Republic of the Philippines
Taguig City University
General Santos, Central Bicutan, Taguig City

DATA ANALYTICS AND SMS NOTIFICATION FOR EFFECTIVE TRUCK QUEUEING MANAGEMENT SYSTEM

A Thesis

Presented to the Faculty College of Information and Communication Technology

Taguig City University

In Partial Fulfillment

of the Requirements For The Degree

Bachelor of Science in Computer Science

By:

Abrina, John Louie A.

Botona, Jay-alou B.

Evangelista, Dianne Tan

Mahinay, Alexis

July 2020

Chapter I

INTRODUCTION

A queue management system is used to control waiting lines. The queue management is a vital part of the logistic companies daily activities in receiving in delivery of local and imported goods to be stored in the warehouse that will distributed to stores nationwide. An effective queue management system will helps in overall productivity, can reduce cost and thus improve profitability. Some logistic companies using a manual queueing process by logging in the information of the delivery in a monitoring log sheet after that truckers given a queuing number and wait for their names to be called that sometimes trucker missed due to they rely only on the guard.The manual queue management system has no automatic feedback in giving a real time queue status.

The proposed system will help the customer service company in storing data of queue, giving an queue status update thru text message, monitor that will display the who's trucker is on queue and a dashboard that will show the analytical and graphical reports of queues.

PROJECT CONTEXT

Nowadays, customer service oriented companies like DHL Supply Chain -Avon facing difficulties of lengthy queues. These problems often occurred and it become worsen every 1st and 2nd week of the month when the high volume of delivery is usually experienced. The improper management of such queues will cause tension and stress among suppliers, clients and also to DHL's employees.

For DHL Supply Chain-Avon time is money. Time is a quantity that is non-renewable and continuous. Any process that saves time and space is considered vital in many operations. Time is very important,that need to be manage efficiently. Wastage of time in a queue is a big problem that needs to be resolved by the proposed system.

To overcome this problem the researchers proposed a new way of handling queue, an Automated Truck Queue Management System with SMS notification.A system that helps service provider to manage customer in efficient way. The system can ease the customer flow management which is useful for manager of the service provider and analytics provided by a queue management system allows to identify key areas that are in need of improvement.

PURPOSE AND DESCRIPTION

The aim of this research is to develop a queue management system that will help to ease the process of queuing. With the help of the new technology the encoding and storing of data becomes easy. The use of sms the queue number and the status of queueing will automatically send via text message. The monitoring of queues will display on the monitor.Data analytics can analyze the delivery history and improve the current operation that will minimize logistical costs. Applying business intelligence, this system also can develop a performance metrics and KPIs, and create computer models to predict delivery issues. As queuing management systems collects data, it should be put to good use. Reporting capabilities will contribute to better visibility and control over future logistics outcomes.

OBJECTIVES

- To implement a queuing management system that will improved queuing process.
- To increase service reliability, as customers are treated fairly and efficient.
- To monitor the queue’s using the live monitor updates and by using SMS notification.
- Producing statistical reports, it helps to determine process condition to develop actions for process improvements.
- To timely monitor the receiving status.

SCOPE

- Displays a real time update via monitor display.
- Supplier’s will be notified on the update via SMS.
- Automated dock assignment.
- Generates a statistical reports that can be use by top management.
- Timely monitoring of Receiving Status.

LIMITATIONS

- This study is focused on the system of DHL Supply Chain Phils-Avon.
- This study is limited to the users w/ provided user access that can modify or operate the system.
- The system does not support Radio Frequency Gun transaction & SMS notification is internet signal dependent.
- The findings are based on the output of this study and proposed as the Data Analytics & SMS Notification for effective truck queueing management system of DHL Supply Chain Phils-Avon.

Chapter II

RELATED LITERATURE

FOREIGN LITERATURE

According to Jhala and Bhathawala (2017), Smart Queue Management System for Banks, addressing the problem of long queues of customers faced by banks at peak hours followed by slacked periods where there were no customer entries. The study sought utilize the Short Messaging Service (SMS) intervention to optimize capacity utilization while saving the customers from having to queue by notifying them when they were up for service. The intervention eased long queues, reduces waiting times, enhanced productivity and motivation of the bank staff and enhanced the bank's image due to improved discipline at the premises. Weaknesses observed included the assumption that these customers would be in the close proximity to the branch in order to quickly receive service. Their proposed model is a small step towards easing out the life from the long queues in banking sector. The problem of waiting for ones turn to come in a long queue could be easily overcome by this project. It upholds the image of the banking firm as the queue system ensures discipline at the premises.With the help of queue management systems it will give comfort to our suppliers. The proponents adapt the idea of Short Messaging Service to use notification for our suppliers. Short Messaging Service help our DHL reduce the waiting time by sending text messages regarding to the status of their queue.

According to Farayibi (2016) Investigating the Application of Queue Theory in the Nigerian Banking System. The Study sought to examine the application of queue theory in the banking system in Nigeria. The study focused on GTBank and Ecobank and employed the multi-server queuing model to analyze queue characteristics in the banks and the performance measures (waiting and operational cost) to determine optimal service levels. In its conclusion, the study emphasized the relevance of queueing theory to optimize service delivery. Additionally, the study recommended server models to the banks management in order to reduce total expected costs while raising customer satisfaction. The study also urged the bank to view queue characteristics from the standpoint of the customers and vary number of service points according to queue characteristics, train staff on queue and motivate employees to improve productivity. The proponents got the idea of Automated Queueing and Multi Server as process in our proposed queueing management system. Now a day's automated queue management is very popular and its usually used in banks,hospital and airport to ease their queue process.

According to Sajid (2017) Data Analytics:How To Use Graphs To Present Your Data Smartly, in Data Analytics the numbers or texts or symbols represents a pieces of information. Using a number is easier to think that numbers has values of quantitative or qualitative variables. It must be taken note of that the term "values" is broad enough to cover everything to which value can be ascribed to. Variables have different types first, there is a quantitative variable also called numerical in the sense that it has a significant meaning as a measurement. A continuous variable is a specific kind of a quantitative variable that describes data in a measurable way. If your data deal with measuring a height, weight, or time, then you have a continuous variable. Here there is the interval, and within this interval, any value can be possible. A discrete variable has a finite number of possible values and does not have the inherent order. Second, there is what is called a categorical or qualitative variable. This is not subject to any quantification because it is descriptive or labels. It describes what it means to measure.Descriptive statistics is the idea of presenting and describing the features of your data. It can be done through various means: graphical representation, tabular representation and summary statistics. First two are called visualization technique. For better understanding the dichotomy of the presentation, it is better to tackle the overview of descriptive vs. inferential statistics. Descriptive statistics are used to present quantitative descriptions in a manageable form. This is a way to see something meaningful of data at hand. In short, you make a statement based on, about, and derived from these data. As a limitation, you are not allowed to make conclusions beyond the data at hand. You cannot make inferences or generalizations. In this system the proponents got the idea of Graphical Presentation or Tabular Presentation and summary statistics for our proposed system that we can use in data statistics as a guide of DHL managers in making decisions by the use of graphs as a visual representation of datas.

According to Silva (2016) An Integrated Queue Management System is using a numbered tickets, served by ticket dispensers, are probably the simplest existing technology for managing waiting lines. A staff member operating a queued service can simply call the next ticket aloud and register the last called number. Alternatively, this same operator can press a button that makes a speaker signal the call and a LED display to show the ticket number being called. This latter example is representative of the current queue management systems deployed in the academic services at Instituto Superior T´ecnico, where only one queue is formed by the customers, independent of the issues they might want to solve. With this current technology the waiting customers have no way to know when their ticket is about to be called. They can only look at the current number in the LCD display, and make an educated guess by watching its progress, not being automatically of the current average time estimate. Likewise, there is no means of providing automatic feedback to the staff about current queue growth or about the effectiveness of their queue operation, based on tickets dispensed and customers served. No record of the overall activity of the services is taken. Recording service activity data in the long run is useful to detect patterns in the service operation, like periods of higher affluence of customers, periods of lower service efficiency or any others patterns that might be found. By having this bulk data, its analysis could pinpoint the weak points of the system and where to act in order to improve the service. With the goal of improving and modernize these services, a queue management product, the SIGA System, is designed and implemented in this work. It provides more information to both customers and staff while keeping record of all service related activities.

After requirements gathering, a server-based solution is proposed, recording all service operation. It provides administrative web interfaces enabling back-office operation for the staff and a queue progress display for the clients. An electronic ticket dispenser connected to the server and configurable by staff shows one or more queues and the respective average time wait for each. Clients obtain a ticket from that dispenser by selecting the queue which best represents their unsolved issue. In a similar way, a client can obtain a virtual tickets from a mobile device, if connected to the internet. In this system, proponents adapt the idea of queue ticket numbering as reference of queue but in our system we used SMS we send the Queue number once the guard done in encoding and detailed information to be display on monitoring screen for us to monitor the real time update on the status of queue.

Al-Barhamtoshy (2014) A Data Analytic Framework for Unstructured Text the Big Data is defined as the amount of data to store, manage, and process in effective manner. Such process includes robust analysis of the data , and capability of tools used in analyzing the data. Accordingly, to the technology capabilities; tens hundreds of terabytes storage are needed to handle big data. Therefore, big data and related analysis are very essential in modern science and business. Such big data are generated from audio, videos, images, event streams, logs, posts, search queries, health records, social networking interactions, online transactions, emails, science data, sensors and mobile phones and their applications. This growth rate is very fast exceeding design capability to handle data effectively and also extract relevant meaning for decision making. Many of governmental sectors emphasize on-how big data create “value” in different domains and across discipline fields. Structured, semi structure and un-structure data will continue to grow. Consequently, different organizations are challenged to manage the big data they have. For example, from the medical field; Fox illustrates how patients medical data record and current health situation are used to plan and predict patient participation in wellness and disease management systems. Therefore, doctors should collect and analyze patient’s data using such systems. Accordingly, many of cloud services require users to share their data like health records for data mining and analytical process, taken into consideration privacy and security. Internet Data Center (IDC) announced that the global big data will increase by 50 times next decade. According to IDC they defines big data as “anew generation of technologies and architectures, designed to economically extract value from very large volumes of a wide variety of data, by enabling high-velocity capture, discovery, and analysis”. This definition includes “three main characteristics of big data”: volume, velocity, and variety. In this system, proponents adapt the idea of handling big data that we can use for analyzation of data that will be stored in our proposed system.With this data we can analyze and interpret it into graphs that wil help our users in making decisions.

LOCAL LITERATURE

According to Austria (2015) Queue Management Practices of Quick Service Restaurant, the problems regarding waiting line in quick service restaurants (QSR) has been one of the main concerns of industries and scholars nowadays. It is because people today demand not only for quality food but also for speed. Quick service restaurant players explore on the approaches to optimize the efficiency of restaurant management. One important area that defines how well and efficient a fast food restaurant delivers its product and services to customers is its waiting line (queue) management practices. The study was conducted at Lipa City, involving five popular quick service restaurants named by the researcher as QSR A, B, C, D, and E. It made used of 363 customer respondents proportionally obtained from five restaurants. It intended to assess the extent of implementation of the queue management practices of the restaurants and the level of satisfaction of the customers in such practices in terms of customer arrival, waiting line and service facility. The findings revealed the queuing system used and the waiting line structured utilized by the restaurants. The extent of implementation of the queue management practices in the three areas mentioned of the five QSR's was presented comparatively. Likewise, the level of customer's satisfaction on the queue management practices was also determined. Significant difference in the extent of implementation and in the level of customer satisfaction were determined if the respondents were grouped according to restaurant’s profile. Recommendations in the improvement of queue were given based on the findings.The use of Automated queue management system in the restaurant affects the level of satisfaction to the customer.Automated queuing number system is preferred by customer’s over the manual queuing system. In this system proponents got an idea by Automating the queueing system will make the process faster base on this study and by the use of automated system it will boost the satisfaction both of DHL and it’s suppliers.

According Miranda et al. (2019) A Dental Advisories and Sms Notification System Utilizing Predictive Analytics the development of Information Technology has been advisable for all establishments and enterprise organizations as nicely as the clinical field. With the use of superior technology, transactions, processing, and operations grew to be faster, correct and useful. This study focuses on the efficiency of advisories and SMS notification, especially to the Dental Industry. The system goals to ease the manual tracking of patients, reducing the time of waiting for patients and increasing the number of patients served in unique Dental Clinics. The system is ubiquitous and efficient as it is online and gives a paperless application that can without problems discover a patient list and preceding transaction. The device is successful of rapidly notifying the sufferers regarding the popularity of their session the usage of SMS Notification and has decision support function that can provide a useful resource in terms of deciding on fabulous sales and advertising strategies for the company. The system also makes use of Predictive Analytics to furnish choice help in terms of Sales strategies for the business. With the help of predictive analytics proponents can easily identify on the process that needs an improvement, also the use of analytical data as business decision.

According to Malala, Nieles and Villanueva (2014) A Queue Management Systems Characteristics, Advantages and Application. Government billings queues are stress-givers for a huge number of industries. To avail a service or a product, the customer must wait for his turn for him to be served. It creates a sense of annoyance to anyone waiting in a line or standing up in queue. Several problems concerning the queuing management still exist up to this date. There are reports that some government offices are having slow queues and rendering slow service to the public. In particular, the Land Transportation Office (LTO) is one of the slowest in terms of the processing service. This dilemma pushed the office to request for additional funds for them to upgrade their service. Even though queuing existed in the past and in the present, numerous ways are being developed to further innovate the system and decrease the total waiting time of customers. Several trials have been made to minimize stressing queues and low client satisfaction ratings, but as time passes by, this situation has changed with the help of technology and computer-based systems that use different programming languages and more powerful devices, giving birth to what is now called queue management systems.The identity of a customer is captured and a customer reference will be provided upon the entering of the customer in the establishment. Content may then be associated with the customer reference according on the customer's identity and the content is displayed to the customer when the customer is called to a service position. With the presence of queue management systems, any customer-orientated institution, like banks, government offices, hospitals, telecommunication companies, and restaurants may be able to widen their horizons and further develop their waiting line system that meets the demand of their customers.This research intends to discuss effectiveness of queue management systems with regards to processing speed and convenience of applicants, its maximum potential and limits on how to improve the system itself and decrease delay time, and the impact of its application in-operations and customer flow in offices. In this study, proponents got the idea of the effectiveness of automated queue management system with regards to processing speed and convenient to the users.

According to Abdullah (2016) A Queue Management System for STI Iligan Registrar. It is better than good for STI Iligan to make use of Queue Management System that is computerized in order to achieve higher efficiency in assisting for queuing while proving a better service. It is accommodating number of students now and will be accommodating more. So processes of transactions must be organized and the time must be spent productively. Queue Management System is a programmable system used to control queues. It improves productivity and reduces students waiting. It has the ability to manage and streamline queues in order to improve students waiting periods and staff productivity. Also, it provides comfort as well as fairness to students by allowing them to maintain their position in the queue while they are seated comfortably or engaged in constructive activity. In this system, proponents got the methods of First In, First Out for the proposed automated queue management system

RELATED STUDIES

FOREIGN STUDIES

According to Woottichaiwat (2015) Efficiency Improvement of Truck Queuing in the Freight Unloading Process Case Study of a Private Port in Songkhla Province. In his study it aimed to improve the efficiency of queuing system of supply trucks in freight unloading process in a private port in Songkhla province. In everyday, suppliers delivered 5 types of supply to the port, including, oil-rig spare parts, consumer products, tools and chemicals, helicopter fuel, and plastic bins. Each truck consumed unequal operation time. The operation process was improved by method analysis and ECRS technique. Then, several models of truck scheduling were compared using Monte Carlo simulation technique. The models composed of first-come, first-served, shortest process time, longest processing time, and analytic hierarchy process of both original and new improved processes. The result showed that New Longest Processing Time was the most efficient model, which reduced 77.2% in mean waiting time, decreased 69.0% in mean total time in system, and improved from 11.94% to 35.30% in system utilization. With this study we the model of first come first serve in our system. In this study proponents got the idea of Data Categorization based on its information and Analytical Hierarchy Process (AHP) as a basis of daily improvement of data.

According to Pad-berg (2015) Big Data and Business Intelligence. A data-driven strategy for e-commerce organizations in the hotel industry Modern organizations do not only want to know what happened and why it happened, but also want to know what is happening right now and what is likely to happen next. Since organizations hunger for these insights and the adoption of the World Wide Web, the generation of data and collection speed has increased exponentially. The demand for all this information and all these rapid technological developments enabled organizations to capture, store, and analyze large amounts of data. With the help of these rapid developments more organizations are shifting their focus to exploring and exploiting all this data. This phenomenon is called "Big Data" and is identified on the emerging technology hype cycle as one of the biggest IT trends of the last few years. In this study, proponents got the idea of big data this will help our system to analyze data's came from our daily suppliers to decide for further decision making.

According to Perry Kuklin (2013) Breaking Down Big Data for Better Queue Management. It's the buzzword to describe the immense volume, velocity, and variety of information that continues to build worldwide at a minimum annual rate of 59%. Big data clearly presents an opportunity for those with the ability to decode and understand it. In the retail space, a 2011 McKinsey Global Institute report estimated that retailers using big data have the potential to increase their operating margins by more than 60%. On the other hand, it also threatens to bog down and leave behind those businesses lacking the means to handle and envision its value. A traditional view of big data is the “3 Vs,” put forth by Gartner’s Doug Laney in a 2001 research report, referring to the volume, velocity, and variety of data. But there’s a more current view of big data: the “3 Ws”– What, So What, and Now What—a view that highlights the need for decisions to be made and actions to be taken from big data. The business challenge ahead is to break down big data into actionable "chunks" from which to make decisions and improve the customer experience.

When it comes to queue management, big data is no stranger. Retailers and other service providers have unprecedented opportunities to capture data relevant to the waiting and checkout experience. Current virtual queuing, electronic queuing, and video analytics systems enable businesses to track all aspects of the customer journey. From the moment of registration all the way through checkout, businesses are given valuable insight for optimizing customer throughput, increasing service efficiency, and enhancing direct customer communication, ultimately providing the customer with a better shopping experience. Today's technology captures metrics such as average wait times, queue length, agent/cashier idle time, staffing allocations, customer arrivals, traffic patterns, etc. In this study, proponents adopt the idea of using Big data in queue management in making decisions. Based on the study Big Data was already using in different types of business to improved performance and customer experience.

According to Ngorsed & Suesaowaluk (2016) A Hospital Service Queue Management System With Wireless Approach the innovation of technologies could bring support to the quality of life for human in various aspects and objectives. However, in order to apply and implement technology system to be used requires the costly investment for itself. This constraint leads to the inescapable archaic management methods, and the systems still coexist alongside the advances in procedures. One of the unavoidable significances is the hospital service for the people, especially among the undeveloped country and developing country. The public hospitals likely support the poor and middle classes which have to patronize the public services in the state hospitals. A growing population base will continue having a pressure to the existing hospital facilities. With the cycle of limited facilities, it leads to the coupled staffing shortages which will guarantee that long queues to remain synonymous anytime visiting a hospital and other public service facilities. The people must take a queue as long as they need the services. Whether the problem is caused by staff shortages, equipment shortages, or the hospital capacity is not sufficient for the population area they serve. Long queues are an unwanted and unnecessary burden to the public as well as the hospital staffs. Long queues are then associated with a negative image of the hospital experience, but most people can’t avoid to be under this present system. For this project, we propose the system with the main objective as to create a visual queue for hospital online where people can access and reserve their queue wirelessly over the Internet. The system allows people to monitor their queuing status from the web service application. This beneficial system is designed to offer the options for people who are waiting for the service; they can go anywhere while they are in the queue rather than standing and presenting themselves in front of the service area. In this study, proponents got the idea of integrating of visual queue that will display real time updates of the queue. Using LCD monitor trucker can monitor their queuing status and has a visibility for them who's on the queue and who's next in queue.

According Udin et al. (2016) An automated queue management system is a system that helps service provider to manage customer in efficient way. The system can ease the customer flow management which is useful for manager of the service provider. The purpose of this project is to develop an Automated Queue Management System for organizing queuing system that can analyze the queue status and take decision which customer to be served first. This project focuses more on the banks queuing system, different queuing algorithm approaches which are used in banks to serve customer and the average waiting time. This queuing architecture model can switch between different scheduling algorithms according to the testing result i.e. the average waiting time by using two different queue control systems, which have developed. There are several process undergo, which control by Intel Galileo Micro-controller that is software-compatible with the Arduino software development environment. Finally, the systems have been tested under different conditions to evaluate its performance. In this proposed system we got the idea of using a controller for queueing management system. The system will help the receiver to organized who will be serve first base in the arrival time of the suppliers.

LOCAL STUDIES

According to Veluya et al. (2015) Queueing Systems to Minimize the Waiting Time during Enrollment in Southern Luzon State University. The mathematical analysis of queues suggests ways to shorten the waiting time and the waiting line which are the basic characteristics of a queueing system. The study of queues deals with qualifying the phenomenon of waiting lines using representative measures of performance, such as average queue length and average waiting time. Saaty (2012) stated that a waiting line or queue involves arriving items that wait to be served at the facility which provides the right service they seek. If the line is long, customers may become impatient and leave, thus causing a loss of profit. The owner of the facility may decide that investment in another check-out counter is worthwhile because its cost is offset by the profits taken from the impatient customers, more of whom now remain to be served. In this proposed system proponents got the idea of processing multiple data in a single process and to use dynamic servers to exceed their limited servers.

According to Lumauag (2016) School Event Notification through SMS. The user of mobile-cellular subscription is almost as many as people in the world with 6.8 billion total subscriptions with more than half in the Asia-Pacific Region which corresponds to 3.5 billion. Mobile devices are becoming popular in use nowadays and Philippines is considered as the texting capital of the world with over 1 billion text messages sent per day. SMS have been developed with different demands and create a new approach for interaction and communication. As we know SMS alert system is useful in some cases for delivering alerts in emergency and there are some applications are developed as pre-disaster warning devices. Educational institutions are expected to extend their services through mobile phones by providing easy access of information to students, teachers, as well as parents. The development of School Event Notification through SMS offers a remarkable improvement of communication between students, parents, teachers, and school through timely delivered SMS messages. The application of this feature will help to ease the calling of suppliers, truckers and vendors.With the help of SMS notification will keep them inform on the status of queues.

According to Tico (2016) An Online student portal is one of the essential parts of a school. As technology evolves alongside, the information revolution, the power of computer makes it easy to distribute and view information needed with just a few clicks.The School Portal System with SMS Notification is web based that will manage all the news and events in university and notify the students and professors or faculty staffs through SMS. The application will help our system to view data and reports online. With the help of dashboard manager can view reports and graphs that can be used in business decision.

According to Cosidon (2016) Student Information System for Kalinga State university .The student information system is an integral part of this technology. This student information system handles every aspect of student data right from admission, class schedules, subject enrolled by the student, overall student performance, and personal information of student. All these elements are integrated into a single database, accessing and tracking data of any student happens with just a click of the mouse! The backbone of the society, the nation is undeniably the education system. The Student Information System is an integral part of this technology. This Student Information System handles every aspect of student data right from admission, class schedules, subject enrolled by the student, overall student performance, and personal information about the student. All these elements are integrated into a single database, accessing and tracking data of any student happens with a click of the mouse! The benefits of Student Information System intuitive user interface with pioneering features. Maximize school management parent’s communication. Smart management of student and staff data makes chaotic schedules, stress-free and easy to manage. Enthuses student performance and success streamlines and simplifies everyday administrative tasks. Enhancing the efficiency of school administration and managing student data is effortless and easy with the Student Information

System software. This system can be customized to include a whole range of activities. It can be easily accessed anytime. Schools can run the Student Management Information System on minimal hardware affordably and gain a competitive advantage of exploiting the latest in technology staying ahead in competition.It this in managing the the information in the system. In this system proponents got the idea of Role-based Security, Data Storing, Data Reusability, Data Maintainability, Data analysis as an application for data that we will use.

Abapo, Torregosa and Tomboc (2017) A UX Case Study: Rose Pharmacy Customer Queue Management System. Rose Pharmacy is one of the top pharmaceutical retailers in the Philippines today. With over 252 branches all over the country, their goal is to provide easy access to quality and affordable medicine to customers. The pharmaceutical retailer's corporate philosophy is focused towards fostering a customer-centered culture, as illustrated in their KSM (Kay Sarap Magmahal) membership card program. As part of their efforts to embody their customer-centric philosophy, Rose Pharmacy has a customer management queue system in place on selected branches. Essentially, its purpose is to manage customer flow in their physical stores. The customer queue management system benefits in two fold. It benefits customers because it reduces their perceived wait time. They'll have a more accurate idea of how long they would have to wait before a staff member will attend to their needs. Also, shorter perceived wait time is associated with more relaxed, happier customers. It benefits Rose Pharmacy employees by ensuring smoother operations through a more organized and centralized flow of information. There researcher observe in one of her visit at Rose Pharmacy Ayala Cebu Branch that Korean couple were having issue using the queueing system. Based on this observation proposed to conduct a design research.They want an answer for the questions. How are existing customers using Rose Pharmacy queuing system,Are users having usability issues with the system?If yes, what are these usability issues and how can we help solve them.T o help them in making the design they use the Stanfords school design thinking process. Make a prototype for testing of usability test. With the use of Stanford and design thinking process and Alan Klemen format for testing. In this system we got the idea of user friendly User Interface(UI) for easy control of queue monitoring, their user group prioritization for data queue and multiple data per user as we do multiple document per user.

Chapter III

TECHNICAL BACKGROUND

HARDWARE

Figure 1: Personal Computer with Windows 10 Operating System.

To perform the specific function of a system proponents need a physical hardware such as computer system that includes a compatible system requirement for other software to be used, composing a keyboard, mouse, Monitor and a System Unit.

Table 1: Minimum requirement specs of suggested Computer Set.

Specification	Details
Operating System	Windows 7 or Higher
CPU	Intel Core i3 or Higher
CPU Speed	2GHz or Higher
Display Adapter	Built-in

Network Adapter	100/1000Mbps
Memory	4gb or Higher
Hard Disk Space Required	50GB
Monitor	Any
Keyboard	Any
Mouse	Any

Figure 2: LCD Monitor

To show the output process of the Queue for the designated dock location

Table 2: Minimum requirement specs of suggested Monitor.

Specification	Details
Screen Size	32 in. Or Higher
Resolution	1366x768 or Higher

Aspect Ratio	0.672916667
Dynamic Contrast Ratio	125.0006944
VESA Mount Compatible	Yes
Connectivity:	VGA
	AV IN
	AV OUT
	S-Video In

Figure 3: Mobile Phone

Mobile Phones receives a message, a notification where you're about to dock and unload, it includes information of a specific Dock or Door assigned to a Driver or Drivers' Assistant in a given number or ticket. Any mobile phone with Active mobile sim that transmits signal is a good one.

SOFTWARE

Figure 4: Visual Studio 2015 (Version 14.0.25431.01 Update 3)

Microsoft Visual Studio is an integrated development environment for developing with Visual Basic.NET, it has a Graphical User Interface that makes more interactive for the user. This software serves as the main user interface that the functions is to develop the main functions such as data analysis, input and output commands, computations and reports.

Figure 5: Microsoft SQL Server 2012 (Version 11.0.7001.0)

Since the system has a database, the developer uses Microsoft SQL Server, Microsoft SQL Server is a relational database management system developed by Microsoft. As a database server, its primary function is to store and retrieve data as requested by software application which may run either on the same computer or on another computer across a network. The proponents want to develop a friendly user interface to be easily access by the users.

SYSTEM ARCHITECTURE

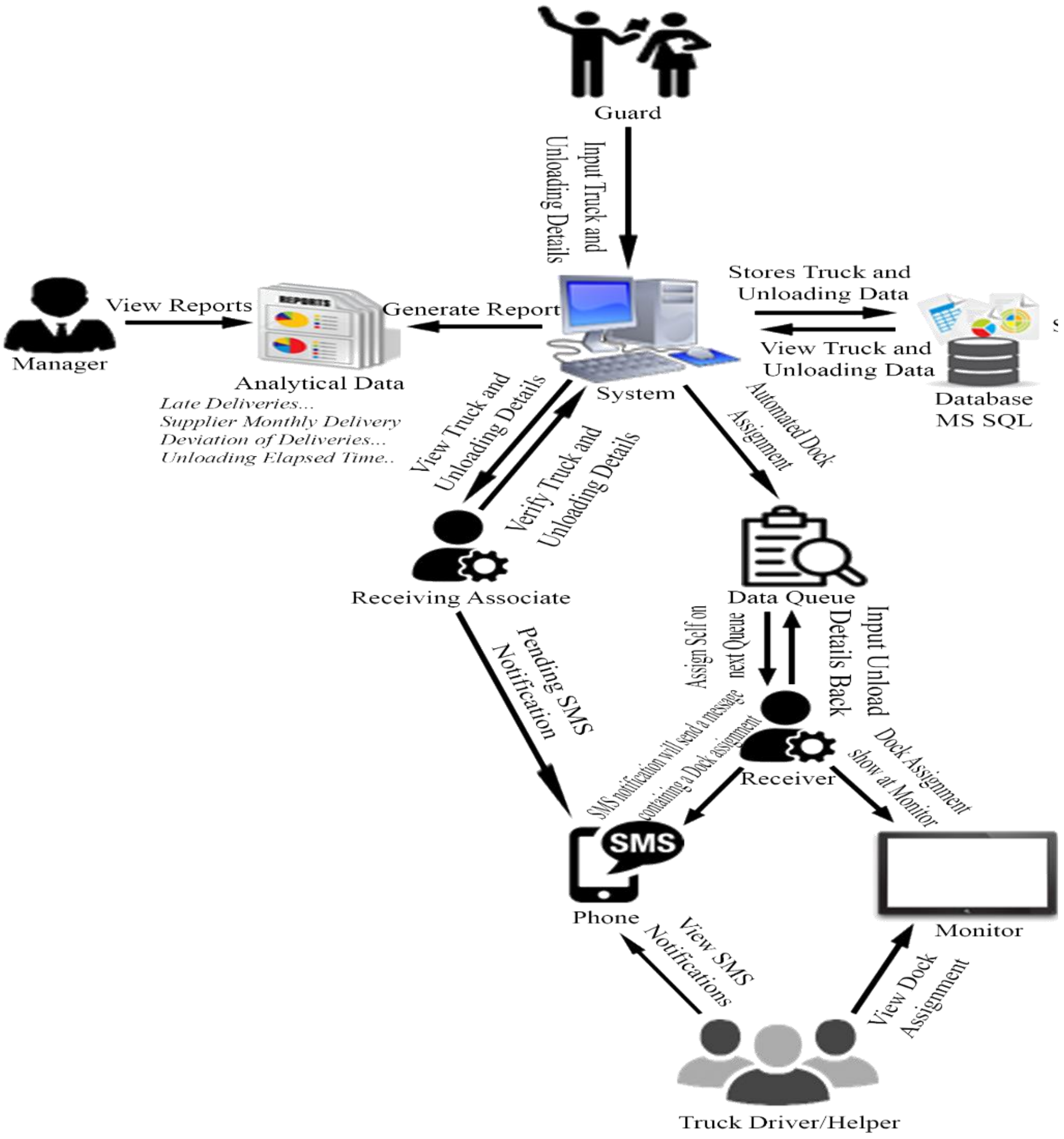

Guard
Input Truck and Unloading Details
Manager
View Reports
Generate Report
Analytical Data
Late Deliveries...
Supplier Monthly Delivery
Deviation of Deliveries...
Unloading Elapsed Time..
System
Stores Truck and Unloading Data
View Truck and Unloading Data
Database
MS SQL
View Truck and Unloading Details
Verify Truck and Unloading Details
Automated Dock Assignment
Receiving Associate
Data Queue
Assign Self on next Queue
Input Unload Details Back
Pending SMS Notification
SMS notification will send a message containing a Dock assignment
Receiver
Dock Assignment show at Monitor
SMS
Phone
Monitor
View SMS Notifications
View Dock Assignment
Truck Driver/Helper

Figure 6: System Architecture

Figure 6. The diagram shows how the system works from the start until the last process. The proponents develop a system that can organize and

manage the queue of the trucks that would be unloading in a limited docking location whereas it can also generate specific output analytical report that shows how fast or accurate their job goes using the Microsoft Visual Studio integrated development environment software figure 4. It also generates different tables and charts that includes the specific data that symbolizes the given output results of a given generated reports. Receiving Associate as a user can view and verify the data that has been encoded by the guard whereas the system can send a notification to the driver if the document where the data encoded by the guard being taken has a problem. Pending Queue is the cost of a problem if their document can still re-verify and proceed to unloading queue. Data that encoded by the guard is the basis of the dock assignment for unloading. Receiver as another user has the controller that can assign their self for the next queue in process, view, input unloading information, prompt a call to the monitor, output the assigned dock to the monitor and prompt a SMS reminder for the driver and its helper.

Flow Chart

Existing Flowchart

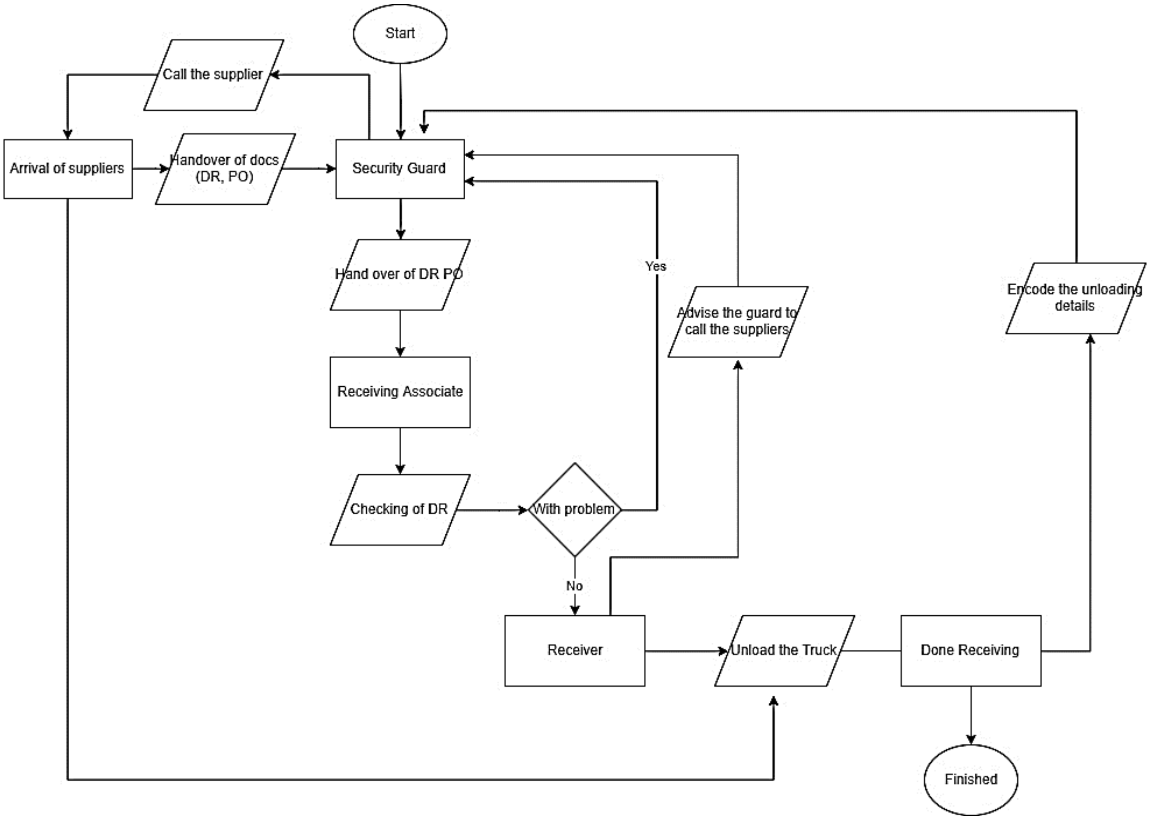


Figure 7. Existing Process Flowchart

In this flowchart shows the existing queue process of DHL Supply Chain.

Proposed Flowchart

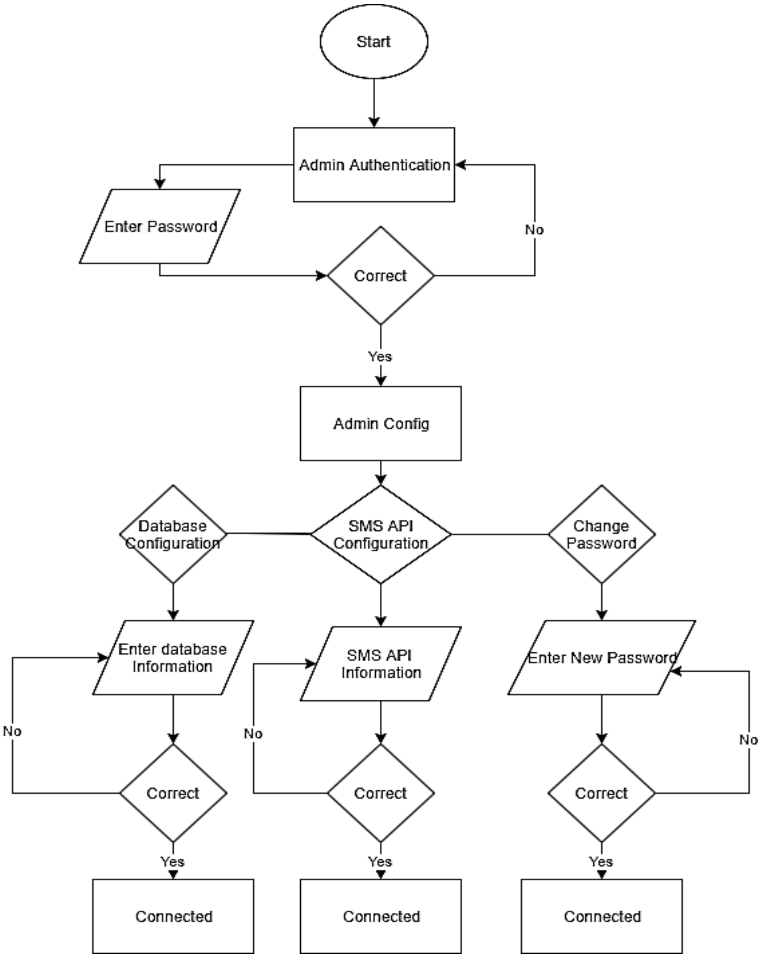


Figure 8. Admin Config.

In this flowchart show the process of Admin Configuration will you can set the Database connection, the SMS API Config and Changing the Password.

Admin Panel Flow Chart

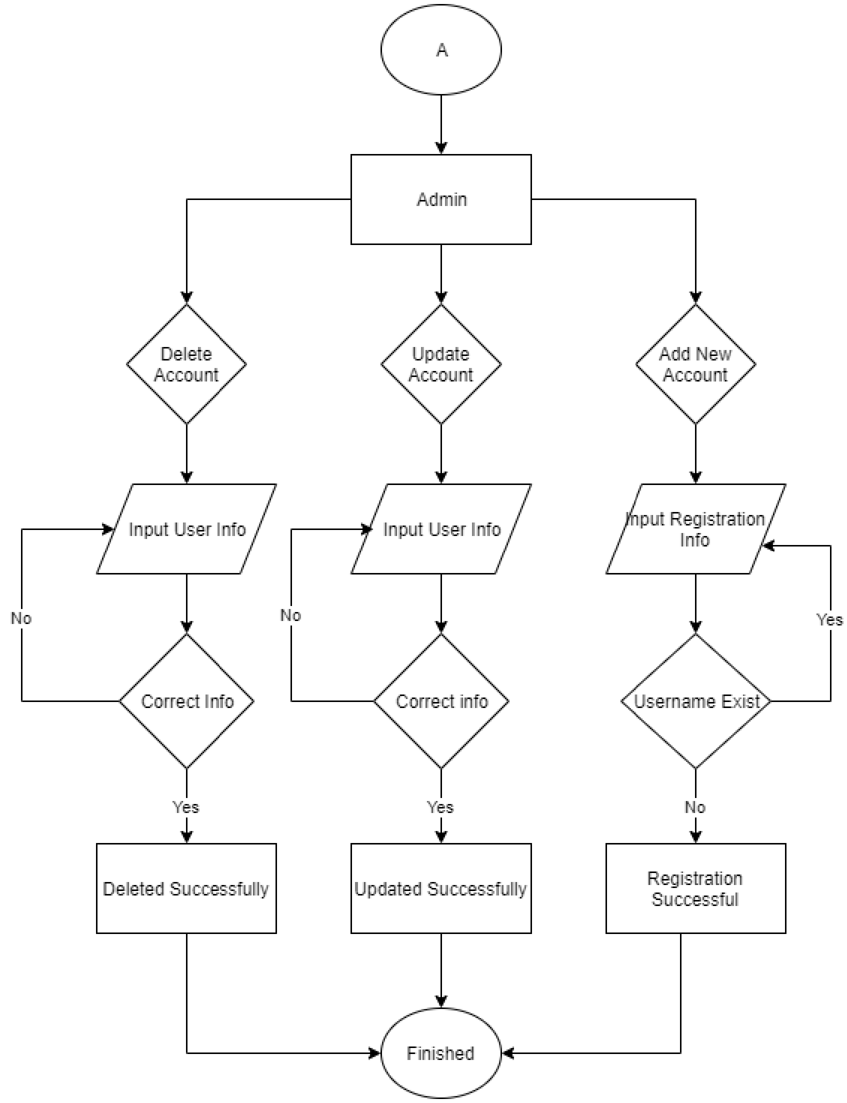


Figure 9. Admin Panel

In this flow chart shows the process of Adding, Updating and Deleting of users.

Log In Panel Flowchart

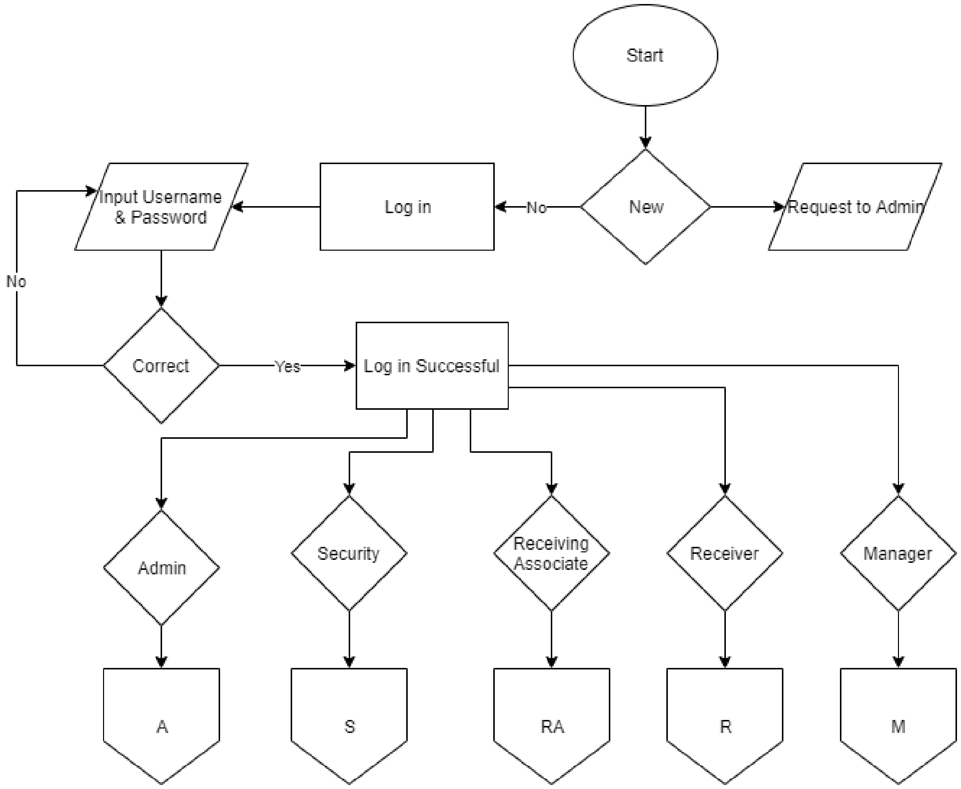


Figure 10. Log In

In this flowchart shows the process of Login. User can access the panel according to the user access. There are five types of user access Admin that handles the user maintenance, Security that handles the encoding of the delivery, Receiving Associate handles the verification of documents, Receiver handles the time and total qty received and Manager can view the reports.

Security Guard Flowchart

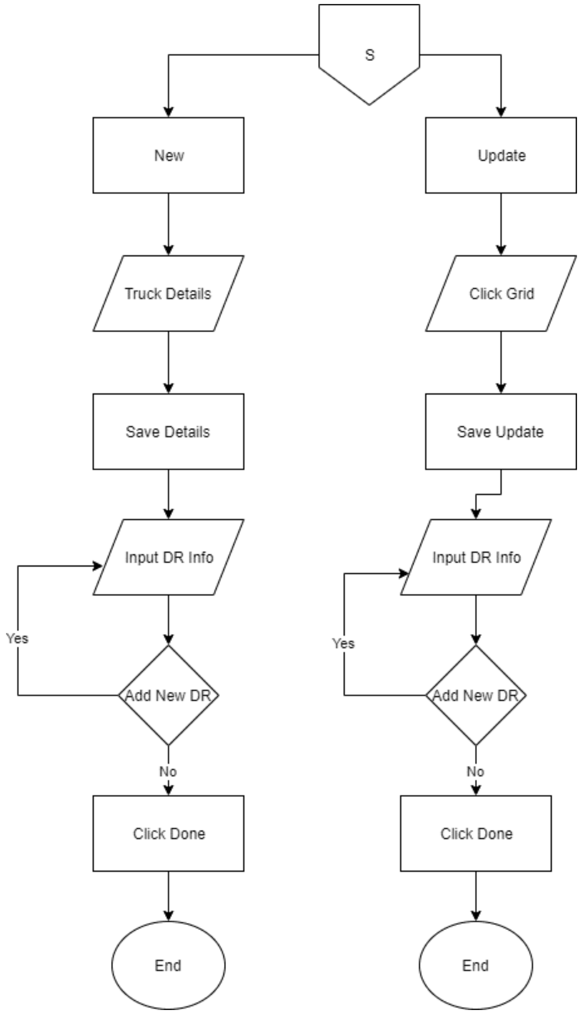


Figure 11. Security Guard’s Panel

In this flowchart shows the process of Security guards in adding new records of deliveries and updating of data entry.

Receiving Associate’s Flowchart

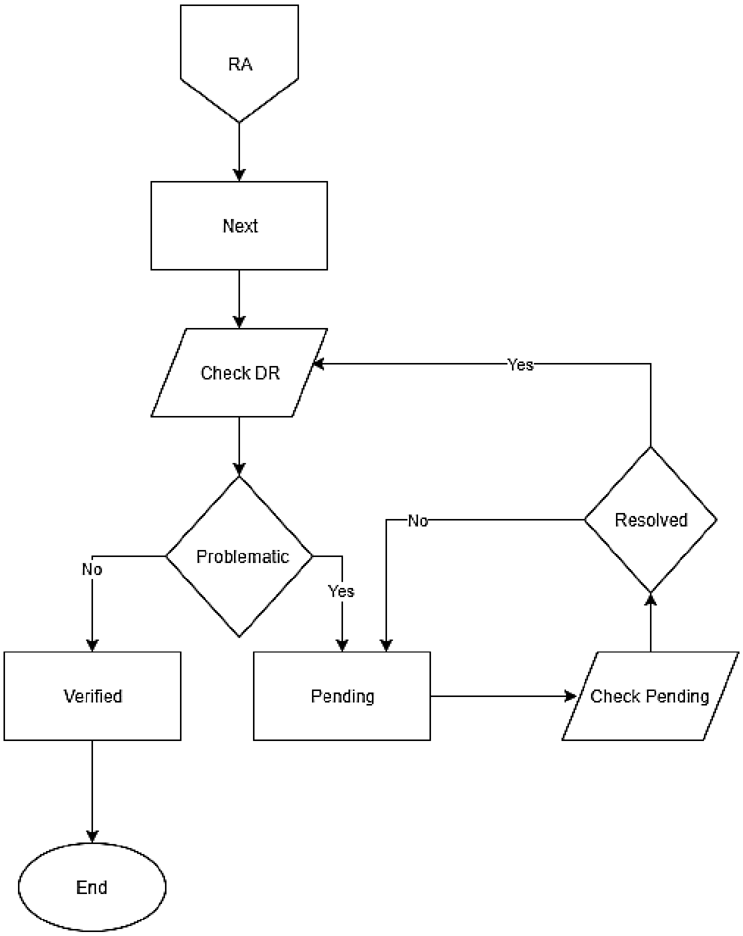


Figure 12. Receiving Associate’s Panel

In this flowchart shows the receiving associate process, the receiving associate check the DR’s that encoded by security guard. If it has a problem DR will go to pending.

Receiver’s Flowchart

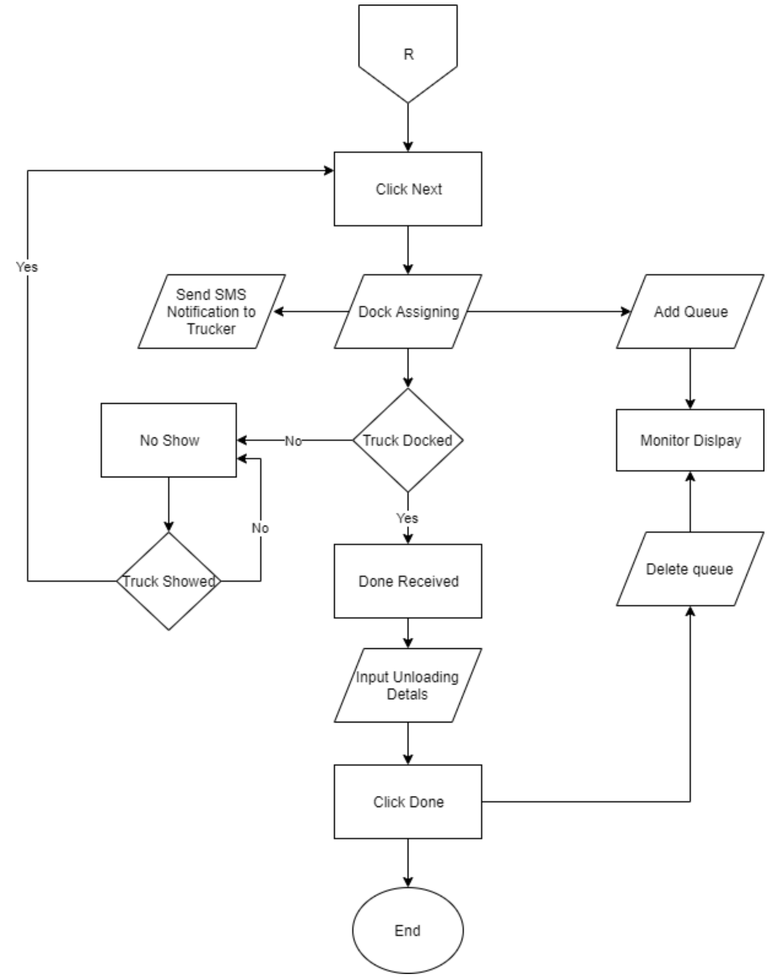


Figure 13. Receiver’s Panel

In This flowchart shows the process of receiver, the receiver click the next button then system will assign a dock depends on the category once done unloading the receiver will input the unloading details.

Manager’s Flowchart

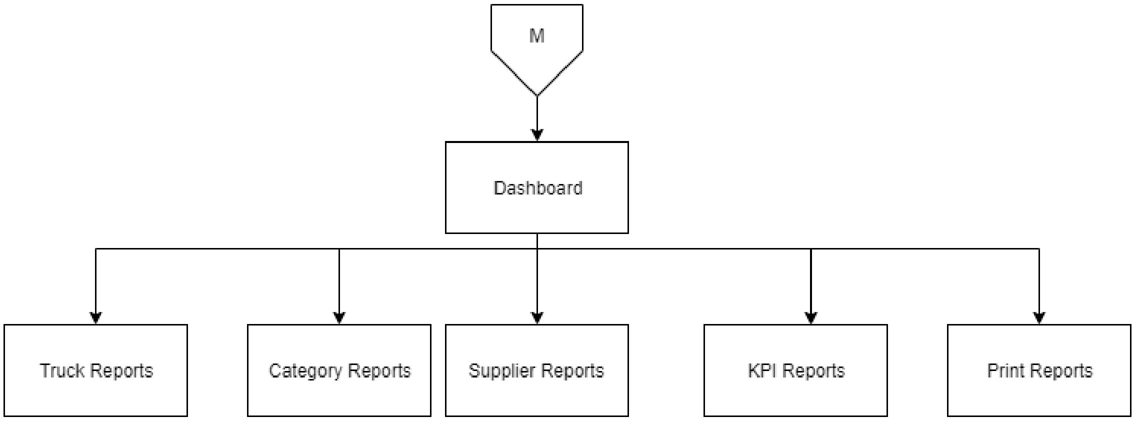


Figure 14. Manager’s Panel

In this flow chart shows the process of Managers panel, Manager can view report such as Truck Report, Category Report, Supplier Report, KPI Report and also can print the reports.

Data Flow Diagram

The DFD Level 0 is the basic process of Truck Queue Management System.

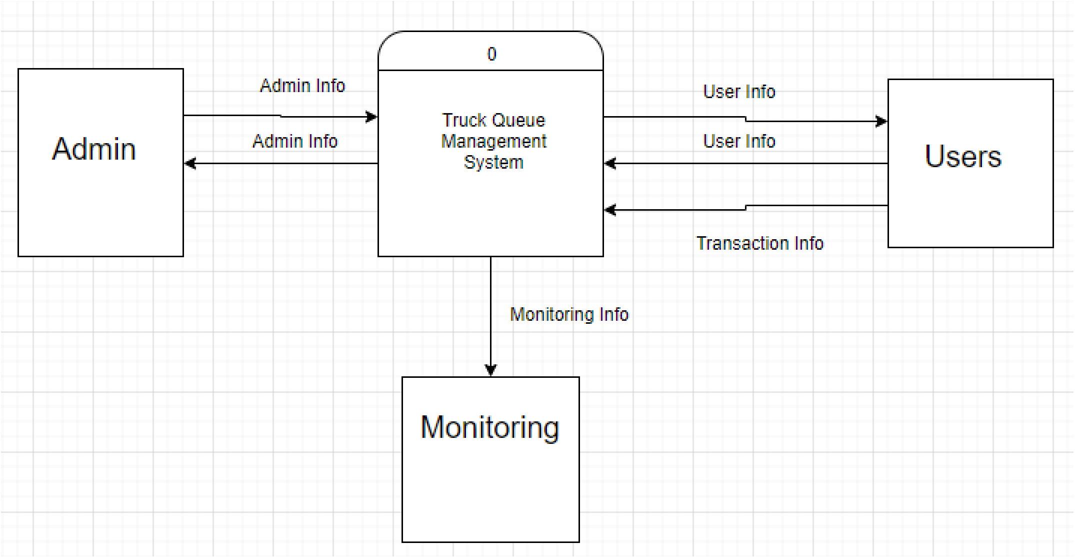


Figure 15. DFD Level 0

The Admin stands to monitor the users and grant a user access base on the request of manager. The User is the person who will input the information and can retrieved the information.

The DFD level 1

The DFD level 1 is what will be the process Queue Management System.

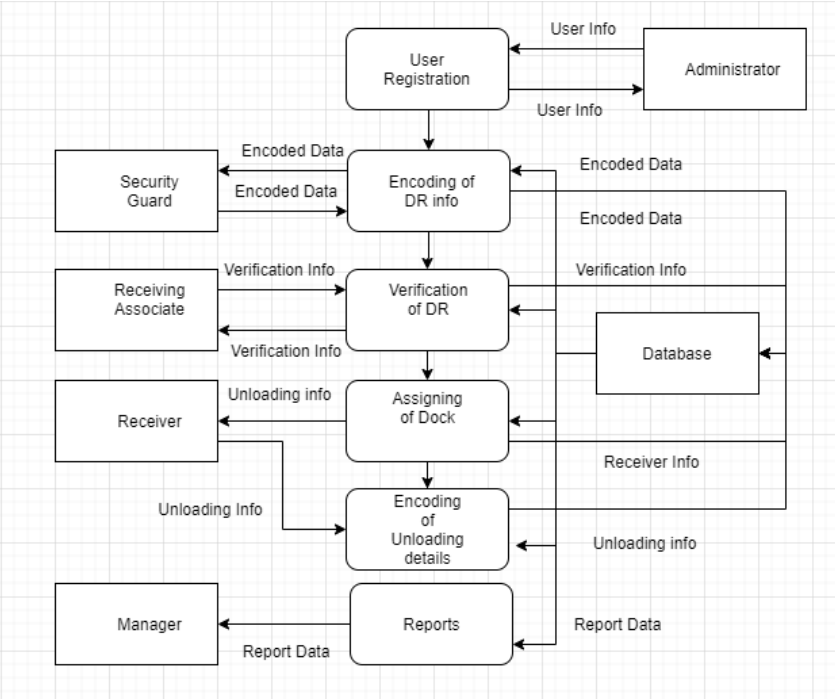


Figure 16. DFD Level 1

The admin is the one who can register or create add, edit or delete of users. Security Guard will encode the DR info and save it into the database. The data that encoded by guard will verified by Receiving associate.The system will assign the truck building and door number for the unloading and the receiver will encode the unloading details.The manager can view report on the daily basis.

