 8 14:13:25 2023	in progress	Added companyOrders ui	Iruobe Akpatason <N1051811@ntu.ac.uk>
b62fa7b72a83d311b3063f4c3eabb1ca8301a86b	Sat Apr 8 14:43:10 2023	in progress	Added companyOrders ui	Iruobe Akpatason <N1051811@ntu.ac.uk>
51a8f0b4570cc63defac153d8825984343b274c6	Sat Apr 8 16:13:43 2023	in progress	Add files via upload Added Code for Program for UI implementation	T0321003 <rick.harith2021@my.ntu.ac.uk>

fdf063a1f70f77 9cb7c928895ce f3f9a3d6d192e	Sun Apr 9 15:52:32 2023	in progress	Updated code and added sign up functionality to Cargo Owner	T0321003 <T0321003@m.y.ntu.ac.uk>
c7ecf2d7a93ee e82f4774571f7 7599c631f5fbdb	Sun Apr 9 16:31:39 2023	in progress	Fixed some issues with the code.	Rick <T0321003@m.y.ntu.ac.uk>
6b88a663950b 7f221b2842b20 b7684941bad1 639	Sun Apr 9 17:04:00 2023	in progress	Fixed some code issues again.	T0321003 <T0321003@m.y.ntu.ac.uk>
149797c69f7a0 bc176304c228d 46031094a419 8e	Mon Apr 10 10:07:29 2023	in progress	added new fields for driverRegistration	Iruobe Akpatason <N1051811@ntu.ac.uk>
31a4aad5277b 0af4b2e31a868 d304b89648aac ec	Mon Apr 10 15:28:36 2023	in progress	Configured driver registration page.	Rick <T0321003@m.y.ntu.ac.uk>
579c1349bd2b3 a7007b38a87a 5e296b6c15a6e 0a	Mon Apr 10 17:16:14 2023	in progress	Implementd company sign up	Rick <T0321003@m.y.ntu.ac.uk>

bd0e30247cc9c bfaece285c8fd1 2de2d3e8da9e 9	Mon Apr 10 18:43:58 2023	in progress	Fixed code and added log in for CARGO OWNER	Rick <T0321003@my.ntu.ac.uk>
759cbee5bb17a 60b13d5e5ac68 6f45f323c09eb4	Mon Apr 10 19:05:00 2023	in progress	added company main page and company calculation pages for the company	Akpatason Iruobe <N1051811@ntu.ac.uk>
c9b4eabdbe278 0f578d00ebfd8 60f5f808ba5a5 2	Mon Apr 10 19:18:55 2023	in progress	Fixed some code with driver sign up	Rick <T0321003@my.ntu.ac.uk>
04be0dd9ffa085 38c070866bd76 ba64a01435bd 4	Tue Apr 11 08:42:29 2023	in progress	Added a proper log in class that tracks log in status, work on the 3 main pages and 3 main logins please	Rick <T0321003@my.ntu.ac.uk>
c36dfef6d335c2 51b2e16a3aad 2a104a740deec 4	Tue Apr 11 11:09:24 2023	in progress	created driver and company login pages and linked them, also added links on	Iruobe Akpatason <N1051811@ntu.ac.uk>

			the home page for the other login pages	
5431916ddde2c e5a373e2b2ae 0263c0682abf6 48	Tue Apr 11 13:07:35 2023	in progress	Added login functionality for all login pages	Rick <T0321003@m y.ntu.ac.uk>
74017eee5773c 19cd925b0676d 6d893802d065 40	Tue Apr 11 14:43:53 2023	in progress	Created back buttons for customer, driver and company logins.	Iruobe Akpatason <N1051811@nt u.ac.uk>
a7f08e0e75d8b be2cd9d058997 7589faf24bd12 6	Tue Apr 11 14:48:58 2023	in progress	Deleted role select files, .h, .ui , .cpp	Iruobe Akpatason <N1051811@nt u.ac.uk>
1cca7975411e6 6799860f6ed9f 62afb89a06541 7	Tue Apr 11 16:21:06 2023	in progress	fixed some buttons.	Iruobe Akpatason <N1051811@nt u.ac.uk>
b74171896df43 09ec701d07019 318bc3b6e8bfc 2	Tue Apr 11 17:04:10 2023	in progress	added customer order page.	Iruobe Akpatason <N1051811@nt u.ac.uk>

01138a901c798 8f89aac69edba 45d3b2a51716 99	Wed Apr 12 02:36:59 2023	in progress	Done with the customer and company ui, created history page and linked button across pages	Iruobe Akpatason <N1051811@ntu.ac.uk>
5f1f25fa1629af 56aac59a0ee0b 55f7c6a5fff91	Wed Apr 12 03:39:46 2023	in progress	Created driver main page	Iruobe Akpatason <N1051811@ntu.ac.uk>
da75f4acd8337 820c8a2031bdb 35aed6cf97aac a	Wed Apr 12 09:30:53 2023	in progress	Need to fix UI	Rick <T0321003@my.ntu.ac.uk>
317f7b022ad94 7e2e2cada0fa6 e278e3535be6 84	Wed Apr 12 10:00:43 2023	in progress	created customer shipping rates	Iruobe Akpatason <N1051811@ntu.ac.uk>
ba1dbb8333d7 0efbf15bcfbadfe 3813b73b73e7 a	Wed Apr 12 10:38:39 2023	in progress	linked edit page to the status page for customer and removed edit button for other pages	Iruobe Akpatason <N1051811@ntu.ac.uk>
063173d93166c 2182ff2135b51a 709b8ef70f492	Wed Apr 12 10:57:13 2023	in progress	removed customer history files and linked buttons	Iruobe Akpatason <N1051811@ntu.ac.uk>

cf4e9af09be71 06303fd1fd918 776a3fd79e29c	Wed Apr 12 11:04:49 2023	in progress	removed customer history files and linked buttons	Iruobe Akpatason <N1051811@ntu.ac.uk>
9ba1295204e0 602f4c3ef9fc77 d42d7403ee0cf b	Wed Apr 12 11:32:03 2023	in progress	Finished organising the company pages	Iruobe Akpatason <N1051811@ntu.ac.uk>
7196023326bf0 6db42362ade8f e8f73caebacf5c	Wed Apr 12 11:45:32 2023	in progress	Changed pickup and destination locations	Iruobe Akpatason <N1051811@ntu.ac.uk>
cc6683b8b905e 4dacb913bd6bf 70014a69bfe04 f	Wed Apr 12 12:08:07 2023	in progress	Remapped the button to take the customer, driver and company to login after registration	Iruobe Akpatason <N1051811@ntu.ac.uk>
66588e4f577d5 9244e9878be1 bf42f3e5833de 99	Wed Apr 12 13:13:33 2023 +0000	in progress	Image uploading implemented for driver	Author: Rick <T0321003@my.ntu.ac.uk>
25052fc51c493 759f098a9079e 3b07947f9f1b9 2	Wed Apr 12 14:34:06 2023 +0000	in progress	created and mapped buttons for driver main page ,edit and view	Akpatason <N1051811@ntu.ac.uk>

39c81ae94f558 81beeb0dd2d3 766a847665df2 af	Wed Apr 12 15:25:10 2023 +0000	in progress	created and mapped buttons for driver main page ,edit and view	Iruobe Akpatason <N1051811@ntu.ac.uk>
7090068fe74a6 8b85917a897e 34039cae61f0a 8c	Wed Apr 12 18:03:01 2023 +0000	in progress	Made profile editing for Cargo Owner and Driver	Rick <T0321003@my.ntu.ac.uk>
commit a40df78ea26f8 64bb7ae14b36 a9fe7a526edf5fd	Wed Apr 12 18:14:06 2023 +0000	in progress	Changed and fixed some code	Rick <T0321003@my.ntu.ac.uk>
fc007218b2002 d889514431d0 57c5619fb0821f 6	Thu Apr 13 09:57:38 2023 +0000	in progress	Reset database	Rick <T0321003@my.ntu.ac.uk>
f3912158f9c164 df135e4b77acd 6a4955159df22	Thu Apr 13 16:02:01 2023 +0000	in progress	Added order making code and new order table	Rick <T0321003@my.ntu.ac.uk>
945016d81eab 485bc7d2ec81f 1ca3d87dfa296 88	Thu Apr 13 21:15:53 2023 +0000	in progress	Created displays for the Cargo Owner and Company	Rick <T0321003@my.ntu.ac.uk>
429d204a83b8 a2965238fb0fc5 5127afe30b24 7	Fri Apr 14 09:38:17 2023 +0000	in progress	Created and added code functionality for order assigning to a driver	Rick <T0321003@my.ntu.ac.uk>

ef8da425e0d7d acef56a2ebe9f2 d93b59fbcc2af	Fri Apr 14 16:33:17 2023 +0000	in progress	Implemented feedback, driver completing orders, order status	Rick <T0321003@my.ntu.ac.uk>
c2ecdf90f022f1 7b8fb4e1e6c88 09849a80dbbc5	Fri Apr 14 18:33:39 2023 +0000	in progress	Implemented invoice	Rick <T0321003@my.ntu.ac.uk>
decca1fbfbe3e3 d48f85bef6646 09db61435d49 2	Fri Apr 14 20:53:29 2023 +0000	in progress	Implemented feedback, calculate commission and company order history	Rick <T0321003@my.ntu.ac.uk>
c6b08d03ebdc9 5f7aa8d975511 3747de3e5841 d9	Sat Apr 15 13:54:00 2023 +0000	in progress	Implemented driver feedback and driver order history	Rick <T0321003@my.ntu.ac.uk>
1c7341400109e d8b8c8ee70cd3 fe28af6d40917 d	Sat Apr 15 18:02:45 2023 +0000	in progress	Added comments, imported files into eclipse and generated Doxygen documentation	Rick <T0321003@my.ntu.ac.uk>
a8b8b34030e9 59897d63f5162	Sat Apr 15 18:07:18 2023 +0000	In progress	Added Doxygen documentation files	Rick <T0321003@my.ntu.ac.uk>

e3fd33df24eaa 77				
114499311ad6d 3c3dbb040929c 60aced97de452 2 (HEAD -> master, origin/master, origin/HEAD)	Sun Apr 16 00:05:47 2023 +0000	Completed	Cleared the database	Rick <T0321003@m y.ntu.ac.uk>

Table Of Contents

ABSTRACT	2
PLAGIARISM DECLARATIONS.....	2
REVISION HISTORY	3
LIST OF TABLES.....	15
INTRODUCTION	16
BACKGROUND RESEARCH.....	17
APPLICATION REQUIREMENTS LIST	17
MITIGATION PLAN.....	20
GANTT CHART.....	21
ADOPTED CODING STANDARDS	22
SOFTWARE ARCHITECTURE.....	23
<i>UML Diagrams.....</i>	23
<i>Use Case Diagram.....</i>	24
<i>Activity Diagram.....</i>	25
<i>Class Diagram.....</i>	26
<i>Sequence Diagram</i>	27
<i>Communication Diagram:</i>	29
<i>Deployment diagram.....</i>	33
MVC ARCHITECTURE	33
ARCHITECTURAL DESIGN	35
ADOPTED CODING STANDARDS	36
IMPLEMENTATION.....	37
TESTING	43
USER MANUAL	91
CONCLUSION AND FUTURE WORK	109
APPENDIX.....	110
GIT LINK	110
REFERENCES	110

List Of Tables

1. Revision history table
2. Requirements table
3. Migration plan table
4. Gantt Chart
5. STL library table
6. Successful cargo owner login test table
7. Successful driver registration test table
8. Empty field test table
9. Invalid length test table
10. Shipping rate calculation test table
11. Receive order from customer test table.
12. Calculate company commission test table.
13. Assign driver test table.
14. Shipment history test table
15. Notify cargo owner test table.
16. View invoice test table
17. Making comment test table

Introduction

The emergence of the e-transport marketplace (e-TM), a digital platform that makes it easier for cargo owners, drivers, and organisations to communicate and work together, has had a profound impact on the transportation sector. In order to produce an effective, dependable, and user-friendly software solution, an organised and methodical development process is necessary. Presenting a detailed examination of the planning, analysis, design, and implementation phases of our e-TM's software development life cycle.

The report starts by talking about the planning stage, which entails determining the software's function, setting project objectives, and detailing the application's scope. The analysis phase, which is the next step, focuses on gathering and analysing needs from stakeholders, such as cargo owners, drivers, and transportation companies. Using these requirements, a comprehensive functional and technical specification is then created.

During the design process, the group concentrates on developing a high-level understanding of the e-transport market while working through the design phase of the software development life cycle. Unified Modelling Language (UML) diagrams are used in this phase to describe the fundamental operations of the software. The UML diagrams aid in visualising the system's relationships, behaviour, and structure.

The report will go into detail about the implementation stage and describe the methodology, development process, and libraries, frameworks, and how the C++ programming language was utilised. Additionally, to guarantee the quality and dependability of the application, the report will outline the various testing techniques and tools.

Finally, it will detail the difficulties encountered during the creation of the e-TM, including technological and logistical problems, and offer solutions. The overall goal of this paper is to give a thorough overview of the e-TM software development life cycle, including all of its features and associated difficulties.

Background Research

Docker: Docker is a platform that allows the development of applications in portable containers. For this project we took advantage of the feature of docker that allows us to run our program on any OS environment. With the use of docker, the dependencies of our application conflicting with other applications were avoided.

QT creator IDE: Qt creator was an excellent choice for this application for several reasons. Its cross-platform compatibility allowed us to create our application and ensures it will be possible to run on different devices and operation systems. Qt creator also hosts an excellent debugging tool helping us fix errors in our code. This was particularly useful.

Eclipse IDE: The Eclipse IDE is used to utilise the Doxygen documentation, importing the QT project into the Eclipse IDE and creating a make file to compile and run the code. This generated documentation will be saved within then Git repository.

Sqlite3: Our applications database is driven by sqlite3. With this library, user information such as passwords, emails, identification numbers and inputs are stored in sqlite3 database. Information concerning the user such as details they intend to edit, and receipts are rendered to the user for the database.

Application Requirements List

No	MoSCoW analysis	Requirements	Description
R1	Must	Cargo owner must sign up and create new account	Sign up creates personalised experience for user and details must be recorded
R2	Must	View profile and edit customer details	Access to manipulate previously provided personal data
R3	Must	Encrypt customer details	Provide security for customer information
R4	Must	Be able to Sign in and log out	Secure access to the system
R5	Must	place cargo order from a source to a destination	Creates detailed description of order

R6	Must	View order status	Track order progress
R7	Must	receive notification when order was accepted by transportation company	Receive order acceptance notification
R8	Could	View invoices issued by transportation company.	Access to billing information
R9	Could	Track lorry on the road	Real-time tracking of order
R10	Must	View delivery notification and report	Confirmation of delivery
R11	Could	View order history/ feedbacks/ rate shipment	Historical information and feedback
R12	Could	Add comments and recommendation	Provew review
R13	Must	Calculate shipping rates	To give the user an estimate based on (source / destination, lorry type)
R14	Must	View/modify driver's details	Access to manipulate previously provided personal data
R15	Must	Be able to Sign in and log out	Secure access to the system
R16	Must	receive notification when order was accepted by transportation company	Receive order acceptance notification
R17	Must	Notify cargo owner	Receive notification
R18	Must	MUST include certain field	Provide more details
R19	Should	View shipments history	Provides history
R20	Should	Send message to cargo owner	Easy communication with the cargo owner
R21	Must	receive notification from customers	Receive customer notification
R22	Must	Forward order to a free driver nearby the source	Ensure no order is left unattended to
R23	Must	issuing invoice to the customer	Provides invoice on order
R24	Must	Calculate company's commission	Provides a total for the company
R25	Could	Track driver location	Provides a system to follow up on order
R26	Must	View order feedbacks	Provides feedback for order and driver

R27	Must	View order history	Provides history
-----	------	--------------------	------------------

Mitigation Plan

1 = Very Low , 2 = Low, 3 = Moderate, 4 = High , 5 = Very High

Risk Definition	Trigger	Impact	Action Plan
Wrong user input	Triggered by the user inputting wrong values for a field(eg. Phone number in address field)	5	To rectify this problem, a <QMessageBox> header that is part of the qt framework will be used. With this, a window will be generated informing the application user of the error the made.
Access control	Using this application will require the user to create an account or login with an already existing account to gain access	3	Implement strong access control measures, including user authentication and authorization, to prevent unauthorized access to the application and its data.
Data loss	After users input their details and comments, these values will need to be stored and reused.	5	Use sqlite 3 database to ensure that data is documented and recoverable .
Maintenance	Unforeseen error may arise	4	Regularly maintain and update the platform to ensure smooth and efficient service delivery
Interoperability	Services within the application may need to communicate	5	Test the application for compatibility with other services and ensure they are connected using foreign keys within the database

	with one another		
--	------------------	--	--

Gantt Chart

PROJECT TITLE: e-Transport Marketplace(e-TM)

Company Name
Project Lead

Project Start: Fri, 4/7/2023

Display Week:

Mar 27, 2023	Apr 3, 2023	Apr 10, 2023
--------------	-------------	--------------

27	28	29	30	31	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
----	----	----	----	----	---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----

TASK	ASSIGNED TO	PROGRESS	START	END	M	T	W	T	F	S	S	M	T	W	T	F	S	S
Project Management Document																		
Project plan	Rick Harith (T0321003)	100%	4/7/23	4/7/23														
Gantt Chart	Iruobe Akpatason (N1051811)	100%	4/7/23	4/7/23														
Project plan	Al Khattab Saif (T0293214)	100%	4/7/23	4/7/23														
Project plan	Aysheh Bkeirat (T0315059)	0%	4/7/23															
Compiling Results	Everyone	100%	4/7/23	4/7/23														
Software Architect(Diagrams)																		
Component and FSM	Rick Harith (T0321003)	100%	4/7/23	4/11/23														
Use case and activity	Iruobe Akpatason (N1051811)	100%	4/7/23	4/11/23														
class and sequence	Al Khattab Saif (T0293214)	100%	4/7/23	4/11/23														
comms and Deployment	Aysheh Bkeirat (T0315059)	0%																
Compiling Results	Everyone	100%	4/7/23	4/11/23														
Source code/product																		
Source code	Rick Harith (T0321003)	100%	4/11/23	4/15/23														
User Interface	Iruobe Akpatason (N1051811)	100%	4/11/23	4/15/23														
Testing	Al Khattab Saif(T0293214)	100%	4/11/23	4/15/23														
	Aysheh Bkeirat (T0315059)	0%																
Compiling Results	Everyone	100%	4/11/23	4/15/23														

Software Testing						
Source Code	Rick Harith (T0321003)	100%	4/11/23	4/15/23		
User Interface	Iruobe Akpatason (N1051811)	100%	4/11/23	4/15/23		
Testing	Al Khattab Saif(T0293214)	100%	4/11/23	4/15/23		
	Aysheh Bkeirat (T0315059)	0%				
Compiling Results	Everyone	100%	4/15/23	4/16/23		
Extra Activities						
Peer review & Report	Rick Harith (T0321003)	100%	4/16/23	4/16/23		
Peer review & Report	Iruobe Akpatason (N1051811)	100%	4/16/23	4/16/23		
Peer review & Report	Al Khattab Saif(T0293214)	100%	4/16/23	4/16/23		
	Aysheh Bkeirat (T0315059)	0%				
Submission	Everyone	100%	4/16/23	4/16/23		

Adopted Coding Standards

For this application the following coding standards were successfully followed. This enhanced the readability, maintainability, and stability of the program. These practices contributed to a robust and efficient system product and an effective development process.

1. Naming convention: This source code demonstrates adherence to the camel case naming convention, which is characterized by the first letter of each word within an identifier being capitalized, except for the initial word. In this application, we can see the camel case convention consistently applied to various variables, class names and file names.
2. Indentation and spacing: Consistent indentation (four spaces) and spacing between operators and operands is present in this application.
3. Code organization: Header files are included at the beginning of each cpp file. The constructor and destructor of the class are defined, followed by other member functions.
4. Error Handling: This program makes use of error handling and user input validation consistently throughout the code with the use of regex.
5. Code together: With the use of collaborative tools such as code together and Microsoft teams, team members simultaneously contributed to each file of document related to this project.

Software Architecture

UML Diagrams

The following Unified Modelling Language (UML) diagrams are used to visualize and design the e-TM. They provide a graphical representation of different aspects of the software, including its structure, behaviour, and interactions between components. These diagrams are used to communicate the architecture and functionality of the software. Each of these diagrams represents different aspects of the software and were used to model, design, and communicate the software system at different levels of abstraction.

Use Case Diagram

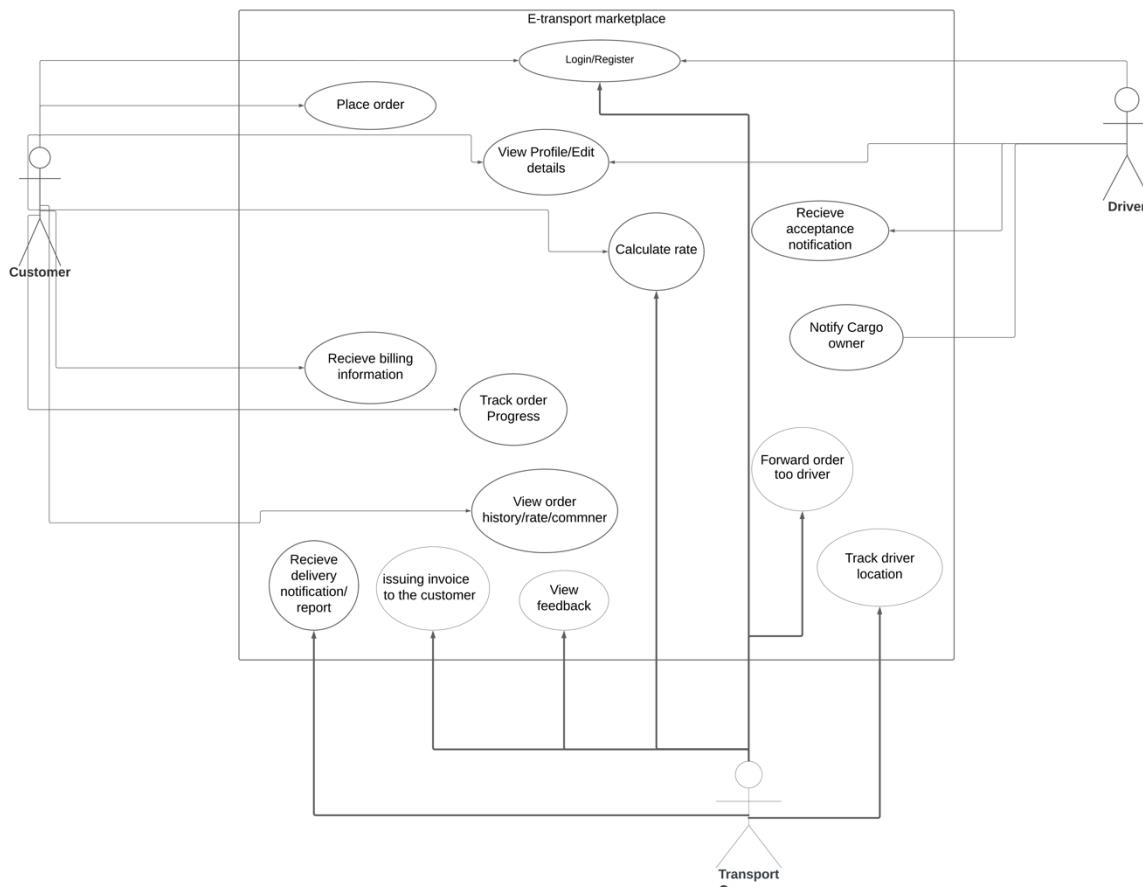


Figure 1. Use Case Diagram

The use case diagram above outlines the interactions between actors and an e-TM system. Each actor interacts with a particular use case, with some being unique to that actor and others being available to multiple actors. The interactions between the actors and the system are represented by the lines with arrows pointing from the actor to the system.

The main actors are customer, driver, and the transportation company. The customers can interact with the system to login, place orders, view and edit their profiles, see the calculated rate for their order, receive billing information, track order progress as well as order history. Drivers can log into the system, receive order acceptance notifications, and notify cargo owner of deliveries. Finally, the transportation company can receive delivery notifications and reports, issue invoices to the customer, view feedback, track driver location, forward deliveries to the driver as well as calculate rates for customers.

Overall, this use case diagram supplies an excellent view of the interactions between multiple actors and the e-TM, and it is also a starting point to the architecture of this application.

