1.1 ABOUT THE PROJECT

The project entitled “Online Cargo Service” is aimed to serve the interest of customers and to cover a widely distributed system. The goal of the system is to computerize the transportation process of a cargo agency.

The entire system has the following parts:

- Branch Details.
- Cargo Details.
- Sending Details.
- Monitoring Sending Details on the Web.

From customer’s level it is a system, which enables them to send cargo to various locations and to track the progress of the transmission using a web site. From companies level it is the system, which can be used to keep all the business information together and can do the business in a more flexible and efficient manner. It is a system, which enables the agency to keep track of all the details regarding the transmission of goods.

The aim of the system is to computerize the cargo processing. All the details regarding the branches, cargo details and sending details are stored in a remote server. All the branches can use these databases. When a user has to send a cargo, the details of the cargo sending is added in cargo details. It includes customer details, item details, quantity etc. also details of sending are added to the sending detail database. When the cargo reaches the neighbouring branch, it is forwarded to the next neighbour branch until it reaches the destination point. There is a website which can track the sending details using an id. The cargo system is a real vast area with too many fields of operation. All these operations are aimed at improving the company resources by attracting customers. The system should be very user-friendly.

2.1 SYSTEM STUDY

2.1.1 Existing system

In existing system, to manage the cargo service we use a manual system. In the present system all the activities are performed manually. All data entry is performed by writing data into the bookings. The record is prepared manually, so there is a chance for occurring errors and the calculations are not so accurate. Various information such as staff details, customer details, package details, payment details are handled manually. When there is need for retrieving details searching is unavoidable, this is a difficult task searching the records manually.

The existing system uses manual method for whole process. This requires a lot of hard work and time consumption to complete the task. This may include human errors. In this existing system, it is difficult to retrieve some particular information. Also, all the records are stored manually and it is tedious task. As a result, the security of these records is always a challenging task. Hence, the computerization of the system of record maintenance is the only solution to reduce the shortcomings of existing system.

2.1.2 Proposed System

The proposed system is interactive, highly user friendly and designed exclusively for the Online Cargo Service. The Cargo Service management system is a database system used to store the information regarding staff details, customer details, package details, payment details, etc.

All the operations and activities related to The Online Cargo Service can be carried out efficiently. The system manages a well-organized database for storing the resources that they are providing by the Online cargo service. This help us to eliminate the entering of invalid data. Most problems of manual system can be solved by this system.

2.1.3 Feasibility Study

Feasibility study is made to see if the project on completion will serve the purpose of the organization for the amount of work, time and effort spent on it.

Study lets the developers to see the future of the project and its usefulness. Finding out whether a new system is required or not.

The study is carried out to the best system that meet performance requirement.

This entails identification, description and evaluation of candidate system and selection of the best system for the job. It simply identifies whether the proposed system is feasible to the organization or not.

There are three aspects in the feasibility study portion of the preliminary investigation

i) Technical feasibility
ii) Economic feasibility
iii) Operational feasibility

2.1.3.1 Technical Feasibility

The Online Cargo Service system must be evaluated from technical view point first. The assessment of this feasibility must be based on an outline design of the system requirement in the terms of input, output, programs and procedure having identified an outline system, the investigation must go on to suggest the type of equipment, required method of developing the system, method of running the system once it has been designed. The project should be developed such that the necessary functions and performance are achieved within the constraints. The project is developed with latest technology. There are only minimal constraints involved in this project.

2.1.3.2 Economic Feasibility

Here an evaluation of development cost is weighted against the ultimate income or benefit derived from the developed system. The cost for the development of the project has been evaluated and we want to check that the cost does not exceed beneficial cost of the system. The economic and financial analysis is used for evaluating the effectiveness of the candidate system. This project also undergone economic feasibility study and found that it is feasible. So, the cost for development does not exceed its beneficial cost. This brought to as the conclusion that the system is economically feasible in the context.

2.1.3.3 Operational Feasibility

In operational feasibility the entire application is checked whether the system will be used if it is developed and implemented. Also, it is checked whether there will be resistance from user that may undermine the possible application benefits. There is no barrier for implementing the system. The system also helps to access the information immediately as need arises. Thus, the system is found to be operational feasible.

2.2 USER CHARACTERISTICS

This software has five users

1. Administrator
2. Branch
3. Customer
4. Staff
5. Delivery boy

Administrator:

The Administrator is the primary user who has the most or maximum control over the software. Administrator administrates over the entire activities of the system and has full control over what all happens in the system. He/she is the only user who can add, view, change or deactivate details of a branch, view package details, customer details, bookings, payment details.

Branch:

Branch is the secondary user. He/she has limited privileges when compared to the administrator. They have the functions like view bookings, add and update staff/delivery boy and can their details etc.

Customer:

The customer will have little privileges when compared to both administrator and branch. They can book the package from their home itself. They can also find their status of the package and all the details related to the package and they can give ratings and feedbacks and they can also view, edit their profile and password.

Staff:

Staff has limited privileges when compared to both administrator and branch. They have the functions like viewing status of the package, package details etc. General interaction with the system is done with the help of the staff.

Delivery boy:

Delivery boy has limited privileges when compared to administrator, branch and staff. They have the functions like view booked packages, collect and transfer package, update package status etc. General interaction with the system is done with the help of the Delivery boy.

2.3 SYSTEM SPECIFICATION

2.3.1 Hardware Specification

The selection of hardware and software configuration is very important task related to system development.

Processor	Intel Pentium IV (3.0 GHz) or above
RAM	1 GB
Hard Disk	80 GB and above
Key Board	Normal or multimedia
Monitor	15’’CRT or LCD monitor
Mouse	Compatible Mouse

2.3.2 Software Specification

Operating System	Windows
Front	PYTHON
Back End	SQL Server 2008

2.3.3 ABOUT THE SOFTWARE TOOLS

FRONT END SPECIFICATION: Python

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

- Python is Interpreted − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
- Python is Interactive − You can actually sit at a Python prompt and interact with the interpreter directly to write your programs.
- Python is Object-Oriented − Python supports Object-Oriented style or technique of programming that encapsulates code within objects.
- Python is a Beginner's Language − Python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

BACK END SPECIFICATION: SQL Server 2008

SQL Server 2008 is an integrated database management system and analysis solution that delivers increased security, scalability and availability to enterprise data and analytical applications, while making them easier to build, deploy and manage. It is comprehensive software that enables to reliably manage mission – critical information and confidently run today's increasingly complex business applications. SQL Server 2008 allows gaining greater insight and achieving faster results for a competitive advantage. The key capabilities of SQL Server 2008 are the following:

High Availability: Ensures business continuity with the highest levels of system availability through technologies that protect data against costly human errors and minimize disaster recovery downtime.

Performance and Scalability: Deliver an infrastructure that has proven record in handling today's large amounts of data and critical enterprise workloads.

Security: Provides a secure environment to address privacy and compliance requirements with built in features that protect data against unauthorized access.

Manageability: Manages infrastructure with automated diagnostics, tuning and configuration to reduce operational costs while reducing maintainance and easily managing very large amounts of data.

Developer Productivity: Build and Deploy critical business ready applications more quickly by improving developer productivity and reducing project lifestyle times.

Business Intelligence: Gain deeper insight into the business with integrated comprehensive analysis and reporting for enhanced decision making.

3. SYSTEM MODELING

The most creative and challenging phase of the system development is system design. It provides the understanding and procedural details necessary for implementing the system recommended in the feasibility study. Design goes through the logical and physical stages of development.

In designing a new system, the system analyst must have a clear understanding of the objectives, which the design is aiming to fulfil. The first step is to determine how the output is to be produced and in what format. Second input data and master files have to be designed to meet the requirements of the proposed output. The operational phases are handled through program construction and testing. The point is to choose such an environment in which we will be able to operate with in a convenient and easy way. The most creative and challenging phase of the system development is system design. It provides the understanding and the procedural details necessary for implementing the system recommended in the feasibility study. The analyst should understand the requirements of the user and develop the system accordingly. Design goes through the logical and physical stages of development. In designing a new system, the system analyst must have a clear understanding of the objectives, which the design is aiming to fulfil. The application program as an interface between the users and the database should be an accurate reflection of the database on the screen; hence a well analyzed and defined structure is needed. The user interface should be easy to understand and operate on for the users. The first step is to determine how the output is to be produced and in what format it has to be produced. Second, input data along with the master files have to be designed to meet the requirements of the proposed output.

The analyst must ensure that the interaction between the user and the interface is simple to understand. To ensure that everything works properly and as it has been expected, test performances have to be done upon the system functionality. Testing plays an important role in identifying any minor errors after system design and it will be corrected.

3.1 MODULES AND DESCRIPTION

Online Two-Wheeler booking Platform is a web-based shopping system which provides us facilities to manage the activities taking place in Online vehicle booking. This system is developed to manage the work flow in an Online vehicle booking. There are 7 modules in this project. They are:

1.0 Branch Management

2.0 Customer Registration

3.0 Staff Management

4.0 Package Management

5.0 Booking Management

6.0 Delivery boy Management

7.0 Payment Management

1.0 BRANCH MANAGEMENT

This module stores the details of all the branch they have and in which all cities. Branch management module also deals with adding new staff/delivery boy to the system, updating the details of the existing staff/delivery boy, maintaining the status of the staff/delivery boy being registered to the system. Here store branch names, no of staffs, delivery boys in each branch, locations, etc.

2.0 CUSTOMER REGISTRATION

Customer Registration Module stores the details of all the customers who register in the webpage. Customers enter the system after registration and their details get stored in this module. They can edit and view their details when they need to.

3.0 STAFF MANAGEMENT

This module deals with the managing the package request and assign delivery boy to deliver a package. Here store the details of the delivery boys like name, contact details etc.

4.0 PACKAGE MANAGEMENT

The main aim of this module is to manage all type of packages in an organized manner. This module stores the details of packages like name, weight, height, width, price etc.

5.0 BOOKING MANAGEMENT

This module coordinates all the information related to package and details of branch, locations, details of the delivery boys, status of the delivery and total amount.

6.0 DELIVERY BOY MANAGEMENT

This module coordinates information related to the delivery boy. Here store the details of the delivery boys like name, contact details etc.

7.0 PAYMENT MANAGEMENT

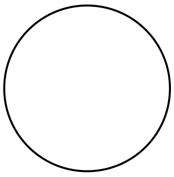
This module keeps the details of all the payments done by the customer. Here store the details of booking, amount paid and status of payment.

3.2 DATA FLOW DIAGRAM (DFD)

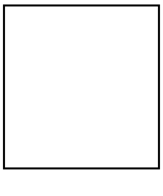
A data flow diagram is graphical tool used to describe and analyze movement of data through a system. These are central tool and the basis from which the other components are developed. The transformation of data from input to output, trough processed, may be described logically and independently of physical components associated with the system. These are known as the logical data flow movement of data between people, departments and workstations. A full description of a system actually consists of a set of data flow diagrams.

A DFD is also known as a "bubble chart" has the purpose of clarifying system requirements and identifying major transformations that will become programs in system design. So, it is the starting point of the design to the lowest level of detail. A DFD consists of a series of bubbles joined by data flows in the system.

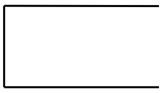
In the DFD, there are four symbols



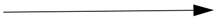
Process that transforms data flow



Source or Destination of data



Data store



Data flow

Rules for drawing data flow diagrams

Rule 1: Establish the context of the data flow diagram by identifying all of the net input and output data flows.

Rule 2: Select a starting point for drawing the DFD.

Rule 3: Give meaningful labels to all data flow lines.

Rule 4: Label all processes with action verbs that relate input and output data flows.

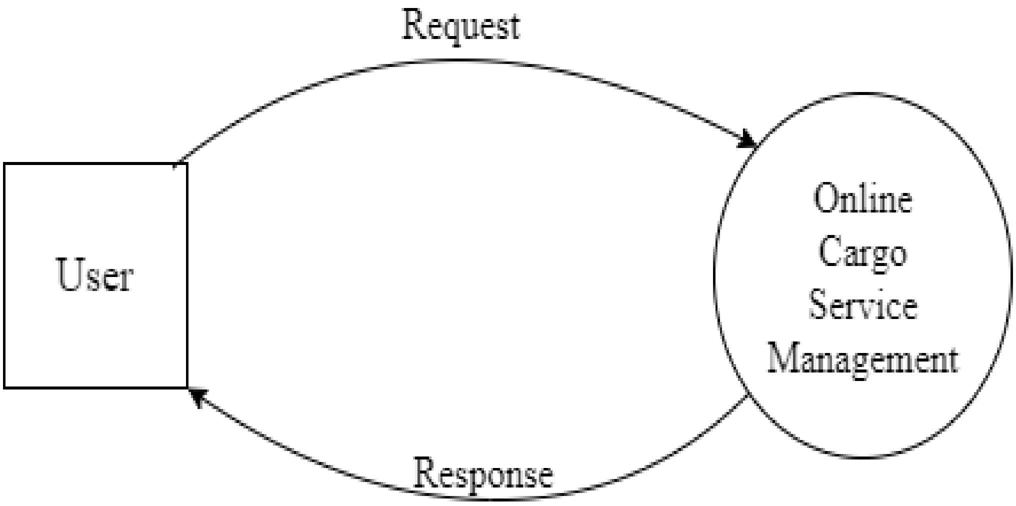
Rule 5: Omit insignificant functions routinely handled in the programming process.

Rule 6: Do not include control or flow of control information.

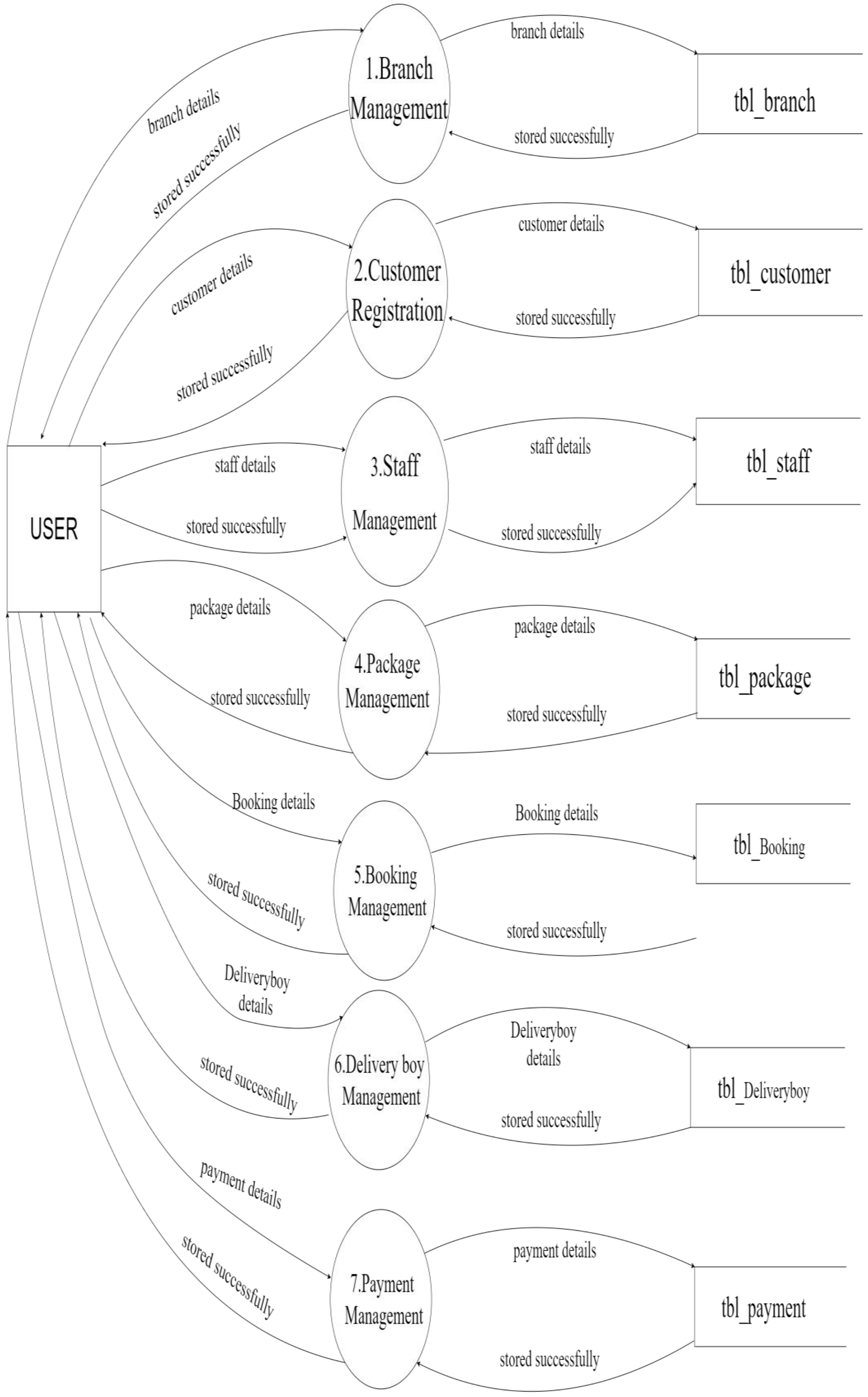
Rule 7: Do not try to put too much information in one DFD.