Design and Methodology

Method and Research

Developmental Research

In this study, the developmental research approach was used. In which research was done simultaneously and combined in the developmental process, allowing for a more agile and efficient work. The research methodology used was the Agile model which made the development progress faster which is outlined in this chapter. In the following sections, the purpose, sample selection, methodology and procedure for data collection is explained.

Conceptual Design

The researchers used Agile modeling as the methodology for this study. Agile model is a combination of iterative and incremental process models with focus on process adaptability and customer satisfaction by rapid delivery of working software product. This involves ongoing planning of the project, testing, and integration of a new or upgraded software or product. This model works well for it assures that quality of the development is maintained. This methodology will help the researchers as it embraces change and adaptability, and aim to deliver working software continuously, rather than at the end of the project.

Research Methodology

Refers to the development of a system or method for a unique situation. Today the term is most often applied to technological fields in reference to web design, software or information system design. Agile methodology is known for its flexibility, so it’s essential for any development just like in this research that focusing on improving the overall functionality. And allows for changes to be made in the research development requirements even after the initial planning has been completed.

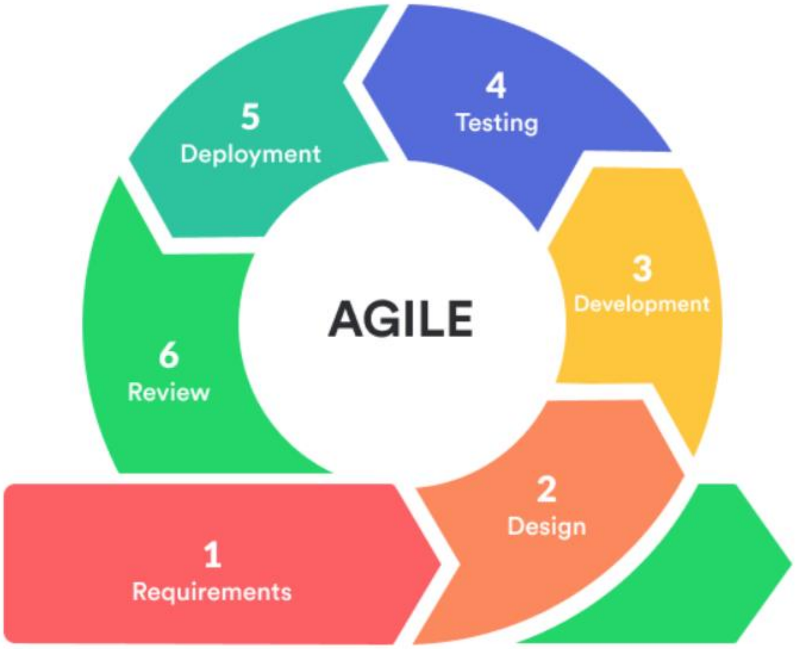


Figure 15. Agile Development

Agile Model is the model that are used to show the process or cycle in this system development. This research start at the left side and goes around the circle. The advantage of using agile model was we can easily determine the process or cycle of the system while the disadvantage was agile model should be detailed and organized.

The purpose of using agile model shows the flow of the system similarly the proposed system which is the development of Truck Queue Management System for DHL Supply Chain.

In requirement phase proponents are thought about what is the suitable software to be used in developing the Data Analytics and SMS notification for Effective Truck Queue Management System and also the materials to be used for queue numbers and for monitoring of queues.The proponents decided to chose Microsoft Visual Studio.NET the tool in developing the system due to it's advantages for the developers such as other built-in tools, like a form designer, which is useful when building GUI applications. Microsoft SQL Server as the database server due to its primary function of storing and retrieving data as requested by other software application which may run across the network. We also get other information such as drivers contact number to be use for sending updates through SMS notification, truck type, weight per box, quantity per box as basis of computation for KPI, category of items to be use for dock assignment, total number of dr as basis for employees target performance in encoding and checking of delivery receipts, count of the helpers in unloading of truck that will be use in data analytics as a part of management business decisions, supplier name to identify the number of deliveries and other data such as driver name, plate number for security record.

After the proponents gathered all the requirements that they needed, proponents continue to the planning phase as per interview were currently using a manual queuing system from it, hence it come up with the idea of automating the system. It plan to adapt the idea of SMS notification to notify or to inform the trucker even if it out of the DHL vicinity. Proponents also plan that the system that will automatically assign specific dock for each category and include Data analytics that will help the management in business decision.

In designing phase proponents made a design that will ease or shorten the process of queuing, proponents made the writing of information in the guard into a new process that is to encode it in the computer for easy storing and retrieval of data and we find it very effective they do not need to re-encode it in excel as their report in the new process they can search the data by date. The queue number will be send to the truck diver through SMS notification.The verification of delivery receipt will update also through SMS notification. The real time update of queuing will be display in a LCD monitor. Manager can also monitor the reports via dashboard anytime.

In the Development phase, developer is starting to install all the software and application in developing the system. He inserted all the functions or the codes to the system. While we are searching for some reference including the coding technique through the use of an internet or web. The logic behind the functions had been converted by the developers in order to create a suitable function for the application. The development of the system was a big challenge for the proponents due to limited time we're worried if we can make it in time and need also to consider our other subjects but with the help of time management we finished the system. Also some of us are not familiar in VB.net and MS SQL Server proponents need to study on how it works by the help of google and youtube it helps us greatly. Data Analytics also is our challenge on how to make graphs and the computation of the KPI through our research, time study and setting of targets we efficiently represent the data. In order to run the program smoothly it needs to undergo in testing phase. Once we are done in one function we do testing to check it's possible error. The first problem that we encountered was the connection of database through server and we find out that this problem be with the proper using of ports. The inclusion of the SMS in our system was also our challenge first we try to used the SMS with driver but it is not applicable so we use the SMS API with http request and find out that it is much better. Date and time error also was encountered during our development. The process continued until all function was tested and all the errors was solved by the developer. Proponents do a testing at DHL, they started with the guard functions to receiver functions they tested it for 1 hour to make them familiarize with the system. After the testing we do the post survey in DHL asses the systems functionality, reliability, usability, efficiency and portability. Also we get the suggestions of our respondents for improvement of the systems quality and to make sure that none of the requirements are missed.

After the testing phase and passed each of it, it will now proceed to deployment phase. We presented it to DHL manager and they find it useful. As of this moment we are waiting for the deployment date.

After the deployment phase proponents will go to review phase where the process of examining software related documents such as requirement documents, design documents, user feedback that will be use for systems continues improvement.

Table 3.Category of Target dock time per truck

Category by Dock Time	
On foot /2 wheeler	5 mins
4w	5 mins
6w	5 mins
6wfwd	10 mins
10w	10 mins
20 ftr	15 mins
40 ftr	15 mins

In this table shows the target dock time per truck from 2 wheeler to Container Van. Smaller trucks has a faster dock time than bigger trucks.

Table 4. Target time by Truck Type

Category by Truck	
on foot /2 wheeler	30 -45 mins
4w	30 -45 mins
6w	75 - 90 mins
10w	90-120 mins
20 ftr	90-20 mins
40 ftr	120 -150 mins

In this table shows the target time per truck from 2 wheeler trucks to 40 ftr container Van. Smaller truck size has a faster time to unload than the bigger trucks.

Table 5. Target time for weight & quantity.

		Weight per box (kg)					
		0 to 1	2 to 5	6 to10	11 to 15	16 to 20	21 to 25
Qty per box	0 to 50	simple	simple	simple	simple	simple	Medium
	51 to 100	simple	simple	simple	simple	Medium	Medium
	101 to 200	simple	simple	simple	Medium	Medium	Medium
	201 to 300	simple	simple	Medium	Medium	Medium	Difficult
	301 to 400	simple	Medium	Medium	Medium	Difficult	Difficult
	401 to 500	Medium	Medium	Medium	Difficult	Difficult	Difficult
	501 to 600	Medium	Medium	Difficult	Difficult	Difficult	Very difficult
	601 to 700	Medium	Difficult	Difficult	Difficult	Very difficult	Very difficult
	701 to 800	Difficult	Difficult	Difficult	Very difficult	Very difficult	Very difficult
	801 to 900	Difficult	Difficult	Very difficult	Very difficult	Very difficult	Very difficult
	901 to 1000	Difficult	Very difficult	Very difficult	Very difficult	Very difficult	Very difficult
	1001 to 1100	Very difficult	Very difficult	Very difficult	Very difficult	Very difficult	Very difficult
	1101 to 1200	Very difficult	Very difficult	Very difficult	Very difficult	Very difficult	Very difficult

In this table shows the range of target time in terms of Quantity per box and weight per box. In this table also analyze the category of items from simple to Very difficult. This category varies on the quantity and weight of the box, lighter box and less number of box falls under category simple the more quantity and heavier box the more it's difficulty to unload.

Chapter IV

Presentation, Analysis and Interpretation of Data

Research Method Used

In this study, the developmental research approach was used. Developmental research has been defined as the systematic study of designing, developing and evaluating instructional programs, processes, and products that must meet the criteria of internal effectiveness. The proponents use developmental research since we all know how the system works. We just want to make it easier and effective.

The next method is Descriptive, the process of descriptive research involved gathering and tabulation of information of the developed system. This method was used to determine the use or help of the system to the user. Under the descriptive method, the proponents gathered all data from interview and observation to acquire information concerning to the study and then we tabulated and organized to form a comprehensive evidence to support the study.

DESCRIPTION OF RESPONDENTS

The respondents of our study are the DHL Employee , driver and helper of Avon Suppliers who used the queue system. They were the ones who have enough knowledge to answer the problems posed in this study. They answered the questionnaire that the researchers relay in which supplies the information that need.

A total of 20 employees and 30 non employees of DHL were the respondents of the study.

RESEARCH INSTRUMENT

In this study, the survey questionnaire instruments were used to determine the main objective of the study. The questionnaire was distributed to all DHL employees and non employees to assess the proposed queue management system. The questionnaire was structured in such a way that respondents will be able to answer it easily. A Likert scale is a rating scale that requires the subject to indicate his or her degree of agreement or disagreement to a statement or question. Thus, the set of questionnaire was structured using the Likert format that has a five-point response scale, below are the designated quantifications used in the questionnaires:

Table 6 : 5 Point Likert Scale System

Likert Scale	Numerical Rating	Verbal Interpretation
5	4.25 - 5.00	Very Satisfied
4	3.25 - 4.24	Satisfied
3	2.25 - 3.24	Partially Satisfied
2	1.25 - 2.24	Dissatisfied
1	0- 1.24	Very Dissatisfied

Table 6, demonstrate the verbal interpretation that was used in calculating the level of acceptance of the proposed system. The point that has 3.25 – 4.00 was the highest score and 0 – 1.24 was the lowest score.

Table 7: 4 Point Likert Scale System

Likert scale	Numerical Rating	Verbal interpretation
1	0-1.24	Serious Problem
2	1.25-2.24	Moderate Problem
3	2.25-3.24	Minor Problem
4	3.25-4.00	Not a problem at all

Table 7, shows the scoring system that was used in calculating the level of acceptance of the proposed system 1.00 – 2.50 was the lowest score and 4.51-5.00 was the highest score.

DATA GATHERING PROCEDURES

The data for this research were collected using a Survey. The survey was created using suitable questions modified form related research and individual questions formed by the researcher. The client will evaluate the application functions, consistency and effectiveness.

The evaluation procedures are the following:

1. Explaining first the problem of the current queue management system.
2. Followed by the main function of the proposed application where the proponents will explain the step by step procedures in how to use the application.

3. Lastly, all the answers gathered in the interview must collect for the clarification purposes.

Internet Research-The proponents use internet to find some related information about the proposed system including the codes for possible action or process. Those codes that have been search will serve as a guidelines or pattern for the developers.

Interview – Gather the important information’s that the researchers need in the study from the truckers to Managers of DHL Supply Chain, collecting an information and data regarding the process, problems that encountered in using the manual queue management system to develop a solution.

GATHERING RESEARCH

The Researchers used internet research to gather information on the system that are about to develop which is the Queue Management System. Here are the information that the researcher gathered through the internet.

Coding system

The Researchers conduct a research through video tutorial at www.youtube.com for reference and to look for similar function that can be used in the system at https://stackoverflow.com

Design for the system

Conducted some research for the example of the dashboard display of our system at www.youtube.com

Information for the documentation

The researchers did about an information of past or old data about the system that used quick response code and agile model picture and description to represent the process of the system at http://www.researchgate.net and Academia.edu.

PRE SURVEY

RESPONDENTS TALLY

1. How satisfied are you with the current queue management system under DHL (Supply chain Parañaque)?

(5) Very Satisfied (4) Satisfied (3) Partially Satisfied (2) Dissatisfied (1) Very Dissatisfied

1.1 Customer Service Satisfaction	5	4	3	2	1
1.1.1 Waiting Time	2	6	10	22	10
1.1.2 Management of delivery	2	10	8	18	12
1.1.3 Unloading of items	2	6	8	20	14
1.1.4 Encoding of data	4	6	10	14	16
1.1.5 Information of the unloading	6	10	12	12	10

2. Rate the time spent in queue

(4) Serious problem (3) Moderate problem (2) Minor problem (1) Not a problem at all

Waiting Time	**4**	**3**	**2**	**1**
2.1.1 Encoding of Docs (DR Info)	4	26	15	5
2.1.2 Checking of Documents (DR vs System)	6	24	12	8
2.1.3 Unloading of item	12	10	20	2
2.1.4 Receiving of Items	4	24	18	4
2.1.5 Docking of Truckers	5	28	16	1

3. What is your level of agreement on the problem of using manual queuing system?
(4) Serious problem (3) Moderate problem (2) Minor problem (1) Not a problem at all

3.1 Manual Queue Management System	4	3	2	1
3.1.1 Long waiting time of queues	20	10	16	4
3.1.2 Missing queuing numbers	20	10	14	6
3.1.3 Manual truck door assignment	8	18	18	6
3.1.4 Availability of wait time information	22	8	14	6
3.1.5 Storage of Data	4	26	10	10
3.1.5 Calling of trucks to be unload	8	26	16	

4. What is the likelihood that you will support the deployment of Automated truck queuing management system in your department?
(5) Extremely likely (4) Likely (3) Medium likelihood (2) Unlikely (1) Extremely unlikely

Automated Truck Management System	**5**	**4**	**3**	**2**	**1**
4.1.1 Organized queue management System	18	20	12		
4.1.2 Paperless queuing number by using SMS	22	18	10		
4.1.3 Database easy retrieval of Data	26	16	6		
4.1.4 Monitor display and SMS notification for time waiting.	26	18	6		
4.1.5 Automatic truck door assignment	26	14	10		

5. What is your level agreement on the identified characteristics of the propose queue management system.

(5) Strongly Agree (4) Agree (3) Partly Agree (2) Disagree (1) Strongly Disagree

5.1 Features and Functionality	**5**	**4**	**3**	**2**	**1**
5.1.1 Features and functionality meet common user goals and objectives.	20	22	8		
5.1.2 Frequently-used tasks are readily available (e.g. easily accessible from the forms) and well supported.	18	28	4		
5.1.3 Users are adequately supported according to their level of expertise	25	23	2		

PRE-SURVEY RESULTS

Table 8: Profile of the Respondents based on Gender

Sex	DHL	Non DHL	Total
Male	9	30	39
Female	11	0	11
			50

The table 8 shows the people of the respondents based on Gender. Table 6 shows that majority of the respondents are male.

Table 9: Profile of Target Respondents

Target Respondents	Frequency	Percentage
DHL	20	40 %
Non DHL	30	60%
Total	50	100%

Table 9 shows the number of target respondents. Most of the respondents are Non DHL with the percentage of 60 % while the 40 % are of the respondents are DHL Employees.

Table 10: Profile of Respondents based on Age.

Age	Frequency	Percentage
18-23	6	12%
24-28	12	24%
29-33	14	28%
34-38	12	24%
39-Above	6	12%
Total	50	100%

Table 10 shows the age of the respondents. Most of the age of respondents are 29-33 years old with 28% while the least of respondents by age are 18-23 years old and 39 and Above yeas old with 6%.

Table 11. Weighted Mean and Findings.

1. How satisfied are you with the current queue management system under DHL (Supply chain Parañaque)?

1.1 Customer Service Satisfaction	**Weight Average**	**Remarks**
1.1.1 Waiting Time	2.2	Dissatisfied
1.1.2 Management of delivery	2.44	Partially Satisfied
1.1.3 Unloading of items	2.24	Dissatisfied
1.1.4 Encoding of data	2.36	Partially Satisfied
1.1.5 Information of the unloading	2.8	Partially Satisfied

Table 11 Represents the level of satisfaction for having a automated queue management system than the existing queuing system. The table states that the overall weight average value of 2.4% is interpreted as "Partially Agree". The indicated statement " Waiting Time" got the highest rating from respondents who got 2.2% or "Dissatisfied" remarks. However, the indicated statement " Information of the unloading" earned the lowest rating of 2.8% or "Partially Satisfied".

Table 12.

2. Rate the time spent in queue

Waiting Time	**Weight Average**	**Remarks**
2.1.1 Encoding of Docs (DR Info)	2.42	Minor Problem
2.1.2 Checking of Documents (DR vs System)	2.44	Minor Problem
2.1.3 Unloading of item	2	Moderate Problem

2.1.4 Receiving of Items	2.44	Minor Problem
2.1.5 Docking of Trucks	2.26	Minor Problem

Table 12. Represents the level of problem in rate of time spent in queue. The table states that the overall weight average value of 2.89 is interpreted as "Minor Problem". The indicated statement "Unloading of Item" got the highest rating from respondents who got 2 or "Moderate Problem" remarks. However, the indicated statement " Docking of Trucks " earned the lowest rating of 2.26 or "Minor Problem".

Table 13.

3. What is your level of agreement on the problem of using manual queuing system?

3.1 Manual Queue Management System	**Weight Average**	**Remarks**
3.1.1 Long waiting time of queues	2.08	Moderate Problem
3.1.2 Missing queuing numbers	2.12	Moderate Problem
3.1.3 Manual truck door assignment	2.44	Minor Problem
3.1.4 Availability of wait time information	2.08	Moderate Problem
3.1.5 Storage of Data	2.52	Minor Problem
3.1.5 Calling of trucks to be unload	2.16	Moderate Problem

Table 13. Represents the level of problem in the existing queuing system . The table states that the overall weight average value of 2.17 is interpreted as "Moderate Problem". The indicated statement "Availability of wait time information " and “ Long waiting time of queues” got the highest rating from respondents who got 2.08 or "Moderate Problem" remarks. However, the indicated statement " Storage of Data " earned the lowest rating of 2.52 or "Minor Problem".

Table 14.

4. What is the likelihood that you will support the deployment of Automated truck queuing management system in your department?

4.1 Automated Truck Management System	**Weight Average**	**Remarks**
4.1.1 Organized queue management System	4.12	Likely
4.1.2 Paperless queuing number by using SMS	4.16	Likely
4.1.3 Database easy retrieval of Data	4.44	Extreme Likely
4.1.4 Monitor display and SMS notification for time waiting.	4.24	Likely
4.1.5 Automatic truck door assignment	4.32	Extreme Likely

Table 14. Represents the level of likelihood of support in deployments of Data Analytics and SMS Notification for Effective Queue Management System. The table states that the overall weight average value of 4.26 is interpreted as "Extreme Likely". The indicated statement " Database easy retreival of Data " got the highest rating from respondents who got 4.44 or "Moderate Problem" remarks. However, the indicated

statement " Organized queue management System " earned the lowest rating of 4.12 or "Minor Problem".

Table 15.

5. What is your level agreement on the identified characteristics of the propose queue management system.

05.1 Features and Functionality	Weight Average	Remarks
5.1.1 Features and functionality meet common user goals and objectives.	4.24	Agree
5.1.2 Frequently-used tasks are readily available (e.g. easily accessible from the forms) and well supported.	4.28	Strongly Agree
5.1.3 Users are adequately supported according to their level of expertise	4.44	Strongly Agree

Table 15. Represents the level of agreement on the identified characteristics of propose queue management system. The table states that the overall weight average value of 4.9 is interpreted as "Strongly Agree". The indicated statement " Users are adequately supported according to their level of expertise " got the highest rating from respondents who got 4.44 or "Strongly Agree" remarks. However, the indicated statement " Features and functionality meet common user goals and objectives. " earned the lowest rating of 4.24 or "Agree".

POST SURVEY

RESPONDENTS TALLY

Table 16. Respondents Tally Post Survey Question

Evaluation	Strongly Agree	Partially Agree	Disagree	Strongly Disagree
	4	**3**	**2**	**1**
Reliability				
1. This system has a provide real time updates.	48	2		
2. The system can provide faster way of processing queues.	46	4		
1. This system helps to reduce the waiting time.	45	5		
Usability				
1. The Development of Truck queue management system is useful to DHL.	48	2		
2. Recommend using this system to others.	47	3		
3. This system help the DHL to ease the long waiting of Queues.	48	2		
4. The Queue management system gives a real time information to stakeholders.	50	0		
Functionality				
1. The system provides a queue no.thru SMS	50	0		
2. The system is capable of giving update through sms.	50	0		

3. The System display information of the queues through monitor display.	50	0		
4. The System can generate Statistical & Graphical Reports	50	0		

Efficiency				
1. The system process queues more quicker than the traditional process.	45	5		
2. The system provides help option once installed.	42	8		
3. System provides real time reports of the queues.	45	5		
4. The System track the queues.	46	4		
Portability				
1. The system is working in windows 7 and above.	45	0		
2. The System provides software ready for online installation.	42	8		
3. The system has a hardware requirement of Laptop computer, 64bit System Architecture and 2GB RAM.	42	8		

SUMMARY

RESPONDENTS ASSESSMENTS

Table17: Profile of Target Respondents

Target Respondents	Frequency	Percentage
DHL	20	40 %
Non DHL	30	60%
Total	50	100%

Table 17. The table shows the number of target respondents. Most of the respondents are Non DHL with the percentage of 60 % while the 40 % are of the respondents are DHL Employees.

Table 18: Frequency and Percentage according to age

AGE	FREQUENCY	PERCENTAGE
15-25	12	24%
26-35	28	56%
Above	15	30%
Total	50	100%

Table 18: The table shows the age of the respondents. Most of the age of respondents are 26-35 years old with 56%. While the least of respondents by age are 15-25 old with 24%.

Table 19: Frequency and Percentage according to gender

GENDER	FREQUENCY	PERCENTAGE
Male	38	76%
Female	12	24%
Total	50	100%

Table 19: This table shows among all the respondents male has the highest number to response to the questionnaires.

Table 20: Reliability of the System

Reliability	Weight Average	Remarks
1. This system has a provide real time updates.	3.96	Very Useful
2. The system can provide faster way of processing queues.	3.92	Very Useful
3. This system helps to reduce the waiting time.	3.9	Very Useful

Table 20. Represents the reliability. The table stated that the overall weight average is 3.53% interpreted as “Very useful”. The indicated statement “Apply proper indentation of codes” got the highest rating from the respondents earned 3.58% or “Very useful” remarks. However, the indicated statement “Display java programming codes” earned the lowest rating of 3.46% or “Useful”.

Table 21: Usability of the System

Usability		
1. The Development of Truck queue management system is useful to DHL.	3.96	Very Useful
2. Recommend using this system to others.	3.94	Very Useful
3. This system help the DHL to ease the long waiting of Queues.	3.96	Very Useful
4. The Queue management system gives a real time information to stakeholders.	4	Very Useful

Table 21. Represents the Usability. The table stated that the overall weight average is 3.53% interpreted as “Very useful”. The indicated statement “Apply proper indentation of codes” got the highest rating from the respondents earned 3.58% or “Very useful” remarks. However, the indicated statement “Display java programming codes” earned the lowest rating of 3.46% or “Useful”.

Table 22: Efficiency of the System

Efficiency		
1. The system process queues more quicker than the traditional process.	3.9	Very Useful
2. The system provides help option once installed.	3.84	Very Useful
3. System provides real time reports of the queues.	3.9	Very Useful
4. The System track the queues.	3.92	Very Useful

Table 22. Represents the Efficiency. The table stated that the overall weight average is 3.53% interpreted as “Very useful”. The indicated statement “Apply proper indentation of codes” got the highest rating from the respondents earned 3.58% or “Very useful” remarks. However, the indicated statement “Display java programming codes” earned the lowest rating of 3.46% or “Useful”.

Table 23: Portability of the System

Portability		
1. The system is working in windows 7 and above.	3.6	Very Useful
2. The System provides software ready for online installation.	3.84	Very Useful
3. The system has a hardware requirement of Laptop computer, 64bit System Architecture and 2GB RAM.	3.84	Very Useful

Table 23. Represents the portability. The table stated that the overall weight average is 3.53% interpreted as “Very useful”. The indicated statement “Apply proper indentation of codes” got the highest rating from the respondents earned 3.58% or “Very useful” remarks. However, the indicated statement “Display java programming codes” earned the lowest rating of 3.46% or “Useful”.

CHAPTER V

RESULT AND DISCUSSION

CONCLUSION

Based on the information gathered the researchers therefore conclude that it is necessary to use the proposed system it is more efficient than the existing system. Based on the time study that the researcher conducted between current and proposed system it show that the proposed system is 37% more efficient than the current system.

Table 24. Time Study of Encoding Time Current vs Proposed.

	Current	**Proposed**		
Count of DR	**Encoding Time Ave**	**Encoding Time Ave**	**Diff encoding**	**Percentage**
1	0:01:23	0:01:02	0:00:21	**26%**
2	0:02:18	0:01:14	0:01:04	**46%**
3	0:02:02	0:01:36	0:00:26	**22%**
4	0:02:59	0:01:53	0:01:06	**37%**
5	0:03:30	0:02:13	0:01:16	**36%**
6	0:06:06	0:03:32	0:02:33	**42%**
7	0:06:51	0:03:45	0:03:06	**45%**
8	0:07:20	0:05:01	0:02:19	**32%**
9	0:08:40	0:06:28	0:02:13	**25%**
10	0:15:07	0:06:50	0:08:17	**55%**

Table 24. In this table shows that in terms of encoding proposed system is much faster than the current queue management system in encoding with the average of 37%

Table 25. Time Study of Checking Time Current vs Proposed.

	Current	Proposed		
Count of DR	**Checking Time Ave**	**Checking Time Ave**	**Diff Checking**	**Percentage**
1	0:01:26	0:00:52	0:00:34	**40%**
2	0:01:47	0:01:02	0:00:45	**42%**
3	0:02:39	0:01:02	0:01:37	**61%**
4	0:06:15	0:03:06	0:03:10	**51%**
5	0:06:09	0:05:12	0:00:57	**15%**
6	0:07:36	0:06:14	0:01:22	**18%**
7	0:07:51	0:07:19	0:00:42	**9%**
8	0:10:05	0:07:50	0:02:15	**22%**
9	0:11:32	0:08:05	0:03:27	**30%**
10	0:12:07	0:09:35	0:02:32	**21%**

Table 25. In this table shows that in terms of checking of DR the proposed system is much faster than the current queue management system in encoding with the average.

RECOMMENDATION

Based on the findings of the study researcher would like to make the following recommendations for future improvements:

1. Future research might apply an android version of receivers panel for easy assigning of docks.
2. Future research should apply the Integration of Avon System to Truck queue management system to lessen the encoded data.
3. Future research should include online truck booking.

REFERENCES

Nityangini Jhala and Pravin Bhathawala.”Smart Queue Management System for Banks”. < http://www.irphouse.com>. October 19, 2019

Farayibi, Adesoji. Investigating the Application of Queue Theory in the Nigerian Banking System.< http://www.papers.ssrn.com>. October 19, 2019

Gonçalo António Rendeiro da Silva. “Integrated Queue Management System - Técnico Lisboa”.< http://www .fenix.tecnico.ulisboa.pt> October 19, 2019.

Al-Barhamtoshy Hassanin.”A Data Analytic Framework for Unstructured Text”.<http://www.researchgate.net>.October 19, 2019.

Austria, Leoven A.” Queue Management Practices of Quick ServiceRestaurants”.< http://www.apjmr.com>. October 19, 2019.

Miranda et al.”A Dental Advisories and Sms Notification System Utilizing Predictive Analytics”.< http://www.ijarcs.com>. October 19, 2019.

Malala, Nieles and Villanueva.” A Queue Management Systems Characteristics, Advantages and Application” < http://www.academia.com>. October 19, 2019.

Abdullah, Jmael.” Queue Management System for STI Iligan Registrar”. <https://www.prezi.com> .October 19, 2019.

Woottichaiwat, Sujin. “Efficiency Improvement of Truck Queuing in the Freight Unloading Process Case Study of a Private Port in Songkhla Province”.< https://www.semanticscholar.org>.October 19, 2019.

Pad-berg, Mike.”Big Data and Business Intelligence”.< https://www.coursehero.com>. October 19, 2019.

Kuklin, Perry. ”Breaking Down Big Data for Better Queue Management”. < https://www.lavi.com/ >. October 19, 2019

Ngorsed, Manoon & Suesaowaluk, Poonphon .”A Hospital Service Queue Management System With Wireless Approach”.< https://www.springerprofessonal.de>. October 19, 2019

Udin et al. “An automated queue management system”. < https://www.semanticscholar.org/>. October 19, 2019

Veluya et al. “Queueing Systems to Minimize the Waiting Time during Enrollment in Southern Luzon State University”.< https://ejournals.ph>. October 19, 2019.

Lumauag,Ramil.”School Event Notification through SMS.” <https://www.researchgate. net> October 19, 2019

Cosidon, Eileen B. “Student Information System for Kalinga State university.” https://docplayer.net October 19, 2019

Abapo, Torregosa and Tomboc. “A UX Case Study: Rose Pharmacy Customer Queue Management System.”< https://medium.com> October 19, 2019

SYSTEM CODES

DBconnection.vb

```
Imports System.Data.SqlClient

Public Class DBconnection

Dim connectionString As String = "Server='" & My.Settings.mServer & "';

Database='" & My.Settings.mDB & "';

User Id='" & My.Settings.mUserDB & "';

Password='" & My.Settings.mPassDB & "';MultipleActiveResultSets=True"

Public DBcon As New SqlConnection(connectionString)

Public Shared ReceiverName As String = ""

End Class
```

Login.vb

```
Imports System.Text.RegularExpressions

Imports System.Data.SqlClient

Imports System.Data

Imports System.ComponentModel

Public Class Login

Dim con As SqlConnection

Dim cmd As SqlCommand
Dim Errorcatch As String
Dim role As String
Dim status As String
Dim receiverpublic As New DBconnection
Private Sub Login_Load(sender As Object, e As EventArgs) Handles MyBase.Load
Me.FormBorderStyle = FormBorderStyle.None
Password.UseSystemPasswordChar = True
Me.KeyPreview = True
End Sub
Private Sub CheckBox1_CheckedChanged(sender As Object, e As EventArgs) Handles CheckBox1.CheckedChanged
If CheckBox1.CheckState = CheckState.Checked Then
Password.UseSystemPasswordChar = False
Else
Password.UseSystemPasswordChar = True
End If
End Sub
Private Sub Login_KeyDown(sender As Object, e As KeyEventArgs) Handles Me.KeyDown
If e.KeyCode = Keys.D And e.Shift And e.Control Then
 AdminPrompt.Show()
End If
End Sub
Private Sub Panel1_Paint(sender As Object, e As PaintEventArgs)
End Sub
Private Sub BunifuFlatButton1_Click(sender As Object, e As EventArgs)
Handles BunifuFlatButton1.Click
If Username.Text = "" Then
 Errorlbl.Text = "Error: Please UserName"
 Errorlbl.ForeColor = Color.Gold
 Username.Focus()
ElseIf Password.Text = "" Then
 Errorlbl.ForeColor = Color.Gold
 Errorlbl.Text = "Error: Please Enter Password"
 Password.Focus()
Else
Dim con As New SqlClient.SqlConnection
Dim connectionString As String = "Server='" & My.Settings.mServer & "';
Database='" & My.Settings.mDB & "';
User Id='" & My.Settings.mUserDB & "';
Password='" & My.Settings.mPassDB & "';"
 con.ConnectionString = connectionString
Dim selectquery As String = "select * from Usertbl where
username=@username and password=@password"
```

```
Try
    con.Open()
    cmd = New SqlCommand(selectquery, con)
    cmd.Parameters.AddWithValue("@username", Username.Text)
    cmd.Parameters.AddWithValue("@password", Password.Text)
    Dim dr As SqlDataReader
    dr = cmd.ExecuteReader()
    If dr.Read Then
        role = dr.Item("UserRole")
        status = dr.Item("Status")
        receiverpublic.ReceiverName = dr.Item("Name")
        If role.Equals("Admin") And status.Equals("Activated") Then
            Me.Hide()
            Admin.Show()
            Username.Clear()
            Password.Clear()
            Errorlbl.Text = ""
          MessageBox.Show("Welcome" + " " + dr("UserRole").ToString() + " " + dr("Name").ToString(), "Welcome")
        ElseIf role.Equals("Manager") And status.Equals("Activated") Then
            Me.Hide()
            Dashboard.Show()
```

```
   Username.Clear()

   Password.Clear()

   Errorlbl.Text = ""

 MessageBox.Show("Welcome" + " " + dr("UserRole").ToString() + "
 " + dr("Name").ToString(), "Welcome")

ElseIf role.Equals("Receiving Associate") And
status.Equals("Activated") Then

   Me.Hide()

   DRchecking.Show()

   Username.Clear()

   Password.Clear()

   Errorlbl.Text = ""

 MessageBox.Show("Welcome" + " " + dr("UserRole").ToString() + "
 " + dr("Name").ToString(), "Welcome")

ElseIf role.Equals("Receiver") And status.Equals("Activated") Then

   Me.Hide()

   Receiver.Show()

   Username.Clear()

   Password.Clear()

   Errorlbl.Text = ""

 MessageBox.Show("Welcome" + " " + dr("UserRole").ToString() + "
 " + dr("Name").ToString(), "Welcome")

ElseIf role.Equals("Guard") And status.Equals("Activated") Then
```

```
     Me.Hide()

     Securityguard.Show()

     Username.Clear()

     Password.Clear()

     Errorlbl.Text = ""

    MessageBox.Show("Welcome" + " " + dr("UserRole").ToString() + "
    " + dr("Name").ToString(), "Welcome")

   Else

    MessageBox.Show("Your Account Is Not Active Yet, Please
    Contact Administrator", "Account Error")

     Username.Clear()

     Password.Clear()

     Username.Focus()

     Errorlbl.Text = ""

   End If

Else

   Errorlbl.Text = "Username and Password are Invalid"

   con.Close()

End If

    Catch ex As Exception

MessageBox.Show("Connection Error", "Error Message")
```

```
 End Try

 con.Close()

End If

 End Sub

Private Sub Login_Closing(sender As Object, e As CancelEventArgs) Handles Me.Closing

End Sub

Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click

Application.Exit()

End Sub

End Class
```

Security.vb

```
Imports System.Text.RegularExpressions

Imports System.Data.SqlClient

Imports System.Data

Public Class Securityguard

    Dim con As SqlConnection

    Dim cmd As SqlCommand

    Dim adapter As SqlDataAdapter

    Dim dr As SqlDataReader

    Dim table As New DataTable
Dim Errorcatch As String
Dim i As Integer
Dim cmd1 As SqlCommand
Dim stat As Integer
Public dbs As New BindingSource
Public dbs1 As New BindingSource
Public dbs2 As New BindingSource
Public dbs3 As New BindingSource
Dim Arrivaldate As String
Dim Adate As DateTime
Dim Arrivaltime As String
Dim Atime As DateTime
Dim queueval As String
Private Sub Label2_Click(sender As Object, e As EventArgs)
End Sub
Public Sub queuetable()
    Dim con As New SqlClient.SqlConnection
       Dim connectionString As String = "Server='" & My.Settings.mServer & "';
Database='" & My.Settings.mDB & "';
User Id='" & My.Settings.mUserDB & "';
Password='" & My.Settings.mPassDB & "';"
```

```
   con.ConnectionString = connectionString
     Dim selectquery As String = "select Queueno as 'Queue No',
     vendorname 'Vendor Name', plate_no as 'Plate No', truck_type as
     'Truck Type', truck_load as 'Truck Load', no_of_helper as 'No of
     Helper', driver_name as 'Driver Name', contact_no as 'Contact no',
     category as 'Category' from truck_details "
   Try
      con.Open()
      cmd = New SqlCommand(selectquery, con)
      Dim da As New SqlDataAdapter(cmd)
      Dim dt As New DataTable()
      da.Fill(dt)
      dbs.DataSource = dt
      truckdetailslist.DataSource = dbs
      truckdetailslist.ReadOnly = True
   Catch ex As Exception
      MessageBox.Show(ex.Message.ToString(), "Error Message")
   End Try
End Sub
Public Sub drtable()
   Dim con As New SqlClient.SqlConnection
     Dim connectionString As String = "Server='" & My.Settings.mServer
     & "';
Database='" & My.Settings.mDB & "';
```

```
    User Id='" & My.Settings.mUserDB & "';
    Password='" & My.Settings.mPassDB & "';"
        con.ConnectionString = connectionString
        Dim selectquery As String = "select queueno as 'Queue no', drno as
'DR no', noofboxes as 'No of Boxes', totalunits as 'Total Units' from drtbl"
        Try
            con.Open()
            cmd = New SqlCommand(selectquery, con)
            Dim da As New SqlDataAdapter(cmd)
            Dim dt As New DataTable()
            da.Fill(dt)
            dbs1.DataSource = dt
            drdetails.DataSource = dbs1
        Catch ex As Exception
            MessageBox.Show(ex.Message.ToString(), "Error Message")
        End Try
    End Sub
    Public Sub drtable2()
        Dim con As New SqlClient.SqlConnection
        Dim connectionString As String = "Server='" & My.Settings.mServer
& "';
    Database='" & My.Settings.mDB & "';
    User Id='" & My.Settings.mUserDB & "';
    Password='" & My.Settings.mPassDB & "';"
        con.ConnectionString = connectionString
        Dim selectquery As String = "select queueno as 'Queue no', drno as 'DR no', noofboxes as 'No of Boxes', totalunits as 'Total Units' from drtbl where queueno=@username"
        Try
            con.Open()
            cmd = New SqlCommand(selectquery, con)
            cmd.Parameters.AddWithValue("@username", queueval)
            Dim da As New SqlDataAdapter(cmd)
            Dim dt As New DataTable()
            da.Fill(dt)
            dbs3.DataSource = dt
            drdetails.DataSource = dbs3
            drdetails.ReadOnly = True
        Catch ex As Exception
            MessageBox.Show(ex.Message.ToString(), "Error Message")
        End Try
    End Sub


    Private Sub Form1_Load(sender As Object, e As EventArgs) Handles MyBase.Load
```

```
truckdetailslist.RowHeadersVisible = False

truckdetailslist.GridColor = Color.Red

drdetails.RowHeadersVisible = False

drdetails.GridColor = Color.Red

Me.FormBorderStyle = FormBorderStyle.None

Dim myname As New DBconnection

plateno.MaxLength = 11

Name = myname.ReceiverName

Label24.Text = Name

Cancel.Hide()

Savedr.Enabled = False

queuetable()

drtable()

drdetails.Show()

Label16.Hide()

Label7.Hide()

drno.Enabled = False

box.Enabled = False

units.Enabled = False


Savedr.Show()

adddr.Hide()
```

```
        done.Hide()

        edit.Hide()

        update.Hide()

        Button1.Hide()

        drdetails.Enabled = False

        truckdetailslist.Enabled = True

        vendorname.Enabled = False

        trucktype.Enabled = False

        plateno.Enabled = False

        helper.Enabled = False

        truckload.Enabled = False

        drivername.Enabled = False

        contact.Enabled = False

        category.Enabled = False

    End Sub

    Private Sub BunifuFlatButton1_Click(sender As Object, e As EventArgs) Handles BunifuFlatButton1.Click

        Cancel.Show()



        BunifuFlatButton1.Enabled = False

        Savedr.Show()
```

```
adddr.Hide()

done.Hide()

edit.Hide()

update.Hide()

Button1.Hide()

vendorname.Enabled = True

trucktype.Enabled = True

trucktype.Focus()

plateno.Enabled = True

Savedr.Enabled = True

helper.Enabled = True

truckload.Enabled = True

drivername.Enabled = True

contact.Enabled = True

category.Enabled = True

vendorname.Text = ""

trucktype.SelectedIndex = -1

plateno.Clear()

helper.Clear()

truckload.SelectedIndex = -1

drivername.Clear()

contact.Clear()
```

```
        category.SelectedIndex = -1

        truckdetailslist.Enabled = True

        drdetails.Enabled = False

        drno.Enabled = False

        box.Enabled = False

        units.Enabled = False

    End Sub

    Private Sub truckdetailslist_CellMouseClick(sender As Object, e As
DataGridViewCellMouseEventArgs) Handles
truckdetailslist.CellMouseClick

        Cancel.Show()

        Label16.Show()

        Label7.Show()

        Dim con As New SqlClient.SqlConnection

        Dim connectionString As String = "Server='" & My.Settings.mServer
& "';

        Database='" & My.Settings.mDB & "';

        User Id='" & My.Settings.mUserDB & "';

        Password='" & My.Settings.mPassDB & "';"




        con.ConnectionString = connectionString
        i = truckdetailslist.CurrentRow.Index

        queueval = truckdetailslist.Item(0, i).Value.ToString

        Dim selectquery As String

        selectquery = "select truck_details.*, Queue_tbl.* from truck_details
inner join Queue_tbl on  truck_details.Queueno=Queue_tbl.Queueno
where truck_details.Queueno=@queue"

        Label7.Text = queueval

        Try

          con.Open()

          cmd = New SqlCommand(selectquery, con)

          cmd.Parameters.AddWithValue("@queue", queueval)

          cmd.ExecuteNonQuery()

          Dim dt As New DataTable()

          Dim adapter As New SqlDataAdapter(cmd)

          adapter.Fill(dt)

          Dim dr As SqlClient.SqlDataReader

          dr = cmd.ExecuteReader(CommandBehavior.CloseConnection)

          While dr.Read

            'Label17.Text = queueval

            Label7.Text = queueval

            vendorname.Text = dr("vendorname").ToString

            trucktype.Text = dr("truck_type").ToString
            plateno.Text = dr("plate_no").ToString

            helper.Text = dr("no_of_helper").ToString

            truckload.Text = dr("truck_load").ToString

            drivername.Text = dr("driver_name").ToString

            contact.Text = dr("contact_no").ToString

            category.Text = dr("category").ToString

            drtable2()

            Savedr.Hide()

            update.Hide()

            vendorname.Enabled = False

            trucktype.Enabled = False

            plateno.Enabled = False

            helper.Enabled = False

            truckload.Enabled = False

            drivername.Enabled = False

            contact.Enabled = False

            category.Enabled = False

            BunifuFlatButton1.Enabled = False

            edit.Show()

          End While

        Catch ex As Exception

          MessageBox.Show(ex.Message.ToString(), "Error Message")
```

```
        End Try
    End Sub
    Private Sub Savedr_Click(sender As Object, e As EventArgs) Handles Savedr.Click
        BunifuFlatButton1.Enabled = False
        If vendorname.Text = "" Then
            Errorlbl.ForeColor = Color.Red
            Errorlbl.Text = "Error: Please Enter Vendor Name"
            vendorname.Focus()
        ElseIf trucktype.Text = "" Then
            Errorlbl.ForeColor = Color.Red
            Errorlbl.Text = "Error: Please Enter Truck Type"
            trucktype.Focus()
        ElseIf plateno.Text = "" Then
            Errorlbl.ForeColor = Color.Red
            Errorlbl.Text = "Error: Please Enter Plate Number"
            plateno.Focus()
        ElseIf helper.Text = "" Then
            Errorlbl.ForeColor = Color.Red
            Errorlbl.Text = "Error: Please Enter No. of Helper"
            helper.Focus()
        ElseIf truckload.Text = "" Then
```

```
      Errorlbl.ForeColor = Color.Red
      Errorlbl.Text = "Error: Please Enter Truckload"
      helper.Focus()
    ElseIf drivername.Text = "" Then
      Errorlbl.ForeColor = Color.Red
      Errorlbl.Text = "Error: Please Enter Driver Name"
      drivername.Focus()
    ElseIf contact.Text = "" Then
      Errorlbl.ForeColor = Color.Red
      Errorlbl.Text = "Error: Please Enter Contact No"
      contact.Focus()
    ElseIf category.Text = "" Then
      Errorlbl.ForeColor = Color.Red
      Errorlbl.Text = "Error: Please Enter Category"
      category.Focus()
    Else
      Dim con As New SqlClient.SqlConnection
      Dim connectionString As String = "Server='" &
My.Settings.mServer & "';
  Database='" & My.Settings.mDB & "';
  User Id='" & My.Settings.mUserDB & "';
```

```
   Password='" & My.Settings.mPassDB & "';
MultipleActiveResultSets=True"

         con.ConnectionString = connectionString

         Dim query As String = String.Empty

         query &= "INSERT INTO Queue_tbl(Arrival_date, Arrival_time, "

         query &= "                 Status,EncodedBy)  "

         query &= "VALUES (convert(date, getdate()), convert(time,
getdate()),@status,@encoded)"

         query &= "INSERT INTO truck_details (vendorname, plate_no,
truck_type,truck_load, no_of_helper, driver_name,contact_no,category)"

         query &= "VALUES                 (@vendor, @plate,
@truck,@truckload, @helper, @driver,@contact,@category)"

         Dim updatequerystat As String = String.Empty

         updatequerystat &= "update Queue_tbl set NotifStatus=@notifstat
where Queueno=@queue"

         Dim selectquery As String = String.Empty

         selectquery &= "select max(queueno) as queuemax from
Queue_tbl"

         Using conn As New SqlConnection(connectionString)

            Using comm As New SqlCommand()

               With comm

                  .Connection = conn

                  .CommandType = CommandType.Text

                  .CommandText = query
```

```
        .Parameters.AddWithValue("@status", "inserting")

        .Parameters.AddWithValue("@encoded", Label24.Text)

        .Parameters.AddWithValue("@type", category.Text)

        .Parameters.AddWithValue("@vendor", vendorname.Text)

        .Parameters.AddWithValue("@truck", trucktype.Text)

        .Parameters.AddWithValue("@plate", plateno.Text)

        .Parameters.AddWithValue("@helper", helper.Text)

        .Parameters.AddWithValue("@truckload", truckload.Text)

        .Parameters.AddWithValue("@driver", drivername.Text)

        .Parameters.AddWithValue("@contact", contact.Text)

        .Parameters.AddWithValue("@category", category.Text)

    End With

    Try

        conn.Open()

        comm.ExecuteNonQuery()

        queuetable()

MsgBox("Successfully Inserted, Please Enter DR details",
MsgBoxStyle.Information)

        Savedr.Hide()

        vendorname.Enabled = False

        trucktype.Enabled = False

        plateno.Enabled = False
```

```
            helper.Enabled = False

            truckload.Enabled = False

            drivername.Enabled = False

            contact.Enabled = False

            category.Enabled = False

            adddr.Show()

            adddr.Enabled = True

            Label7.Enabled = True

            drno.Enabled = True

            box.Enabled = True

            units.Enabled = True

            drno.Focus()

            Cancel.Show()

            conn.Close()

        Catch ex As Exception

MessageBox.Show(ex.Message.ToString(), "Error Message")

        End Try

        conn.Close()

    End Using

    Using comm1 As New SqlCommand()

        With comm1

            .Connection = conn
            .CommandType = CommandType.Text
            .CommandText = selectquery
          End With
          Try
            conn.Open()
            comm1.ExecuteNonQuery()
            Dim dr As SqlDataReader
            dr = comm1.ExecuteReader()
            If dr.Read Then
              Label7.Text = dr.Item("queuemax")
              Try
                Label7.Show()
                Label16.Show()
Dim smsmessage As String = ("Good Day! Mr. " +
drivername.Text.ToString + " Plate no.: " + plateno.Text.ToString + " Your
Queue No. is: " + dr.Item("queuemax").ToString)
Dim res As String = itexmo(contact.Text, smsmessage,
My.Settings.smsAPI, "(z}v7vymks")
If res = "0" Then
MsgBox("Queue Message Has been Sent!")
cmd1 = New SqlCommand(updatequerystat, conn)
cmd1.Parameters.AddWithValue("@notifstat", "Message Sent!")
     cmd1.Parameters.AddWithValue("@queue", dr.Item("queuemax"))
```

```
    Try

    cmd1.ExecuteNonQuery()

    Catch ex As Exception

    MsgBox("Update Error")

    End Try

Else

MsgBox("Error on Sending Message, Please Contact Administrator")

cmd1 = New SqlCommand(updatequerystat, conn)

cmd1.Parameters.AddWithValue("@notifstat", "Message NOT Sent!")

cmd1.Parameters.AddWithValue("@queue", dr.Item("queuemax"))

             Try

               cmd1.ExecuteNonQuery()

             Catch ex As Exception

               MsgBox("Update Error")

             End Try

            End If

          Catch ex As Exception

            MsgBox("Error on Sending Message, Please Contact Administrator")

            cmd1 = New SqlCommand(updatequerystat, conn)

            cmd1.Parameters.AddWithValue("@notifstat", "Message NOT Sent!")
```

```
                            cmd1.Parameters.AddWithValue("@queue",
                            dr.Item("queuemax"))

                            Try

                                cmd1.ExecuteNonQuery()

                            Catch ex1 As Exception

                                MsgBox("Update Error")

                            End Try

                        End Try

                    Else

                        Errorlbl.Text = "NO RECORD FOUND"

                    End If

                Catch ex As Exception

                  MessageBox.Show(ex.Message.ToString(), "Error
                  Message")

                End Try

                conn.Close()

            End Using

            conn.Close()

        End Using

    End If

End Sub

Private Sub Adddr_Click(sender As Object, e As EventArgs) Handles
adddr.Click
```

```
    If drno.Text = "" Then

        Errorlbl.ForeColor = Color.Red

        Errorlbl.Text = "Error: Please Enter Dr No."

        drno.Focus()

    ElseIf box.Text = "" Then

        Errorlbl.ForeColor = Color.Red

        Errorlbl.Text = "Error: Please Enter No of box."

        box.Focus()

    ElseIf units.Text = "" Then

        Errorlbl.ForeColor = Color.Red

        Errorlbl.Text = "Error: Please Enter Total no. of Units"

        units.Focus()

    Else

      Dim con As New SqlClient.SqlConnection

       Dim connectionString As String = "Server='" & My.Settings.mServer
       & "';

Database='" & My.Settings.mDB & "';

User Id='" & My.Settings.mUserDB & "';

           Password='" & My.Settings.mPassDB & "';
       MultipleActiveResultSets=True"


        con.ConnectionString = connectionString
```

```
 Dim query As String = String.Empty

 query &= "INSERT INTO drtbl(Queueno,drno, "

 query &= "                    noofboxes, totalunits)  "

 query &= "VALUES ( @queue,@drno,@box, @units)"

          Dim selectquery = "select drno from drtbl where
queueno=@queueno and drno=@drno"

 Dim insertquery = "insert into
drtbl(queueno,drno,noofboxes,totalunits)
values(@queue,@drno,@box,@units)"

 Using conn As New SqlConnection(connectionString)

     cmd1 = New SqlCommand(selectquery, conn)

     cmd1.Parameters.AddWithValue("@queueno", Label7.Text)

     cmd1.Parameters.AddWithValue("@drno", drno.Text)

     Try

         conn.Open()

         cmd1.ExecuteNonQuery()

         Dim dr As SqlDataReader

         dr = cmd1.ExecuteReader()

         MsgBox("DR verified", MsgBoxStyle.Information)

         If dr.HasRows Then

             Errorlbl.Text = ("DR no. already exist at this Queue no.")

             Errorlbl.ForeColor = Color.Red

         Else
```

```
        cmd = New SqlCommand(insertquery, conn)

            cmd.Parameters.AddWithValue("@queue", Label7.Text)

            cmd.Parameters.AddWithValue("@drno", drno.Text)

            cmd.Parameters.AddWithValue("@box", box.Text)

            cmd.Parameters.AddWithValue("@units", units.Text)

            cmd.ExecuteNonQuery()

            drtable()

      MsgBox("DR Successfully Added",
      MsgBoxStyle.Information)

            Errorlbl.Text = ""

            drno.Clear()

            drno.Focus()

            box.Clear()

            units.Clear()

            done.Show()

            done.Enabled = True

            conn.Close()

        End If

    conn.Close()

Catch ex As Exception

    MessageBox.Show(ex.Message.ToString(), "Error Message")

End Try
```

```
    End Using

    End If

End Sub

        Private Sub Done_Click(sender As Object, e As EventArgs)
      Handles done.Click

    Dim con As New SqlClient.SqlConnection

            Dim connectionString As String = "Server='" &
      My.Settings.mServer & "';

Database='" & My.Settings.mDB & "';

User Id='" & My.Settings.mUserDB & "';

Password='" & My.Settings.mPassDB & "';"

    con.ConnectionString = connectionString

    Dim query As String = String.Empty

            query &= "UPDATE Queue_tbl set Status=@status ,
      EndEncoded= convert(time, getdate()) where Queueno=@queue "

    Using conn As New SqlConnection(connectionString)

        Using comm As New SqlCommand()

            With comm

                .Connection = conn

                .CommandType = CommandType.Text

                .CommandText = query

                .Parameters.AddWithValue("@status", "On process")

                .Parameters.AddWithValue("@queue", Label7.Text)
```

```
End With
Try
    conn.Open()
    comm.ExecuteNonQuery()
    MsgBox("Successfully Saved", MsgBoxStyle.Information)
    truckdetailslist.Enabled = True
    vendorname.Text = ""
    trucktype.Text = ""
    plateno.Clear()
    helper.Clear()
    truckload.Text = ""
    drivername.Clear()
    contact.Clear()
    category.Text = ""
    Errorlbl.Text = ""
    drno.Enabled = False
    box.Enabled = False
    units.Enabled = False
    adddr.Enabled = False
    done.Enabled = False
    BunifuFlatButton1.Enabled = True
    Button1.Hide()
```

```
                done.Hide()

                adddr.Hide()

                Cancel.Hide()

                drno.Text = ""

                box.Text = ""

                units.Text = ""

                Label7.Text = ""

            Catch ex As Exception

                MessageBox.Show(ex.Message.ToString(), "Error Message")

            End Try

        End Using

    End Using

    con.Close()

End Sub

           Private Sub Edit_Click(sender As Object, e As EventArgs)
       Handles edit.Click

    Dim con As New SqlClient.SqlConnection

               Dim connectionString As String = "Server='" &
       My.Settings.mServer & "';

Database='" & My.Settings.mDB & "';

User Id='" & My.Settings.mUserDB & "';

Password='" & My.Settings.mPassDB & "';"
```

```
con.ConnectionString = connectionString

Dim query As String = String.Empty

        query &= "select Status from Queue_tbl where
  Queueno=@queueno"

Dim status As String

Using conn As New SqlConnection(connectionString)

   conn.Open()

   cmd = New SqlCommand(query, conn)

   cmd.Parameters.AddWithValue("@queueno", Label7.Text)

   Try

      cmd.ExecuteNonQuery()

      Dim dr As SqlClient.SqlDataReader

      dr = cmd.ExecuteReader(CommandBehavior.CloseConnection)

      While dr.Read

         status = dr("Status").ToString

         If status = "Verified" Then

  Errorlbl.Text = "Data has already been verified, You can't edit it
  anymore!!!"

            Errorlbl.ForeColor = Color.Red

         Else

            vendorname.Enabled = True

            trucktype.Enabled = True
```

```
                    plateno.Enabled = True

                    helper.Enabled = True

                    truckload.Enabled = True

                    drivername.Enabled = True

                    contact.Enabled = True

                    category.Enabled = True

                    edit.Hide()

                    update.Show()

                End If

            End While

            conn.Close()

        Catch ex As Exception

            MsgBox(ex.ToString + " Please Contact Administrator")

            conn.Close()

        End Try

    End Using

End Sub

       Private Sub Update_Click(sender As Object, e As EventArgs)
       Handles update.Click

    If vendorname.Text = "" Then

        Errorlbl.ForeColor = Color.Red

        Errorlbl.Text = "Error: Please Enter Vendor Name"
    vendorname.Focus()
ElseIf trucktype.Text = "" Then
    Errorlbl.ForeColor = Color.Red
    Errorlbl.Text = "Error: Please Enter Truck Type"
    trucktype.Focus()
ElseIf plateno.Text = "" Then
    Errorlbl.ForeColor = Color.Red
    Errorlbl.Text = "Error: Please Enter Plate Number"
    plateno.Focus()
ElseIf helper.Text = "" Then
    Errorlbl.ForeColor = Color.Red
    Errorlbl.Text = "Error: Please Enter No. of Helper"
    helper.Focus()
ElseIf truckload.Text = "" Then
    Errorlbl.ForeColor = Color.Red
    Errorlbl.Text = "Error: Please Enter Truckload"
    helper.Focus()
ElseIf drivername.Text = "" Then
    Errorlbl.ForeColor = Color.Red
    Errorlbl.Text = "Error: Please Enter Driver Name"
    drivername.Focus()
ElseIf contact.Text = "" Then
```

```
      Errorlbl.ForeColor = Color.Red

      Errorlbl.Text = "Error: Please Enter Contact No"

      contact.Focus()

   ElseIf category.Text = "" Then

      Errorlbl.ForeColor = Color.Red

      Errorlbl.Text = "Error: Please Enter Category"

      category.Focus()

   Else

      Dim con As New SqlClient.SqlConnection

              Dim connectionString As String = "Server='" &
     My.Settings.mServer & "';

Database='" & My.Settings.mDB & "';

User Id='" & My.Settings.mUserDB & "';

Password='" & My.Settings.mPassDB & "';"

      con.ConnectionString = connectionString

      Dim query As String = String.Empty

              query &= "update truck_details set vendorname=@vendor,
     plate_no=@plate, truck_type=@truck, "

      query &= " truck_load=@truckload, no_of_helper=@helper,
     driver_name=@driver,contact_no=@contact,category=@category
     where Queueno=@queue"
```

```
Using conn As New SqlConnection(connectionString)

    Using comm As New SqlCommand()

        With comm

            .Connection = conn

            .CommandType = CommandType.Text

            .CommandText = query

            .Parameters.AddWithValue("@queue", Label7.Text)

            .Parameters.AddWithValue("@vendor", vendorname.Text)

            .Parameters.AddWithValue("@truck", trucktype.Text)

            .Parameters.AddWithValue("@plate", plateno.Text)

            .Parameters.AddWithValue("@helper", helper.Text)

            .Parameters.AddWithValue("@truckload", truckload.Text)

            .Parameters.AddWithValue("@driver", drivername.Text)

            .Parameters.AddWithValue("@contact", contact.Text)

            .Parameters.AddWithValue("@category", category.Text)

        End With

        Try

            conn.Open()

            comm.ExecuteNonQuery()

            queuetable()

          MsgBox("Successfully Updated, Please Enter DR details",
          MsgBoxStyle.Information)
            truckdetailslist.Enabled = False

            drdetails.Enabled = True

            Savedr.Hide()

            vendorname.Enabled = False

            trucktype.Enabled = False

            plateno.Enabled = False

            helper.Enabled = False

            truckload.Enabled = False

            drivername.Enabled = False

            contact.Enabled = False

            category.Enabled = False

            adddr.Show()

            adddr.Enabled = True

            Label7.Enabled = True

            drno.Enabled = True

            box.Enabled = True

            units.Enabled = True

            update.Hide()

            Button1.Show()


            conn.Close()

        Catch ex As Exception
```

```
      MessageBox.Show(ex.Message.ToString(), "Error Message")

                End Try

                conn.Close()

            End Using

        End Using

    End If

End Sub

Public Sub details()

End Sub

          Private Sub drdetails_CellMouseClick(sender As Object, e As
      DataGridViewCellMouseEventArgs) Handles
      drdetails.CellMouseClick

    Dim con As New SqlClient.SqlConnection

              Dim connectionString As String = "Server='" &
      My.Settings.mServer & "';

    Database='" & My.Settings.mDB & "';

    User Id='" & My.Settings.mUserDB & "';

    Password='" & My.Settings.mPassDB & "';"

    con.ConnectionString = connectionString

    i = drdetails.CurrentRow.Index


    Dim drnoval As String

    Dim boxesval As String
Dim unitsval As String

queueval = drdetails.Item(0, i).Value.ToString

drnoval = drdetails.Item(1, i).Value.ToString

boxesval = drdetails.Item(2, i).Value.ToString

unitsval = drdetails.Item(3, i).Value.ToString

Dim selectquery As String

        selectquery = "select * from drtbl where queueno=@queue
  and drno=@dr and noofboxes=@box and totalunits=@units"

Try

    con.Open()

    cmd = New SqlCommand(selectquery, con)

    cmd.Parameters.AddWithValue("@queue", queueval)

    cmd.Parameters.AddWithValue("@dr", drnoval)

    cmd.Parameters.AddWithValue("@box", boxesval)

    cmd.Parameters.AddWithValue("@units", unitsval)

    cmd.ExecuteNonQuery()

    Dim dt As New DataTable()

    Dim adapter As New SqlDataAdapter(cmd)

    adapter.Fill(dt)

    Dim dr As SqlClient.SqlDataReader

    dr = cmd.ExecuteReader(CommandBehavior.CloseConnection)

    While dr.Read
```

```
            Label7.Text = queueval

            drno.Text = dr("drno").ToString

            box.Text = dr("noofboxes").ToString

            units.Text = dr("totalunits").ToString

            drno.Enabled = True

            box.Enabled = True

            units.Enabled = True

            Savedr.Hide()

            Button1.Show()

            vendorname.Enabled = False

            trucktype.Enabled = False

            plateno.Enabled = False

            helper.Enabled = False

            truckload.Enabled = False

            drivername.Enabled = False

            contact.Enabled = False

            category.Enabled = False

            truckdetailslist.Enabled = False

        End While

    Catch ex As Exception

        MessageBox.Show(ex.Message.ToString(), "Error Message")

    End Try
End Sub

Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click

    Dim dialog As DialogResult

    If drno.Text = "" Then

        Errorlbl.ForeColor = Color.Red

        Errorlbl.Text = "Error: Please Enter Dr No. OR Select on the DR List"

        drno.Focus()

    ElseIf box.Text = "" Then

        Errorlbl.ForeColor = Color.Red

        Errorlbl.Text = "Error: Please Enter No of box."

        box.Focus()

    ElseIf units.Text = "" Then

        Errorlbl.ForeColor = Color.Red

        Errorlbl.Text = "Error: Please Enter Total no. of Units"

        units.Focus()

    Else

        dialog = MessageBox.Show("Are you sure you want to Update DR?", "Update", MessageBoxButtons.YesNo)


    If dialog = DialogResult.Yes Then

            Dim con As New SqlClient.SqlConnection
            Dim connectionString As String = "Server='" &
    My.Settings.mServer & "';
Database='" & My.Settings.mDB & "';
User Id='" & My.Settings.mUserDB & "';
Password='" & My.Settings.mPassDB & "';"
            con.ConnectionString = connectionString
            i = drdetails.CurrentRow.Index
            Dim drnoval As String
            drnoval = drdetails.Item(1, i).Value.ToString
            con.ConnectionString = connectionString
            Dim query As String = String.Empty
                 query &= "UPDATE drtbl Set
         drno=@dr,noofboxes=@box,totalunits=@units where
         queueno=@queue and drno=@dr"
            Using conn As New SqlConnection(connectionString)
              Using comm As New SqlCommand()
                With comm
                  .Connection = conn
                  .CommandType = CommandType.Text
                  .CommandText = query
                  .Parameters.AddWithValue("@queue", Label7.Text)
                  .Parameters.AddWithValue("@dr", drno.Text)
                  .Parameters.AddWithValue("@box", box.Text)
```

```
                .Parameters.AddWithValue("@units", units.Text)
            End With
            Try
                conn.Open()
                comm.ExecuteNonQuery()
MsgBox("Updated Successfully", MsgBoxStyle.Information)
                drtable2()
                drno.Clear()
                box.Clear()
                units.Clear()
                BunifuFlatButton1.Enabled = True
                Button1.Hide()
                done.Enabled = True
                update.Enabled = True
            Catch ex As Exception
MessageBox.Show(ex.Message.ToString(), "Error Message")
            End Try
        End Using
    End Using
    con.Close()
    BunifuFlatButton1.Enabled = True
    Button1.Hide()
    trucktype.Text = ""
Else
    Savedr.Hide()
    adddr.Hide()
    done.Hide()
    edit.Hide()
    update.Hide()
    Button1.Show()
    vendorname.Enabled = True
    trucktype.Enabled = True
    plateno.Enabled = True
    helper.Enabled = True
    truckload.Enabled = True
    drivername.Enabled = True
    contact.Enabled = True
    category.Enabled = True
    vendorname.Text = ""
    trucktype.Text = ""
    plateno.Clear()
    helper.Clear()
    truckload.Text = ""
    drivername.Clear()
```

```
                        contact.Clear()

                        category.Text = ""

                        truckdetailslist.Enabled = True

                        drdetails.Enabled = False

                        drno.Enabled = False

                        box.Enabled = False

                        units.Enabled = False

                        drno.Clear()

                        box.Clear()

                        units.Clear()

                        Button1.Hide()

                        BunifuFlatButton1.Enabled = True

                        Savedr.Hide()

                    End If

                End If

            End Sub

            Private Sub Contact_TextChanged(sender As Object, e As EventArgs)
Handles contact.TextChanged

                contact.MaxLength = 11

            End Sub

            Private Sub contact_LostFocus(sender As Object, e As EventArgs)
        Handles contact.LostFocus
        Dim strAllowedChars As String = "0123456789"

        If Len(contact.Text) = 0 Then

            Exit Sub

        Else

            For i = 0 To Len(contact.Text) - 1

                If InStr(1, strAllowedChars, contact.Text(i)) = 0 Then

                    Errorlbl.Text = "Invalid Number"

                    Errorlbl.ForeColor = Color.Red

                    contact.Focus()

                    Exit Sub

                End If

            Next

        End If

    End Sub

    Private Sub contact_GotFocus(sender As Object, e As EventArgs)
Handles contact.GotFocus

        If contact.TextLength.Equals(0) Then

            contact.Text = "09"

        End If

    End Sub

               Private Sub drno_KeyPress(sender As Object, e As
           KeyPressEventArgs) Handles drno.KeyPress
```

```
   Select Case e.KeyChar
              Case "0", "1", "2", "3", "4", "5", "6", "7", "8", "9", vbBack
         e.Handled = False
      Case Else
         e.Handled = True
   End Select
End Sub
        Private Sub box_KeyPress(sender As Object, e As
     KeyPressEventArgs) Handles box.KeyPress
   Select Case e.KeyChar
              Case "0", "1", "2", "3", "4", "5", "6", "7", "8", "9", vbBack
         e.Handled = False
      Case Else
         e.Handled = True
   End Select
End Sub
Private Sub units_KeyPress(sender As Object, e As
KeyPressEventArgs) Handles units.KeyPress
   Select Case e.KeyChar
              Case "0", "1", "2", "3", "4", "5", "6", "7", "8", "9",
         e.Handled = False
      Case Else
```

```
                e.Handled = True

        End Select

    End Sub

    Private Sub helper_KeyPress(sender As Object, e As
KeyPressEventArgs) Handles helper.KeyPress

        Select Case e.KeyChar

            Case "0", "1", "2", "3", "4", "5", "6", "7", "8", "9", vbBack

                e.Handled = False

            Case Else

                e.Handled = True

        End Select

    End Sub

    Private Sub contact_KeyPress(sender As Object, e As
KeyPressEventArgs) Handles contact.KeyPress

        Select Case e.KeyChar

            Case "0", "1", "2", "3", "4", "5", "6", "7", "8", "9", vbBack

                e.Handled = False

            Case Else

                e.Handled = True

        End Select

    End Sub

    Private Sub Cancel_Click(sender As Object, e As EventArgs) Handles
Cancel.Click
Cancel.Hide()

Errorlbl.Text = ""

BunifuFlatButton1.Enabled = True

Savedr.Enabled = False

queuetable()

drtable()

drdetails.Show()

Label16.Hide()

'Label17.Hide()

Label7.Hide()

drno.Enabled = False

box.Enabled = False

units.Enabled = False

Savedr.Show()

adddr.Hide()

done.Hide()

edit.Hide()

update.Hide()

Button1.Hide()

drno.Text = ""

box.Text = ""

units.Text = ""
```

```
        drdetails.Enabled = False
        truckdetailslist.Enabled = True
        vendorname.Enabled = False
        trucktype.Enabled = False
        plateno.Enabled = False
        helper.Enabled = False
        truckload.Enabled = False
        drivername.Enabled = False
        contact.Enabled = False
        category.Enabled = False
        vendorname.Text = ""
        trucktype.SelectedIndex = -1
        plateno.Clear()
        helper.Clear()
        truckload.SelectedIndex = -1
        drivername.Clear()
        contact.Clear()
        category.SelectedIndex = -1
    End Sub


    Private Sub Logout_Click(sender As Object, e As EventArgs) Handles
Logout.Click
```

```
        Dim dialog As DialogResult

        dialog = MessageBox.Show("Are you sure you want to Logout?",
"Exit", MessageBoxButtons.YesNo)

        If dialog = DialogResult.No Then

        Else

            Dim logout As New DBconnection

            logout.ReceiverName = ""

            Me.Close()

            Login.Show()

        End If

    End Sub

    Private Sub Timer1_Tick(sender As Object, e As EventArgs) Handles
Timer1.Tick

        Label5.Text = DateTime.Now.ToString("MMMM dd, yyyy h:mm:ss tt")

    End Sub

End Class
```