Activity Diagram

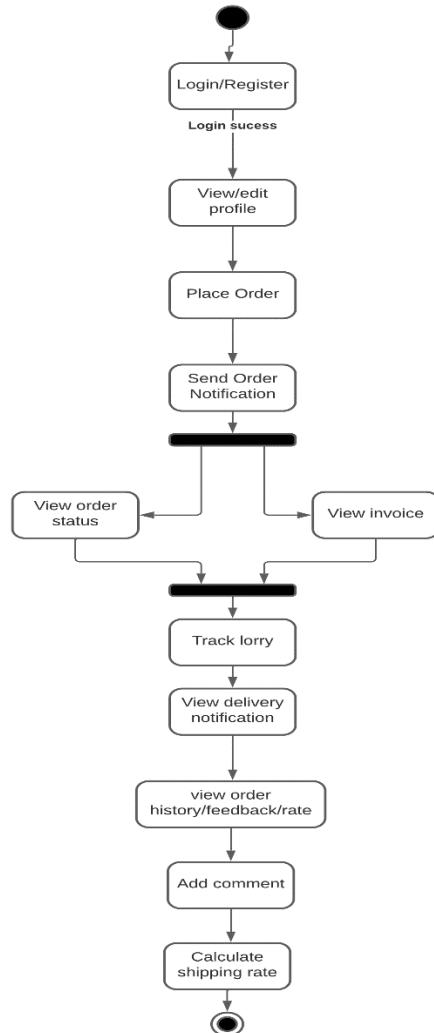


Figure 2. Activity Diagram

This activity diagram models the workflow of processes in the e-TM and the order in which they may occur. Following the start node of this diagram, the first action stem involves the user logging in. After authorisation, the user is given the option to view or edit their information. The flow then continues with the user placing an order and a notification of said placed order being sent.

The next step on the flow involves a fork. An option is given to view the order status or view the invoice of the order and after a decision is made tracking on the delivery lorry begins. As the delivery is concluded the next flow involves being able to view order history, give

feedback on order, rate the order and leave a comment. After shipping the rate is then calculated and the session is then ended.

Class Diagram

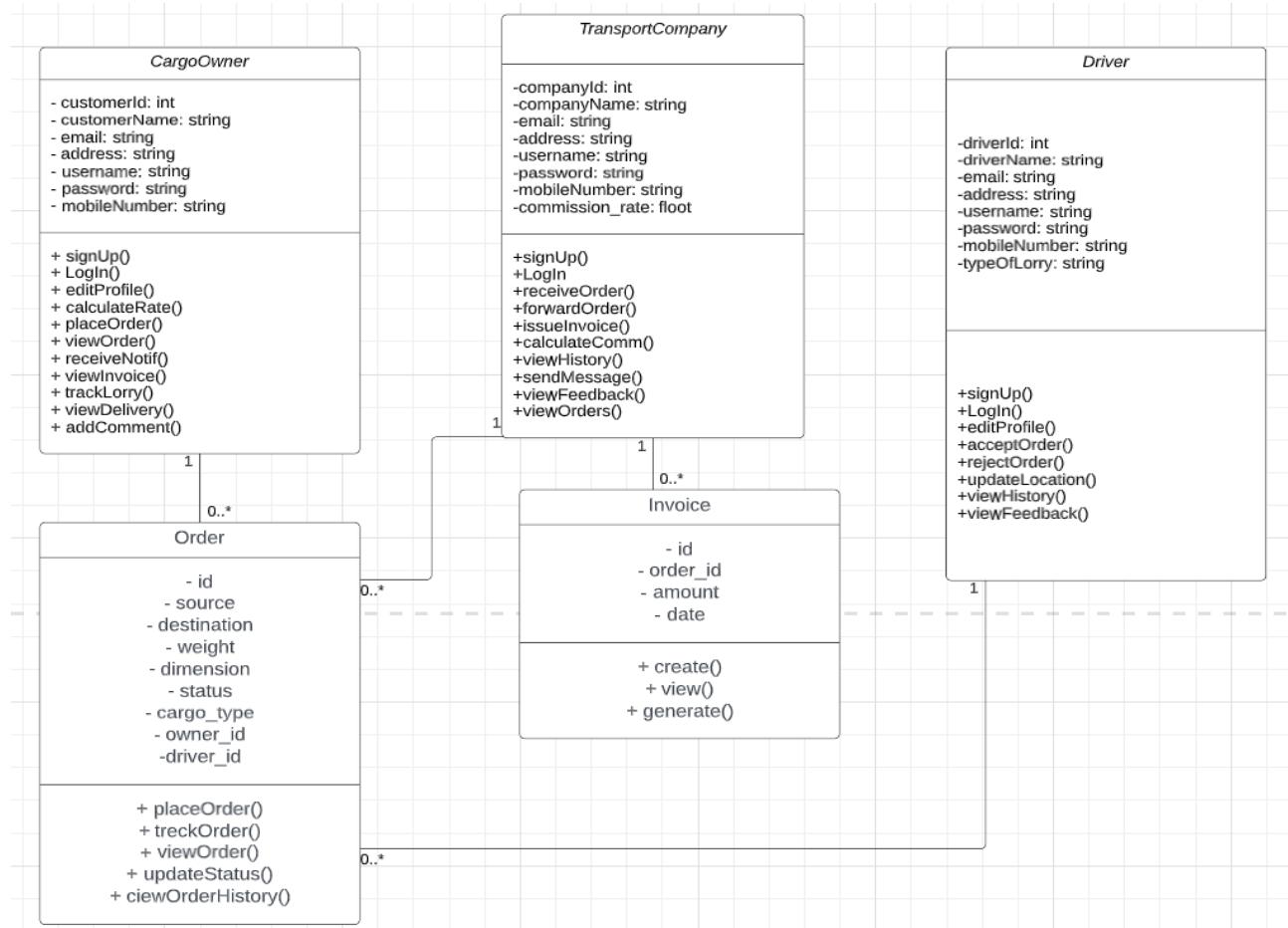


Figure 3. Class Diagram

This class diagram represents the e-Tm platform and its three main classes: CargoOwner, Driver, and TransportationCompany. The CargoOwner class has methods to calculate shipping rates, place cargo orders, view order status, receive notifications, view invoices, track lorries, view delivery notifications and reports, view order history and feedback, and add comments and recommendations. The Driver class has methods to view and modify driver details and is related to the TransportationCompany class, which assigns drivers to new orders and tracks their location and status. The TransportationCompany class has methods to receive orders from customers, forward orders to drivers, issue invoices to customers, calculate commission, track driver location and status, and view order and driver

feedback. The Order class is related to all three classes and has attributes such as order ID, source, destination, weight, dimensions, and status. The relationships between classes include the following:

- CargoOwner has a one-to-many relationship with Order
- Driver has a one-to-many relationship with Order
- TransportationCompany has a one-to-many relationship with Order and Invoice
- Order has a many-to-one relationship with CargoOwner, Driver, and TransportationCompany
- Invoice has a many-to-one relationship with TransportationCompany

Sequence Diagram

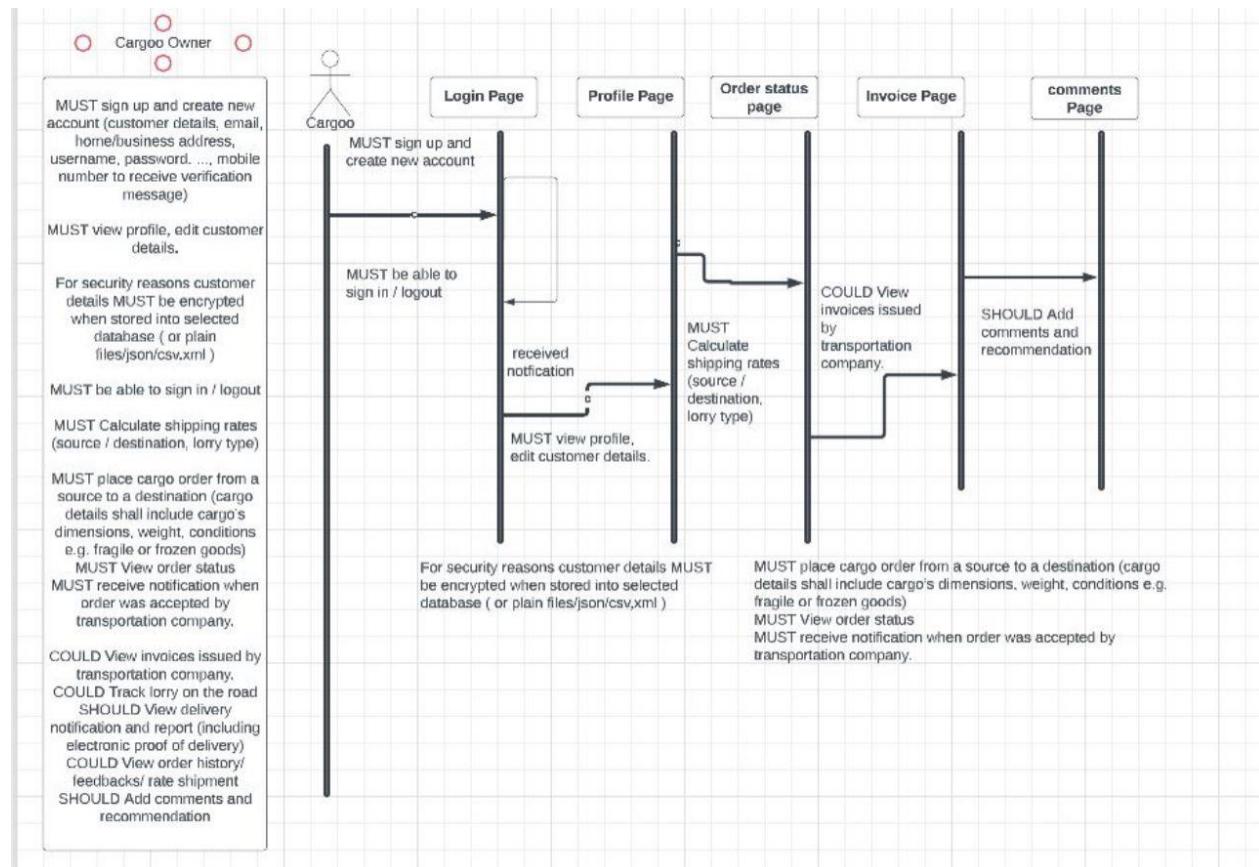


Figure 4. Sequence Diagram

the sequence diagram would begin with the cargo owner accessing the login page of the application. Upon successful login, the user is directed to their profile page where they can update their personal information, such as their contact details and shipping preferences.

From the profile page, the cargo owner can access the order status page, which displays the current status of their shipment. This page shows details such as the current location of the cargo and the estimated time of delivery. If there are any delays or issues with the shipment, the status page will provide alerts and notifications to the cargo owner.

Next, the sequence diagram would show the cargo owner accessing the invoice page, which provides details of the billing and payment process. The invoice page would display the total cost of the shipment and any applicable taxes or fees.

Finally, the sequence diagram would show the cargo owner accessing the comments page, where they can leave feedback or ask questions about their shipment. The comments page provides a platform for the cargo owner to communicate with the logistics company and resolve any issues that may arise during the shipping process.

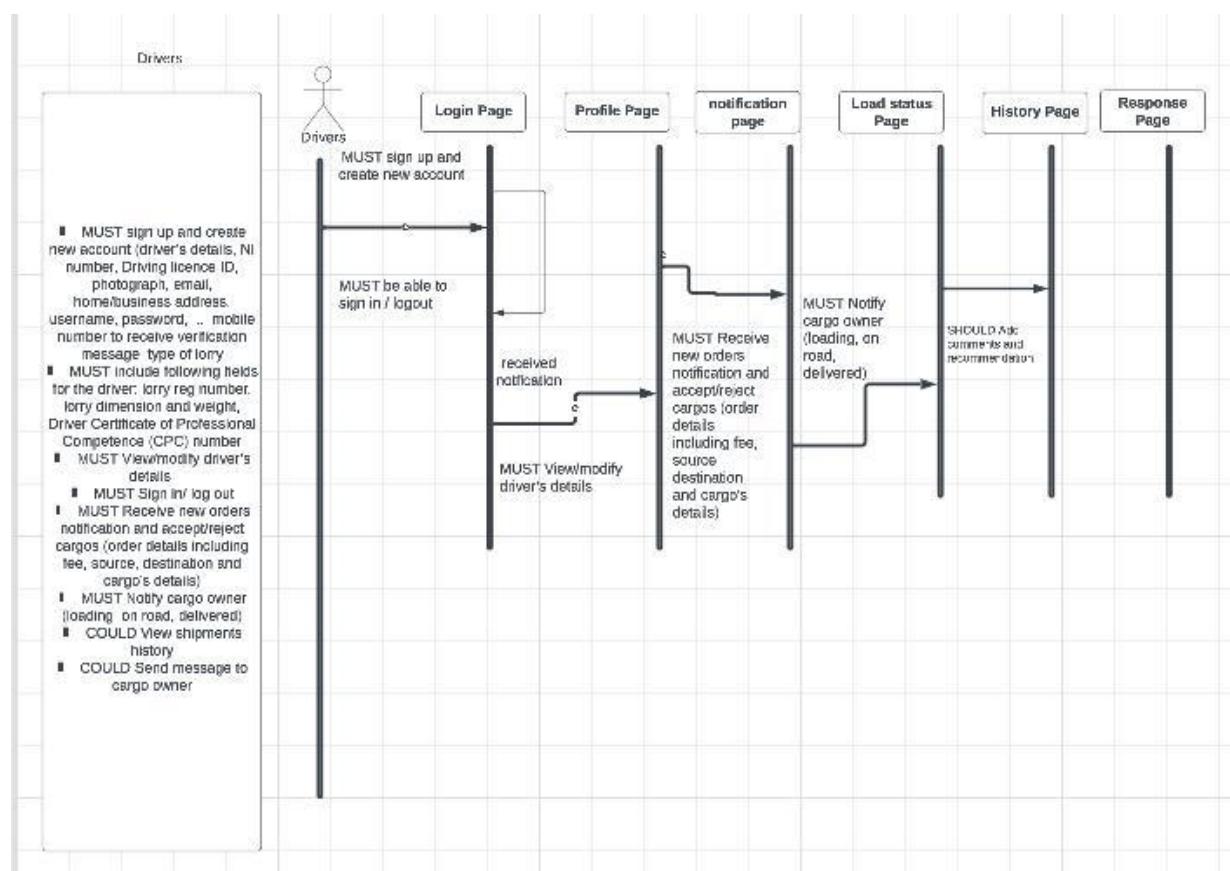


Figure 5, Sequence Diagram

The diagram would begin with the driver initiating the login process by entering their credentials into the login page. The system would then verify the credentials and grant the driver access to the profile page. At this point, the driver would be able to view and update their personal information.

If there are any notifications for the driver, the system would display them on the notification page. The driver could then choose to respond to the notifications by clicking on them, which would take them to the response page.

The long status page would display information about any long trips the driver has taken or is currently taking. The driver would be able to view details such as the start and end locations, the duration of the trip, and any other relevant information.

The history page would show the driver's trip history, including details such as the date, time, and duration of each trip, as well as the start and end locations.

Finally, the response page would allow the driver to respond to any notifications they have received. They could choose to accept or decline requests, provide feedback, or take any other appropriate action.

Communication Diagram:

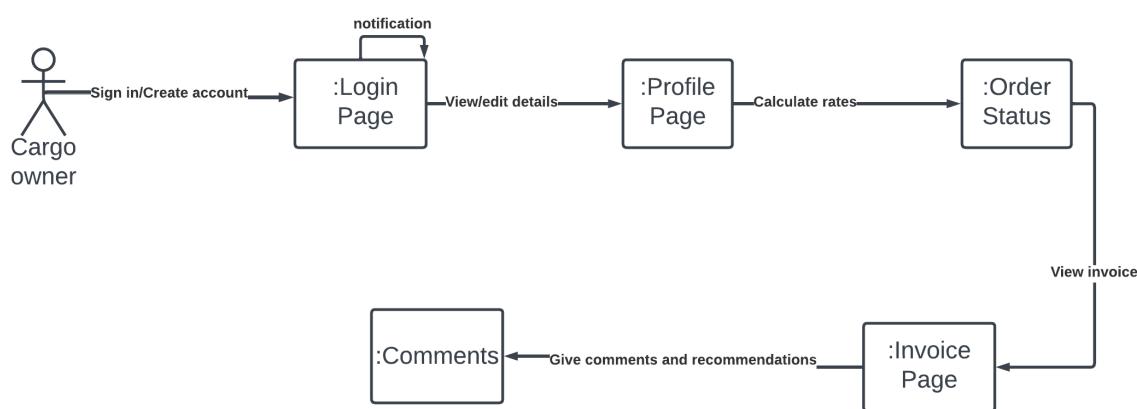


Figure 6 Cargo Owner Communication Diagram

The diagram likely shows the cargo owner as the primary actor, with the other components represented as boxes or nodes connected by lines. The login page would likely be the first component encountered by the cargo owner, allowing them to enter their login credentials

and access their account. From there, the cargo owner would be able to access their profile page, where they can view and edit their personal information, such as their name, address, and contact details.

The order status component would allow the cargo owner to view the status of their current orders, including any updates or changes to delivery times or locations. The invoice page would likely show any outstanding payments or bills related to the cargo owner's account. Finally, the comments component would allow the cargo owner to communicate with other users or support staff, either to ask questions or provide feedback.

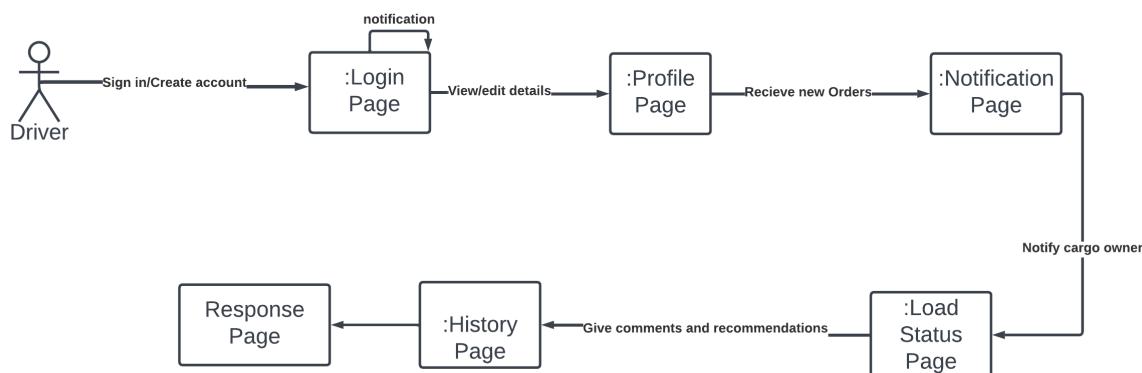


Figure 7 Driver Communication Diagram

The Communication Diagram represents the interactions between a driver and various pages within a system. The driver is the primary actor, and the other components are depicted as boxes or nodes connected by lines. To access their account, the driver first encounters the login page where they enter their login credentials. From there, they can access their profile page to view and edit their personal information, such as their contact details and license information.

The notification page displays any relevant notifications related to the driver's job, such as new pickups, deliveries, or updates to existing orders. The Load status page shows the driver the status of their current load, including any changes to the delivery schedule or location. The history page provides a record of the driver's past deliveries and pickups, including information such as the date, time, and location of each job. Finally, the response page allows the driver to respond to any messages or notifications received from the system, such as confirming a delivery or providing additional information about a job.

Component Diagram:

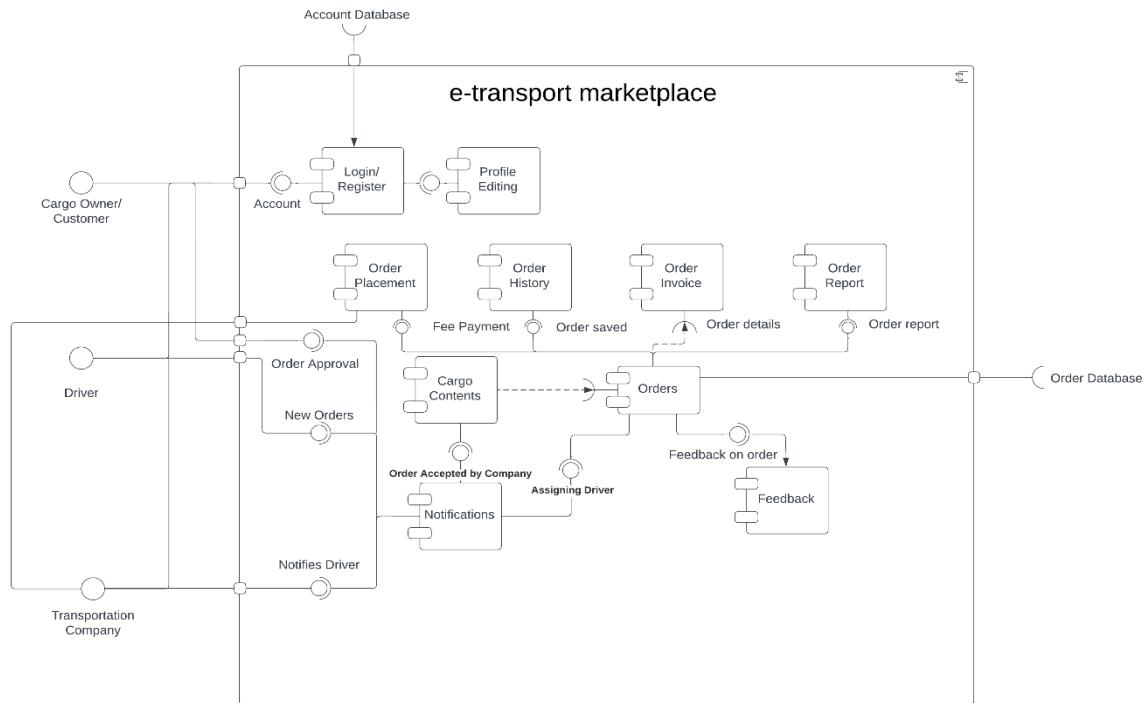


Figure 8. Component Diagram

The component diagram details the multiple components that are present within the system. The three major types of users are Cargo Owner (Customer), Driver and Transportation Company, an account is required to interact with the e-transport marketplace system. The major components are Order, Login/Registration and Notifications. The three types of users will be signed up with their respective types of accounts, this information will be stored within the user database.

Orders and cargo contents have a dependency, order invoice and orders have a dependency as well.

When an order is placed, the notification system handles sending all the relevant info to each of the users. Once the order is confirmed, details are saved for the history, the placing of the order, invoice, and report. All this information is saved in the order database.

FSM Diagram:

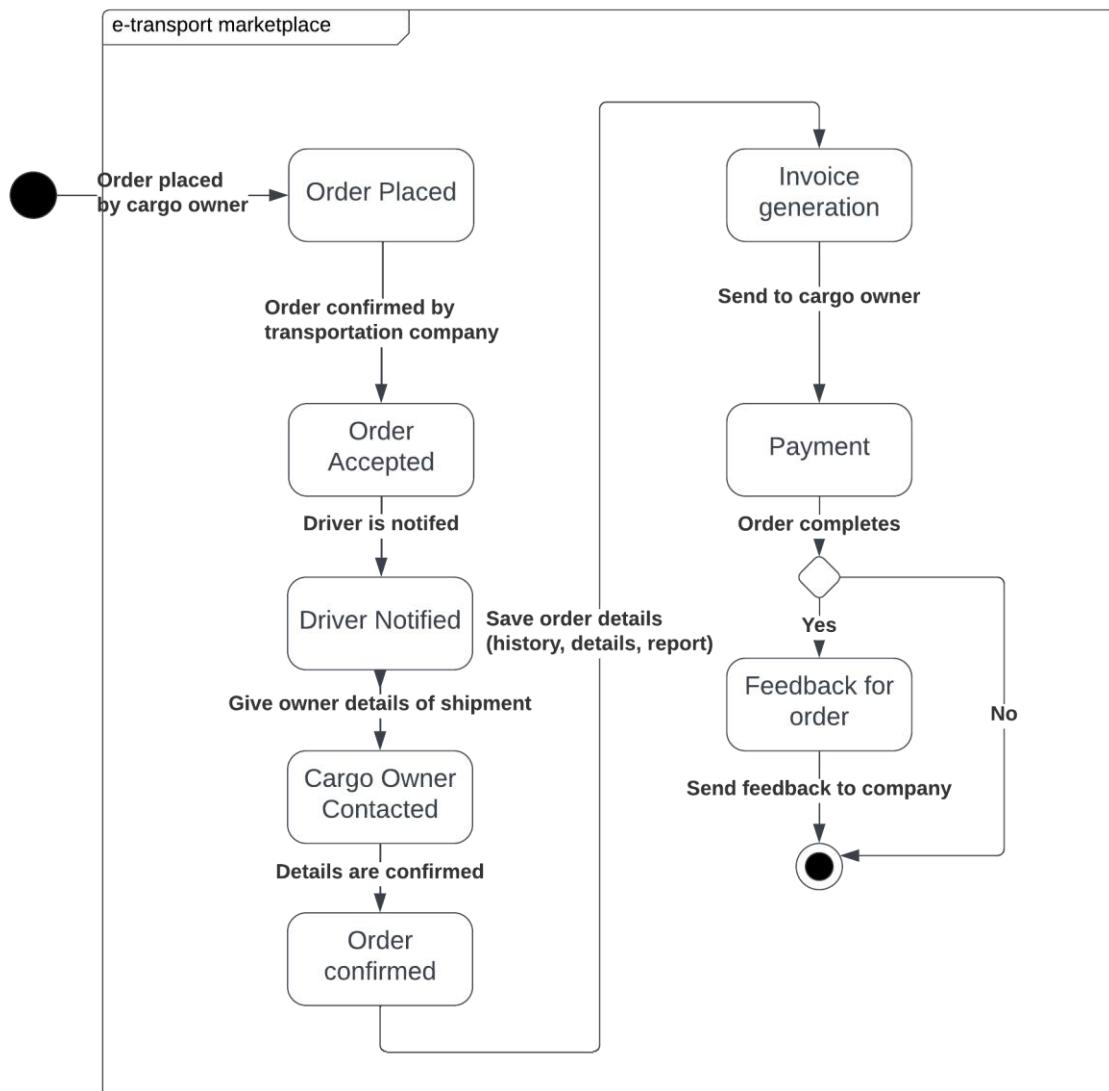


Figure 9. FSM Diagram

This FSM diagram describes the ordering process for a cargo to be shipped. When a cargo owner places an order for a cargo to be shipped, a transportation company can accept the order and notify a driver for the job. Once this is done, the driver will notify the cargo owner of the shipping details. The order confirmation is sent to the transportation company, the order details are saved for having a history, report, and invoice. This invoice is sent to the cargo owner for payment, the order becomes ongoing when payment is accepted. After the order is complete, the cargo owner can provide optional feedback.