Rule 8: Be prepared to start over

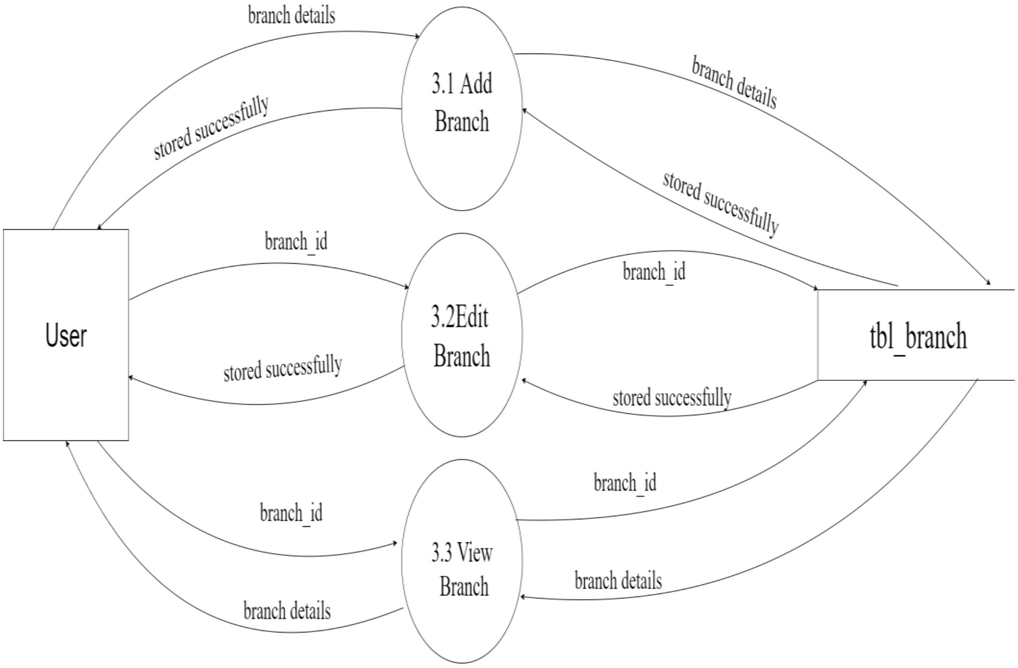
Level 0 DFD Showing Online Cargo Service Management



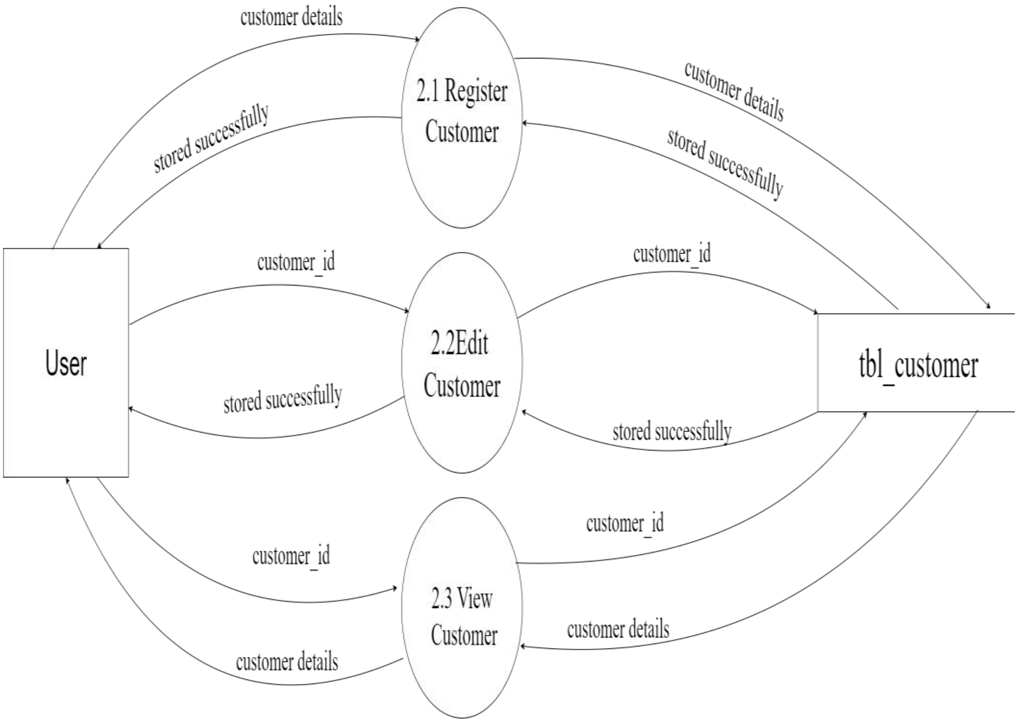
Level 1 DFD Showing Online Cargo Service Management



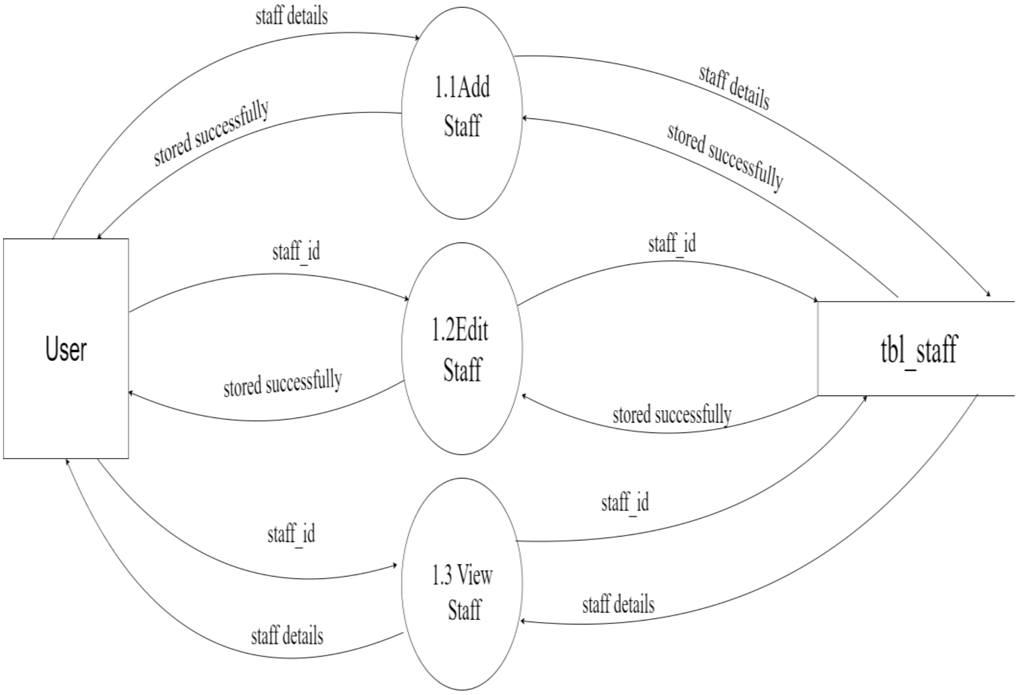
Level 2 DFD Showing Branch Management



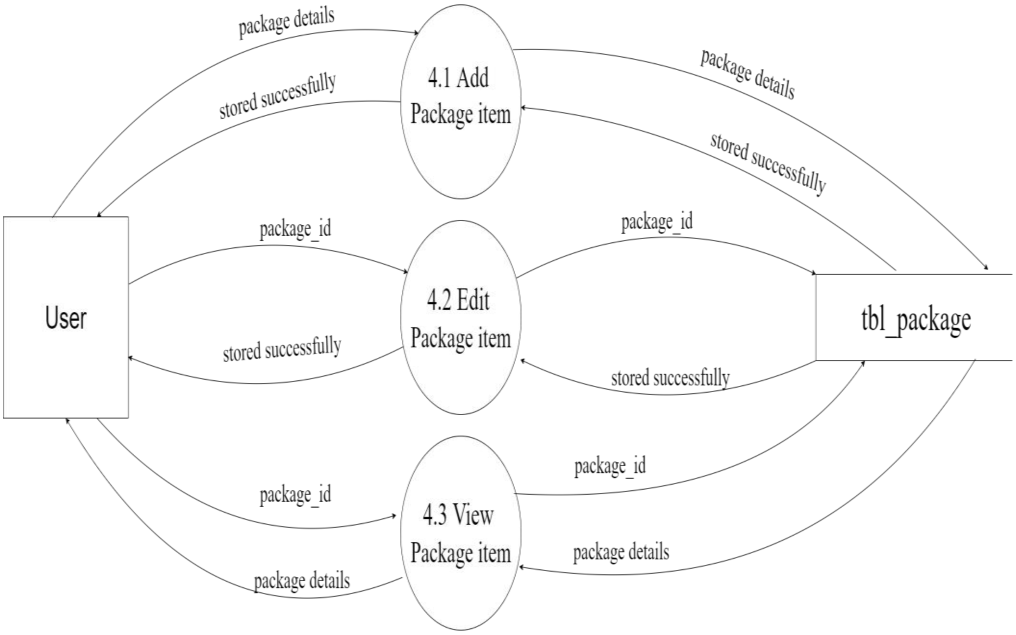
Level 2 DFD Showing Customer Registration



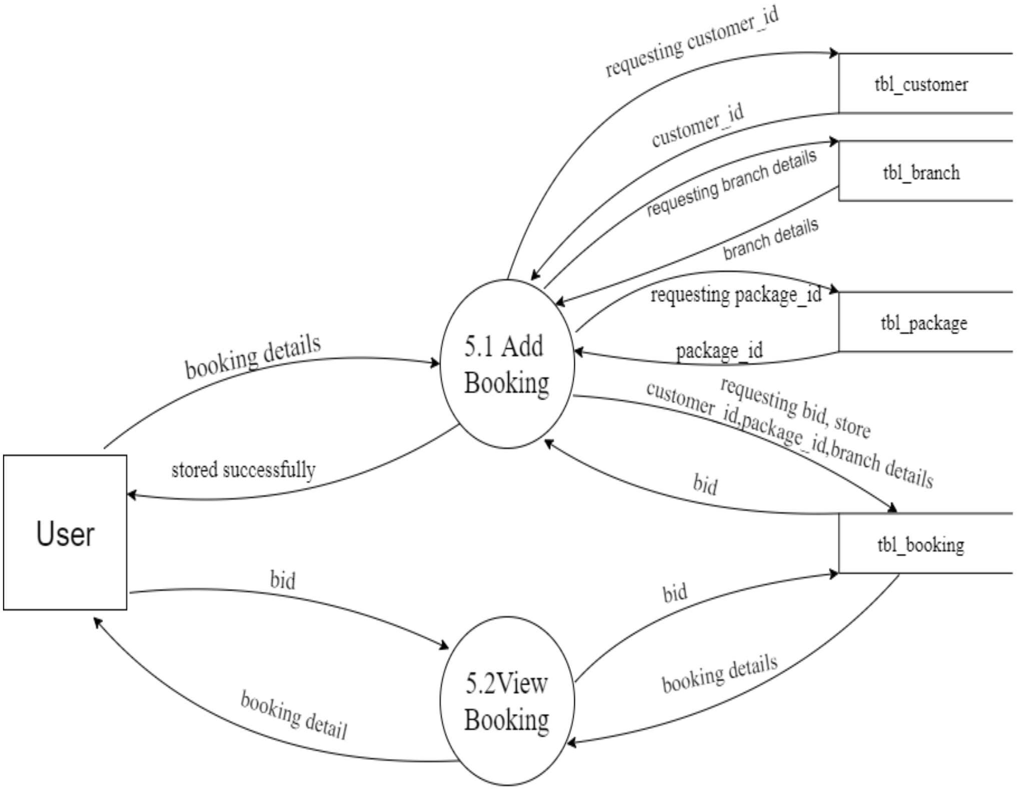
Level 2 DFD Showing Staff Management



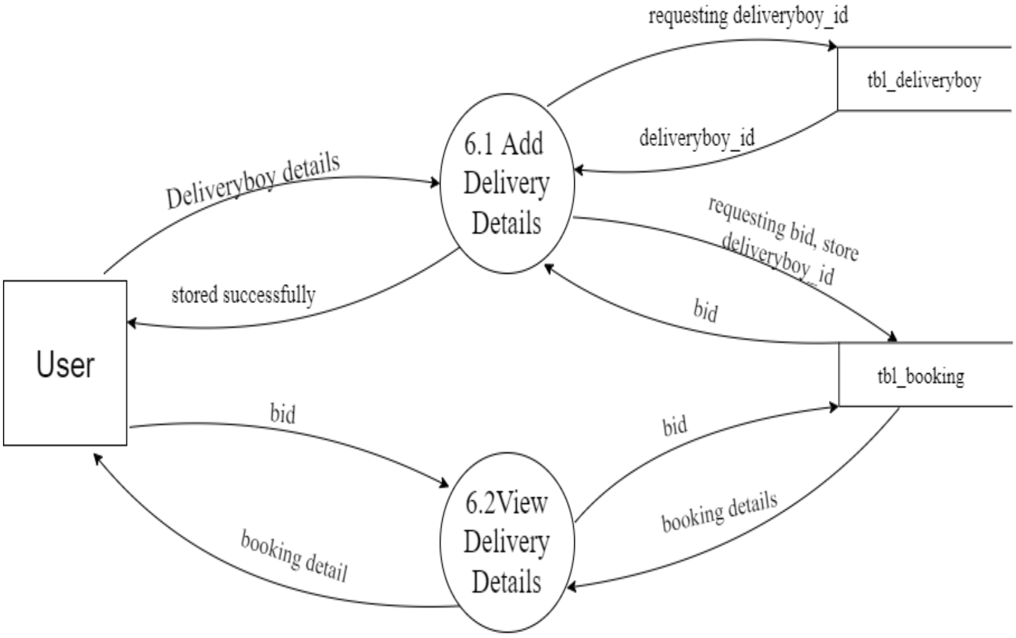
Level 2 DFD Showing Package Management



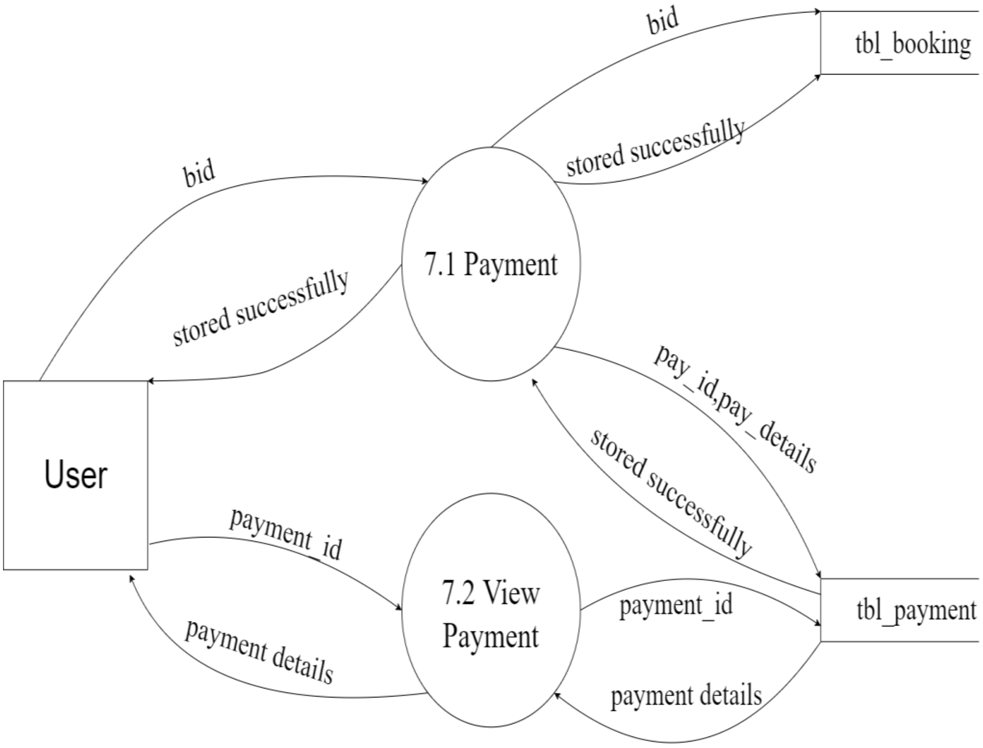
Level 2 DFD Showing Booking Management



Level 2 DFD Showing Delivery Boy Management

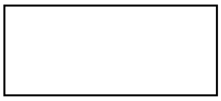


Level 2 DFD Showing Payment Management

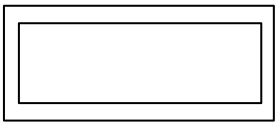


3.3. Entity Relationship Diagram

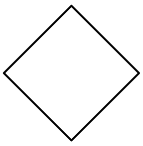
The ER model is a conceptual data model that views the real world as a construct of entities and associations or relationships between entities. A basic component of the model is the Entity-Relationship diagram, which is used to visually represent data objects. The ER modelling technique is frequently used for the conceptual design of database applications and many database applications and many database design tools employ its concepts.