DRchecking.vb

```
Imports System.Text.RegularExpressions

Imports System.Data.SqlClient

Imports System.Data

Public Class DRchecking

    Dim con As SqlConnection
    Dim cmd As SqlCommand

    Dim cmd1 As SqlCommand

    Dim cmd2 As SqlCommand

    Dim cmd3 As SqlCommand

    Dim cmd4 As SqlCommand

    Dim cmd5 As SqlCommand

    Dim adapter As SqlDataAdapter

    Dim dr As SqlDataReader

    Dim dr1 As SqlDataReader

    Dim table As New DataTable

    Public dbs As New BindingSource

    Public dbs1 As New BindingSource

    Public dbs2 As New BindingSource

    Public dbs3 As New BindingSource

    Dim i As Integer

    Private Sub Form4_Load(sender As Object, e As EventArgs) Handles MyBase.Load

        Pendinglist.GridColor = Color.Red

        queuelist.GridColor = Color.Red

        Me.FormBorderStyle = FormBorderStyle.None

        queuelist.RowHeadersVisible = False

        Pendinglist.RowHeadersVisible = False
        DataGridView4.RowHeadersVisible = False

        LoadTable()

        ' LoadVerificationTable()

        LoadPendingTable()

        Dim myname As New DBconnection

        Name = myname.ReceiverName

        Label9.Text = Name

        Queuenotxt.Enabled = False

        vendortxt.Enabled = False

        categorytxt.Enabled = False

        countofdrtxt.Enabled = False

        noofboxtxt.Enabled = False

        totalunittxt.Enabled = False

        Button2.Enabled = False

        Button3.Enabled = False

    End Sub

    Public Sub LoadTable()

        Dim con As New SqlClient.SqlConnection

        Dim connectionString As String = "Server='" & My.Settings.mServer
& "';

    Database='" & My.Settings.mDB & "';

    User Id='" & My.Settings.mUserDB & "';
```

```
Password='" & My.Settings.mPassDB & "';"

   Dim queueno As String = "On process"

   con.ConnectionString = connectionString

     Dim selectquery As String = "select Queueno as 'Queue Number'
     from Queue_tbl where status=@queueno"

   Try

      con.Open()

      cmd = New SqlCommand(selectquery, con)

      cmd.Parameters.AddWithValue("@queueno", queueno)

      Dim da As New SqlDataAdapter(cmd)

      Dim dt As New DataTable()

      da.Fill(dt)

      dbs.DataSource = dt

      queuelist.DataSource = dbs

      queuelist.ReadOnly = True

      con.Close()

   Catch ex As Exception

      MessageBox.Show(ex.Message.ToString(), "Error Message")

      con.Close()

   End Try


End Sub
```

```
Public Sub LoadPendingTable()

   Dim con As New SqlClient.SqlConnection

     Dim connectionString As String = "Server='" &
     My.Settings.mServer & "';

Database='" & My.Settings.mDB & "';

User Id='" & My.Settings.mUserDB & "';

Password='" & My.Settings.mPassDB & "';"

   Dim queueno As String = "Pending"

   con.ConnectionString = connectionString

     Dim selectquery As String = "select Queueno as 'Queue Number'
     from Queue_tbl where status=@queueno"

   Try

      con.Open()

      cmd = New SqlCommand(selectquery, con)

      cmd.Parameters.AddWithValue("@queueno", queueno)

      Dim da As New SqlDataAdapter(cmd)

      Dim dt As New DataTable()

      da.Fill(dt)

      dbs3.DataSource = dt

      Pendinglist.DataSource = dbs3

      Pendinglist.ReadOnly = True

      con.Close()
   Catch ex As Exception

      MessageBox.Show(ex.Message.ToString(), "Error Message")

      con.Close()

   End Try

End Sub

     Private Sub Callbtn_Click(sender As Object, e As EventArgs)
     Handles Callbtn.Click

   Callbtn.Enabled = False

      Button4.Enabled = False

      Button2.Enabled = True

      Button3.Enabled = True

      Dim queueno As String

      queueno = queuelist.Item(0, 0).Value.ToString

      Dim con As New SqlClient.SqlConnection

     Dim connectionString As String = "Server='" &
     My.Settings.mServer & "';

Database='" & My.Settings.mDB & "';

User Id='" & My.Settings.mUserDB & "';

     Password='" & My.Settings.mPassDB & "';
     MultipleActiveResultSets=True"



      con.ConnectionString = connectionString
```

```
 i = queuelist.CurrentRow.Index

Dim selectquery As String = "select q.Queueno, q.Status, t.vendorname, t.category, d.drno from Queue_tbl q inner join truck_details t on q.Queueno = t.Queueno inner join drtbl d on t.Queueno = d.Queueno where q.Queueno =@queueno "

Dim drquery As String = "select queueno,drno, noofboxes, totalunits from drtbl where queueno=@queueno"

Dim drupdatequery As String = "UPDATE Queue_tbl Set Status=@Status, StartVerify=convert(time, getdate()) where queueno=@queueno"

Dim countquery As String = "select sum(noofboxes) totalbox, sum(totalunits) totalunit from drtbl where queueno=@queueno"

Dim countdr As String = "SELECT count(*) as drcount from drtbl where queueno = @queueno"

 Try

 con.Open()

    cmd = New SqlCommand(selectquery, con)

    cmd.Parameters.AddWithValue("@queueno", queueno)

    cmd1 = New SqlCommand(drquery, con)

    cmd1.Parameters.AddWithValue("@queueno", queueno)

    cmd2 = New SqlCommand(drupdatequery, con)

    cmd2.Parameters.AddWithValue("@queueno", queueno)

    cmd2.Parameters.AddWithValue("@Status", "Verification")

    cmd3 = New SqlCommand(countquery, con)

    cmd3.Parameters.AddWithValue("@queueno", queueno)
    cmd4 = New SqlCommand(countdr, con)
    cmd4.Parameters.AddWithValue("@queueno", queueno)
    cmd.ExecuteNonQuery()
    cmd1.ExecuteNonQuery()
    cmd2.ExecuteNonQuery()
    cmd3.ExecuteNonQuery()
    cmd4.ExecuteNonQuery()
    Dim dt As New DataTable()
    Dim dt1 As New DataTable()
    Dim adapter As New SqlDataAdapter(cmd)
    Dim adapter1 As New SqlDataAdapter(cmd1)
    adapter.Fill(dt)
    adapter1.Fill(dt1)
    Dim dr As SqlClient.SqlDataReader
dr = cmd.ExecuteReader(CommandBehavior.CloseConnection)
    While dr.Read
        Queuenotxt.Text = dr("Queueno").ToString
        vendortxt.Text = dr("vendorname").ToString
        categorytxt.Text = dr("category").ToString
    End While
    Dim dr1 As SqlClient.SqlDataReader
dr1 = cmd3.ExecuteReader(CommandBehavior.CloseConnection)
```

```
         While dr1.Read

            noofboxtxt.Text = dr1("totalbox").ToString

            totalunittxt.Text = dr1("totalunit").ToString

         End While

         Dim dr2 As SqlClient.SqlDataReader

     dr2 = cmd4.ExecuteReader(CommandBehavior.CloseConnection)

         While dr2.Read

            countofdrtxt.Text = dr2("drcount").ToString

         End While

         dbs1.DataSource = dt1

         DataGridView4.DataSource = dbs1

         DataGridView4.ReadOnly = True

         con.Close()

      Catch ex As Exception

         MessageBox.Show(ex.Message.ToString(), "Error Message")

         con.Close()

      End Try

End Sub

     Private Sub Timer1_Tick(sender As Object, e As EventArgs)
     Handles Timer1.Tick

   LoadTable()

   LoadPendingTable()
```

```
      Label16.Text = DateTime.Now.ToString("MMMM dd, yyyy
      h:mm:ss tt")

    If queuelist.Item(0, 0).Value Is Nothing Then

       Callbtn.Enabled = False

    ElseIf Queuenotxt.TextLength > 0 Then

       Callbtn.Enabled = False

    Else

       Callbtn.Enabled = True

    End If

    If Pendinglist.Item(0, 0).Value Is Nothing Then

       Button4.Enabled = False

    ElseIf Queuenotxt.TextLength > 0 Then

       Button4.Enabled = False

    Else

       Button4.Enabled = True

    End If

End Sub

      Private Sub Button3_Click(sender As Object, e As EventArgs)
      Handles Button3.Click

      If TextBox10.Text = "" Or TextBox11.Text = "" Or TextBox9.Text =
      "" Or TextBox12.Text = "" Or TextBox13.Text = "" Then

       Errorlbl.Text = "Please input Pending Details, Put 0 if none"

       Errorlbl.ForeColor = Color.Red
    Else

        Errorlbl.Text = ""

        Dim con As New SqlClient.SqlConnection

      Dim connectionString As String = "Server='" &
      My.Settings.mServer & "';

Database='" & My.Settings.mDB & "';

User Id='" & My.Settings.mUserDB & "';

          Password='" & My.Settings.mPassDB &
      "';MultipleActiveResultSets=True"

        con.ConnectionString = connectionString

      Dim query As String = "update truck_details Set
      advancedel=@advancedel, shortqty=@shortqty,
      overqty=@overqty, reenterpo=@reenterpo, unknown=@unknown
      where queueno=@queueno"

      Dim statquery As String = "update Queue_tbl Set Status=@status,
      Pending='Yes' where queueno=@queueno"

      Dim selectqueuecontact As String = "select contact_no from
      truck_details where Queueno=@queueno"

        Dim updatequerystat As String = String.Empty

      updatequerystat &= "update Queue_tbl set
      NotifPendingStatus=@notifstat where Queueno=@queueno"

        Using conn As New SqlConnection(connectionString)

            Using comm As New SqlCommand()

                With comm

                    .Connection = conn
          .CommandType = CommandType.Text
          .CommandText = query
.Parameters.AddWithValue("@advancedel", TextBox10.Text)
          .Parameters.AddWithValue("@shortqty", TextBox11.Text)
          .Parameters.AddWithValue("@overqty", TextBox9.Text)
.Parameters.AddWithValue("@reenterpo", TextBox12.Text)
.Parameters.AddWithValue("@unknown", TextBox13.Text)
.Parameters.AddWithValue("@queueno", Queuenotxt.Text)
       End With
       Try
          conn.Open()
          comm.ExecuteNonQuery()
MsgBox("Pending Successfully", MsgBoxStyle.Information)
          LoadTable()
          conn.Close()
       Catch ex As Exception
MessageBox.Show(ex.Message.ToString(), "Error Message")
          conn.Close()
       End Try
    End Using
    conn.Open()
   Dim queueno As String = Queuenotxt.Text
```

```
        cmd2 = New SqlCommand(selectqueuecontact, conn)

        cmd2.Parameters.AddWithValue("@queueno", queueno)

        cmd2.ExecuteNonQuery()

        Dim dr As SqlClient.SqlDataReader

   dr = cmd2.ExecuteReader(CommandBehavior.CloseConnection)

        While dr.Read

            Dim classpublic As New DBconnection

            Dim contactnumber As String = dr("contact_no").ToString

   Dim smsmessage As String = ("Your Queue No. is: " + queueno +
   " Your Status is PENDING, Please Proceed to Verification Office
   and look for " + classpublic.ReceiverName)

            Try

   Dim res As String = itexmo(contactnumber, smsmessage,
   My.Settings.smsAPI, "(z}v7vymks")

                If res = "0" Then

   MsgBox("Queue Message Has been Sent to the Driver!")

cmd3 = New SqlCommand(updatequerystat, conn)

cmd3.Parameters.AddWithValue("@notifstat", "Message Sent!")

cmd3.Parameters.AddWithValue("@queueno", queueno)

                    Try

                        cmd3.ExecuteNonQuery()

                    Catch ex As Exception

                        MsgBox(ex.ToString + "Update Error")
                End Try
            Else
MsgBox("Error on Sending Message, Please Contact
Administrator")
cmd3 = New SqlCommand(updatequerystat, conn)
cmd3.Parameters.AddWithValue("@notifstat", "Message NOT
Sent!")
cmd3.Parameters.AddWithValue("@queueno", queueno)
                Try
                    cmd3.ExecuteNonQuery()
                Catch ex As Exception
                    MsgBox(ex.ToString + "Update Error")
                End Try
            End If
        Catch ex As Exception
MsgBox("Error on Sending Message, Please Contact
Administrator")
cmd3 = New SqlCommand(updatequerystat, conn)
cmd3.Parameters.AddWithValue("@notifstat", "Message NOT
Sent!")
cmd3.Parameters.AddWithValue("@queueno", Queuenotxt.Text)
            Try
                cmd3.ExecuteNonQuery()
            Catch ex1 As Exception
                MsgBox(ex1.ToString + "Update Error")
            End Try
        End Try
    End While
    conn.Close()
    Using comm1 As New SqlCommand()
        With comm1
            .Connection = conn
            .CommandType = CommandType.Text
            .CommandText = statquery
            .Parameters.AddWithValue("@status", "Pending")
.Parameters.AddWithValue("@queueno", Queuenotxt.Text)
        End With
        Try
            conn.Open()
            comm1.ExecuteNonQuery()
MsgBox("Pending Successfully", MsgBoxStyle.Information)
            LoadTable()
            Callbtn.Enabled = True
            Button3.Enabled = False
            Button2.Enabled = False
```

```
                    Button4.Enabled = True
                    Queuenotxt.Clear()
                    vendortxt.Clear()
                    categorytxt.Clear()
                    countofdrtxt.Clear()
                    noofboxtxt.Clear()
                    totalunittxt.Clear()
                    TextBox10.Clear()
                    TextBox11.Clear()
                    TextBox9.Clear()
                    TextBox12.Clear()
                    TextBox13.Clear()
                    DataGridView4.DataSource = DBNull.Value
                    conn.Close()
                Catch ex As Exception
       MessageBox.Show(ex.Message.ToString(), "Error Message")
                    conn.Close()
                End Try
            End Using
        End Using
    End If
End Sub
```

```
    Private Sub Button4_Click(sender As Object, e As EventArgs)
    Handles Button4.Click

  Callbtn.Enabled = False

  Button4.Enabled = False

  Button2.Enabled = True

  Button3.Enabled = True

  Dim queueno As String

  queueno = Pendinglist.Item(0, 0).Value.ToString

  Dim con As New SqlClient.SqlConnection

    Dim connectionString As String = "Server='" &
    My.Settings.mServer & "';

Database='" & My.Settings.mDB & "';

User Id='" & My.Settings.mUserDB & "';

    Password='" & My.Settings.mPassDB & "';
    MultipleActiveResultSets=True"

  con.ConnectionString = connectionString

    Dim selectquery As String = "select q.Queueno, q.Status,
    t.vendorname, t.category, t.advancedel, t.shortqty, t.overqty,
    t.reenterpo, t.unknown, d.drno from Queue_tbl q inner join
    truck_details t on q.Queueno = t.Queueno inner join drtbl d on
    t.Queueno = d.Queueno where q.Queueno =@queueno "

    Dim drquery As String = "select queueno,drno, noofboxes,
    totalunits from drtbl where queueno=@queueno"

    Dim drupdatequery As String = "UPDATE Queue_tbl Set
    Status=@Status  where queueno=@queueno"
  Dim countquery As String = "select sum(noofboxes) totalbox, sum(totalunits) totalunit from drtbl where queueno=@queueno"

  Dim countdr As String = "SELECT count(*) as drcount from drtbl where queueno = @queueno"

Try

   con.Open()

   cmd = New SqlCommand(selectquery, con)

   cmd.Parameters.AddWithValue("@queueno", queueno)

   cmd1 = New SqlCommand(drquery, con)

   cmd1.Parameters.AddWithValue("@queueno", queueno)

   cmd2 = New SqlCommand(drupdatequery, con)

   cmd2.Parameters.AddWithValue("@queueno", queueno)

   cmd2.Parameters.AddWithValue("@Status", "Verification")

   cmd3 = New SqlCommand(countquery, con)

   cmd3.Parameters.AddWithValue("@queueno", queueno)

   cmd4 = New SqlCommand(countdr, con)

   cmd4.Parameters.AddWithValue("@queueno", queueno)

   cmd.ExecuteNonQuery()

   cmd1.ExecuteNonQuery()

   cmd2.ExecuteNonQuery()

   cmd3.ExecuteNonQuery()

   cmd4.ExecuteNonQuery()
```

```
Dim dt As New DataTable()

Dim dt1 As New DataTable()

Dim adapter As New SqlDataAdapter(cmd)

Dim adapter1 As New SqlDataAdapter(cmd1)

adapter.Fill(dt)

adapter1.Fill(dt1)

Dim dr As SqlClient.SqlDataReader

dr = cmd.ExecuteReader(CommandBehavior.CloseConnection)

While dr.Read

    Queuenotxt.Text = dr("Queueno").ToString

    vendortxt.Text = dr("vendorname").ToString

    categorytxt.Text = dr("category").ToString

    TextBox10.Text = dr("advancedel").ToString

    TextBox11.Text = dr("shortqty").ToString

    TextBox9.Text = dr("overqty").ToString

    TextBox12.Text = dr("reenterpo").ToString

    TextBox13.Text = dr("unknown").ToString

End While

Dim dr1 As SqlClient.SqlDataReader

dr1 = cmd3.ExecuteReader(CommandBehavior.CloseConnection)

While dr1.Read

    noofboxtxt.Text = dr1("totalbox").ToString
```

```
            totalunittxt.Text = dr1("totalunit").ToString
         End While
         Dim dr2 As SqlClient.SqlDataReader
        dr2 = cmd4.ExecuteReader(CommandBehavior.CloseConnection)
         While dr2.Read
            countofdrtxt.Text = dr2("drcount").ToString
         End While
         dbs1.DataSource = dt1
         DataGridView4.DataSource = dbs1
         DataGridView4.ReadOnly = True
         con.Close()
    Catch ex As Exception
         MessageBox.Show(ex.Message.ToString(), "Error Message")
         con.Close()
    End Try
End Sub
        Private Sub Button2_Click(sender As Object, e As EventArgs)
        Handles Button2.Click
    Dim getname As New DBconnection
    Dim name As String = getname.ReceiverName
    Dim con As New SqlClient.SqlConnection
```

```
          Dim connectionString As String = "Server='" &
          My.Settings.mServer & "';

 Database='" & My.Settings.mDB & "';

 User Id='" & My.Settings.mUserDB & "';

          Password='" & My.Settings.mPassDB & "';
          MultipleActiveResultSets=True"

      con.ConnectionString = connectionString

          Dim query As String = "update truck_details Set
          countofdr=@countofdr, totalnoofboxes=@totalnoofboxes,
          totaltotalunits=@totaltotalunits, advancedel=@advancedel,
          shortqty=@shortqty, overqty=@overqty, reenterpo=@reenterpo,
          unknown=@unknown where queueno=@queueno"

          Dim statquery As String = "update Queue_tbl Set Status=@status,
          VerifiedBy=@verifiedby, EndVerify=convert(time, getdate()) where
          queueno=@queueno"

          Dim insertaerosolquery As String = "insert into Bldg6
          (queueno,Bldg,Dock) values(@queueno,@bldg,@dock)"

          Dim insertcftquery As String = "insert into Bldg3
          (Queueno,Bldg,Dock) values(@queueno,@bldg,@dock)"

          Dim insertdr1query As String = "insert into bldg4d1
          (queueno,Bldg,Dock) values(@queueno,@bldg,@dock)"

          Dim insertdr2query As String = "insert into bldg4d2
          (queueno,Bldg,Dock) values(@queueno,@bldg,@dock)"

          Dim insertdr3query As String = "insert into Bldg4d3
          (queueno,Bldg,Dock) values(@queueno,@bldg,@dock)"

          Dim insertdr4query As String = "insert into Bldg4d4
          (queueno,Bldg,Dock) values(@queueno,@bldg,@dock)"
```

```
  Dim countquery As String = "SELECT  (SELECT COUNT(*)
  FROM   bldg4d1) AS d1, (SELECT COUNT(*) FROM   bldg4d2)
  AS d2,(SELECT COUNT(*)FROM   bldg4d3) AS d3,(SELECT
  COUNT(*)FROM   bldg4d4) AS d4"

Using conn As New SqlConnection(connectionString)

   Using comm As New SqlCommand()

        With comm

            .Connection = conn

            .CommandType = CommandType.Text

            .CommandText = query

  .Parameters.AddWithValue("@countofdr", countofdrtxt.Text)

  .Parameters.AddWithValue("@totalnoofboxes", noofboxtxt.Text)

  .Parameters.AddWithValue("@totaltotalunits", totalunittxt.Text)

  .Parameters.AddWithValue("@advancedel", TextBox10.Text)

  .Parameters.AddWithValue("@shortqty", TextBox11.Text)

  .Parameters.AddWithValue("@overqty", TextBox9.Text)

  .Parameters.AddWithValue("@reenterpo", TextBox12.Text)

  .Parameters.AddWithValue("@unknown", TextBox13.Text)

  .Parameters.AddWithValue("@queueno", Queuenotxt.Text)

        End With

        Try

            conn.Open()

            comm.ExecuteNonQuery()
```

```
MsgBox("Verified Successfully", MsgBoxStyle.Information)
                   LoadTable()
                   conn.Close()
               Catch ex As Exception
  MessageBox.Show(ex.Message.ToString(), "Error Message")
                   conn.Close()
               End Try
           End Using
           Using comm1 As New SqlCommand()
               With comm1
                   .Connection = conn
                   .CommandType = CommandType.Text
                   .CommandText = statquery
                   .Parameters.AddWithValue("@status", "Verified")
               .Parameters.AddWithValue("@queueno", Queuenotxt.Text)
               .Parameters.AddWithValue("@verifiedby", name)
           End With
               Try
                   conn.Open()
                   comm1.ExecuteNonQuery()
  MsgBox("Verified Successfully", MsgBoxStyle.Information)
                   LoadTable()
```

```
                conn.Close()

            Catch ex As Exception

  MessageBox.Show(ex.Message.ToString(), "Error Message")

                conn.Close()

            End Try

            conn.Close()

        End Using

        conn.Close()

    If categorytxt.Text = "AEROSOL" Then

        Using comm2 As New SqlCommand()

            With comm2

                .Connection = conn

                .CommandType = CommandType.Text

                .CommandText = insertaerosolquery

.Parameters.AddWithValue("@queueno", Queuenotxt.Text)

.Parameters.AddWithValue("@bldg", "6")

.Parameters.AddWithValue("@dock", "1")

            End With

            Try

                conn.Open()

                comm2.ExecuteNonQuery()

MsgBox("Verified Successfully", MsgBoxStyle.Information)
```

```
          LoadTable()

          Callbtn.Enabled = True

          Button3.Enabled = False

          Button2.Enabled = False

          Button4.Enabled = True

          Queuenotxt.Clear()

          vendortxt.Clear()

          categorytxt.Clear()

          countofdrtxt.Clear()

          noofboxtxt.Clear()

          totalunittxt.Clear()

          TextBox10.Clear()

          TextBox11.Clear()

          TextBox9.Clear()

          TextBox12.Clear()

          TextBox13.Clear()

          DataGridView4.DataSource = DBNull.Value

          conn.Close()

        Catch ex As Exception

MessageBox.Show(ex.Message.ToString(), "Error Message")

          conn.Close()

        End Try
        conn.Close()
    End Using
    conn.Close()
ElseIf categorytxt.Text = "CFT" Then
    Using comm3 As New SqlCommand()
        With comm3
            .Connection = conn
            .CommandType = CommandType.Text
            .CommandText = insertcftquery
.Parameters.AddWithValue("@queueno", Queuenotxt.Text)
.Parameters.AddWithValue("@bldg", "3")
.Parameters.AddWithValue("@dock", "1")
        End With
        Try
            conn.Open()
            comm3.ExecuteNonQuery()
MsgBox("Verified Successfully", MsgBoxStyle.Information)
            LoadTable()
            Callbtn.Enabled = True
            Button3.Enabled = False
            Button2.Enabled = False
            Button4.Enabled = True
        Queuenotxt.Clear()

        vendortxt.Clear()

        categorytxt.Clear()

        countofdrtxt.Clear()

        noofboxtxt.Clear()

        totalunittxt.Clear()

        TextBox10.Clear()

        TextBox11.Clear()

        TextBox9.Clear()

        TextBox12.Clear()

        TextBox13.Clear()

        DataGridView4.DataSource = DBNull.Value


        conn.Close()

    Catch ex As Exception

 MessageBox.Show(ex.Message.ToString(), "Error Message")

        conn.Close()

    End Try

    conn.Close()

End Using

conn.Close()
```

```
    ElseIf categorytxt.Text = "NCFT" Then
        Dim dr1 As String
        Dim dr2 As String
        Dim dr3 As String
        Dim dr4 As String
        Try
            conn.Open()
            cmd4 = New SqlCommand(countquery, conn)
            cmd4.ExecuteNonQuery()
            Dim dr As SqlClient.SqlDataReader
dr = cmd4.ExecuteReader(CommandBehavior.CloseConnection)
            While dr.Read
                dr1 = dr("d1").ToString
                dr2 = dr("d2").ToString
                dr3 = dr("d3").ToString
                dr4 = dr("d4").ToString
                If (dr1 = dr2 And dr1 = dr3 And dr1 = dr4) Then
    cmd5 = New SqlCommand(insertdr1query, conn)
    cmd5.Parameters.AddWithValue("@queueno", Queuenotxt.Text)
    cmd5.Parameters.AddWithValue("@bldg", "4")
    cmd5.Parameters.AddWithValue("@dock", "1")
     cmd5.ExecuteNonQuery()
```

```
            Callbtn.Enabled = True
            Button4.Enabled = True
            Button2.Enabled = False
            Button3.Enabled = False
            Queuenotxt.Clear()
            vendortxt.Clear()
            categorytxt.Clear()
            countofdrtxt.Clear()
            noofboxtxt.Clear()
            totalunittxt.Clear()
            TextBox10.Clear()
            TextBox11.Clear()
            TextBox9.Clear()
            TextBox12.Clear()
            TextBox13.Clear()
            DataGridView4.DataSource = DBNull.Value
        ElseIf (dr1 > dr2 And dr2 = dr3 And dr2 = dr4) Then
cmd5 = New SqlCommand(insertdr2query, conn)
cmd5.Parameters.AddWithValue("@queueno", Queuenotxt.Text)
cmd5.Parameters.AddWithValue("@bldg", "4")
 cmd5.Parameters.AddWithValue("@dock", "2")
 cmd5.ExecuteNonQuery()
```

```
          Callbtn.Enabled = True

          Button4.Enabled = True

          Button2.Enabled = False

          Button3.Enabled = False

          Queuenotxt.Clear()

          vendortxt.Clear()

          categorytxt.Clear()

          countofdrtxt.Clear()

          noofboxtxt.Clear()

          totalunittxt.Clear()

          TextBox10.Clear()

          TextBox11.Clear()

          TextBox9.Clear()

          TextBox12.Clear()

          TextBox13.Clear()

          DataGridView4.DataSource = DBNull.Value

       ElseIf (dr1 = dr2 And dr2 > dr3 And dr3 = dr4) Then

cmd5 = New SqlCommand(insertdr3query, conn)

cmd5.Parameters.AddWithValue("@queueno", Queuenotxt.Text)

cmd5.Parameters.AddWithValue("@bldg", "4")

 cmd5.Parameters.AddWithValue("@dock", "3")

 cmd5.ExecuteNonQuery()
```

```
                Callbtn.Enabled = True

                Button4.Enabled = True

                Button2.Enabled = False

                Button3.Enabled = False

                Queuenotxt.Clear()

                vendortxt.Clear()

                categorytxt.Clear()

                countofdrtxt.Clear()

                noofboxtxt.Clear()

                totalunittxt.Clear()

                TextBox10.Clear()

                TextBox11.Clear()

                TextBox9.Clear()

                TextBox12.Clear()

                TextBox13.Clear()

                DataGridView4.DataSource = DBNull.Value

            ElseIf (dr1 = dr2 And dr2 = dr3 And dr3 > dr4) Then

cmd5 = New SqlCommand(insertdr4query, conn)

cmd5.Parameters.AddWithValue("@queueno", Queuenotxt.Text)

cmd5.Parameters.AddWithValue("@bldg", "4")

cmd5.Parameters.AddWithValue("@dock", "4")

 cmd5.ExecuteNonQuery()
```

```
                Callbtn.Enabled = True

                Button4.Enabled = True

                Button2.Enabled = False

                Button3.Enabled = False

                Queuenotxt.Clear()

                vendortxt.Clear()

                categorytxt.Clear()

                countofdrtxt.Clear()

                noofboxtxt.Clear()

                totalunittxt.Clear()

                TextBox10.Clear()

                TextBox11.Clear()

                TextBox9.Clear()

                TextBox12.Clear()

                TextBox13.Clear()

                dbs1.Clear()

                DataGridView4.DataSource = DBNull.Value

            End If

        End While

        conn.Close()

    Catch ex As Exception

MessageBox.Show(ex.Message.ToString(), "Error Message")
```

```
            End Try
            conn.Close()
        End If
    End Using
    con.Close()
End Sub
      Private Sub Logout_Click(sender As Object, e As EventArgs) Handles Logout.Click
    If Queuenotxt.TextLength > 0 Then
      MessageBox.Show("You are not allowed to Logout while you're still in Process")
    Else
        Dim dialog As DialogResult
      dialog = MessageBox.Show("Are you sure you want to Logout?", "Exit", MessageBoxButtons.YesNo)
        If dialog = DialogResult.No Then
        Else
            Dim logout As New DBconnection
            logout.ReceiverName = ""
            Me.Close()
            Login.Show()
        End If
    End If
```

```
 End Sub
End Class
```