Deployment diagram

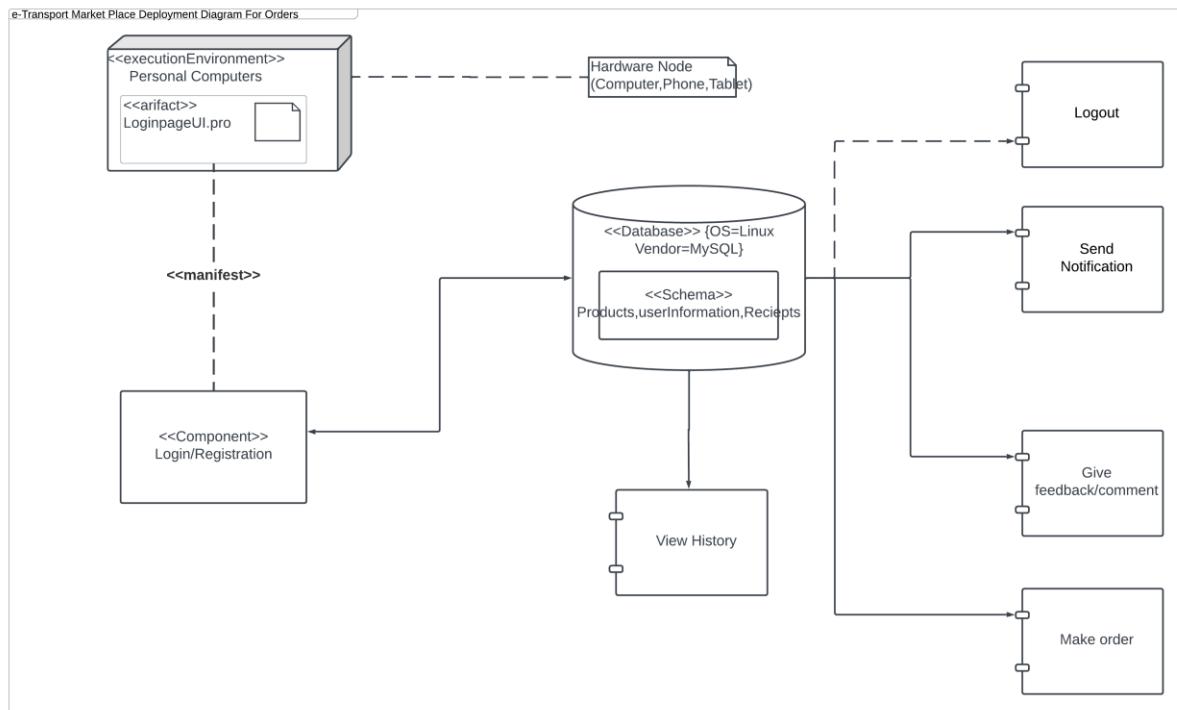


Figure 10. Deployment Diagram

In the deployment diagram above, the execution environment node contains the LoginpageUI.pro artefact. This software is the execution file for our application. This artifact interacts with login and registration component and retrieves data from the database. Within the database the user details, order history, feedback, order, and notification values are stored.

MVC Architecture

Model:

The Model component in our e-TM system will consist of three sub-components:

1. **Cargo Order:** This will represent an order placed by a customer to transport their cargo from a source location to a destination location. It will store details such as cargo dimensions, weight, conditions, source and destination locations, and customer details.
2. **Driver:** This will represent a driver who will accept orders placed by customers and transport the cargo from the source to the destination location. It will store details

such as the driver's name, driving license ID, photograph, lorry registration number, lorry dimensions and weight, and availability status.

3. Transport Company: This will represent a transport company that will receive orders placed by customers and assign them to available drivers. It will store details such as company name, commission rate, order history, and feedback from customers and drivers.

View:

The View component in our e-TM system will consist of three sub-components:

1. Customer Interface: This will be the interface through which customers will place orders, view their order history, track their cargo, and receive notifications about order status and delivery.
2. Driver Interface: This will be the interface through which drivers will receive and accept orders, view their order history, and update their availability status.
3. Transport Company Interface: This will be the interface through which the transport company will receive orders from customers, assign orders to available drivers, view their order history, and generate reports.

Controller:

The Controller component in our e-TM system will consist of three sub-components:

1. Customer Controller: This will handle customer actions such as placing orders, viewing order history, and tracking their cargo.
2. Driver Controller: This will handle driver actions such as accepting orders, updating availability status, and viewing order history.

3. Transport Company Controller: This will handle actions such as receiving orders from customers, assigning orders to available drivers, generating reports, and viewing order history.

Architectural Design

1. Client Tier:

The client tier is the front-end of the e-transport marketplace that will be used by customers, drivers, and transport companies to interact with the system. It will consist of web and mobile applications that will be accessible via the internet.

2. Application Tier:

The application tier will handle the business logic of the e-transport marketplace. It will consist of the following sub-components:

- Authentication and Authorization: This sub-component will be responsible for managing user authentication and authorization to ensure that only authorized users can access the system.
- Order Management: This sub-component will be responsible for managing cargo orders, assigning them to available drivers, and tracking their status until they are delivered.
- Driver Management: This sub-component will be responsible for managing drivers, their availability, and their assignments.

3. Data Tier:

The data tier will store all the data required by the e-transport marketplace. It will consist of the following sub-components:

- Customer Database: This sub-component will store all customer information such as personal details, order history, and feedback.

- Driver Database: This sub-component will store all driver information such as personal details, lorry details, availability, and order history.
- Transport Company Database: This sub-component will store all transport company information such as commission rates, order history, and feedback.
- Order Database: This sub-component will store all order information such as cargo details, source and destination locations, driver assignments, and status updates.

Adopted Coding Standards

For this application the following coding standards were successfully followed. This enhanced the readability, maintainability, and stability of the program. These practices contributed to a robust and efficient system product and an effective development process.

1. Naming convention: This source code demonstrates adherence to the camel case naming convention, which is characterized by the first letter of each word within an identifier being capitalized, except for the initial word. In this application, we can see the camel case convention consistently applied to various variables, class names and file names.
2. Indentation and spacing: Consistent indentation (four spaces) and spacing between operators and operands is present in this application.
3. Code organization: Header files are included at the beginning of each cpp file. The constructor and destructor of the class are defined, followed by other member functions.
4. Error Handling: This program makes use of error handling and user input validation consistently throughout the code with the use of regex.
5. Code together: With the use of collaborative tools such as code together and Microsoft teams, team members simultaneously contributed to each file of document related to this project.

Implementation

SQLite 3 is required, paste the following command in the command line:

```
sudo apt-get install sqlite3 libsqlite3-dev
```

STL Libraries Used:

Library	Description
sqlite3.h	A library for embedding SQLite, used to interact with the database used to store, manipulate and query user information, order information and more
iostream	a library that gives C++ users access to the common input and output operations. It defines functions like getline() and endl for reading and writing data, as well as objects like cin, cout, and cerr that allow input and output to the console. The library was used to debug certain errors in the console with cout.
string	a library that offers a C++ class for dealing with character strings. For concatenating, comparing, and manipulating strings as well as converting between strings and other data types, it defines functions and operators.
regex	a C++ library with regular expression matching capabilities. Using regular expressions, a standardised vocabulary for expressing intricate text patterns, it enables the program to look and compare patterns in strings. This was used for exception handling.
sstream	a library that gives C++ programmers a mechanism to handle strings as input and output streams. It includes classes like istringstream and ostringstream to read

	data from strings just like input streams or write data to strings just like output streams.
iomanip	a library that offers C++ manipulators for input and output formatting. When reading or writing data, it enables the program to specify different formatting options, such as the number of decimal places to display or the width of a field. For setting these parameters, it additionally has functions like setprecision() and setw(). This library was used to handle dimensions, weight and currency values.

Functionalities:

Numerous capabilities are included in the program's implementation for various user roles, including Cargo Owner, Drivers, and Transportation Company.

One of the features of Cargo Owner is the option to register and open a new account with information like client information, email, address, and mobile number. They can view and modify the information on their profile. Additionally, Cargo Owners have the option to calculate shipping rates based on the source and destination. Additionally, they can examine their order status throughout the delivery process. Cargo Owners also have the choice to inspect their completed order history, view invoices and send feedback.

Drivers can register and create a new account with their information, including their NI number, driving licence ID, picture, and kind of vehicle. Additionally, they can access and edit their personal information, which includes their driver certificate of professional competence (CPC) number, vehicle's dimensions, weight, and registration number. With order specifics, they can view the shipping rates and final price, source, destination, and cargo's details, drivers can accept or reject cargo orders too. The status of the shipment, including loading, on the road, and delivery, must be reported to the cargo owner. They can also see the history of their completed deliveries and leave feedback as well.

The Transportation Company can register and set up an account using information about their business. They have the ability to log in, log out, and send orders to a driver. Rejected

orders will also reappear in the main page to be assigned to another driver. They can check the completed order history and feedback for each order from the cargo owner and driver, and they can calculate the company's commission for each order.

The functionalities that have not been achieved include the notification system, the encryption of cargo order, automatic driver rejection that cycles to the next driver if an order is rejected, driver availability and assigning a driver that is close to the source and destination.

Architecture:

The Model, the View, and the Controller are the three unique components that make up the Model-View-Controller (MVC) architecture. For each of these components, separate classes must be created, and their relationships and roles must be specified to implement MVC architecture using classes.

The implementation of the program does not meet the Model-View-Controller (MVC) architecture. The Model, View, and Controller components are not clearly separated from one another. Most of the code handles database transactions, executes UI-related tasks like refreshing the UI and back-end related code all together in a single function.

Furthermore, the implementation doesn't follow any design pattern to divide concerns and tasks, which could make it more difficult to maintain and expand the program in the future.

The code does not explicitly define any access modifiers for its members or methods, which brings us to our final point on access modifiers. Class members and methods are by default private in C++ unless otherwise specified. To maintain adequate encapsulation and to prevent unauthorised access, it is crucial to explicitly define access modifiers for each member and method.

STL Data Structures:

The program is primarily intended to serve as an interface for interacting with a SQLite database. To manage database connections, run SQL queries, and get data from the database, it significantly depends on the features offered by the SQLite3 library.

While using STL data structures like vectors, lists, and maps may be advantageous in some circumstances, it might not be the ideal choice for a program that is so strongly focused on

database interaction. In comparison to creating data structures using STL, the SQLite library already offers a wide range of data structures and algorithms for processing data. Using STL data structures for the program would not be desirable as handling large amounts of data would make the program be prone to errors.

Furthermore, the decision to utilise SQLite was influenced by the efficiency and dependability in managing massive volumes of data, which can be challenging to accomplish with just STL data structures. Therefore, based on the requirements and the program's design, using SQLite to manage the data in the database is far more efficient than utilising STL data structures.

STL Search and Sort Algorithms:

As mentioned earlier, the program heavily relies on the SQLite C++ Library and databases to manage data, the program uses SQL queries to retrieve data from a SQLite3 database and shows that data in QTableWidgets. The user can examine their orders and their current status using the program.

Search and sort algorithms are used to quickly locate or arrange data. To obtain data from the database, however, the program relies on SQL queries to retrieve the data. Using search and sort algorithms within the program would not offer any value because these queries are made to efficiently obtain particular data from the database.

If the program were to use search and sort algorithms, it must first need to obtain all the data from the database and store it in a data structure, such an array or a vector, then would it be able to utilise search and sort algorithms. Under this hypothetical, the data within the data structure could then be found or sorted.

Exception handling:

Using methods and regular expressions, exception handling has been built into this program. Regex patterns for validating most input fields, methods like 'empty()', 'isdigit' and more have been used too. An error message is displayed using a QMessageBox and the appropriate input field is cleared to prohibit further execution until the error has been resolved if any of the inputs are empty or do not match the anticipated regex pattern.

Additionally, the program creates and maintains a database using the sqlite3 library. When a UNIQUE constraint fails during a SQL query, the programme detects the error and uses the sqlite3_errmsg() function to extract the error message. After that, the error message is transformed to a string and examined to determine if the substring "UNIQUE constraint failed" is present. If the UNIQUE constraint is violated, the program addresses the error by informing the user about it with an error message.

In conclusion, a more reliable and error-free programme can be created by using methods and regular expressions for handling exceptions and the SQLite3 package for database management. These capabilities enable the programme to deliver helpful error messages and stop the user from inputting the wrong information to all fields.

Access Modifiers:

To regulate the accessibility of class members, this program has introduced access modifiers. Access modifiers are used to limit class members' visibility and shield them from unauthorised access or alteration.

Private members can only be accessed within the class that declared them, whereas public members can be accessed from anywhere in the programme. With this knowledge, private access modifiers have been used to restrict the visibility of sensitive member variables and functions within their respective classes. Public access modifiers have been used to make certain member functions and variables accessible to other parts of the program. In addition, UserSession, DriverSession and CompanySession class is declared as extern in the code, indicating that it is defined elsewhere in the code but will be used in other files. This is helpful if, for instance, a large program has several source files and needs to share the same instance of a class amongst other files.

The program has should have data hiding and encapsulation by employing access modifiers. This guarantees that only the intended methods are used to modify objects, protecting sensitive data from unauthorised changes. By making it clearer where specific variables or functions are used or modified, this method also makes the code easier to maintain and debug.

User Interface:

The program is set up to use QT Creator's GUI, the user interacts with the program through a Graphical User Interface, there is no console-based interactions to go through for the user.

The program's interface (GUI) is designed to let users interact with the application in a fluid and user-friendly manner. This makes the application more approachable and simpler to use by removing the requirement that the user have any prior training or familiarity with programming or command-line interfaces. Additionally, the GUI makes it possible for users to engage with the programme graphically, offering a more simple and effective manner to explore all of the program's features and functionalities. In general, this strategy improves the user experience and makes it easier to utilise the application.

Use of Multi-Thread:

Concurrent programming techniques are not used in the program. The technique of creating programs that can carry out several tasks at once is known as concurrent programming. Multiple strategies, such as parallelism, multiprocessing, and multithreading, can be used to accomplish this.

A concurrent programme creates many threads of execution to carry out various activities concurrently, improving performance and responsiveness. Writing concurrent programmes, however, can be difficult since it necessitates carefully taking into account thread synchronisation, data sharing, and other potential problems that might occur from parallel execution.

In conclusion, the program implemented does not meet this criterion.

Use of source control tools

Git repositories, which allow for version control was used to maintain the code. Git makes it simple to branch, merge, and track code changes, which helps to prevent disputes and preserve code integrity. Additionally, the Eclipse IDE, which offered an automated method of producing code documentation, was used to generate the Doxygen documentation. This made it simple to comprehend how the code worked. It was possible to create documentation using Doxygen in a variety of forms, including HTML, LaTeX, and RTF. As a result, the documentation was easier to obtain and could be tailored to the team's requirements. Overall, using Git and Doxygen has streamlined the code management process, the source code and documentation are both saved in the Git repository.

Testing

ID	001	Description:	This test plan aims to ensure that the Cargo Owner account creation process is working as expected, without any errors or issues.
Test type	Quantity / Quality	Success criteria:	The customer should be able to create a new account successfully. The account details provided by the Cargo Owner should be saved accurately in the database. The customer should be able to log in using the same account details.
Number of attempts:	6	Comments:	-
List of equipment/requirements	QT Creator		
Setup instructions	1. Open Qt Creator. 2. Open the project containing the customer account creation code. 3. Build the project to ensure that it is free of errors. 4. Start the application. 5. click into create new account button. 5. click into Cargo Owner. 6. Fill in the required details, such as name, username, password, email, address, and phone. 7. Click on the "Register" button.		
Failure correction procedure	If the account creation fails, an error message should be displayed. Check the error message and take corrective action as necessary. Retry the account creation process.		
Engineer(s)/Technician(s)	-		
Individual results	Pass / Fail		
Test Date:	09/04/20 23	Date:	09/04/2023

successfully signed up for cargo owner:

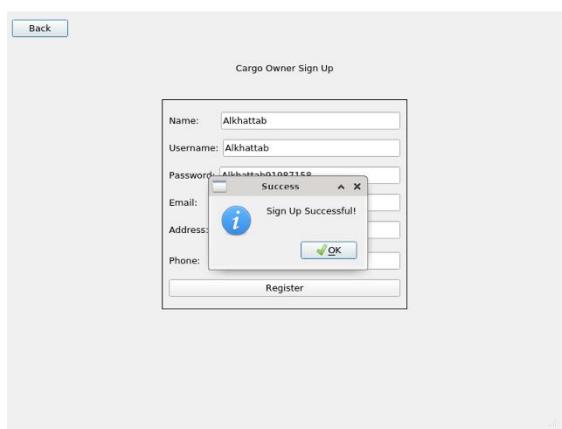


Figure 11

Error message pop up when input invalid email form:

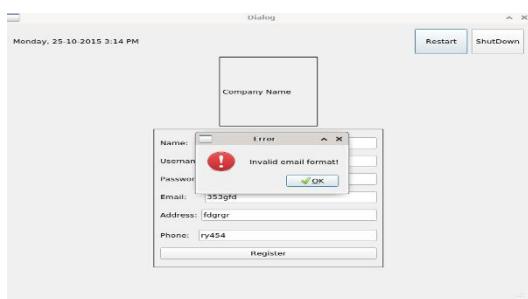


Figure 12

Error message pop up when input a wrong phone number:

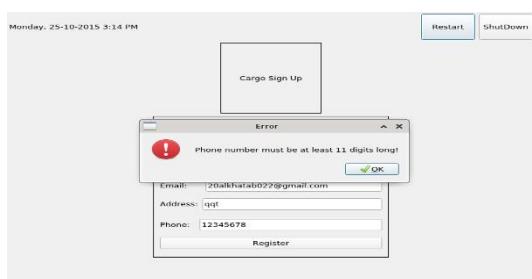


Figure 13

Error message come when input an empty Address:

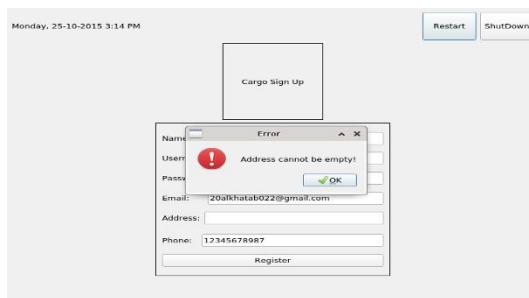


Figure 14

Error Message pop up when input Invalid password format:

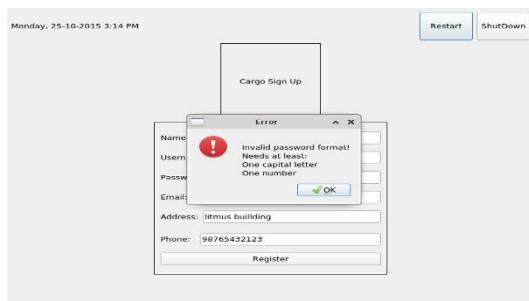


Figure 15

Error Message pop up when input short Username:

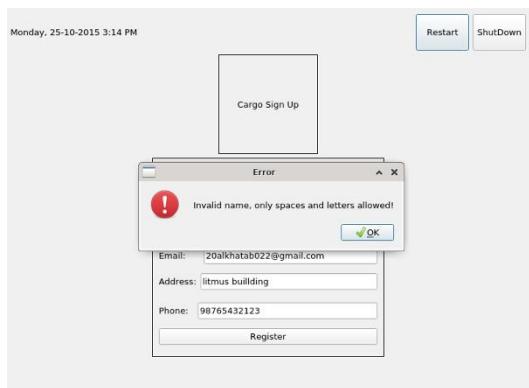


Figure 16

Test Report for Creating a New Customer Account

Test Plan ID: 001

Test Date: 2023-04-09

Tester:

Introduction

The purpose of this test was to verify that a new customer account can be created successfully, and that the account details are saved accurately in the database. The test was performed on a computer with an internet connection and Qt Creator.

Conclusion

The test was successful, and it can be concluded that the customer account creation process is working as expected. No issues or errors were encountered during the test, and the account details were saved accurately in the database. Therefore, the test plan has passed.

Recommendations

No recommendations for improvements were identified during the testing process.