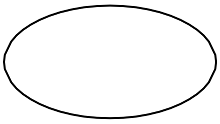
Entity Type



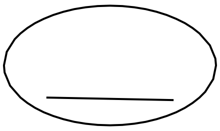
Weak Entity Type



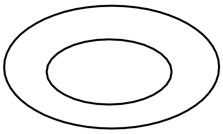
Relationship Type



Attribute

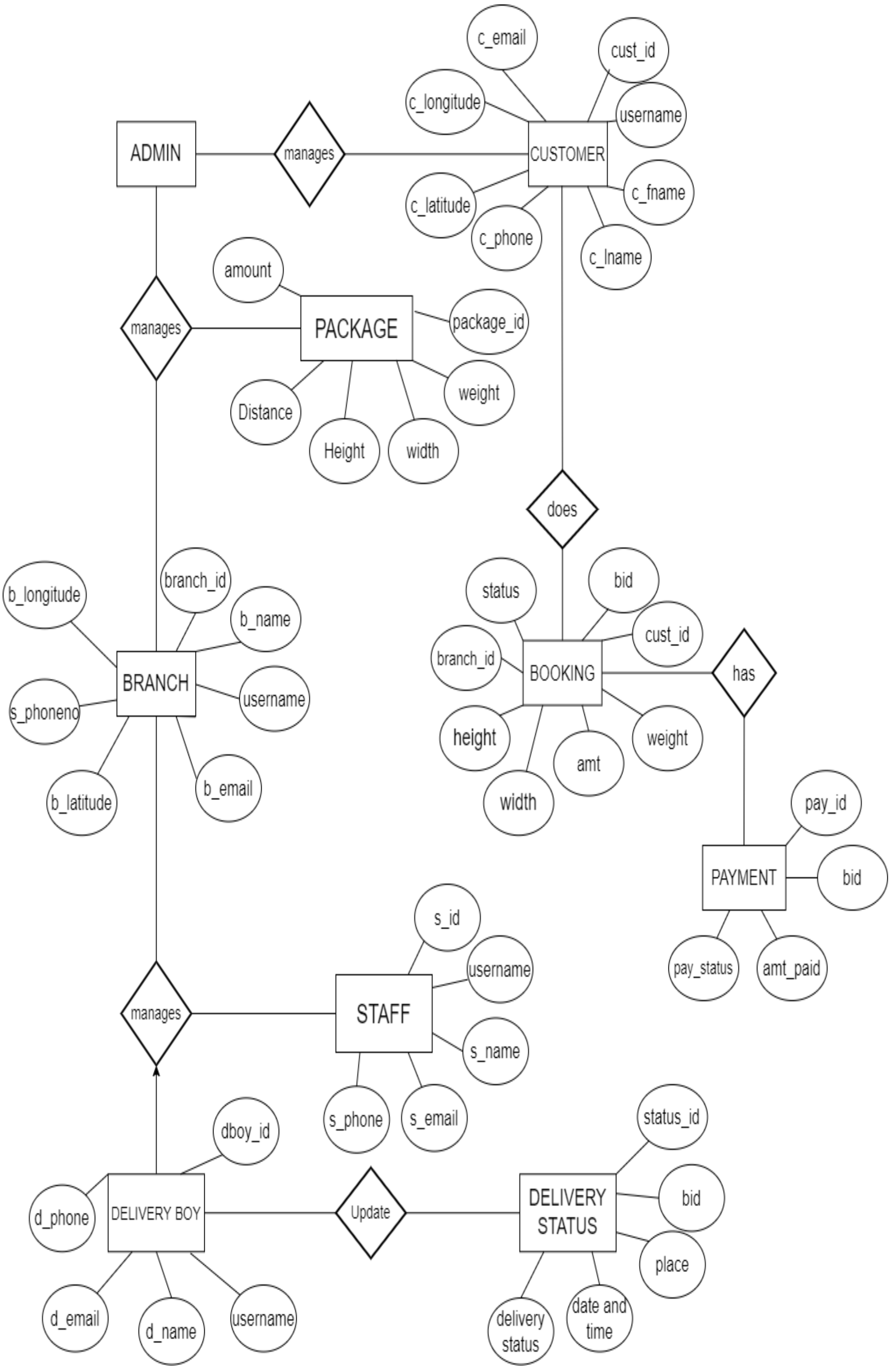


Key Attribute



Multi-valued Attribute

ER DIAGRAM


c_email
cust_id
c_longitude
username
ADMIN
manages
CUSTOMER
c_fname
c_latitude
c_phone
c_lname
amount
PACKAGE
package_id
manages
weight
Distance
Height
width
does
b_longitude
branch_id
b_name
status
bid
cust_id
branch_id
BOOKING
has
BRANCH
username
s_phoneno
height
amt
weight
b_latitude
b_email
width
pay_id
PAYMENT
bid
pay_status
amt_paid
s_id
username
manages
STAFF
s_name
s_phone
s_email
dboy_id
status_id
d_phone
DELIVERY BOY
Update
DELIVERY STATUS
bid
place
d_email
d_name
username
delivery status
date and time

4.1 INPUT DESIGN

Input design is the process of converting a user-oriented description of the inputs to a computer-based business system into a programmer-oriented specification. The quality of system input determines the quality of system output. Input specification describes the manner in which data enter the system for processing. Input design features can ensure the reliability of the system and produce result from accurate data or they can result in the production of errors.

The input design also determines whether the user can interact efficiently with the system. Input design requires consideration of the needs of the data entry operator. Three data entry considerations are:

- The field length must be documented

- The sequence of fields must match the sequence of the fields on the source document.

- The data format must be identified to the data entry operator.

In our system almost all inputs are being taken from the databases. To provide adequate inputs we have to select necessary values from the databases and arrange it to the appropriate controls.

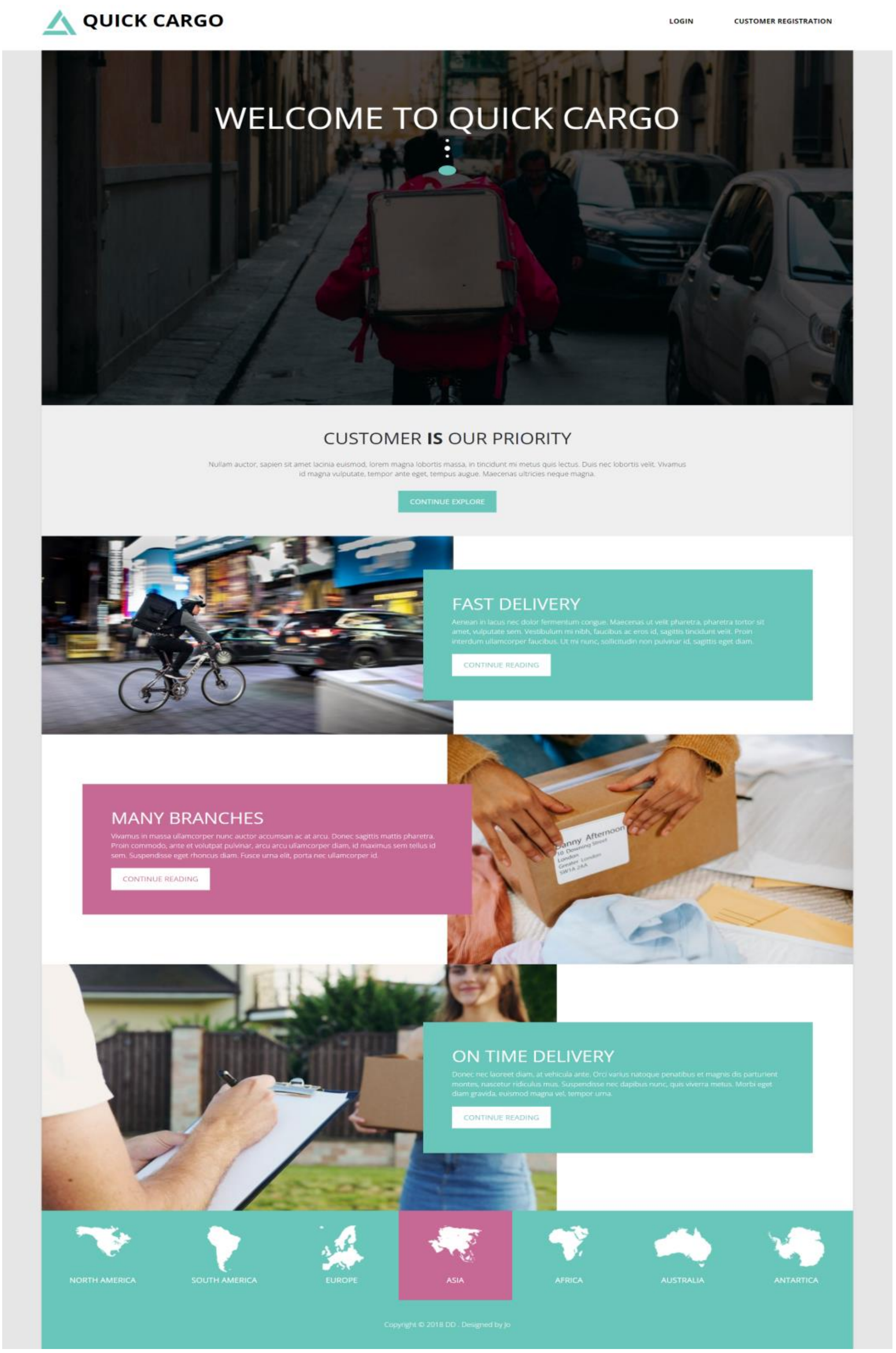
Inaccurate input data are the most common cause of errors in data processing. Errors entered by data entry can be controlled by input design. Input design is the process of converting user-oriented inputs to a computer based format. There are three major approaches for entering data into the computer.

They are menus, formatted forms and prompts. A menu is a selection list that simplifies computer data access or entry. Instead of remembering what to enter, the user choices from a list of option.

A formatted form is a pre printed form or a template that request the user to enter data in appropriate location. It is a fill-in-the-blank type form. The form is flashed on the screen as a unit. In prompt the system displays one enquiry at a time, asking the user for a response.

Home Page

Description: This form shows the Home page for all the users.



CUSTOMER **IS** OUR PRIORITY

Nullam auctor, sapien sit amet lacinia euismod, lorem magna lobortis massa, in tincidunt mi metus quis lectus. Duis nec lobortis velit. Vivamus id magna vulputate, tempor ante eget, tempus augue. Maecenas ultricies neque magna.

FAST DELIVERY

Aenean in lacus nec dolor fermentum congue. Maecenas ut velit pharetra, pharetra tortor sit amet, vulputate sem. Vestibulum mi nibh, faucibus ac eros id, sagittis tincidunt velit. Proin interdum ullamcorper faucibus. Ut mi nunc, sollicitudin non pulvinar id, sagittis eget diam.

MANY BRANCHES

Vivamus in massa ullamcorper nunc auctor accumsan ac at arcu. Donec sagittis mattis pharetra. Proin commodo, ante et volutpat pulvinar, arcu arcu ullamcorper diam, id maximus sem tellus id sem. Suspendisse eget rhoncus diam. Fusce urna elit, porta nec ullamcorper id.

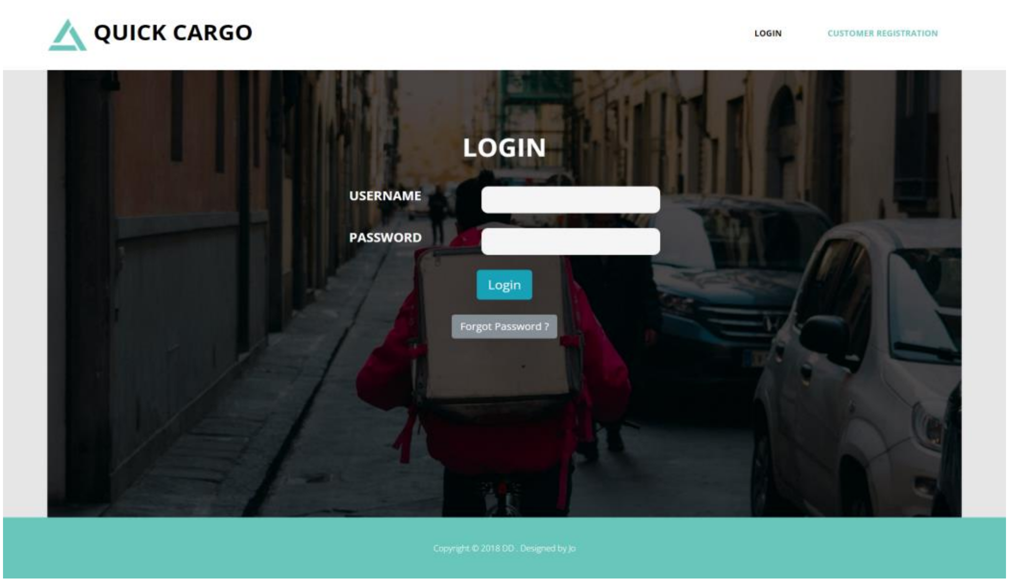
ON TIME DELIVERY

Donec nec laoreet diam, at vehicula ante. Orci varius natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Suspendisse nec dapibus nunc, quis viverra metus. Morbi eget diam gravida, euismod magna vel, tempor urna.

Copyright © 2018 DD . Designed by Jo

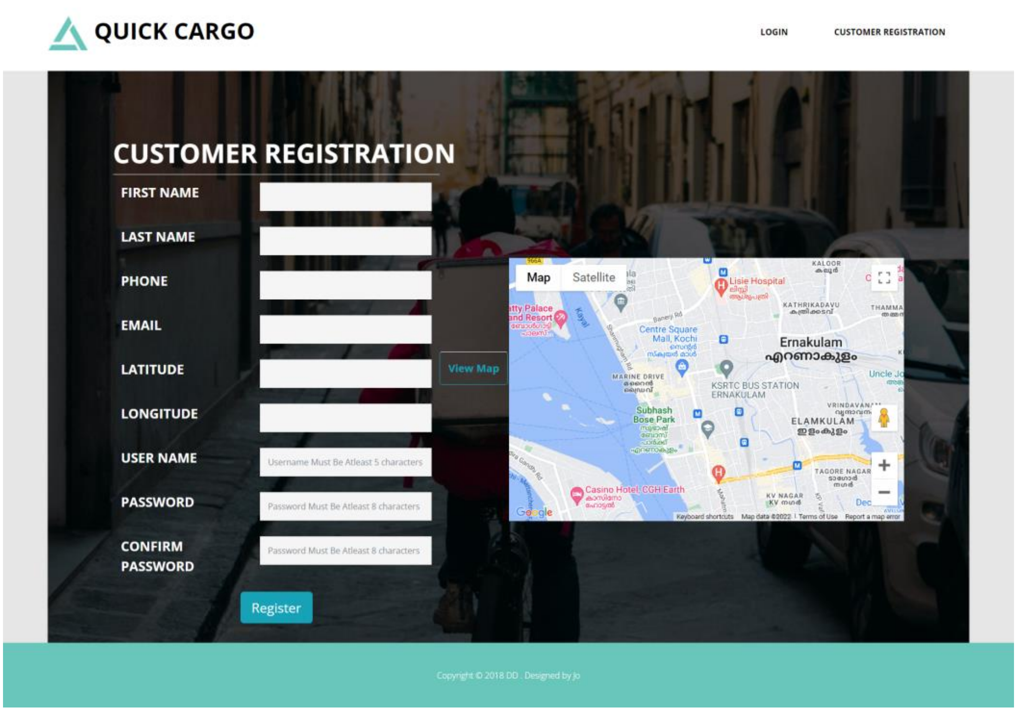
Login Page

Description: This form shows the Login page for all the users.