Receiver.vb

```
Imports System.Text.RegularExpressions
Imports System.Data.SqlClient
Imports System.Data
Public Class Receiver
Dim con As SqlConnection
Dim cmd As SqlCommand
Dim cmd1 As SqlCommand
Dim cmd2 As SqlCommand
Dim cmd3 As SqlCommand
Dim cmd4 As SqlCommand
Dim cmd5 As SqlCommand
Dim cmd6 As SqlCommand
Dim cmd7 As SqlCommand
Dim adapter As SqlDataAdapter
Dim dr As SqlDataReader
Dim dr1 As SqlDataReader
Dim table As New DataTable
Public dbs As New BindingSource
Public dbs1 As New BindingSource
Public dbs2 As New BindingSource

Public dbs3 As New BindingSource

Public dbs4 As New BindingSource

Dim i As Integer

Dim receiverpublic As New DBconnection

Public Sub LoadNewTable()

Dim con As New SqlClient.SqlConnection

Dim connectionString As String = "Server='" & My.Settings.mServer
& "';

 Database='" & My.Settings.mDB & "';

 User Id='" & My.Settings.mUserDB & "';

 Password='" & My.Settings.mPassDB & "';
 MultipleActiveResultSets=True"

        Dim selectquery As String = "select drno as 'DR No.'
 ,actualnoofboxes as 'No. of Boxes',weight as 'Weight per Box',
 actualtotalunits as 'No. of Units' from drtbl where
 queueno=@queueno"

        con.ConnectionString = connectionString

        Using conn As New SqlConnection(connectionString)

            Try

                conn.Open()

 cmd5 = New SqlCommand(selectquery, conn)

 cmd5.Parameters.AddWithValue("@queueno", Label15.Text)

 Dim da As New SqlDataAdapter(cmd5)
```

```
            Dim dt As New DataTable()

            da.Fill(dt)

            dbs4.DataSource = dt

            DataGridView1.DataSource = dbs4

            DataGridView1.ReadOnly = True

Catch ex As Exception

MessageBox.Show(ex.Message.ToString(), "Error Message")

          End Try

      conn.Close()

    End Using

  End Sub

  Public Sub LoadTable()

 Dim con As New SqlClient.SqlConnection

 Dim connectionString As String = "Server='" &
 My.Settings.mServer & "';

  Database='" & My.Settings.mDB & "';

  User Id='" & My.Settings.mUserDB & "';

  Password='" & My.Settings.mPassDB & "';"

    con.ConnectionString = connectionString

Dim selectquery As String = "select Queueno as 'Queue No.',
ReceiverName as 'Receiver Name', Bldg as 'Building No.', Dock as
'Dock No.' from NoShow"
```

```
        Try

            con.Open()

            cmd = New SqlCommand(selectquery, con)

            Dim da As New SqlDataAdapter(cmd)

            Dim dt As New DataTable()

            da.Fill(dt)

            dbs1.DataSource = dt

            NoShowlist.DataSource = dbs1

            NoShowlist.ReadOnly = True

            con.Close()

        Catch ex As Exception

            MessageBox.Show(ex.Message.ToString(), "Error
Message")

        End Try

    End Sub

    Private Sub Button2_Click(sender As Object, e As EventArgs)
Handles Button2.Click

        Dim con As New SqlClient.SqlConnection

        Dim connectionString As String = "Server='" &
My.Settings.mServer & "';

    Database='" & My.Settings.mDB & "';

    User Id='" & My.Settings.mUserDB & "';
```

```
   Password='" & My.Settings.mPassDB & "';
MultipleActiveResultSets=True"

        con.ConnectionString = connectionString

        Dim query As String = "select top 1 * from (select a.* from
Bldg3 a where ReceiverName is NULL union select b.* from Bldg6
b where ReceiverName is NULL union select c.* from bldg4d1 c
where ReceiverName is NULL union select d.* from bldg4d2 d
where ReceiverName is NULL union select e.* from Bldg4d3 e
where ReceiverName is NULL union select f.* from Bldg4d4 f where
ReceiverName is NULL) as t"

        Dim bldg4dr1query As String = "update bldg4d1 Set
ReceiverName=@receivername where queueno=@queueno AND
Bldg=@bldg AND Dock=@dock"

        Dim bldg4dr2query As String = "update bldg4d2 Set
ReceiverName=@receivername where queueno=@queueno AND
Bldg=@bldg AND Dock=@dock"

        Dim bldg4dr3query As String = "update Bldg4d3 Set
ReceiverName=@receivername where queueno=@queueno AND
Bldg=@bldg AND Dock=@dock"

        Dim bldg4dr4query As String = "update Bldg4d4 Set
ReceiverName=@receivername where queueno=@queueno AND
Bldg=@bldg AND Dock=@dock"

        Dim bldg3dr1query As String = "update Bldg3 Set
ReceiverName=@receivername where queueno=@queueno AND
Bldg=@bldg AND Dock=@dock"

        Dim bldg6dr1query As String = "update Bldg6 Set
ReceiverName=@receivername where queueno=@queueno AND
Bldg=@bldg AND Dock=@dock"

        Dim updatequerystat As String = String.Empty
        updatequerystat &= "update Queue_tbl set
NotifDockStatus=@notifstat where Queueno=@queue"

        Dim selectqueuecontact As String = "select contact_no from
truck_details where Queueno=@queueno"

        Dim updatereceivetime As String = "update truck_details Set
startreceive=getdate() where Queueno=@queueno"

        Dim selectquery As String = "select drno as 'DR No.'
,actualnoofboxes as 'No. of Boxes',weight as 'Weight per Box',
actualtotalunits as 'No. of Units' from drtbl where
queueno=@queueno"

        Using conn As New SqlConnection(connectionString)

            Dim queueno As String

            Dim bldg As String

            Dim Dock As String

            Try

                conn.Open()

                cmd = New SqlCommand(query, conn)

                cmd.ExecuteNonQuery()

                Dim dr As SqlClient.SqlDataReader

dr = cmd.ExecuteReader(CommandBehavior.CloseConnection)

                While dr.Read

                    queueno = dr("Queueno").ToString

                    bldg = dr("Bldg").ToString

                    Dock = dr("Dock").ToString
        Label15.Text = queueno

        Label10.Text = bldg

        Label11.Text = Dock

        If bldg.Equals("4") And Dock.Equals("1") Then

cmd1 = New SqlCommand(bldg4dr1query, conn)

cmd1.Parameters.AddWithValue("@receivername",
receiverpublic.ReceiverName)

cmd1.Parameters.AddWithValue("@queueno", queueno)

cmd1.Parameters.AddWithValue("@bldg", "4")

cmd1.Parameters.AddWithValue("@dock", "1")

cmd1.ExecuteNonQuery()

Button2.Enabled = False

Button3.Enabled = True

Button1.Enabled = True

ElseIf bldg.Equals("4") And Dock.Equals("2") Then

cmd1 = New SqlCommand(bldg4dr2query, conn)

cmd1.Parameters.AddWithValue("@receivername",
receiverpublic.ReceiverName)

cmd1.Parameters.AddWithValue("@queueno", queueno)

cmd1.Parameters.AddWithValue("@bldg", "4")

cmd1.Parameters.AddWithValue("@dock", "2")

cmd1.ExecuteNonQuery()
```

```
        Button2.Enabled = False

        Button3.Enabled = True

        Button1.Enabled = True

    ElseIf bldg.Equals("4") And Dock.Equals("3") Then

        cmd1 = New SqlCommand(bldg4dr3query, conn)

  cmd1.Parameters.AddWithValue("@receivername",
receiverpublic.ReceiverName)

  cmd1.Parameters.AddWithValue("@queueno", queueno)

        cmd1.Parameters.AddWithValue("@bldg", "4")

        cmd1.Parameters.AddWithValue("@dock", "3")

        cmd1.ExecuteNonQuery()

        Button2.Enabled = False

        Button3.Enabled = True

        Button1.Enabled = True

    ElseIf bldg.Equals("4") And Dock.Equals("4") Then

        cmd1 = New SqlCommand(bldg4dr4query, conn)

      cmd1.Parameters.AddWithValue("@receivername",
receiverpublic.ReceiverName)

cmd1.Parameters.AddWithValue("@queueno", queueno)

        cmd1.Parameters.AddWithValue("@bldg", "4")

        cmd1.Parameters.AddWithValue("@dock", "4")

        cmd1.ExecuteNonQuery()
```

```
           Button2.Enabled = False

           Button3.Enabled = True

           Button1.Enabled = True

         ElseIf bldg.Equals("3") And Dock.Equals("1") Then

           cmd1 = New SqlCommand(bldg3dr1query, conn)

      cmd1.Parameters.AddWithValue("@receivername",
      receiverpublic.ReceiverName)

      cmd1.Parameters.AddWithValue("@queueno", queueno)

           cmd1.Parameters.AddWithValue("@bldg", "3")

           cmd1.Parameters.AddWithValue("@dock", "1")

           cmd1.ExecuteNonQuery()

           Button2.Enabled = False

           Button3.Enabled = True

           Button1.Enabled = True

         Else

           cmd1 = New SqlCommand(bldg6dr1query, conn)

 cmd1.Parameters.AddWithValue("@receivername",
 receiverpublic.ReceiverName)

cmd1.Parameters.AddWithValue("@queueno", queueno)

cmd1.Parameters.AddWithValue("@bldg", "6")

cmd1.Parameters.AddWithValue("@dock", "1")

cmd1.ExecuteNonQuery()
```

```
            Button2.Enabled = False

            Button3.Enabled = True

            Button1.Enabled = True

          End If

cmd2 = New SqlCommand(selectqueuecontact, conn)

cmd2.Parameters.AddWithValue("@queueno", queueno)

cmd2.ExecuteNonQuery()

Dim dr1 As SqlClient.SqlDataReader

dr1 = cmd2.ExecuteReader(CommandBehavior.CloseConnection)

          While dr1.Read

            Dim contactnumber As String =
dr1("contact_no").ToString

            Dim smsmessage As String = ("Your Queue No. is: "
+ queueno + " Please Proceed to Building: " + bldg + " Dock: " +
Dock + " with your Receiver: " + receiverpublic.ReceiverName)

            Try

              Dim res As String = itexmo(contactnumber,
smsmessage, My.Settings.smsAPI, "(z}v7vymks")

              If res = "0" Then

MsgBox("Queue Message Has been Sent to the Driver!")

cmd3 = New SqlCommand(updatequerystat, conn)

cmd3.Parameters.AddWithValue("@notifstat", "Message Sent!")

cmd3.Parameters.AddWithValue("@queue", queueno)
```

```
                Try
                  cmd3.ExecuteNonQuery()
                Catch ex As Exception
                  MsgBox("Update Error")
                End Try
              Else
MsgBox("Error on Sending Message, Please Contact
Administrator")
cmd3 = New SqlCommand(updatequerystat, conn)
cmd3.Parameters.AddWithValue("@notifstat", "Message NOT
Sent!")
cmd3.Parameters.AddWithValue("@queue", queueno)
                Try
                  cmd3.ExecuteNonQuery()
                Catch ex As Exception
                  MsgBox("Update Error")
                End Try
              End If
            Catch ex As Exception
MsgBox("Error on Sending Message, Please Contact
Administrator")
cmd3 = New SqlCommand(updatequerystat, conn)
```

```
cmd3.Parameters.AddWithValue("@notifstat", "Message NOT Sent!")
cmd3.Parameters.AddWithValue("@queue", queueno)
              Try
                cmd3.ExecuteNonQuery()
              Catch ex1 As Exception
                MsgBox("Update Error")
              End Try
            End Try
          End While
cmd4 = New SqlCommand(updatereceivetime, conn)
cmd4.Parameters.AddWithValue("@queueno", queueno)
cmd4.ExecuteNonQuery()
          Try
cmd5 = New SqlCommand(selectquery, conn)
cmd5.Parameters.AddWithValue("@queueno", queueno)
            Dim da As New SqlDataAdapter(cmd5)
            Dim dt As New DataTable()
            da.Fill(dt)
            dbs4.DataSource = dt
            DataGridView1.DataSource = dbs4
            DataGridView1.ReadOnly = True
```

```
                    Catch ex As Exception
MessageBox.Show(ex.Message.ToString(), "Error Message")
                    End Try
                End While
                conn.Close()
            Catch ex As Exception
MessageBox.Show(ex.Message.ToString(), "Error Message")
            End Try
        End Using
    End Sub
    Private Sub Receiver_Load(sender As Object, e As EventArgs)
Handles MyBase.Load
        DataGridView1.AllowUserToAddRows = False
        DataGridView1.RowHeadersVisible = False
        DataGridView1.GridColor = Color.Red
        NoShowlist.RowHeadersVisible = False
        NoShowlist.GridColor = Color.Red
        Me.FormBorderStyle = FormBorderStyle.None
        Dim myname As New DBconnection
        Name = myname.ReceiverName
        Label21.Text = Name
        Button3.Enabled = False
        Button1.Enabled = False

        LoadTable()

        Dim con As New SqlClient.SqlConnection

Dim connectionString As String = "Server='" & My.Settings.mServer
& "';

     Database='" & My.Settings.mDB & "';

     User Id='" & My.Settings.mUserDB & "';

     Password='" & My.Settings.mPassDB & "';
MultipleActiveResultSets=True"

Dim selectexist As String = "select a.* from Bldg3 a where
ReceiverName=@receivername union select b.* from Bldg6 b
where ReceiverName=@receivername union select c.* from
bldg4d1 c where ReceiverName=@receivername union select d.*
from bldg4d2 d where ReceiverName=@receivername union select
e.* from Bldg4d3 e where ReceiverName=@receivername union
select f.* from Bldg4d4 f where ReceiverName=@receivername"

Dim selectquery As String = "select drno as 'DR No.'
,actualnoofboxes as 'No. of Boxes',weight as 'Weight per Box',
actualtotalunits as 'No. of Units' from drtbl where
queueno=@queueno"

        con.ConnectionString = connectionString

        Using conn As New SqlConnection(connectionString)

            Dim queueno As String

            Dim bldg As String

            Dim Dock As String

            Try
            conn.Open()

            cmd = New SqlCommand(selectexist, conn)

            cmd.Parameters.AddWithValue("@receivername",
receiverpublic.ReceiverName)

            cmd.ExecuteNonQuery()

            Dim dr As SqlClient.SqlDataReader

            dr =
cmd.ExecuteReader(CommandBehavior.CloseConnection)

            While dr.Read

              queueno = dr("Queueno").ToString

              bldg = dr("Bldg").ToString

              Dock = dr("Dock").ToString

              Label15.Text = queueno

              Label10.Text = bldg

              Label11.Text = Dock

              Button2.Enabled = False

              Button3.Enabled = True

              Button1.Enabled = True

              Try

cmd5 = New SqlCommand(selectquery, conn)

cmd5.Parameters.AddWithValue("@queueno", queueno)

Dim da As New SqlDataAdapter(cmd5)
```

```
                        Dim dt As New DataTable()

                        da.Fill(dt)

                        dbs4.DataSource = dt

                        DataGridView1.DataSource = dbs4

                        DataGridView1.ReadOnly = True

                    Catch ex As Exception

MessageBox.Show(ex.Message.ToString(), "Error Message")

                    End Try

                End While

                conn.Close()

            Catch ex As Exception

MessageBox.Show(ex.Message.ToString(), "Error Message")

            End Try

        End Using

    End Sub

    Private Sub Button1_Click(sender As Object, e As EventArgs)
Handles Button1.Click

If TextBox1.Text = "" Or Label7.Text = "" Or Label13.Text = "" Then

            errorlbl.Text = "Please complete all fields"

            errorlbl.ForeColor = Color.Red

        Else

            Button3.Enabled = False
```

```
Button2.Enabled = True

Dim con As New SqlClient.SqlConnection

Dim connectionString As String = "Server='" & My.Settings.mServer
& "';

    Database='" & My.Settings.mDB & "';

    User Id='" & My.Settings.mUserDB & "';

    Password='" & My.Settings.mPassDB & "';
MultipleActiveResultSets=True"

Dim statquery As String = "update truck_details Set
docktime=@docktime, startunload=@startunload,
finishedunload=@finishedunload, finishedreceive=getdate() where
queueno=@queueno"

Dim selectnumber As String = "select totalnoofboxes, totaltotalunits
from truck_details where queueno=@queueno"

Dim insertquery As String = "insert into Actualunloaddetails
(Queueno,totalnoofboxes,totaltotalunits,totalnoofpallet,actualtotalwe
ight,differenceinboxes,differenceinunits)
values(@queueno,@totalnoofboxes,@totaltotalunits,@totalnoofpall
et,@actualtotalweight,@differenceinboxes,@differenceinunits)"

Dim insertbldg4dr1query As String = "insert into DoneQueuetbl
(queueno,ReceiverName,Bldg,Dock)
values(@queueno,@receivername,@bldg,@dock)"

Dim insertbldg4dr2query As String = "insert into DoneQueuetbl
(queueno,ReceiverName,Bldg,Dock)
values(@queueno,@receivername,@bldg,@dock)"

Dim insertbldg4dr3query As String = "insert into DoneQueuetbl
(queueno,ReceiverName,Bldg,Dock)
values(@queueno,@receivername,@bldg,@dock)"
```

```
Dim insertbldg4dr4query As String = "insert into DoneQueuetbl
(queueno,ReceiverName,Bldg,Dock)
values(@queueno,@receivername,@bldg,@dock)"

Dim insertbldg3dr1query As String = "insert into DoneQueuetbl
(queueno,ReceiverName,Bldg,Dock)
values(@queueno,@receivername,@bldg,@dock)"

Dim insertbldg6dr1query As String = "insert into DoneQueuetbl
(queueno,ReceiverName,Bldg,Dock)
values(@queueno,@receivername,@bldg,@dock)"

Dim deletebldg4dr1query As String = "delete from bldg4d1 where
Queueno=@queueno AND Bldg=@bldg AND Dock=@dock"

Dim deletebldg4dr2query As String = "delete from bldg4d2 where
Queueno=@queueno AND Bldg=@bldg AND Dock=@dock"

Dim deletebldg4dr3query As String = "delete from Bldg4d3 where
Queueno=@queueno AND Bldg=@bldg AND Dock=@dock"

Dim deletebldg4dr4query As String = "delete from Bldg4d4 where
Queueno=@queueno AND Bldg=@bldg AND Dock=@dock"

Dim deletebldg3dr1query As String = "delete from Bldg3 where
Queueno=@queueno AND Bldg=@bldg AND Dock=@dock"

Dim deletebldg6dr1query As String = "delete from Bldg6 where
Queueno=@queueno AND Bldg=@bldg AND Dock=@dock"

Dim selecttransno As String = "select max(transno) as queuemax
from DoneQueuetbl where Queueno =@queueno"

Dim insertqueuetbl As String = "INSERT INTO DoneQueueInfotbl (
Arrival_date, Arrival_time, Status, NotifStatus,
NotifDockStatus,NotifPendingStatus,VerifiedBy,EncodedBy,EndEnc
oded,StartVerify,EndVerify,Pending,transno )SELECT
Queue_tbl.Arrival_date, Queue_tbl.Arrival_time, Queue_tbl.Status,
Queue_tbl.NotifStatus, Queue_tbl.NotifDockStatus,
Queue_tbl.NotifPendingStatus,
Queue_tbl.VerifiedBy,Queue_tbl.EncodedBy,Queue_tbl.EndEncode
d,Queue_tbl.StartVerify,Queue_tbl.EndVerify,Queue_tbl.Pending,
@maxqueue FROM Queue_tbl where queueno = @queueno"

Dim deletequeuetbl As String = "delete from Queue_tbl where
Queueno=@queueno"

Dim inserttruckdetailstbl As String = "INSERT INTO
Donetruckdetails (vendorname, plate_no, truck_type, truck_load,
no_of_helper,driver_name,contact_no,category,countofdr,totalnoofb
oxes,totaltotalunits,docktime,startunload,finishedunload,difference,s
tartreceive,finishedreceive,advancedel,shortqty,overqty,reenterpo,u
nknown,transno )SELECT  truck_details.vendorname,
truck_details.plate_no, truck_details.truck_type,
truck_details.truck_load, truck_details.no_of_helper,
truck_details.driver_name, truck_details.contact_no,
truck_details.category, truck_details.countofdr,
truck_details.totalnoofboxes, truck_details.totaltotalunits,
truck_details.docktime, truck_details.startunload,
truck_details.finishedunload,@difference, truck_details.startreceive,
truck_details.finishedreceive, truck_details.advancedel,
truck_details.shortqty, truck_details.overqty,
truck_details.reenterpo, truck_details.unknown, @maxqueue
FROM truck_details where Queueno = @queueno"

Dim deletetruckdetailstbl As String = "delete from truck_details
where Queueno=@queueno"

Dim insertactualunloaddetailstbl As String = "INSERT INTO
DoneActualunloaddetails ( actualtotalnoofboxes, actualtotalunits,
actualtotalnoofpallet,actualtotalweight, transno,
differenceinboxes,differenceinunits)SELECT
Actualunloaddetails.totalnoofboxes,
Actualunloaddetails.totaltotalunits,
Actualunloaddetails.totalnoofpallet,Actualunloaddetails.actualtotalw
```

```
eight, @maxqueue, Actualunloaddetails.differenceinboxes,
Actualunloaddetails.differenceinunits FROM Actualunloaddetails
where queueno = @queueno"

Dim deleteactualunloaddetailstbl As String = "delete from
Actualunloaddetails where Queueno=@queueno"

Dim insertdrtbl As String = "INSERT INTO Donedrtbl ( drno,
noofboxes, noofunits,weight,actualnoofboxes,actualtotalunits,target,
transno)SELECT  drtbl.drno, drtbl.noofboxes, drtbl.totalunits,
drtbl.weight,
drtbl.actualnoofboxes,drtbl.actualtotalunits,drtbl.target,@maxqueue
FROM drtbl where queueno = @queueno"

Dim deletedrtbl As String = "delete from drtbl where
Queueno=@queueno"

Dim Encode As String = "INSERT INTO
Encode(EncodeDiff,TargetEncode,GuardGrade,Excess,Within,trans
no) SELECT DATEDIFF(mi, dq.Arrival_time,dq.EndENcoded)
,countofdr*2,

Case When
DATEDIFF(mi,dq.Arrival_time,dq.EndENcoded)<=dqi.countofdr*2
Then 'Passed'

Else 'Failed' End as GuardGrade,

Case When DATEDIFF(mi,
dq.Arrival_time,dq.EndENcoded)<=dqi.countofdr*2 Then '0'

Else DATEDIFF(mi, dq.Arrival_time,dq.EndENcoded)-
dqi.countofdr*2 End,

Case When DATEDIFF(mi,
dq.Arrival_time,dq.EndENcoded)<=dqi.countofdr*2 Then
DATEDIFF(mi, dq.Arrival_time,dq.EndENcoded)

Else dqi.countofdr*2 End,@maxqueue
```

```
from  DoneQueueInfotbl dq inner join Donetruckdetails dqi on
dq.transno=dqi.transno inner join DoneQueuetbl dqii on
dq.transno=dqii.transno where dqii.transno = @maxqueue"

Dim selectverifier As String = "Select VerifiedBy from
DoneQueueInfotbl where transno= @maxqueue"

Dim GV As String = "SELECT TOP 1 DATEDIFF ( mi , StartVerify ,
EndVerify )+1 as Target  From

(select Top 2 * from DoneQueueInfotbl where VerifiedBy =
@verifier ORDER BY
try_convert(int,Right(convert(varchar(10),transno),2)) DESC) x

ORDER BY try_convert(int,Right(convert(varchar(10),transno),2))"

Dim GVTime As String = "INSERT INTO
GV(GVDiff,GVTarget,GVGrade,GVExcess,GVWithin,transno)
SELECT DATEDIFF(mi, dq.EndENcoded,dq.StartVerify),@target,

Case When
DATEDIFF(mi,dq.EndENcoded,dq.StartVerify)<=@target Then
'Passed'

Else 'Failed' End,

Case When DATEDIFF(mi,
dq.EndENcoded,dq.StartVerify)<=@target Then '0'

Else DATEDIFF(mi, dq.EndENcoded,dq.StartVerify)-@target End,

Case When DATEDIFF(mi,
dq.EndENcoded,dq.StartVerify)<=@target Then DATEDIFF(mi,
dq.EndENcoded,dq.StartVerify)

Else @target End, @maxqueue from  DoneQueueInfotbl dq inner
join Donetruckdetails dqi on dq.transno=dqi.transno inner join
DoneQueuetbl dqii on dq.transno=dqii.transno where dqii.transno =
@maxqueue"
```

```
Dim VerifyTime As String = "INSERT INTO
Verify(Vdiff,VTarget,VGrade,VExcess,VWithin,transno) SELECT
DATEDIFF(mi, dq.StartVerify,dq.EndVerify) ,dqi.countofdr*4,

Case When
DATEDIFF(mi,dq.StartVerify,dq.EndVerify)<=dqi.countofdr*4 Then
'Passed'

Else 'Failed' End as GuardGrade,

Case When DATEDIFF(mi,
dq.StartVerify,dq.EndVerify)<=dqi.countofdr*4 Then '0'

Else DATEDIFF(mi, dq.StartVerify,dq.EndVerify)-dqi.countofdr*4
End,

Case When DATEDIFF(mi,
dq.StartVerify,dq.EndVerify)<=dqi.countofdr*4 Then DATEDIFF(mi,
dq.StartVerify,dq.EndVerify)

Else dqi.countofdr*4 End,@maxqueue

from  DoneQueueInfotbl dq inner join Donetruckdetails dqi on
dq.transno=dqi.transno inner join DoneQueuetbl dqii on
dq.transno=dqii.transno where dqii.transno = @maxqueue"

Dim SelectCategory As String = "Select category from
Donetruckdetails where transno= @maxqueue"

Dim VRTarget As String = "SELECT TOP 1 DATEDIFF ( mi ,
startreceive , finishedreceive )+1 As Target From

(select Top 2 * from Donetruckdetails where category = @category
ORDER BY try_convert(int,Right(convert(varchar(10),transno),2))
DESC) x

ORDER BY try_convert(int,Right(convert(varchar(10),transno),2))"
```

```
Dim VRTime As String = "INSERT INTO
VR(VRDiff,VRTarget,VRGrade,VRExcess,VRWithin,transno)
SELECT DATEDIFF(mi, CAST(dq.Arrival_date AS
DATETIME)+CAST(dq.EndVerify AS DATETIME),dqi.startreceive)
,@target,

Case When DATEDIFF(mi,CAST(dq.Arrival_date AS
DATETIME)+CAST(dq.EndVerify AS
DATETIME),dqi.startreceive)<=@target Then 'Passed'

Else 'Failed' End as GuardGrade,

Case When DATEDIFF(mi, CAST(dq.Arrival_date AS
DATETIME)+CAST(dq.EndVerify AS
DATETIME),dqi.startreceive)<=@target Then '0'

Else DATEDIFF(mi, CAST(dq.Arrival_date AS
DATETIME)+CAST(dq.EndVerify AS DATETIME),dq.EndVerify)-
@target End,

Case When DATEDIFF(mi, CAST(dq.Arrival_date AS
DATETIME)+CAST(dq.EndVerify AS
DATETIME),dqi.startreceive)<=@target Then DATEDIFF(mi,
CAST(dq.Arrival_date AS DATETIME)+CAST(dq.EndVerify AS
DATETIME),dqi.startreceive)

Else @target End,@maxqueue

from  DoneQueueInfotbl dq inner join Donetruckdetails dqi on
dq.transno=dqi.transno inner join DoneQueuetbl dqii on
dq.transno=dqii.transno where dqii.transno = @maxqueue"

Dim RDTime As String = "INSERT INTO
RD(RDDiff,RDTarget,RDGrade,RDExcess,RDWithin,transno)
SELECT DATEDIFF(mi,startreceive,dqi.docktime) ,

CASE WHEN dqi.truck_type = '4W'  THEN '5'

WHEN dqi.truck_type = '6W' THEN '5'
WHEN dqi.truck_type = '10W' THEN '10'
WHEN dqi.truck_type = '40FTR' THEN '15'
WHEN dqi.truck_type = 'WINGVAN' THEN '10'
WHEN dqi.truck_type = 'L300' THEN '5'
WHEN dqi.truck_type = '20FTR' THEN '15'
WHEN dqi.truck_type = 'APMI' THEN '10'
WHEN dqi.truck_type = '6WFWD' THEN '10'
WHEN dqi.truck_type = 'ON FOOT' THEN '5'
END,
Case When DATEDIFF(mi,startreceive,dqi.docktime)<=CASE
WHEN dqi.truck_type = '4W'  THEN '5'
WHEN dqi.truck_type = '6W' THEN '5'
WHEN dqi.truck_type = '10W' THEN '10'
WHEN dqi.truck_type = '40FTR' THEN '15'
WHEN dqi.truck_type = 'WINGVAN' THEN '10'
WHEN dqi.truck_type = 'L300' THEN '5'
WHEN dqi.truck_type = '20FTR' THEN '15'
WHEN dqi.truck_type = 'APMI' THEN '10'
WHEN dqi.truck_type = '6WFWD' THEN '10'
WHEN dqi.truck_type = 'ON FOOT' THEN '5'
END Then 'Passed'
Else 'Failed' End,
Case When DATEDIFF(mi,startreceive,dqi.docktime)<=CASE WHEN dqi.truck_type = '4W'  THEN '5'

WHEN dqi.truck_type = '6W' THEN '5'

WHEN dqi.truck_type = '10W' THEN '10'

WHEN dqi.truck_type = '40FTR' THEN '15'

WHEN dqi.truck_type = 'WINGVAN' THEN '10'

WHEN dqi.truck_type = 'L300' THEN '5'

WHEN dqi.truck_type = '20FTR' THEN '15'

WHEN dqi.truck_type = 'APMI' THEN '10'

WHEN dqi.truck_type = '6WFWD' THEN '10'

WHEN dqi.truck_type = 'ON FOOT' THEN '5'

END Then '0'

Else DATEDIFF(mi,startreceive,dqi.docktime)-CASE WHEN dqi.truck_type = '4W'  THEN '5'

WHEN dqi.truck_type = '6W' THEN '5'

WHEN dqi.truck_type = '10W' THEN '10'

WHEN dqi.truck_type = '40FTR' THEN '15'

WHEN dqi.truck_type = 'WINGVAN' THEN '10'

WHEN dqi.truck_type = 'L300' THEN '5'

WHEN dqi.truck_type = '20FTR' THEN '15'

WHEN dqi.truck_type = 'APMI' THEN '10'

WHEN dqi.truck_type = '6WFWD' THEN '10'
WHEN dqi.truck_type = 'ON FOOT' THEN '5'

END End,

Case When DATEDIFF(mi,startreceive,dqi.docktime)<=CASE
WHEN dqi.truck_type = '4W'  THEN '5'

WHEN dqi.truck_type = '6W' THEN '5'

WHEN dqi.truck_type = '10W' THEN '10'

WHEN dqi.truck_type = '40FTR' THEN '15'

WHEN dqi.truck_type = 'WINGVAN' THEN '10'

WHEN dqi.truck_type = 'L300' THEN '5'

WHEN dqi.truck_type = '20FTR' THEN '15'

WHEN dqi.truck_type = 'APMI' THEN '10'

WHEN dqi.truck_type = '6WFWD' THEN '10'

WHEN dqi.truck_type = 'ON FOOT' THEN '5'

END Then DATEDIFF(mi,startreceive,dqi.docktime)

Else CASE WHEN dqi.truck_type = '4W'  THEN '5'

WHEN dqi.truck_type = '6W' THEN '5'

WHEN dqi.truck_type = '10W' THEN '10'

WHEN dqi.truck_type = '40FTR' THEN '15'

WHEN dqi.truck_type = 'WINGVAN' THEN '10'

WHEN dqi.truck_type = 'L300' THEN '5'

WHEN dqi.truck_type = '20FTR' THEN '15'

WHEN dqi.truck_type = 'APMI' THEN '10'
WHEN dqi.truck_type = '6WFWD' THEN '10'

WHEN dqi.truck_type = 'ON FOOT' THEN '5'

END End,@maxqueue

from  DoneQueueInfotbl dq inner join Donetruckdetails dqi on
dq.transno=dqi.transno inner join DoneQueuetbl dqii on
dq.transno=dqii.transno where dqii.transno = @maxqueue"

Dim DU As String = "INSERT INTO
DU(DUDiff,DUTarget,DUGrade,DUExcess,DUWithin,transno)
SELECT DATEDIFF(mi, dqi.docktime,dqi.startunload) , 2,

Case When DATEDIFF(mi,dqi.docktime,dqi.startunload)<=2 Then
'Passed'

Else 'Failed' End,

Case When DATEDIFF(mi, dqi.docktime,dqi.startunload)<=2 Then
'0'

Else DATEDIFF(mi, dqi.docktime,dqi.startunload)-2 End,

Case When DATEDIFF(mi, dqi.docktime,dqi.startunload)<=2 Then
DATEDIFF(mi, dqi.docktime,dqi.startunload)

Else 2 End,@maxqueue

from  DoneQueueInfotbl dq inner join Donetruckdetails dqi on
dq.transno=dqi.transno inner join DoneQueuetbl dqii on
dq.transno=dqii.transno where dqii.transno = @maxqueue"

Dim Unload As String = "INSERT INTO
Unload(UnloadDiff,UnloadTarget,UnloadGrade,UnloadExcess,Unlo
adWithin,transno) SELECT DATEDIFF(mi,
dqi.startunload,dqi.finishedunload),SUM(dqiii.target),
Case When DATEDIFF(mi,
dqi.startunload,dqi.finishedunload)<=SUM(dqiii.target) Then
'Passed'

Else 'Failed' End,

Case When DATEDIFF(mi,
dqi.startunload,dqi.finishedunload)<=SUM(dqiii.target) Then '0'

Else DATEDIFF(mi, dqi.startunload,dqi.finishedunload)-
SUM(dqiii.target) End,

Case When DATEDIFF(mi,
dqi.startunload,dqi.finishedunload)<=SUM(dqiii.target) Then
DATEDIFF(mi, dqi.startunload,dqi.finishedunload)

Else SUM(dqiii.target) End,@maxqueue from  DoneQueueInfotbl
dq inner join Donetruckdetails dqi on dq.transno=dqi.transno inner
join DoneQueuetbl dqii on dq.transno=dqii.transno inner join
Donedrtbl dqiii on dqiii.transno=dqi.transno where dqii.transno =
@maxqueue group by dqi.transno, startunload,finishedunload,
EndEncoded, StartVerify"

Dim UR As String = "INSERT INTO
UR(URDiff,URTarget,URGrade,URExcess,URWithin,transno)
SELECT DATEDIFF(mi, dqi.finishedunload,dqi.finishedreceive) ,5,

Case When DATEDIFF(mi,
dqi.finishedunload,dqi.finishedreceive)<=5 Then 'Passed'

Else 'Failed' End as GuardGrade,

Case When DATEDIFF(mi,
dqi.finishedunload,dqi.finishedreceive)<=5 Then '0'

Else DATEDIFF(mi, dqi.finishedunload,dqi.finishedreceive)-5 End,
Case When DATEDIFF(mi,
dqi.finishedunload,dqi.finishedreceive)<=5 Then DATEDIFF(mi,
dqi.finishedunload,dqi.finishedreceive)

Else 5 End,@maxqueue

from  DoneQueueInfotbl dq inner join Donetruckdetails dqi on
dq.transno=dqi.transno inner join DoneQueuetbl dqii on
dq.transno=dqii.transno where dqii.transno = @maxqueue"

          con.ConnectionString = connectionString

          Using conn As New SqlConnection(connectionString)

                 Dim bldg As String = Label10.Text

                 Dim Dock As String = Label11.Text

                 Dim queueno As String = Label15.Text

                 Dim queuemax As String

                 Dim differenceinboxes As Integer

                 Dim differenceinunits As Integer

                 Dim boxes As String

                 Dim units As String

                 Dim Actualboxes As Integer =
Convert.ToInt32(Label7.Text)

                 Dim Actualunits As Integer =
Convert.ToInt32(Label13.Text)

                 Try

                     conn.Open()

                     cmd7 = New SqlCommand(selectnumber, conn)
```

```
cmd7.Parameters.AddWithValue("@queueno", queueno)

cmd7.ExecuteNonQuery()

Dim dr1 As SqlClient.SqlDataReader

dr1 = cmd7.ExecuteReader(CommandBehavior.CloseConnection)

            While dr1.Read

              boxes = dr1("totalnoofboxes").ToString

              units = dr1("totaltotalunits").ToString

              differenceinboxes = Actualboxes - CInt(boxes)

              differenceinunits = Actualunits - CInt(units)

            End While

            conn.Close()

          Catch ex As Exception

MessageBox.Show(ex.Message.ToString(), "Error Message")

            conn.Close()

          End Try

          Try

            conn.Open()

cmd = New SqlCommand(statquery, conn)

cmd.Parameters.AddWithValue("@queueno", Label15.Text)

cmd.Parameters.AddWithValue("@docktime",
DateTimePicker2.Value)
```

```
cmd.Parameters.AddWithValue("@startunload",
DateTimePicker1.Value)

cmd.Parameters.AddWithValue("@finishedunload",
DateTimePicker5.Value)

cmd.ExecuteNonQuery()

            conn.Close()

          Catch ex As Exception

MessageBox.Show(ex.Message.ToString(), "Error Message")

          End Try

Try

conn.Open()

cmd1 = New SqlCommand(insertquery, conn)

cmd1.Parameters.AddWithValue("@queueno", Label15.Text)

cmd1.Parameters.AddWithValue("@totalnoofboxes", Label7.Text)

cmd1.Parameters.AddWithValue("@totaltotalunits", Label13.Text)

cmd1.Parameters.AddWithValue("@totalnoofpallet",
TextBox1.Text)

cmd1.Parameters.AddWithValue("@differenceinboxes",
differenceinboxes)

cmd1.Parameters.AddWithValue("@differenceinunits",
differenceinunits)

cmd1.Parameters.AddWithValue("@actualtotalweight",
Label14.Text)
```

```
          cmd1.ExecuteNonQuery()

          conn.Close()

        Catch ex As Exception

MessageBox.Show(ex.Message.ToString(), "Error Message")

        End Try

        Try

          conn.Open()

If bldg.Equals("4") And Dock.Equals("1") Then

cmd2 = New SqlCommand(insertbldg4dr1query, conn)

cmd2.Parameters.AddWithValue("@receivername", receiverpublic.ReceiverName)

cmd2.Parameters.AddWithValue("@queueno", queueno)

cmd2.Parameters.AddWithValue("@bldg", "4")

cmd2.Parameters.AddWithValue("@dock", "1")

cmd2.ExecuteNonQuery()

            Try

cmd3 = New SqlCommand(deletebldg4dr1query, conn)

cmd3.Parameters.AddWithValue("@receivername", receiverpublic.ReceiverName)

cmd3.Parameters.AddWithValue("@queueno", queueno)

cmd3.Parameters.AddWithValue("@bldg", "4")

cmd3.Parameters.AddWithValue("@dock", "1")
```

```
cmd3.ExecuteNonQuery()

              conn.Close()

            Catch ex As Exception

MessageBox.Show(ex.Message.ToString(), "Error Message")

              conn.Close()

            End Try

ElseIf bldg.Equals("4") And Dock.Equals("2") Then

cmd2 = New SqlCommand(insertbldg4dr2query, conn)

cmd2.Parameters.AddWithValue("@receivername",
receiverpublic.ReceiverName)

cmd2.Parameters.AddWithValue("@queueno", queueno)

cmd2.Parameters.AddWithValue("@bldg", "4")

cmd2.Parameters.AddWithValue("@dock", "2")

cmd2.ExecuteNonQuery()

            Try

cmd3 = New SqlCommand(deletebldg4dr2query, conn)

cmd3.Parameters.AddWithValue("@receivername",
receiverpublic.ReceiverName)

cmd3.Parameters.AddWithValue("@queueno", queueno)

cmd3.Parameters.AddWithValue("@bldg", "4")

cmd3.Parameters.AddWithValue("@dock", "2")

cmd3.ExecuteNonQuery()
```

```
         conn.Close()

        Catch ex As Exception

MessageBox.Show(ex.Message.ToString(), "Error Message")

         conn.Close()

        End Try


ElseIf bldg.Equals("4") And Dock.Equals("3") Then

cmd2 = New SqlCommand(insertbldg4dr3query, conn)

cmd2.Parameters.AddWithValue("@receivername", receiverpublic.ReceiverName)

cmd2.Parameters.AddWithValue("@queueno", queueno)

cmd2.Parameters.AddWithValue("@bldg", "4")

cmd2.Parameters.AddWithValue("@dock", "3")

cmd2.ExecuteNonQuery()

        Try

cmd3 = New SqlCommand(deletebldg4dr3query, conn)

cmd3.Parameters.AddWithValue("@receivername", receiverpublic.ReceiverName)

cmd3.Parameters.AddWithValue("@queueno", queueno)

cmd3.Parameters.AddWithValue("@bldg", "4")

cmd3.Parameters.AddWithValue("@dock", "3")

cmd3.ExecuteNonQuery()
```

```
                    conn.Close()

                 Catch ex As Exception

MessageBox.Show(ex.Message.ToString(), "Error Message")

                    conn.Close()

                 End Try

               ElseIf bldg.Equals("4") And Dock.Equals("4") Then

cmd2 = New SqlCommand(insertbldg4dr4query, conn)

cmd2.Parameters.AddWithValue("@receivername",
receiverpublic.ReceiverName)

cmd2.Parameters.AddWithValue("@queueno", queueno)

cmd2.Parameters.AddWithValue("@bldg", "4")

cmd2.Parameters.AddWithValue("@dock", "4")

cmd2.ExecuteNonQuery()

                 Try

cmd3 = New SqlCommand(deletebldg4dr4query, conn)

cmd3.Parameters.AddWithValue("@receivername",
receiverpublic.ReceiverName)

cmd3.Parameters.AddWithValue("@queueno", queueno)

cmd3.Parameters.AddWithValue("@bldg", "4")

cmd3.Parameters.AddWithValue("@dock", "4")

cmd3.ExecuteNonQuery()

                    conn.Close()
               Catch ex As Exception
MessageBox.Show(ex.Message.ToString(), "Error Message")
                 conn.Close()
               End Try

ElseIf bldg.Equals("3") And Dock.Equals("1") Then
cmd2 = New SqlCommand(insertbldg3dr1query, conn)
cmd2.Parameters.AddWithValue("@receivername",
receiverpublic.ReceiverName)
cmd2.Parameters.AddWithValue("@queueno", queueno)
cmd2.Parameters.AddWithValue("@bldg", "3")
cmd2.Parameters.AddWithValue("@dock", "1")
cmd2.ExecuteNonQuery()
               Try
cmd3 = New SqlCommand(deletebldg3dr1query, conn)
cmd3.Parameters.AddWithValue("@receivername",
receiverpublic.ReceiverName)
cmd3.Parameters.AddWithValue("@queueno", queueno)
cmd3.Parameters.AddWithValue("@bldg", "3")
cmd3.Parameters.AddWithValue("@dock", "1")
cmd3.ExecuteNonQuery()
                 conn.Close()
```

```
        Catch ex As Exception
MessageBox.Show(ex.Message.ToString(), "Error Message")
          conn.Close()
        End Try
      Else
cmd2 = New SqlCommand(insertbldg6dr1query, conn)
cmd2.Parameters.AddWithValue("@receivername",
receiverpublic.ReceiverName)
cmd2.Parameters.AddWithValue("@queueno", queueno)
cmd2.Parameters.AddWithValue("@bldg", "6")
cmd2.Parameters.AddWithValue("@dock", "1")
cmd2.ExecuteNonQuery()
        Try
cmd3 = New SqlCommand(deletebldg6dr1query, conn)
cmd3.Parameters.AddWithValue("@receivername",
receiverpublic.ReceiverName)
cmd3.Parameters.AddWithValue("@queueno", queueno)
cmd3.Parameters.AddWithValue("@bldg", "6")
cmd3.Parameters.AddWithValue("@dock", "1")
cmd3.ExecuteNonQuery()
          conn.Close()
        Catch ex As Exception
```

```
MessageBox.Show(ex.Message.ToString(), "Error Message")
                                conn.Close()
                            End Try
                        End If
                    Catch ex As Exception
                    End Try
                    Try
                        conn.Open()
cmd4 = New SqlCommand(selecttransno, conn)
cmd4.Parameters.AddWithValue("@queueno", queueno)
cmd4.ExecuteNonQuery()
Dim dr As SqlClient.SqlDataReader
dr = cmd4.ExecuteReader(CommandBehavior.CloseConnection)
While dr.Read
queuemax = dr("queuemax").ToString
cmd5 = New SqlCommand(insertqueuetbl, conn)
cmd5.Parameters.AddWithValue("@maxqueue", queuemax)
cmd5.Parameters.AddWithValue("@queueno", queueno)
cmd5.ExecuteNonQuery()
Dim cmd8 As New SqlCommand(inserttruckdetailstbl, conn)
Dim span As TimeSpan = DateTimePicker5.Value - DateTimePicker1.Value

Dim minutes As Decimal = CDec(span.TotalMinutes)

Dim Decimalrounded As Decimal = Decimal.Round(minutes, 2, MidpointRounding.AwayFromZero)

cmd8.Parameters.AddWithValue("@difference", Decimalrounded)

cmd8.Parameters.AddWithValue("@maxqueue", queuemax)

cmd8.Parameters.AddWithValue("@queueno", queueno)

cmd8.ExecuteNonQuery()

Dim cmd10 As New SqlCommand(insertactualunloaddetailstbl, conn)

cmd10.Parameters.AddWithValue("@maxqueue", queuemax)

cmd10.Parameters.AddWithValue("@queueno", queueno)

cmd10.ExecuteNonQuery()

Dim cmd12 As New SqlCommand(insertdrtbl, conn)

cmd12.Parameters.AddWithValue("@maxqueue", queuemax)

cmd12.Parameters.AddWithValue("@queueno", queueno)

cmd12.ExecuteNonQuery()

Dim cmd13 As New SqlCommand(Encode, conn)

cmd13.Parameters.AddWithValue("@maxqueue", queuemax)

cmd13.ExecuteNonQuery()

Dim cmd14 As New SqlCommand(selectverifier, conn)

cmd14.Parameters.AddWithValue("@maxqueue", queuemax)
```

```
cmd14.ExecuteNonQuery()

Dim dr1 As SqlClient.SqlDataReader

dr1 = cmd14.ExecuteReader(CommandBehavior.CloseConnection)

While dr1.Read

Dim cmd15 As New SqlCommand(GV, conn)

cmd15.Parameters.AddWithValue("@verifier",
dr1("VerifiedBy").ToString)

cmd15.ExecuteNonQuery()

Dim dr2 As SqlClient.SqlDataReader

dr2 = cmd15.ExecuteReader(CommandBehavior.CloseConnection)

                While dr2.Read

                    Dim targetGV As String

                    If dr2.IsDBNull(0) Then

                        targetGV = "0"

Dim cmd16 As New SqlCommand(GVTime, conn)


cmd16.Parameters.AddWithValue("@target", targetGV)


cmd16.Parameters.AddWithValue("@maxqueue", queuemax)

cmd16.ExecuteNonQuery()

                    Else

                        targetGV = dr2("Target").ToString

Dim cmd17 As New SqlCommand(GVTime, conn)
```

```
cmd17.Parameters.AddWithValue("@target", targetGV)

cmd17.Parameters.AddWithValue("@maxqueue", queuemax)
cmd17.ExecuteNonQuery()
                  End If
                End While
              End While
Dim cmd18 As New SqlCommand(VerifyTime, conn)
cmd18.Parameters.AddWithValue("@maxqueue", queuemax)
cmd18.ExecuteNonQuery()
Dim cmd19 As New SqlCommand(SelectCategory, conn)
cmd19.Parameters.AddWithValue("@maxqueue", queuemax)
cmd19.ExecuteNonQuery()
Dim dr3 As SqlClient.SqlDataReader
dr3 = cmd19.ExecuteReader(CommandBehavior.CloseConnection)
              While dr3.Read
Dim cmd20 As New SqlCommand(VRTarget, conn)
cmd20.Parameters.AddWithValue("@category",
dr3("category").ToString)
cmd20.ExecuteNonQuery()
Dim dr4 As SqlClient.SqlDataReader
dr4 = cmd20.ExecuteReader(CommandBehavior.CloseConnection)
```

```
                While dr4.Read
                  Dim targetVR As String
                  If dr4.IsDBNull(0) Then
                    targetVR = "0"
Dim cmd21 As New SqlCommand(VRTime, conn)
cmd21.Parameters.AddWithValue("@target", targetVR)
cmd21.Parameters.AddWithValue("@maxqueue", queuemax)
cmd21.ExecuteNonQuery()
                  Else
                    targetVR = dr4("Target").ToString
Dim cmd22 As New SqlCommand(VRTime, conn)
cmd22.Parameters.AddWithValue("@target", targetVR)
cmd22.Parameters.AddWithValue("@maxqueue", queuemax)
cmd22.ExecuteNonQuery()
                  End If
                End While
              End While
Dim cmd23 As New SqlCommand(RDTime, conn)
cmd23.Parameters.AddWithValue("@maxqueue", queuemax)
cmd23.ExecuteNonQuery()
Dim cmd24 As New SqlCommand(DU, conn)
cmd24.Parameters.AddWithValue("@maxqueue", queuemax)
```

```
cmd24.ExecuteNonQuery()
Dim cmd25 As New SqlCommand(Unload, conn)
cmd25.Parameters.AddWithValue("@maxqueue", queuemax)
cmd25.ExecuteNonQuery()
Dim cmd26 As New SqlCommand(UR, conn)
cmd26.Parameters.AddWithValue("@maxqueue", queuemax)
cmd26.ExecuteNonQuery()
          End While
            conn.Close()
          Catch ex As Exception
MessageBox.Show(ex.Message.ToString(), "Error Message")
            conn.Close()
          End Try
          Try
            conn.Open()
cmd6 = New SqlCommand(deletequeuetbl, conn)
cmd6.Parameters.AddWithValue("@queueno", queueno)
cmd6.ExecuteNonQuery()
            conn.Close()
          Catch ex As Exception
MessageBox.Show(ex.Message.ToString(), "Error Message")
            conn.Close()
          End Try
          Try
            conn.Open()
Dim cmd9 As New SqlCommand(deletetruckdetailstbl, conn)
cmd9.Parameters.AddWithValue("@queueno", queueno)
cmd9.ExecuteNonQuery()
conn.Close()
          Catch ex As Exception
MessageBox.Show(ex.Message.ToString(), "Error Message")
            conn.Close()
          End Try
          Try
            conn.Open()
Dim cmd11 As New SqlCommand(deleteactualunloaddetailstbl, conn)
cmd11.Parameters.AddWithValue("@queueno", queueno)
cmd11.ExecuteNonQuery()
            conn.Close()
          Catch ex As Exception
MessageBox.Show(ex.Message.ToString(), "Error Message")
            conn.Close()
          End Try
          Try
            conn.Open()
Dim cmd13 As New SqlCommand(deletedrtbl, conn)
cmd13.Parameters.AddWithValue("@queueno", queueno)
cmd13.ExecuteNonQuery()
            conn.Close()
          Catch ex As Exception
MessageBox.Show(ex.Message.ToString(), "Error Message")
            conn.Close()
          End Try
        End Using
        Button1.Enabled = False
        Label15.Text = ""
        Label7.Text = ""
        Label13.Text = ""
        TextBox1.Text = ""
        Label10.Text = ""
        Label11.Text = ""
        errorlbl.Text = ""
        DataGridView1.DataSource = Nothing
        DataGridView1.Refresh()
        End If

    End Sub

    Private Sub Button3_Click(sender As Object, e As EventArgs)
Handles Button3.Click

        DataGridView1.DataSource = Nothing

        DataGridView1.Refresh()

        Dim con As New SqlClient.SqlConnection

        Dim connectionString As String = "Server='" &
My.Settings.mServer & "';

    Database='" & My.Settings.mDB & "';

    User Id='" & My.Settings.mUserDB & "';

    Password='" & My.Settings.mPassDB & "';
MultipleActiveResultSets=True"

Dim insertquery As String = "insert into NoShow
(Queueno,ReceiverName,Bldg,Dock)
values(@queueno,@ReceiverName,@Bldg,@Dock)"

Dim deletebldg4dr1query As String = "delete from bldg4d1 where
Queueno=@queueno AND ReceiverName=@receivername AND
Bldg=@bldg AND Dock=@dock"

Dim deletebldg4dr2query As String = "delete from bldg4d2 where
Queueno=@queueno AND ReceiverName=@receivername AND
Bldg=@bldg AND Dock=@dock"

Dim deletebldg4dr3query As String = "delete from Bldg4d3 where
Queueno=@queueno AND ReceiverName=@receivername AND
Bldg=@bldg AND Dock=@dock"
```

```
Dim deletebldg4dr4query As String = "delete from Bldg4d4 where
Queueno=@queueno AND ReceiverName=@receivername AND
Bldg=@bldg AND Dock=@dock"

Dim deletebldg3dr1query As String = "delete from Bldg3 where
Queueno=@queueno AND ReceiverName=@receivername AND
Bldg=@bldg AND Dock=@dock"

Dim deletebldg6dr1query As String = "delete from Bldg6 where
Queueno=@queueno AND ReceiverName=@receivername AND
Bldg=@bldg AND Dock=@dock"

con.ConnectionString = connectionString

        Using conn As New SqlConnection(connectionString)

            Dim bldg As String = Label10.Text

            Dim Dock As String = Label11.Text

            Dim queueno As String = Label15.Text

            Try

                conn.Open()

                cmd = New SqlCommand(insertquery, conn)

                cmd.Parameters.AddWithValue("@queueno", queueno)

             cmd.Parameters.AddWithValue("@ReceiverName",
             receiverpublic.ReceiverName)

                cmd.Parameters.AddWithValue("@Bldg", bldg)

                cmd.Parameters.AddWithValue("@Dock", Dock)

                cmd.ExecuteNonQuery()

                MsgBox("Insert to Noshow successful")
            If bldg = "4" And Dock = "1" Then
                Try
cmd3 = New SqlCommand(deletebldg4dr1query, conn)
cmd3.Parameters.AddWithValue("@receivername",
receiverpublic.ReceiverName)
cmd3.Parameters.AddWithValue("@queueno", queueno)
cmd3.Parameters.AddWithValue("@bldg", "4")
cmd3.Parameters.AddWithValue("@dock", "1")
cmd3.ExecuteNonQuery()
                    Button3.Enabled = False
                    Button2.Enabled = True
                    Label15.Text = ""
                    Label10.Text = ""
                    Label11.Text = ""
                    TextBox1.Text = ""
                    Label7.Text = ""
                    Label13.Text = ""
                    MsgBox("Delete from queue good")
                Catch ex As Exception
                End Try
            ElseIf bldg.Equals("4") And Dock.Equals("2") Then
                cmd3 = New SqlCommand(deletebldg4dr2query, conn)
```

```
cmd3.Parameters.AddWithValue("@receivername",
receiverpublic.ReceiverName)

cmd3.Parameters.AddWithValue("@queueno", queueno)

        cmd3.Parameters.AddWithValue("@bldg", "4")

        cmd3.Parameters.AddWithValue("@dock", "2")

        cmd3.ExecuteNonQuery()

        Button3.Enabled = False

        Button2.Enabled = True

        Label15.Text = ""

        Label10.Text = ""

        Label11.Text = ""

        TextBox1.Text = ""

        Label7.Text = ""

        Label13.Text = ""

      ElseIf bldg.Equals("4") And Dock.Equals("3") Then

cmd3 = New SqlCommand(deletebldg4dr3query, conn)

cmd3.Parameters.AddWithValue("@receivername",
receiverpublic.ReceiverName)

cmd3.Parameters.AddWithValue("@queueno", queueno)

cmd3.Parameters.AddWithValue("@bldg", "4")

cmd3.Parameters.AddWithValue("@dock", "3")

cmd3.ExecuteNonQuery()
```

```
          Button3.Enabled = False

          Button2.Enabled = True

          Label15.Text = ""

          Label10.Text = ""

          Label11.Text = ""

          TextBox1.Text = ""

          Label7.Text = ""

          Label13.Text = ""

        ElseIf bldg.Equals("4") And Dock.Equals("4") Then

          cmd3 = New SqlCommand(deletebldg4dr4query, conn)

cmd3.Parameters.AddWithValue("@receivername",
receiverpublic.ReceiverName)

cmd3.Parameters.AddWithValue("@queueno", queueno)

          cmd3.Parameters.AddWithValue("@bldg", "4")

          cmd3.Parameters.AddWithValue("@dock", "4")

          cmd3.ExecuteNonQuery()

          Button3.Enabled = False

          Button2.Enabled = True

          Label15.Text = ""

          Label10.Text = ""

          Label11.Text = ""

          TextBox1.Text = ""
          Label7.Text = ""

          Label13.Text = ""

        ElseIf bldg = "3" And Dock = "1" Then

          cmd3 = New SqlCommand(deletebldg3dr1query, conn)

cmd3.Parameters.AddWithValue("@receivername",
receiverpublic.ReceiverName)

cmd3.Parameters.AddWithValue("@queueno", queueno)

          cmd3.Parameters.AddWithValue("@bldg", "3")

          cmd3.Parameters.AddWithValue("@dock", "1")

          cmd3.ExecuteNonQuery()

          Button3.Enabled = False

          Button2.Enabled = True

          Label15.Text = ""

          Label10.Text = ""

          Label11.Text = ""

          TextBox1.Text = ""

          Label7.Text = ""

          Label13.Text = ""

        ElseIf bldg.Equals("6") And Dock.Equals("1") Then

cmd3 = New SqlCommand(deletebldg6dr1query, conn)

cmd3.Parameters.AddWithValue("@receivername",
receiverpublic.ReceiverName)
```

```
cmd3.Parameters.AddWithValue("@queueno", queueno)
                    cmd3.Parameters.AddWithValue("@bldg", "6")
                    cmd3.Parameters.AddWithValue("@dock", "1")
                    cmd3.ExecuteNonQuery()
                    Button3.Enabled = False
                    Button2.Enabled = True
                    Label15.Text = ""
                    Label10.Text = ""
                    Label11.Text = ""
                    TextBox1.Text = ""
                    Label7.Text = ""
                    Label13.Text = ""
                End If
                conn.Close()
            Catch ex As Exception
MessageBox.Show(ex.Message.ToString(), "Error Message")
            End Try
        End Using
    End Sub
    Private Sub Timer1_Tick(sender As Object, e As EventArgs)
Handles Timer1.Tick
        LoadTable()
        Label17.Text = DateTime.Now.ToString("MMMM dd, yyyy h:mm:ss tt")

        If NoShowlist.Item(0, 0).Value Is Nothing Then

            Button4.Enabled = False

        Else

            Button4.Enabled = True

        End If

        Dim Box As Integer = 0

        Dim Weight As Integer = 0

        Dim Units As Integer = 0

        For index As Integer = 0 To DataGridView1.RowCount - 1

            Box += Convert.ToInt32(DataGridView1.Rows(index).Cells(1).Value)

            Weight += Convert.ToInt32(DataGridView1.Rows(index).Cells(2).Value)

            Units += Convert.ToInt32(DataGridView1.Rows(index).Cells(3).Value)

        Next

        Label7.Text = Box

        Label13.Text = Units

        Label14.Text = Weight / DataGridView1.RowCount

End Sub
```

```
    Private Sub Button4_Click(sender As Object, e As EventArgs)
Handles Button4.Click

          Dim queueno As String

          Dim Bldg As String

          Dim Dock As String

          Dim receivername As String

          Button3.Enabled = False

          queueno = NoShowlist.Item(0, 0).Value.ToString

          receivername = NoShowlist.Item(1, 0).Value.ToString

          Bldg = NoShowlist.Item(2, 0).Value.ToString

          Dock = NoShowlist.Item(3, 0).Value.ToString

          Dim con As New SqlClient.SqlConnection

          Dim connectionString As String = "Server='" &
My.Settings.mServer & "';

    Database='" & My.Settings.mDB & "';

    User Id='" & My.Settings.mUserDB & "';

    Password='" & My.Settings.mPassDB & "';
MultipleActiveResultSets=True"

Dim deletequery As String = "delete from NoShow where
Queueno=@queueno And ReceiverName=@ReceiverName AND
Bldg=@Bldg AND Dock=@Dock"

Dim insertbldg4dr1query As String = "insert into bldg4d1
(queueno,Bldg,Dock) values(@queueno,@bldg,@dock)"
```

```
Dim insertbldg4dr2query As String = "insert into bldg4d2
(queueno,Bldg,Dock) values(@queueno,@bldg,@dock)"

Dim insertbldg4dr3query As String = "insert into Bldg4d3
(queueno,Bldg,Dock) values(@queueno,@bldg,@dock)"

Dim insertbldg4dr4query As String = "insert into Bldg4d4
(queueno,Bldg,Dock) values(@queueno,@bldg,@dock)"

Dim insertbldg3dr1query As String = "insert into Bldg3
(queueno,Bldg,Dock) values(@queueno,@bldg,@dock)"

Dim insertbldg6dr1query As String = "insert into Bldg6
(queueno,Bldg,Dock) values(@queueno,@bldg,@dock)"

        con.ConnectionString = connectionString

        Using conn As New SqlConnection(connectionString)

            Try

                conn.Open()

If Bldg.Equals("4") And Dock.Equals("1") Then

cmd3 = New SqlCommand(insertbldg4dr1query, conn)

cmd3.Parameters.AddWithValue("@queueno", queueno)

cmd3.Parameters.AddWithValue("@bldg", "4")

cmd3.Parameters.AddWithValue("@dock", "1")

cmd3.ExecuteNonQuery()

                    Try

cmd = New SqlCommand(deletequery, conn)

cmd.Parameters.AddWithValue("@queueno", queueno)

cmd.Parameters.AddWithValue("@ReceiverName", receivername)
```

```
            cmd.Parameters.AddWithValue("@Bldg", Bldg)
            cmd.Parameters.AddWithValue("@Dock", Dock)
            cmd.ExecuteNonQuery()
          Catch ex As Exception
MessageBox.Show(ex.Message.ToString(), "Error Message")
          End Try
        ElseIf Bldg.Equals("4") And Dock.Equals("2") Then
cmd3 = New SqlCommand(insertbldg4dr2query, conn)
cmd3.Parameters.AddWithValue("@queueno", queueno)
cmd3.Parameters.AddWithValue("@bldg", "4")
cmd3.Parameters.AddWithValue("@dock", "2")
cmd3.ExecuteNonQuery()
          Try
cmd = New SqlCommand(deletequery, conn)
cmd.Parameters.AddWithValue("@queueno", queueno)
cmd.Parameters.AddWithValue("@ReceiverName", receivername)
cmd.Parameters.AddWithValue("@Bldg", Bldg)
cmd.Parameters.AddWithValue("@Dock", Dock)
cmd.ExecuteNonQuery()
Catch ex As Exception
MessageBox.Show(ex.Message.ToString(), "Error Message")
          End Try
          ElseIf Bldg.Equals("4") And Dock.Equals("3") Then
cmd3 = New SqlCommand(insertbldg4dr3query, conn)
cmd3.Parameters.AddWithValue("@queueno", queueno)
cmd3.Parameters.AddWithValue("@bldg", "4")
cmd3.Parameters.AddWithValue("@dock", "3")
cmd3.ExecuteNonQuery()
            Try
cmd = New SqlCommand(deletequery, conn)
cmd.Parameters.AddWithValue("@queueno", queueno)
cmd.Parameters.AddWithValue("@ReceiverName", receivername)
cmd.Parameters.AddWithValue("@Bldg", Bldg)
cmd.Parameters.AddWithValue("@Dock", Dock)
cmd.ExecuteNonQuery()
 Catch ex As Exception
MessageBox.Show(ex.Message.ToString(), "Error Message")
            End Try
          ElseIf Bldg.Equals("4") And Dock.Equals("4") Then
cmd3 = New SqlCommand(insertbldg4dr4query, conn)
cmd3.Parameters.AddWithValue("@queueno", queueno)
cmd3.Parameters.AddWithValue("@bldg", "4")
cmd3.Parameters.AddWithValue("@dock", "4")
cmd3.ExecuteNonQuery()
```

```
            Try
cmd = New SqlCommand(deletequery, conn)
cmd.Parameters.AddWithValue("@queueno", queueno)
cmd.Parameters.AddWithValue("@ReceiverName", receivername)
cmd.Parameters.AddWithValue("@Bldg", Bldg)
cmd.Parameters.AddWithValue("@Dock", Dock)
cmd.ExecuteNonQuery()
            Catch ex As Exception
MessageBox.Show(ex.Message.ToString(), "Error Message")
            End Try
ElseIf Bldg.Equals("3") And Dock.Equals("1") Then
cmd3 = New SqlCommand(insertbldg3dr1query, conn)
cmd3.Parameters.AddWithValue("@queueno", queueno)
cmd3.Parameters.AddWithValue("@bldg", "3")
cmd3.Parameters.AddWithValue("@dock", "1")
cmd3.ExecuteNonQuery()
            Try
cmd = New SqlCommand(deletequery, conn)
cmd.Parameters.AddWithValue("@queueno", queueno)
cmd.Parameters.AddWithValue("@ReceiverName", receivername)
cmd.Parameters.AddWithValue("@Bldg", Bldg)
cmd.Parameters.AddWithValue("@Dock", Dock)
cmd.ExecuteNonQuery()
 Catch ex As Exception
MessageBox.Show(ex.Message.ToString(), "Error Message")
          End Try
        ElseIf Bldg.Equals("6") And Dock.Equals("1") Then
cmd3 = New SqlCommand(insertbldg6dr1query, conn)
cmd3.Parameters.AddWithValue("@queueno", queueno)
cmd3.Parameters.AddWithValue("@bldg", "6")
cmd3.Parameters.AddWithValue("@dock", "1")
cmd3.ExecuteNonQuery()
          Try
cmd = New SqlCommand(deletequery, conn)
cmd.Parameters.AddWithValue("@queueno", queueno)
cmd.Parameters.AddWithValue("@ReceiverName", receivername)
cmd.Parameters.AddWithValue("@Bldg", Bldg)
cmd.Parameters.AddWithValue("@Dock", Dock)
cmd.ExecuteNonQuery()
          Catch ex As Exception
MessageBox.Show(ex.Message.ToString(), "Error Message")
          End Try
        End If
```

```
                conn.Close()

            Catch ex As Exception

            End Try

        End Using

    End Sub

    Private Sub Logout_Click(sender As Object, e As EventArgs)
Handles Logout.Click

        Dim dialog As DialogResult

        dialog = MessageBox.Show("Are you sure you want to
Logout?", "Exit", MessageBoxButtons.YesNo)

        If dialog = DialogResult.No Then

        Else

            Dim logout As New DBconnection

            logout.ReceiverName = ""

            Me.Close()

            Login.Show()

        End If

    End Sub

    Private Sub DataGridView1_CellClick(sender As Object, e As
DataGridViewCellEventArgs) Handles DataGridView1.CellClick

Dim con As New SqlClient.SqlConnection

Dim connectionString As String = "Server='" & My.Settings.mServer
& "';
```

```
Database='" & My.Settings.mDB & "';

User Id='" & My.Settings.mUserDB & "';

Password='" & My.Settings.mPassDB & "';"

con.ConnectionString = connectionString

i = DataGridView1.CurrentRow.Index

Dim drval As String

Dim queueno As String

Dim nobox As String

Dim noWeight As String

Dim Units As String

drval = DataGridView1.Item(0, i).Value.ToString

nobox = DataGridView1.Item(1, i).Value.ToString

noWeight = DataGridView1.Item(2, i).Value.ToString

Units = DataGridView1.Item(3, i).Value.ToString

queueno = Label15.Text

Dim form As New Weight

form.Label5.Text = queueno

form.Label3.Text = drval

form.TextBox1.Text = nobox

form.TextBox2.Text = Units

form.TextBox3.Text = noWeight

form.ShowDialog()
```

```
    End Sub

    Private Sub DataGridView1_CellValueChanged(sender As Object, e As DataGridViewCellEventArgs) Handles DataGridView1.CellValueChanged

        Dim Box As Integer = 0

        Dim Weight As Integer = 0

        Dim Units As Integer = 0

        For index As Integer = 0 To DataGridView1.RowCount - 1

Box += Convert.ToInt32(DataGridView1.Rows(index).Cells(1).Value)

Weight += Convert.ToInt32(DataGridView1.Rows(index).Cells(2).Value)

Units += Convert.ToInt32(DataGridView1.Rows(index).Cells(3).Value)

        Next

        Label7.Text = Box

        Label13.Text = Units

        Label14.Text = Weight

    End Sub

End Class
```