ID	002	Description:	This test plan aims to ensure that a new driver account can be created successfully with the provided driver, lorry, and lorry type details.
Test type	Quantity <i>/Quality</i>	Success criteria:	- A new driver account should be created with the provided details. - The driver, lorry, and lorry type details should be saved accurately in the database.
Number of attempts:	24	Comments:	-

List of equipment/requirements	QT Creator.
Setup instructions	<ol style="list-style-type: none"> 1. Open Qt Creator. 2. Open the project containing the customer account creation code. 3. Build the project to ensure that it is free of errors. 4. Start the application. 5. click into create new account button. 6. Fill in the required driver details, Lorry Details and Lorry Type. 7. Click on the "Register" button.
Failure correction procedure	If the account creation fails, an error message should be displayed. Check the error message and take corrective action as necessary. Retry the account creation process.
Engineer(s)/Technician(s)	-
Individual results	Pass / Fail
Test Date:	10/04/20 23
Date:	10/04/2023

Successful Register to Driver:

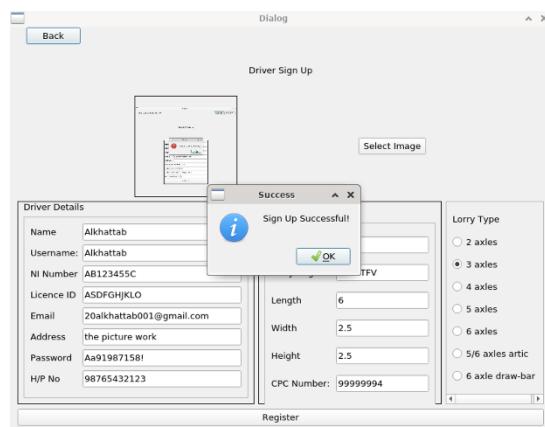


Figure 17

Input Invalid email format:

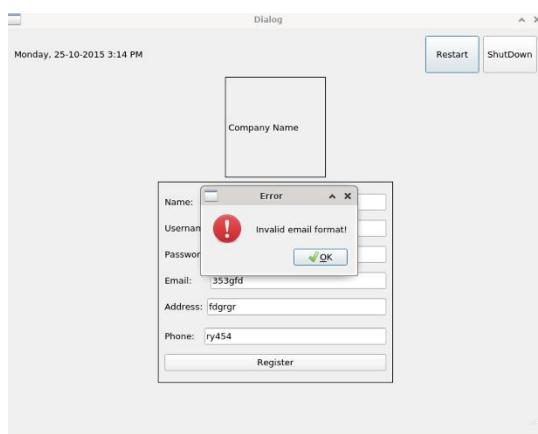


Figure 18

Input Invalid name:

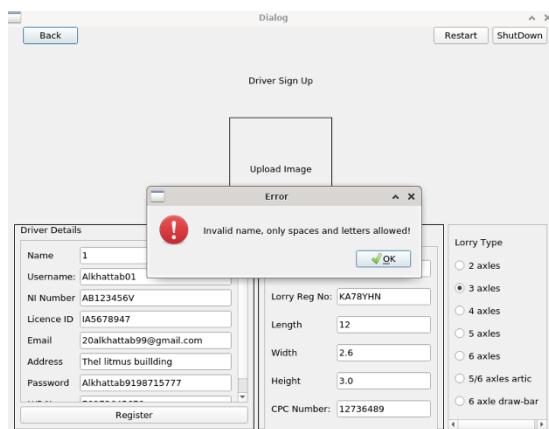


Figure 19

Input invalid Username format:

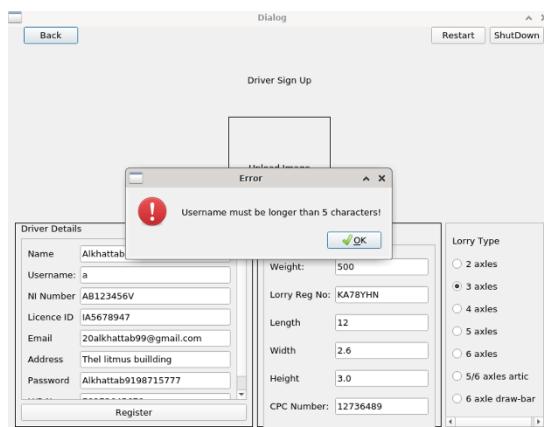


Figure 20

Invalid NI number format:

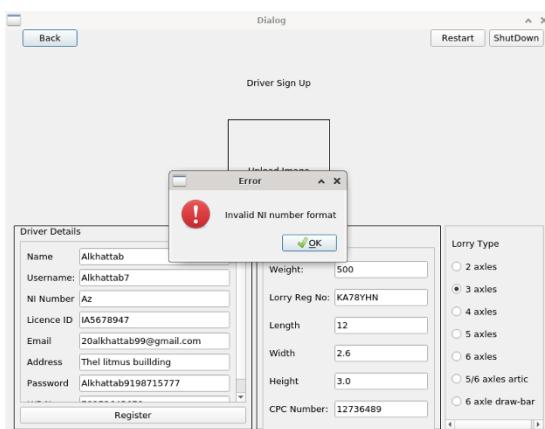


Figure 21

Input empty Driving License ID:

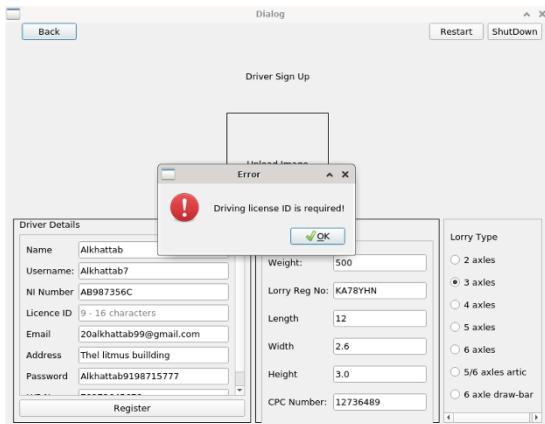


Figure 22

Input empty Address:

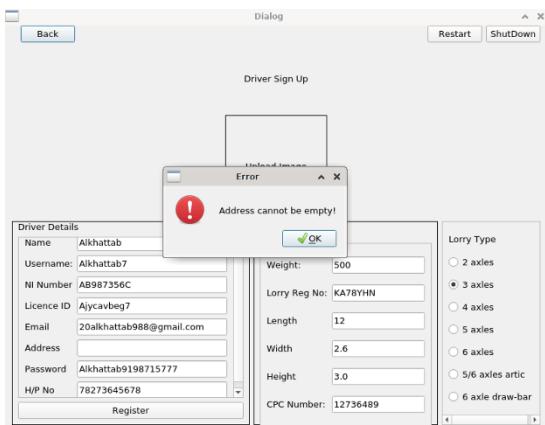


Figure 23

Password should not be empty:

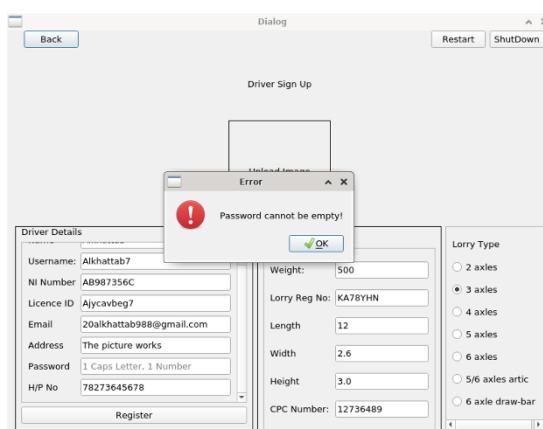


Figure 24

Input Invalid password format:

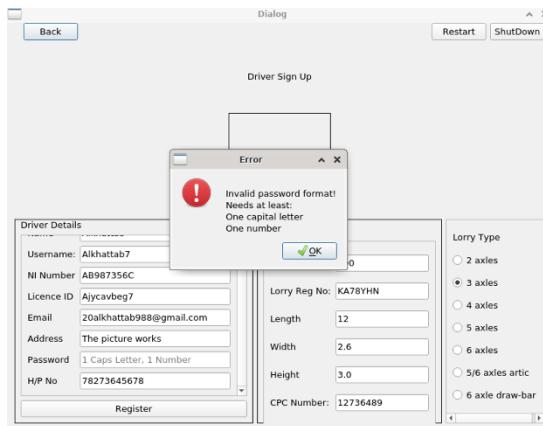


Figure 25

Do not input a number:

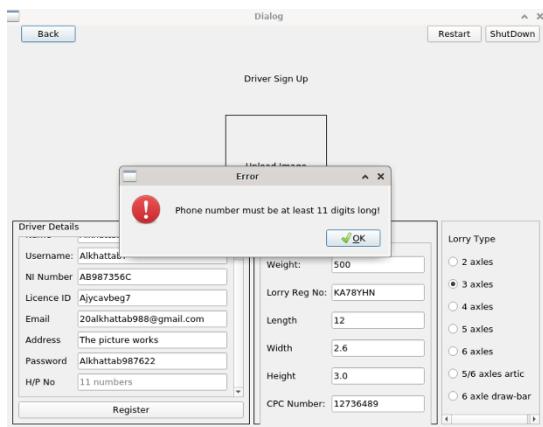


Figure 26

Lorry weight cannot be empty:

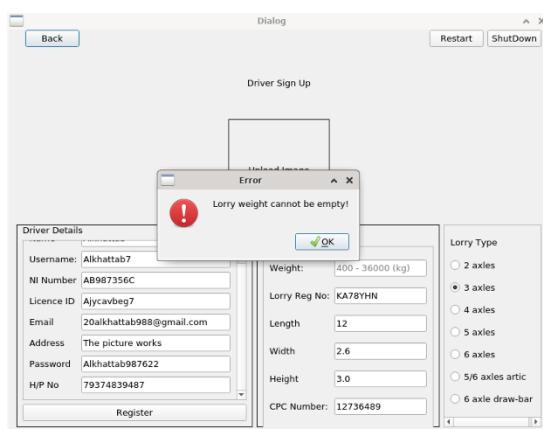


Figure 27

Input Invalid NI number format:

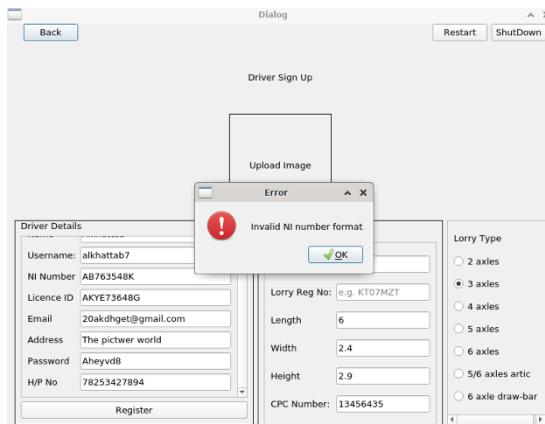


Figure 28

Input empty Lorry length:

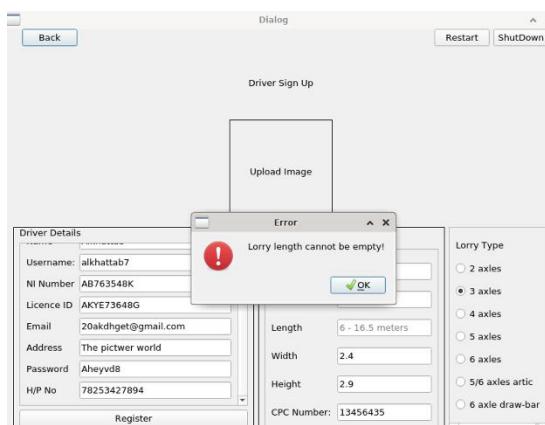


Figure 29

Empty Lorry width:

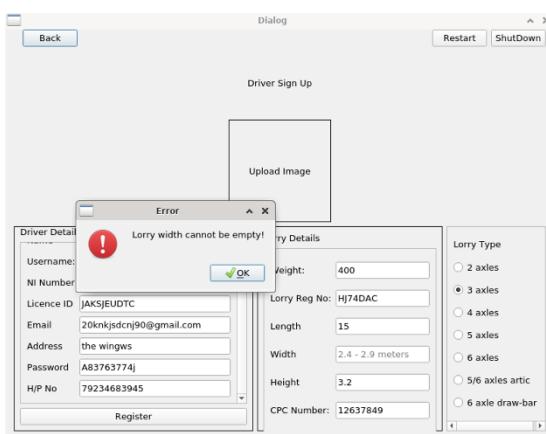


Figure 30

Input empty Lorry height

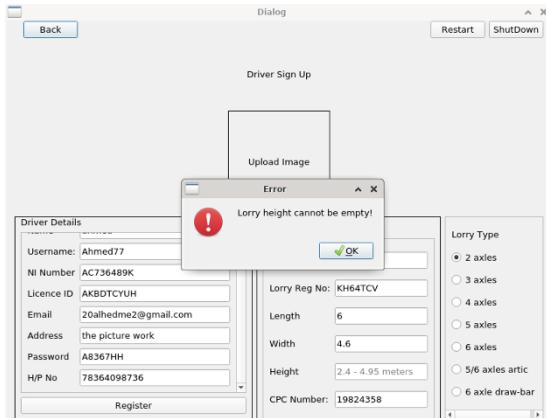


Figure 31

Input empty CPC Number:

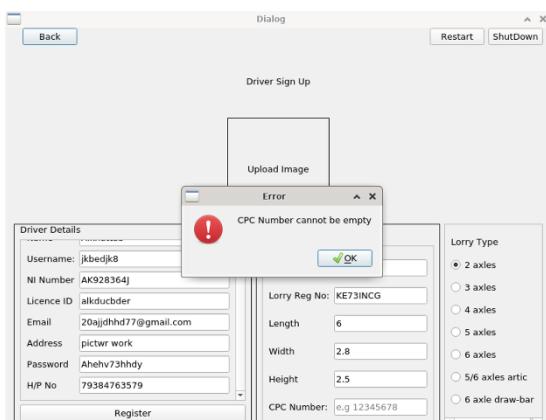


Figure 32

Input invalid format of CPC:

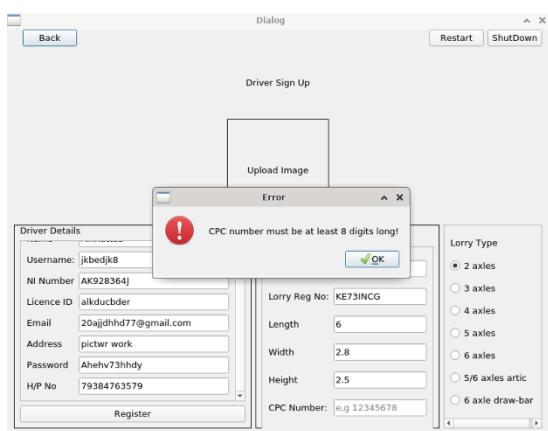


Figure 33

Did not Select a vehicle type:

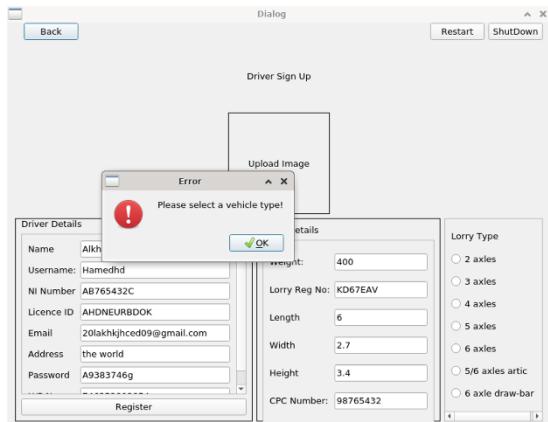


Figure 34

Input Driver's ID has been used:

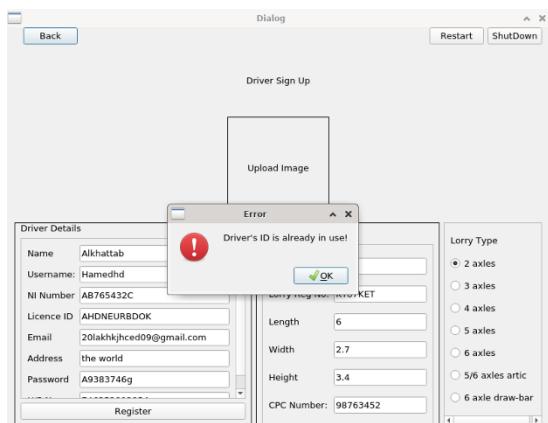


Figure 35

Input NI number has been used:

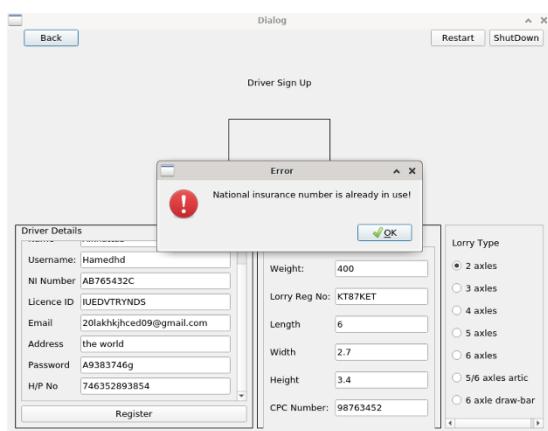


Figure 36

Input Phone number has been used:



Figure 37

Input Username has been used:

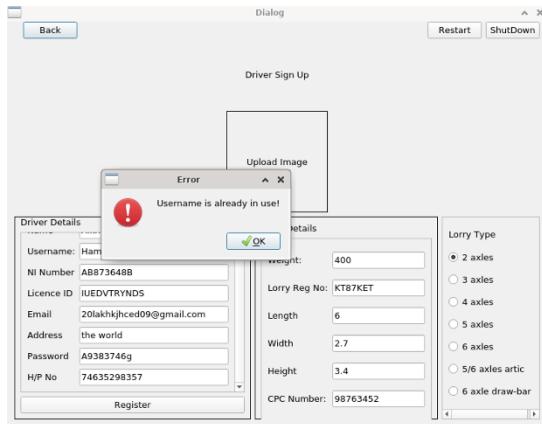


Figure 38

Input Email address is already use:

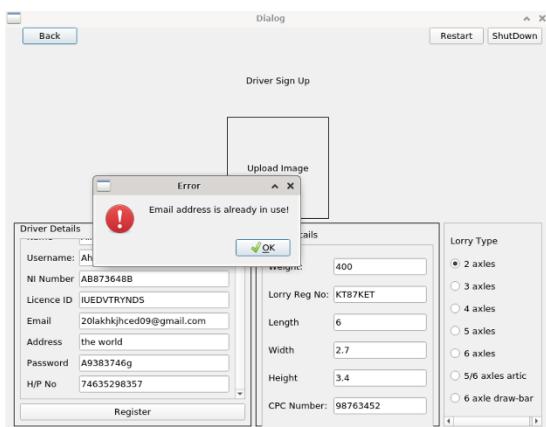


Figure 39

Test Report for Creating a New Driver Account

Test Plan ID: 002

Test Date: 2023-04-10

Tester: -

Introduction

The purpose of this test was to verify that a new driver account can be created successfully with the provided driver, lorry, and lorry type details, and that the details are saved accurately in the database. The test was performed on Qt Creator.

Conclusion

The test was successful, and it can be concluded that the driver account creation process is working as expected. No issues or errors were encountered during the test, and the details were saved accurately in the database. Therefore, the test plan has passed.

Recommendations

No recommendations for improvements were identified during the testing process.

ID	003	Description:	This test plan aims to ensure that a new company account can be created successfully with the provided details including company name, address, number of employees and contact information.
Test type	Quantity /Quality	Success criteria:	- A new company account should be created with the provided details. - The company details should be saved accurately in the database.
Number of attempts:	12	Comments:	-
List of equipment/requirements	QT Creator.		
Setup instructions	1. Open Qt Creator. 2. Open the project containing the customer account creation code. 3. Build the project to ensure that it is free of errors. 4. Start the application. 5. click into create new account button. 5. click into Company. 6. Fill in the required Name, Username, Password, Email, Address, Phone Number, Company Number, Company Address, and Number of Employees. 7. Click on the "Register" button.		
Failure correction procedure	1. If the account creation fails, an error message should be displayed. 2. Check the error message and take corrective action as necessary. 3. Retry the account creation process.		
Engineer(s)/Technician(s)	-		
Individual results	Pass / Fail		
Test Date:	10/04/20 23	Date:	10/04/2023

Input an empty name:

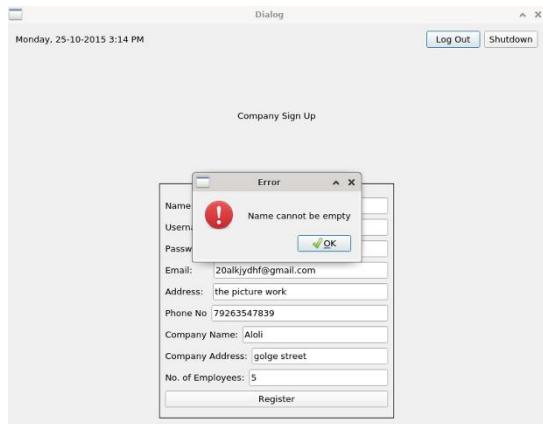


Figure 40

Input invalid username:

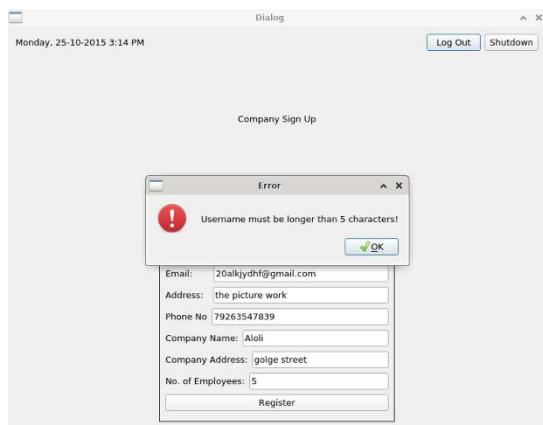


Figure 41

Input Invalid password format:

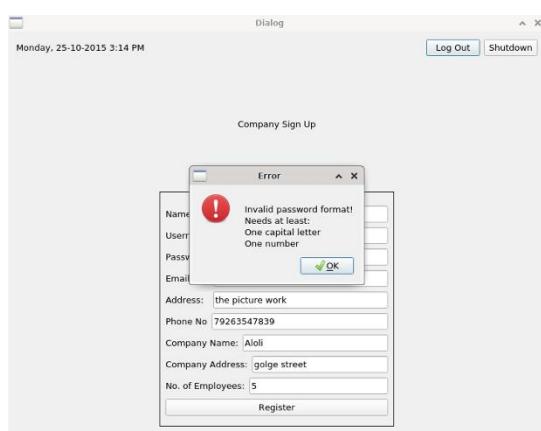


Figure 42

Input invalid email format:

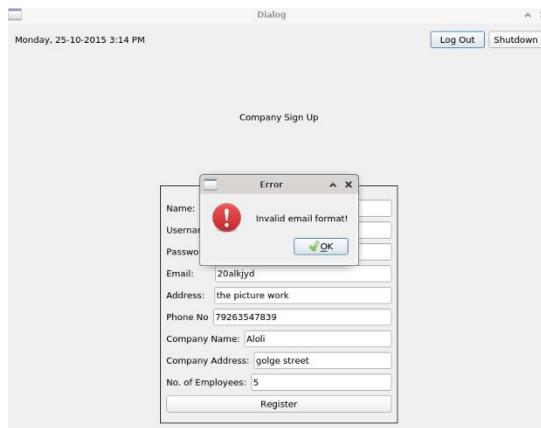


Figure 43

Input empty address:



Figure 44

Input empty phone number:

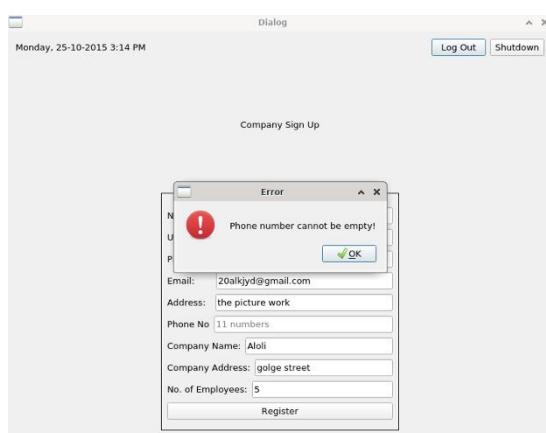


Figure 45

Empty Company name:

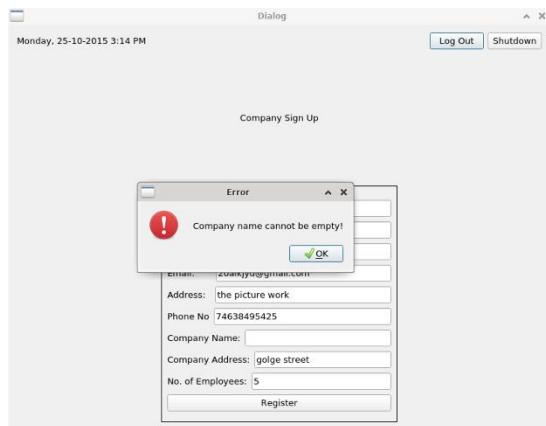


Figure 46

Input invalid Company name:

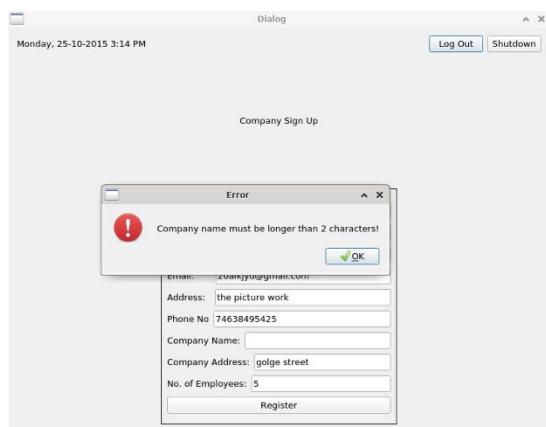


Figure 47

Input empty Company address:

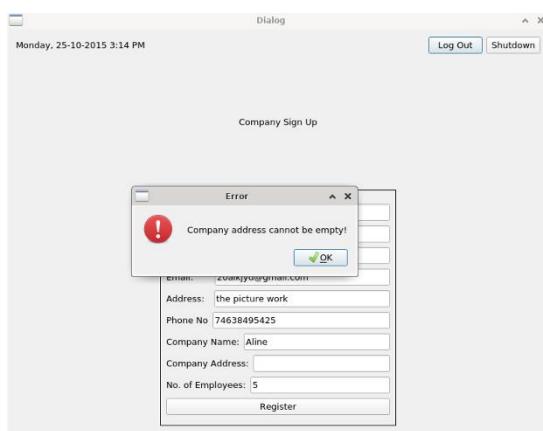


Figure 48

Empty number of employees:

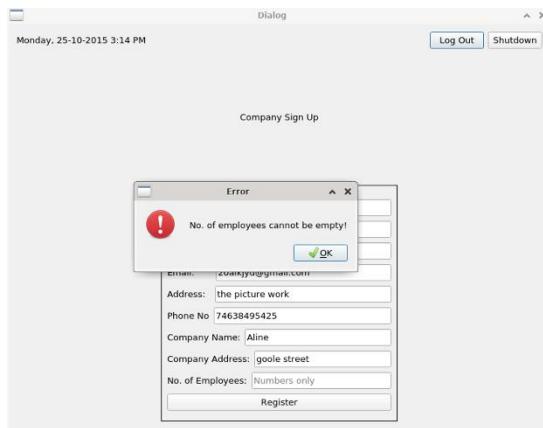


Figure 49

Input invalid input:

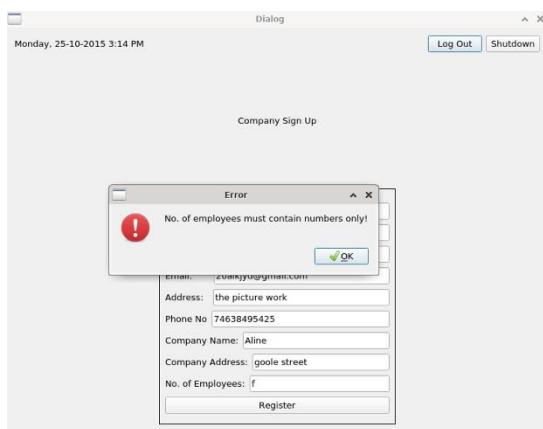


Figure 50

Successfully sign into company account:

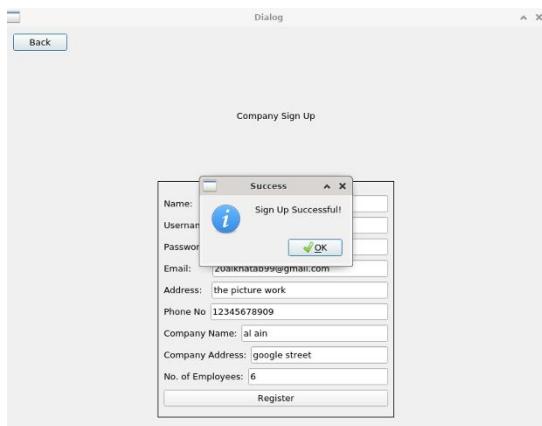


Figure 51

Test Report for Creating a New Company Account

Test Plan ID: 003

Test Date: 2023-04-10

Tester: -

Introduction

The purpose of this test was to verify that a new company account can be created successfully with the provided company details, and that the details are saved accurately in the database. The test was performed on a computer with Qt Creator.