Sign-up Page

Description: This form shows the Signup page for customer.



Username Must Be Atleast 5 characters

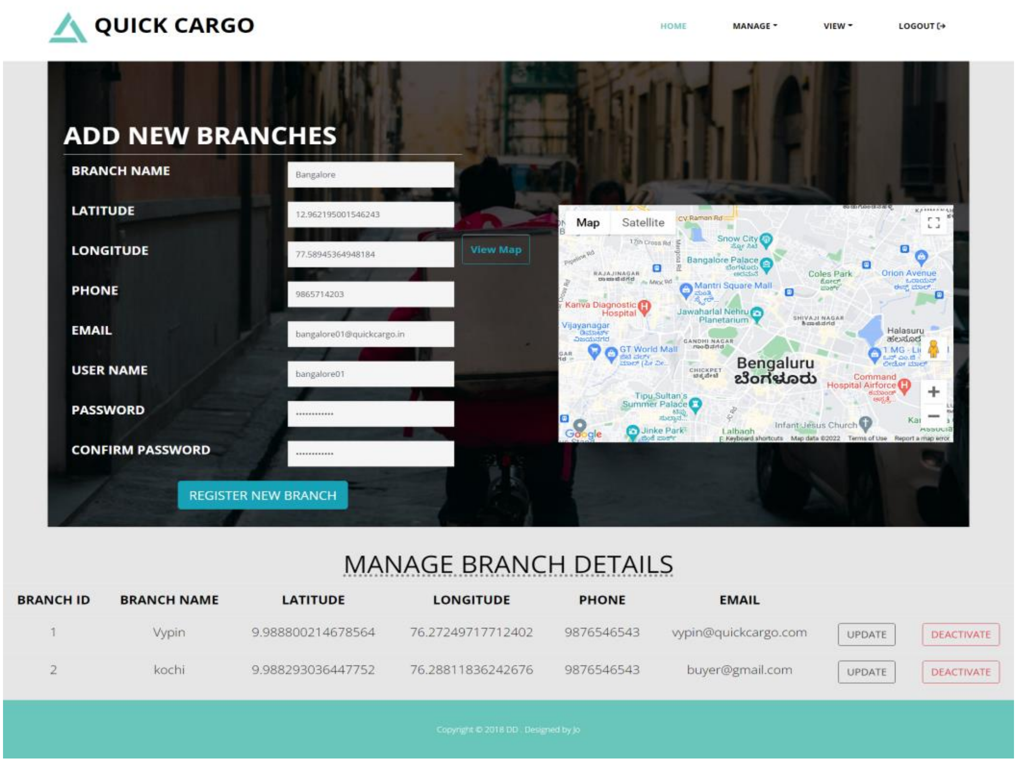
Password Must Be Atleast 8 characters

Password Must Be Atleast 8 characters

.

Add Branch Page

Description: This form shows the add new branch by admin.



ADD NEW BRANCHES

BRANCH NAME	Bangalore
LATITUDE	12.962195001546243
LONGITUDE	77.58945364948184
PHONE	9865714203
EMAIL	bangalore01@quickcargo.in
USER NAME	bangalore01
PASSWORD	············
CONFIRM PASSWORD	············

REGISTER NEW BRANCH

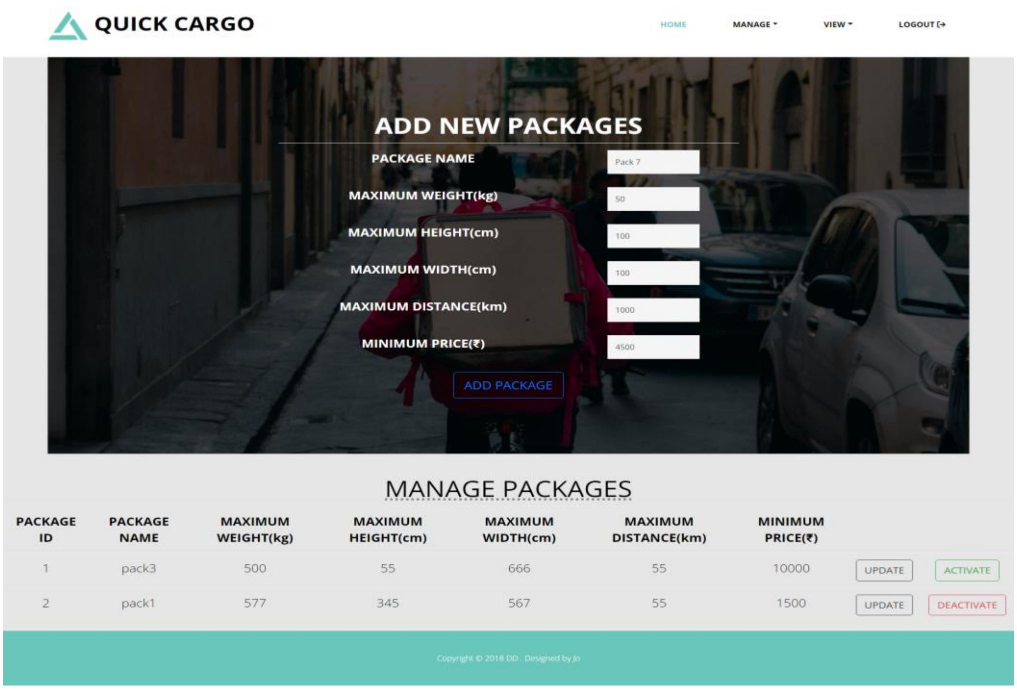
MANAGE BRANCH DETAILS

BRANCH ID	BRANCH NAME	LATITUDE	LONGITUDE	PHONE	EMAIL		
1	Vypin	9.988800214678564	76.27249717712402	9876546543	vypin@quickcargo.com	UPDATE	DEACTIVATE
2	kochi	9.988293036447752	76.28811836242676	9876546543	buyer@gmail.com	UPDATE	DEACTIVATE

Copyright © 2018 DD . Designed by Jo

Add Package Page

Description: This form shows add cargo package by admin.



ADD NEW PACKAGES

PACKAGE NAME	Pack 7
MAXIMUM WEIGHT(kg)	50
MAXIMUM HEIGHT(cm)	100
MAXIMUM WIDTH(cm)	100
MAXIMUM DISTANCE(km)	1000
MINIMUM PRICE(₹)	4500

ADD PACKAGE

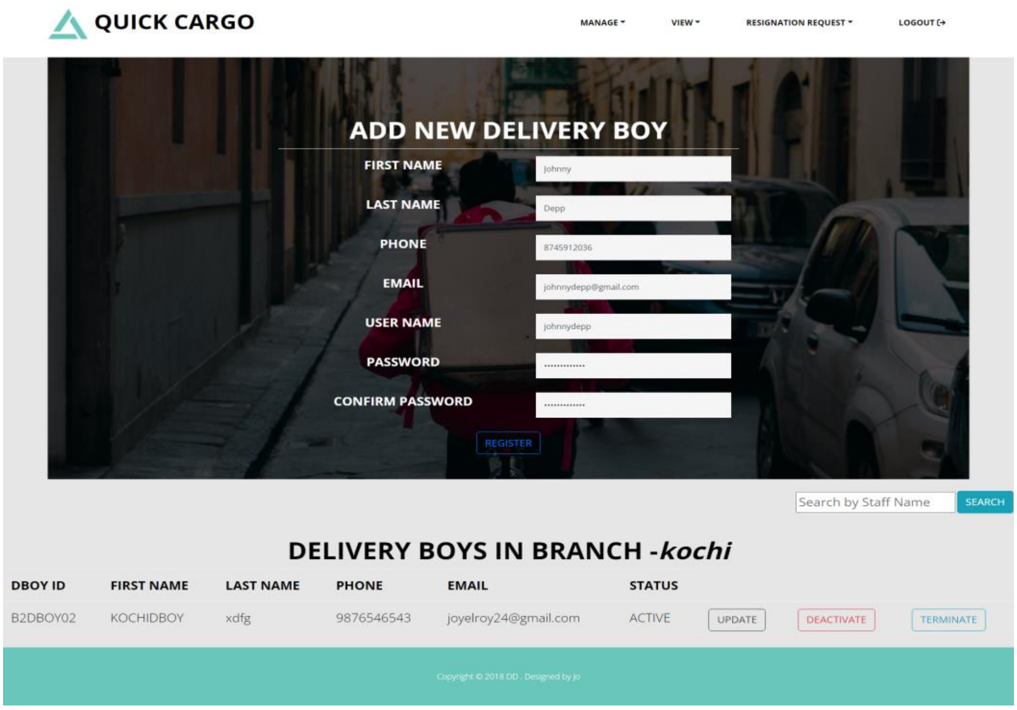
MANAGE PACKAGES

PACKAGE ID	PACKAGE NAME	MAXIMUM WEIGHT(kg)	MAXIMUM HEIGHT(cm)	MAXIMUM WIDTH(cm)	MAXIMUM DISTANCE(km)	MINIMUM PRICE(₹)		
1	pack3	500	55	666	55	10000	UPDATE	ACTIVATE
2	pack1	577	345	567	55	1500	UPDATE	DEACTIVATE

Copyright © 2018 DD . Designed by Jo

Add Delivery boy Page

Description: This form shows the add new delivery boy by branch.



ADD NEW DELIVERY BOY

FIRST NAME	Johnny
LAST NAME	Depp
PHONE	8745912036
EMAIL	johnnydepp@gmail.com
USER NAME	johnnydepp
PASSWORD	··············
CONFIRM PASSWORD	··············

Search by Staff Name

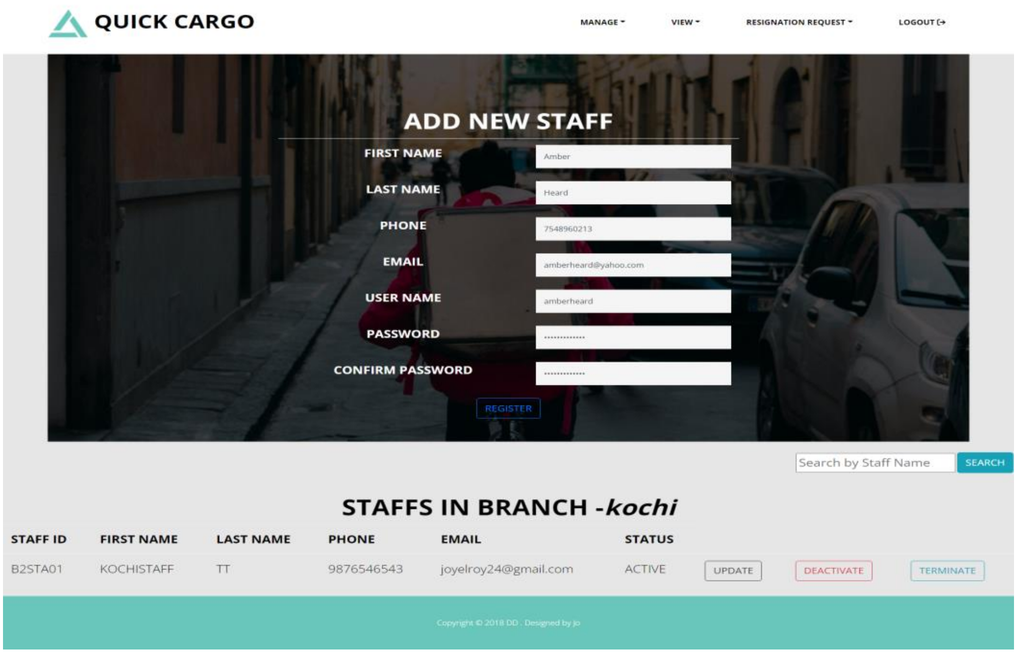
DELIVERY BOYS IN BRANCH -*kochi*

DBOY ID	FIRST NAME	LAST NAME	PHONE	EMAIL	STATUS			
B2DBOY02	KOCHIDBOY	xdfg	9876546543	joyelroy24@gmail.com	ACTIVE	UPDATE	DEACTIVATE	TERMINATE

Copyright © 2018 DD . Designed by jo

Add Staff Page

Description: This form shows the add new staff by branch.



ADD NEW STAFF

FIRST NAME	Amber
LAST NAME	Heard
PHONE	7548960213
EMAIL	amberheard@yahoo.com
USER NAME	amberheard
PASSWORD	··············
CONFIRM PASSWORD	··············

Search by Staff Name

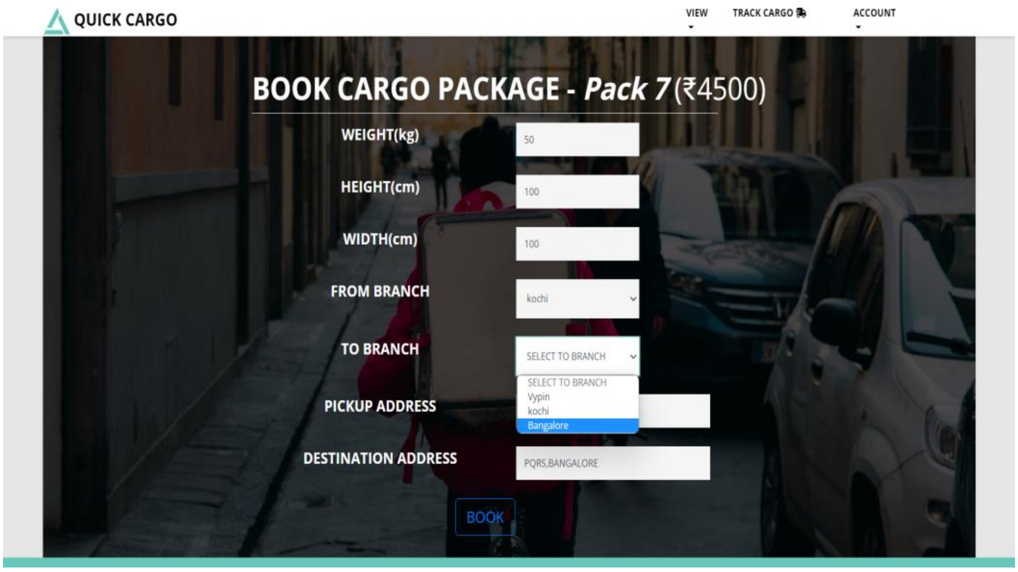
STAFFS IN BRANCH -*kochi*

STAFF ID	FIRST NAME	LAST NAME	PHONE	EMAIL	STATUS			
B2STA01	KOCHISTAFF	TT	9876546543	joyelroy24@gmail.com	ACTIVE	UPDATE	DEACTIVATE	TERMINATE

Copyright © 2018 DD . Designed by jo

Book Package Page

Description: This form shows book cargo package by customer.