Dashboard.vb

```
Imports System.Text.RegularExpressions

Imports System.Data.SqlClient

Imports System.Data
Imports CrystalDecisions.CrystalReports.Engine

Imports CrystalDecisions.Shared

Public Class Dashboard

    Public dbs As New BindingSource

    Public dbs1 As New BindingSource

    Public dbs2 As New BindingSource

    Public dbs3 As New BindingSource

    Public dbs4 As New BindingSource

    Public dbs5 As New BindingSource

    Dim cmd5 As New SqlCommand

    Dim i As Integer

    Dim transno As String

    Private Sub LoadDailyQueueingWeightKPI()

        Chart29.Series("Series1").Points.Clear()

        Chart29.DataSource = GetQueueWeightKPIDay()

        Chart29.Series("Series1").XValueMember = "Day"

        Chart29.Series("Series1").YValueMembers = "Average"

        Chart30.Series("Series1").Points.Clear()

        Chart30.DataSource = GetQueueWeightKPIMonth()

        Chart30.Series("Series1").XValueMember = "Month"

        Chart30.Series("Series1").YValueMembers = "Average"

    End Sub
    Private Sub LoadCustomWeightKPIData()

        Chart32.Series("Series1").Points.Clear()

        Chart32.DataSource = GetCustomWeightKPIDataDay()

        Chart32.Series("Series1").XValueMember = "Day"

        Chart32.Series("Series1").YValueMembers = "Average"

    End Sub

    Private Sub LoadDailyQueueingKPI()

        Chart28.Series("Series1").Points.Clear()

        Chart28.DataSource = GetQueueKPIDay()

        Chart28.Series("Series1").XValueMember = "Day"

        Chart28.Series("Series1").YValueMembers = "Daily KPI Queueing(min.)"

    End Sub

    Private Sub LoadCustomDailyQueueingKPI()

        Chart31.Series("Series1").Points.Clear()

        Chart31.DataSource = GetCustomQueueKPIDay()

        Chart31.Series("Series1").XValueMember = "Day"

        Chart31.Series("Series1").YValueMembers = "Daily KPI Queueing(min.)"

    End Sub

    Private Function GetCustomWeightKPIDataDay() As DataTable

        Dim dtChartdata As New DataTable()
```

```
Dim con As New SqlClient.SqlConnection

Dim connectionString As String = "Server='" & My.Settings.mServer
& "';

    Database='" & My.Settings.mDB & "';

    User Id='" & My.Settings.mUserDB & "';

    Password='" & My.Settings.mPassDB & "';"

Dim selectquery As String = "select AVG(dq.difference) as Average,
convert(varchar(5),dqi.Arrival_date,110) AS Day from
Donetruckdetails dq Inner Join DoneQueueInfotbl dqi on
dq.transno=dqi.transno where dqi.Arrival_date=@date1 group by
convert(varchar(5),dqi.Arrival_date,110)"

        con.ConnectionString = connectionString

        Try

            con.Open()

Dim cmd As New SqlCommand(selectquery, con)

cmd.Parameters.AddWithValue("@date1",
CDate(DateTimePicker16.Text).ToString("yyyy-MM-dd"))

            cmd.ExecuteNonQuery()

            Dim dr As SqlDataReader = cmd.ExecuteReader()

            dtChartdata.Load(dr)

        Catch ex As Exception

            MsgBox(ex.ToString)

        End Try
        Return dtChartdata

    End Function

    Private Function getAverage()

Dim con As New SqlClient.SqlConnection

Dim connectionString As String = "Server='" & My.Settings.mServer
& "';

    Database='" & My.Settings.mDB & "';

    User Id='" & My.Settings.mUserDB & "';

    Password='" & My.Settings.mPassDB &
"';MultipleActiveResultSets=True"

Dim selectave As String = "select COUNT(transno) as transcount
from DoneQueueInfotbl where Arrival_date=@date1"

Dim selectave1 As String = "select COUNT(dq.transno) as
'totalfailed', ROUND(AVG(CAST(UN.UnloadDiff as float)),2) as
difffailed, ROUND(AVG(CAST(UN.UnloadDiff as float)-
CAST(UN.UnloadTarget as float)),2) as intervalfailed,
ROUND(AVG(CAST(DA.actualtotalweight as float)),2) as
weightfailed, ROUND(AVG(CAST(DA.actualtotalnoofboxes as
float)),2) as boxesfailed ,ROUND(AVG(CAST(DA.actualtotalunits as
float)),2) as unitsfailed,ROUND(AVG(CAST(DT.no_of_helper as
float)),2) as helperfailed from DoneQueueInfotbl dq inner join
Unload UN on dq.transno=UN.transno inner join
DoneActualunloaddetails DA on dq.transno=DA.transno inner join
Donetruckdetails DT on dq.transno=DT.transno where
Arrival_date=@date1 and UN.UnloadGrade='Failed'"

Dim selectave2 As String = "select COUNT(dq.transno) as
totalpassed, Round(AVG(CAST(UN.UnloadDiff as float)),2) as
diffpassed, Round(AVG(CAST(UN.UnloadDiff as float)-
CAST(UN.UnloadTarget as float)),2) as intervalpassed,
Round(AVG(CAST(DA.actualtotalweight as float)),2) as
weightpassed, Round(AVG(CAST(DA.actualtotalnoofboxes as
float)),2) as boxespassed ,Round(AVG(CAST(DA.actualtotalunits
as float)),2) as unitspassed,Round(AVG(CAST(DT.no_of_helper as
float)),2) as helperpassed from DoneQueueInfotbl dq inner join
Unload UN on dq.transno=UN.transno inner join
DoneActualunloaddetails DA on dq.transno=DA.transno inner join
Donetruckdetails DT on dq.transno=DT.transno where
Arrival_date=@Date1 and UN.UnloadGrade='Passed'"

    con.ConnectionString = connectionString

    Try

      con.Open()

Dim cmd As New SqlCommand(selectave, con)

cmd.Parameters.AddWithValue("@date1",
CDate(DateTimePicker16.Text).ToString("yyyy-MM-dd"))

      cmd.ExecuteNonQuery()

      Dim dr As SqlDataReader

dr = cmd.ExecuteReader(CommandBehavior.CloseConnection)

      While dr.Read

Label37.Text = dr("transcount").ToString

      End While

Dim cmd1 As New SqlCommand(selectave1, con)

cmd1.Parameters.AddWithValue("@date1",
CDate(DateTimePicker16.Text).ToString("yyyy-MM-dd"))

cmd1.ExecuteNonQuery()

Dim dr2 As SqlDataReader
```

```
          dr2 =
cmd1.ExecuteReader(CommandBehavior.CloseConnection)

          While dr2.Read

             Label55.Text = dr2("totalfailed").ToString

             Label57.Text = dr2("difffailed").ToString

             Label59.Text = dr2("intervalfailed").ToString

             Label61.Text = dr2("weightfailed").ToString

             Label63.Text = dr2("boxesfailed").ToString

             Label65.Text = dr2("unitsfailed").ToString

             Label67.Text = dr2("helperfailed").ToString

          End While

          Dim cmd2 As New SqlCommand(selectave2, con)

cmd2.Parameters.AddWithValue("@date1",
CDate(DateTimePicker16.Text).ToString("yyyy-MM-dd"))

cmd2.ExecuteNonQuery()

Dim dr3 As SqlDataReader

dr3 = cmd2.ExecuteReader(CommandBehavior.CloseConnection)

          While dr3.Read

             Label56.Text = dr3("totalpassed").ToString

             Label58.Text = dr3("diffpassed").ToString

             Label60.Text = dr3("intervalpassed").ToString

             Label62.Text = dr3("weightpassed").ToString
```

```
                Label64.Text = dr3("boxespassed").ToString

                Label66.Text = dr3("unitspassed").ToString

                Label68.Text = dr3("helperpassed").ToString

            End While

            con.Close()

        Catch ex As Exception

            MsgBox(ex.ToString)

            con.Close()

        End Try

    End Function

    Private Function GetQueueWeightKPIDay() As DataTable

        Dim dtChartdata As New DataTable()

        Dim con As New SqlClient.SqlConnection

             Dim connectionString As String = "Server='" &
             My.Settings.mServer & "';

    Database='" & My.Settings.mDB & "';

    User Id='" & My.Settings.mUserDB & "';

    Password='" & My.Settings.mPassDB & "';"

Dim selectquery As String = "select Round(AVG(dq.difference),2)
as Average,  convert(varchar(5),dqi.Arrival_date,110) AS Day from
Donetruckdetails dq Inner Join DoneQueueInfotbl dqi on
dq.transno=dqi.transno group by
convert(varchar(5),dqi.Arrival_date,110)"

        con.ConnectionString = connectionString
        Try
            con.Open()
            Dim cmd As New SqlCommand(selectquery, con)
            cmd.ExecuteNonQuery()
            Dim dr As SqlDataReader = cmd.ExecuteReader()
            dtChartdata.Load(dr)
        Catch ex As Exception
            MsgBox(ex.ToString)
        End Try
        Return dtChartdata
    End Function
    Private Function GetQueueWeightKPIMonth() As DataTable
        Dim dtChartdata As New DataTable()
        Dim con As New SqlClient.SqlConnection
        Dim connectionString As String = "Server='" &
My.Settings.mServer & "';
    Database='" & My.Settings.mDB & "';
    User Id='" & My.Settings.mUserDB & "';
    Password='" & My.Settings.mPassDB & "';"
Dim selectquery As String = "select Round(AVG(dq.difference),2)
as Average, DATEPART(mm, dqi.Arrival_date) AS Month from
Donetruckdetails dq Inner Join DoneQueueInfotbl dqi on
dq.transno=dqi.transno group by DATEPART(mm,
dqi.Arrival_date)"
```

```
        con.ConnectionString = connectionString

        Try

            con.Open()

            Dim cmd As New SqlCommand(selectquery, con)

            cmd.ExecuteNonQuery()

            Dim dr As SqlDataReader = cmd.ExecuteReader()

            dtChartdata.Load(dr)

        Catch ex As Exception

            MsgBox(ex.ToString)

        End Try

        Return dtChartdata

    End Function

    Private Function GetQueueKPIDay() As DataTable

        Dim dtChartdata As New DataTable()

        Dim con As New SqlClient.SqlConnection

        Dim connectionString As String = "Server='" &
My.Settings.mServer & "';

    Database='" & My.Settings.mDB & "';

    User Id='" & My.Settings.mUserDB & "';

    Password='" & My.Settings.mPassDB & "';"

Donetruckdetails dq Inner Join DoneQueueInfotbl dqi on
dq.transno=dqi.transno group by DATEPART(dd, dqi.Arrival_date)"
```

```
Dim selectquery As String = "SELECT
CAST(AVG(DATEDIFF(mi,SUBSTRING(CONVERT(VARCHAR,CA
ST(Arrival_date AS DATETIME)+CAST(Arrival_time AS
DATETIME),120), 1, 19) ,dq.startunload)) AS Float)*60 as 'Daily
KPI Queueing(min.)',convert(varchar(5),Arrival_date,110) AS Day
From Donetruckdetails dq Inner Join DoneQueueInfotbl dqi on
dq.transno=dqi.transno group by
convert(varchar(5),dqi.Arrival_date,110)"

        con.ConnectionString = connectionString

        Try

            con.Open()

            Dim cmd As New SqlCommand(selectquery, con)

            cmd.ExecuteNonQuery()

            Dim dr As SqlDataReader = cmd.ExecuteReader()

            dtChartdata.Load(dr)

        Catch ex As Exception

            MsgBox(ex.ToString)

        End Try

        Return dtChartdata

    End Function

    Private Function GetCustomQueueKPIDay() As DataTable

        Dim dtChartdata As New DataTable()

        Dim con As New SqlClient.SqlConnection

             Dim connectionString As String = "Server='" &
             My.Settings.mServer & "';
```

```
     Database='" & My.Settings.mDB & "';

     User Id='" & My.Settings.mUserDB & "';

     Password='" & My.Settings.mPassDB & "';"

Dim selectquery As String = "Select CAST(AVG(DateDiff(mi,
SUBSTRING(Convert(VARCHAR, CAST(Arrival_date As
DATETIME) + CAST(Arrival_time As DATETIME), 120), 1, 19),
dq.startunload)) As Float)*60  As 'Daily KPI
Queueing(min.)',convert(varchar(5),Arrival_date,110) AS Day From
Donetruckdetails dq Inner Join DoneQueueInfotbl dqi on
dq.transno=dqi.transno where dqi.Arrival_date between @date1
and @date2 group by convert(varchar(5),dqi.Arrival_date,110)"

          con.ConnectionString = connectionString

          Try

               con.Open()

               Dim cmd As New SqlCommand(selectquery, con)

               cmd.Parameters.AddWithValue("@date1",
CDate(DateTimePicker14.Text).ToString("yyyy-MM-dd"))

               cmd.Parameters.AddWithValue("@date2",
CDate(DateTimePicker13.Text).ToString("yyyy-MM-dd"))

               cmd.ExecuteNonQuery()

               Dim dr As SqlDataReader = cmd.ExecuteReader()

               dtChartdata.Load(dr)

          Catch ex As Exception

               MsgBox(ex.ToString)

          End Try
```

```
    Return dtChartdata
End Function
Private Sub LoadCustomData()
    Chart22.Series("Series1").Points.Clear()
    Chart22.DataSource = CustomTruckDay()
    Chart22.Series("Series1").XValueMember = "Day"
    Chart22.Series("Series1").YValueMembers = "total"
    Chart23.Series("Series1").Points.Clear()
    Chart23.DataSource = CustomPalletsDay()
    Chart23.Series("Series1").XValueMember = "Day"
    Chart23.Series("Series1").YValueMembers = "total"
    Chart24.Series("Series1").Points.Clear()
    Chart24.DataSource = CustomBoxesDay()
    Chart24.Series("Series1").XValueMember = "Day"
    Chart24.Series("Series1").YValueMembers = "total"
    Chart25.Series("Series1").Points.Clear()
    Chart25.DataSource = CustomUnitsDay()
    Chart25.Series("Series1").XValueMember = "Day"
    Chart25.Series("Series1").YValueMembers = "total"
End Sub
Private Sub LoadCustomDataVS()
    Chart22.Series("Series1").Points.Clear()
    Chart22.DataSource = CustomTruckDayVS()
    Chart22.Series("Series1").IsXValueIndexed = True
    Chart22.Series("Series1").XValueMember = "Day"
    Chart22.Series("Series1").YValueMembers = "total"
    Chart23.Series("Series1").Points.Clear()
    Chart23.DataSource = CustomPalletsDayVS()
    Chart23.Series("Series1").IsXValueIndexed = True
    Chart23.Series("Series1").XValueMember = "Day"
    Chart23.Series("Series1").YValueMembers = "total"
    Chart24.Series("Series1").Points.Clear()
    Chart24.DataSource = CustomBoxesDayVS()
    Chart24.Series("Series1").IsXValueIndexed = True
    Chart24.Series("Series1").XValueMember = "Day"
    Chart24.Series("Series1").YValueMembers = "total"
    Chart25.Series("Series1").Points.Clear()
    Chart25.DataSource = CustomUnitsDayVS()
    Chart25.Series("Series1").IsXValueIndexed = True
    Chart25.Series("Series1").XValueMember = "Day"
    Chart25.Series("Series1").YValueMembers = "total"
End Sub
Private Sub LoadDailyData()
    Chart7.Series("Series1").Points.Clear()
    Chart7.DataSource = GetTruckDataDay()

    Chart7.Series("Series1").XValueMember = "Day"

    Chart7.Series("Series1").YValueMembers = "total"

    Chart13.Series("Series1").Points.Clear()

    Chart13.DataSource = GetSumPalletsDataDay()

    Chart13.Series("Series1").XValueMember = "Day"

    Chart13.Series("Series1").YValueMembers = "Total"

    Chart16.Series("Series1").Points.Clear()

    Chart16.DataSource = GetSumUnitsDataDay()

    Chart16.Series("Series1").XValueMember = "Day"

    Chart16.Series("Series1").YValueMembers = "Total"

    Chart10.Series("Series1").Points.Clear()

    Chart10.DataSource = GetSumBoxDataDay()

    Chart10.Series("Series1").XValueMember = "Day"

    Chart10.Series("Series1").YValueMembers = "Total"

End Sub

Private Sub LoadMonthlyData()

    Chart8.Series("Series1").Points.Clear()

    Chart8.DataSource = GetTruckDataMonth()

    Chart8.Series("Series1").XValueMember = "Month"

    Chart8.Series("Series1").YValueMembers = "total"

    Chart14.Series("Series1").Points.Clear()
    Chart14.DataSource = GetSumPalletsDataMonth()
    Chart14.Series("Series1").XValueMember = "Month"
    Chart14.Series("Series1").YValueMembers = "Total"
    Chart11.Series("Series1").Points.Clear()
    Chart11.DataSource = GetSumBoxDataMonth()
    Chart11.Series("Series1").XValueMember = "Month"
    Chart11.Series("Series1").YValueMembers = "Total"
    Chart17.Series("Series1").Points.Clear()
    Chart17.DataSource = GetSumUnitsDataMonth()
    Chart17.Series("Series1").XValueMember = "Month"
    Chart17.Series("Series1").YValueMembers = "Total"
End Sub
Private Sub LoadYearlyData()
    Chart9.Series("Series1").Points.Clear()
    Chart9.DataSource = GetTruckDataYear()
    Chart9.Series("Series1").XValueMember = "Year"
    Chart9.Series("Series1").YValueMembers = "total"
    Chart12.Series("Series1").Points.Clear()
    Chart12.DataSource = GetSumBoxDataYear()
    Chart12.Series("Series1").XValueMember = "Year"
    Chart12.Series("Series1").YValueMembers = "Total"
```

```
        Chart15.Series("Series1").Points.Clear()

        Chart15.DataSource = GetSumPalletsDataYear()

        Chart15.Series("Series1").XValueMember = "Year"

        Chart15.Series("Series1").YValueMembers = "Total"

        Chart18.Series("Series1").Points.Clear()

        Chart18.DataSource = GetSumUnitsDataYear()

        Chart18.Series("Series1").XValueMember = "Year"

        Chart18.Series("Series1").YValueMembers = "Total"

    End Sub

    Private Sub LoadCategory()

        Chart1.Series.Clear()

Chart1.DataBindCrossTable(GetData().DefaultView, "category",
"Year", "Total", "Label=Total")

        Chart2.Series.Clear()

Chart2.DataBindCrossTable(GetDataMonth().DefaultView,
"category", "Month", "Total", "Label=Total")

        Chart3.Series.Clear()

Chart3.DataBindCrossTable(GetDataDay().DefaultView, "category",
"Day", "Total", "Label=Total")

    End Sub

    Private Sub LoadSupplier()

        Chart6.Series.Clear()
```

```
Chart6.DataBindCrossTable(GetDeliveryDataYear().DefaultView,
"vendorname", "Year", "Total", "Label=Total")

        Chart5.Series.Clear()

Chart5.DataBindCrossTable(GetDeliveryDataMonth().DefaultView,
"vendorname", "Month", "Total", "Label=Total")

        Chart4.Series.Clear()

Chart4.DataBindCrossTable(GetDeliveryDataDay().DefaultView,
"vendorname", "Day", "Total", "Label=Total")

    End Sub

    Private Sub LoadData()

        Chart19.Series("Series1").Points.Clear()

        Chart19.DataSource = GetKPIDataDay()

        Chart19.Series("Series1").XValueMember = "Day"

        Chart19.Series("Series1").YValueMembers = "Average"

        Chart20.Series("Series1").Points.Clear()

        Chart20.DataSource = GetKPIDataMonth()

        Chart20.Series("Series1").XValueMember = "Month"

        Chart20.Series("Series1").YValueMembers = "Average"

    End Sub

    Private Sub LoadCustomKPIData()

        Chart21.Series("Series1").Points.Clear()

        Chart21.DataSource = GetCustomKPIDataDay()
        Chart21.Series("Series1").XValueMember = "Day"

        Chart21.Series("Series1").YValueMembers = "Average"

    End Sub

    Private Sub LoadCustomSupplier()

        Chart27.Series.Clear()

Chart27.DataBindCrossTable(GetCustomSupplierData().DefaultVie
w, "vendorname", "Day", "total", "Label=total")

    End Sub

    Private Sub LoadCustomSupplierVS()

        Chart27.Series.Clear()

Chart27.DataBindCrossTable(GetCustomSupplierDataVS().Default
View, "vendorname", "Day", "Total", "Label=Total")

    End Sub

    Private Sub LoadCustomCategory()

        Chart26.Series.Clear()

Chart26.DataBindCrossTable(GetCustomData().DefaultView,
"category", "Day", "Total", "Label=Total")

    End Sub

    Private Sub LoadCustomCategoryVS()

        Chart26.Series.Clear()

Chart26.DataBindCrossTable(GetCustomDataVS().DefaultView,
"category", "Day", "Total", "Label=Total")
```

```
    End Sub

    Public Sub Queuechecktbl()

        Dim con As New SqlClient.SqlConnection

        Dim connectionString As String = "Server='" &
My.Settings.mServer & "';

    Database='" & My.Settings.mDB & "';

    User Id='" & My.Settings.mUserDB & "';

    Password='" & My.Settings.mPassDB & "';"

        con.ConnectionString = connectionString

Dim selectquery2 As String = "Select dq.Arrival_date as Date,
dq.transno as 'Trans. No.', E.GuardGrade as 'Guard Grade',
GV.GVGrade as 'Guard-Verify Grade', V.VGrade as 'Verify Grade',
VR.VRGrade as 'Verify-Receive Grade',RD.RDGrade as 'Receiver-
Dock Grade',DU.DUGrade as 'Dock-Unload Grade',U.UnloadGrade
as 'Unloading Grade',UR.URGrade as 'Unload-End Receive Grade'
from DoneQueueInfotbl dq inner join Encode E on
dq.transno=E.transno inner join GV GV on GV.transno=dq.transno
inner join Verify V on dq.transno=V.transno inner join VR VR on
dq.transno=VR.transno inner join RD RD on
dq.transno=RD.transno inner join DU DU on
dq.transno=DU.transno inner join Unload U on
dq.transno=U.transno inner join UR UR on dq.transno=UR.transno

WHERE dq.Arrival_date between @date1 and @date2 order by
len(dq.transno), dq.transno"

        Try

            con.Open()
```

```
cmd5 = New SqlCommand(selectquery2, con)

cmd5.Parameters.AddWithValue("@date1",
CDate(DateTimePicker14.Text).ToString("yyyy-MM-dd"))

cmd5.Parameters.AddWithValue("@date2",
CDate(DateTimePicker13.Text).ToString("yyyy-MM-dd"))

            Dim da As New SqlDataAdapter(cmd5)

            Dim dt As New DataTable()

            da.Fill(dt)

            dbs4.DataSource = dt

            DataGridView5.DataSource = dbs4

            DataGridView5.ReadOnly = True

        Catch ex As Exception

MessageBox.Show(ex.Message.ToString(), "Error Message")

        End Try

    End Sub

    Private Sub LoadArrived()

        Dim con As New SqlClient.SqlConnection

        Dim connectionString As String = "Server='" &
My.Settings.mServer & "';

        Database ='" & My.Settings.mDB & "';

    User Id='" & My.Settings.mUserDB & "';

    Password='" & My.Settings.mPassDB &
"';MultipleActiveResultSets=True"
```

```
        Dim selectquery As String = ""

        con.ConnectionString = connectionString

Dim selecttruck As String = "select count(transno) as total from
DoneQueueInfotbl where Arrival_date between @date1 AND
@date2"

Dim selectboxes As String = "Select
sum(Cast(dq.actualtotalnoofboxes As INT)) As Total from
DoneActualunloaddetails dq Inner Join DoneQueueInfotbl dqi On
dq.transno=dqi.transno where dqi.Arrival_date between @date1
And @date2"

Dim selectpallets As String = "select
sum(Cast(dq.actualtotalnoofpallet AS INT)) As Total from
DoneActualunloaddetails dq Inner Join DoneQueueInfotbl dqi on
dq.transno=dqi.transno where dqi.Arrival_date between @date1
And @date2"

Dim selectunits As String = "select sum(Cast(dq.actualtotalunits AS
INT)) as Total from DoneActualunloaddetails dq Inner Join
DoneQueueInfotbl dqi on  dq.transno=dqi.transno where
dqi.Arrival_date between @date1 And @date2"

Dim selectkpi As String = "select AVG(dq.difference)/60 as Average
from Donetruckdetails dq Inner Join DoneQueueInfotbl dqi on
dq.transno=dqi.transno where dqi.Arrival_date between @date1
And @date2"

Dim selectadvance As String = "Select sum(Cast(dq.advancedel As
INT)) As Total from Donetruckdetails dq Inner Join
DoneQueueInfotbl dqi On  dq.transno=dqi.transno where
dqi.Arrival_date between @date1 And @date2"

Dim selectshort As String = "Select sum(Cast(dq.shortqty As INT))
As Total from Donetruckdetails dq Inner Join DoneQueueInfotbl dqi
```

```
On  dq.transno=dqi.transno where dqi.Arrival_date between
@date1 And @date2"

Dim selectover As String = "Select sum(Cast(dq.overqty As INT))
As Total from Donetruckdetails dq Inner Join DoneQueueInfotbl dqi
On  dq.transno=dqi.transno where dqi.Arrival_date between
@date1 And @date2"

Dim selectreenterpo As String = "Select sum(Cast(dq.reenterpo As
INT)) As Total from Donetruckdetails dq Inner Join
DoneQueueInfotbl dqi On  dq.transno=dqi.transno where
dqi.Arrival_date between @date1 And @date2"

Dim selectunknownitem As String = "Select sum(Cast(dq.unknown
As INT)) As Total from Donetruckdetails dq Inner Join
DoneQueueInfotbl dqi On  dq.transno=dqi.transno where
dqi.Arrival_date between @date1 And @date2"

        Try

            con.Open()

            Dim cmd As New SqlCommand(selecttruck, con)

            cmd.Parameters.AddWithValue("@date1",
CDate(DateTimePicker1.Text).ToString("yyyy-MM-dd"))

            cmd.Parameters.AddWithValue("@date2",
CDate(DateTimePicker2.Text).ToString("yyyy-MM-dd"))

            Try

                cmd.ExecuteNonQuery()

                Dim dr As SqlClient.SqlDataReader

dr = cmd.ExecuteReader(CommandBehavior.CloseConnection)

                While dr.Read
```

```
                    Label1.Text = dr("total").ToString
                End While
            Catch ex As Exception
                MsgBox(ex.ToString)
                con.Close()
            End Try
            con.Close()
        Catch ex As Exception
            MsgBox(ex.ToString)
            con.Close()
        End Try
        Try
            con.Open()
            Dim cmd As New SqlCommand(selectboxes, con)
            cmd.Parameters.AddWithValue("@date1",
CDate(DateTimePicker1.Text).ToString("yyyy-MM-dd"))
            cmd.Parameters.AddWithValue("@date2",
CDate(DateTimePicker2.Text).ToString("yyyy-MM-dd"))
            Try
                cmd.ExecuteNonQuery()
                Dim dr1 As SqlClient.SqlDataReader
dr1 = cmd.ExecuteReader(CommandBehavior.CloseConnection)
```

```
        While dr1.Read
            If dr1.IsDBNull(0) Then
                Label3.Text = "0"
            Else
                Label3.Text = dr1("Total").ToString
            End If
        End While
    Catch ex As Exception
        MsgBox(ex.ToString)
        con.Close()
    End Try
    con.Close()
Catch ex As Exception
    MsgBox(ex.ToString)
    con.Close()
End Try
Try
    con.Open()
    Dim cmd As New SqlCommand(selectpallets, con)
    cmd.Parameters.AddWithValue("@date1",
CDate(DateTimePicker1.Text).ToString("yyyy-MM-dd"))
```

```
cmd.Parameters.AddWithValue("@date2",
CDate(DateTimePicker2.Text).ToString("yyyy-MM-dd"))
            Try
                cmd.ExecuteNonQuery()
                Dim dr2 As SqlClient.SqlDataReader
                dr2 =
cmd.ExecuteReader(CommandBehavior.CloseConnection)
                While dr2.Read
                    If dr2.IsDBNull(0) Then
                        Label4.Text = "0"
                    Else
                        Label4.Text = dr2("Total").ToString
                    End If
                End While
            Catch ex As Exception
                MsgBox(ex.ToString)
                con.Close()
            End Try
            con.Close()
        Catch ex As Exception
            MsgBox(ex.ToString)
            con.Close()
        End Try
        Try
            con.Open()
            Dim cmd As New SqlCommand(selectunits, con)
            cmd.Parameters.AddWithValue("@date1",
CDate(DateTimePicker1.Text).ToString("yyyy-MM-dd"))
            cmd.Parameters.AddWithValue("@date2",
CDate(DateTimePicker2.Text).ToString("yyyy-MM-dd"))
            Try
                cmd.ExecuteNonQuery()
                Dim dr3 As SqlClient.SqlDataReader
                dr3 =
cmd.ExecuteReader(CommandBehavior.CloseConnection)
                While dr3.Read
                    If dr3.IsDBNull(0) Then
                        Label5.Text = "0"
                    Else
                        Label5.Text = dr3("Total").ToString
                    End If
                End While
            Catch ex As Exception
                MsgBox(ex.ToString)
                con.Close()
            End Try
            con.Close()
        Catch ex As Exception
            MsgBox(ex.ToString)
            con.Close()
        End Try
        Try
            con.Open()
            Dim cmd As New SqlCommand(selectkpi, con)
            cmd.Parameters.AddWithValue("@date1", CDate(DateTimePicker1.Text).ToString("yyyy-MM-dd"))
            cmd.Parameters.AddWithValue("@date2", CDate(DateTimePicker2.Text).ToString("yyyy-MM-dd"))
            Try
                cmd.ExecuteNonQuery()
                Dim dr4 As SqlClient.SqlDataReader
                dr4 = cmd.ExecuteReader(CommandBehavior.CloseConnection)
                While dr4.Read
                    If dr4.IsDBNull(0) Then
                        Label6.Text = "0"
                    Else
                        Label6.Text = dr4("Average").ToString
                    End If
                End While
            Catch ex As Exception
                MsgBox(ex.ToString)
                con.Close()
            End Try
            con.Close()
        Catch ex As Exception
            MsgBox(ex.ToString)
            con.Close()
        End Try
        Try
            con.Open()
            Dim cmd As New SqlCommand(selectadvance, con)
            cmd.Parameters.AddWithValue("@date1",
CDate(DateTimePicker1.Text).ToString("yyyy-MM-dd"))
            cmd.Parameters.AddWithValue("@date2",
CDate(DateTimePicker2.Text).ToString("yyyy-MM-dd"))
            Try
                cmd.ExecuteNonQuery()
                Dim dr5 As SqlClient.SqlDataReader
dr5 = cmd.ExecuteReader(CommandBehavior.CloseConnection)
```

```
                While dr5.Read
                    If dr5.IsDBNull(0) Then
                        Label38.Text = "0"
                    Else
                        Label38.Text = dr5("Total").ToString
                    End If
                End While
            Catch ex As Exception
                MsgBox(ex.ToString)
                con.Close()
            End Try
            con.Close()
        Catch ex As Exception
            MsgBox(ex.ToString)
            con.Close()
        End Try
        Try
            con.Open()
            Dim cmd As New SqlCommand(selectshort, con)
cmd.Parameters.AddWithValue("@date1",
CDate(DateTimePicker1.Text).ToString("yyyy-MM-dd"))
```

```
            cmd.Parameters.AddWithValue("@date2",
CDate(DateTimePicker2.Text).ToString("yyyy-MM-dd"))

            Try

                cmd.ExecuteNonQuery()

                Dim dr9 As SqlClient.SqlDataReader

dr9 = cmd.ExecuteReader(CommandBehavior.CloseConnection)

                While dr9.Read

                    If dr9.IsDBNull(0) Then

                        Label39.Text = "0"

                    Else

                        Label39.Text = dr9("Total").ToString

                    End If

                End While

            Catch ex As Exception

                MsgBox(ex.ToString)

                con.Close()

            End Tr

            con.Close()

        Catch ex As Exception

            MsgBox(ex.ToString)

            con.Close()

        End Try
```

```
Try
    con.Open()
    Dim cmd As New SqlCommand(selectover, con)
    cmd.Parameters.AddWithValue("@date1", CDate(DateTimePicker1.Text).ToString("yyyy-MM-dd"))
    cmd.Parameters.AddWithValue("@date2", CDate(DateTimePicker2.Text).ToString("yyyy-MM-dd"))
    Try
        cmd.ExecuteNonQuery()
        Dim dr6 As SqlClient.SqlDataReader
        dr6 = cmd.ExecuteReader(CommandBehavior.CloseConnection)
        While dr6.Read
            If dr6.IsDBNull(0) Then
                Label40.Text = "0"
            Else
                Label40.Text = dr6("Total").ToString
            End If
        End While
    Catch ex As Exception
        MsgBox(ex.ToString)
        con.Close()
    End Try
```

```
            con.Close()

        Catch ex As Exception

            MsgBox(ex.ToString)

            con.Close()

        End Try

        Try

            con.Open()

            Dim cmd As New SqlCommand(selectreenterpo, con)

            cmd.Parameters.AddWithValue("@date1",
CDate(DateTimePicker1.Text).ToString("yyyy-MM-dd"))

            cmd.Parameters.AddWithValue("@date2",
CDate(DateTimePicker2.Text).ToString("yyyy-MM-dd"))

            Try

                cmd.ExecuteNonQuery()

                Dim dr7 As SqlClient.SqlDataReader

dr7 = cmd.ExecuteReader(CommandBehavior.CloseConnection)

                While dr7.Read

                    If dr7.IsDBNull(0) Then

                        Label41.Text = "0"

                    Else

                        Label41.Text = dr7("Total").ToString

                    End If
                End While
            Catch ex As Exception
                MsgBox(ex.ToString)
                con.Close()
            End Try
            con.Close()
        Catch ex As Exception
            MsgBox(ex.ToString)
            con.Close()
        End Try
        Try
            con.Open()
            Dim cmd As New SqlCommand(selectunknownitem, con)
            cmd.Parameters.AddWithValue("@date1",
CDate(DateTimePicker1.Text).ToString("yyyy-MM-dd"))
            cmd.Parameters.AddWithValue("@date2",
CDate(DateTimePicker2.Text).ToString("yyyy-MM-dd"))
            Try
                cmd.ExecuteNonQuery()
                Dim dr8 As SqlClient.SqlDataReader
dr8 = cmd.ExecuteReader(CommandBehavior.CloseConnection)
                While dr8.Read
```

```
                    If dr8.IsDBNull(0) Then
                        Label42.Text = "0"
                    Else
                        Label42.Text = dr8("Total").ToString
                    End If
                End While
            Catch ex As Exception
                MsgBox(ex.ToString)
                con.Close()
            End Try
            con.Close()
        Catch ex As Exception
            MsgBox(ex.ToString)
            con.Close()
        End Try
    End Sub
    Private Function GetCustomData() As DataTable
        Dim dtChartdata As New DataTable()
        Dim con As New SqlClient.SqlConnection
             Dim connectionString As String = "Server='" &
             My.Settings.mServer & "'";
        Database ='" & My.Settings.mDB & "'";
```

```
   User Id='" & My.Settings.mUserDB & "';

   Password='" & My.Settings.mPassDB & "';"

Dim selectquery As String = "select count(dhl.category) as Total
,dhl.category, dqi.Arrival_date AS Day from
DoneActualunloaddetails dq Inner Join DoneQueueInfotbl dqi on
dq.transno=dqi.transno Inner Join Donetruckdetails dhl on
dqi.transno=dhl.transno where dqi.Arrival_date between @date1
AND @date2 group by dhl.category,dqi.Arrival_date"

      con.ConnectionString = connectionString

      Try

         con.Open()

         Dim cmd As New SqlCommand(selectquery, con)

         cmd.Parameters.AddWithValue("@date1",
CDate(DateTimePicker6.Text).ToString("yyyy-MM-dd"))

         cmd.Parameters.AddWithValue("@date2",
CDate(DateTimePicker5.Text).ToString("yyyy-MM-dd"))

         cmd.ExecuteNonQuery()

         Dim dr As SqlDataReader = cmd.ExecuteReader()

         dtChartdata.Load(dr)

      Catch ex As Exception

         MsgBox(ex.ToString)

      End Try

      Return dtChartdata

   End Function
```

```
    Private Function GetCustomDataVS() As DataTable

Dim dtChartdata As New DataTable()

Dim con As New SqlClient.SqlConnection

Dim connectionString As String = "Server='" & My.Settings.mServer
& "';

        Database ='" & My.Settings.mDB & "';

    User Id='" & My.Settings.mUserDB & "';

    Password='" & My.Settings.mPassDB & "';"

Dim selectquery As String = "select count(dhl.category) as Total
,dhl.category, dqi.Arrival_date AS Day from
DoneActualunloaddetails dq Inner Join DoneQueueInfotbl dqi on
dq.transno=dqi.transno Inner Join Donetruckdetails dhl on
dqi.transno=dhl.transno where dqi.Arrival_date IN( @date1 ,
@date2) group by dhl.category,dqi.Arrival_date"

        con.ConnectionString = connectionString

        Try

            con.Open()

            Dim cmd As New SqlCommand(selectquery, con)

            cmd.Parameters.AddWithValue("@date1",
CDate(DateTimePicker6.Text).ToString("yyyy-MM-dd"))

            cmd.Parameters.AddWithValue("@date2",
CDate(DateTimePicker5.Text).ToString("yyyy-MM-dd"))

            cmd.ExecuteNonQuery()

            Dim dr As SqlDataReader = cmd.ExecuteReader()

            dtChartdata.Load(dr)
        Catch ex As Exception

            MsgBox(ex.ToString)

        End Try

        Return dtChartdata

    End Function

    Private Function GetCustomSupplierData() As DataTable

        Dim dtChartdata1 As New DataTable()

        Dim con As New SqlClient.SqlConnection

        Dim connectionString As String = "Server='" &
My.Settings.mServer & "';

        Database ='" & My.Settings.mDB & "';

    User Id='" & My.Settings.mUserDB & "';

    Password='" & My.Settings.mPassDB & "';"

Dim selectquery1 As String = "select count(dq.vendorname) as
total,dq.vendorname, convert(varchar(5),dqi.Arrival_date,110) AS
Day from Donetruckdetails dq Inner Join DoneQueueInfotbl dqi on
dq.transno=dqi.transno where dqi.Arrival_date between @date1
AND @date2 group by
dq.vendorname,convert(varchar(5),dqi.Arrival_date,110)"

        con.ConnectionString = connectionString

        Try

            con.Open()

            Dim cmd1 As New SqlCommand(selectquery1, con)
```

```
          cmd1.Parameters.AddWithValue("@date1",
CDate(DateTimePicker8.Text).ToString("yyyy-MM-dd"))

          cmd1.Parameters.AddWithValue("@date2",
CDate(DateTimePicker7.Text).ToString("yyyy-MM-dd"))

          cmd1.ExecuteNonQuery()

          Dim dr2 As SqlDataReader = cmd1.ExecuteReader()

          dtChartdata1.Load(dr2)

      Catch ex As Exception

          MsgBox(ex.ToString)

      End Try

      Return dtChartdata1

   End Function

   Private Function GetCustomSupplierDataVS() As DataTable

      Dim dtChartdata As New DataTable()

      Dim con As New SqlClient.SqlConnection

          Dim connectionString As String = "Server='" &
          My.Settings.mServer & "';

      Database ='" & My.Settings.mDB & "';

   User Id='" & My.Settings.mUserDB & "';

   Password='" & My.Settings.mPassDB & "';"

Dim selectquery As String = "Select count(dq.vendorname) As
total,dq.vendorname, convert(varchar(5),dqi.Arrival_date,110) As
Day from Donetruckdetails dq Inner Join DoneQueueInfotbl dqi On
dq.transno=dqi.transno where dqi.Arrival_date IN (@date1,
@date2) group by
dq.vendorname,convert(varchar(5),dqi.Arrival_date,110)"

      con.ConnectionString = connectionString

      Try

          con.Open()

          Dim cmd As New SqlCommand(selectquery, con)

          cmd.Parameters.AddWithValue("@date1",
CDate(DateTimePicker8.Text).ToString("yyyy-MM-dd"))

          cmd.Parameters.AddWithValue("@date2",
CDate(DateTimePicker7.Text).ToString("yyyy-MM-dd"))

          cmd.ExecuteNonQuery()

          Dim dr As SqlDataReader = cmd.ExecuteReader()

          dtChartdata.Load(dr)

      Catch ex As Exception

          MsgBox(ex.ToString)

      End Try

      Return dtChartdata

   End Function

   Private Function GetData() As DataTable

      Dim dtChartdata As New DataTable()

      Dim con As New SqlClient.SqlConnection

          Dim connectionString As String = "Server='" &
          My.Settings.mServer & "';
      Database ='" & My.Settings.mDB & "';
   User Id='" & My.Settings.mUserDB & "';
   Password='" & My.Settings.mPassDB & "';"
Dim selectquery As String = "select count(dq.category) as
total,dq.category, DATEPART(yyyy, dqi.Arrival_date) AS Year from
Donetruckdetails dq Inner Join DoneQueueInfotbl dqi on
dq.transno=dqi.transno group by dq.category,DATEPART(yyyy,
dqi.Arrival_date)"
      con.ConnectionString = connectionString
      Try
          con.Open()
          Dim cmd As New SqlCommand(selectquery, con)
          cmd.ExecuteNonQuery()
          Dim dr As SqlDataReader = cmd.ExecuteReader()
          dtChartdata.Load(dr)
      Catch ex As Exception
          MsgBox(ex.ToString)
      End Try
      Return dtChartdata
   End Function
   Private Function GetDataMonth() As DataTable
      Dim dtChartdata As New DataTable()
        Dim con As New SqlClient.SqlConnection

        Dim connectionString As String = "Server='" &
My.Settings.mServer & "';

    Database='" & My.Settings.mDB & "';

    User Id='" & My.Settings.mUserDB & "';

    Password='" & My.Settings.mPassDB & "';"

Dim selectquery As String = "select count(dq.category) as
total,dq.category, DATEPART(mm, dqi.Arrival_date) AS Month from
Donetruckdetails dq Inner Join DoneQueueInfotbl dqi on
dq.transno=dqi.transno group by dq.category,DATEPART(mm,
dqi.Arrival_date)"

        con.ConnectionString = connectionString

        Try

            con.Open()

            Dim cmd As New SqlCommand(selectquery, con)

            cmd.ExecuteNonQuery()

            Dim dr As SqlDataReader = cmd.ExecuteReader()

            dtChartdata.Load(dr)

        Catch ex As Exception

            MsgBox(ex.ToString)

        End Try

        Return dtChartdata

    End Function
    Private Function GetDataDay() As DataTable

        Dim dtChartdata As New DataTable()

        Dim con As New SqlClient.SqlConnection

        Dim connectionString As String = "Server='" &
My.Settings.mServer & "';

    Database='" & My.Settings.mDB & "';

    User Id='" & My.Settings.mUserDB & "';

    Password='" & My.Settings.mPassDB & "';"

Dim selectquery As String = "select count(dq.category) as
total,dq.category, convert(varchar(5),dqi.Arrival_date,110) AS Day
from Donetruckdetails dq Inner Join DoneQueueInfotbl dqi on
dq.transno=dqi.transno group by
dq.category,convert(varchar(5),dqi.Arrival_date,110)"

        con.ConnectionString = connectionString

        Try

            con.Open()

            Dim cmd As New SqlCommand(selectquery, con)

            cmd.ExecuteNonQuery()

            Dim dr As SqlDataReader = cmd.ExecuteReader()

            dtChartdata.Load(dr)

        Catch ex As Exception

            MsgBox(ex.ToString)

        End Try
        Return dtChartdata
    End Function
    Private Function GetDeliveryDataYear() As DataTable
        Dim dtChartdata As New DataTable()
        Dim con As New SqlClient.SqlConnection
        Dim connectionString As String = "Server='" &
My.Settings.mServer & "';
    Database='" & My.Settings.mDB & "';
    User Id='" & My.Settings.mUserDB & "';
    Password='" & My.Settings.mPassDB & "';"
Dim selectquery As String = "select count(dq.vendorname) as
total,dq.vendorname, DATEPART(yyyy, dqi.Arrival_date) AS Year
from Donetruckdetails dq Inner Join DoneQueueInfotbl dqi on
dq.transno=dqi.transno group by dq.vendorname,DATEPART(yyyy,
dqi.Arrival_date)"
        con.ConnectionString = connectionString
        Try
            con.Open()
            Dim cmd As New SqlCommand(selectquery, con)
            cmd.ExecuteNonQuery()
            Dim dr As SqlDataReader = cmd.ExecuteReader()
            dtChartdata.Load(dr)
        Catch ex As Exception
            MsgBox(ex.ToString)
        End Try

        Return dtChartdata

    End Function

    Private Function GetDeliveryDataMonth() As DataTable

        Dim dtChartdata As New DataTable()

        Dim con As New SqlClient.SqlConnection

        Dim connectionString As String = "Server='" &
My.Settings.mServer & "';

    Database='" & My.Settings.mDB & "';

    User Id='" & My.Settings.mUserDB & "';

    Password='" & My.Settings.mPassDB & "';"

Dim selectquery As String = "select count(dq.vendorname) as
total,dq.vendorname, DATEPART(mm, dqi.Arrival_date) AS Month
from Donetruckdetails dq Inner Join DoneQueueInfotbl dqi on
dq.transno=dqi.transno group by dq.vendorname,DATEPART(mm,
dqi.Arrival_date)"

        con.ConnectionString = connectionString

        Try

            con.Open()

            Dim cmd As New SqlCommand(selectquery, con)

            cmd.ExecuteNonQuery()

            Dim dr As SqlDataReader = cmd.ExecuteReader()

            dtChartdata.Load(dr)
        Catch ex As Exception

            MsgBox(ex.ToString)

        End Try

        Return dtChartdata

    End Function

    Private Function GetDeliveryDataDay() As DataTable

        Dim dtChartdata As New DataTable()

        Dim con As New SqlClient.SqlConnection

        Dim connectionString As String = "Server='" &
My.Settings.mServer & "';

    Database='" & My.Settings.mDB & "';

    User Id='" & My.Settings.mUserDB & "';

    Password='" & My.Settings.mPassDB & "';"

Dim selectquery As String = "select count(dq.vendorname) as
total,dq.vendorname, convert(varchar(5),dqi.Arrival_date,110) AS
Day from Donetruckdetails dq Inner Join DoneQueueInfotbl dqi on
dq.transno=dqi.transno group by
dq.vendorname,convert(varchar(5),dqi.Arrival_date,110)"

        con.ConnectionString = connectionString

        Try

            con.Open()

            Dim cmd As New SqlCommand(selectquery, con)

            cmd.ExecuteNonQuery()
            Dim dr As SqlDataReader = cmd.ExecuteReader()

            dtChartdata.Load(dr)

        Catch ex As Exception

            MsgBox(ex.ToString)

        End Try

        Return dtChartdata

    End Function

    Private Function GetTruckDataDay() As DataTable

        Dim dtChartdata As New DataTable()

        Dim con As New SqlClient.SqlConnection

        Dim connectionString As String = "Server='" &
My.Settings.mServer & "';

    Database='" & My.Settings.mDB & "';

    User Id='" & My.Settings.mUserDB & "';

    Password='" & My.Settings.mPassDB & "';"

Dim selectquery As String = "select count(dq.transno) as total,
convert(varchar(5),Arrival_date,110) AS Day from Donetruckdetails
dq Inner Join DoneQueueInfotbl dqi on  dq.transno=dqi.transno
group by convert(varchar(5),Arrival_date,110) "

        con.ConnectionString = connectionString

        Try

            con.Open()

            Dim cmd As New SqlCommand(selectquery, con)
            cmd.ExecuteNonQuery()

            Dim dr As SqlDataReader = cmd.ExecuteReader()

            dtChartdata.Load(dr)

        Catch ex As Exception

            MsgBox(ex.ToString)

        End Try

        Return dtChartdata

    End Function

    Private Function GetTruckDataMonth() As DataTable

        Dim dtChartdata As New DataTable()

        Dim con As New SqlClient.SqlConnection

        Dim connectionString As String = "Server='" &
My.Settings.mServer & "';

    Database='" & My.Settings.mDB & "';

    User Id='" & My.Settings.mUserDB & "';

    Password='" & My.Settings.mPassDB & "';"

Dim selectquery As String = "select count(dq.transno) as total,
DATEPART(mm, dqi.Arrival_date) AS Month from Donetruckdetails
dq Inner Join DoneQueueInfotbl dqi on  dq.transno=dqi.transno
group by DATEPART(mm, dqi.Arrival_date)"

        con.ConnectionString = connectionString

        Try

            con.Open()
            Dim cmd As New SqlCommand(selectquery, con)

            cmd.ExecuteNonQuery()

            Dim dr As SqlDataReader = cmd.ExecuteReader()

            dtChartdata.Load(dr)

        Catch ex As Exception

            MsgBox(ex.ToString)

        End Try

        Return dtChartdata

    End Function

    Private Function GetTruckDataYear() As DataTable

        Dim dtChartdata As New DataTable()

        Dim con As New SqlClient.SqlConnection

        Dim connectionString As String = "Server='" &
My.Settings.mServer & "';

    Database='" & My.Settings.mDB & "';

    User Id='" & My.Settings.mUserDB & "';

    Password='" & My.Settings.mPassDB & "';"

Dim selectquery As String = "select count(dq.transno) as total,
DATEPART(yyyy, dqi.Arrival_date) AS Year from Donetruckdetails
dq Inner Join DoneQueueInfotbl dqi on  dq.transno=dqi.transno
group by DATEPART(yyyy, dqi.Arrival_date)"

        con.ConnectionString = connectionString

        Try
```

```
            con.Open()

            Dim cmd As New SqlCommand(selectquery, con)

            cmd.ExecuteNonQuery()

            Dim dr As SqlDataReader = cmd.ExecuteReader()

            dtChartdata.Load(dr)

        Catch ex As Exception

            MsgBox(ex.ToString)

        End Try

        Return dtChartdata

    End Function

    Private Function GetSumBoxDataDay() As DataTable

        Dim dtChartdata As New DataTable()

        Dim con As New SqlClient.SqlConnection

        Dim connectionString As String = "Server='" &
My.Settings.mServer & "';

    Database='" & My.Settings.mDB & "';

    User Id='" & My.Settings.mUserDB & "';

    Password='" & My.Settings.mPassDB & "';"

Dim selectquery As String = "select
sum(Cast(dq.actualtotalnoofboxes AS INT))as Total,
convert(varchar(5),Arrival_date,110) AS Day from
DoneActualunloaddetails dq Inner Join DoneQueueInfotbl dqi on
dq.transno=dqi.transno group by
convert(varchar(5),Arrival_date,110) "
```

```
        con.ConnectionString = connectionString
        Try
            con.Open()
            Dim cmd As New SqlCommand(selectquery, con)
            cmd.ExecuteNonQuery()
            Dim dr As SqlDataReader = cmd.ExecuteReader()
            dtChartdata.Load(dr)
        Catch ex As Exception
            MsgBox(ex.ToString)
        End Try
        Return dtChartdata
    End Function
    Private Function GetSumBoxDataMonth() As DataTable
        Dim dtChartdata As New DataTable()
        Dim con As New SqlClient.SqlConnection
        Dim connectionString As String = "Server='" &
My.Settings.mServer & "';
    Database='" & My.Settings.mDB & "';
    User Id='" & My.Settings.mUserDB & "';
    Password='" & My.Settings.mPassDB & "';"
Dim selectquery As String = "select
sum(Cast(dq.actualtotalnoofboxes AS INT))as Total ,
DATEPART(mm, dqi.Arrival_date) AS Month from
```

```
DoneActualunloaddetails dq Inner Join DoneQueueInfotbl dqi on
dq.transno=dqi.transno group by DATEPART(mm,
dqi.Arrival_date)"

        con.ConnectionString = connectionString

        Try

            con.Open()

            Dim cmd As New SqlCommand(selectquery, con)

            cmd.ExecuteNonQuery()

            Dim dr As SqlDataReader = cmd.ExecuteReader()

            dtChartdata.Load(dr)

        Catch ex As Exception

            MsgBox(ex.ToString)

        End Try

        Return dtChartdata

    End Function

    Private Function GetSumBoxDataYear() As DataTable

        Dim dtChartdata As New DataTable()

        Dim con As New SqlClient.SqlConnection

        Dim connectionString As String = "Server='" &
My.Settings.mServer & "';

    Database='" & My.Settings.mDB & "';

    User Id='" & My.Settings.mUserDB & "';

    Password='" & My.Settings.mPassDB & "';"
```

```
Dim selectquery As String = "select
sum(Cast(dq.actualtotalnoofboxes AS INT))as Total ,
DATEPART(yyyy, dqi.Arrival_date) AS Year from
DoneActualunloaddetails dq Inner Join DoneQueueInfotbl dqi on
dq.transno=dqi.transno group by DATEPART(yyyy,
dqi.Arrival_date)"

        con.ConnectionString = connectionString

        Try

            con.Open()

            Dim cmd As New SqlCommand(selectquery, con)

            cmd.ExecuteNonQuery()

            Dim dr As SqlDataReader = cmd.ExecuteReader()

            dtChartdata.Load(dr)

        Catch ex As Exception

            MsgBox(ex.ToString)

        End Try

        Return dtChartdata

    End Function

    Private Function GetSumPalletsDataDay() As DataTable

        Dim dtChartdata As New DataTable()

        Dim con As New SqlClient.SqlConnection

        Dim connectionString As String = "Server='" &
My.Settings.mServer & "';

    Database='" & My.Settings.mDB & "';
    User Id='" & My.Settings.mUserDB & "';

    Password='" & My.Settings.mPassDB & "';"

Dim selectquery As String = "select
sum(Cast(dq.actualtotalnoofpallet AS INT))as Total,
convert(varchar(5),Arrival_date,110) AS Day from
DoneActualunloaddetails dq Inner Join DoneQueueInfotbl dqi on
dq.transno=dqi.transno group by
convert(varchar(5),Arrival_date,110) "

        con.ConnectionString = connectionString

        Try

            con.Open()

            Dim cmd As New SqlCommand(selectquery, con)

            cmd.ExecuteNonQuery()

            Dim dr As SqlDataReader = cmd.ExecuteReader()

            dtChartdata.Load(dr)

        Catch ex As Exception

            MsgBox(ex.ToString)

        End Try


        Return dtChartdata

    End Function

    Private Function GetSumPalletsDataMonth() As DataTable

        Dim dtChartdata As New DataTable()
        Dim con As New SqlClient.SqlConnection

        Dim connectionString As String = "Server='" &
My.Settings.mServer & "';

    Database='" & My.Settings.mDB & "';

    User Id='" & My.Settings.mUserDB & "';

    Password='" & My.Settings.mPassDB & "';"

        Dim selectquery As String = "select
sum(Cast(dq.actualtotalnoofpallet AS INT))as Total ,
DATEPART(mm, dqi.Arrival_date) AS Month from
DoneActualunloaddetails dq Inner Join DoneQueueInfotbl dqi on
dq.transno=dqi.transno group by DATEPART(mm,
dqi.Arrival_date)"

        con.ConnectionString = connectionString

        Try

            con.Open()

            Dim cmd As New SqlCommand(selectquery, con)

            cmd.ExecuteNonQuery()

            Dim dr As SqlDataReader = cmd.ExecuteReader()

            dtChartdata.Load(dr)

        Catch ex As Exception

            MsgBox(ex.ToString)

        End Try

        Return dtChartdata

    End Function
    Private Function GetSumPalletsDataYear() As DataTable
        Dim dtChartdata As New DataTable()
        Dim con As New SqlClient.SqlConnection
        Dim connectionString As String = "Server='" &
My.Settings.mServer & "';
    Database='" & My.Settings.mDB & "';
    User Id='" & My.Settings.mUserDB & "';
    Password='" & My.Settings.mPassDB & "';"
        Dim selectquery As String = "select
sum(Cast(dq.actualtotalnoofpallet AS INT))as Total ,
DATEPART(yyyy, dqi.Arrival_date) AS Year from
DoneActualunloaddetails dq Inner Join DoneQueueInfotbl dqi on
dq.transno=dqi.transno group by DATEPART(yyyy,
dqi.Arrival_date)"
        con.ConnectionString = connectionString
        Try
            con.Open()
            Dim cmd As New SqlCommand(selectquery, con)
            cmd.ExecuteNonQuery()
            Dim dr As SqlDataReader = cmd.ExecuteReader()
            dtChartdata.Load(dr)
        Catch ex As Exception
            MsgBox(ex.ToString)
        End Try
        Return dtChartdata

    End Function

    Private Function GetSumUnitsDataDay() As DataTable

        Dim dtChartdata As New DataTable()

        Dim con As New SqlClient.SqlConnection

        Dim connectionString As String = "Server='" &
My.Settings.mServer & "';

    Database='" & My.Settings.mDB & "';

    User Id='" & My.Settings.mUserDB & "';

    Password='" & My.Settings.mPassDB & "';"

        Dim selectquery As String = "select
sum(Cast(dq.actualtotalunits AS INT))as Total,
convert(varchar(5),Arrival_date,110) AS Day from
DoneActualunloaddetails dq Inner Join DoneQueueInfotbl dqi on
dq.transno=dqi.transno group by
convert(varchar(5),Arrival_date,110) "

        con.ConnectionString = connectionString

        Try

            con.Open()

            Dim cmd As New SqlCommand(selectquery, con)

            cmd.ExecuteNonQuery()

            Dim dr As SqlDataReader = cmd.ExecuteReader()

            dtChartdata.Load(dr)
        Catch ex As Exception

            MsgBox(ex.ToString)

        End Try

        Return dtChartdata

    End Function

    Private Function GetSumUnitsDataMonth() As DataTable

        Dim dtChartdata As New DataTable()

        Dim con As New SqlClient.SqlConnection

        Dim connectionString As String = "Server='" &
My.Settings.mServer & "';

    Database='" & My.Settings.mDB & "';

    User Id='" & My.Settings.mUserDB & "';

    Password='" & My.Settings.mPassDB & "';"

Dim selectquery As String = "select sum(Cast(dq.actualtotalunits
AS INT))as Total , DATEPART(mm, dqi.Arrival_date) AS Month
from DoneActualunloaddetails dq Inner Join DoneQueueInfotbl dqi
on  dq.transno=dqi.transno group by DATEPART(mm,
dqi.Arrival_date)"

        con.ConnectionString = connectionString

        Try

            con.Open()

            Dim cmd As New SqlCommand(selectquery, con)

            cmd.ExecuteNonQuery()
            Dim dr As SqlDataReader = cmd.ExecuteReader()

            dtChartdata.Load(dr)

        Catch ex As Exception

            MsgBox(ex.ToString)

        End Try

        Return dtChartdata

    End Function

    Private Function GetSumUnitsDataYear() As DataTable

        Dim dtChartdata As New DataTable()

        Dim con As New SqlClient.SqlConnection

        Dim connectionString As String = "Server='" &
My.Settings.mServer & "';

    Database='" & My.Settings.mDB & "';

    User Id='" & My.Settings.mUserDB & "';

    Password='" & My.Settings.mPassDB & "';"

Dim selectquery As String = "select sum(Cast(dq.actualtotalunits
AS INT))as Total , DATEPART(yyyy, dqi.Arrival_date) AS Year from
DoneActualunloaddetails dq Inner Join DoneQueueInfotbl dqi on
dq.transno=dqi.transno group by DATEPART(yyyy,
dqi.Arrival_date)"

        con.ConnectionString = connectionString

        Try

            con.Open()

            Dim cmd As New SqlCommand(selectquery, con)
            cmd.ExecuteNonQuery()

            Dim dr As SqlDataReader = cmd.ExecuteReader()

            dtChartdata.Load(dr)

        Catch ex As Exception

            MsgBox(ex.ToString)

        End Try

        Return dtChartdata

    End Function

    Private Function GetKPIDataDay() As DataTable

        Dim dtChartdata As New DataTable()

        Dim con As New SqlClient.SqlConnection

        Dim connectionString As String = "Server='" &
My.Settings.mServer & "';

    Database='" & My.Settings.mDB & "';

    User Id='" & My.Settings.mUserDB & "';

    Password='" & My.Settings.mPassDB & "';"

        Dim selectquery As String = "select AVG(dq.difference) as
Average,  convert(varchar(5),dqi.Arrival_date,110) AS Day from
Donetruckdetails dq Inner Join DoneQueueInfotbl dqi on
dq.transno=dqi.transno group by
convert(varchar(5),dqi.Arrival_date,110)"

        con.ConnectionString = connectionString

        Try

            con.Open()
            Dim cmd As New SqlCommand(selectquery, con)

            cmd.ExecuteNonQuery()

            Dim dr As SqlDataReader = cmd.ExecuteReader()

            dtChartdata.Load(dr)

        Catch ex As Exception

            MsgBox(ex.ToString)

        End Try

        Return dtChartdata

    End Function

    Private Function GetKPIDataMonth() As DataTable

        Dim dtChartdata As New DataTable()

        Dim con As New SqlClient.SqlConnection

        Dim connectionString As String = "Server='" &
My.Settings.mServer & "';

    Database='" & My.Settings.mDB & "';

    User Id='" & My.Settings.mUserDB & "';

    Password='" & My.Settings.mPassDB & "';"

Dim selectquery As String = "select AVG(dq.difference) as Average,
DATEPART(mm, dqi.Arrival_date) AS Month from Donetruckdetails
dq Inner Join DoneQueueInfotbl dqi on  dq.transno=dqi.transno
group by DATEPART(mm, dqi.Arrival_date)"

        con.ConnectionString = connectionString

        Try
```

```
            con.Open()

            Dim cmd As New SqlCommand(selectquery, con)

            cmd.ExecuteNonQuery()

            Dim dr As SqlDataReader = cmd.ExecuteReader()

            dtChartdata.Load(dr)

        Catch ex As Exception

            MsgBox(ex.ToString)

        End Try

        Return dtChartdata

    End Function

    Private Function GetKPIDataYear() As DataTable

        Dim dtChartdata As New DataTable()

        Dim con As New SqlClient.SqlConnection

        Dim connectionString As String = "Server='" &
My.Settings.mServer & "';

    Database='" & My.Settings.mDB & "';

    User Id='" & My.Settings.mUserDB & "';

    Password='" & My.Settings.mPassDB & "';"

Dim selectquery As String = "select AVG(dq.difference) as Average,
DATEPART(yyyy, dqi.Arrival_date) AS Year from Donetruckdetails
dq Inner Join DoneQueueInfotbl dqi on  dq.transno=dqi.transno
group by DATEPART(yyyy, dqi.Arrival_date)"

        con.ConnectionString = connectionString
        Try
            con.Open()
            Dim cmd As New SqlCommand(selectquery, con)
            cmd.ExecuteNonQuery()
            Dim dr As SqlDataReader = cmd.ExecuteReader()
            dtChartdata.Load(dr)
        Catch ex As Exception
            MsgBox(ex.ToString)
        End Try
        Return dtChartdata
    End Function
    Private Function GetCustomKPIDataDay() As DataTable
        Dim dtChartdata As New DataTable()
        Dim con As New SqlClient.SqlConnection
        Dim connectionString As String = "Server='" &
My.Settings.mServer & "';
    Database='" & My.Settings.mDB & "';
    User Id='" & My.Settings.mUserDB & "';
    Password='" & My.Settings.mPassDB & "';"
Dim selectquery As String = "select AVG(dq.difference) as Average,
convert(varchar(5),dqi.Arrival_date,110) AS Day from
Donetruckdetails dq Inner Join DoneQueueInfotbl dqi on
dq.transno=dqi.transno where dqi.Arrival_date between @date1
and @date2 group by  convert(varchar(5),dqi.Arrival_date,110)"
```

```
        con.ConnectionString = connectionString
        Try
            con.Open()
            Dim cmd As New SqlCommand(selectquery, con)
            cmd.Parameters.AddWithValue("@date1",
CDate(DateTimePicker10.Text).ToString("yyyy-MM-dd"))
            cmd.Parameters.AddWithValue("@date2",
CDate(DateTimePicker9.Text).ToString("yyyy-MM-dd"))
            cmd.ExecuteNonQuery()
            Dim dr As SqlDataReader = cmd.ExecuteReader()
            dtChartdata.Load(dr)
        Catch ex As Exception
            MsgBox(ex.ToString)
        End Try
        Return dtChartdata
    End Function
    Private Function CustomTruckDay() As DataTable
        Dim dtChartdata As New DataTable()
        Dim con As New SqlClient.SqlConnection
        Dim connectionString As String = "Server='" &
My.Settings.mServer & "';
    Database='" & My.Settings.mDB & "';
    User Id='" & My.Settings.mUserDB & "';
    Password='" & My.Settings.mPassDB & "';"

Dim selectquery As String = "select count(dq.transno) as total,
convert(varchar(5),dqi.Arrival_date,110) AS Day from
Donetruckdetails dq Inner Join DoneQueueInfotbl dqi on
dq.transno=dqi.transno where dqi.Arrival_date between @date1
AND @date2 group by convert(varchar(5),Arrival_date,110) "

        con.ConnectionString = connectionString

        Try

            con.Open()

            Dim cmd As New SqlCommand(selectquery, con)

cmd.Parameters.AddWithValue("@date1",
CDate(DateTimePicker1.Text).ToString("yyyy-MM-dd"))

cmd.Parameters.AddWithValue("@date2",
CDate(DateTimePicker2.Text).ToString("yyyy-MM-dd"))

            cmd.ExecuteNonQuery()

            Dim dr As SqlDataReader = cmd.ExecuteReader()

            dtChartdata.Load(dr)

        Catch ex As Exception

            MsgBox(ex.ToString)

        End Try

        Return dtChartdata

    End Function

    Private Function CustomPalletsDay() As DataTable

        Dim dtChartdata As New DataTable()
        Dim con As New SqlClient.SqlConnection

        Dim connectionString As String = "Server='" &
My.Settings.mServer & "';

    Database='" & My.Settings.mDB & "';

    User Id='" & My.Settings.mUserDB & "';

    Password='" & My.Settings.mPassDB & "';"

Dim selectquery As String = "select
sum(Cast(dq.actualtotalnoofpallet AS INT))as total,
convert(varchar(5),Arrival_date,110) AS Day from
DoneActualunloaddetails dq Inner Join DoneQueueInfotbl dqi on
dq.transno=dqi.transno where dqi.Arrival_date between @date1
AND @date2 group by convert(varchar(5),Arrival_date,110) "

        con.ConnectionString = connectionString

        Try

            con.Open()

            Dim cmd As New SqlCommand(selectquery, con)

            cmd.Parameters.AddWithValue("@date1",
CDate(DateTimePicker1.Text).ToString("yyyy-MM-dd"))

            cmd.Parameters.AddWithValue("@date2",
CDate(DateTimePicker2.Text).ToString("yyyy-MM-dd"))

            cmd.ExecuteNonQuery()

            Dim dr As SqlDataReader = cmd.ExecuteReader()

            dtChartdata.Load(dr)
        Catch ex As Exception
            MsgBox(ex.ToString)
        End Try
        Return dtChartdata
    End Function
    Private Function CustomBoxesDay() As DataTable
        Dim dtChartdata As New DataTable()
        Dim con As New SqlClient.SqlConnection
        Dim connectionString As String = "Server='" &
My.Settings.mServer & "';
    Database='" & My.Settings.mDB & "';
    User Id='" & My.Settings.mUserDB & "';
    Password='" & My.Settings.mPassDB & "';"
Dim selectquery As String = "select
sum(Cast(dq.actualtotalnoofboxes AS INT))as total,
convert(varchar(5),dqi.Arrival_date,110) AS Day from
DoneActualunloaddetails dq Inner Join DoneQueueInfotbl dqi on
dq.transno=dqi.transno where dqi.Arrival_date between @date1
AND @date2 group by convert(varchar(5),dqi.Arrival_date,110) "
        con.ConnectionString = connectionString
        Try
            con.Open()
            Dim cmd As New SqlCommand(selectquery, con)
            cmd.Parameters.AddWithValue("@date1",
CDate(DateTimePicker1.Text).ToString("yyyy-MM-dd"))

            cmd.Parameters.AddWithValue("@date2",
CDate(DateTimePicker2.Text).ToString("yyyy-MM-dd"))

            cmd.ExecuteNonQuery()

            Dim dr As SqlDataReader = cmd.ExecuteReader()

            dtChartdata.Load(dr)

        Catch ex As Exception

            MsgBox(ex.ToString)

        End Try

        Return dtChartdata

    End Function

    Private Function CustomUnitsDay() As DataTable

        Dim dtChartdata As New DataTable()

        Dim con As New SqlClient.SqlConnection

        Dim connectionString As String = "Server='" &
My.Settings.mServer & "';

    Database='" & My.Settings.mDB & "';

    User Id='" & My.Settings.mUserDB & "';

    Password='" & My.Settings.mPassDB & "';"

Dim selectquery As String = "select sum(Cast(dq.actualtotalunits
AS INT))as Total, convert(varchar(5),dqi.Arrival_date,110) AS Day
from DoneActualunloaddetails dq Inner Join DoneQueueInfotbl dqi
```

```
on  dq.transno=dqi.transno where dqi.Arrival_date between @date1
AND @date2 group by convert(varchar(5),Arrival_date,110) "

        con.ConnectionString = connectionString

        Try

            con.Open()

            Dim cmd As New SqlCommand(selectquery, con)

            cmd.Parameters.AddWithValue("@date1",
CDate(DateTimePicker1.Text).ToString("yyyy-MM-dd"))

            cmd.Parameters.AddWithValue("@date2",
CDate(DateTimePicker2.Text).ToString("yyyy-MM-dd"))

            cmd.ExecuteNonQuery()

            Dim dr As SqlDataReader = cmd.ExecuteReader()

            dtChartdata.Load(dr)

        Catch ex As Exception

            MsgBox(ex.ToString)

        End Try

        Return dtChartdata

    End Function

    Private Function CustomTruckDayVS() As DataTable

        Dim dtChartdata6 As New DataTable()

        Dim con As New SqlClient.SqlConnection

        Dim connectionString As String = "Server='" &
My.Settings.mServer & "';
```

```
    Database='" & My.Settings.mDB & "';

    User Id='" & My.Settings.mUserDB & "';

    Password='" & My.Settings.mPassDB & "';"

Dim selectquery1 As String = "select count(dq.transno) as total,
dqi.Arrival_date as Day from Donetruckdetails dq Inner Join
DoneQueueInfotbl dqi on  dq.transno=dqi.transno where
dqi.Arrival_date IN( @date1 , @date2) group by Arrival_date"

        con.ConnectionString = connectionString

        Try

            con.Open()

            Dim cmd6 As New SqlCommand(selectquery1, con)

            cmd6.Parameters.AddWithValue("@date1",
CDate(DateTimePicker1.Text).ToString("yyyy-MM-dd"))

            cmd6.Parameters.AddWithValue("@date2",
CDate(DateTimePicker2.Text).ToString("yyyy-MM-dd"))

            cmd6.ExecuteNonQuery()

            Dim dr6 As SqlDataReader = cmd6.ExecuteReader()

            dtChartdata6.Load(dr6)

        Catch ex As Exception

            MsgBox(ex.ToString)

        End Try

        Return dtChartdata6

    End Function

    Private Function CustomPalletsDayVS() As DataTable
```

```
        Dim dtChartdata2 As New DataTable()

        Dim con As New SqlClient.SqlConnection

        Dim connectionString As String = "Server='" &
My.Settings.mServer & "';

    Database='" & My.Settings.mDB & "';

    User Id='" & My.Settings.mUserDB & "';

    Password='" & My.Settings.mPassDB & "';"

Dim selectquery2 As String = "select
sum(Cast(dq.actualtotalnoofpallet AS INT))as total, dqi.Arrival_date
AS Day from DoneActualunloaddetails dq Inner Join
DoneQueueInfotbl dqi on  dq.transno=dqi.transno where
dqi.Arrival_date IN( @date1 , @date2) group by dqi.Arrival_date"

        con.ConnectionString = connectionString

        Try

            con.Open()

            Dim cmd2 As New SqlCommand(selectquery2, con)

            cmd2.Parameters.AddWithValue("@date1",
CDate(DateTimePicker1.Text).ToString("yyyy-MM-dd"))

            cmd2.Parameters.AddWithValue("@date2",
CDate(DateTimePicker2.Text).ToString("yyyy-MM-dd"))

            cmd2.ExecuteNonQuery()

            Dim dr2 As SqlDataReader = cmd2.ExecuteReader()

            dtChartdata2.Load(dr2)
        Catch ex As Exception

            MsgBox(ex.ToString)

        End Try

        Return dtChartdata2

    End Function

    Private Function CustomBoxesDayVS() As DataTable

        Dim dtChartdata3 As New DataTable()

        Dim con As New SqlClient.SqlConnection

        Dim connectionString As String = "Server='" &
My.Settings.mServer & "';

    Database='" & My.Settings.mDB & "';

    User Id='" & My.Settings.mUserDB & "';

    Password='" & My.Settings.mPassDB & "';"

Dim selectquery3 As String = "select
sum(Cast(dq.actualtotalnoofboxes AS INT))as total, dqi.Arrival_date
AS Day from DoneActualunloaddetails dq Inner Join
DoneQueueInfotbl dqi on  dq.transno=dqi.transno where
dqi.Arrival_date IN( @date1 , @date2) group by dqi.Arrival_date"

        con.ConnectionString = connectionString

        Try

            con.Open()

            Dim cmd3 As New SqlCommand(selectquery3, con)

cmd3.Parameters.AddWithValue("@date1",
CDate(DateTimePicker1.Text).ToString("yyyy-MM-dd"))
```

```
cmd3.Parameters.AddWithValue("@date2",
CDate(DateTimePicker2.Text).ToString("yyyy-MM-dd"))

cmd3.ExecuteNonQuery()

            Dim dr3 As SqlDataReader = cmd3.ExecuteReader()

            dtChartdata3.Load(dr3)

        Catch ex As Exception

            MsgBox(ex.ToString)

        End Try

        Return dtChartdata3

    End Function

    Private Function CustomUnitsDayVS() As DataTable

        Dim dtChartdata4 As New DataTable()

        Dim con As New SqlClient.SqlConnection

        Dim connectionString As String = "Server='" &
My.Settings.mServer & "';

    Database='" & My.Settings.mDB & "';

    User Id='" & My.Settings.mUserDB & "';

    Password='" & My.Settings.mPassDB & "';"

Dim selectquery4 As String = "select sum(Cast(dq.actualtotalunits
AS INT))as total, dqi.Arrival_date AS Day from
DoneActualunloaddetails dq Inner Join DoneQueueInfotbl dqi on
dq.transno=dqi.transno where dqi.Arrival_date IN( @date1 ,
@date2) group by dqi.Arrival_date"
```

```
            con.ConnectionString = connectionString

            Try

                con.Open()

                Dim cmd4 As New SqlCommand(selectquery4, con)

                cmd4.Parameters.AddWithValue("@date1",
CDate(DateTimePicker1.Text).ToString("yyyy-MM-dd"))

                cmd4.Parameters.AddWithValue("@date2",
CDate(DateTimePicker2.Text).ToString("yyyy-MM-dd"))

                cmd4.ExecuteNonQuery()

                Dim dr4 As SqlDataReader = cmd4.ExecuteReader()

                dtChartdata4.Load(dr4)

            Catch ex As Exception

                MsgBox(ex.ToString)

            End Try

            Return dtChartdata4

        End Function

        Private Sub Dashboard_Load(sender As Object, e As EventArgs)
Handles MyBase.Load

            DataGridView1.AllowUserToAddRows = False

            DataGridView4.AllowUserToAddRows = False

            DataGridView5.AllowUserToAddRows = False

            DataGridView2.AllowUserToAddRows = False

            DataGridView3.AllowUserToAddRows = False
            DataGridView6.AllowUserToAddRows = False

            DataGridView6.GridColor = Color.Red

            DataGridView5.GridColor = Color.Red

            DataGridView2.GridColor = Color.Red

            DataGridView3.GridColor = Color.Red

            DataGridView6.RowHeadersVisible = False

            DataGridView5.RowHeadersVisible = False

            DataGridView2.RowHeadersVisible = False

            Me.FormBorderStyle = FormBorderStyle.None

            RadioButton2.Checked = True

            RadioButton3.Checked = True

            RadioButton5.Checked = True

            DataGridView1.RowHeadersVisible = False

            DataGridView3.RowHeadersVisible = False

            DataGridView4.RowHeadersVisible = False

            Dim myname As New DBconnection

            Name = myname.ReceiverName

            Label24.Text = Name

        End Sub

        Private Sub Button1_Click(sender As Object, e As EventArgs)
Handles Button1.Click

            LoadArrived()
            If RadioButton1.Checked = True Then
                TabControl9.TabPages(3).Enabled = True
                TabControl9.SelectedIndex = 3
                LoadCustomData()
                Timer1.Enabled = False
                Timer2.Enabled = False
                Timer3.Enabled = True
            ElseIf RadioButton2.Checked = True Then
                TabControl9.TabPages(3).Enabled = True
                TabControl9.SelectedIndex = 3
                LoadCustomDataVS()
                Timer1.Enabled = False
                Timer3.Enabled = False
                Timer2.Enabled = True
            End If
        End Sub
        Private Sub Timer1_Tick(sender As Object, e As EventArgs) Handles Timer1.Tick
            Label22.Text = DateTime.Now.ToString("MMMM dd, yyyy h:mm:ss tt")
            If TabControl9.SelectedIndex = 0 Then
                LoadDailyData()
            ElseIf TabControl9.SelectedIndex = 1 Then
                LoadMonthlyData()
            ElseIf TabControl9.SelectedIndex = 2 Then
                LoadYearlyData()
            End If
        End Sub
        Private Sub Logout_Click(sender As Object, e As EventArgs)
Handles Logout.Click
            Dim dialog As DialogResult
            dialog = MessageBox.Show("Are you sure you want to
Logout?", "Exit", MessageBoxButtons.YesNo)
            If dialog = DialogResult.No Then
            Else
                Dim logout As New DBconnection
                logout.ReceiverName = ""
                Me.Close()
                Login.Show()
            End If
        End Sub
        Public Sub queuetable()
            Dim con As New SqlClient.SqlConnection
            Dim connectionString As String = "Server='" &
My.Settings.mServer & "';
```

```
   Database='" & My.Settings.mDB & "';

   User Id='" & My.Settings.mUserDB & "';

   Password='" & My.Settings.mPassDB & "';"

      con.ConnectionString = connectionString

Dim selectquery As String = "select DoneQueueInfotbl.Arrival_date
as 'Date', SUBSTRING(CONVERT(VARCHAR,
DoneQueueInfotbl.Arrival_time,120), 1, 17) as 'Arrival Time',
SUBSTRING(CONVERT(VARCHAR,
Donetruckdetails.startunload,120), 1, 17) as 'Unload Start',
SUBSTRING(CONVERT(VARCHAR,
Donetruckdetails.finishedunload,120), 1, 17) as 'Unload End',
SUBSTRING(CONVERT(VARCHAR,
Donetruckdetails.startreceive,120), 1, 17) as 'Receive
Start',SUBSTRING(CONVERT(VARCHAR,
Donetruckdetails.finishedreceive,120), 1, 17) as 'Receive
End',Donetruckdetails.difference as
'Difference',Donetruckdetails.vendorname as
'Vendor',Donetruckdetails.truck_type as 'Truck',
Donetruckdetails.truck_load as 'Load',Donetruckdetails.plate_no as
'Plate No.',Donetruckdetails.no_of_helper as 'No. of
Helper',DoneActualunloaddetails.actualtotalnoofpallet as 'Total
Pallets',Donetruckdetails.countofdr as 'Total
DR',DoneActualunloaddetails.actualtotalnoofboxes as 'Total
Boxes',DoneActualunloaddetails.actualtotalunits as 'Total
Units',Donetruckdetails.category as 'Category' from
DoneQueueInfotbl inner join Donetruckdetails on
DoneQueueInfotbl.transno = Donetruckdetails.transno inner join
DoneActualunloaddetails on
DoneActualunloaddetails.transno=Donetruckdetails.transno inner
join DoneQueuetbl on DoneQueuetbl.transno =
Donetruckdetails.transno where DoneQueueInfotbl.Arrival_date
between @date1 And @date2"
```

```
        Try

            con.Open()

            cmd5 = New SqlCommand(selectquery, con)

            cmd5.Parameters.AddWithValue("@date1",
CDate(DateTimePicker4.Text).ToString("yyyy-MM-dd"))

            cmd5.Parameters.AddWithValue("@date2",
CDate(DateTimePicker3.Text).ToString("yyyy-MM-dd"))

            Dim da As New SqlDataAdapter(cmd5)

            Dim dt As New DataTable()

            da.Fill(dt)

            dbs.DataSource = dt

            DataGridView1.DataSource = dbs

            DataGridView1.ReadOnly = True

        Catch ex As Exception

            MessageBox.Show(ex.Message.ToString(), "Error
Message")

        End Try

    End Sub

    Public Sub queuetable1()

        Dim con As New SqlClient.SqlConnection

        Dim connectionString As String = "Server='" &
My.Settings.mServer & "';

    Database='" & My.Settings.mDB & "';
    User Id='" & My.Settings.mUserDB & "';

    Password='" & My.Settings.mPassDB & "';"

        con.ConnectionString = connectionString

Dim selectquery As String = "select DoneQueueInfotbl.Arrival_date
as Date, DoneQueueInfotbl.Arrival_time as
Time,Donetruckdetails.vendorname as 'Vendor
Name',Donetruckdetails.truck_type as 'Truck Type',
Donetruckdetails.truck_load as 'Truck
Load',Donetruckdetails.plate_no as 'Plate
No.',Donetruckdetails.no_of_helper as 'No. of
Helper',DoneActualunloaddetails.actualtotalnoofpallet as 'No. of
Pallet',DoneQueuetbl.ReceiverName as 'Receiver
Name',DoneActualunloaddetails.actualtotalnoofboxes as 'No. of
Boxes',DoneActualunloaddetails.actualtotalunits as 'No of
Units',Donetruckdetails.category as 'Category' from
DoneQueueInfotbl inner join Donetruckdetails on
DoneQueueInfotbl.transno = Donetruckdetails.transno inner join
DoneActualunloaddetails on
DoneActualunloaddetails.transno=Donetruckdetails.transno inner
join DoneQueuetbl on DoneQueuetbl.transno =
Donetruckdetails.transno where DoneQueueInfotbl.Arrival_date
between @date1 And @date2"

Dim selectquery1 As String = "select
DoneQueueInfotbl.Arrival_date as Date,
DoneQueueInfotbl.Arrival_time as
Time,Donetruckdetails.vendorname as 'Vendor
Name',Donetruckdetails.truck_type as 'Truck Type',
Donetruckdetails.truck_load as 'Truck
Load',Donetruckdetails.plate_no as 'Plate
No.',Donetruckdetails.no_of_helper as 'No. of
Helper',DoneActualunloaddetails.actualtotalnoofpallet as 'No. of
Pallet',DoneQueuetbl.ReceiverName as 'Receiver
```

```
Name',DoneActualunloaddetails.actualtotalnoofboxes as 'No. of
Boxes',DoneActualunloaddetails.actualtotalunits as 'No of
Units',Donetruckdetails.category as 'Category' from
DoneQueueInfotbl inner join Donetruckdetails on
DoneQueueInfotbl.transno = Donetruckdetails.transno inner join
DoneActualunloaddetails on
DoneActualunloaddetails.transno=Donetruckdetails.transno inner
join DoneQueuetbl on DoneQueuetbl.transno =
Donetruckdetails.transno where DoneQueueInfotbl.Arrival_date IN
(@date1, @date2)"

        Try

            con.Open()

            If RadioButton3.Checked = True Then

                cmd5 = New SqlCommand(selectquery1, con)

            Else

                cmd5 = New SqlCommand(selectquery, con)

            End If

            cmd5.Parameters.AddWithValue("@date1",
CDate(DateTimePicker6.Text).ToString("yyyy-MM-dd"))

            cmd5.Parameters.AddWithValue("@date2",
CDate(DateTimePicker5.Text).ToString("yyyy-MM-dd"))

            Dim da As New SqlDataAdapter(cmd5)

            Dim dt As New DataTable()

            da.Fill(dt)

            dbs1.DataSource = dt

            DataGridView2.DataSource = dbs1
            DataGridView2.ReadOnly = True
        Catch ex As Exception
            MessageBox.Show(ex.Message.ToString(), "Error
Message")
        End Try
    End Sub
    Public Sub queuetable2()
        Dim con As New SqlClient.SqlConnection
        Dim connectionString As String = "Server='" &
My.Settings.mServer & "';
    Database='" & My.Settings.mDB & "';
    User Id='" & My.Settings.mUserDB & "';
    Password='" & My.Settings.mPassDB & "';"
        con.ConnectionString = connectionString
Dim selectquery As String = "select DoneQueueInfotbl.Arrival_date
as Date, DoneQueueInfotbl.Arrival_time as Time,
Donetruckdetails.vendorname as 'Vendor
Name',Donetruckdetails.truck_type as 'Truck Type',
Donetruckdetails.truck_load as 'Truck
Load',Donetruckdetails.plate_no as 'Plate
No.',Donetruckdetails.no_of_helper as 'No. of
Helper',DoneActualunloaddetails.actualtotalnoofpallet as 'No. of
Pallet',DoneQueuetbl.ReceiverName as 'Receiver
Name',DoneActualunloaddetails.actualtotalnoofboxes as 'No. of
Boxes',DoneActualunloaddetails.actualtotalunits as 'No of
Units',Donetruckdetails.category as 'Category' from
DoneQueueInfotbl inner join Donetruckdetails on
```

```
DoneQueueInfotbl.transno = Donetruckdetails.transno inner join
DoneActualunloaddetails on
DoneActualunloaddetails.transno=Donetruckdetails.transno inner
join DoneQueuetbl on DoneQueuetbl.transno =
Donetruckdetails.transno where Donetruckdetails.vendorname=
@vendor and DoneQueueInfotbl.Arrival_date between @date1 And
@date2"

        Try

            con.Open()

            cmd5 = New SqlCommand(selectquery, con)

            cmd5.Parameters.AddWithValue("@date1",
CDate(DateTimePicker8.Text).ToString("yyyy-MM-dd"))

            cmd5.Parameters.AddWithValue("@date2",
CDate(DateTimePicker7.Text).ToString("yyyy-MM-dd"))

            cmd5.Parameters.AddWithValue("@vendor",
TextBox11.Text)

            Dim da As New SqlDataAdapter(cmd5)

            Dim dt As New DataTable()

            da.Fill(dt)

            dbs2.DataSource = dt

            DataGridView3.DataSource = dbs2

            DataGridView3.ReadOnly = True

        Catch ex As Exception

            MessageBox.Show(ex.Message.ToString(), "Error
Message")

        End Try
```

```
    End Sub

    Public Sub queuetable3()

        Dim con As New SqlClient.SqlConnection

        Dim connectionString As String = "Server='" &
My.Settings.mServer & "';

    Database='" & My.Settings.mDB & "';

    User Id='" & My.Settings.mUserDB & "';

    Password='" & My.Settings.mPassDB & "';"

        con.ConnectionString = connectionString

Dim selectquery As String = "select DoneQueueInfotbl.transno as
'Trans No.',DoneQueueInfotbl.Arrival_date as 'Date',
SUBSTRING(CONVERT(VARCHAR,
Donetruckdetails.startunload,120), 1, 16)as 'Unload Start',
SUBSTRING(CONVERT(VARCHAR,
Donetruckdetails.finishedunload,120), 1, 16)as 'Unload End',
Donetruckdetails.difference As 'Actual Unload Time',

CASE WHEN truck_type = '4W'  THEN '45'

WHEN truck_type = '6W' THEN '90'

WHEN truck_type = '10W' THEN '120'

WHEN truck_type = '40FTR' THEN '240'

WHEN truck_type = 'WINGVAN' THEN '120'

WHEN truck_type = 'L300' THEN '45'

WHEN truck_type = '20FTR' THEN '120'

WHEN truck_type = 'APMI' THEN '30'
WHEN truck_type = '6WFWD' THEN '90'

WHEN truck_type = 'ON FOOT' THEN '30'

END As Target,(Donetruckdetails.difference-CASE WHEN
truck_type = '4W'  THEN '45'

WHEN truck_type = '6W' THEN '90'

WHEN truck_type = '10W' THEN '120'

WHEN truck_type = '40FTR' THEN '240'

WHEN truck_type = 'WINGVAN' THEN '120'

WHEN truck_type = 'L300' THEN '45'

WHEN truck_type = '20FTR' THEN '120'

WHEN truck_type = 'APMI' THEN '30'

WHEN truck_type = '6WFWD' THEN '90'

WHEN truck_type = 'ON FOOT' THEN '30'

END ) as 'Interval Time' ,

Case WHEN Donetruckdetails.difference<=CASE WHEN
truck_type = '4W'  THEN '45'

WHEN truck_type = '6W' THEN '90'

WHEN truck_type = '10W' THEN '120'

WHEN truck_type = '40FTR' THEN '240'

WHEN truck_type = 'WINGVAN' THEN '120'

WHEN truck_type = 'L300' THEN '45'

WHEN truck_type = '20FTR' THEN '120'
WHEN truck_type = 'APMI' THEN '30'

WHEN truck_type = '6WFWD' THEN '90'

WHEN truck_type = 'ON FOOT' THEN '30'

END then Donetruckdetails.difference Else CASE WHEN truck_type = '4W'  THEN '45'

WHEN truck_type = '6W' THEN '90'

WHEN truck_type = '10W' THEN '120'

WHEN truck_type = '40FTR' THEN '240'

WHEN truck_type = 'WINGVAN' THEN '120'

WHEN truck_type = 'L300' THEN '45'

WHEN truck_type = '20FTR' THEN '120'

WHEN truck_type = 'APMI' THEN '30'

WHEN truck_type = '6WFWD' THEN '90'

WHEN truck_type = 'ON FOOT' THEN '30'

END End as 'Within',

CASE WHEN Donetruckdetails.difference <=CASE WHEN truck_type = '4W'  THEN '45'

WHEN truck_type = '6W' THEN '90'

WHEN truck_type = '10W' THEN '120'

WHEN truck_type = '40FTR' THEN '240'

WHEN truck_type = 'WINGVAN' THEN '120'

WHEN truck_type = 'L300' THEN '45'
WHEN truck_type = '20FTR' THEN '120'
WHEN truck_type = 'APMI' THEN '30'
WHEN truck_type = '6WFWD' THEN '90'
WHEN truck_type = 'ON FOOT' THEN '30'
END then 0 Else Donetruckdetails.difference-CASE WHEN
truck_type = '4W'  THEN '45'
WHEN truck_type = '6W' THEN '90'
WHEN truck_type = '10W' THEN '120'
WHEN truck_type = '40FTR' THEN '240'
WHEN truck_type = 'WINGVAN' THEN '120'
WHEN truck_type = 'L300' THEN '45'
WHEN truck_type = '20FTR' THEN '120'
WHEN truck_type = 'APMI' THEN '30'
WHEN truck_type = '6WFWD' THEN '90'
WHEN truck_type = 'ON FOOT' THEN '30'
END End As Excess

,
Case when Donetruckdetails.difference <= CASE WHEN truck_type
= '4W'  THEN '45'
WHEN truck_type = '6W' THEN '90'
WHEN truck_type = '10W' THEN '120'
WHEN truck_type = '40FTR' THEN '240'
WHEN truck_type = 'WINGVAN' THEN '120'

WHEN truck_type = 'L300' THEN '45'

WHEN truck_type = '20FTR' THEN '120'

WHEN truck_type = 'APMI' THEN '30'

WHEN truck_type = '6WFWD' THEN '90'

WHEN truck_type = 'ON FOOT' THEN '30'

END Then 'PASSED' Else 'FAILED' END AS 'Grade',

Donetruckdetails.vendorname As
'Vendor',Donetruckdetails.truck_type As Truck,
Donetruckdetails.truck_load as Load,Donetruckdetails.no_of_helper
as 'No. of Helper',DoneActualunloaddetails.actualtotalnoofpallet as
'Total Pallet',DoneQueuetbl.ReceiverName as 'Receiver
Name',Donetruckdetails.countofdr as 'No of
DR',DoneActualunloaddetails.actualtotalnoofboxes as 'Total
Boxes',DoneActualunloaddetails.actualtotalunits as 'Total
Units',Donetruckdetails.category as 'Category' from
DoneQueueInfotbl inner join Donetruckdetails on
DoneQueueInfotbl.transno = Donetruckdetails.transno inner join
DoneActualunloaddetails on
DoneActualunloaddetails.transno=Donetruckdetails.transno inner
join DoneQueuetbl on DoneQueuetbl.transno =
Donetruckdetails.transno where DoneQueueInfotbl.Arrival_date
between @date1 And @date2 group by
DoneQueueInfotbl.Arrival_date,DoneQueueInfotbl.transno,DoneQu
eueInfotbl.Arrival_time,
Donetruckdetails.startunload,Donetruckdetails.finishedunload,
Donetruckdetails.startreceive,Donetruckdetails.finishedreceive,Don
etruckdetails.difference,Donetruckdetails.vendorname,Donetruckdet
ails.truck_type,
Donetruckdetails.truck_load,Donetruckdetails.plate_no,Donetruckd
etails.no_of_helper,DoneActualunloaddetails.actualtotalnoofpallet,D
```

```
oneQueuetbl.ReceiverName,Donetruckdetails.countofdr,DoneActualunloaddetails.actualtotalnoofboxes,DoneActualunloaddetails.actualtotalunits,Donetruckdetails.category"

        Try

            con.Open()

            cmd5 = New SqlCommand(selectquery, con)

            cmd5.Parameters.AddWithValue("@date1", CDate(DateTimePicker10.Text).ToString("yyyy-MM-dd"))

            cmd5.Parameters.AddWithValue("@date2", CDate(DateTimePicker9.Text).ToString("yyyy-MM-dd"))

            Dim da As New SqlDataAdapter(cmd5)

            Dim dt As New DataTable()

            da.Fill(dt)

            dbs3.DataSource = dt

            DataGridView4.DataSource = dbs3

            DataGridView4.ReadOnly = True

        Catch ex As Exception

            MessageBox.Show(ex.Message.ToString(), "Error Message")

        End Try

    End Sub

    Public Sub queuetable4()

        Dim con As New SqlClient.SqlConnection
        Dim connectionString As String = "Server='" &
My.Settings.mServer & '";

    Database='" & My.Settings.mDB & '";

    User Id='" & My.Settings.mUserDB & '";

    Password='" & My.Settings.mPassDB & '";"

        con.ConnectionString = connectionString

Dim selectquery As String = "select DoneQueueInfotbl.transno as
'Trans No.',DoneQueueInfotbl.Arrival_date as 'Date',
SUBSTRING(CONVERT(VARCHAR,
Donetruckdetails.startunload,120), 1, 16)as 'Unload Start',

SUBSTRING(CONVERT(VARCHAR,
Donetruckdetails.finishedunload,120), 1, 16)as 'Unload End',
Unload.UnloadDiff As 'Actual Unload Time',

              Unload.UnloadTarget as
Target,AVG(CAST(Unload.UnloadDiff as float)-
CAST(Unload.UnloadTarget as float)) as 'Time
Interval',Unload.UnloadWithin as Within,Unload.UnloadExcess as
Excess,Unload.UnloadGrade as 'Unload Grade',

        Donetruckdetails.vendorname As
'Vendor',Donetruckdetails.truck_type As Truck,
Donetruckdetails.truck_load as Load,Donetruckdetails.no_of_helper
as 'No. of Helper',DoneActualunloaddetails.actualtotalnoofpallet as
'Total Pallet',DoneQueuetbl.ReceiverName as 'Receiver
Name',Donetruckdetails.countofdr as 'No of
DR',DoneActualunloaddetails.actualtotalnoofboxes as 'Total
Boxes',DoneActualunloaddetails.actualtotalunits as 'Total
Units',Donetruckdetails.category as 'Category' from
DoneQueueInfotbl inner join Donetruckdetails on
DoneQueueInfotbl.transno = Donetruckdetails.transno inner join
DoneActualunloaddetails on
```

```
DoneActualunloaddetails.transno=Donetruckdetails.transno inner
join DoneQueuetbl on DoneQueuetbl.transno =
Donetruckdetails.transno inner join Unload on
Unload.transno=DoneQueuetbl.transno where
DoneQueueInfotbl.Arrival_date=@date1 group by
DoneQueueInfotbl.Arrival_date,DoneQueueInfotbl.transno,DoneQu
eueInfotbl.Arrival_time,
Donetruckdetails.startunload,Donetruckdetails.finishedunload,
Donetruckdetails.startreceive,Donetruckdetails.finishedreceive,Don
etruckdetails.vendorname,Donetruckdetails.truck_type,
Donetruckdetails.truck_load,Donetruckdetails.plate_no,Donetruckd
etails.no_of_helper,DoneActualunloaddetails.actualtotalnoofpallet,D
oneQueuetbl.ReceiverName,Donetruckdetails.countofdr,DoneActua
lunloaddetails.actualtotalnoofboxes,DoneActualunloaddetails.actual
totalunits,Donetruckdetails.category,Unload.UnloadDiff,Unload.Unlo
adTarget,Unload.UnloadWithin,Unload.UnloadExcess,Unload.Unlo
adGrade"

    Try

        con.Open()

        cmd5 = New SqlCommand(selectquery, con)

        cmd5.Parameters.AddWithValue("@date1",
CDate(DateTimePicker16.Text).ToString("yyyy-MM-dd"))

        Dim da As New SqlDataAdapter(cmd5)

        Dim dt As New DataTable()

        da.Fill(dt)

        dbs5.DataSource = dt

        DataGridView6.DataSource = dbs5

        DataGridView6.ReadOnly = True
    Catch ex As Exception

        MessageBox.Show(ex.Message.ToString(), "Error Message")

    End Try

End Sub

Private Sub Button2_Click(sender As Object, e As EventArgs) Handles Button2.Click

    queuetable()

End Sub

Private Sub Button3_Click(sender As Object, e As EventArgs) Handles Button3.Click

    queuetable1()

    If RadioButton4.Checked = True Then

        TabControl1.TabPages(3).Enabled = True

        TabControl1.SelectedIndex = 3

        Timer5.Enabled = True

        Timer1.Enabled = False

        Timer2.Enabled = False

        Timer3.Enabled = False

        Timer4.Enabled = False

        Timer6.Enabled = False

    ElseIf RadioButton3.Checked = True Then

        TabControl1.TabPages(3).Enabled = True
        TabControl1.SelectedIndex = 3

        Timer5.Enabled = False

        Timer1.Enabled = False

        Timer2.Enabled = False

        Timer3.Enabled = False

        Timer4.Enabled = False

        Timer6.Enabled = True

    End If

End Sub

Private Sub Button4_Click(sender As Object, e As EventArgs)
Handles Button4.Click

    queuetable2()

    If RadioButton6.Checked = True Then

        TabControl2.TabPages(3).Enabled = True

        TabControl2.SelectedIndex = 3

        Timer1.Enabled = False

        Timer2.Enabled = False

        Timer3.Enabled = False

        Timer4.Enabled = False

        Timer5.Enabled = False

        Timer6.Enabled = False

        Timer7.Enabled = False
        Timer8.Enabled = True

        Timer9.Enabled = False

    ElseIf RadioButton5.Checked = True Then

        TabControl2.TabPages(3).Enabled = True

        TabControl2.SelectedIndex = 3

        Timer1.Enabled = False

        Timer2.Enabled = False

        Timer3.Enabled = False

        Timer4.Enabled = False

        Timer5.Enabled = False

        Timer6.Enabled = False

        Timer7.Enabled = False

        Timer8.Enabled = False

        Timer9.Enabled = True

    End If

End Sub

Private Sub Button5_Click(sender As Object, e As EventArgs)
Handles Button5.Click

    queuetable3()

    LoadCustomKPIData()

    TabControl8.TabPages(2).Enabled = True

    TabControl8.SelectedIndex = 2
End Sub
Private Sub Button6_Click(sender As Object, e As EventArgs)
End Sub
Private Sub Timer2_Tick(sender As Object, e As EventArgs)
Handles Timer2.Tick
    LoadCustomDataVS()
    LoadArrived()
End Sub
Private Sub Timer3_Tick(sender As Object, e As EventArgs)
Handles Timer3.Tick
    LoadCustomData()
    LoadArrived()
End Sub
Private Sub TabControl9_SelectedIndexChanged(sender As
Object, e As EventArgs) Handles
TabControl9.SelectedIndexChanged
    Select Case TabControl9.SelectedIndex
        Case 0
            Timer1.Enabled = True
            Timer2.Enabled = False
            Timer3.Enabled = False
            Timer4.Enabled = False
            Timer5.Enabled = False
            Timer6.Enabled = False

            Timer7.Enabled = False

            Timer8.Enabled = False

            Timer9.Enabled = False

        Case 1

            Timer1.Enabled = True

            Timer2.Enabled = False

            Timer3.Enabled = False

            Timer4.Enabled = False

            Timer5.Enabled = False

            Timer6.Enabled = False

            Timer7.Enabled = False

            Timer8.Enabled = False

            Timer9.Enabled = False

        Case 2

            Timer1.Enabled = True

            Timer2.Enabled = False

            Timer3.Enabled = False

            Timer4.Enabled = False

            Timer5.Enabled = False

            Timer6.Enabled = False

            Timer7.Enabled = False
```

```
                Timer8.Enabled = False

                Timer9.Enabled = False

            Case 3

                Timer1.Enabled = False

                Timer4.Enabled = False

                Timer5.Enabled = False

                Timer6.Enabled = False

                Timer7.Enabled = False

                Timer8.Enabled = False

                Timer9.Enabled = False

                Button1.PerformClick()

        End Select

    End Sub

    Private Sub TabControl4_SelectedIndexChanged(sender As
Object, e As EventArgs) Handles
TabControl4.SelectedIndexChanged

        Select Case TabControl4.SelectedIndex

            Case 0

                Timer1.Enabled = True

                Timer2.Enabled = False

                Timer3.Enabled = False

                Timer4.Enabled = False
    Timer5.Enabled = False
    Timer6.Enabled = False
    Timer7.Enabled = False
    Timer8.Enabled = False
    Timer9.Enabled = False
Case 1
    Timer4.Enabled = True
    Timer1.Enabled = False
    Timer2.Enabled = False
    Timer3.Enabled = False
    Timer5.Enabled = False
    Timer6.Enabled = False
    Timer7.Enabled = False
    Timer8.Enabled = False
    Timer9.Enabled = False
Case 2
    Timer4.Enabled = False
    Timer1.Enabled = False
    Timer2.Enabled = False
    Timer3.Enabled = False
    Timer5.Enabled = False
    Timer6.Enabled = False
```

```
                Timer7.Enabled = True
                Timer8.Enabled = False
                Timer9.Enabled = False
            Case 3
                LoadData()
            Case 6
                LoadDailyQueueingKPI()
            Case 7
                LoadDailyQueueingWeightKPI()
        End Select
    End Sub
    Private Sub Timer4_Tick(sender As Object, e As EventArgs) Handles Timer4.Tick
        LoadCategory()
    End Sub
    Private Sub TabControl1_SelectedIndexChanged(sender As Object, e As EventArgs) Handles TabControl1.SelectedIndexChanged
        Select Case TabControl1.SelectedIndex
            Case 0
                Timer4.Enabled = True
                Timer1.Enabled = False
                Timer2.Enabled = False
    Timer3.Enabled = False
    Timer5.Enabled = False
    Timer6.Enabled = False
    Timer7.Enabled = False
    Timer8.Enabled = False
    Timer9.Enabled = False
Case 1
    Timer4.Enabled = True
    Timer1.Enabled = False
    Timer2.Enabled = False
    Timer3.Enabled = False
    Timer5.Enabled = False
    Timer6.Enabled = False
    Timer7.Enabled = False
    Timer8.Enabled = False
    Timer9.Enabled = False
Case 2
    Timer4.Enabled = True
    Timer1.Enabled = False
    Timer2.Enabled = False
    Timer3.Enabled = False
    Timer5.Enabled = False
```

```
            Timer6.Enabled = False

            Timer7.Enabled = False

            Timer8.Enabled = False

            Timer9.Enabled = False

         Case 3

            Timer4.Enabled = False

            Timer1.Enabled = False

            Timer2.Enabled = False

            Timer3.Enabled = False

            Timer7.Enabled = False

            Timer8.Enabled = False

            Timer9.Enabled = False

            Button3.PerformClick()

      End Select

   End Sub

   Private Sub TabControl2_SelectedIndexChanged(sender As Object, e As EventArgs) Handles TabControl2.SelectedIndexChanged

      Select Case TabControl2.SelectedIndex

         Case 0

            Timer4.Enabled = False

            Timer1.Enabled = False
```

```
    Timer2.Enabled = False

    Timer3.Enabled = False

    Timer5.Enabled = False

    Timer6.Enabled = False

    Timer7.Enabled = True

    Timer8.Enabled = False

    Timer9.Enabled = False

Case 1

    Timer4.Enabled = False

    Timer1.Enabled = False

    Timer2.Enabled = False

    Timer3.Enabled = False

    Timer5.Enabled = False

    Timer6.Enabled = False

    Timer7.Enabled = True

    Timer8.Enabled = False

    Timer9.Enabled = False

Case 2

    Timer4.Enabled = False

    Timer1.Enabled = False

    Timer2.Enabled = False

    Timer3.Enabled = False
```

```
                Timer5.Enabled = False

                Timer6.Enabled = False

                Timer7.Enabled = True

                Timer8.Enabled = False

                Timer9.Enabled = False

            Case 3

                Timer4.Enabled = False

                Timer1.Enabled = False

                Timer2.Enabled = False

                Timer3.Enabled = False

                Timer5.Enabled = False

                Timer6.Enabled = False

                Timer7.Enabled = False

                Button4.PerformClick()

        End Select

    End Sub

    Private Sub Timer5_Tick(sender As Object, e As EventArgs) Handles Timer5.Tick

        LoadCustomCategory()

    End Sub
    Private Sub Timer6_Tick(sender As Object, e As EventArgs)
Handles Timer6.Tick

        LoadCustomCategoryVS()

    End Sub

    Private Sub Timer7_Tick(sender As Object, e As EventArgs)
Handles Timer7.Tick

        LoadSupplier()

    End Sub

    Private Sub Timer8_Tick(sender As Object, e As EventArgs)
Handles Timer8.Tick

        LoadCustomSupplier()

    End Sub

    Private Sub Timer9_Tick(sender As Object, e As EventArgs)
Handles Timer9.Tick

        LoadCustomSupplierVS()

    End Sub

    Private Sub Button7_Click(sender As Object, e As EventArgs)
Handles Button7.Click

        Queuechecktbl()

        LoadCustomDailyQueueingKPI()

        TabControl3.TabPages(1).Enabled = True

        TabControl3.SelectedIndex = 1

    End Sub
```

```
   Private Sub DataGridView5_CellMouseClick(sender As Object, e
As DataGridViewCellMouseEventArgs) Handles
DataGridView5.CellMouseClick

        Dim con As New SqlClient.SqlConnection

        Dim connectionString As String = "Server='" &
My.Settings.mServer & "';

        Database='" & My.Settings.mDB & "';

        User Id='" & My.Settings.mUserDB & "';

        Password='" & My.Settings.mPassDB & "';"

        con.ConnectionString = connectionString

        i = DataGridView5.CurrentRow.Index

        transno = DataGridView5.Item(1, i).Value.ToString

        Dim selectquery As String

selectquery = "SELECT
SUBSTRING(CONVERT(VARCHAR,CAST(Arrival_date AS
DATETIME)+CAST(Arrival_time AS DATETIME),120), 1, 19) as
ArrivalTime,SUBSTRING(CONVERT(VARCHAR,CAST(Arrival_dat
e AS DATETIME)+CAST(EndEncoded AS DATETIME),120), 1, 19)
as
EndEncode,SUBSTRING(CONVERT(VARCHAR,CAST(Arrival_dat
e AS DATETIME)+CAST(StartVerify AS DATETIME),120), 1, 19)
as StartV,SUBSTRING(CONVERT(VARCHAR,CAST(Arrival_date
AS DATETIME)+CAST(EndVerify AS DATETIME),120), 1, 19) as
EndV,SUBSTRING(CONVERT(VARCHAR, startreceive,120), 1,
19) as StartR,SUBSTRING(CONVERT(VARCHAR, docktime,120),
1, 19) as DT,SUBSTRING(CONVERT(VARCHAR,
startunload,120), 1, 19) as
SUnload,SUBSTRING(CONVERT(VARCHAR, finishedunload,120),
1, 19) as FUnload,SUBSTRING(CONVERT(VARCHAR,
finishedreceive,120), 1, 19) as FReceive, * from DoneQueueInfotbl
dq inner join Encode E on dq.transno=E.transno

inner join GV G on dq.transno=G.transno inner join Verify V on
dq.transno=V.transno inner join

Donetruckdetails dqi on dq.transno=dqi.transno inner join VR on
dq.transno=VR.transno inner join

RD on dq.transno=RD.transno inner join DU on
dq.transno=DU.transno inner join Unload U on
dq.transno=U.transno

inner join UR on dq.transno=UR.transno inner join DoneQueuetbl
dqiii on dq.transno=dqiii.transno where dq.transno=@transno order
by len(dqi.transno), dqi.transno"

      Try

         Dim form As New ViewData

         con.Open()

         Dim cmd As New SqlCommand(selectquery, con)

         cmd.Parameters.AddWithValue("@transno", transno)

         cmd.ExecuteNonQuery()

         Dim dr As SqlClient.SqlDataReader

dr = cmd.ExecuteReader(CommandBehavior.CloseConnection)

         While dr.Read

            form.Label3.Text = dr("transno").ToString

            form.Label4.Text = dr("ArrivalTime").ToString

            form.Label6.Text = dr("EndEncode").ToString
            form.Label8.Text = dr("EncodeDiff").ToString

            form.Label10.Text = dr("TargetEncode").ToString

            form.Label93.Text = dr("Within").ToString

            form.Label95.Text = dr("Excess").ToString

            form.Label12.Text = dr("GuardGrade").ToString

            form.Label14.Text = dr("EndEncode").ToString

            form.Label16.Text = dr("StartV").ToString

            form.Label18.Text = dr("GVDiff").ToString

            form.Label20.Text = dr("GVTarget").ToString

            form.Label99.Text = dr("GVWithin").ToString

            form.Label97.Text = dr("GVExcess").ToString

            form.Label22.Text = dr("GVGrade").ToString

            form.Label24.Text = dr("StartV").ToString

            form.Label26.Text = dr("EndV").ToString

            form.Label28.Text = dr("Vdiff").ToString

            form.Label30.Text = dr("VTarget").ToString

            form.Label103.Text = dr("VWithin").ToString

            form.Label101.Text = dr("VExcess").ToString

            form.Label32.Text = dr("VGrade").ToString

            form.Label42.Text = dr("EndV").ToString

            form.Label40.Text = dr("StartR").ToString

            form.Label38.Text = dr("VRDiff").ToString
            form.Label36.Text = dr("VRTarget").ToString

            form.Label107.Text = dr("VRWithin").ToString

            form.Label105.Text = dr("VRExcess").ToString

            form.Label34.Text = dr("VRGrade").ToString

            form.Label44.Text = dr("RDGrade").ToString

            form.Label46.Text = dr("RDTarget").ToString

            form.Label48.Text = dr("RDDiff").ToString

            form.Label111.Text = dr("RDWithin").ToString

            form.Label109.Text = dr("RDExcess").ToString

            form.Label50.Text = dr("DT").ToString

            form.Label52.Text = dr("StartR").ToString

            form.Label62.Text = dr("DT").ToString

            form.Label60.Text = dr("SUnload").ToString

            form.Label58.Text = dr("DUDiff").ToString

            form.Label56.Text = dr("DUTarget").ToString

            form.Label115.Text = dr("DUWithin").ToString

            form.Label113.Text = dr("DUExcess").ToString

            form.Label54.Text = dr("DUGrade").ToString

            form.Label72.Text = dr("SUnload").ToString

            form.Label70.Text = dr("FUnload").ToString

            form.Label68.Text = dr("UnloadDiff").ToString

            form.Label66.Text = dr("UnloadTarget").ToString
```

```
            form.Label119.Text = dr("UnloadWithin").ToString

            form.Label117.Text = dr("UnloadExcess").ToString

            form.Label64.Text = dr("UnloadGrade").ToString

            form.Label82.Text = dr("FUnload").ToString

            form.Label80.Text = dr("FReceive").ToString

            form.Label78.Text = dr("URDiff").ToString

            form.Label76.Text = dr("URTarget").ToString

            form.Label123.Text = dr("URWithin").ToString

            form.Label121.Text = dr("URExcess").ToString

            form.Label74.Text = dr("URGrade").ToString

            form.Label92.Text = dr("driver_name").ToString

            form.Label86.Text = dr("EncodedBy").ToString

            form.Label88.Text = dr("VerifiedBy").ToString

            form.Label90.Text = dr("ReceiverName").ToString

            form.Label127.Text = dr("no_of_helper").ToString

            form.Label129.Text = dr("countofdr").ToString

            form.Label131.Text = dr("Pending").ToString

            form.Show()

        End While

    Catch ex As Exception

        MessageBox.Show(ex.Message.ToString(), "Error Message")
        End Try

    End Sub

    Private Sub Button8_Click(sender As Object, e As EventArgs)
Handles Button8.Click

        LoadCustomWeightKPIData()

        getAverage()

        queuetable4()

        TabControl5.TabPages(2).Enabled = True

        TabControl5.SelectedIndex = 2

    End Sub

    Private Sub DataGridView6_CellMouseClick(sender As Object, e
As DataGridViewCellMouseEventArgs) Handles
DataGridView6.CellMouseClick

        Dim con As New SqlClient.SqlConnection

        Dim connectionString As String = "Server='" &
My.Settings.mServer & "';

        Database='" & My.Settings.mDB & "';

        User Id='" & My.Settings.mUserDB & "';

        Password='" & My.Settings.mPassDB & "';"

        con.ConnectionString = connectionString

        i = DataGridView6.CurrentRow.Index
        transno = DataGridView6.Item(0, i).Value.ToString

        Dim selectquery As String

selectquery = "SELECT
SUBSTRING(CONVERT(VARCHAR,CAST(Arrival_date AS
DATETIME)+CAST(Arrival_time AS DATETIME),120), 1, 19) as
ArrivalTime,SUBSTRING(CONVERT(VARCHAR,CAST(Arrival_dat
e AS DATETIME)+CAST(EndEncoded AS DATETIME),120), 1, 19)
as
EndEncode,SUBSTRING(CONVERT(VARCHAR,CAST(Arrival_dat
e AS DATETIME)+CAST(StartVerify AS DATETIME),120), 1, 19)
as StartV,SUBSTRING(CONVERT(VARCHAR,CAST(Arrival_date
AS DATETIME)+CAST(EndVerify AS DATETIME),120), 1, 19) as
EndV,SUBSTRING(CONVERT(VARCHAR, startreceive,120), 1,
19) as StartR,SUBSTRING(CONVERT(VARCHAR, docktime,120),
1, 19) as DT,SUBSTRING(CONVERT(VARCHAR,
startunload,120), 1, 19) as
SUnload,SUBSTRING(CONVERT(VARCHAR, finishedunload,120),
1, 19) as FUnload,SUBSTRING(CONVERT(VARCHAR,
finishedreceive,120), 1, 19) as FReceive, * from DoneQueueInfotbl
dq inner join Encode E on dq.transno=E.transno

inner join GV G on dq.transno=G.transno inner join Verify V on
dq.transno=V.transno inner join

Donetruckdetails dqi on dq.transno=dqi.transno inner join VR on
dq.transno=VR.transno inner join

RD on dq.transno=RD.transno inner join DU on
dq.transno=DU.transno inner join Unload U on
dq.transno=U.transno

inner join UR on dq.transno=UR.transno inner join DoneQueuetbl
dqiii on dq.transno=dqiii.transno where dq.transno=@transno order
by len(dqi.transno), dqi.transno"
```

```
Try
    Dim form As New ViewData
    con.Open()
    Dim cmd As New SqlCommand(selectquery, con)
    cmd.Parameters.AddWithValue("@transno", transno)
    cmd.ExecuteNonQuery()
    Dim dr As SqlClient.SqlDataReader
    dr = cmd.ExecuteReader(CommandBehavior.CloseConnection)
    While dr.Read
        form.Label3.Text = dr("transno").ToString
        form.Label4.Text = dr("ArrivalTime").ToString
        form.Label6.Text = dr("EndEncode").ToString
        form.Label8.Text = dr("EncodeDiff").ToString
        form.Label10.Text = dr("TargetEncode").ToString
        form.Label93.Text = dr("Within").ToString
        form.Label95.Text = dr("Excess").ToString
        form.Label12.Text = dr("GuardGrade").ToString
        form.Label14.Text = dr("EndEncode").ToString
        form.Label16.Text = dr("StartV").ToString
        form.Label18.Text = dr("GVDiff").ToString
        form.Label20.Text = dr("GVTarget").ToString
        form.Label99.Text = dr("GVWithin").ToString
        form.Label97.Text = dr("GVExcess").ToString
        form.Label22.Text = dr("GVGrade").ToString
        form.Label24.Text = dr("StartV").ToString
        form.Label26.Text = dr("EndV").ToString
        form.Label28.Text = dr("Vdiff").ToString
        form.Label30.Text = dr("VTarget").ToString
        form.Label103.Text = dr("VWithin").ToString
        form.Label101.Text = dr("VExcess").ToString
        form.Label32.Text = dr("VGrade").ToString
        form.Label42.Text = dr("EndV").ToString
        form.Label40.Text = dr("StartR").ToString
        form.Label38.Text = dr("VRDiff").ToString
        form.Label36.Text = dr("VRTarget").ToString
        form.Label107.Text = dr("VRWithin").ToString
        form.Label105.Text = dr("VRExcess").ToString
        form.Label34.Text = dr("VRGrade").ToString
        form.Label44.Text = dr("RDGrade").ToString
        form.Label46.Text = dr("RDTarget").ToString
        form.Label48.Text = dr("RDDiff").ToString
        form.Label111.Text = dr("RDWithin").ToString
        form.Label109.Text = dr("RDExcess").ToString
        form.Label50.Text = dr("DT").ToString

        form.Label52.Text = dr("StartR").ToString

        form.Label62.Text = dr("DT").ToString

        form.Label60.Text = dr("SUnload").ToString

        form.Label58.Text = dr("DUDiff").ToString

        form.Label56.Text = dr("DUTarget").ToString

        form.Label115.Text = dr("DUWithin").ToString

        form.Label113.Text = dr("DUExcess").ToString

        form.Label54.Text = dr("DUGrade").ToString

        form.Label72.Text = dr("SUnload").ToString

        form.Label70.Text = dr("FUnload").ToString

        form.Label68.Text = dr("UnloadDiff").ToString

        form.Label66.Text = dr("UnloadTarget").ToString

        form.Label119.Text = dr("UnloadWithin").ToString

        form.Label117.Text = dr("UnloadExcess").ToString

        form.Label64.Text = dr("UnloadGrade").ToString

        form.Label82.Text = dr("FUnload").ToString

        form.Label80.Text = dr("FReceive").ToString

        form.Label78.Text = dr("URDiff").ToString

        form.Label76.Text = dr("URTarget").ToString

        form.Label123.Text = dr("URWithin").ToString

        form.Label121.Text = dr("URExcess").ToString
```

```
                form.Label74.Text = dr("URGrade").ToString

                form.Label92.Text = dr("driver_name").ToString

                form.Label86.Text = dr("EncodedBy").ToString

                form.Label88.Text = dr("VerifiedBy").ToString

                form.Label90.Text = dr("ReceiverName").ToString

                form.Label127.Text = dr("no_of_helper").ToString

                form.Label129.Text = dr("countofdr").ToString

                form.Label131.Text = dr("Pending").ToString

                form.Show()

            End While

        Catch ex As Exception

            MessageBox.Show(ex.Message.ToString(), "Error
Message")

        End Try

    End Sub

    Private Sub Button9_Click(sender As Object, e As EventArgs)
Handles Button9.Click

        Dim con As New SqlClient.SqlConnection

        Dim connectionString As String = "Server='" &
My.Settings.mServer & "';

    Database='" & My.Settings.mDB & "';

    User Id='" & My.Settings.mUserDB & "';

    Password='" & My.Settings.mPassDB & "';"
```

```
        con.ConnectionString = connectionString

        Dim ds As New DataSet1

        Dim sql As String

        con.Open()

sql = "select
convert(varchar(5),DoneQueueInfotbl.Arrival_date,110) as 'Date',
convert(varchar(3),DATENAME(WEEKDAY,
DoneQueueInfotbl.Arrival_date) ,110) as WeekDay,
SUBSTRING(CONVERT(VARCHAR,
DoneQueueInfotbl.Arrival_time,120), 1, 5) as 'Arrival Time',
CONVERT(VARCHAR(5),Donetruckdetails.startunload,108) AS
'Unload
Start',CONVERT(VARCHAR(5),Donetruckdetails.finishedunload,10
8) AS 'Unload End',
CONVERT(VARCHAR(5),Donetruckdetails.startreceive,108) as
'Receive
Start',CONVERT(VARCHAR(5),Donetruckdetails.finishedreceive,10
8) as 'Receive End',Donetruckdetails.difference as
'Difference',Donetruckdetails.vendorname as
'Vendor',Donetruckdetails.truck_type as 'Truck',
Donetruckdetails.truck_load as 'Load',Donetruckdetails.plate_no as
'Plate No.',Donetruckdetails.no_of_helper as 'No. of
Helper',DoneActualunloaddetails.actualtotalnoofpallet as 'Total
Pallets',Donetruckdetails.countofdr as 'Total
DR',DoneActualunloaddetails.actualtotalnoofboxes as 'Total
Boxes',DoneActualunloaddetails.actualtotalunits as 'Total
Units',Donetruckdetails.category as
'Category',DoneQueuetbl.ReceiverName from DoneQueueInfotbl
inner join Donetruckdetails on DoneQueueInfotbl.transno =
Donetruckdetails.transno inner join DoneActualunloaddetails on
DoneActualunloaddetails.transno=Donetruckdetails.transno inner
join DoneQueuetbl on DoneQueuetbl.transno =
Donetruckdetails.transno where DoneQueueInfotbl.Arrival_date
between @date1 And @date2"

        cmd5 = New SqlCommand(sql, con)

        cmd5.Parameters.AddWithValue("@date1",
CDate(DateTimePicker4.Text).ToString("yyyy-MM-dd"))

        cmd5.Parameters.AddWithValue("@date2",
CDate(DateTimePicker3.Text).ToString("yyyy-MM-dd"))

        Dim dscmd As New SqlDataAdapter(cmd5)

        dscmd.Fill(ds, "DataTable1")

        Dim objRpt As New CrystalReport1

        Dim reportprint As New PrintPreview

        objRpt.SetDataSource(ds.Tables(0))

        PrintPreview.CrystalReportViewer1.ReportSource = objRpt

        con.Close()

        PrintPreview.Show()

        Me.Hide()

    End Sub

End Class
```