Conclusion

The test was successful, and it can be concluded that the company account creation process is working as expected. No issues or errors were encountered during the test, and the details were saved accurately in the database. Therefore, the test plan has passed.

Recommendations

No recommendations for improvements were identified during the testing process.

ID	004	Description:	The purpose of this test is to verify that a cargo owner can log in to the system using their credentials.
Test type	Quantity / Quality	Success criteria:	The cargo owner is able to log in to the system using their correct credentials. The cargo owner is denied access if the provided credentials are incorrect.
Number of attempts:	4	Comments:	-
List of equipment/requirements	QT Creator.		
Setup instructions	1. Open Qt Creator. 2. Open the project containing the customer account creation code. 3. Build the project to ensure that it is free of errors. 4. Start the application. 5. click into Log In button for cargo owner.		
Failure correction procedure	If the cargo owner is unable to log in to the system, the following steps will be taken to correct the issue: -Verify that the correct username and password have been entered. -Have not registered.		
Engineer(s)/Technician(s)	-		
Individual results	Pass / Fail		
Test Date:	11/04/20 23	Date:	11/04/2023

Sign in successful:

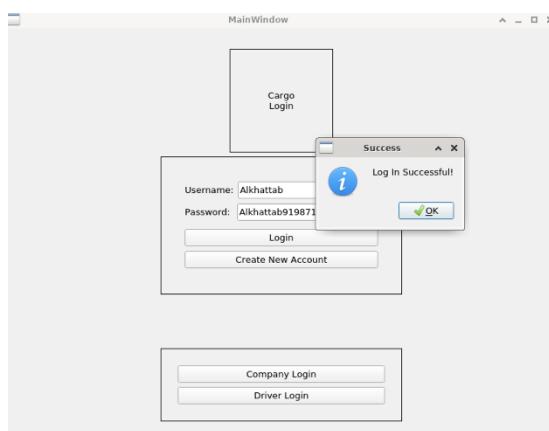


Figure 52

Input wrong details:

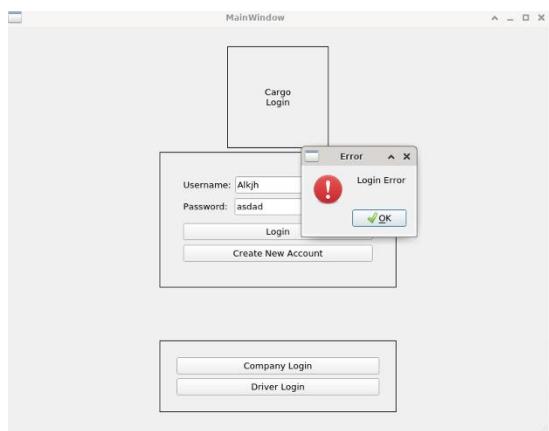


Figure 53

Input empty username:

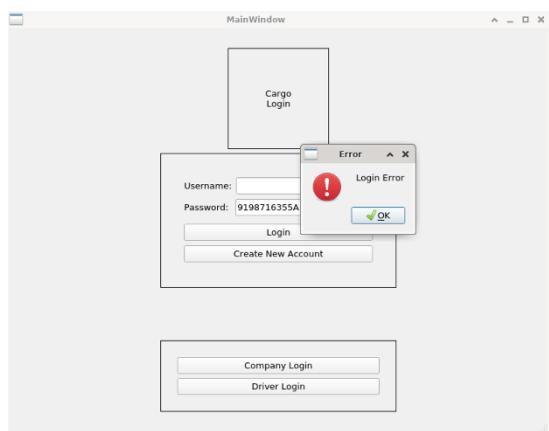


Figure 54

Input wrong password:

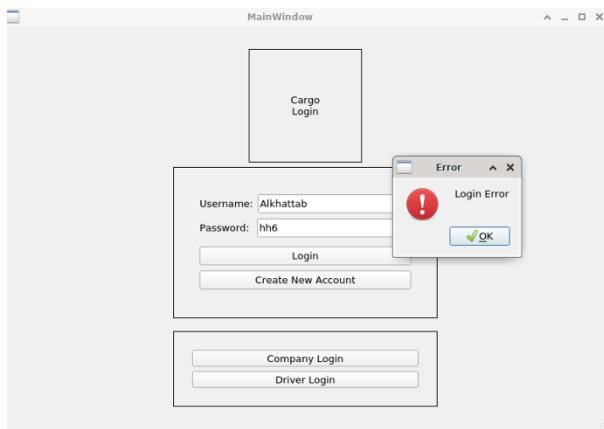


Figure 55

Test Report: Cargo Owner Login

Test Plan ID: 004

Test Date: 2023-05-11

Tester: -

Summary

The purpose of this test was to verify that a cargo owner can log in to the system using their credentials. The test was successful, as the cargo owner was able to log in to the system using their correct credentials on attempt.

Conclusion

Based on the test results, it can be concluded that the cargo owner login process is functioning as expected. The success criteria were met, and the cargo owner was able to log in to the system using their correct credentials.

Recommendations

No issues were identified during the test, and there is a small recommendation when the cargo owner logs in successful tow successfully, messages pop up.

ID	005	Description:	The purpose of this test is to verify that a driver can log in to the system using their credentials.
Test type	Quantity / Quality	Success criteria:	The driver is able to log in to the system using their correct credentials. The driver is denied access if the provided credentials are incorrect.
Number of attempts:	4	Comments:	-
List of equipment/requirements	QT Creator.		
Setup instructions	1. Open Qt Creator. 2. Open the project containing the customer account creation code. 3. Build the project to ensure that it is free of errors. 4. Start the application. 5. Click into Log In button for driver. 6. Enter your details than click log in.		
Failure correction procedure	If the driver is unable to log in to the system, the following steps will be taken to correct the issue: -Verify that the correct username and password have been entered. -Have not registered.		
Engineer(s)/Technician(s)	-		
Individual results	Pass / Fail		
Test Date:	11/04/20 23	Date:	11/04/2023

Successful log in:

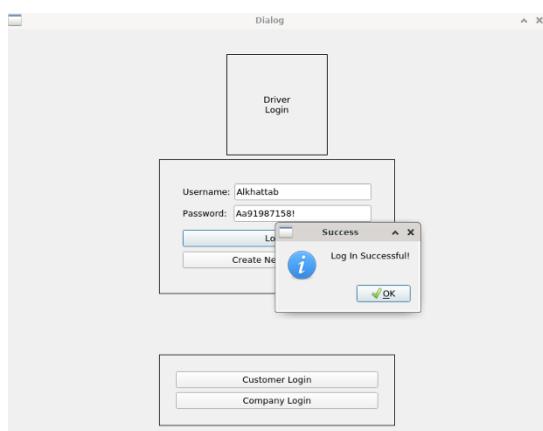


Figure 56

Input invalid details:

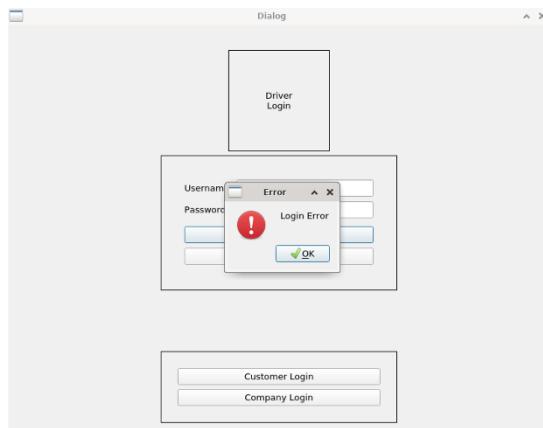


Figure 57

Input wrong password:

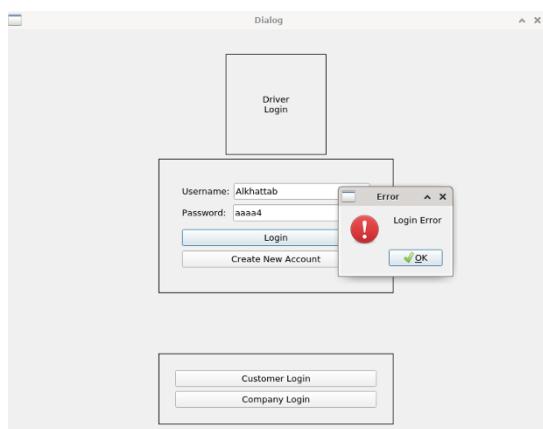


Figure 58

Input empty details:

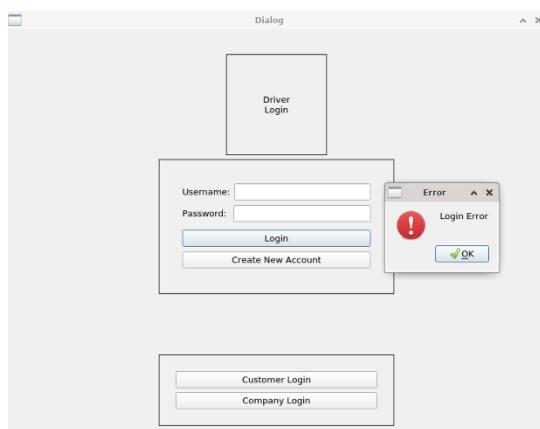


Figure 59

Test Report: driver Login

Test Plan ID: 005

Test Date: 2023-05-11

Tester: -

Summary

The purpose of this test was to verify that a driver can log in to the system using their credentials. The test was successful, as the driver was able to log in to the system using their correct credentials on attempt.

Conclusion

Based on the test results, it can be concluded that the driver login process is functioning as expected. The success criteria were met, and the driver was able to log in to the system using their correct credentials.

Recommendations

No issues were identified during the test, and there is a small recommendation when the driver logs in successfully tow successfully, messages pop up.

ID	006	Description:	The purpose of this test is to verify that a company can log in to the system using their credentials.
Test type	Quantity <i>/Quality</i>	Success criteria:	The company can log in to the system using the correct credentials. The company is denied access if the provided credentials are incorrect.
Number of attempts:	5	Comments:	-
List of equipment/requirements	QT Creator.		
Setup instructions	1. Open Qt Creator. 2. Open the project containing the customer account creation code. 3. Build the project to ensure that it is free of errors. 4. Start the application. 5. click into Log In button for company. 6. Enter your details than click log in.		
Failure correction procedure	If the company is unable to log in to the system, the following steps will be taken to correct the issue: -Verify that the correct username and password have been entered. -Have not registered.		
Engineer(s)/Technician(s)	-		
Individual results	Pass / Fail		
Test Date:	11/04/20 23	Date:	11/04/2023

Log in successfully:

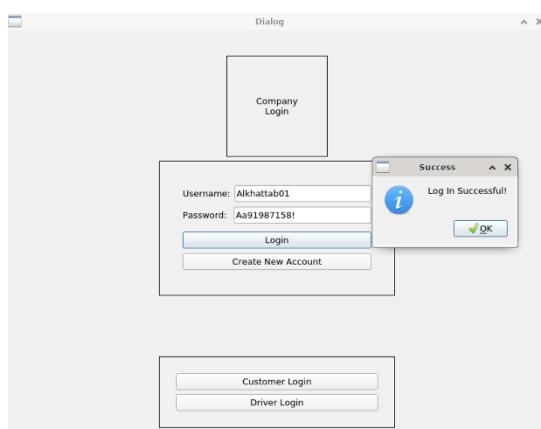


Figure 60

Input empty password:

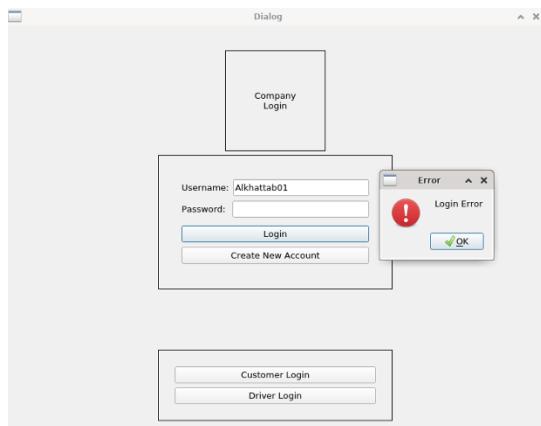


Figure 61

Input empty username:

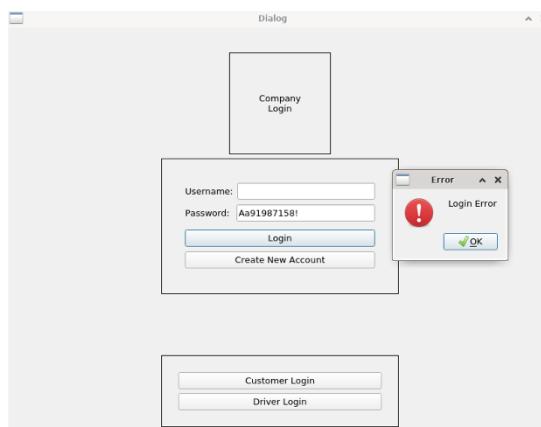


Figure 62

Input wrong username:

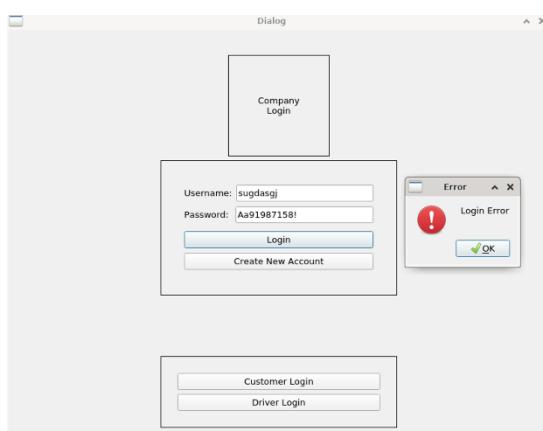


Figure 63

Input invalid details:

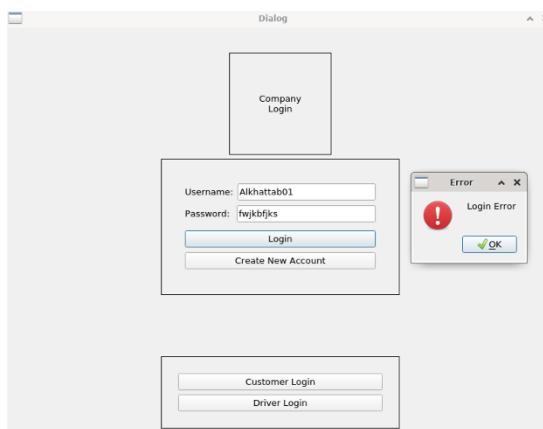


Figure 64

Test Report: Company Login

Test Plan ID: 004

Test Date: 2023-05-11

Tester: -

Summary

The purpose of this test was to verify that a Company can log in to the system using their credentials. The test was successful, as the Company was able to log in to the system using their correct credentials on attempt.

Conclusion

Based on the test results, it can be concluded that the Company login process is functioning as expected. The success criteria were met, and the Company was able to log in to the system using their correct credentials.

Recommendations

No issues were identified during the test, and there is a small recommendation when the Company logs in successful tow successfully, messages pop up.

ID	007	Description:	Verify that customers can view and edit their profile details.
Test type	Quantity / Quality	Success criteria:	1. The customer is able to view their profile details. 2. The customer is able to edit their profile details such as Name, Username, Password, Email, Address, Phone. 3. The edited profile details are saved correctly.
Number of attempts:	6	Comments:	Ensure that all information is displayed correctly and that the customer can view and edit their information
List of equipment/requirements	QT Creator.		
Setup instructions	1. Open Qt Creator. 2. Open the project containing the customer account creation code. 3. Build the project to ensure that it is free of errors. 4. Start the application. 5. enter your Log In details. 6. click into edit details button. 6. edit details and View profile.		
Failure correction procedure	In case of a failure, the engineer or technician should investigate the root cause of the issue and try to fix it. If the issue persists, the test should be considered a failure.		

Engineer(s)/ Technician(s)	-		
Individual results	Pass / Fail		
Test Date:	12/04/2023	Date:	12/04/2023

Name has been changed:

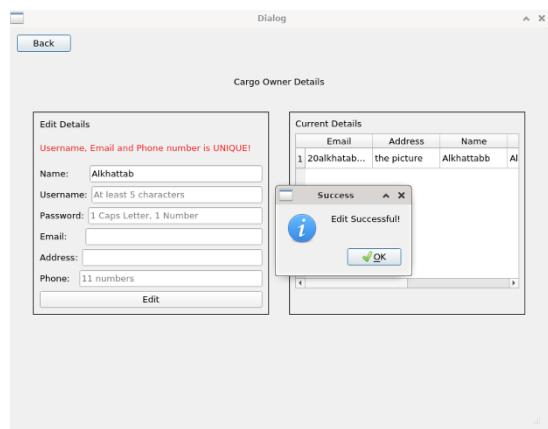


Figure 65

Username has been changed:

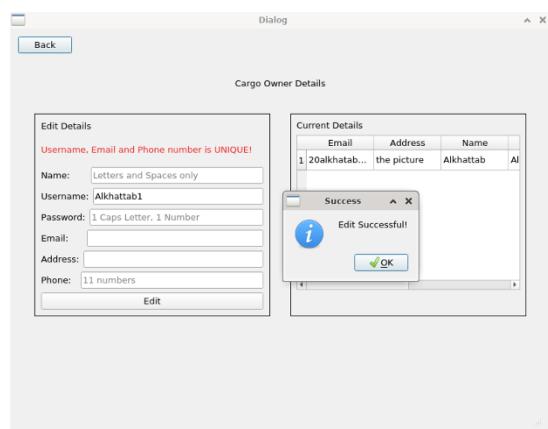


Figure 66

Password has been changed:

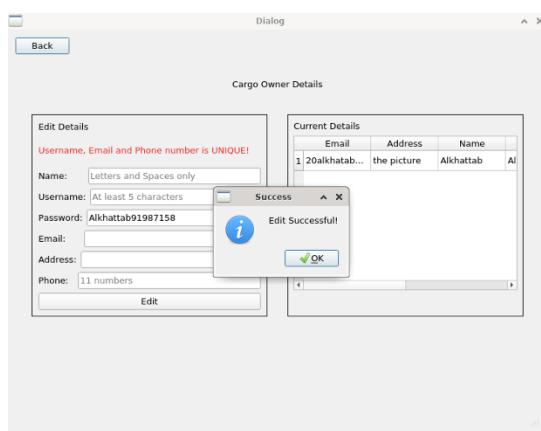


Figure 67

Email has been changed:

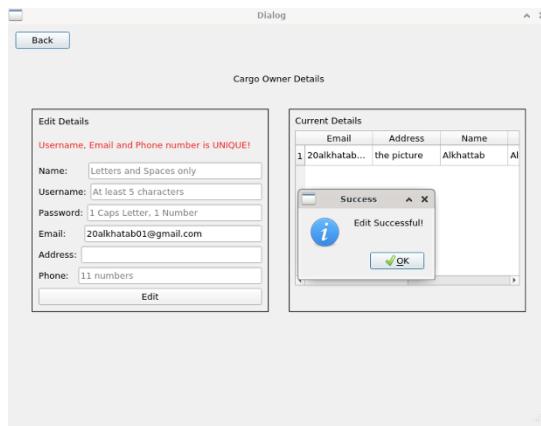


Figure 68

Address has been changed:

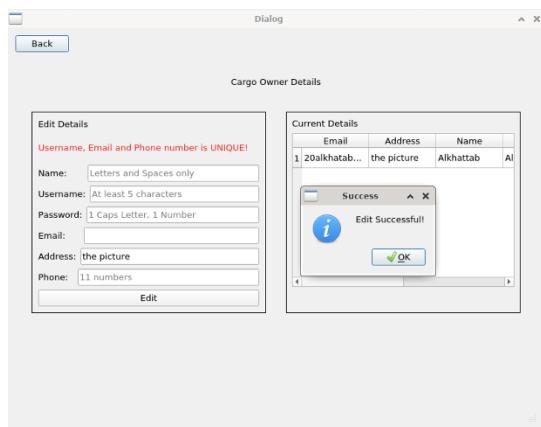


Figure 69

Phone number has been changed:

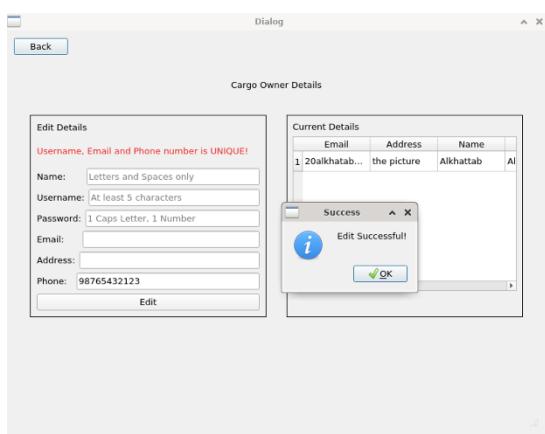


Figure 70

Introduction:

This test plan is designed to ensure that the system can accurately view and edit customer details. The testing process will be carried out in different stages to ensure all aspects of the system have been checked and are functioning as expected.

Conclusion:

The testing process has been carried out successfully, and all aspects of the system have been checked for accuracy and functionality. The tests have been performed in different stages to ensure that all aspects of the system are covered, and the system is working as expected. Any errors that were identified were documented and addressed, ensuring that the system is operating optimally.

Recommendations

No recommendations for improvements were identified during the testing process.

ID	008	Description:	Place cargo order from source to destination
Test type	Quantity / Quality	Success criteria:	Cargo order is successfully placed with accurate details
Number of attempts:	6	Comments:	-
List of equipment/requirements	QT Creator.		