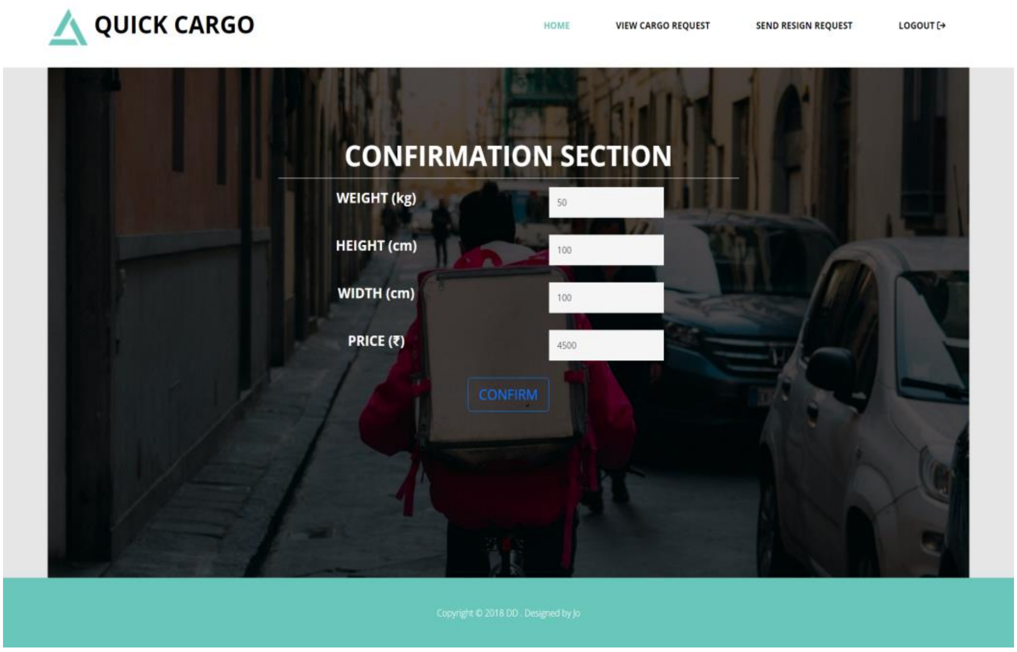


BOOK CARGO PACKAGE - *Pack 7* (₹4500)

WEIGHT(kg)	50
HEIGHT(cm)	100
WIDTH(cm)	100
FROM BRANCH	kochi
TO BRANCH	SELECT TO BRANCH SELECT TO BRANCH Vypin kochi Bangalore
PICKUP ADDRESS	
DESTINATION ADDRESS	PQRS,BANGALORE

Package Confirmation Page

Description: This form shows the confirmation form for staff of the package booked by customer.



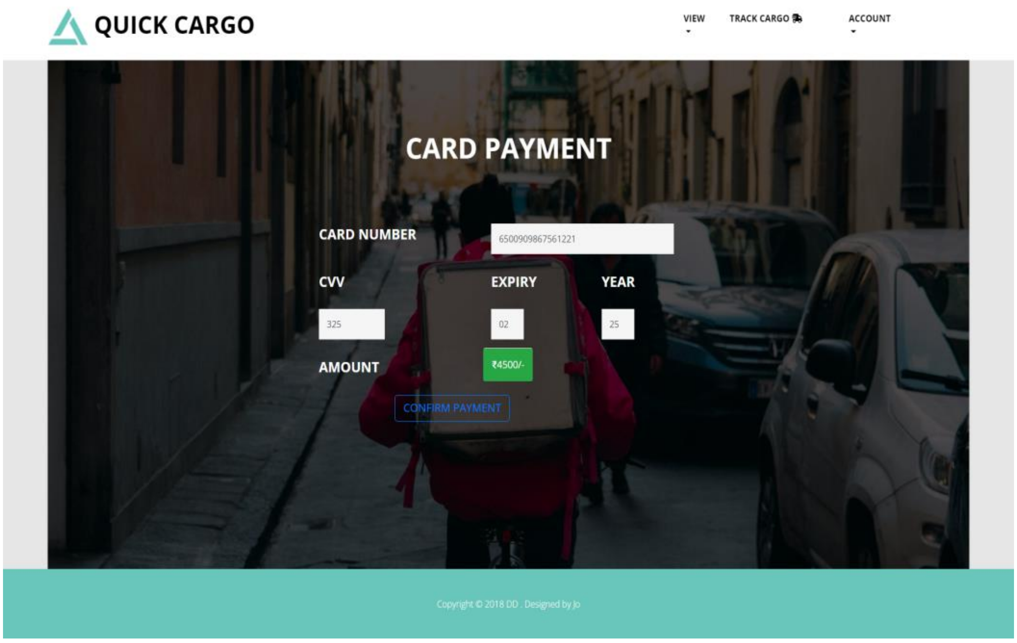
CONFIRMATION SECTION

WEIGHT (kg)	50
HEIGHT (cm)	100
WIDTH (cm)	100
PRICE (₹)	4500

Copyright © 2018 DD . Designed by Jo

Payment Page

Description: This form shows card payment by customer.



CARD PAYMENT

CARD NUMBER 6500909867561221

CVV 325

EXPIRY 02

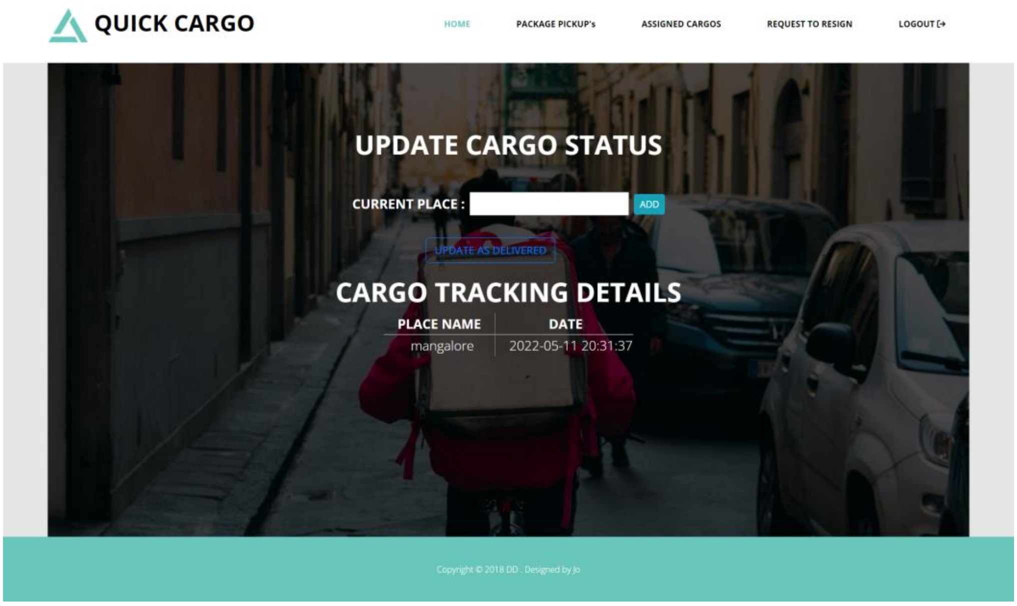
YEAR 25

AMOUNT ₹4500/-

Copyright © 2018 DD . Designed by Jo

Package Delivery Page

Description: This form is to update the status of package by delivery boy.



UPDATE CARGO STATUS

CURRENT PLACE :

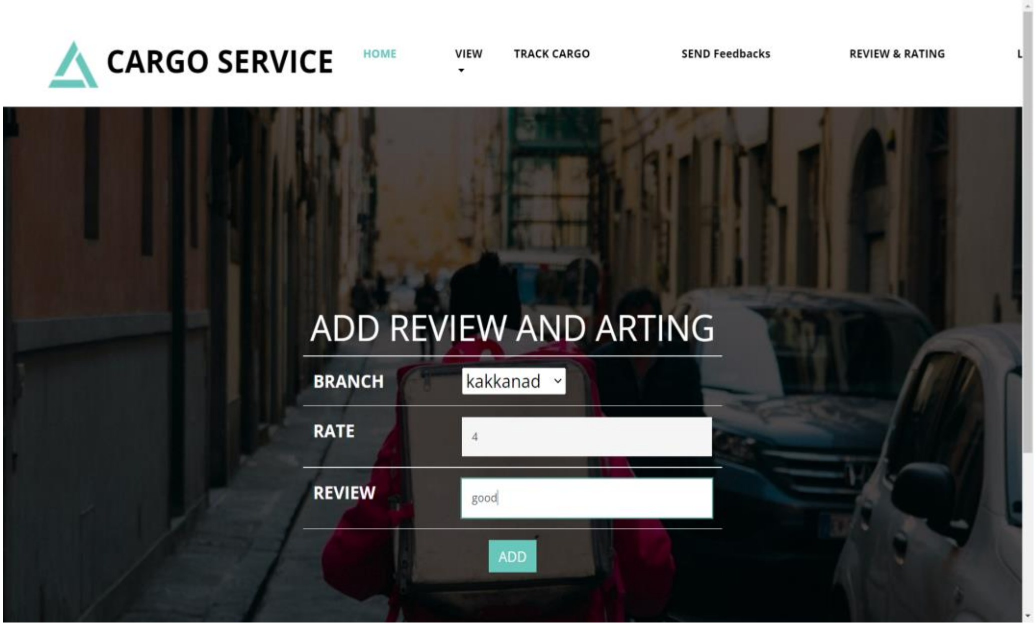
CARGO TRACKING DETAILS

PLACE NAME	DATE
mangalore	2022-05-11 20:31:37

Copyright © 2018 DD . Designed by Jo

Review and Feedback Page

Description: This form shows to send feedbacks, rating and review for customer.

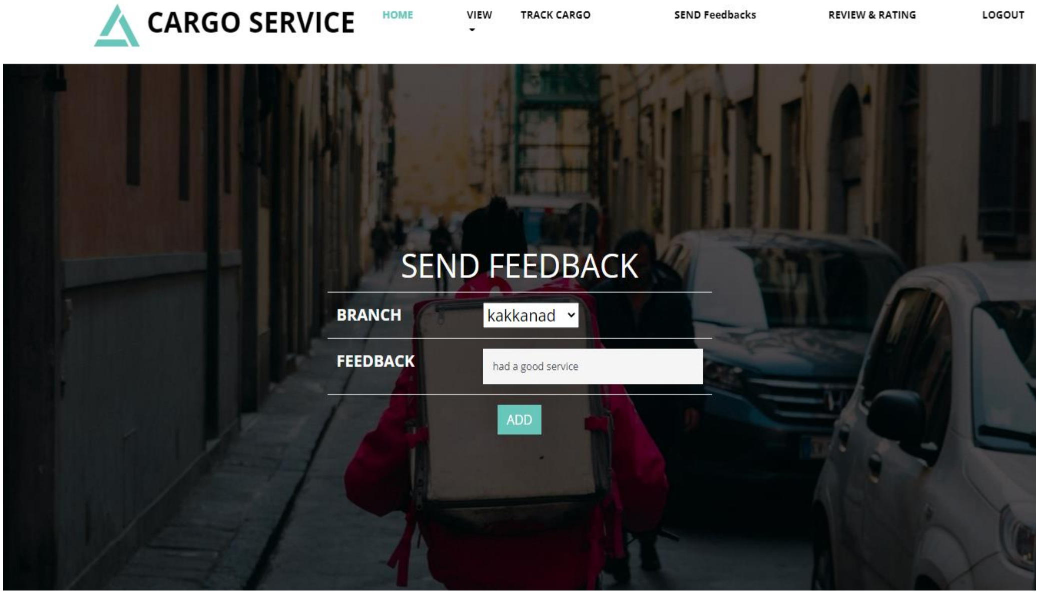


ADD REVIEW AND ARTING

BRANCH kakkanad

RATE 4

REVIEW good



SEND FEEDBACK

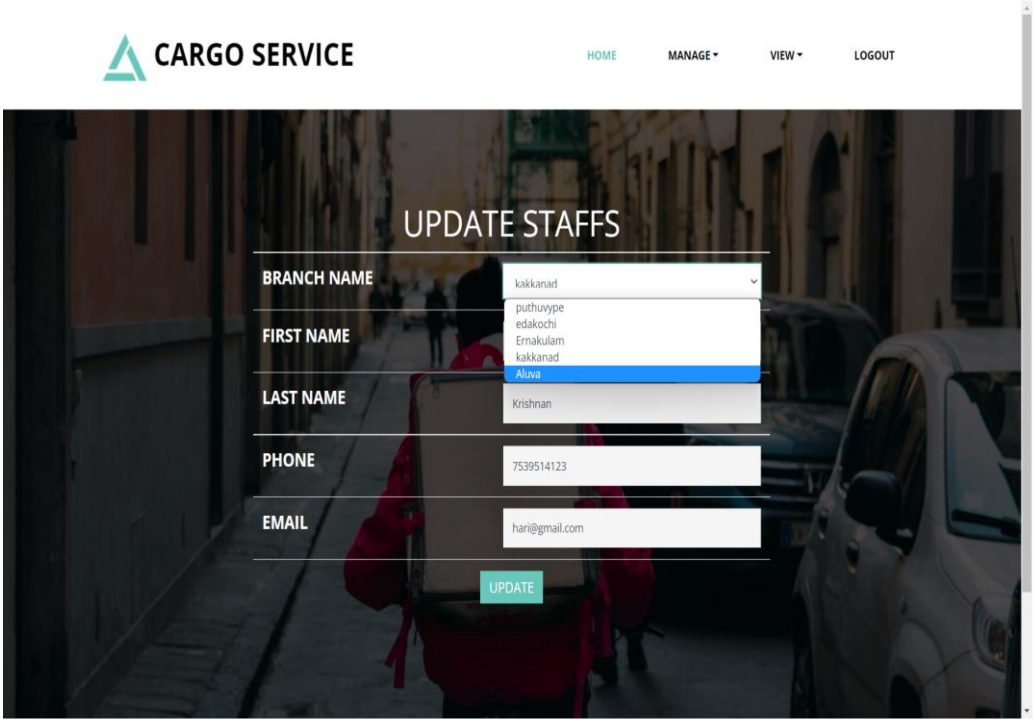
BRANCH kakkanad

FEEDBACK had a good service

.

Update Staff Page

Description: This form shows update staff details by branch.

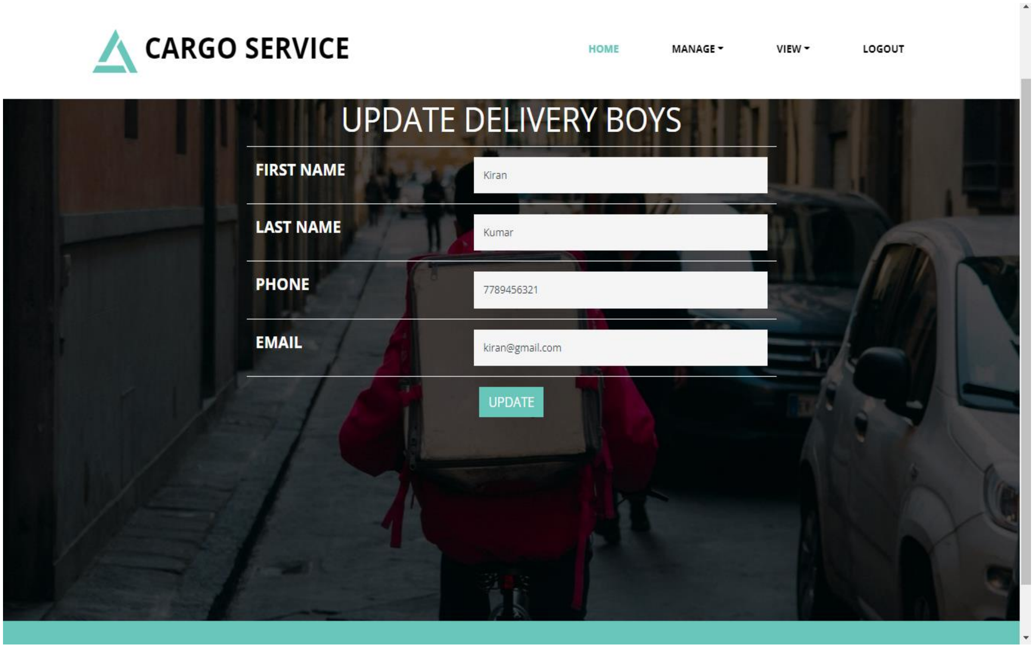


UPDATE STAFFS

BRANCH NAME	kakkanad puthuvype edakochi Ernakulam kakkanad Aluva
FIRST NAME	
LAST NAME	Krishnan
PHONE	7539514123
EMAIL	hari@gmail.com

Update Delivery boy Page

Description: This form shows update delivery boy details by branch.