Weight.vb

```
Imports System.Text.RegularExpressions

Imports System.Data.SqlClient

Imports System.Data

Public Class Weight
```

```
    Dim cmd As SqlCommand

    Dim cmd1 As SqlCommand

    Dim cmd2 As SqlCommand

    Dim cmd3 As SqlCommand

    Private Sub Button2_Click(sender As Object, e As EventArgs)
Handles Button2.Click

        Me.Close()

    End Sub

    Private Sub Weight_Load(sender As Object, e As EventArgs)
Handles MyBase.Load

        Me.FormBorderStyle = FormBorderStyle.None

    End Sub

    Private Sub Button1_Click(sender As Object, e As EventArgs)
Handles Button1.Click

        Dim con As New SqlClient.SqlConnection

        Dim connectionString As String = "Server='" &
My.Settings.mServer & "';

    Database='" & My.Settings.mDB & "';

    User Id='" & My.Settings.mUserDB & "';

    Password='" & My.Settings.mPassDB & "';
MultipleActiveResultSets=True"

Dim statquery As String = "update drtbl Set
actualnoofboxes=@actualnoofboxes, weight=@weight,
actualtotalunits=@actualtotalunits ,target=@target where
queueno=@queueno and drno=@drno"
```

```
        con.ConnectionString = connectionString

        Using conn As New SqlConnection(connectionString)

            Try

                conn.Open()

cmd = New SqlCommand(statquery, conn)

cmd.Parameters.AddWithValue("@queueno", Label5.Text)

cmd.Parameters.AddWithValue("@drno", Label3.Text)

cmd.Parameters.AddWithValue("@actualnoofboxes", TextBox1.Text)

cmd.Parameters.AddWithValue("@actualtotalunits", TextBox2.Text)

cmd.Parameters.AddWithValue("@weight", TextBox3.Text)

If CInt(TextBox1.Text) <= 50 And CInt(TextBox3.Text) <= 20 Then

cmd.Parameters.AddWithValue("@target", 60)

ElseIf CInt(TextBox1.Text) <= 50 And CInt(TextBox3.Text) >= 21
And CInt(TextBox3.Text) <= 25 Then

cmd.Parameters.AddWithValue("@target", 120)

ElseIf CInt(TextBox1.Text) >= 51 And CInt(TextBox1.Text) <= 100
And CInt(TextBox3.Text) >= 0 And CInt(TextBox3.Text) <= 15 Then

cmd.Parameters.AddWithValue("@target", 60)

ElseIf CInt(TextBox1.Text) >= 51 And CInt(TextBox1.Text) <= 100
And CInt(TextBox3.Text) >= 16 And CInt(TextBox3.Text) <= 25 Then

cmd.Parameters.AddWithValue("@target", 120)

ElseIf CInt(TextBox1.Text) >= 101 And CInt(TextBox1.Text) <= 200
And CInt(TextBox3.Text) >= 0 And CInt(TextBox3.Text) <= 10 Then
```

```
cmd.Parameters.AddWithValue("@target", 60)

ElseIf CInt(TextBox1.Text) >= 101 And CInt(TextBox1.Text) <= 200
And CInt(TextBox3.Text) >= 11 And CInt(TextBox3.Text) <= 25 Then

cmd.Parameters.AddWithValue("@target", 120)

ElseIf CInt(TextBox1.Text) >= 201 And CInt(TextBox1.Text) <= 300
And CInt(TextBox3.Text) >= 0 And CInt(TextBox3.Text) <= 5 Then

cmd.Parameters.AddWithValue("@target", 60)

ElseIf CInt(TextBox1.Text) >= 201 And CInt(TextBox1.Text) <= 300
And CInt(TextBox3.Text) >= 6 And CInt(TextBox3.Text) <= 20 Then

cmd.Parameters.AddWithValue("@target", 120)

ElseIf CInt(TextBox1.Text) >= 201 And CInt(TextBox1.Text) <= 300
And CInt(TextBox3.Text) >= 21 And CInt(TextBox3.Text) <= 25 Then

cmd.Parameters.AddWithValue("@target", 240)

ElseIf CInt(TextBox1.Text) >= 301 And CInt(TextBox1.Text) <= 400
And CInt(TextBox3.Text) >= 0 And CInt(TextBox3.Text) <= 3 Then

cmd.Parameters.AddWithValue("@target", 60)

ElseIf CInt(TextBox1.Text) >= 301 And CInt(TextBox1.Text) <= 400
And CInt(TextBox3.Text) >= 4 And CInt(TextBox3.Text) <= 15 Then

cmd.Parameters.AddWithValue("@target", 120)

ElseIf CInt(TextBox1.Text) >= 301 And CInt(TextBox1.Text) <= 400
And CInt(TextBox3.Text) >= 4 And CInt(TextBox3.Text) <= 15 Then

cmd.Parameters.AddWithValue("@target", 120)

ElseIf CInt(TextBox1.Text) >= 301 And CInt(TextBox1.Text) <= 400
And CInt(TextBox3.Text) >= 16 And CInt(TextBox3.Text) <= 25 Then

          cmd.Parameters.AddWithValue("@target", 240)
```

```
ElseIf CInt(TextBox1.Text) >= 401 And CInt(TextBox1.Text) <= 500
And CInt(TextBox3.Text) >= 0 And CInt(TextBox3.Text) <= 10 Then

cmd.Parameters.AddWithValue("@target", 120)

ElseIf CInt(TextBox1.Text) >= 401 And CInt(TextBox1.Text) <= 500
And CInt(TextBox3.Text) >= 11 And CInt(TextBox3.Text) <= 25 Then

cmd.Parameters.AddWithValue("@target", 240)

ElseIf CInt(TextBox1.Text) >= 501 And CInt(TextBox1.Text) <= 600
And CInt(TextBox3.Text) >= 0 And CInt(TextBox3.Text) <= 5 Then

cmd.Parameters.AddWithValue("@target", 120)

ElseIf CInt(TextBox1.Text) >= 501 And CInt(TextBox1.Text) <= 600
And CInt(TextBox3.Text) >= 6 And CInt(TextBox3.Text) <= 20 Then

cmd.Parameters.AddWithValue("@target", 240)

ElseIf CInt(TextBox1.Text) >= 501 And CInt(TextBox1.Text) <= 600
And CInt(TextBox3.Text) >= 21 And CInt(TextBox3.Text) <= 25 Then

cmd.Parameters.AddWithValue("@target", 360)

ElseIf CInt(TextBox1.Text) >= 601 And CInt(TextBox1.Text) <= 700
And CInt(TextBox3.Text) >= 0 And CInt(TextBox3.Text) <= 3 Then

cmd.Parameters.AddWithValue("@target", 120)

ElseIf CInt(TextBox1.Text) >= 601 And CInt(TextBox1.Text) <= 700
And CInt(TextBox3.Text) >= 4 And CInt(TextBox3.Text) <= 15 Then

cmd.Parameters.AddWithValue("@target", 240)

ElseIf CInt(TextBox1.Text) >= 601 And CInt(TextBox1.Text) <= 700
And CInt(TextBox3.Text) >= 16 And CInt(TextBox3.Text) <= 25 Then

cmd.Parameters.AddWithValue("@target", 360)
```

```
ElseIf CInt(TextBox1.Text) >= 701 And CInt(TextBox1.Text) <= 800
And CInt(TextBox3.Text) >= 0 And CInt(TextBox3.Text) <= 3 Then

cmd.Parameters.AddWithValue("@target", 120)

ElseIf CInt(TextBox1.Text) >= 701 And CInt(TextBox1.Text) <= 800
And CInt(TextBox3.Text) >= 4 And CInt(TextBox3.Text) <= 10 Then

cmd.Parameters.AddWithValue("@target", 240)

ElseIf CInt(TextBox1.Text) >= 701 And CInt(TextBox1.Text) <= 800
And CInt(TextBox3.Text) >= 11 And CInt(TextBox3.Text) <= 25 Then

cmd.Parameters.AddWithValue("@target", 360)

ElseIf CInt(TextBox1.Text) >= 801 And CInt(TextBox1.Text) <= 900
And CInt(TextBox3.Text) >= 0 And CInt(TextBox3.Text) <= 5 Then

cmd.Parameters.AddWithValue("@target", 240)

ElseIf CInt(TextBox1.Text) >= 801 And CInt(TextBox1.Text) <= 900
And CInt(TextBox3.Text) >= 6 And CInt(TextBox3.Text) <= 25 Then

cmd.Parameters.AddWithValue("@target", 360)

ElseIf CInt(TextBox1.Text) >= 901 And CInt(TextBox1.Text) <= 1000
And CInt(TextBox3.Text) >= 0 And CInt(TextBox3.Text) <= 3 Then

cmd.Parameters.AddWithValue("@target", 240)

ElseIf CInt(TextBox1.Text) >= 901 And CInt(TextBox1.Text) <= 1000
And CInt(TextBox3.Text) >= 4 And CInt(TextBox3.Text) <= 25 Then

cmd.Parameters.AddWithValue("@target", 360)

        Else

          cmd.Parameters.AddWithValue("@target", 360)

        End If
```

```
                  cmd.ExecuteNonQuery()

                  Dim Reloadtbl As New Receiver

                  Receiver.LoadNewTable()

                  conn.Close()

                  Me.Close()

              Catch ex As Exception

                  MessageBox.Show(ex.Message.ToString(), "Error
  Message")

                  conn.Close()

              End Try

          End Using

      End Sub

  End Class
```