Setup instructions	1. Open Qt Creator. 2. Open the project containing the customer account creation code. 3. Build the project to ensure that it is free of errors. 4. Start the application. 5. click into Log In button for company. 6. Enter your details than click log in.
Failure correction procedure	If the order cannot be placed, verify the details entered and retry the process.
Engineer(s)/Technician(s)	-
Individual results	Pass / Fail
Test Date:	12/04/2023 23
Date:	14/04/2023

Input invalid length:

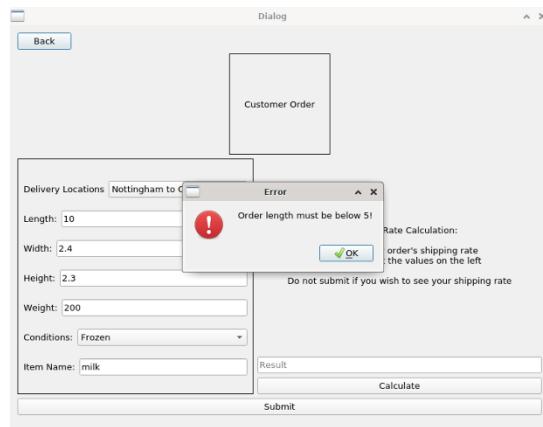


Figure 71

Input invalid width:

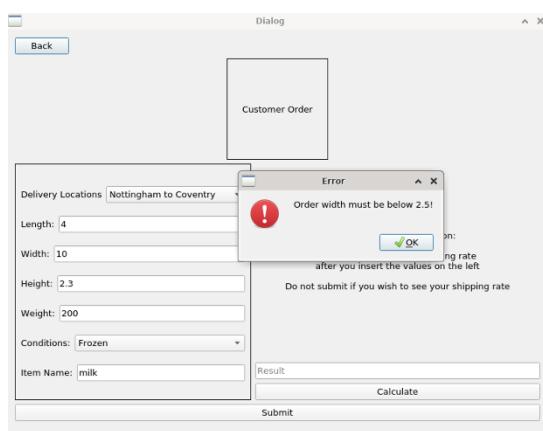


Figure 72

Input invalid height:

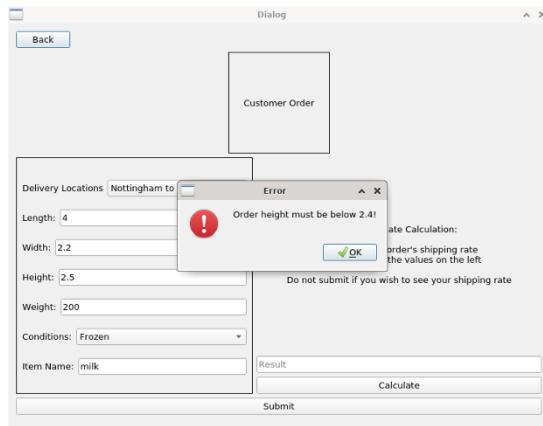


Figure 73

Input invalid weight:

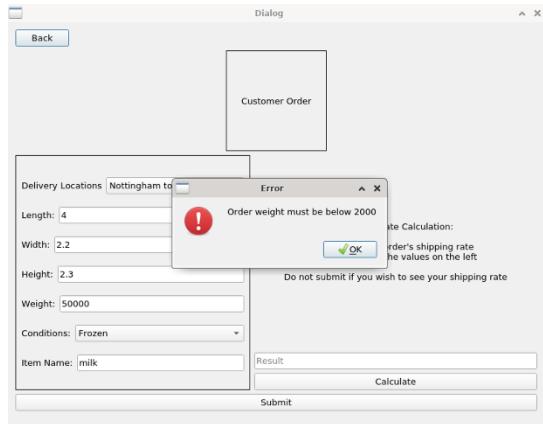


Figure 74

Input empty name:

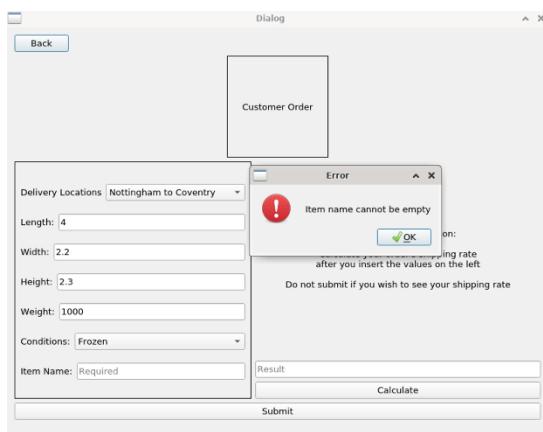


Figure 75

Cargo owner have been made successfully:

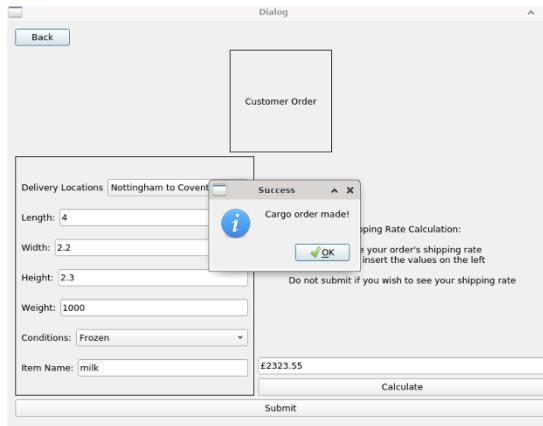


Figure 76

Introduction:

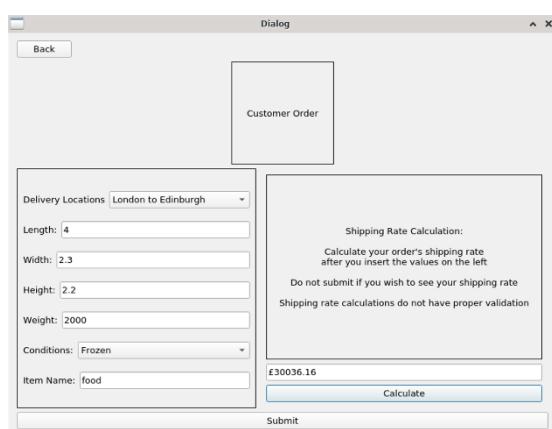
The purpose of this test plan is to ensure that the functionality of placing a cargo order from a source to a destination with accurate details is tested thoroughly. The test plan covers the required fields and checks that are needed to ensure a successful cargo order placement.

Conclusion:

The test plan for placing a cargo order was successful in ensuring that the functionality of placing a cargo order from a source to a destination with accurate details was thoroughly tested.

ID	009	Description:	Calculate shipping rates based on source/destination and lorry type.
Test type	Quantity / Quality	Success criteria:	The system should accurately calculate shipping rates based on the input parameters.
Number of attempts:	3	Comments:	-
List of equipment/requirements	QT Creator.		
Setup instructions	1. Open Qt Creator. 2. Open the project containing the customer account creation code. 3. Build the project to ensure that it is free of errors. 4. Start the application. 5.log into driver.		
Failure correction procedure	Correct any input errors and try again.		
Engineer(s)/Technician(s)	-		
Individual results	Pass / Fail		
Test Date:	14/04/20 23	Date:	14/04/2023

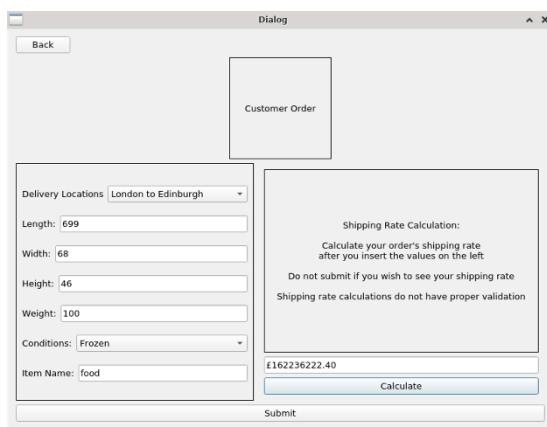
Shipping rates has been calculated successfully:



The screenshot shows a 'Dialog' window with the title 'Customer Order'. On the left, there is a form with fields for 'Delivery Locations' (set to 'London to Edinburgh'), 'Length' (4), 'Width' (2.3), 'Height' (2.2), 'Weight' (2000), 'Conditions' (Frozen), and 'Item Name' (food). On the right, there is a 'Shipping Rate Calculation' section with a note: 'Calculate your order's shipping rate after you insert the values on the left. Do not submit if you wish to see your shipping rate. Shipping rate calculations do not have proper validation'. Below this note, there is a text field showing '£30036.16' and two buttons: 'Calculate' and 'Submit'.

Figure 77

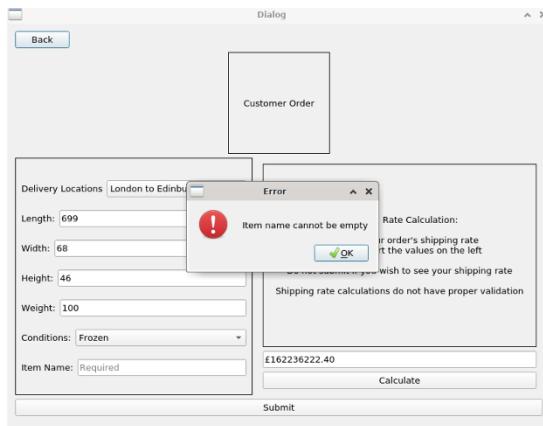
Input invalid numbers:



The screenshot shows a 'Dialog' window titled 'Customer Order'. On the left, there's a form with fields for Delivery Locations (set to 'London to Edinburgh'), Length (699), Width (68), Height (46), Weight (100), Conditions (Frozen), and Item Name (empty). On the right, a panel titled 'Shipping Rate Calculation' contains instructions: 'Calculate your order's shipping rate after you insert the values on the left', 'Do not submit if you wish to see your shipping rate', and 'Shipping rate calculations do not have proper validation'. Below these are a text input field containing '£162236222.40' and a 'Calculate' button. At the bottom is a 'Submit' button.

Figure 78

Input empty name:



This screenshot is similar to Figure 78, showing the 'Customer Order' dialog. The 'Item Name' field is empty. A modal error dialog box appears in the center, titled 'Error', with the message 'Item name cannot be empty'. It has an 'OK' button. The background form remains mostly the same, with the 'Item Name' field highlighted in red.

Figure 79

Introduction:

The purpose of this test plan is to ensure that the shipping rate calculation feature of the system is tested thoroughly to meet the desired success criteria. This feature is critical as it ensures that customers are provided with accurate shipping rates based on the source, destination, and lorry type.

Conclusion:

The shipping rates have been calculated successfully; just one thing to mention if the user inputs an invalid number of the shipping rates, the calculator will calculate those numbers.

ID	011	Description:	Receive order from customer.
Test type	Quantity/ Quality	Success criteria:	The company should receive a notification whenever a customer places an order.
Number of attempts:	1	Comments:	-
List of equipment/requirements	QT Creator.		
Setup instructions	1. Open Qt Creator. 2. Open the project containing the customer account creation code. 3. Build the project to ensure that it is free of errors. 4. Start the application. 5.log into company.		
Failure correction procedure	-		
Engineer(s)/Technician(s)	-		
Individual results	Pass / Fail		
Test Date:	15/04/2023	Date:	15/04/2023

Successfully received order from customer:

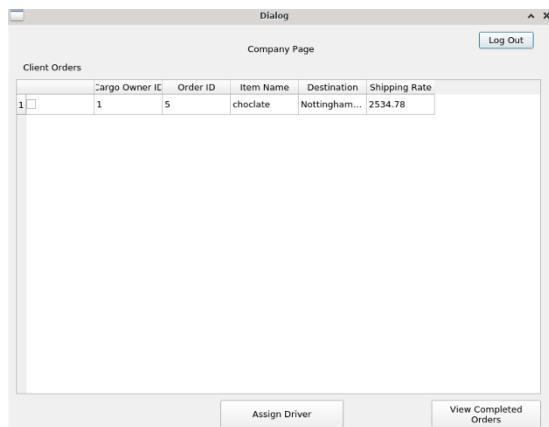


Figure 80

Result:

In this scenario, the company received an order notification within the specified time frame, and the test result is a pass.

ID	012	Description:	Calculate company commission for each order
----	-----	--------------	---

Test type	Quantity/ Quality	Success criteria:	The system accurately calculates the company commission for each order
Number of attempts:	1	Comments:	-
List of equipment/requirements	QT Creator.		
Setup instructions	1. Open Qt Creator. 2. Open the project containing the customer account creation code. 3. Build the project to ensure that it is free of errors. 4. Start the application. 5.log into company.		
Failure correction procedure	Identify the issue and retest with different orders to isolate the problem.		
Engineer(s)/Technician(s)	-		
Individual results	Pass / Fail		
Test Date:	15/04/2023	Date:	15/04/2023

Successfully calculate company commission for order:

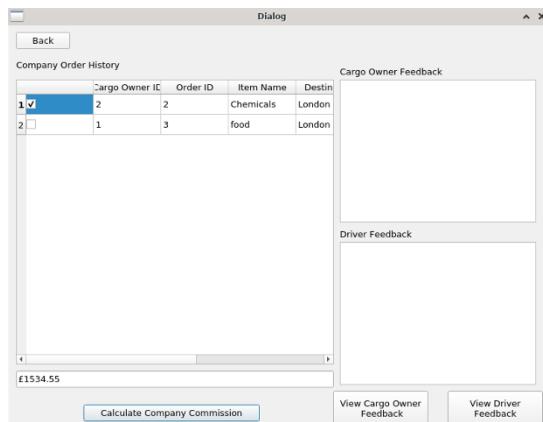


Figure 81

Result: The commission was calculated accurately and matched the expected value.

ID	013	Description:	This test plan is designed to test the functionality of the system for the company to view feedback for each cargo owner and drive
Test type	Quantity/ Quality	Success criteria:	The test will be considered successful if the system allows the

			company to view feedback for each cargo owner and driver accurately and efficiently.
Number of attempts:	1	Comments:	-
List of equipment/requirements	QT Creator.		
Setup instructions	1. Open Qt Creator. 2. Open the project containing the customer account creation code. 3. Build the project to ensure that it is free of errors. 4. Start the application. 5.log into company.		
Failure correction procedure	Feedback should be displayed in a clear and concise manner.		
Engineer(s)/Technician(s)	-		
Individual results	Pass / Fail		
Test Date:	15/04/2023	Date:	15/04/2023

Result:

The test was successfully passed. The system was able to display feedback for each cargo owner and driver accurately and efficiently. The results were documented accurately for each cargo owner and driver.

Successfully view feedback from cargo owner and Driver:

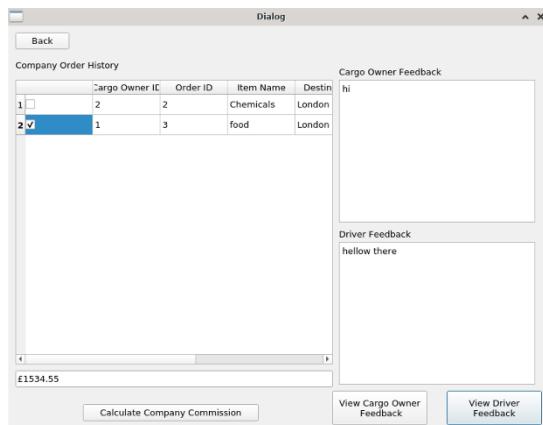


Figure 82

ID	014	Description:	Company can issuing invoice to the customer when the order was accepted by a driver.
Test type	Quantity/ Quality	Success criteria:	The system should generate and send an accurate invoice to the customer immediately after the order is accepted by a driver.
Number of attempts:	1	Comments:	-
List of equipment/requirements	QT Creator.		
Setup instructions	1. Open Qt Creator. 2. Open the project containing the customer account creation code. 3. Build the project to ensure that it is free of errors. 4. Start the application.		
Failure correction procedure	-		
Engineer(s)/Technician(s)	-		
Individual results	Pass / Fail		
Test Date:	15/04/2023	Date:	15/04/2023

Driver has been assigned successfully:

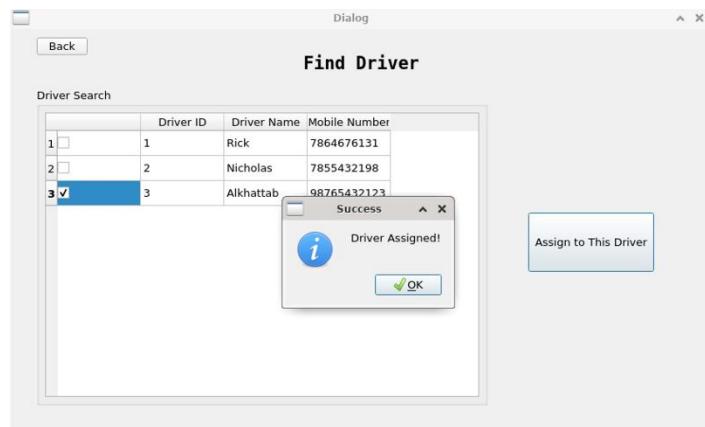


Figure 83

ID	015	Description:	This test plan is designed to ensure that the driver is able to accept or
----	-----	--------------	---

			reject cargos that are assigned to them.
Test type	Quantity/ Quality	Success criteria:	The driver should be able to accept or reject the assigned cargos without any issues.
Number of attempts:	2	Comments:	-
List of equipment/requirements	QT Creator.		
Setup instructions	1. Open Qt Creator. 2. Open the project containing the customer account creation code. 3. Build the project to ensure that it is free of errors. 4. Start the application. 5.log into driver.		
Failure correction procedure			
Engineer(s)/Technician(s)	-		
Individual results	Pass / Fail		
Test Date:	15/04/2023	Date:	15/04/2023

Order has been accepted successfully:

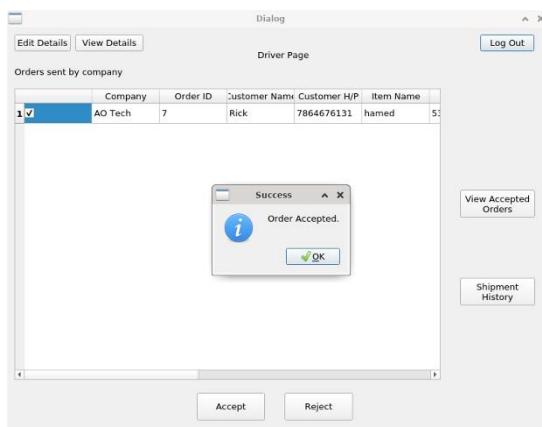


Figure 84

Order has been rejected successfully:

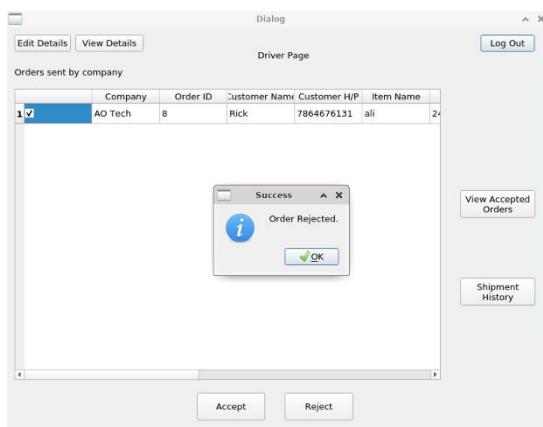


Figure 85

Result:

The system has passed the test as the driver was able to accept and reject cargo requests without any issues. The test was successful in verifying the functionality of the system for this feature.

ID	016	Description:	Test the ability of a driver to view their shipment history.
Test type	Quantity/Quality	Success criteria:	<ul style="list-style-type: none"> Driver is able to access their shipment history. Shipment history displays all previous shipments. Shipment history includes relevant information about each shipment such as pickup/delivery locations, dates, and cargo details. Shipment history is up-to-date and accurate.
Number of attempts:	1	Comments:	-
List of equipment/requirements	QT Creator.		
Setup instructions	1. Open Qt Creator. 2. Open the project containing the customer account creation code. 3. Build the project to ensure that it is free of errors.		

	4. Start the application. 5.log into driver.		
Failure correction procedure			
Engineer(s)/Technician(s)	-		
Individual results	Pass / Fail		
Test Date:	15/04/2023	Date:	15/04/2023

Shipment history has been viewed successfully:

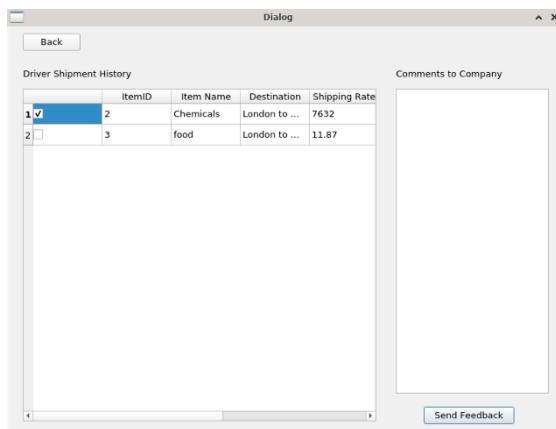


Figure 86

Results:

Pass: Driver is able to view their shipment history and all information is accurate and up-to-date.

ID	017	Description:	Test the ability of the driver to notify the cargo owner of the shipment status.
Test type	Quantity/Quality	Success criteria:	The driver is able to notify the cargo owner when the cargo is on the road.

			The driver is able to notify the cargo owner when the cargo has been delivered.
Number of attempts:	2	Comments:	-
List of equipment/requirements	QT Creator.		
Setup instructions	1. Open Qt Creator. 2. Open the project containing the customer account creation code. 3. Build the project to ensure that it is free of errors. 4. Start the application. 5.log into driver.		
Failure correction procedure	-		
Engineer(s)/Technician(s)	-		
Individual results	Pass / Fail		
Test Date:	15/04/2023	Date:	15/04/2023

Notified cargo owner successfully:

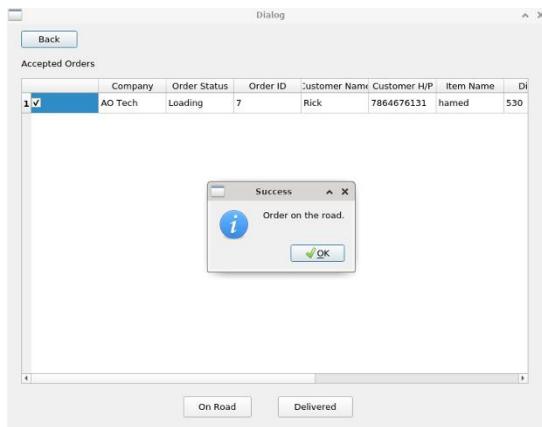


Figure 87

Notified cargo owner that order has been delivered:

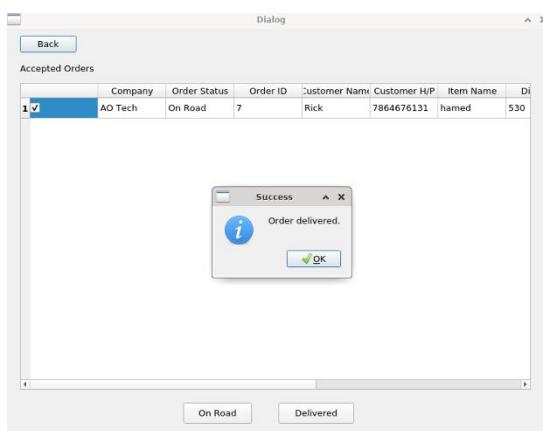


Figure 88

Result:

The test was successful, and the driver was able to notify the cargo owner of the shipment status without any issues.

ID	018	Description:	This test plan verifies that the cargo owner is able to view the invoice generated by the transportation company.
Test type	Quantity/Quality	Success criteria:	The cargo owner should be able to view the invoice details.
Number of attempts:	3	Comments:	-
List of equipment/requirements	QT Creator.		
Setup instructions	1. Open Qt Creator. 2. Open the project containing the customer account creation code. 3. Build the project to ensure that it is free of errors. 4. Start the application. 5.log into cargo owner.		
Failure correction procedure			
Engineer(s)/Technician(s)	-		
Individual results	Pass / Fail		
Test Date:	15/04/2023	Date:	15/04/2023

Successfully view invoice issued by transportation company:

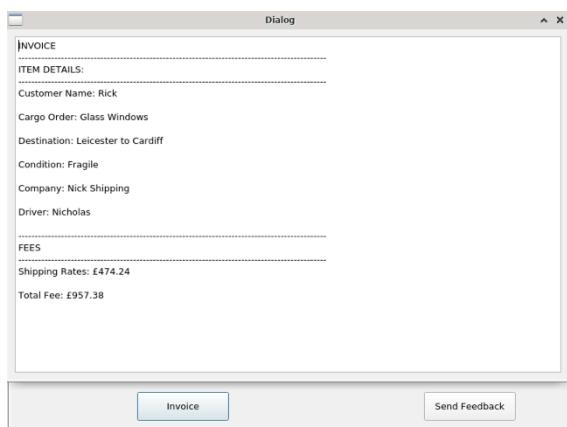


Figure 89

Conclusion:

This test plan has verified that the cargo owner is able to view the invoice generated by the transportation company for the cargo transportation service. The test results indicate that the invoice details are accurate and match the cargo transportation service provided.

ID	019	Description:	The purpose of this test plan is to verify that users can add comments and recommendations
Test type	Quantity/ Quality	Success criteria:	User is able to add a comment to a specific cargo or shipment. User is able to add a recommendation to a specific cargo or shipment.
Number of attempts:	1	Comments:	-
List of equipment/requirements	QT Creator.		
Setup instructions	1. Open Qt Creator. 2. Open the project containing the customer account creation code. 3. Build the project to ensure that it is free of errors. 4. Start the application. 5. log into cargo owner.		

Failure correction procedure			
Engineer(s)/Technician(s)	-		
Individual results	Pass / Fail		
Test Date:	15/04/2023	Date:	15/04/2023

Successfully added comment and recommendations:

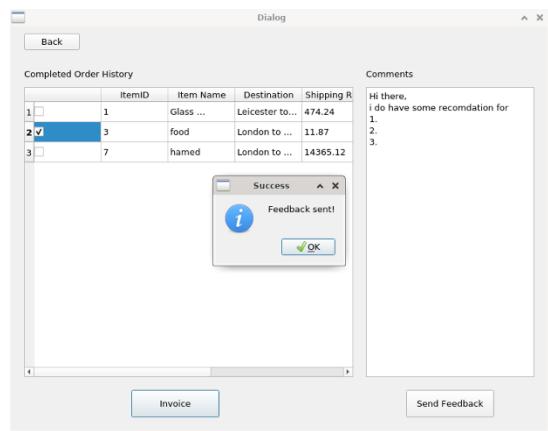


Figure 90

Result:

The user was able to add comments and recommendations successfully.

User Manual

Rick Richard Harith (T0321003)

There are three user types:

- Cargo Owner
- Drivers
- Transportation Company

The manual will be divided into these three users.