UPDATE DELIVERY BOYS

FIRST NAME	Kiran
LAST NAME	Kumar
PHONE	7789456321
EMAIL	kiran@gmail.com

4.2 OUTPUT DESIGN

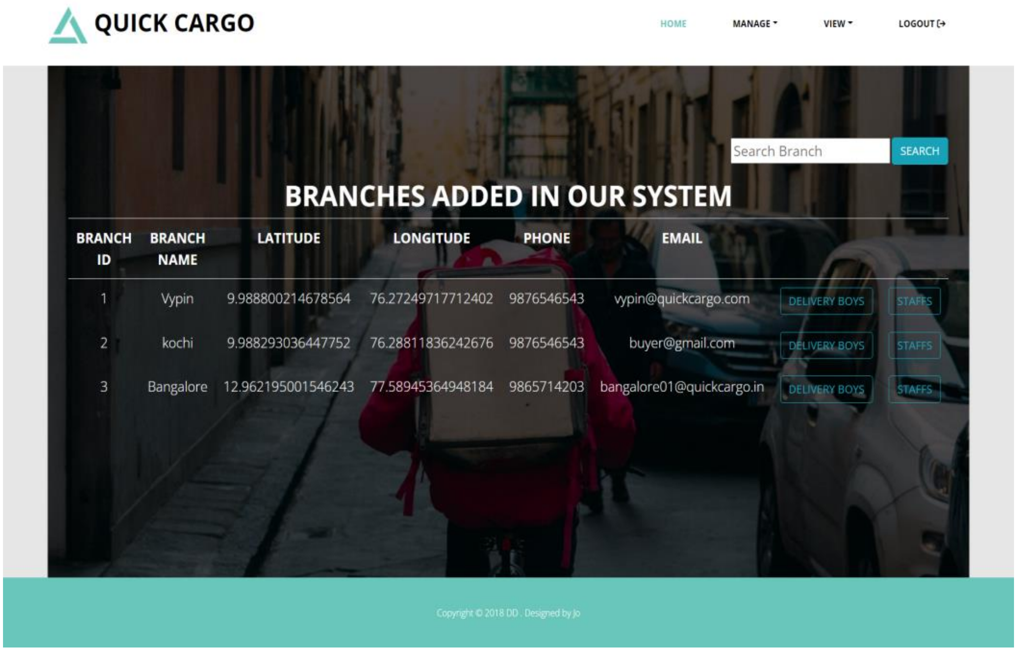
One of the important features of an information system for users is the output it produces. Output is the information delivered to users through the information system. Without quality output, the entire system appears to be unnecessary that users will avoid using it. Uses generally merit the system solely by its output. In order to create the most useful output possible. One works closely with the user through an interactive process, until the result is considered to be satisfactory.

Output design has been an ongoing activity almost from the beginning of the project. In the study phase, outputs were identified and described general in the project directive. A tentative output medium was then selected and sketches made for each output. In the feasibility analysis, a “best” new system was selected; its description identified the input and output media. In the design phase the system has included an evaluation and selection of specific equipment for the system.

Outputs from computer systems are required primarily to communicate the results of processing to the user. They are also used to provide a permanent copy of these results for later consultation.

View Branches

Description: This page shows the all the branches and their details for the admin.

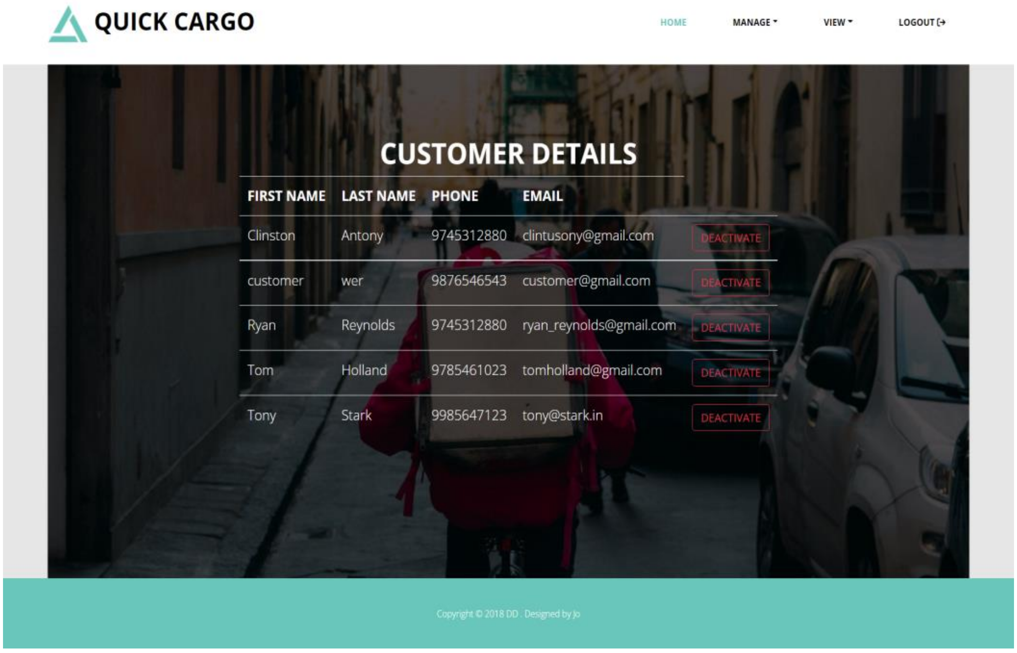


BRANCHES ADDED IN OUR SYSTEM

BRANCH ID	BRANCH NAME	LATITUDE	LONGITUDE	PHONE	EMAIL		
1	Vypin	9.988800214678564	76.27249717712402	9876546543	vypin@quickcargo.com	DELIVERY BOYS	STAFFS
2	kochi	9.988293036447752	76.28811836242676	9876546543	buyer@gmail.com	DELIVERY BOYS	STAFFS
3	Bangalore	12.962195001546243	77.58945364948184	9865714203	bangalore01@quickcargo.in	DELIVERY BOYS	STAFFS

View Customers Details Page

Description: This page shows the customer details page for admin.

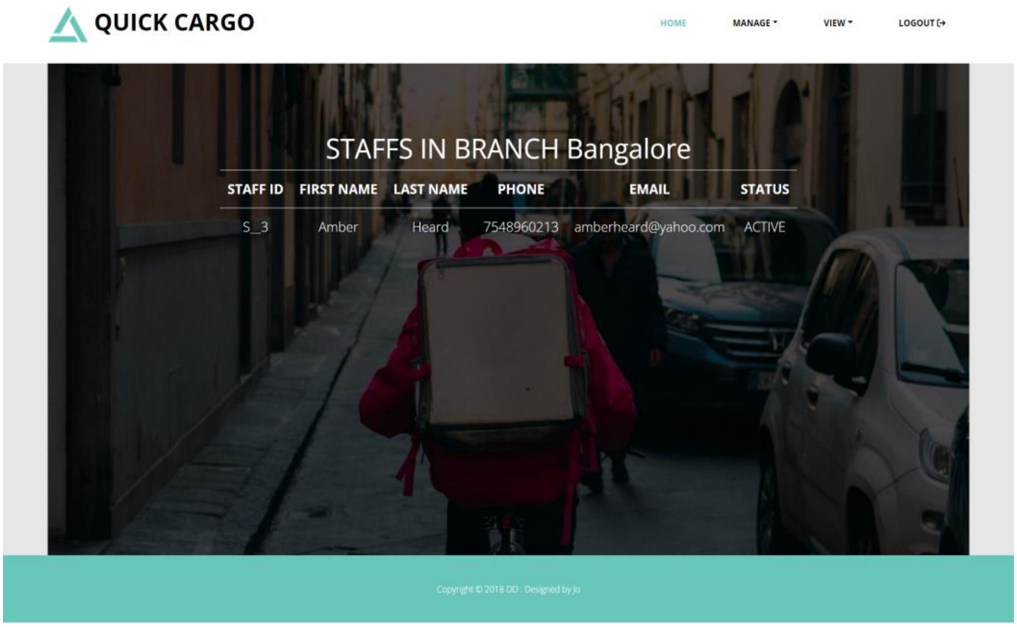


CUSTOMER DETAILS

FIRST NAME	LAST NAME	PHONE	EMAIL	
Clinston	Antony	9745312880	clintusony@gmail.com	DEACTIVATE
customer	wer	9876546543	customer@gmail.com	DEACTIVATE
Ryan	Reynolds	9745312880	ryan_reynolds@gmail.com	DEACTIVATE
Tom	Holland	9785461023	tomholland@gmail.com	DEACTIVATE
Tony	Stark	9985647123	tony@stark.in	DEACTIVATE

View Staffs and Delivery boys Details

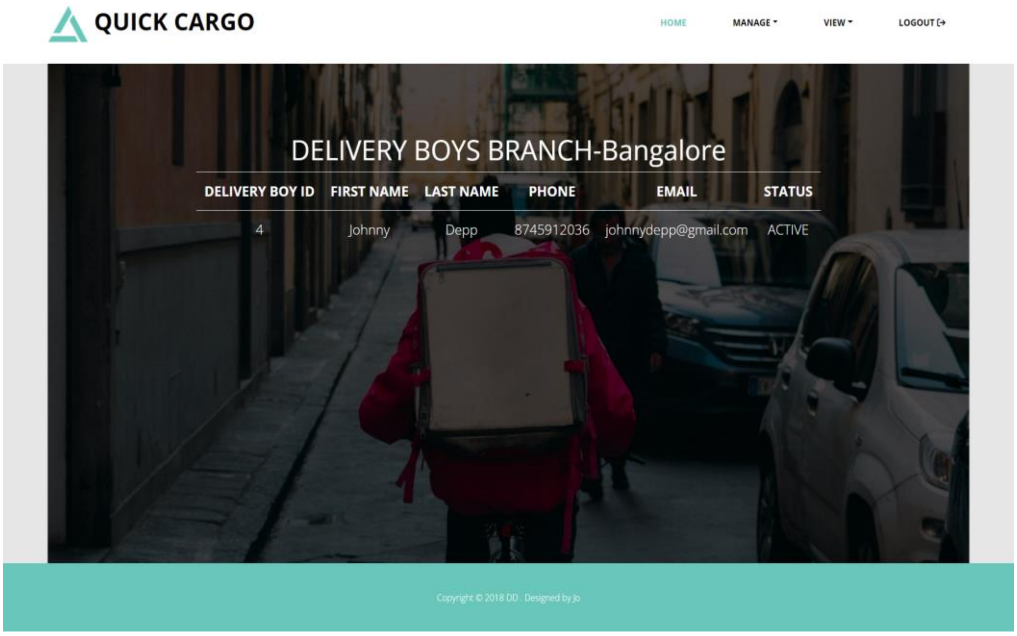
Description: This page shows the all staffs and delivery boys details of one branch for the admin and branch.



STAFFS IN BRANCH Bangalore

STAFF ID	FIRST NAME	LAST NAME	PHONE	EMAIL	STATUS
S_3	Amber	Heard	7548960213	amberheard@yahoo.com	ACTIVE

Copyright © 2018 DD . Designed by Jo



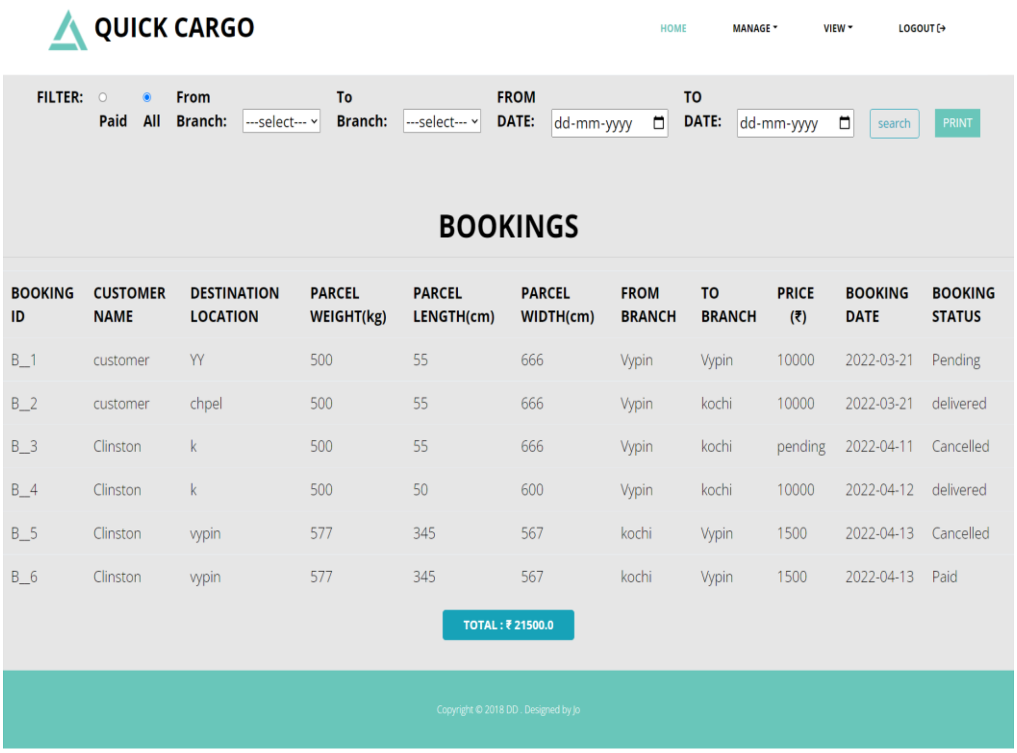
DELIVERY BOYS BRANCH-Bangalore

DELIVERY BOY ID	FIRST NAME	LAST NAME	PHONE	EMAIL	STATUS
4	Johnny	Depp	8745912036	johnnydepp@gmail.com	ACTIVE

Copyright © 2018 DD . Designed by Jo

View Bookings Page

Description: This page shows the booking report page for admin.



FILTER: ○ Paid ● All From Branch: ---select--- To Branch: ---select--- FROM DATE: dd-mm-yyyy TO DATE: dd-mm-yyyy

BOOKINGS

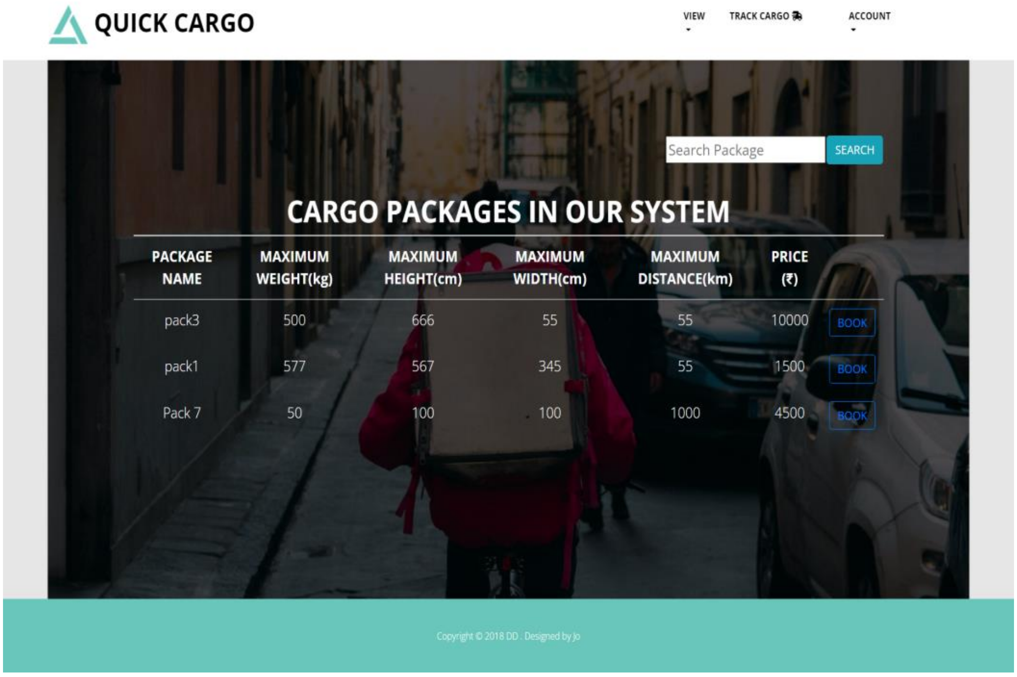
BOOKING ID	CUSTOMER NAME	DESTINATION LOCATION	PARCEL WEIGHT(kg)	PARCEL LENGTH(cm)	PARCEL WIDTH(cm)	FROM BRANCH	TO BRANCH	PRICE (₹)	BOOKING DATE	BOOKING STATUS
B_1	customer	YY	500	55	666	Vypin	Vypin	10000	2022-03-21	Pending
B_2	customer	chpel	500	55	666	Vypin	kochi	10000	2022-03-21	delivered
B_3	Clinston	k	500	55	666	Vypin	kochi	pending	2022-04-11	Cancelled
B_4	Clinston	k	500	50	600	Vypin	kochi	10000	2022-04-12	delivered
B_5	Clinston	vypin	577	345	567	kochi	Vypin	1500	2022-04-13	Cancelled
B_6	Clinston	vypin	577	345	567	kochi	Vypin	1500	2022-04-13	Paid

TOTAL : ₹ 21500.0

Copyright © 2018 DD . Designed by Jo

View Packages Page

Description: This page shows the package details to book for customer.