ViewData.vb

```
Imports System.Text.RegularExpressions

Imports System.Data.SqlClient

Imports System.Data

Public Class ViewData

    Public dbs As New BindingSource

    Private Sub ViewData_Load(sender As Object, e As EventArgs)
Handles MyBase.Load

        DataGridView1.AllowUserToAddRows = False
        DataGridView1.GridColor = Color.Red

        DataGridView1.RowHeadersVisible = False

        queuetable()

        Me.FormBorderStyle = FormBorderStyle.None

    End Sub

    Private Sub Button1_Click(sender As Object, e As EventArgs)
Handles Button1.Click

        Me.Close()

    End Sub

    Public Sub queuetable()

        Dim con As New SqlClient.SqlConnection

        Dim connectionString As String = "Server='" &
My.Settings.mServer & "';

    Database='" & My.Settings.mDB & "';

    User Id='" & My.Settings.mUserDB & "';

    Password='" & My.Settings.mPassDB & "';"

        con.ConnectionString = connectionString

        Dim selectquery As String = "select drno as 'DR
No.',actualnoofboxes as 'Number of Boxes',weight as 'Weight per
Box', target as 'Target Time' from Donedrtbl where transno =
@transno"

        Try

            con.Open()

            Dim cmd = New SqlCommand(selectquery, con)
            cmd.Parameters.AddWithValue("@transno", Label3.Text.ToString)

            Dim da As New SqlDataAdapter(cmd)

            Dim dt As New DataTable()

            da.Fill(dt)

            dbs.DataSource = dt

            DataGridView1.DataSource = dbs

            DataGridView1.ReadOnly = True

        Catch ex As Exception

            MessageBox.Show(ex.Message.ToString(), "Error Message")

        End Try

    End Sub

End Class
```

sms.vb

```
Module sms

    Function itexmo(ByVal Number As String, ByVal Message As String, ByVal API As String, ByVal ApiPassword As String)

        Using client As New Net.WebClient

            Dim parameter As New Specialized.NameValueCollection

            Dim url As String = "https://www.itexmo.com/php_api/api.php"

            parameter.Add("1", Number)

            parameter.Add("2", Message)
            parameter.Add("3", API)

            parameter.Add("passwd", ApiPassword)

            Dim rpb = client.UploadValues(url, "POST", parameter)

            itexmo = (New System.Text.UTF8Encoding).GetString(rpb)

        End Using

    End Function

End Module
```

PrintPreview.vb

```
Imports System.Text.RegularExpressions

Imports System.Data.SqlClient

Imports System.Data

Imports CrystalDecisions.CrystalReports.Engine

Imports CrystalDecisions.Shared

Public Class PrintPreview

    Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click

        Dashboard.Show()

        Me.Close()

    End Sub

End Class
```

AdminPrompt.vb

```
Imports System.Data.SqlClient

Imports System.Data

Public Class AdminPrompt

    Dim con As SqlConnection

    Dim cmd As SqlCommand

    Dim Errorcatch As String

    Dim role As String

    Dim status As String

    Private Sub AdminPrompt_Load(sender As Object, e As EventArgs) Handles MyBase.Load

        Password.UseSystemPasswordChar = True

    End Sub

    Private Sub ShowPass_CheckedChanged(sender As Object, e As EventArgs) Handles ShowPass.CheckedChanged

        If ShowPass.CheckState = CheckState.Checked Then

            Password.UseSystemPasswordChar = False

        Else

            Password.UseSystemPasswordChar = True

        End If

    End Sub
    Private Sub Button2_Click(sender As Object, e As EventArgs)
Handles Button2.Click
        Me.Close()
    End Sub
    Private Sub Button1_Click(sender As Object, e As EventArgs)
Handles Button1.Click
        If Password.Text <> My.Settings.defaultPass Then
            Errorlbl.ForeColor = Color.Red
            Errorlbl.Text = "Password incorrect"
            Password.Focus()
        Else
            Me.Hide()
            AdminDBConfig.Show()
        End If
    End Sub
    Private Sub PassLbl_Click(sender As Object, e As EventArgs)
Handles PassLbl.Click
    End Sub
End Class
```

AdminDBConfig.vb

```
Public Class AdminDBConfig
    Private Sub AdminDBConfig_Load(sender As Object, e As
EventArgs) Handles MyBase.Load
```

```
Me.FormBorderStyle = FormBorderStyle.None
If AdminPrompt.Password.Text = "" Then
    TabControl1.SelectedTab = TabPage3
    Errorlbl2.Text = "Default Password is Being used, Please Create New Password for your protection!"
    Errorlbl2.ForeColor = Color.Red
    TextBox1.UseSystemPasswordChar = True
    TextBox3.UseSystemPasswordChar = True
    DBpasswordtxt.UseSystemPasswordChar = True
    APIKEYtxt.UseSystemPasswordChar = True
    APIKEYtxt.Text = My.Settings.smsAPI
    DBusernametxt.Text = My.Settings.mUserDB
    DBpasswordtxt.Text = My.Settings.mPassDB
    Servertxt.Text = My.Settings.mServer
    DBnametxt.Text = My.Settings.mDB
Else
    TextBox1.UseSystemPasswordChar = True
    TextBox3.UseSystemPasswordChar = True
    DBpasswordtxt.UseSystemPasswordChar = True
    APIKEYtxt.UseSystemPasswordChar = True
    APIKEYtxt.Text = My.Settings.smsAPI
    DBusernametxt.Text = My.Settings.mUserDB
```

```
        DBpasswordtxt.Text = My.Settings.mPassDB

        Servertxt.Text = My.Settings.mServer

        DBnametxt.Text = My.Settings.mDB

    End If

End Sub

Private Sub savebtn_Click(sender As Object, e As EventArgs)

    If Servertxt.Text = "" Then

        Errorlbl.Text = "Please type (Server IP Address)(,)(Port)!"

        Errorlbl.ForeColor = Color.Red

    ElseIf DBusernametxt.Text = "" Then

        Errorlbl.Text = "Please enter valid database username!"

        Errorlbl.ForeColor = Color.Red

    ElseIf DBnametxt.Text = "" Then

        Errorlbl.Text = "Please enter valid database Name!"

        Errorlbl.ForeColor = Color.Red

    Else

        My.Settings.mUserDB = DBusernametxt.Text

        My.Settings.mPassDB = DBpasswordtxt.Text

        My.Settings.mServer = Servertxt.Text

        My.Settings.mDB = DBnametxt.Text

        My.Settings.Save()
```

```
            MsgBox("Your server configuration has saved !",
MsgBoxStyle.Information, MsgBoxResult.Ok)

            My.Settings.Reload()

            Application.Restart()

        End If

    End Sub

    Private Sub testbtn_Click(sender As Object, e As EventArgs)

        Errorlbl.Text = ""

        Dim con As New SqlClient.SqlConnection

        Dim connectionString As String = "Server='" &
My.Settings.mServer & "';

    Database='" & My.Settings.mDB & "';

    User Id='" & My.Settings.mUserDB & "';

    Password='" & My.Settings.mPassDB & "';"

        con.ConnectionString = connectionString

        If Servertxt.Text = "" Then

            Errorlbl.Text = "Please type (Server IP Address)(,)(Port)!"

            Errorlbl.ForeColor = Color.Red

        ElseIf DBusernametxt.Text = "" Then

            Errorlbl.Text = "Please enter valid database username!"

            Errorlbl.ForeColor = Color.Red

        ElseIf DBnametxt.Text = "" Then
```

```
            Errorlbl.Text = "Please enter valid database Name!"
            Errorlbl.ForeColor = Color.Red
        Else
            If con.State = ConnectionState.Closed Then
                Try
                    con.Open()
                    MsgBox("Connected")
                Catch ex As Exception
                    MsgBox("Error on Connection Occured")
                    Errorlbl.Text = "Please check and follow Server IP format
with port"
                    Errorlbl.ForeColor = Color.Red
                End Try
            End If
        End If
    End Sub
    Private Sub CheckBox1_CheckedChanged(sender As Object, e As
EventArgs) Handles CheckBox1.CheckedChanged
        If CheckBox1.CheckState = CheckState.Checked Then
            'IF TRUE, IT SHOW THE TEXT
            DBpasswordtxt.UseSystemPasswordChar = False
        Else
```

```
            'IF FALSE, IT WILL HIDE THE TEXT AND IT WILL TURN IT INTIO BULLETS.

            DBpasswordtxt.UseSystemPasswordChar = True

        End If

    End Sub

    Private Sub Button1_Click(sender As Object, e As EventArgs)

        Me.Close()

    End Sub

    Private Sub Servertxt_TextChanged(sender As Object, e As EventArgs)

    End Sub

    Private Sub savebtn_Click_1(sender As Object, e As EventArgs) Handles savebtn.Click

        If Servertxt.Text = "" Then

            Errorlbl.Text = "Please type (Server IP Address)(,)(Port)!"

            Errorlbl.ForeColor = Color.Red

        ElseIf DBusernametxt.Text = "" Then

            Errorlbl.Text = "Please enter valid database username!"

            Errorlbl.ForeColor = Color.Red

        ElseIf DBnametxt.Text = "" Then

            Errorlbl.Text = "Please enter valid database Name!"

            Errorlbl.ForeColor = Color.Red

        Else
```

```
            My.Settings.mUserDB = DBusernametxt.Text

            My.Settings.mPassDB = DBpasswordtxt.Text

            My.Settings.mServer = Servertxt.Text

            My.Settings.mDB = DBnametxt.Text

            My.Settings.Save()

            My.Settings.Reload()

            MsgBox("Your server configuration has saved !",
MsgBoxStyle.Information, MsgBoxResult.Ok)

        End If

    End Sub

    Private Sub Button3_Click(sender As Object, e As EventArgs)
Handles Button3.Click

        If APIKEYtxt.Text = "" Then

            Errorlbl1.Text = "Please Insert Valid API Key"

            Errorlbl1.ForeColor = Color.Red

        ElseIf TextBox2.Text = "" Then

            Errorlbl1.Text = "Please Insert Mobile Number"

            Errorlbl1.ForeColor = Color.Red

        ElseIf RichTextBox1.Text = "" Then

            Errorlbl1.Text = "Please Insert Message"

            Errorlbl1.ForeColor = Color.Red

        Else
```

```
            Dim res As String = itexmo(TextBox2.Text, RichTextBox1.Text,
APIKEYtxt.Text, "(z}v7vymks")
            If res = "0" Then
                My.Settings.smsAPI = APIKEYtxt.Text
                MsgBox("Test Message Has been Sent! API code has been
saved")
                My.Settings.Save()
                My.Settings.Reload()
            Else
                MsgBox("Error on Sending Message, invalid API code")
            End If
        End If
    End Sub
    Private Sub Button2_Click(sender As Object, e As EventArgs)
Handles Button2.Click
        Application.Restart()
        Me.Close()
    End Sub
    Private Sub testbtn_Click_1(sender As Object, e As EventArgs)
Handles testbtn.Click
        Errorlbl.Text = ""
        Dim con As New SqlClient.SqlConnection
        Dim connectionString As String = "Server='" &
My.Settings.mServer & "';
```

```
Database='" & My.Settings.mDB & "';
User Id='" & My.Settings.mUserDB & "';
Password='" & My.Settings.mPassDB & "';"

    con.ConnectionString = connectionString
    If Servertxt.Text = "" Then
        Errorlbl.Text = "Please type (Server IP Address)(,)(Port)!"
        Errorlbl.ForeColor = Color.Red
    ElseIf DBusernametxt.Text = "" Then
        Errorlbl.Text = "Please enter valid database username!"
        Errorlbl.ForeColor = Color.Red
    ElseIf DBnametxt.Text = "" Then
        Errorlbl.Text = "Please enter valid database Name!"
        Errorlbl.ForeColor = Color.Red
    Else
        If con.State = ConnectionState.Closed Then
            Try
                con.Open()
                MsgBox("Connected")
            Catch ex As Exception
                MsgBox("Error on Connection Occured")
Errorlbl.Text = "Please check and follow Server IP format with port"
```

```
                    Errorlbl.ForeColor = Color.Red
                End Try
            End If
        End If
    End Sub
    Private Sub Button1_Click_1(sender As Object, e As EventArgs) Handles Button1.Click
        Application.Restart()
        Me.Close()
    End Sub
    Private Sub CheckBox2_CheckedChanged(sender As Object, e As EventArgs) Handles CheckBox2.CheckedChanged
        If CheckBox2.CheckState = CheckState.Checked Then
            APIKEYtxt.UseSystemPasswordChar = False
        Else
            APIKEYtxt.UseSystemPasswordChar = True
        End If
    End Sub
    Private Sub Button4_Click(sender As Object, e As EventArgs) Handles Button4.Click
        If APIKEYtxt.Text = "" Then
            Errorlbl1.Text = "Please Insert Valid API Key"
            Errorlbl1.ForeColor = Color.Red
        Else
            My.Settings.smsAPI = APIKEYtxt.Text
            MsgBox("API Code has been saved")
            My.Settings.Save()
            My.Settings.Reload()
        End If
    End Sub
    Private Sub Button6_Click(sender As Object, e As EventArgs)
Handles Button6.Click
        Application.Restart()
        Me.Close()
    End Sub
    Private Sub Button5_Click(sender As Object, e As EventArgs)
Handles Button5.Click
        If TextBox1.Text = "" Then
            Errorlbl2.Text = "Please Input Password!"
            Errorlbl2.ForeColor = Color.Red
        ElseIf TextBox3.Text = "" Then
            Errorlbl2.Text = "Please Input Confirm Password!"
            Errorlbl2.ForeColor = Color.Red
        ElseIf TextBox1.Text <> TextBox3.Text Then
```

```
            Errorlbl2.Text = "Input Password are not the same!"

            Errorlbl2.ForeColor = Color.Red

        Else

            My.Settings.defaultPass = TextBox1.Text

            My.Settings.Save()

            My.Settings.Reload()

            MsgBox("New Password has been saved")

        End If

    End Sub

    Private Sub CheckBox3_CheckedChanged(sender As Object, e As EventArgs) Handles CheckBox3.CheckedChanged

        If CheckBox3.CheckState = CheckState.Checked Then

            TextBox1.UseSystemPasswordChar = False

            TextBox3.UseSystemPasswordChar = False

        Else

            TextBox1.UseSystemPasswordChar = True

            TextBox3.UseSystemPasswordChar = True

        End If

    End Sub

End Class
```

Admin.vb

```
Imports System.Text.RegularExpressions

Imports System.Data.SqlClient

Imports System.Data

Imports System.ComponentModel

Public Class Admin

    Dim con As SqlConnection

    Dim cmd As SqlCommand

    Dim adapter As SqlDataAdapter

    Dim dr As SqlDataReader

    Dim table As New DataTable

    Dim Errorcatch As String

    Dim i As Integer

    Dim stat As Integer

    Public dbs As New BindingSource

    Public Sub LoadTable()

        Dim con As New SqlClient.SqlConnection

        Dim connectionString As String = "Server='" &
My.Settings.mServer & "';

    Database='" & My.Settings.mDB & "';

    User Id='" & My.Settings.mUserDB & "';

    Password='" & My.Settings.mPassDB & "';"
```

```
        con.ConnectionString = connectionString

        Dim selectquery As String = "select IDnumber,Username, Name from Usertbl"

        Try

            con.Open()

            cmd = New SqlCommand(selectquery, con)

            Dim da As New SqlDataAdapter(cmd)

            Dim dt As New DataTable()

            da.Fill(dt)

            dbs.DataSource = dt

            AccountList.DataSource = dbs

            AccountList.ReadOnly = True

        Catch ex As Exception

            MessageBox.Show(ex.Message.ToString(), "Error Message")

        End Try

    End Sub

    Private Sub Admin_Load(sender As Object, e As EventArgs) Handles MyBase.Load

        AccountList.RowHeadersVisible = False

        AccountList.GridColor = Color.Red

        Me.FormBorderStyle = FormBorderStyle.None

        FilterComboBox.Text = "ID number"
        Dim myname As New DBconnection

        Name = myname.ReceiverName

        Label21.Text = Name

        stat = 0

        Dim con As New SqlClient.SqlConnection

        Dim connectionString As String = "Server='" &
My.Settings.mServer & "';

    Database='" & My.Settings.mDB & "';

    User Id='" & My.Settings.mUserDB & "';

    Password='" & My.Settings.mPassDB & "';"

        con.ConnectionString = connectionString

        Idtxt.MaxLength = 10

        Contacttxt.MaxLength = 11

        Pass.UseSystemPasswordChar = True

        conPass.UseSystemPasswordChar = True

        If con.State = ConnectionState.Open Then

            con.Close()

        End If

        con.Open()

        LoadTable()

    End Sub
    Private Sub CheckBox1_CheckedChanged(sender As Object, e As EventArgs) Handles CheckBox1.CheckedChanged

        If CheckBox1.CheckState = CheckState.Checked Then

            Pass.UseSystemPasswordChar = False

            conPass.UseSystemPasswordChar = False

        Else

            Pass.UseSystemPasswordChar = True

            conPass.UseSystemPasswordChar = True

        End If

    End Sub

    Private Sub Button2_Click(sender As Object, e As EventArgs) Handles Button2.Click

        If stat = 1 Then

            If Pass.Text = "" And conPass.Text = "" Then

                Dim regex As Regex = New Regex("([a-zA-Z0-9_\-\.]+)@((\[[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\." +

                                           ")|(([a-zA-Z0-9\-]+\.)+))([a-zA-Z]{2,4}|[0-9]{1,3})",

                                           RegexOptions.IgnoreCase _

                                           Or RegexOptions.CultureInvariant _

                                           Or RegexOptions.IgnorePatternWhitespace _

                                           Or RegexOptions.Compiled

                                           )
```

```
Dim IsMatch As Boolean = regex.IsMatch(Emailtxt.Text)
If Len(Trim(Nametxt.Text)) = 0 Then
    Errorlbl.ForeColor = Color.Red
    Errorlbl.Text = "Error: Please Enter Name"
    conPasslbl.ForeColor = Color.Black
    Namelbl.ForeColor = Color.Red
    Idlbl.ForeColor = Color.Black
    Passlbl.ForeColor = Color.Black
    Emaillbl.ForeColor = Color.Black
    Usernamelbl.ForeColor = Color.Black
    Contactlbl.ForeColor = Color.Black
    Statuslbl.ForeColor = Color.Black
    Nametxt.Focus()
ElseIf Idtxt.Text = "" Then
    Errorlbl.ForeColor = Color.Red
    Errorlbl.Text = "Error: Please Enter ID number"
    conPasslbl.ForeColor = Color.Black
    Namelbl.ForeColor = Color.Black
    Idlbl.ForeColor = Color.Red
    Passlbl.ForeColor = Color.Black
    Emaillbl.ForeColor = Color.Black
    Usernamelbl.ForeColor = Color.Black
```

```
        Contactlbl.ForeColor = Color.Black

        Statuslbl.ForeColor = Color.Black

        Idtxt.Focus()

    ElseIf Emailtxt.Text = "" Then

        Errorlbl.ForeColor = Color.Red

        Errorlbl.Text = "Error: Please Enter Email Address"

        conPasslbl.ForeColor = Color.Black

        Namelbl.ForeColor = Color.Black

        Idlbl.ForeColor = Color.Black

        Passlbl.ForeColor = Color.Black

        Emaillbl.ForeColor = Color.Red

        Usernamelbl.ForeColor = Color.Black

        Contactlbl.ForeColor = Color.Black

        Statuslbl.ForeColor = Color.Black

        Emailtxt.Focus()

    ElseIf Emailtxt.Text <>
(regex.Match(Emailtxt.Text).ToString) Then

        Errorlbl.ForeColor = Color.Red

        Errorlbl.Text = "Sorry.  Invalid email address format."

        conPasslbl.ForeColor = Color.Black

        Namelbl.ForeColor = Color.Black

        Idlbl.ForeColor = Color.Black
    Passlbl.ForeColor = Color.Black
    Emaillbl.ForeColor = Color.Red
    Usernamelbl.ForeColor = Color.Black
    Contactlbl.ForeColor = Color.Black
    UserRolelbl.ForeColor = Color.Black
    Statuslbl.ForeColor = Color.Black
    Emailtxt.Focus()
ElseIf Contacttxt.Text = "" Then
    Errorlbl.ForeColor = Color.Red
    Errorlbl.Text = "Error: Please Enter Contact Number"
    conPasslbl.ForeColor = Color.Black
    Namelbl.ForeColor = Color.Black
    Idlbl.ForeColor = Color.Black
    Passlbl.ForeColor = Color.Black
    Emaillbl.ForeColor = Color.Black
    Usernamelbl.ForeColor = Color.Black
    Contactlbl.ForeColor = Color.Red
    UserRolelbl.ForeColor = Color.Black
    Statuslbl.ForeColor = Color.Black
    Contacttxt.Focus()
ElseIf Contacttxt.TextLength < 7 Then
    Errorlbl.ForeColor = Color.Red
```

```
        Errorlbl.Text = "Sorry.  Invalid Contact number format"
        conPasslbl.ForeColor = Color.Black
        Namelbl.ForeColor = Color.Black
        Idlbl.ForeColor = Color.Black
        Passlbl.ForeColor = Color.Black
        Emaillbl.ForeColor = Color.Black
        Usernamelbl.ForeColor = Color.Black
        Contactlbl.ForeColor = Color.Red
        UserRolelbl.ForeColor = Color.Black
        Statuslbl.ForeColor = Color.Black
        Contacttxt.Focus()
    ElseIf Usernametxt.Text = "" Then
        Errorlbl.ForeColor = Color.Red
        Errorlbl.Text = "Error: Please Enter Username"
        conPasslbl.ForeColor = Color.Black
        Namelbl.ForeColor = Color.Black
        Idlbl.ForeColor = Color.Black
        Passlbl.ForeColor = Color.Black
        Emaillbl.ForeColor = Color.Black
        Usernamelbl.ForeColor = Color.Red
        Contactlbl.ForeColor = Color.Black
        UserRolelbl.ForeColor = Color.Black
        Statuslbl.ForeColor = Color.Black
        Usernametxt.Focus()
    ElseIf UserRole.Text = "" Then
        Errorlbl.ForeColor = Color.Red
        Errorlbl.Text = "Error: Please Select User Role"
        conPasslbl.ForeColor = Color.Black
        Namelbl.ForeColor = Color.Black
        Idlbl.ForeColor = Color.Black
        Passlbl.ForeColor = Color.Black
        Emaillbl.ForeColor = Color.Black
        Usernamelbl.ForeColor = Color.Black
        Contactlbl.ForeColor = Color.Black
        UserRolelbl.ForeColor = Color.Red
        Statuslbl.ForeColor = Color.Black
        UserRole.Focus()
    ElseIf Status.Text = "" Then
        Errorlbl.ForeColor = Color.Red
        Errorlbl.Text = "Error: Please Select Status"
        conPasslbl.ForeColor = Color.Black
        Namelbl.ForeColor = Color.Black
        Idlbl.ForeColor = Color.Black
        Passlbl.ForeColor = Color.Black
        Emaillbl.ForeColor = Color.Black

        Usernamelbl.ForeColor = Color.Black

        Contactlbl.ForeColor = Color.Black

        UserRolelbl.ForeColor = Color.Black

        Statuslbl.ForeColor = Color.Red

        Status.Focus()

    ElseIf Pass.Text <> conPass.Text Then

        Pass.Focus()

        Passlbl.ForeColor = Color.Red

        conPasslbl.ForeColor = Color.Red

        Errorlbl.ForeColor = Color.Red

        Errorlbl.Text = "Error: Password did not Match"

        Pass.Text = ""

        conPass.Text = ""

    Else

        Dim con As New SqlClient.SqlConnection

        Dim connectionString As String = "Server='" &
My.Settings.mServer & "';

   Database='" & My.Settings.mDB & "';

   User Id='" & My.Settings.mUserDB & "';

   Password='" & My.Settings.mPassDB & "';"
```

```
         con.ConnectionString = connectionString

         Dim query As String = String.Empty

       query &= "UPDATE Usertbl Set Name=@Name,
       IDnumber=@IDnumber, Email=@Email, "

  query &=
  "Contactno=@Contactno,UserRole=@UserRole,Status=@Status
  where Username =@Username  "

         Using conn As New SqlConnection(connectionString)

           Using comm As New SqlCommand()

             With comm

.Connection = conn

.CommandType = CommandType.Text

.CommandText = query

.Parameters.AddWithValue("@Name", Nametxt.Text)

.Parameters.AddWithValue("@IDnumber", Idtxt.Text)

.Parameters.AddWithValue("@Username", Usernametxt.Text)

.Parameters.AddWithValue("@Email", Emailtxt.Text)

.Parameters.AddWithValue("@Contactno", Contacttxt.Text)

.Parameters.AddWithValue("@UserRole", UserRole.Text)

.Parameters.AddWithValue("@Status", Status.Text)

             End With

             Try

               conn.Open()
```

```
                                comm.ExecuteNonQuery()
MsgBox("Updated Successfully", MsgBoxStyle.Information)
                                LoadTable()
                                Nametxt.Clear()
                                Idtxt.Clear()
                                Emailtxt.Clear()
                                Contacttxt.Clear()
                                Usernametxt.Clear()
                                Pass.Clear()
                                conPass.Clear()
                                Status.Text = ""
                                UserRole.Text = ""
                                AccountList.ClearSelection()
                            Catch ex As Exception
MessageBox.Show(ex.Message.ToString(), "Error Message")
                            End Try
                        End Using
                    End Using
                    con.Close()
                End If
            Else
```

```
Dim regex As Regex = New Regex("([a-zA-Z0-9_\-\.]+)@((\[[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\." +

")|(([a-zA-Z0-9\-]+\.)+))([a-zA-Z]{2,4}|[0-9]{1,3})",

RegexOptions.IgnoreCase _

Or RegexOptions.CultureInvariant _

Or RegexOptions.IgnorePatternWhitespace _

Or RegexOptions.Compiled

)

Dim IsMatch As Boolean = regex.IsMatch(Emailtxt.Text)

        If Len(Trim(Nametxt.Text)) = 0 Then

          Errorlbl.ForeColor = Color.Red

          Errorlbl.Text = "Error: Please Enter Name"

          conPasslbl.ForeColor = Color.Black

          Namelbl.ForeColor = Color.Red

          Idlbl.ForeColor = Color.Black

          Passlbl.ForeColor = Color.Black

          Emaillbl.ForeColor = Color.Black

          Usernamelbl.ForeColor = Color.Black

          Contactlbl.ForeColor = Color.Black

          Statuslbl.ForeColor = Color.Black

          Nametxt.Focus()
```

```
ElseIf Idtxt.Text = "" Then
    Errorlbl.ForeColor = Color.Red
    Errorlbl.Text = "Error: Please Enter ID number"
    conPasslbl.ForeColor = Color.Black
    Namelbl.ForeColor = Color.Black
    Idlbl.ForeColor = Color.Red
    Passlbl.ForeColor = Color.Black
    Emaillbl.ForeColor = Color.Black
    Usernamelbl.ForeColor = Color.Black
    Contactlbl.ForeColor = Color.Black
    Statuslbl.ForeColor = Color.Black
    Idtxt.Focus()
ElseIf Emailtxt.Text = "" Then
    Errorlbl.ForeColor = Color.Red
    Errorlbl.Text = "Error: Please Enter Email Address"
    conPasslbl.ForeColor = Color.Black
    Namelbl.ForeColor = Color.Black
    Idlbl.ForeColor = Color.Black
    Passlbl.ForeColor = Color.Black
    Emaillbl.ForeColor = Color.Red
    Usernamelbl.ForeColor = Color.Black
    Contactlbl.ForeColor = Color.Black
```

```
        Statuslbl.ForeColor = Color.Black

        Emailtxt.Focus()

    ElseIf Emailtxt.Text <>
(regex.Match(Emailtxt.Text).ToString) Then

        Errorlbl.ForeColor = Color.Red

        Errorlbl.Text = "Sorry.  Invalid email address format."

        conPasslbl.ForeColor = Color.Black

        Namelbl.ForeColor = Color.Black

        Idlbl.ForeColor = Color.Black

        Passlbl.ForeColor = Color.Black

        Emaillbl.ForeColor = Color.Red

        Usernamelbl.ForeColor = Color.Black

        Contactlbl.ForeColor = Color.Black

        UserRolelbl.ForeColor = Color.Black

        Statuslbl.ForeColor = Color.Black

        Emailtxt.Focus()

    ElseIf Contacttxt.Text = "" Then

        Errorlbl.ForeColor = Color.Red

        Errorlbl.Text = "Error: Please Enter Contact Number"

        conPasslbl.ForeColor = Color.Black

        Namelbl.ForeColor = Color.Black

        Idlbl.ForeColor = Color.Black
    Passlbl.ForeColor = Color.Black
    Emaillbl.ForeColor = Color.Black
    Usernamelbl.ForeColor = Color.Black
    Contactlbl.ForeColor = Color.Red
    UserRolelbl.ForeColor = Color.Black
    Statuslbl.ForeColor = Color.Black
    Contacttxt.Focus()
ElseIf Contacttxt.TextLength < 7 Then
    Errorlbl.ForeColor = Color.Red
    Errorlbl.Text = "Sorry.  Invalid Contact number format"
    conPasslbl.ForeColor = Color.Black
    Namelbl.ForeColor = Color.Black
    Idlbl.ForeColor = Color.Black
    Passlbl.ForeColor = Color.Black
    Emaillbl.ForeColor = Color.Black
    Usernamelbl.ForeColor = Color.Black
    Contactlbl.ForeColor = Color.Red
    UserRolelbl.ForeColor = Color.Black
    Statuslbl.ForeColor = Color.Black
    Contacttxt.Focus()
ElseIf Usernametxt.Text = "" Then
    Errorlbl.ForeColor = Color.Red
    Errorlbl.Text = "Error: Please Enter Username"
    conPasslbl.ForeColor = Color.Black
    Namelbl.ForeColor = Color.Black
    Idlbl.ForeColor = Color.Black
    Passlbl.ForeColor = Color.Black
    Emaillbl.ForeColor = Color.Black
    Usernamelbl.ForeColor = Color.Red
    Contactlbl.ForeColor = Color.Black
    UserRolelbl.ForeColor = Color.Black
    Statuslbl.ForeColor = Color.Black
    Usernametxt.Focus()
ElseIf Pass.Text = "" Then
    Errorlbl.ForeColor = Color.Red
    Errorlbl.Text = "Error: Please Enter Password"
    conPasslbl.ForeColor = Color.Black
    Namelbl.ForeColor = Color.Black
    Idlbl.ForeColor = Color.Black
    Passlbl.ForeColor = Color.Red
    Emaillbl.ForeColor = Color.Black
    Usernamelbl.ForeColor = Color.Black
    Contactlbl.ForeColor = Color.Black
    UserRolelbl.ForeColor = Color.Black
    Statuslbl.ForeColor = Color.Black
    Pass.Focus()
ElseIf conPass.Text = "" Then
    Errorlbl.ForeColor = Color.Red
    Errorlbl.Text = "Error: Please Enter Confirm Password"
    conPasslbl.ForeColor = Color.Red
    Namelbl.ForeColor = Color.Black
    Idlbl.ForeColor = Color.Black
    Passlbl.ForeColor = Color.Black
    Emaillbl.ForeColor = Color.Black
    Usernamelbl.ForeColor = Color.Black
    Contactlbl.ForeColor = Color.Black
    UserRolelbl.ForeColor = Color.Black
    Statuslbl.ForeColor = Color.Black
    conPass.Focus()
ElseIf UserRole.Text = "" Then
    Errorlbl.ForeColor = Color.Red
    Errorlbl.Text = "Error: Please Select User Role"
    conPasslbl.ForeColor = Color.Black
    Namelbl.ForeColor = Color.Black
    Idlbl.ForeColor = Color.Black
    Passlbl.ForeColor = Color.Black
    Emaillbl.ForeColor = Color.Black
    Usernamelbl.ForeColor = Color.Black
    Contactlbl.ForeColor = Color.Black
    UserRolelbl.ForeColor = Color.Red
    Statuslbl.ForeColor = Color.Black
    UserRole.Focus()
ElseIf Status.Text = "" Then
    Errorlbl.ForeColor = Color.Red
    Errorlbl.Text = "Error: Please Select Status"
    conPasslbl.ForeColor = Color.Black
    Namelbl.ForeColor = Color.Black
    Idlbl.ForeColor = Color.Black
    Passlbl.ForeColor = Color.Black
    Emaillbl.ForeColor = Color.Black
    Usernamelbl.ForeColor = Color.Black
    Contactlbl.ForeColor = Color.Black
    UserRolelbl.ForeColor = Color.Black
    Statuslbl.ForeColor = Color.Red
    Status.Focus()
ElseIf Pass.Text <> conPass.Text Then
    Pass.Focus()
    Passlbl.ForeColor = Color.Red
```

```
                    conPasslbl.ForeColor = Color.Red

                    Errorlbl.ForeColor = Color.Red

                    Errorlbl.Text = "Error: Password did not Match"

                    Pass.Text = ""

                    conPass.Text = ""

                Else

                    Dim con As New SqlClient.SqlConnection

                    '   Dim con As New DBconnection

                    Dim connectionString As String = "Server='" &
My.Settings.mServer & "';

    Database='" & My.Settings.mDB & "';

    User Id='" & My.Settings.mUserDB & "';

    Password='" & My.Settings.mPassDB & "';"

                    con.ConnectionString = connectionString

                    Dim query As String = String.Empty

                    query &= "UPDATE Usertbl Set Name=@Name,
IDnumber=@IDnumber, Email=@Email, "

                    query &=
"Contactno=@Contactno,Password=@Password,UserRole=@UserRo
le,Status=@Status where Username =@Username  "

                    Using conn As New SqlConnection(connectionString)

                        Using comm As New SqlCommand()

                            With comm
```

```
.Connection = conn
.CommandType = CommandType.Text
.CommandText = query
.Parameters.AddWithValue("@Name", Nametxt.Text)
.Parameters.AddWithValue("@IDnumber", Idtxt.Text)
.Parameters.AddWithValue("@Email", Emailtxt.Text)
.Parameters.AddWithValue("@Contactno", Contacttxt.Text)
.Parameters.AddWithValue("@Username", Usernametxt.Text)
.Parameters.AddWithValue("@Password", Pass.Text)
.Parameters.AddWithValue("@UserRole", UserRole.Text)
.Parameters.AddWithValue("@Status", Status.Text)
                    End With
                    Try
                       conn.Open()
                       comm.ExecuteNonQuery()
MsgBox("Password Updated Successfully", MsgBoxStyle.Information)
                       LoadTable()
                       Nametxt.Clear()
                       Idtxt.Clear()
                       Emailtxt.Clear()
                       Contacttxt.Clear()
                       Usernametxt.Clear()
```

```
                                Pass.Clear()

                                conPass.Clear()

                                Status.Text = ""

                                UserRole.Text = ""

                                AccountList.ClearSelection()

                            Catch ex As Exception

MessageBox.Show(ex.Message.ToString(), "Error Message")

                            End Try

                        End Using

                    End Using

                    con.Close()

                End If

            End If

        Else

Dim regex As Regex = New Regex("([a-zA-Z0-9_\-\.]+)@((\[[0-
9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\." +

")|(([a-zA-Z0-9\-]+\.)+))([a-zA-Z]{2,4}|[0-9]{1,3})",

RegexOptions.IgnoreCase _

Or RegexOptions.CultureInvariant _

Or RegexOptions.IgnorePatternWhitespace _

Or RegexOptions.Compiled

)
```

```
Dim IsMatch As Boolean = regex.IsMatch(Emailtxt.Text)
If Len(Trim(Nametxt.Text)) = 0 Then
    Errorlbl.ForeColor = Color.Red
    Errorlbl.Text = "Error: Please Enter Name"
    conPasslbl.ForeColor = Color.Black
    Namelbl.ForeColor = Color.Red
    Idlbl.ForeColor = Color.Black
    Passlbl.ForeColor = Color.Black
    Emaillbl.ForeColor = Color.Black
    Usernamelbl.ForeColor = Color.Black
    Contactlbl.ForeColor = Color.Black
    Statuslbl.ForeColor = Color.Black
    Nametxt.Focus()
ElseIf Idtxt.Text = "" Then
    Errorlbl.ForeColor = Color.Red
    Errorlbl.Text = "Error: Please Enter ID number"
    conPasslbl.ForeColor = Color.Black
    Namelbl.ForeColor = Color.Black
    Idlbl.ForeColor = Color.Red
    Passlbl.ForeColor = Color.Black
    Emaillbl.ForeColor = Color.Black
    Usernamelbl.ForeColor = Color.Black
```

```
            Contactlbl.ForeColor = Color.Black

            Statuslbl.ForeColor = Color.Black

            Idtxt.Focus()

        ElseIf Emailtxt.Text = "" Then

            Errorlbl.ForeColor = Color.Red

            Errorlbl.Text = "Error: Please Enter Email Address"

            conPasslbl.ForeColor = Color.Black

            Namelbl.ForeColor = Color.Black

            Idlbl.ForeColor = Color.Black

            Passlbl.ForeColor = Color.Black

            Emaillbl.ForeColor = Color.Red

            Usernamelbl.ForeColor = Color.Black

            Contactlbl.ForeColor = Color.Black

            Statuslbl.ForeColor = Color.Black

            Emailtxt.Focus()

        ElseIf Emailtxt.Text <> (regex.Match(Emailtxt.Text).ToString)
Then

            Errorlbl.ForeColor = Color.Red

            Errorlbl.Text = "Sorry.  Invalid email address format."

            conPasslbl.ForeColor = Color.Black

            Namelbl.ForeColor = Color.Black

            Idlbl.ForeColor = Color.Black
```

```
    Passlbl.ForeColor = Color.Black

    Emaillbl.ForeColor = Color.Red

    Usernamelbl.ForeColor = Color.Black

    Contactlbl.ForeColor = Color.Black

    UserRolelbl.ForeColor = Color.Black

    Statuslbl.ForeColor = Color.Black

    Emailtxt.Focus()

ElseIf Contacttxt.Text = "" Then

    Errorlbl.ForeColor = Color.Red

    Errorlbl.Text = "Error: Please Enter Contact Number"

    conPasslbl.ForeColor = Color.Black

    Namelbl.ForeColor = Color.Black

    Idlbl.ForeColor = Color.Black

    Passlbl.ForeColor = Color.Black

    Emaillbl.ForeColor = Color.Black

    Usernamelbl.ForeColor = Color.Black

    Contactlbl.ForeColor = Color.Red

    UserRolelbl.ForeColor = Color.Black

    Statuslbl.ForeColor = Color.Black

    Contacttxt.Focus()

ElseIf Contacttxt.TextLength < 7 Then

    Errorlbl.ForeColor = Color.Red
    Errorlbl.Text = "Sorry.  Invalid Contact number format"
    conPasslbl.ForeColor = Color.Black
    Namelbl.ForeColor = Color.Black
    Idlbl.ForeColor = Color.Black
    Passlbl.ForeColor = Color.Black
    Emaillbl.ForeColor = Color.Black
    Usernamelbl.ForeColor = Color.Black
    Contactlbl.ForeColor = Color.Red
    UserRolelbl.ForeColor = Color.Black
    Statuslbl.ForeColor = Color.Black
    Contacttxt.Focus()
ElseIf Usernametxt.Text = "" Then
    Errorlbl.ForeColor = Color.Red
    Errorlbl.Text = "Error: Please Enter Username"
    conPasslbl.ForeColor = Color.Black
    Namelbl.ForeColor = Color.Black
    Idlbl.ForeColor = Color.Black
    Passlbl.ForeColor = Color.Black
    Emaillbl.ForeColor = Color.Black
    Usernamelbl.ForeColor = Color.Red
    Contactlbl.ForeColor = Color.Black
    UserRolelbl.ForeColor = Color.Black
    Statuslbl.ForeColor = Color.Black
    Usernametxt.Focus()
ElseIf Pass.Text = "" Then
    Errorlbl.ForeColor = Color.Red
    Errorlbl.Text = "Error: Please Enter Password"
    conPasslbl.ForeColor = Color.Black
    Namelbl.ForeColor = Color.Black
    Idlbl.ForeColor = Color.Black
    Passlbl.ForeColor = Color.Red
    Emaillbl.ForeColor = Color.Black
    Usernamelbl.ForeColor = Color.Black
    Contactlbl.ForeColor = Color.Black
    UserRolelbl.ForeColor = Color.Black
    Statuslbl.ForeColor = Color.Black
    Pass.Focus()
ElseIf conPass.Text = "" Then
    Errorlbl.ForeColor = Color.Red
    Errorlbl.Text = "Error: Please Enter Confirm Password"
    conPasslbl.ForeColor = Color.Red
    Namelbl.ForeColor = Color.Black
    Idlbl.ForeColor = Color.Black
    Passlbl.ForeColor = Color.Black
    Emaillbl.ForeColor = Color.Black
    Usernamelbl.ForeColor = Color.Black
    Contactlbl.ForeColor = Color.Black
    UserRolelbl.ForeColor = Color.Black
    Statuslbl.ForeColor = Color.Black
    conPass.Focus()
ElseIf UserRole.Text = "" Then
    Errorlbl.ForeColor = Color.Red
    Errorlbl.Text = "Error: Please Select User Role"
    conPasslbl.ForeColor = Color.Black
    Namelbl.ForeColor = Color.Black
    Idlbl.ForeColor = Color.Black
    Passlbl.ForeColor = Color.Black
    Emaillbl.ForeColor = Color.Black
    Usernamelbl.ForeColor = Color.Black
    Contactlbl.ForeColor = Color.Black
    UserRolelbl.ForeColor = Color.Red
    Statuslbl.ForeColor = Color.Black
    UserRole.Focus()
ElseIf Status.Text = "" Then
    Errorlbl.ForeColor = Color.Red
    Errorlbl.Text = "Error: Please Select Status"
    conPasslbl.ForeColor = Color.Black

    Namelbl.ForeColor = Color.Black

    Idlbl.ForeColor = Color.Black

    Passlbl.ForeColor = Color.Black

    Emaillbl.ForeColor = Color.Black

    Usernamelbl.ForeColor = Color.Black

    Contactlbl.ForeColor = Color.Black

    UserRolelbl.ForeColor = Color.Black

    Statuslbl.ForeColor = Color.Red

    Status.Focus()

ElseIf Pass.Text <> conPass.Text Then

    Pass.Focus()

    Passlbl.ForeColor = Color.Red

    conPasslbl.ForeColor = Color.Red

    Errorlbl.ForeColor = Color.Red

    Errorlbl.Text = "Error: Password did not Match"

    Pass.Text = ""

    conPass.Text = ""

Else

    Dim con As New SqlClient.SqlConnection

   Dim connectionString As String = "Server='" &
   My.Settings.mServer & "';
```

```
    Database='" & My.Settings.mDB & "';

    User Id='" & My.Settings.mUserDB & "';

    Password='" & My.Settings.mPassDB & "';"

                con.ConnectionString = connectionString

                Dim selectquery As String = "select Username from Usertbl
where username=@username"

                con.Open()

                cmd = New SqlCommand(selectquery, con)

                cmd.Parameters.AddWithValue("@username",
Usernametxt.Text)

                Dim dr As SqlDataReader

                dr = cmd.ExecuteReader()

                While dr.Read

                    Errorlbl.Text = "Username Already Taken"

                    con.Close()

                    Exit Sub

                End While

                Dim query As String = String.Empty

                query &= "INSERT INTO Usertbl (Name, IDnumber, Email, "

                query &= "                    Contactno, Username,
                Password,UserRole,Status)  "

                query &= "VALUES (@Name,@IDnumber, @Email,
                @Contactno,@Username, @Password, @UserRole,
                @Status)"
                Using conn As New SqlConnection(connectionString)
                    Using comm As New SqlCommand()
                        With comm
.Connection = conn
.CommandType = CommandType.Text
.CommandText = query
.Parameters.AddWithValue("@Name", Nametxt.Text)
.Parameters.AddWithValue("@IDnumber", Idtxt.Text)
.Parameters.AddWithValue("@Email", Emailtxt.Text)
.Parameters.AddWithValue("@Contactno", Contacttxt.Text)
.Parameters.AddWithValue("@Username", Usernametxt.Text)
.Parameters.AddWithValue("@Password", Pass.Text)
.Parameters.AddWithValue("@UserRole", UserRole.Text)
.Parameters.AddWithValue("@Status", Status.Text)
                        End With
                        Try
                            Dim dialog As DialogResult
dialog = MessageBox.Show("Are you sure you want to Create this New Account", "Exit", MessageBoxButtons.YesNo)
                            If dialog = DialogResult.No Then
                            Else
                                conn.Open()
                                comm.ExecuteNonQuery()
                                LoadTable()
MsgBox("Registered Successfully", MsgBoxStyle.Information)
                                Nametxt.Clear()
                                Idtxt.Clear()
                                Emailtxt.Clear()
                                Contacttxt.Clear()
                                Usernametxt.Clear()
                                Pass.Clear()
                                conPass.Clear()
                                Status.Text = ""
                                UserRole.Text = ""
                            End If
                        Catch ex As Exception
MessageBox.Show(ex.Message.ToString(), "Error Message")
                        End Try
                    End Using
                End Using
                con.Close()
```

```
            End If
        End If
    End Sub
    Private Sub Contacttxt_KeyPress(sender As Object, e As KeyPressEventArgs) Handles Contacttxt.KeyPress
        Select Case e.KeyChar
            Case "0", "1", "2", "3", "4", "5", "6", "7", "8", "9", vbBack '// your pre-selected Characters and Backspace.
                e.Handled = False
            Case Else
                e.Handled = True
        End Select
    End Sub
    Private Sub Contacttxt_GotFocus(sender As Object, e As EventArgs) Handles Contacttxt.GotFocus
        If Contacttxt.TextLength.Equals(0) Then
            Contacttxt.Text = "09"
        End If
    End Sub
    Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
        stat = 0
        conPasslbl.ForeColor = Color.Black
        Namelbl.ForeColor = Color.Black

        Idlbl.ForeColor = Color.Black

        Passlbl.ForeColor = Color.Black

        Emaillbl.ForeColor = Color.Black

        Usernamelbl.ForeColor = Color.Black

        Contactlbl.ForeColor = Color.Black

        UserRolelbl.ForeColor = Color.Black

        Statuslbl.ForeColor = Color.Black

        Errorlbl.Text = ""

        Nametxt.Clear()

        Idtxt.Clear()

        Emailtxt.Clear()

        Contacttxt.Clear()

        Usernametxt.Clear()

        Pass.Clear()

        conPass.Clear()

        Status.Text = ""

        UserRole.Text = ""

        AccountList.ClearSelection()

    End Sub
```

```
    Private Sub Button3_Click(sender As Object, e As EventArgs)
Handles Button3.Click

        If Nametxt.Text = "" Then

            MessageBox.Show("Select Account to Delete", "Error
Message")

        Else

            Dim con As New SqlClient.SqlConnection

            Dim connectionString As String = "Server='" &
My.Settings.mServer & "';

    Database='" & My.Settings.mDB & "';

    User Id='" & My.Settings.mUserDB & "';

    Password='" & My.Settings.mPassDB & "';"

            con.ConnectionString = connectionString

            con.Open()

            Dim query As String = String.Empty

query &= "INSERT INTO UserArchieves (Name, IDnumber, Email, "

query &= "                    Contactno, Username,
Password,UserRole,Status)  "

query &= "SELECT t1.Name, t1.IDnumber, t1.Email, t1.Contactno,
t1.Username, t1.Password, t1.UserRole, t1.Status "

query &= "from Usertbl t1 where username =@Username"


Dim deletequery As String = "delete from Usertbl where username
=@username AND idnumber =@idnumber AND name =@name "
```

```
Dim QueryString As String = String.Concat(query, ";", deletequery)
            Using conn As New SqlConnection(connectionString)
                Using comm As New SqlCommand()
                    With comm
.Connection = conn
.CommandType = CommandType.Text
.CommandText = QueryString
.Parameters.AddWithValue("@Name", Nametxt.Text)
.Parameters.AddWithValue("@IDnumber", Idtxt.Text)
.Parameters.AddWithValue("@Email", Emailtxt.Text)
.Parameters.AddWithValue("@Contactno", Contacttxt.Text)
.Parameters.AddWithValue("@Username", Usernametxt.Text)
.Parameters.AddWithValue("@Password", Pass.Text)
.Parameters.AddWithValue("@UserRole", UserRole.Text)
.Parameters.AddWithValue("@Status", Status.Text)
                    End With
                    Try
                        Dim dialog As DialogResult
dialog = MessageBox.Show("Are you sure you want to Delete this?",
"Exit", MessageBoxButtons.YesNo)
                        If dialog = DialogResult.No Then
                        Else
                            conn.Open()
                            comm.ExecuteNonQuery()
                            LoadTable()
                            MsgBox("Delete Successfully",
MsgBoxStyle.Information)
                            Nametxt.Clear()
                            Idtxt.Clear()
                            Emailtxt.Clear()
                            Contacttxt.Clear()
                            Usernametxt.Clear()
                            Pass.Clear()
                            conPass.Clear()
                            Status.Text = ""
                            UserRole.Text = ""
                        End If
                    Catch ex As Exception
MessageBox.Show(ex.Message.ToString(), "Error Message")
                    End Try
                End Using
            End Using
            con.Close()
```

```
        End If
    End Sub
    Private Sub AccountList_CellMouseClick(sender As Object, e As
DataGridViewCellMouseEventArgs) Handles
AccountList.CellMouseClick
        stat = 1
        Dim con As New SqlClient.SqlConnection
        Dim connectionString As String = "Server='" &
My.Settings.mServer & "';
    Database='" & My.Settings.mDB & "';
    User Id='" & My.Settings.mUserDB & "';
    Password='" & My.Settings.mPassDB & "';"
        con.ConnectionString = connectionString
        i = AccountList.CurrentRow.Index
        Dim usernameval As String
        Dim IDnumberval As String
        Dim nameval As String
        IDnumberval = AccountList.Item(0, i).Value.ToString
        usernameval = AccountList.Item(1, i).Value.ToString
        nameval = AccountList.Item(2, i).Value.ToString
Dim selectquery As String = "select * from Usertbl where username
=@username AND idnumber =@idnumber AND name =@name "
        Try
        con.Open()

        cmd = New SqlCommand(selectquery, con)

        cmd.Parameters.AddWithValue("@username", usernameval)

        cmd.Parameters.AddWithValue("@idnumber", IDnumberval)

        cmd.Parameters.AddWithValue("@name", nameval)

        cmd.ExecuteNonQuery()

        Dim dt As New DataTable()

        Dim adapter As New SqlDataAdapter(cmd)

        adapter.Fill(dt)

        Dim dr As SqlClient.SqlDataReader

dr = cmd.ExecuteReader(CommandBehavior.CloseConnection)

        While dr.Read

           Nametxt.Text = dr("Name").ToString

           Idtxt.Text = dr("IDnumber").ToString

           Emailtxt.Text = dr("EMail").ToString

           Contacttxt.Text = dr("Contactno").ToString

           Usernametxt.Text = dr("Username").ToString

           UserRole.Text = dr("UserRole").ToString

           Status.Text = dr("Status").ToString
```

```
            End While
        Catch ex As Exception
MessageBox.Show(ex.Message.ToString(), "Error Message")
        End Try
    End Sub
    Private Sub AccountList_CellContentClick(sender As Object, e As DataGridViewCellEventArgs) Handles AccountList.CellContentClick
    End Sub
    Private Sub Idtxt_TextChanged(sender As Object, e As EventArgs) Handles Idtxt.TextChanged
        If Idtxt.Text.Length >= 5 Then
            Dim s As String = Idtxt.Text.Replace("-", "")
            For x As Integer = 6 To s.Length Step 4
                s = s.Insert(x + ((x \ 4) - 1), "-")
            Next
            s = s.TrimEnd("-"c)
            Idtxt.Text = s
            Idtxt.SelectionStart = s.Length
        End If
    End Sub
    Private Sub Idtxt_KeyPress(sender As Object, e As KeyPressEventArgs) Handles Idtxt.KeyPress
        Select Case e.KeyChar
```

```
            Case "0", "1", "2", "3", "4", "5", "6", "7", "8", "9", vbBack '// your pre-selected Characters and Backspace.

                e.Handled = False '// allow.

            Case Else

                e.Handled = True '// block.

        End Select

    End Sub

    Private Sub Button4_Click(sender As Object, e As EventArgs) Handles Button4.Click

        conPasslbl.ForeColor = Color.Black

        Namelbl.ForeColor = Color.Black

        Idlbl.ForeColor = Color.Black

        Passlbl.ForeColor = Color.Black

        Emaillbl.ForeColor = Color.Black

        Usernamelbl.ForeColor = Color.Black

        Contactlbl.ForeColor = Color.Black

        UserRolelbl.ForeColor = Color.Black

        Statuslbl.ForeColor = Color.Black

        Errorlbl.Text = ""

        Nametxt.Clear()

        Idtxt.Clear()

        Emailtxt.Clear()
        Contacttxt.Clear()

        Usernametxt.Clear()

        Pass.Clear()

        conPass.Clear()

        Status.Text = ""

        UserRole.Text = ""

        AccountList.ClearSelection()

    End Sub

    Private Sub TextBox1_TextChanged(sender As Object, e As
EventArgs) Handles TextBox1.TextChanged

        If FilterComboBox.Text.Equals("ID number") Then

            dbs.Filter = String.Format("IDnumber LIKE '%{0}%'",
TextBox1.Text)

        ElseIf FilterComboBox.Text.Equals("Username") Then

            dbs.Filter = String.Format("Username LIKE '%{0}%'",
TextBox1.Text)

        ElseIf FilterComboBox.Text.Equals("Name") Then

            dbs.Filter = String.Format("Name LIKE '%{0}%'",
TextBox1.Text)

        End If

    End Sub

    Private Sub Logout_Click(sender As Object, e As EventArgs)
Handles Logout.Click

        Dim dialog As DialogResult
        dialog = MessageBox.Show("Are you sure you want to Logout?", "Exit", MessageBoxButtons.YesNo)

        If dialog = DialogResult.No Then

        Else

            Me.Hide()

            Login.Show()

            Dim logout As New DBconnection

            logout.ReceiverName = ""

        End If

    End Sub

    Private Sub Timer1_Tick(sender As Object, e As EventArgs) Handles Timer1.Tick

        Label17.Text = DateTime.Now.ToString("MMMM dd, yyyy h:mm:ss tt")

    End Sub

End Class
```