CARGO OWNER:

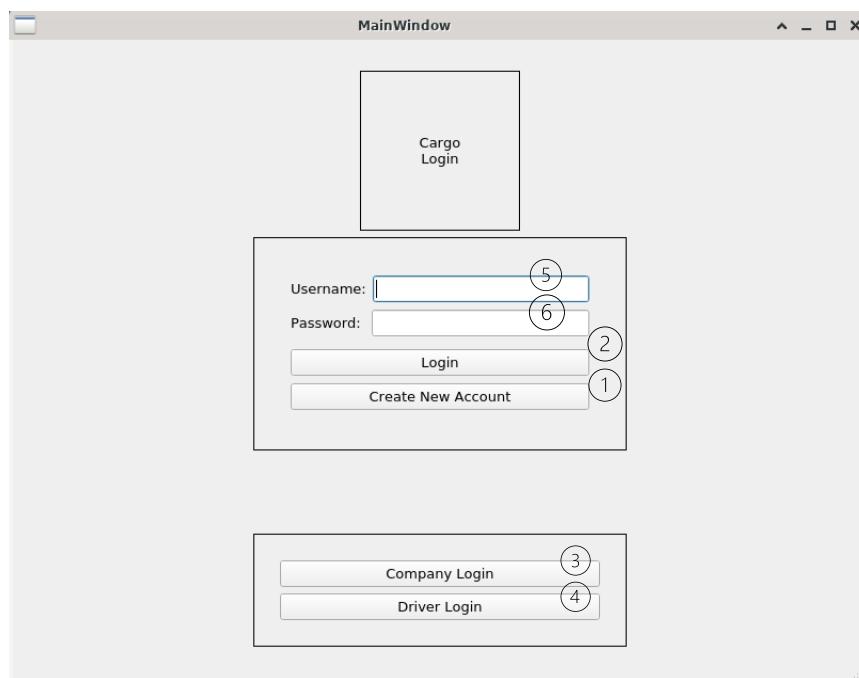


Figure 91. Cargo Owner Login Page

1. Cargo Owners should create a new account if no account exists.
2. If an account exists, log in with your account credentials.
3. This leads to the company login page.
4. This leads to the driver login page.
5. Username field for login.
6. Password field for login.

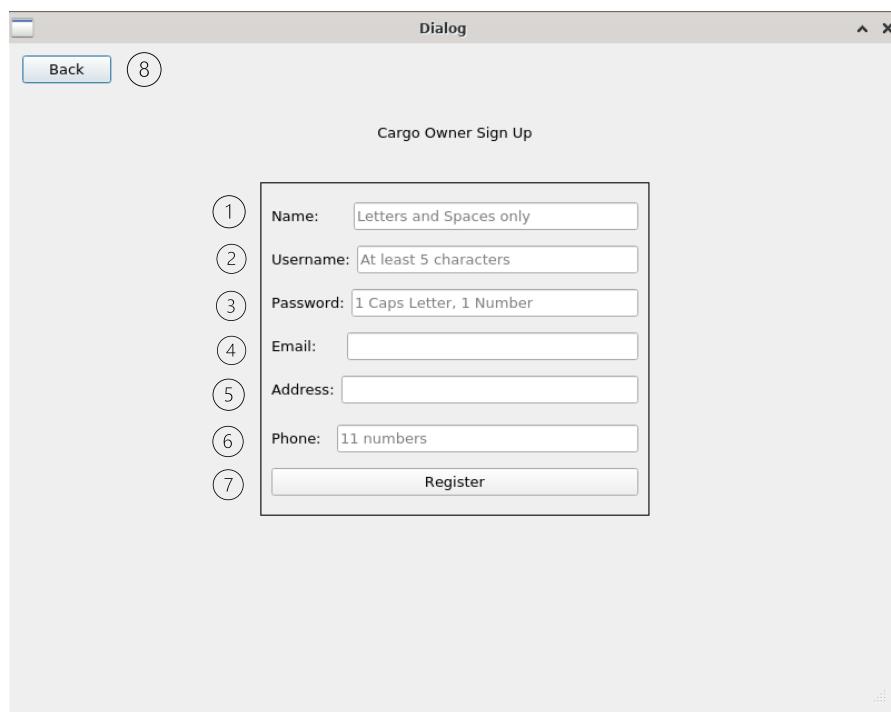


Figure 92. Cargo Owner Sign Up

1. Name field for your name.
2. Username needs to be at least 5 characters.
3. Password must be at least 5 characters.
4. E-mail field for e-mail, it must conform to the correct format.
5. Home address field takes any input.
6. Mobile number field must be more than eleven numbers.
7. Once all fields are filled, press the register button to sign up as a cargo owner.
8. Back button to navigate to the previous page.

After signing up, navigate back to the login page.

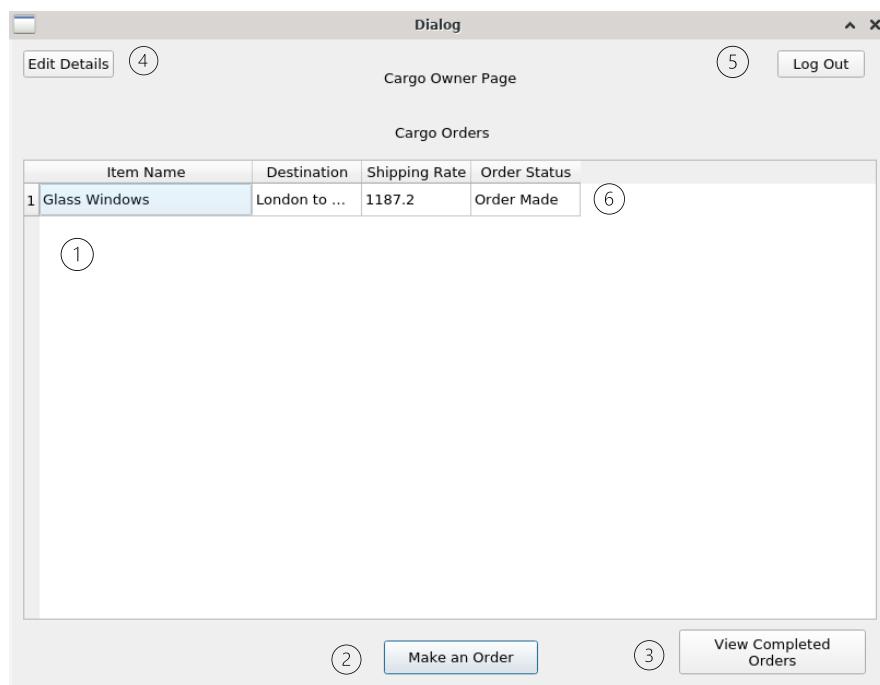
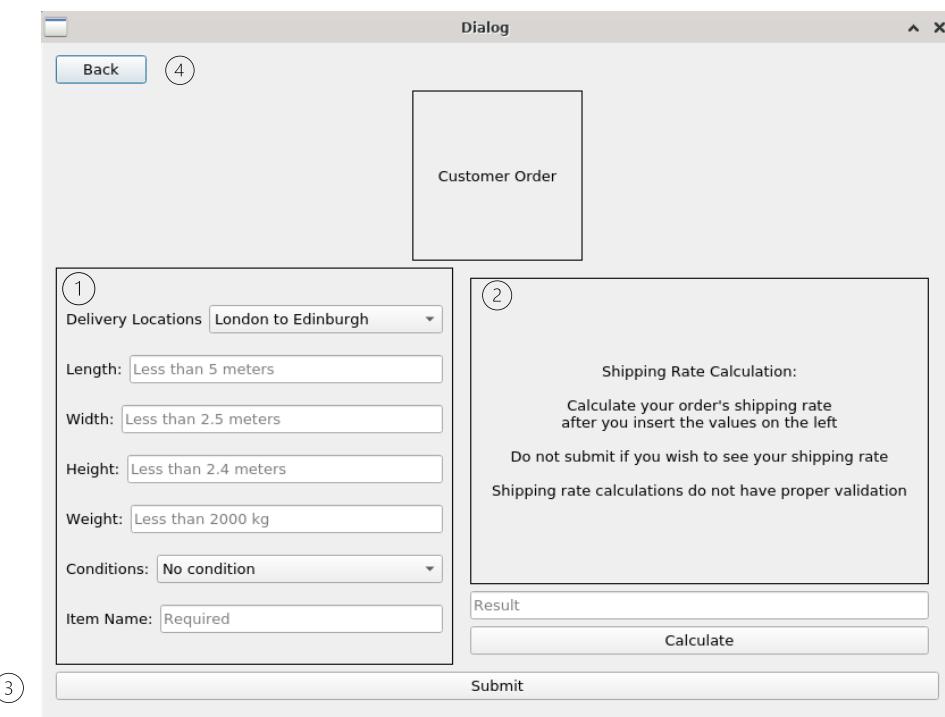


Figure 93. Cargo Owner Main Page

1. This is the table that displays all the cargo orders made, if no orders are made, it will be empty.
2. Leads to the page to place an order.
3. View the completed and delivered orders.
4. Edit profile details of the account.
5. Logs the user out, this will lead back to the login pages.
6. The orders made that appear in the table will have an order status, take note of the status to determine how the order is progressing, delivered orders will appear in the View Completed Orders page.



The screenshot shows a Windows-style dialog box titled "Dialog". At the top right are standard window controls for minimizing, maximizing, and closing. Below the title bar is a navigation bar with a "Back" button and a circled "4" button. The main area is divided into four sections:

- Section 1 (Left):** Contains input fields for "Delivery Locations" (set to "London to Edinburgh"), "Length" ("Less than 5 meters"), "Width" ("Less than 2.5 meters"), "Height" ("Less than 2.4 meters"), "Weight" ("Less than 2000 kg"), "Conditions" ("No condition"), and "Item Name" (marked as "Required").
- Section 2 (Right):** Contains a heading "Shipping Rate Calculation:" followed by the instruction "Calculate your order's shipping rate after you insert the values on the left". It also includes the note "Do not submit if you wish to see your shipping rate" and "Shipping rate calculations do not have proper validation". Below this is a "Result" field and a "Calculate" button.
- Section 3 (Bottom Left):** A "Submit" button.
- Section 4 (Top Left):** A "Back" button.

Figure 94. Place an Order Page

1. That form has the input fields required to place an order, be sure to follow the requirements for each field or the program will not place an order.
2. After the input fields for filled, cargo owners can calculate the shipping rates and the result will be printed above the calculate button.
3. The submit button should be pressed after all fields are typed, this will place an order for a transportation company to handle your order and have it delivered.

Cargo owner users will be returned back to the main page mentioned earlier, the order will appear in the main page with an order status to show how the order is going.

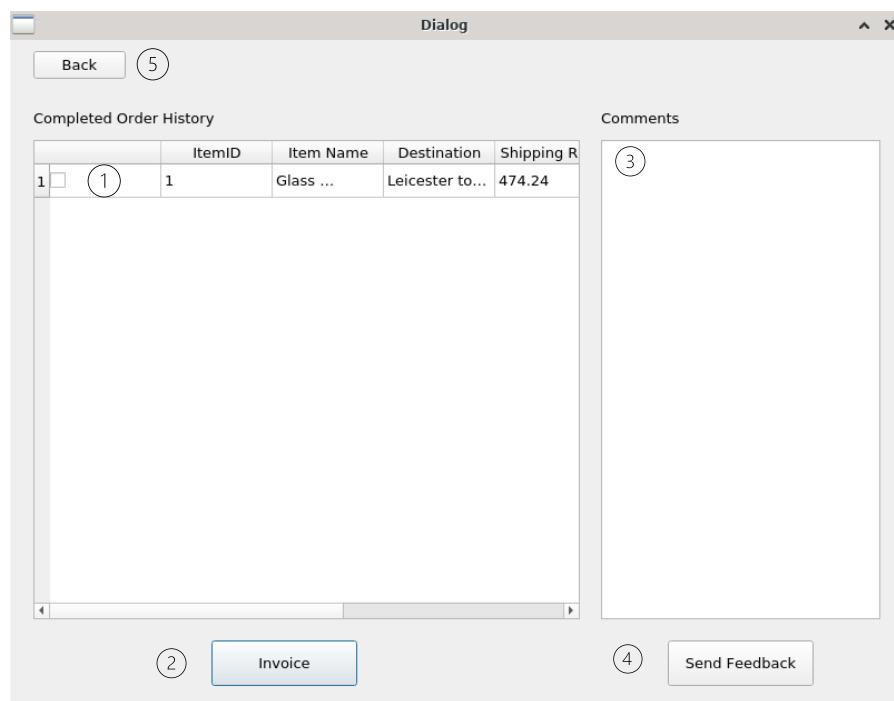
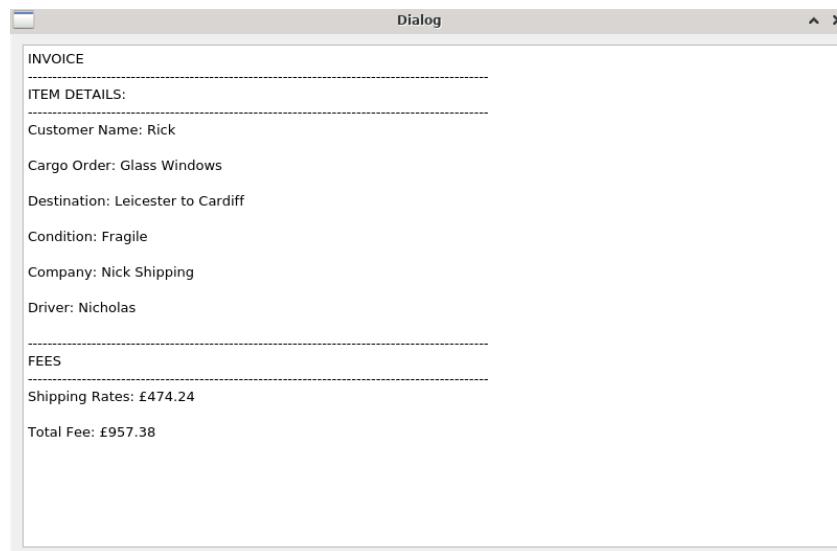


Figure 95. Completed Orders Page

1. This table shows completed orders for cargo owners, invoices can be viewed for the order and the feedback can be sent, the user is required to check the checkbox to do these functionalities.
2. An invoice can be viewed for the selected order.
3. Feedback can be written in this field and sent to the company handling the order.
4. Click the send feedback button after writing the feedback.
5. Leads cargo owners back to the main page.

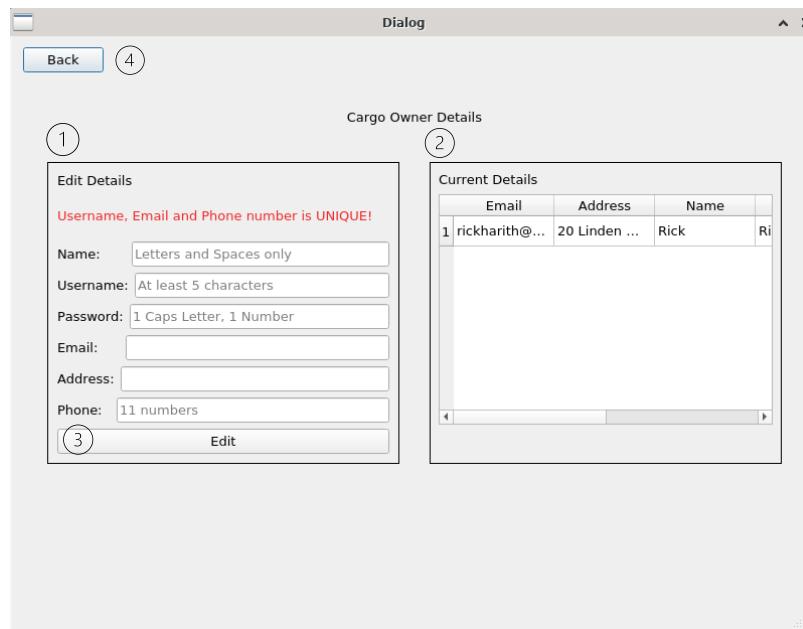
Here is an example of a generated invoice for an order:



The screenshot shows a 'Dialog' window titled 'INVOICE'. It contains two main sections: 'ITEM DETAILS:' and 'FEES'. Under 'ITEM DETAILS:', the following information is listed:
 Customer Name: Rick
 Cargo Order: Glass Windows
 Destination: Leicester to Cardiff
 Condition: Fragile
 Company: Nick Shipping
 Driver: Nicholas
 Under 'FEES', the following information is listed:
 Shipping Rates: £474.24
 Total Fee: £957.38

Figure 96. Invoice of an order

Cargo owners can update profile details through the edit details button in the main page.



The screenshot shows a 'Dialog' window titled 'Cargo Owner Details'. It has two main sections: 'Edit Details' and 'Current Details'.
 Section 1 (Edit Details):
 - A red error message: 'Username, Email and Phone number is UNIQUE!'
 - Input fields: Name (Letters and Spaces only), Username (At least 5 characters), Password (1 Caps Letter, 1 Number), Email, Address, Phone.
 - An 'Edit' button.
 Section 2 (Current Details):
 - A table showing current details:

Email	Address	Name
1 rickharith@...	20 Linden ...	Rick

 A 'Back' button and a circled '4' are also visible.

Figure 97. Edit Cargo Owner Profile

1. This form allows cargo owners to edit profile details. Username, email and phone number is unique meaning that there can be no duplicate e-mails and such.
2. This table displays the cargo owner's details, it will show the up to date details after an edit.
3. Edit button to update the cargo owner details.
4. Leads back to cargo owner main page.

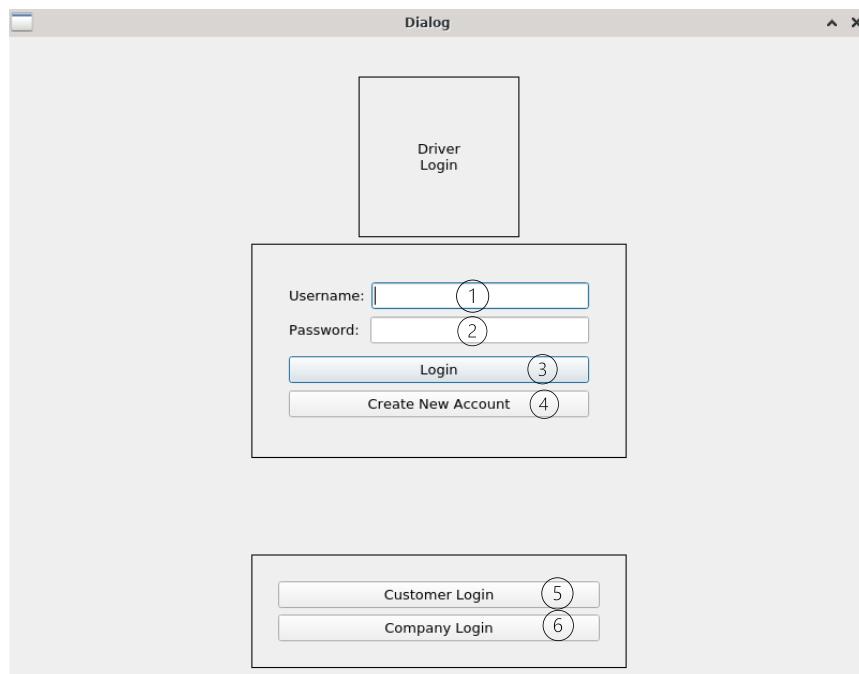
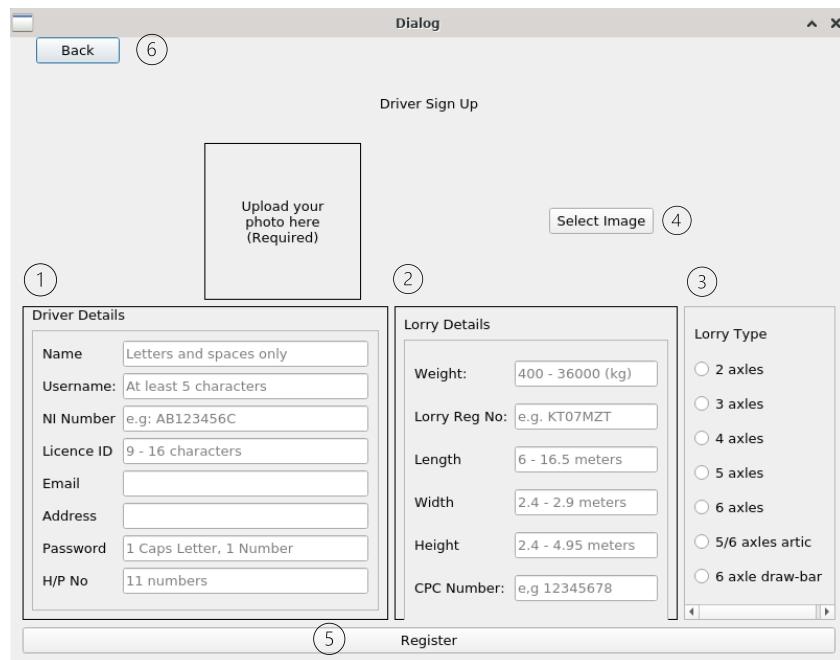
DRIVER:

Figure 98. Driver Login Page

The driver should navigate to the driver login page.

1. Username field for logging into the driver's page.
2. Password field for logging into the driver's page, ensure the username and password match your sign-up information.
3. Login button to log driver user in.
4. Create a new account if no account exists.
5. Customer login for cargo owners.
6. Company login for transportation companies.

Drivers should create a new account if no account exists for login, navigate to the create a new account page.



The screenshot shows a 'Dialog' window titled 'Driver Sign Up'. On the left, there's a placeholder box labeled 'Upload your photo here (Required)' with a 'Select Image' button next to it. In the center, there are two main sections: 'Driver Details' and 'Lorry Details'. The 'Driver Details' section contains fields for Name, Username, NI Number, Licence ID, Email, Address, Password, and H/P No. The 'Lorry Details' section contains fields for Weight, Lorry Reg No., Length, Width, Height, and CPC Number. To the right of these details is a 'Lorry Type' section with radio buttons for axle counts: 2, 3, 4, 5, 6, 5/6 artic, and 6 axle draw-bar. At the bottom, there's a 'Register' button. The numbered callouts point to specific elements: 1 points to the 'Driver Details' section; 2 points to the 'Lorry Details' section; 3 points to the 'Lorry Type' section; 4 points to the 'Select Image' button; 5 points to the 'Register' button; and 6 points to the top-left corner of the window.

Figure 99. Driver Sign Up Page

1. Driver details is where the general driver information must be typed. Fields such as NI Number have a specific format to follow or the sign up will fail.
2. Lorry details is where lorry information should be typed. The dimensions follow strict values to ensure users do not input unrealistic values. Lorry registration number follows a format and sign up will fail if this format is not met.
3. Drivers must pick a lorry type, picking multiple options is not possible.
4. Picks a driver photo to be uploaded.
5. Click the register button once all fields are inputted correctly.
6. Back button to return to the login pages.

Once the driver completes the sign up, return to the login page and enter the correct credentials to log in.

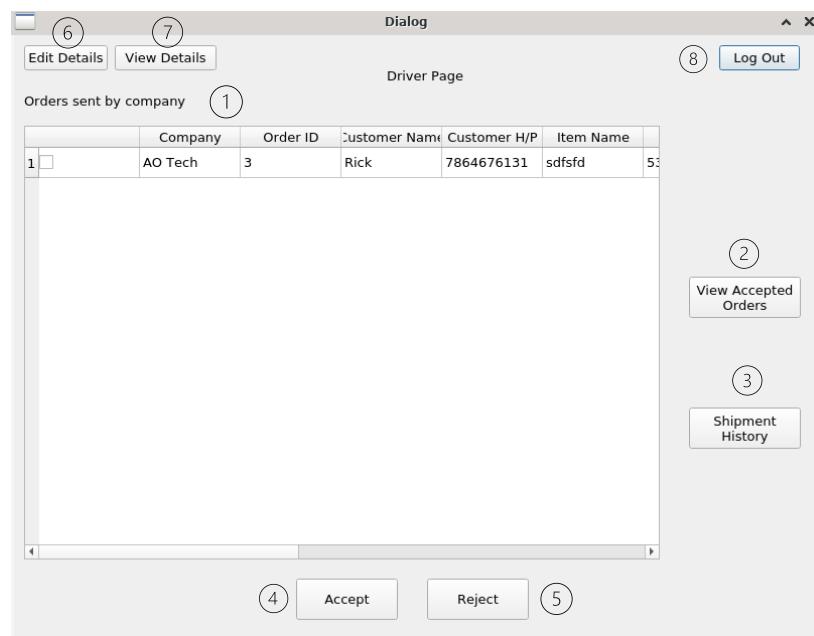


Figure 100. Drivers Main Page

1. The table shows all the orders that a company assigns to the driver, the checkbox in the first column must be ticked to select a specific order.
2. Accepted orders will appear in the View Accepted Orders page.
3. Shipment history is where delivered orders will appear, this will be the page where completed and delivered orders are shown.
4. The order must be checked in the checkbox to select an order, the driver can choose to accept, accepted orders will have order status changed to 'Loading'.
5. This button is for rejecting order, orders that are rejected will be sent back to the transportation company to assign to a different driver.
6. Leads to the edit profile page for driver.
7. Current profile details will appear in this page.
8. Logs the driver out and returns to the main login page.

Once a driver accepts an order, navigate to the View Accepted Orders page.