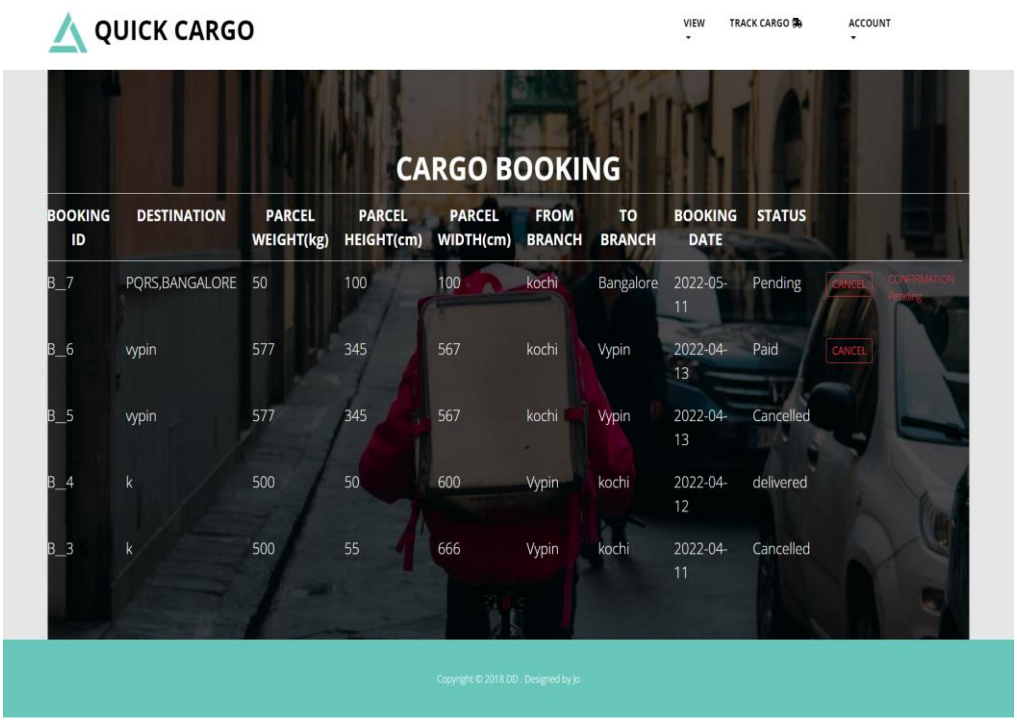
CARGO PACKAGES IN OUR SYSTEM

PACKAGE NAME	MAXIMUM WEIGHT(kg)	MAXIMUM HEIGHT(cm)	MAXIMUM WIDTH(cm)	MAXIMUM DISTANCE(km)	PRICE (₹)	
pack3	500	666	55	55	10000	BOOK
pack1	577	567	345	55	1500	BOOK
Pack 7	50	100	100	1000	4500	BOOK

Copyright © 2018 DD . Designed by Jo

View Customer Bookings Page

Description: This page shows the booking details for customer.



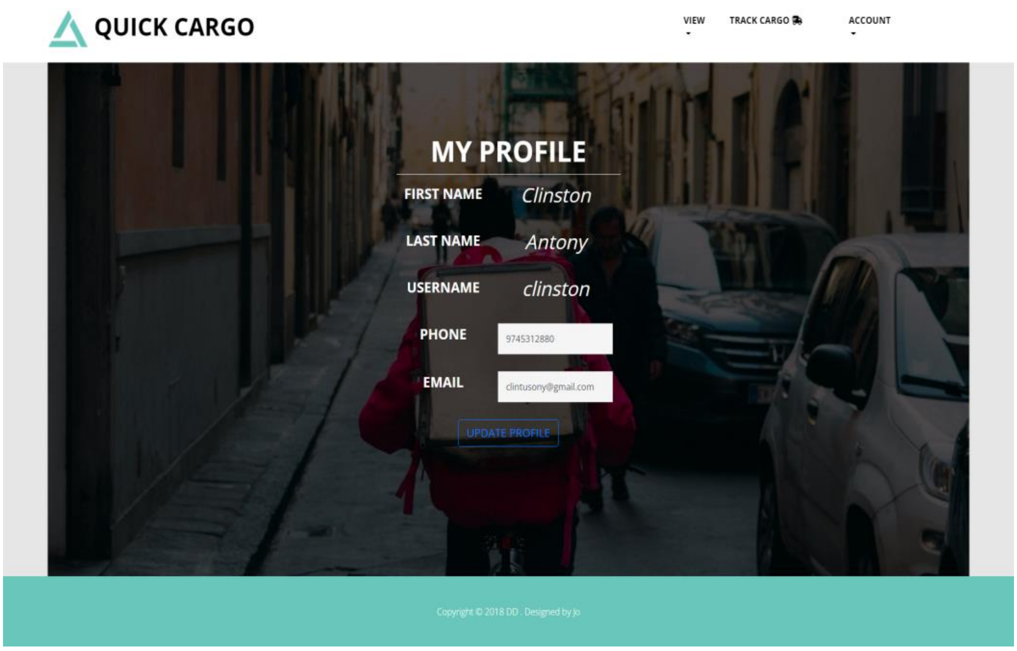
CARGO BOOKING

BOOKING ID	DESTINATION	PARCEL WEIGHT(kg)	PARCEL HEIGHT(cm)	PARCEL WIDTH(cm)	FROM BRANCH	TO BRANCH	BOOKING DATE	STATUS		
B_7	PQRS,BANGALORE	50	100	100	kochi	Bangalore	2022-05-11	Pending	CANCEL	CONFIRMATION Pending
B_6	vypin	577	345	567	kochi	Vypin	2022-04-13	Paid	CANCEL	
B_5	vypin	577	345	567	kochi	Vypin	2022-04-13	Cancelled		
B_4	k	500	50	600	Vypin	kochi	2022-04-12	delivered		
B_3	k	500	55	666	Vypin	kochi	2022-04-11	Cancelled		

Copyright © 2018 DD . Designed by Jo

View Profile Page

Description: This page shows the profile for customer.



MY PROFILE

FIRST NAME: Clinston

LAST NAME: Antony

USERNAME: clinston

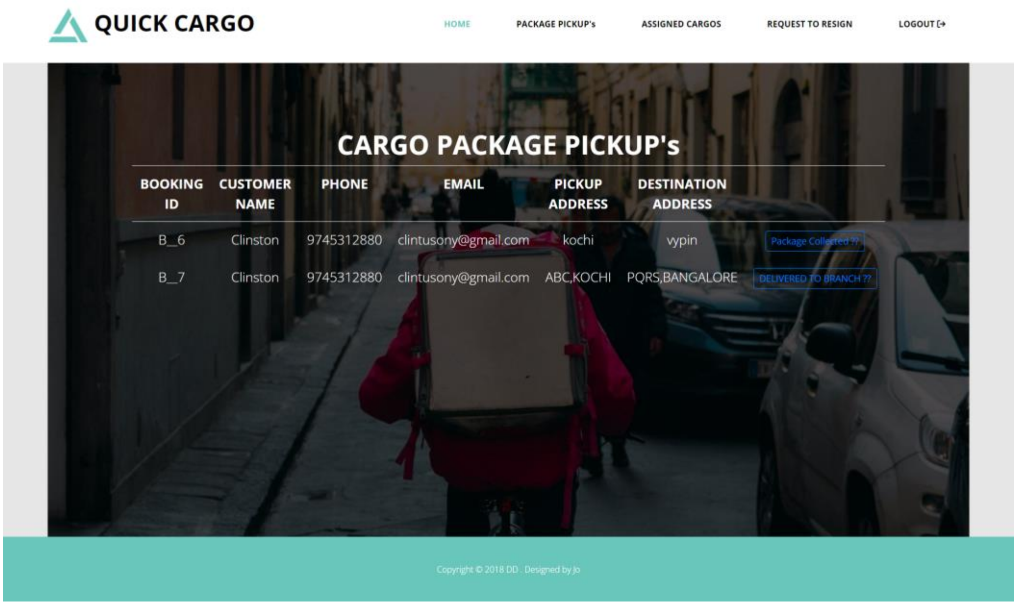
PHONE: 9745312880

EMAIL: clintusony@gmail.com

Copyright © 2018 DD . Designed by Jo

View Package Collection Page

Description: This page shows the package details for delivery boy to transfer.



CARGO PACKAGE PICKUP's

BOOKING ID	CUSTOMER NAME	PHONE	EMAIL	PICKUP ADDRESS	DESTINATION ADDRESS	
B_6	Clinston	9745312880	clintusony@gmail.com	kochi	vypin	Package Collected ??
B_7	Clinston	9745312880	clintusony@gmail.com	ABC,KOCHI	PQRS,BANGALORE	DELIVERED TO BRANCH ??

4.3 DATABASE DESIGN

4.3.1 NORMALIZATION

Designing a database is a complex task and the normalization theory is a useful aid in this design process. The process of normalization is concerned with transformation of conceptual schema into computer representation form.

A bad database design may lead to certain undesirable situation such us,

- Repetition of information
- Inability to represent certain information
- Loss of information

To minimize these anomalies, normalization mat be used. If the database is in a normalization form, the data can be restructured and can maintain it easily. This is important that the database using that we are using may free from data redundancy and inconsistency. For this need we maintain the tables in a normalized manner.

First Normal Form

A relation is in first normal form (INF), if and only if all its attributes are based on single domain. The objective of normalizing a table is in to remove its repeating groups and ensure that all entries of the resulting table have at most single value.

Second Normal Form

A table is said to be in Second Normal Form (2NF), when it is in 1NF and every attribute in the record is functionally dependent upon the whole key, and not just a part of the key.

Third Normal Form

A table is in third Normal Form (3NF), when it is in 2NF and every non-key attribute is functionally dependent on just the primary key.

Table name: tbl_login

Description: Stores the login details of users that exist in the system.

FIELD	DATATYPE	CONSTRAINTS	DESCRIPTION
username	Varchar(30)	Primary key	Login Username
usertype	Varchar(20)	Not null	Type of user
password	Varchar(30)	Not null	Password

Table name: tbl_branch

Description: Table which stores branch details

FIELD	DATATYPE	CONSTRAINTS	DESCRIPTION
branch_id	Varchar(10)	Primary key	Branch ID
username	Varchar(10)	Unique	Username for branch login
b_name	Varchar(30)	Unique	Branch Name
b_email	Varchar(30)	Unique	Branch Email id
b_phone	Numeric(10)	Unique	Branch Phone Number
b_latitude	Varchar(40)	Not null	Branch latitude
b_longitude	Varchar(30)	Not null	Branch longitude

Table name: tbl_customer

Description: Stores the details of customers

FIELD	DATATYPE	CONSTRAINTS	DESCRIPTION
cust_id	Varchar(10)	Primary key	Customer ID
username	Varchar(25)	Unique	Username for customer
c_phoneno	Numeric(10)	Unique	Customer Phone Number
c_email	Varchar(30)	Unique	Customer Email
c_fname	Varchar(30)	Not null	Customer First Name
c_lname	Varchar(30)	Not null	Customer Last Name
c_latitude	Varchar(30)	Not null	Customer latitude
c_longitude	Varchar(30)	Not null	Customer longitude

Table name: tbl_ packages

Description: Table which stores package details

FIELD	DATATYPE	CONSTRAINTS	DESCRIPTION
package_id	Varchar(10)	Primary key	Package ID
packname	Varchar(10)	Unique	Package Name
weight	number(10)	Not null	Max weight
height	number(10)	Not null	Max Height
width	number(10)	Not null	Max Width
distance	number(10)	Not null	Max distance
amount	Decimal(10,2)	Not null	Amount for the package
p_status	Varchar(30)	Not null	Package status

Table name: tbl_staff

Description: Stores the details of staff working in the cargo service

FIELD	**DATATYPE**	**CONSTRAINTS**	**DESCRIPTION**
staff_id	Varchar(10)	Primary key	Staff ID
branch_id	Varchar(30)	Foreign Key	Staff's Branch ID
username	Varchar(20)	Unique	Username of staff
s_ phoneno	Numeric(10)	Unique	Staff Phone's Phone Number
s_ email	Varchar(30)	Unique	Staff's Email
s_ fname	Varchar(30)	Not null	Staff's First Name
s_ lname	Varchar(30)	Not null	Staff's Last Name

Table name: tbl_deliveryboy

Description: Stores the details of deliveryboys working in the cargo

FIELD	**DATATYPE**	**CONSTRAINTS**	**DESCRIPTION**
dboy_id	Varchar(10)	Primary key	Delivery boy ID
branch_id	Varchar(30)	Foreign Key	Delivery boy's Branch ID
username	Varchar(20)	Unique	Username of Delivery boy
d_email	Varchar(30)	Unique	Delivery boy's Email
d_phoneno	Numeric(10)	Unique	Delivery boy's Phone Number
d_fname	Varchar(30)	Not null	Delivery boy's First Name
d_lname	Varchar(30)	Not null	Delivery boy's Last Name

Table name: tbl_booking

Description: Table which stores payment details

FIELD	DATATYPE	CONSTRAINTS	DESCRIPTION
booking_id	Varchar(10)	Primary key	Booking ID
cust_id	Varchar(10)	Foreign key	Customer ID
from_branch _id	Varchar(10)	Foreign key	From Branch ID
to_branch_id	Varchar(10)	Foreign key	To Branch ID
width	number(10)	Not null	Max Width
weight	number(10)	Not null	Max Weight
height	number(10)	Not null	Max Height
loc_from	Varchar(10)	Not null	From Location
loc_to	Varchar(10)	Not null	To Location
booking_time	Date	Not null	Booking date and time
amt	Decima(10,2)	Not null	Amount to pay
booking_ status	Varchar(20)	Not null	Booking Status

Table name: tbl_delivery_status

Description: Table which stores payment details

FIELD	DATATYPE	CONSTRAINTS	DESCRIPTION
delivery_ status_id	Varchar(10)	Primary key	Delivery Status ID
booking_id	Varchar(10)	Foreign key	Booking ID
placename	Decima(10,2)	Not null	Name of the place where the package reached
status_date	Date	Not null	Date and Time of status updated
delivery_ status	Varchar(20)	Not null	Delivery Status

Table name: tbl_payment

Description: Table which stores payment details

FIELD	DATATYPE	CONSTRAINTS	DESCRIPTION
payment_id	Varchar(10)	Primary key	Payment ID
booking_id	Varchar(10)	Foreign key	Booking ID
paid	Decima(10,2)	Not null	Amount Paid
payment_ date	Date	Not null	Date of Payment
payment_ status	Varchar(20)	Not null	Payment Status (Paid/Not paid)

5.1 INTRODUCTION

Testing is the process of examining the software to compare the actual behavior with that of the excepted behavior. The major goal of software testing is to demonstrate that faults are not present. In order to achieve this goal, the tester executes the program with the intent of finding errors. Though testing cannot show absence of errors but by not showing their presence it is considered that these are not present.

System testing is the first Stage of implementation, which is aimed at ensuring that the system works accurately and efficiently before live operations commences. Testing is vital to the success of the system. System testing makes a logical assumption that if all the parts of the system are correct and the goal will be successfully achieved. A series of testing are performed for the proposed system before the proposed system is ready for user acceptance testing.

Levels of Testing

1. Unit testing
2. Integration testing
3. Validation testing
4. Output testing

Unit testing

In this each module is tested individually before integrating it to the final system. Unit test focuses verification in the smallest unit of software design in each module. This is also known as module testing as here each module is tested to check whether it is producing the desired output and to see if any error occurs.

Integration testing

Integration testing (sometimes called integration and testing, abbreviated I&T) is the phase in software testing in which individual software modules are combined and tested as a group. It occurs after unit testing and before validation testing.

Integration testing takes as its input modules that have been unit tested, groups them in larger aggregates, applies tests defined in an integration test plan to those aggregates, and delivers as its output the integrated system ready for system testing. The purpose of integration testing is to verify functional, performance, and reliability requirements placed on major design items.

Output testing

No system could be useful if it does not produce the required output in the specific format. Output testing is performed to ensure the correctness of the output and its format. The output generated or displayed by the system is tested asking the users about the format required by them.