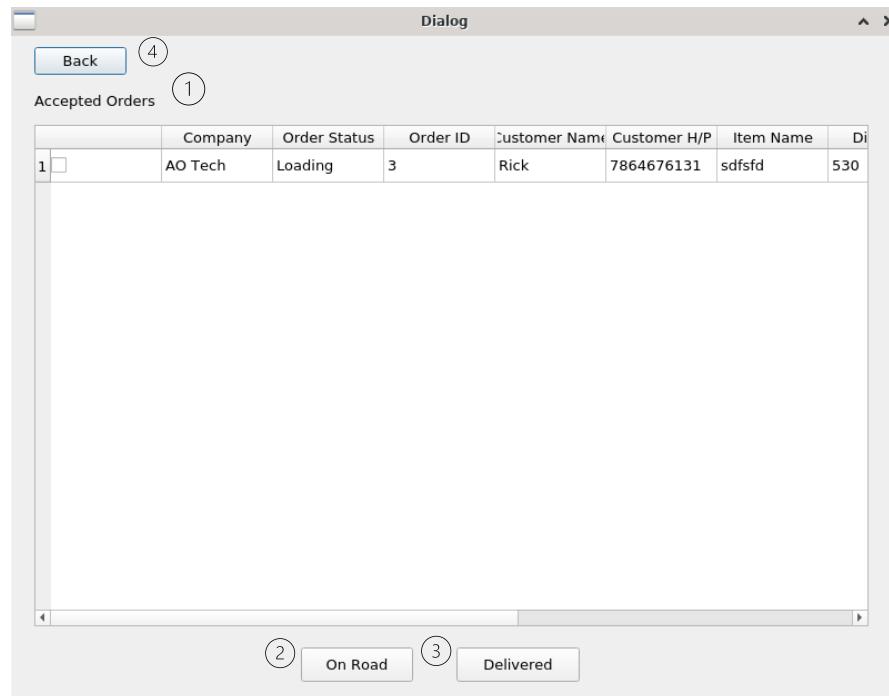


Figure 101. Accepted Orders Page

1. Accepted orders table displays all orders the driver accepts, the checkbox must be ticked to select an order, it is located at the first column of the table.
2. The 'On Road' button changes the order status to on road to indicate to the cargo owner that the order is on the way.
3. The 'Delivered' button completes the order and indicates that the order has been delivered.
4. Back button leads back to the driver main page.

Drivers can now navigate to the Shipment History page to view all completed orders.

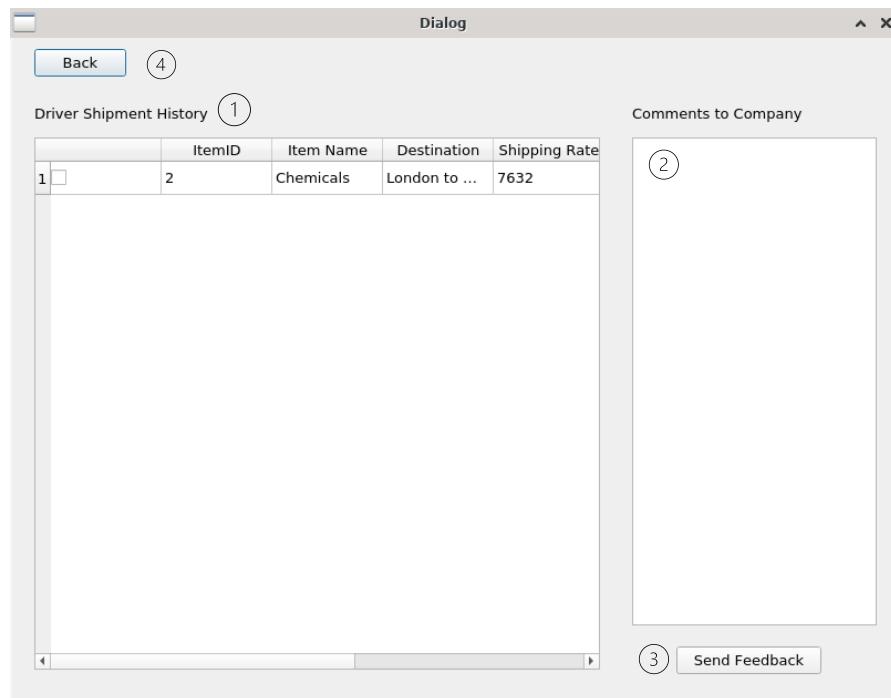
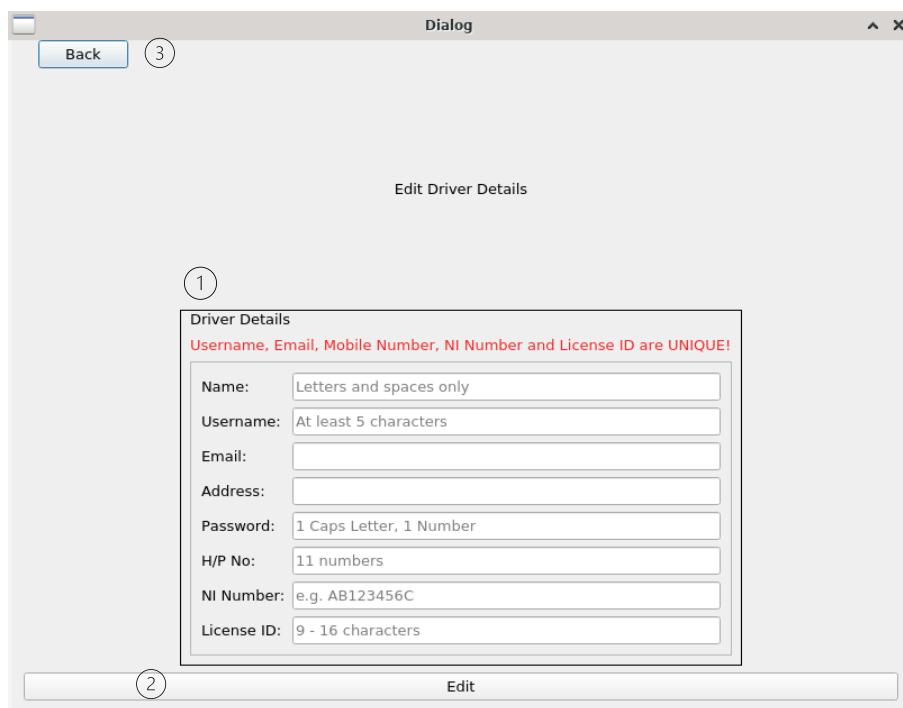


Figure 102. Shipment History Page

1. This is the driver shipment history table, it displays all completed and delivered orders that the driver has made. Checkbox must be ticked to select a specific order.
2. This field is used to send feedback to the transportation company regarding the order. Select an order to send feedback on by ticking the checkbox in the shipment history table.
3. Press the send feedback button to give feedback to the company that assigned said order.
4. Back button leads back to the driver main page.



Driver Details
Username, Email, Mobile Number, NI Number and License ID are UNIQUE!

Name:	Letters and spaces only
Username:	At least 5 characters
Email:	
Address:	
Password:	1 Caps Letter, 1 Number
H/P No:	11 numbers
NI Number:	e.g. AB123456C
License ID:	9 - 16 characters

(2) Edit

Figure 103. Edit Driver Details Page

1. Driver edit form for changing the driver details, certain fields are unique, meaning that duplicate information cannot exist, to prevent multiple accounts with the same emails and such.
2. The edit button will update driver details in the profile, if the information is invalid in format, details will not be updated.
3. Back button will lead back to the driver main page.

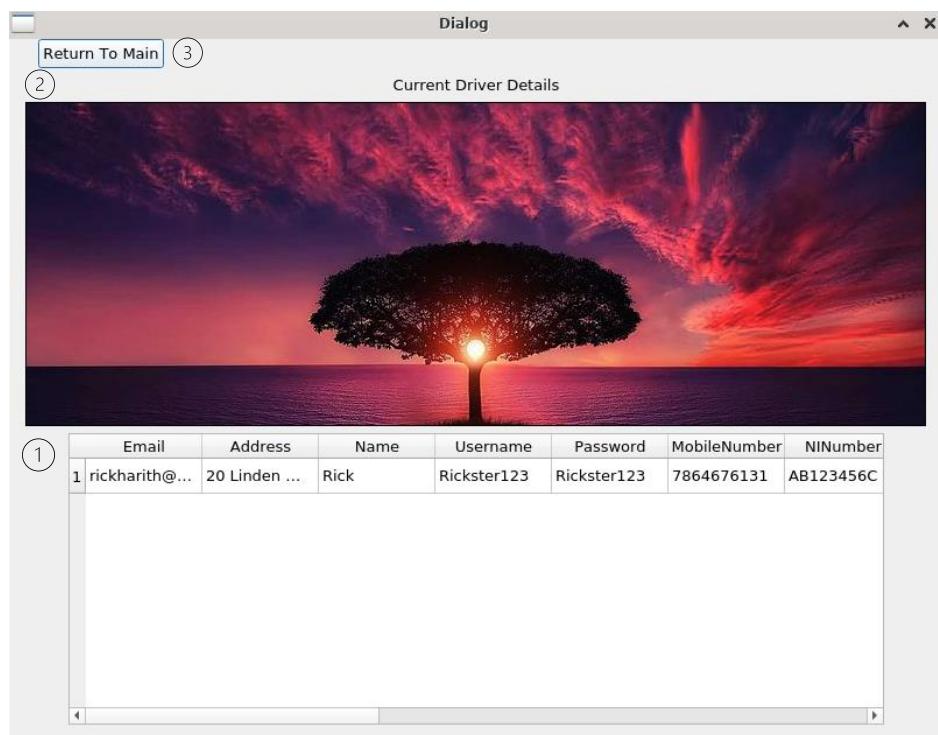


Figure 104. Edit Driver Details

1. This table displays the driver details, any changes made in edit profile will be updated when viewed.
2. This is where the driver photo is displayed.
3. Leads back to the drivers main page.

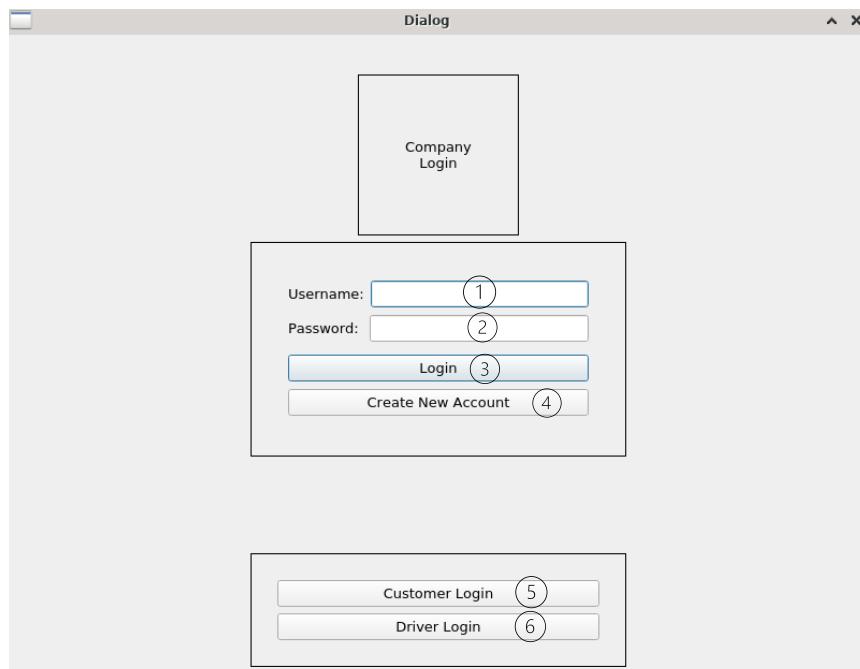
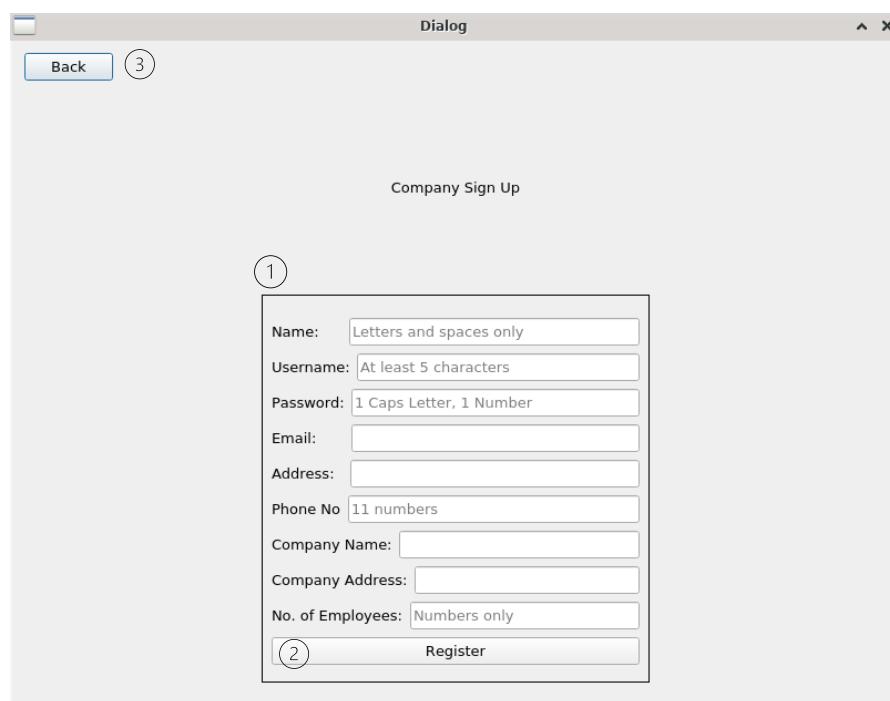
TRANSPORTATION COMPANY:

Figure 105. Transportation Company Login

Transportation companies should navigate to the company login page.

1. Username field for logging into transportation company page.
2. Password field for logging into transportation company page.
3. Click the login button to log in.
4. If no transportation company account exists, create a new account.
5. Leads to customer login page.
6. Leads to driver login page.



Company Sign Up

1

Name:	Letters and spaces only
Username:	At least 5 characters
Password:	1 Caps Letter, 1 Number
Email:	
Address:	
Phone No	11 numbers
Company Name:	
Company Address:	
No. of Employees:	Numbers only
2 Register	

3

Figure 106. Company Sign Up Page

1. The form is for transportation company sign up, all fields must be typed and must conform to the format of each fields.
2. Click the register button to sign up if all fields are typed and valid.
3. Back button leads back to login page.

Once the sign up is complete, navigate back to the login page to sign in.

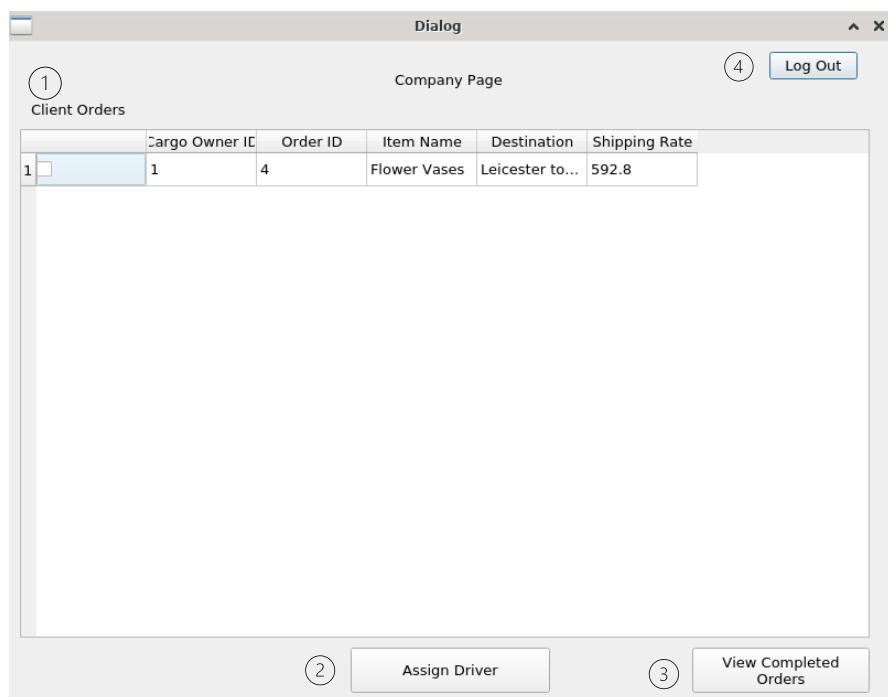


Figure 107. Transportation Company Main Page

1. This table displays all orders made by clients (Cargo Owners), the checkbox must be ticked to select a specific order. Note that any rejected will re-appear in the table.
2. The assign driver button will let the selected order have a driver assigned to, a new window will appear.
3. Leads to the View Completed Orders page where delivered orders will appear.
4. Logs out button to log out of the currently signed in account.

Once an order is selected and the assign driver button is clicked, a new window will appear for driver assigning.

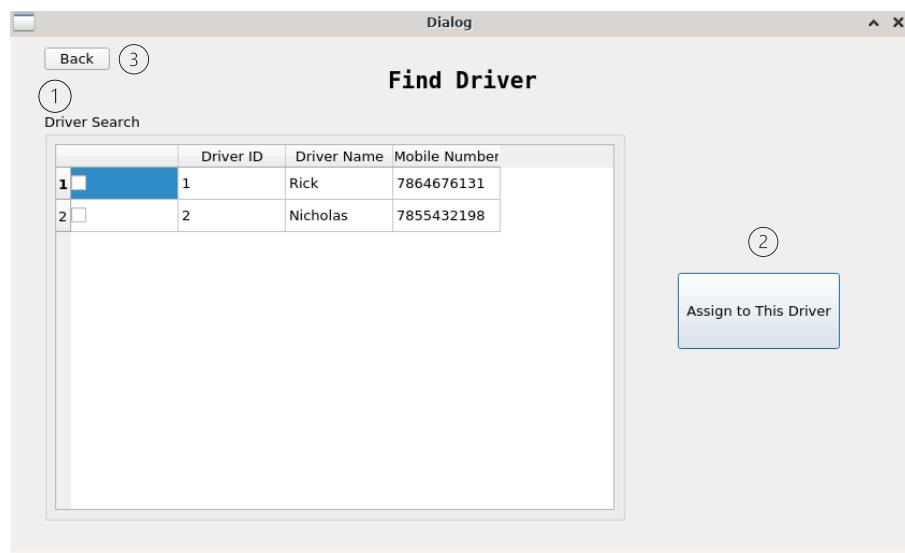


Figure 108. Assign a Driver Page

1. This is the driver search table where all drivers that are signed up appear, companies can assign any driver to the selected order from the window previously.
2. The checkbox in the driver search table must be ticked to select a driver, once a driver is selected, the button can be clicked to assign a driver to the order.
3. Leads back to the company main page.

Once a driver completes the delivery, the order will be added to the View Completed Orders page, it functions as an order history.

Dialog

Back (7)

Company Order History (1)

	Cargo Owner ID	Order ID	Item Name	Destin
1 ✓	2	2	Chemicals	London

£1534.55

Cargo Owner Feedback (3)
Cargo arrived, safely, thank you

Driver Feedback (4)
Good delivery, cargo was sent safely

Calculate Company Commission (2) (5)

View Cargo Owner Feedback (6)

View Driver Feedback

Figure 109. Company Order History Page

1. This is the company order history table, this is where the completed orders that have the cargo delivered are displayed. The checkbox in the first column must be ticked to select a specific order.
2. This calculates the company commission from the order, the amount will appear at the field above.
3. Cargo owner feedback for the specific selected order.
4. Driver feedback for the specific selected order.
5. Button to display the feedback for cargo owner in the field.
6. Button to display the feedback for driver in the field.

Conclusion and Future Work

Conclusion:

In conclusion, the e-transport marketplace (e-TM) is a comprehensive platform that provides various functionalities to help cargo owners, drivers, and transportation companies manage their cargo transportation needs efficiently. The platform offers essential features such as secure account management, order placement, order tracking, and invoice management. Additionally, the system ensures the security and privacy of customer details by encrypting them when stored in the database. The e-TM platform is a valuable tool that can enhance the efficiency of cargo transportation and provide an optimal experience for all stakeholders.

Future Work:

The e-transport marketplace (e-TM) has the potential for further enhancements and improvements to provide an even better experience for users. Here are some suggestions for future work:

1. Integration with other transportation modes: The e-TM platform can be expanded to include other transportation modes such as air or sea transport, enabling cargo owners to choose the best mode of transportation for their needs.
2. Improved communication: The platform can be improved to include more efficient communication channels between cargo owners, drivers, and transportation companies, such as chatbots or live chat.
3. Advanced tracking features: The platform can be enhanced with advanced tracking features that provide real-time tracking and updates on cargo location and status.
4. Integration with IoT devices: The platform can be integrated with IoT devices, such as sensors or GPS trackers, to provide more accurate and real-time data on cargo location, temperature, and other conditions.

5. Expansion to international markets: The e-TM platform can be expanded to international markets, enabling cargo owners to transport goods across borders and providing a more extensive network of transportation companies and drivers.
6. Concurrent programming should be implemented so multiple instances of the program can be run.
7. Some of the requirements that were not met the notification system, the encryption of cargo order, automatic driver rejection that cycles to the next driver if an order is rejected, driver availability and assigning a driver that is close to the source and destination. These requirements should have been implemented.

In conclusion, the e-transport marketplace (e-TM) has significant potential for further enhancements and improvements, and future work can focus on expanding its capabilities to provide an even better experience for users.

Appendix

The source code is maintained in the Git repository of this link:

<https://olympuss.ntu.ac.uk/N1051811/SDIuserInterface>

This application is documented using Doxygen and is available at:

<https://olympuss.ntu.ac.uk/N1051811/SDIuserInterface/tree/master/LoginPageUI>

References

- algoLancer (2020). *Sqlite C/C++ for Beginners*. YouTube. Retrieved from: https://www.youtube.com/watch?v=-C5HSdPU3TI&t=869s&ab_channel=algoLancer [Accessed 3 April 2023].
- Carlos va libre (2016). *Tutorial SQLite3 Ubuntu 14.04 Eclipse C++*. YouTube. Retrieved from: https://www.youtube.com/watch?v=Ezwug_wCx1o&t=83s&ab_channel=Carlosvalibre [Accessed 1 April 2023].

- computingmrh (2018). *Linking buttons to functions in QT5 using generated UI code*. YouTube. Retrieved from: https://www.youtube.com/watch?v=OGVuD0-5INY&ab_channel=computingmrh [Accessed 6 April 2023].
- Hello C & CPP By ProgrammingKnowledge (2022). *QT C++ GUI Tutorial - How to connect Qt to SQLite*. YouTube. Retrieved from: https://www.youtube.com/watch?v=N-on2AdmF6I&ab_channel=HelloC%26CPPByProgrammingKnowledge [Accessed 5 April 2023].
- linuxhint (2021). *Callback Function in C++*. [online] Linuxhint.com. Retrieved from: <https://linuxhint.com/callback-function-in-cpp/> [Accessed 12 April 2023].
- roseindiatutorials (2015). *How to import or open the existing project in Eclipse?* YouTube. Retrieved from: https://www.youtube.com/watch?v=o-t2e0hxNHc&ab_channel=roseindiatutorials [Accessed 10 April 2023].
- Stack Overflow (2011). *How can I add a checkbox/radio button to QTableWidget*. [online] Stack Overflow. Retrieved from: <https://stackoverflow.com/questions/5369300/how-can-i-add-a-checkbox-radio-button-to-qtablewidget> [Accessed 9 April 2023].
- Stack Overflow (2020). *Searching In QTableWidget*. [online] Stack Overflow. Retrieved from: <https://stackoverflow.com/questions/65235164/searching-in-qtablewidget> [Accessed 9 April 2023].
- Stack Overflow (2012). *Inserting data from sqlite database to QTableWidget using PyQt in Python*. [online] Stack Overflow. Retrieved from: <https://stackoverflow.com/questions/11754825/inserting-data-from-sqlite-database-to-qtablewidget-using-pyqt-in-python> [Accessed 9 April 2023].
- Stack Overflow (2021). *QTableWidget set header labels*. [online] Stack Overflow. Retrieved from: <https://stackoverflow.com/questions/66293142/qtablewidget-set-header-labels> [Accessed 9 April 2023].
- Stack Overflow (2016). *Connect button to a function QT c++*. [online] Stack Overflow. Retrieved from: <https://stackoverflow.com/questions/37848140/connect-button-to-a-function-qt-c> [Accessed 5 April 2023].

- Sqlite.org. (n.d.). *An Introduction To The SQLite C/C++ Interface*. [online] Retrieved from: <https://www.sqlite.org/cintro.html> [Accessed 4 April 2023].
- Qt (n.d.). *Qt Creator Manual*. [online] Qt.io. Retrieved from: <https://doc.qt.io/qtcreator/> [Accessed 7 April 2023].
- Qt (n.d.). *A Quick Start to Qt Designer | Qt Designer Manual*. [online] Qt.io. Retrieve from: <https://doc.qt.io/qt-6/designer-quick-start.html> [Accessed 7 April 2023].