Validation testing

In software project management, software testing, and software engineering, validation is the process of checking that a software system meets specifications and that it fulfills its intended purpose. The errors which are uncovered during integration testing are connected during this phase.

5.2 TEST CASES

A test plan documents the strategy that will be used to verify and ensure that a product or system meets its design specification and other requirements. A test plan is usually prepared by or with significant input from Test Engineers.

Unit Testing

Form	Procedure	Expected Result	Actual Result	Status
Entry Form	Choose whether to Login or Register, About us			Pass
Login Form	Enter valid username and password	Should validate user and provide link to user accounts	Got entry to accounts	Pass
Branch Form	Enter all mandatory fields	Should validate all entered fields and flash a message indicating successful registration	Message indicating successful registration is shown	Pass
Customer Form	Enter all mandatory fields	Should validate all entered fields and flash a message indicating successful entered	Message indicating successful registration is shown	Pass
Staff Form	Enter all mandatory fields	Should validate all entered fields and flash a message indicating successful registration	Message indicating successful registration is shown	Pass
Delivery boy Form	Enter all mandatory fields	Should validate all entered fields and flash a message indicating successful registration	Message indicating successful registration is shown	Pass

Package Form	Enter all mandatory fields	Should validate all entered fields and flash a message indicating successful entered	Message indicating successful registration is shown	Pass
Booking Form	Enter all mandatory fields	Should validate all entered fields and flash a message indicating successful entered	Message indicating successful registration is shown	Pass
Payment Form	Enter all mandatory fields	Should validate all entered fields and flash a message indicating successful entered	Message indicating successful registration is shown	Pass

Integration testing

Form	Expected Result	Actual Result	Status
Login and user account forms	Get entry to appropriate user page	Appropriate user page is displayed	Pass
Customer Form	Must add customer details successfully	Insertion is successfully	Pass
Branch form	Must add branch details successfully	Specified entry updated	Pass
Staff form	Must add staff details successfully	Insertion is successfully	Pass
Delivery boy form	Must add delivery boy details successfully	Insertion is successfully	Pass
Package form	Must add package details successfully	Insertion is successfully	Pass
Booking form	Must update the specified entry in the database	Specified entry updated	Pass
Payment Form	Must update the specified entry in the database	Specified entry updated	Pass
Report	Must display reports	Reports displayed	Pass

Validation Testing

Form	Expected Result	Actual Result	Status
Create user	Check all mandatory fields and validate all entered data fields	If any error found display message and the same screen is displayed else record saved and confirmed	Pass
Edit User	Edit the row corresponding to the value entered	If the value entered is invalid error message is thrown otherwise message indicating successful deletion	Pass

6.1 INTRODUCTION

Implementation is the stage in the project where theoretical design is turned into a working system and is giving confidence on the new system for the users which will work efficiently and effectively. It involves careful planning, investigation of the current system and its constraints on implementations, design of methods to achieve the changeover, an evaluation, of change over methods. Apart from planning major tasks of preparing the implementation are education and training of users. The major complex system being implemented the more evolved will be the system analysis and the design effort required just for implementation. An implementation coordination committee based on policies of individual organisation has been appointed. The implementation process begins with preparing plan for implementation of the system. According to this plan the activities are to be carried out discussions made regarding the equipment and resources and the additional equipment has to be acquired to implement the new system.

Implementation is the final and important phase. The most critical stage in achieving a successful new system and in giving the users confidence that the new system and in giving the users confidence that the new system will work and be effective. The system can be implemented only after thorough testing is done and if found to working according to the specification.

6.2 INSTALLATION PROCEDURE

Installation of software refers to the final installation of the package in the real environment, to the satisfaction of the intended users and the successful operation of the system. In many organizations, those who commission the software development project will not be the one to operate them. In the initial stage, the person who is not sure that the software will make the jobs easier will doubt about the software.

Implementation is the stage of the project where the theoretical design is turned into a working system. At this stage, the main work load, the greatest upheaval and the major impact on the existing system shifts to the user department. If the implementation is not carefully planned and controlled, it can cause confusion.

Implementation includes all those activities that take place to convert from the old system to new one. Proper implementation is essential to provide a reliable system to meet the organizational requirements. Successful implementation may guarantee improvement in the organization using the new system, but improper installation will prevent it. The process of putting the developed system in to actual use is called system implementation.

6.3 IMPLEMENTATION PLAN

The Implementation Plan describes how the information system will be deployed, installed and transitioned into an operational system. The plan contains an overview of the system, a brief description of the major tasks involved in the implementation, the overall resources needed to support the implementation effort, and any site-specific implementation requirements. The plan is developed during the Design Phase and is updated during the Development Phase the final version is provided in the Integration and Test Phase and is used for guidance during the Implementation Phase.

7.1 FUTURE ENHANCEMENT

We have tried our best to present the information effectively, yet there can be further enhancement in the application. We have taken care of all the critical aspects, which were needed to be taken care of. Because of fast changes in the world of programming this system will gradually get outdated and less effective. For the time being it's possible to overcome problems by amendments and minor modifications to acknowledge the need of fundamental design.

Though the new system provides base for improving the efficiency of operations, there are a lot of future enhancements that can be added to this project. Keeping this in view, a provision has been made in the system to facilities easy modification updating in the future. Any modification will not affect the normal working of the system.

The development system is very interactive, coded in such a way to ensure maximum user friendliness and also allows flexibility for future.
It can be extended in such a way that:

- Improve the security.
- Improve the performance.

BIBLIOGRAPHY

Websites

Following websites are referred to create this project reports.

- www.stackoverflow.com –

https://stackoverflow.com/

https://stackoverflow.com/questions/51562789/get-values-in-flask-query

https://stackoverflow.com/questions/14343812/redirecting-to-url-in-flask

https://stackoverflow.com/questions/11912322/select-location-from-map-by-user-to-get-latitude-and-longitude

- www.mysqltutorial.com –

https://www.mysqltutorial.org/mysql-inner-join.aspx/

- www.studentsprojectsguide.com
- https://flask.palletsprojects.com/en/2.1.x/
- www.geeksforgeeks.org -

https://www.geeksforgeeks.org/html-table-rules-attribute/

- www.dreamincode.net
- https://getbootstrap.com/
- www.w3schools.com –

https://www.w3schools.com/html/default.asp

https://www.w3schools.com/html/html_tables.asp

https://www.w3schools.com/python/python_mysql_create_db.asp

https://www.w3schools.com/python/python_mysql_getstarted.asp

https://www.w3schools.com/python/python_for_loops.asp

https://www.w3schools.com/python/python_modules.asp

https://www.w3schools.com/python/python_functions.asp

APPENDICES

APPENDIX A

Sample Source Code/Pseudo Code:

```
#main.py

from flask import *
from public import public
from admin import admin
from branch import branch
from staff import staff
from dboy import dboy
from customer import customer
import smtplib
from email.mime.text import MIMEText
from flask_mail import Mail

app=Flask(__name__)
app.secret_key="hello"
app.register_blueprint(public)
app.register_blueprint(branch,url_prefix='/branch')
app.register_blueprint(admin,url_prefix='/admin')
app.register_blueprint(staff,url_prefix='/staff')
app.register_blueprint(dboy,url_prefix='/dboy')
app.register_blueprint(customer,url_prefix='/customer')

mail=Mail(app)
app.config['MAIL_SERVER']='smtp.gmail.com'
app.config['MAIL_PORT'] = 587
app.config['MAIL_USERNAME'] = '**********@*********'
app.config['MAIL_PASSWORD'] = '*************'
app.config['MAIL_USE_TLS'] = False
app.config['MAIL_USE_SSL'] = True
app.run(debug=True,port=5030)

#database.py


import mysql.connector
password=""
database = "c_binr"
port=3306
def select(q):
    cnx = mysql.connector.connect(user="root", password=password,
host="localhost", database=database,port=port)
    cur = cnx.cursor(dictionary=True)
    cur.execute(q)
    result = cur.fetchall()
    cur.close()
    cnx.close()
    return result
def update(q):
    cnx = mysql.connector.connect(user="root", password=password,
host="localhost", database=database,port=port)
    cur = cnx.cursor(dictionary=True)
    cur.execute(q)
    cnx.commit()
    result = cur.rowcount
    cur.close()
    cnx.close()
    return result
def delete(q):
    cnx = mysql.connector.connect(user="root",
password=password,host="localhost", database=database,port=port)
    cur = cnx.cursor(dictionary=True)
    cur.execute(q)
    cnx.commit()
    result = cur.rowcount
    cur.close()
    cnx.close()
    return result
def insert(q):
    cnx = mysql.connector.connect(user="root", password=password,
host="localhost", database=database,port=port)
    cur = cnx.cursor(dictionary=True)
    cur.execute(q)
    cnx.commit()
    result = cur.lastrowid
    cur.close()
    cnx.close()
    return result
#public.py
from flask import *
from database import *
public=Blueprint('public',__name__)
@public.route('/',methods=['get','post'])
def home():
        return render_template('home.html')
@public.route('/login',methods=['get','post'])
def login():
    data={}
    if 'submit' in request.form:
        uname=request.form['uname']
        password=request.form['password']
        q="select * from login where username='%s' and
password='%s'"%(uname,password)
        res=select(q)
```

```
 if res:
        session['username']=res[0]['username']
        if res[0]['user_type']=='admin':
            return redirect(url_for('admin.adminhome'))
        if res[0]['user_type']=='resigned':
            flash("YOU ARE RESIGNED")
            return redirect(url_for('public.login'))
        if res[0]['user_type']=='inactive':
            flash("YOU ARE BLOCKED")
            return redirect(url_for('public.login'))
        if res[0]['user_type']=='resigned':
            flash("YOU ARE RESIGNED")
            return redirect(url_for('public.login'))
        if res[0]['user_type']=='branch':
            q="select branch_id,branch_name from branches where
username='%s'"%(uname)
            res=select(q)
            session['bname']=res[0]['branch_name']
            session['bid']=res[0]['branch_id']
            return redirect(url_for('branch.branchhome'))
        if res[0]['user_type']=='staff':
            q="select * from staffs where username='%s'"%(uname)
            res=select(q)
            print(res)
            session['sid']=res[0]['staff_id']
            session['sname']=res[0]['first_name']+" "+res[0]['last_name']
            return redirect(url_for('staff.staffhome'))
        if res[0]['user_type']=='dboy':
            q="select * from deliveryboys where username='%s'"%(uname)
            res=select(q)
            print(res)
            session['did']=res[0]['boy_id']
```

```
    session['dname']=res[0]['first_name']+" "+res[0]['last_name']
                return redirect(url_for('dboy.dboyhome'))
            if res[0]['user_type']=='customer':
                q="select * from customers where username='%s'"%(uname)
                res=select(q)
                print(res)
                session['cid']=res[0]['customer_id']
                session['cname']=res[0]['first_name']+" "+res[0]['last_name']
                return redirect(url_for('customer.customerhome'))
        else:
            data['invalid']="Invalid Login Details"
    return render_template('login.html',data=data)
@public.route('/customerreg',methods=['get','post'])
def customerreg():
    data={}
    if 'submit' in request.form:
        fname=request.form['fname']
        lname=request.form['lname']
        ph=request.form['phone']
        email=request.form['email']
        lat=request.form['lat']
        lon=request.form['lon']
        uname=request.form['uname']
        password=request.form['password']
        cpassword=request.form['cpassword']
        if cpassword==password:
            q="select * from login where username='%s'"%(uname)
            res=select(q)
            if res:
                flash('THIS USER NAME  ALREADY TAKEN BY ANOTHER 
USER')
                return redirect(url_for('public.customerreg'))
```

```
  else:
            q="select * from branches where phone='%s' or
email='%s'"%(ph,email)
            res1=select(q)
            q="select * from deliveryboys where phone='%s' or
email='%s'"%(ph,email)
            res2=select(q)
            q="select * from staffs where phone='%s' or email='%s'"%(ph,email)
            res3=select(q)
            if res1 or res2 or res3:
               flash("ALREADY YOU HAVE AN ACCOUNT")
               return redirect(url_for('public.customerreg'))
            else:
               q="insert into login values('%s','%s','customer')"%(uname,password)
               lid=insert(q)
               q="insert into customers
values(NULL,'%s','%s','%s','%s','%s','%s','%s')"%(uname,fname,lname,ph,email,l
at,lon)
               insert(q)
               return redirect(url_for('public.customerreg'))
      else:
         flash("PASSWORD AND CONFORM PASSWORD DOES NOT
MATCH")
   return render_template('customerreg.html',data=data)

<! login.html -->

{% include 'publicheader.html' %}
<div class="tm-page-wrap mx-auto">
         <section class="tm-banner">
            <div class="tm-container-outer tm-banner-bg">
               <div class="container">
```

```
                <div class="row tm-banner-row tm-banner-row-header">
                  <div class="col-xs-12">
                    <div class="tm-banner-header">
                      <center>
                    <form method="post">
                    <h1><B>LOGIN</B></h1> <br>
                    <table class="table" style="width: 500px" rules=none>
                      {% if data['invalid'] %}
                        <tr style="border-bottom-style: hidden;">
                          <th class="btn btn-danger">{{data['invalid']}}</th>
                        </tr>
                        {% endif %}
                      <tr style="border-bottom-style: hidden;">
                        <th>USERNAME</th>
                        <td><input type="text" name="uname" style="border-
radius: 10px;" class="form-control"></td>
                      </tr>
                      <tr style="border-bottom-style: hidden;">
                        <th>PASSWORD</th>
                        <td><input type="password" style="border-radius:
10px;" name="password" class="form-control"></td>
                      </tr>
                      <tr style="border-bottom-style: hidden;">
                        <td colspan="2" align="center"><input type="submit"
name="submit" value="Login" class="btn btn-info btn-lg"></td>
                      </tr>
                      <tr>
                        <td colspan="2" align="center"><a
href="public_chanagepassword" class="btn btn-secondary">Forgot Password
?</a></td>
                      </tr>
                    </table>
```

```
                                 </form>
                             </center>
                                 </div>
                             </div>  <!-- col-xs-12 -->
                         </div> <!-- row -->
                       <div class="tm-banner-overlay"></div>
                    </div>  <!-- .container -->
                </div>     <!-- .tm-container-outer -->
            </section>
        </div>
{% include 'footer.html' %}

<! publicheader.html -->

<!DOCTYPE html>
<html lang="en">
<head>
     <style type="text/css">
        th,td{
            font-size: 20px
        }
    </style>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1">
        <link rel="icon" href="/static/images/img/logo.png" type="image/icon
type">
    <title>CARGO</title>
    <link rel="stylesheet" href="font-awesome-4.7.0/css/font-awesome.min.css">
     <link rel="stylesheet" href="/static/css/bootstrap.min.css">
     <link rel="stylesheet" type="text/css" href="/static/slick/slick.css"/>
    <link rel="stylesheet" type="text/css" href="/static/slick/slick-theme.css"/>
```

```
   </head>
   <body>
    <div class="tm-main-content" id="top">
      <div class="tm-top-bar-bg"></div>
      <div class="tm-top-bar" id="tm-top-bar">
        <div class="tm-container-outer">
          <div class="row">
            <nav class="navbar navbar-expand-lg narbar-light">
              <a class="navbar-brand mr-auto" href="/">
                <img src="/static/img/logo.png" alt="Site logo">
                QUICK CARGO</a>
              <button type="button" id="nav-toggle" class="navbar-toggler
collapsed" data-toggle="collapse" data-target="#mainNav" aria-expanded="false"
aria-label="Toggle navigation">
                <span class="navbar-toggler-icon"></span>
              </button>
              <div id="mainNav" class="collapse navbar-collapse tm-bg-
white">
               <ul class="navbar-nav ml-auto">
                 <li class="nav-item">
                  <a class="nav-link" href="login">LOGIN</a>
                </li>
                <li class="nav-item">
                  <a class="nav-link" href="/customerreg">CUSTOMER
REGISTRATION</a>
                </li>
              </ul>
            </div>
          </nav>
        </div> <!-- row -->
      </div> <!-- container -->
    </div> <!-- .tm-top-bar -->  <! publicheader.html ends here-->
```

APPENDIX B

Acronyms

ER- Entity relation

CSS –Cascading Style Sheet

MySQL-Structured Query Language

DFD-Data Flow Diagram

INF-First Normal Form

2NF-Second Normal Form

3NF-Third Normal Form