Monitor.vb

```
Imports System.Text.RegularExpressions

Imports System.Data.SqlClient

Imports System.Data

Public Class Monitor

    Dim con As SqlConnection

    Dim cmd As SqlCommand
    Dim adapter As SqlDataAdapter

    Dim dr As SqlDataReader

    Public dbs As New BindingSource

    Dim table As New DataTable

    Public Sub LoadTable()

        Dim con As New SqlClient.SqlConnection

        Dim connectionString As String = "Server='" &
My.Settings.mServer & "';

    Database='" & My.Settings.mDB & "';

    User Id='" & My.Settings.mUserDB & "';

    Password='" & My.Settings.mPassDB & "';"

        con.ConnectionString = connectionString

        Dim selectquery As String = "select Queueno from Queue_tbl
where Status='Pending'"

        Try

            con.Open()

            cmd = New SqlCommand(selectquery, con)

            Dim da As New SqlDataAdapter(cmd)

            Dim dt As New DataTable()

            da.Fill(dt)

            dbs.DataSource = dt

            PendingList.DataSource = dbs
            PendingList.ReadOnly = True

            con.Close()

        Catch ex As Exception

            MessageBox.Show(ex.Message.ToString(), "Error Message")

            con.Close()

        End Try

    End Sub

    Public Sub LoadB4D1()

        Dim con As New SqlClient.SqlConnection

        Dim connectionString As String = "Server='" &
My.Settings.mServer & "';

    Database='" & My.Settings.mDB & "';

    User Id='" & My.Settings.mUserDB & "';

    Password='" & My.Settings.mPassDB & "';"

        con.ConnectionString = connectionString

        Dim selectquery As String = "select Top 1 Queueno from bldg4d1
where ReceiverName is not null"

        Using conn As New SqlConnection(connectionString)

            conn.Open()

            Try

                cmd = New SqlCommand(selectquery, conn)

                cmd.ExecuteNonQuery()
                Dim dr As SqlClient.SqlDataReader
dr = cmd.ExecuteReader(CommandBehavior.CloseConnection)
                While dr.Read
                    Label9.Text = dr("Queueno").ToString
                End While
            Catch ex As Exception
MessageBox.Show(ex.Message.ToString(), "Error Message")
                conn.Close()
            End Try
            conn.Close()
        End Using
    End Sub
    Public Sub LoadB4D2()
        Dim con As New SqlClient.SqlConnection
        Dim connectionString As String = "Server='" &
My.Settings.mServer & "';
    Database='" & My.Settings.mDB & "';
    User Id='" & My.Settings.mUserDB & "';
    Password='" & My.Settings.mPassDB & "';"
        con.ConnectionString = connectionString
Dim selectquery As String = "select Top 1 Queueno from bldg4d2
where ReceiverName is not null"
```

```
        Using conn As New SqlConnection(connectionString)
            conn.Open()
            Try
                cmd = New SqlCommand(selectquery, conn)
                cmd.ExecuteNonQuery()
                Dim dr As SqlClient.SqlDataReader
dr = cmd.ExecuteReader(CommandBehavior.CloseConnection)
                While dr.Read
                    Label10.Text = dr("Queueno").ToString
                End While
            Catch ex As Exception
MessageBox.Show(ex.Message.ToString(), "Error Message")
                conn.Close()
            End Try
            conn.Close()
        End Using
    End Sub
    Public Sub LoadB4D3()
        Dim con As New SqlClient.SqlConnection
        Dim connectionString As String = "Server='" &
My.Settings.mServer & "';
    Database='" & My.Settings.mDB & "';
    User Id='" & My.Settings.mUserDB & "';

    Password='" & My.Settings.mPassDB & "';"

        con.ConnectionString = connectionString

        Dim selectquery As String = "select Top 1 Queueno from
Bldg4d3 where ReceiverName is not null"

        Using conn As New SqlConnection(connectionString)

            conn.Open()

            Try

                cmd = New SqlCommand(selectquery, conn)

                cmd.ExecuteNonQuery()

                Dim dr As SqlClient.SqlDataReader

                dr =
cmd.ExecuteReader(CommandBehavior.CloseConnection)

                While dr.Read

                    Label11.Text = dr("Queueno").ToString

                End While

            Catch ex As Exception

MessageBox.Show(ex.Message.ToString(), "Error Message")

                conn.Close()

            End Try

            conn.Close()

        End Using
    End Sub

    Public Sub LoadB4D4()

        Dim con As New SqlClient.SqlConnection

        Dim connectionString As String = "Server='" &
My.Settings.mServer & "';

    Database='" & My.Settings.mDB & "';

    User Id='" & My.Settings.mUserDB & "';

    Password='" & My.Settings.mPassDB & "';"

        con.ConnectionString = connectionString

Dim selectquery As String = "select Top 1 Queueno from Bldg4d4
where ReceiverName is not null"

        Using conn As New SqlConnection(connectionString)

            conn.Open()

            Try

                cmd = New SqlCommand(selectquery, conn)

                cmd.ExecuteNonQuery()

                Dim dr As SqlClient.SqlDataReader

dr = cmd.ExecuteReader(CommandBehavior.CloseConnection)

                While dr.Read

                    Label12.Text = dr("Queueno").ToString

                End While

            Catch ex As Exception
```

```
MessageBox.Show(ex.Message.ToString(), "Error Message")
                conn.Close()
            End Try
            conn.Close()
        End Using
    End Sub
    Public Sub LoadB3D1()
        panelbuilding3.Dock = DockStyle.Fill
        Dim con As New SqlClient.SqlConnection
        Dim connectionString As String = "Server='" &
My.Settings.mServer & "';
    Database='" & My.Settings.mDB & "';
    User Id='" & My.Settings.mUserDB & "';
    Password='" & My.Settings.mPassDB & "';"
        con.ConnectionString = connectionString
        Dim selectquery As String = "select Top 1 Queueno from Bldg3
where ReceiverName is not null"
        Using conn As New SqlConnection(connectionString)
            conn.Open()
            Try
                cmd = New SqlCommand(selectquery, conn)
                cmd.ExecuteNonQuery()
                Dim dr As SqlClient.SqlDataReader
dr = cmd.ExecuteReader(CommandBehavior.CloseConnection)
                While dr.Read
                    Label4.Text = dr("Queueno").ToString
                End While
            Catch ex As Exception
                MessageBox.Show(ex.Message.ToString(), "Error
Message")
                conn.Close()
            End Try
            conn.Close()
        End Using
    End Sub
    Public Sub LoadB6D1()
        Dim con As New SqlClient.SqlConnection
        Dim connectionString As String = "Server='" &
My.Settings.mServer & "';
    Database='" & My.Settings.mDB & "';
    User Id='" & My.Settings.mUserDB & "';
    Password='" & My.Settings.mPassDB & "';"
        con.ConnectionString = connectionString
Dim selectquery As String = "select Top 1 Queueno from Bldg6 where
ReceiverName is not null"
```

```
        Using conn As New SqlConnection(connectionString)

            conn.Open()

            Try

                cmd = New SqlCommand(selectquery, conn)

                cmd.ExecuteNonQuery()

                Dim dr As SqlClient.SqlDataReader

dr = cmd.ExecuteReader(CommandBehavior.CloseConnection)

                While dr.Read

                    Label8.Text = dr("Queueno").ToString

                End While

            Catch ex As Exception

MessageBox.Show(ex.Message.ToString(), "Error Message")

                conn.Close()

            End Try

            conn.Close()

        End Using

    End Sub

    Private Sub Monitor_Load(sender As Object, e As EventArgs)
Handles MyBase.Load

        Dim con As New SqlClient.SqlConnection

        Dim connectionString As String = "Server='" &
My.Settings.mServer & "';
```

```
    Database='" & My.Settings.mDB & "';
    User Id='" & My.Settings.mUserDB & "';
    Password='" & My.Settings.mPassDB &
"';MultipleActiveResultSets=True"
        con.ConnectionString = connectionString
        Dim selectquery As String = "select max(date) as dateme from
resetdate"
        Dim resetqueuetbl As String = "DBCC CHECKIDENT
(Queue_tbl, RESEED, 0)"
        Dim resettrucktbl As String = "DBCC CHECKIDENT
(truck_details, RESEED, 0)"
        Dim insertdate As String = "insert into resetdate select convert
(date, getdate())"
        If con.State = ConnectionState.Closed Then
            Try
                con.Open()
                Dim datetoday As String
                Try
                    Dim Command As New SqlCommand(selectquery, con)
                    Command.ExecuteNonQuery()
                    Dim dr As SqlClient.SqlDataReader
dr = Command.ExecuteReader(CommandBehavior.CloseConnection)
                    While dr.Read
                        datetoday = dr("dateme").ToString
                        If datetoday = Date.Today.ToString Then
                        Else
Dim Command1 As New SqlCommand(insertdate, con)
Command1.ExecuteNonQuery()
Dim Command2 As New SqlCommand(resetqueuetbl, con)
Command2.ExecuteNonQuery()
Dim Command3 As New SqlCommand(resettrucktbl, con)
Command3.ExecuteNonQuery()
                        End If
                    End While
                Catch ex As Exception
MessageBox.Show(ex.Message.ToString(), "Error Message")
                End Try
                Timer1.Enabled = True
            Catch ex As Exception
                Timer1.Enabled = False
                MsgBox("Error on Connection Occured")
                AdminPrompt.Show()
            End Try
        End If
        Me.KeyPreview = True
Me.MaximumSize =
Screen.FromRectangle(Me.Bounds).WorkingArea.Size

    End Sub

    Public Sub Timer1_Tick(sender As Object, e As EventArgs)
Handles Timer1.Tick

        Label4.Text = ""

        Label9.Text = ""

        Label8.Text = ""

        Label10.Text = ""

        Label11.Text = ""

        Label12.Text = ""

        LoadB3D1()

        LoadB4D1()

        LoadB4D2()

        LoadB4D3()

        LoadB4D4()

        LoadB6D1()

        LoadTable()

    End Sub

    Private Sub Monitor_KeyDown(sender As Object, e As
KeyEventArgs) Handles Me.KeyDown

        If e.KeyCode = Keys.D And e.Shift And e.Control Then

            AdminPrompt.Show()
            Timer1.Enabled = False
        ElseIf e.KeyCode = Keys.F4 And e.Alt Then
        End If
    End Sub
End Class
```

EVALUATION TOOL

Survey Questionnaire

Dear Respondent:

This survey is part of BS in Computer Science thesis student project at the Taguig City University. To ensure your anonymity, all of your answers are kept in utmost confidentiality. Your completed survey answers will be only being seen by the researcher and their adviser. Regardless of the situation, no individual responses will be identified on any of the questionnaires.

Directions. We are interested in your views about Data Analytics for Effective Truck Queuing Management System, please take your time but try not to linger on any one question, your first response to the question is usually your true belief.

Thanks Again!

Please read each question carefully and indicate your response by selecting the most appropriate choice.

Researcher
Abrina, John Louie A.
Botona, Jay-alou B.
Evangelista, Dianne Tan
Mahinay, Alexis

PRE-SURVEY QUESTIONNAIRE

Part A. Demographic Profile

1. Are you a DHL or Non DHL Employee?

□ Associate

□ Leadership

2. Job Position.

□ Manager

□ Supervisor

□ Team Leader

□ System Associate

□ Trucker Helper

□ Trucker Driver

□ Others: _________

3. Gender

□ Male

□ Female

4. Civil Status

□ Single

□ Married

□ Widowed

5. Age

□ 18 - 23

□ 24 - 28

□ 29 - 33

□ 34 – 38

□ 38 & up

6. No. of years in the company ______

Part B. Questionnaire Proper

1. What is your level of agreement on the problem of using manual queuing system?

(4) Serious problem (3) Moderate problem (2) Minor problem (1) Not at all a problem

1.1 Manual Queue Management System	4	3	2	1
1.1.1 Long waiting time of queues				
1.1.2 Missing queuing numbers				
1.1.3 Manual truck door assignment				
1.1.4 Availability of wait time information				
1.1.5 Storage of Data				
1.1.5 Calling of trucks to be unload				

2. What is the likelihood that you will support the deployment of Automated truck queuing management system in your department?

(5) Extremely likely (4) Likely (3) Neutral (2) Unlikely (1) Extremely unlikely

Automated Truck Management System	5	4	3	2	1
2.1.1 Organized queue management System					
2.1.2 Paperless queuing number by using SMS					
2.1.3 Online database for Data					
2.1.4 Monitor display and SMS notification for time waiting.					
2.1.5 Automatic truck door assignment					

3. What is your level agreement on the identified characteristics of the system?

(5) Strongly Agree (4) Agree (3) Partly Agree (2) Disagree (1) Strongly Disagree

3.1 Features and Functionality	**5**	**4**	**3**	**2**	**1**
3.1.1 Features and functionality meet common user goals and objectives.					
3.1.2 Frequently-used tasks are readily available (e.g. easily accessible from the forms) and well supported.					
3.1.3 Users are adequately supported according to their level of expertise					
3.1.4 Calls to action (e.g. Save, submit) are clear, well labelled and appear clickable.					

3.2 Navigation	**5**	**4**	**3**	**2**	**1**
3.2.1 Users can easily access the site or application.					
3.2.2 The navigational scheme is easy to find, intuitive and consistent.					
3.2.3 The navigation has sufficient flexibility to allow users to navigate by their desired means. .					
3.2.4 The site or application structure is clear, easily understood and addresses common user goals					
3.2.5 Menus, descriptive and well labelled.					

4. What is your level of acceptance on the proposed subscription model of the software prototype?

(5) Perfectly Acceptable (4) Acceptable (3) Partly Acceptable (2) Slightly Acceptable (1) Unacceptable

4.1 Predictable Revenue for the school and the researchers	**5**	**4**	**3**	**2**	**1**
4.1.1 Allows the university and the researcher for additional source of income					
4.1.2 Can create accurate forecasts if required for possible publication from authors and subscribers					
4.1.3 Increase in revenue would mean better projects from both ends					

5. What is your level of agreement on the implementation of a strong security backbone of the system?

5 – Strongly Agree 4 – Agree 3 – Partly Agree 2 – Disagree 1 – Strongly Disagree

5.1 Strong passwords	**5**	**4**	**3**	**2**	**1**
5.1.1 Using a different password than					
5.1.2 Creating a password that has a mixture of capital letters, lowercase letters, numbers, and symbols					
5.1.3 Avoiding dictionary words, anniversaries, birthdays, or other combinations that could be easily guessed					
5.1.4 Prioritizing length — the longer and more complex a password is, the harder it is to crack, even by a program					

--Thank You--

POST-SURVEY QUESTIONNAIRE

Part A: Respondents Profile

The Researchers are conducting an assessment about some aspects of the proposed system Truck Queue Management System.

Name(Optional):_____________________________

Date:___________________

□ **DHL** □ **Non DHL**

Gender: □ **Male** □ **Female**

Age: □ **18-25 years old** □ **26-35 years old** □ **above**

Part B: System Evaluation

Please answer the question by putting a check mark (**/**) on the appropriate column.

Evaluation	**Strongly Agree 4**	**Partially Agree 3**	**Disagree 2**	**Strongly Disagree 1**
Reliability				
1. This system has a provide real time updates.				
2. The system can provide faster way of processing queues.				

3. This system helps to reduce the waiting time.				
Usability				
1. The Development of Truck queue management system is useful to DHL.				
2. Recommend using this system to others.				
3. This system help the DHL to ease the long waiting of Queues.				
4. The Queue management system gives a real time information to stakeholders.				

Functionality				
1. The system provides queue no.				
2. The system is capable of giving update through sms.				
3. The System display information of the queues through monitor display.				

4. The System can generate Statistical & Graphical Reports				
Efficiency				
1. The system process queues more quicker than the traditional process.				
2. The system provides help option once installed.				
3. System provides real time reports of the queues.				
4. The System track the queues.				
Portability				
1. The system is working in windows 7 and above.				
2. The System provides software ready for online installation.				
3. The system has a hardware requirement of Laptop computer, 64bit System Architecture and 2GB RAM.				

--Thank You—

DATA ANALYTICS AND SMS NOTIFICATION FOR EFFECTIVE QUEUEING MANAGEMENT SYSTEM
V 1.0

User Manual

1 Content

2 Introduction

Data Analytics And Sms Notification For Effective Queueing Management System will help to ease the process of queuing. With the help of the new technology the encoding and storing of data becomes easy. The use of sms the queue number and the status of queueing will automatically send via text message. The monitoring of queues will display on the monitor.Data analytics can analyze the delivery history and improve the current operation that will minimize logistical costs. Applying business intelligence, this system also can develop a performance metrics and KPIs, and create computer models to predict delivery issues. As queuing management systems collects data, it should be put to good use. Reporting capabilities will contribute to better visibility and control over future logistics outcomes.

3 Features

- Displays a real time update via monitor display.
- Automated Dock Assignment
- Generates a statistical reports that can be use as management decision making.
- Timely monitoring of Receiving Status.
- Can be accessed thru wired and wireless network.
- SMS notification for Dock Assignment and Delivery Receipt Status.
- Daily Reports Printing

4 System Requirements

- **Server Requirements**

1. Microsoft SQL Server 2012
2. Windows 7 or Higher
3. Install the latest service packs
4. Install the latest Framework

Recommended Server Hardware:

CPU	Intel Core i3 or Higher
CPU Speed	2GHz or Higher
Display Adapter	Built-in
Network Adapter	100/1000Mbps
Memory	4gb or Higher
Hard Disk Space Required	1TB
Monitor	Any
Keyboard	Any
Mouse	Any

- **User Requirements**

1. Windows 7 or Higher
2. Install the latest service packs
3. Install the latest Framework

Recommended User Hardware:

CPU	Intel Core i3 or Higher
CPU Speed	2GHz or Higher
Display Adapter	Built-in
Network Adapter	100/1000Mbps
Memory	4gb or Higher
Hard Disk Space Required	50GB
Monitor	Any
Keyboard	Any
Mouse	Any

- **Monitoring Requirements**

1. Windows 7 or Higher
2. Install the latest service packs
3. Install the latest Framework

Recommended Monitoring Desktop Hardware:

CPU	Intel Core i3 or Higher
CPU Speed	2GHz or Higher
Display Adapter	Built-in
Network Adapter	100/1000Mbps
Memory	4gb or Higher
Hard Disk Space Required	50GB
Keyboard	Any
Mouse	Any

Recommended Monitor Specification:

Screen Size	32 in. Or Higher
Resolution	1366x768 or Higher
Aspect Ratio	16:9
Dynamic Contrast Ratio	3000:1
VESA Mount Compatible	Yes
Connectivity:	VGA AV IN AV OUT S-Video In

- **Networking Requirements**

  1. Lan Cables
  2. Switch

  For better performance, we recommend 1000M Ethernet network environment.

5 Quick Start

5.1 Install Microsoft SQL Server 2012 -Server Side

Inside the Queueing Management System Folder, Run SQLEXPR_x64_ENU.exe as administrator

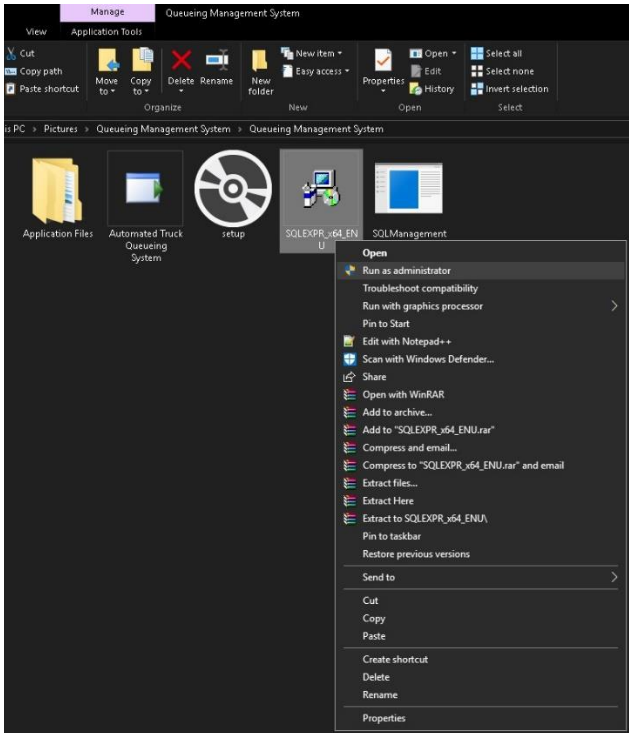
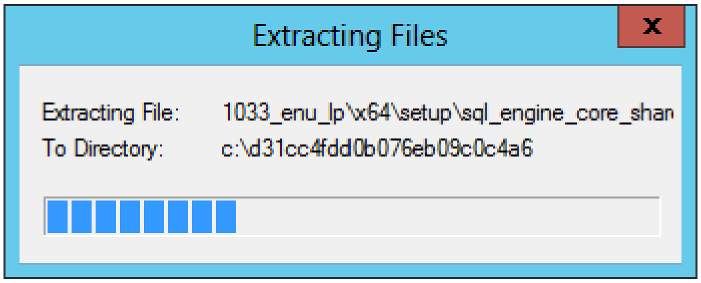


Figure 5-1

The set up file will start extracting the installation files

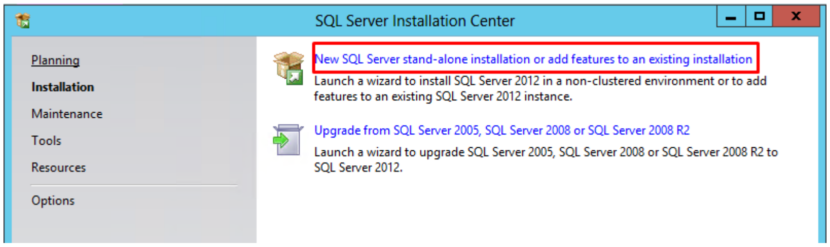


Extracting Files

Extracting File: 1033_enu_lp\x64\setup\sql_engine_core_shar

To Directory: c:\d31cc4fdd0b076eb09c0c4a6

On the next screen choose the top option to install a new stand-alone installation.



SQL Server Installation Center

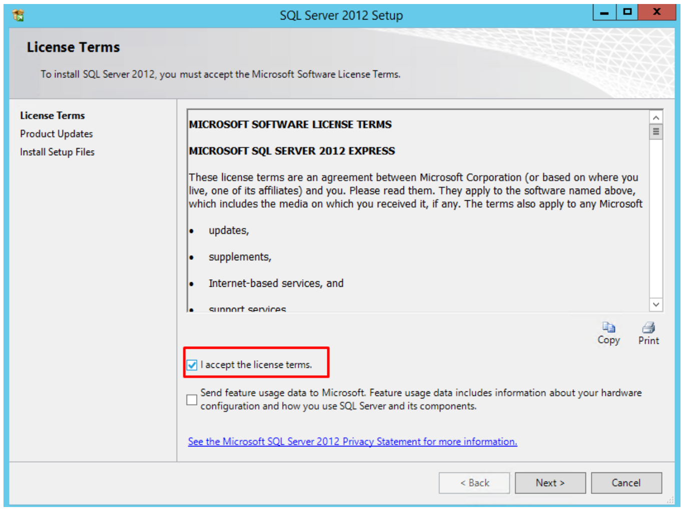
New SQL Server stand-alone installation or add features to an existing installation

Launch a wizard to install SQL Server 2012 in a non-clustered environment or to add features to an existing SQL Server 2012 instance.

Upgrade from SQL Server 2005, SQL Server 2008 or SQL Server 2008 R2

Launch a wizard to upgrade SQL Server 2005, SQL Server 2008 or SQL Server 2008 R2 to SQL Server 2012.

The set up will process the basic operations and you will see the licensing screen where you have to click the check box for “I accept the license terms”, then click next.



SQL Server 2012 Setup

License Terms

To install SQL Server 2012, you must accept the Microsoft Software License Terms.

MICROSOFT SOFTWARE LICENSE TERMS

MICROSOFT SQL SERVER 2012 EXPRESS

These license terms are an agreement between Microsoft Corporation (or based on where you live, one of its affiliates) and you. Please read them. They apply to the software named above, which includes the media on which you received it, if any. The terms also apply to any Microsoft

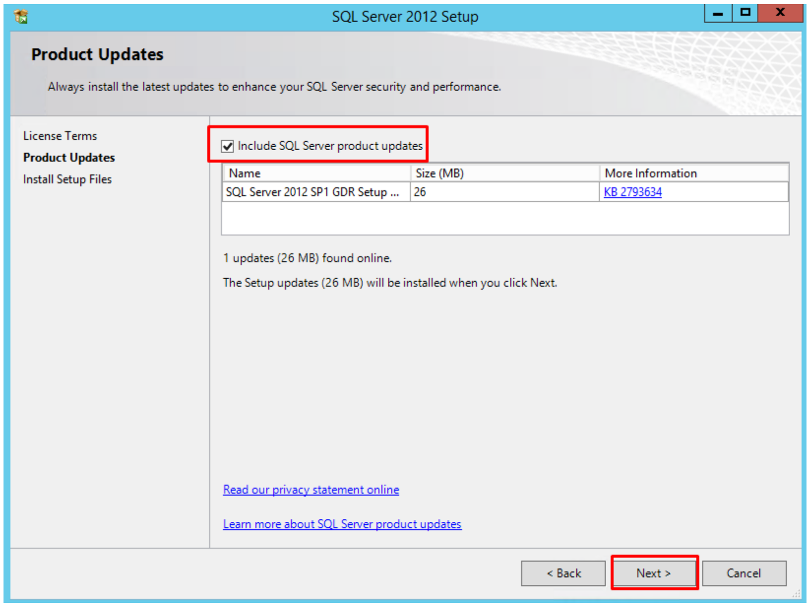
- updates,
- supplements,
- Internet-based services, and
- support services

☑ I accept the license terms.

□ Send feature usage data to Microsoft. Feature usage data includes information about your hardware configuration and how you use SQL Server and its components.

See the Microsoft SQL Server 2012 Privacy Statement for more information.

The next screen will be Product Updates, Leave it as it is and click on next.



SQL Server 2012 Setup

Product Updates

Always install the latest updates to enhance your SQL Server security and performance.

License Terms
Product Updates
Install Setup Files

☑ Include SQL Server product updates

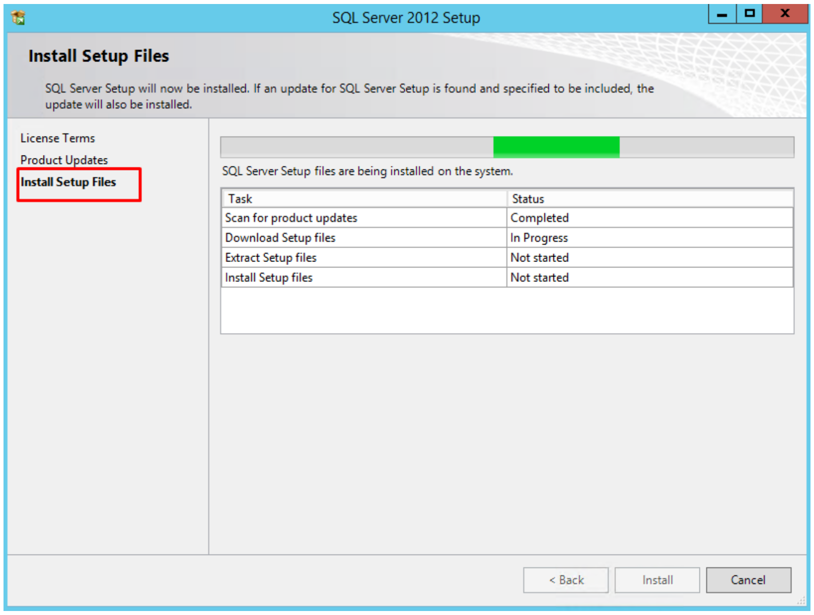
Name	Size (MB)	More Information
SQL Server 2012 SP1 GDR Setup ...	26	KB 2793634

1 updates (26 MB) found online.

The Setup updates (26 MB) will be installed when you click Next.

Read our privacy statement online
Learn more about SQL Server product updates
< Back
Next >
Cancel

On the next screen you will see that the set up is completing the installations.



SQL Server 2012 Setup

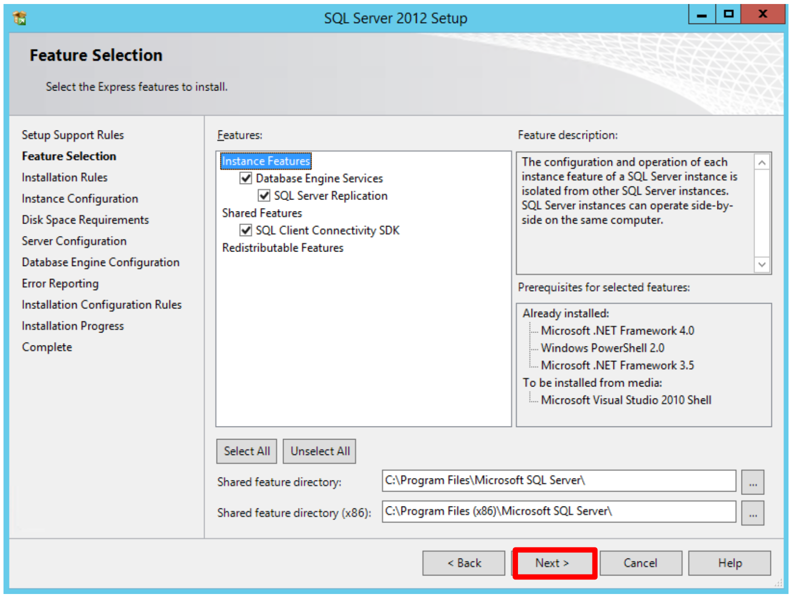
Install Setup Files

SQL Server Setup will now be installed. If an update for SQL Server Setup is found and specified to be included, the update will also be installed.

SQL Server Setup files are being installed on the system.

Task	Status
Scan for product updates	Completed
Download Setup files	In Progress
Extract Setup files	Not started
Install Setup files	Not started

On the Feature Selection screen, please leave it as it is and click on Next.



SQL Server 2012 Setup

Feature Selection

Select the Express features to install.

Setup Support Rules
Feature Selection
Installation Rules
Instance Configuration
Disk Space Requirements
Server Configuration
Database Engine Configuration
Error Reporting
Installation Configuration Rules
Installation Progress
Complete

Features:

Instance Features
- ☑ Database Engine Services
  - ☑ SQL Server Replication

Shared Features
- ☑ SQL Client Connectivity SDK

Redistributable Features

Feature description:

The configuration and operation of each instance feature of a SQL Server instance is isolated from other SQL Server instances. SQL Server instances can operate side-by-side on the same computer.

Prerequisites for selected features:

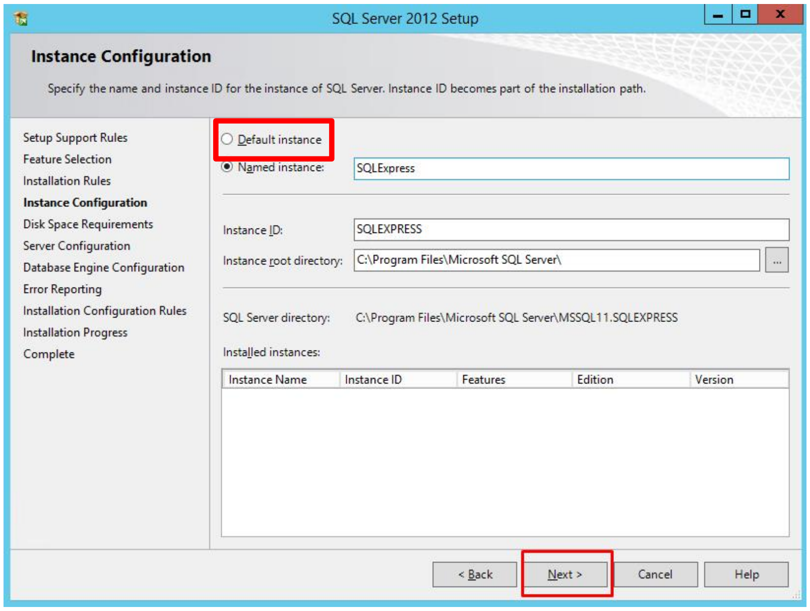
Already installed:
- Microsoft .NET Framework 4.0
- Windows PowerShell 2.0
- Microsoft .NET Framework 3.5

To be installed from media:
- Microsoft Visual Studio 2010 Shell

Shared feature directory: C:\Program Files\Microsoft SQL Server\

Shared feature directory (x86): C:\Program Files (x86)\Microsoft SQL Server\

On the Instance Configuration screen, select default instance then click on Next.



SQL Server 2012 Setup

Instance Configuration

Specify the name and instance ID for the instance of SQL Server. Instance ID becomes part of the installation path.

Setup Support Rules
Feature Selection
Installation Rules
Instance Configuration
Disk Space Requirements
Server Configuration
Database Engine Configuration
Error Reporting
Installation Configuration Rules
Installation Progress
Complete

○ Default instance

● Named instance: SQLExpress

Instance ID: SQLEXPRESS

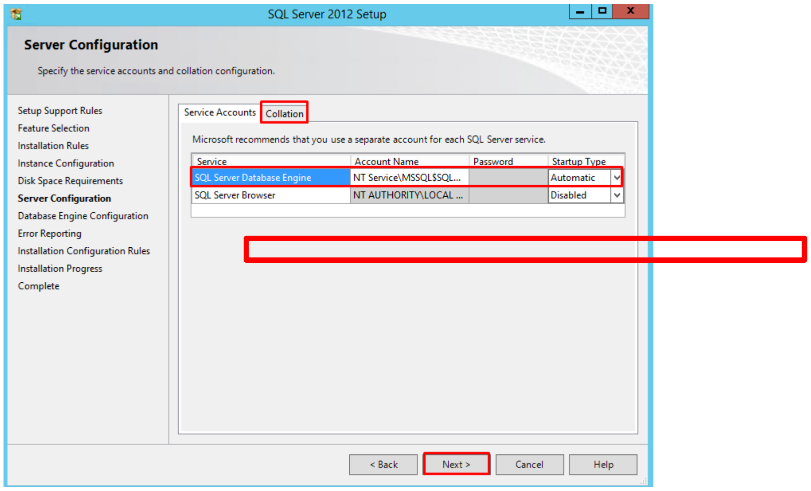
Instance root directory: C:\Program Files\Microsoft SQL Server\

SQL Server directory: C:\Program Files\Microsoft SQL Server\MSSQL11.SQLEXPRESS

Installed instances:

Instance Name	Instance ID	Features	Edition	Version

The next step is server configuration, change the Startup Type for SQL Server Database Engine and SQL Server Browser into Automatic, then click on Next.



SQL Server 2012 Setup

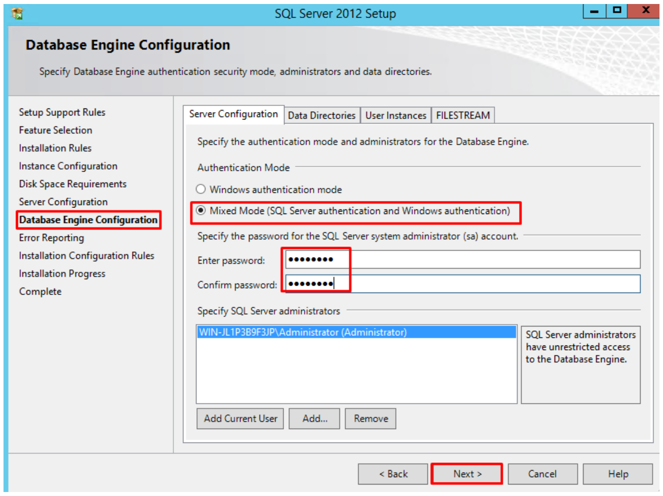
Server Configuration

Specify the service accounts and collation configuration.

Microsoft recommends that you use a separate account for each SQL Server service.

Service	Account Name	Password	Startup Type
SQL Server Database Engine	NT Service\MSSQL$SQL...		Automatic
SQL Server Browser	NT AUTHORITY\LOCAL ...		Disabled

The next step is Database Engine configuration, on Authentication Mode, choose Mixed Mode and input our default password that we used.
Password: 1234567890
Then click on Next.



SQL Server 2012 Setup

Database Engine Configuration

Specify Database Engine authentication security mode, administrators and data directories.

Specify the authentication mode and administrators for the Database Engine.

Authentication Mode

○ Windows authentication mode

● Mixed Mode (SQL Server authentication and Windows authentication)

Specify the password for the SQL Server system administrator (sa) account.

Enter password: ••••••••

Confirm password: ••••••••

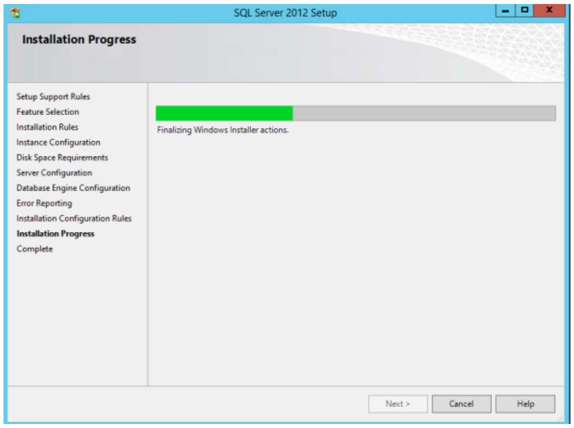
Specify SQL Server administrators

WIN-JL1P3B9F3JP\Administrator (Administrator)

SQL Server administrators have unrestricted access to the Database Engine.

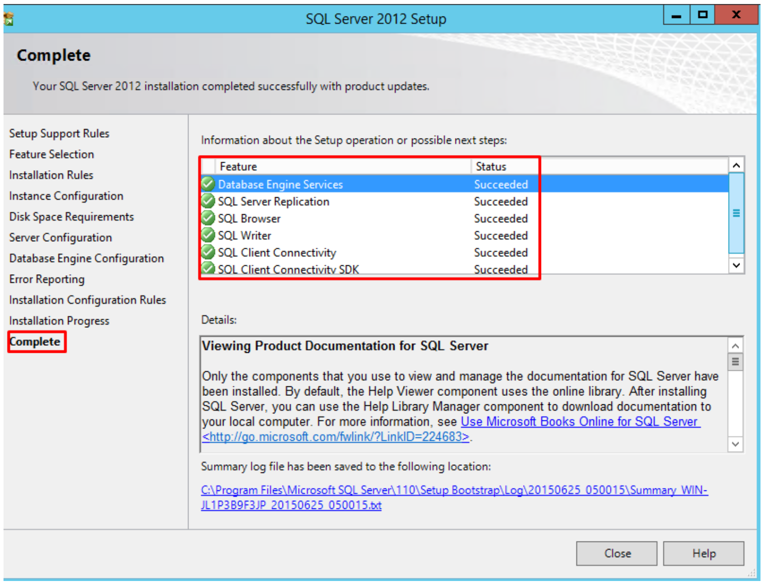
The next screen is Error Reporting screen. Leave it as it is. Just click on Next.

The following screen is where SQL server will install on your computer. This could take a while to complete depending on the server specification you are using.



Finalizing Windows Installer actions.

Once the installation has completed, you will see a screen showing the details of what was completed and if there were any problems occurred.



Your SQL Server 2012 installation completed successfully with product updates.

Information about the Setup operation or possible next steps:

Feature	Status
Database Engine Services	Succeeded
SQL Server Replication	Succeeded
SQL Browser	Succeeded
SQL Writer	Succeeded
SQL Client Connectivity	Succeeded
SQL Client Connectivity SDK	Succeeded

Details:

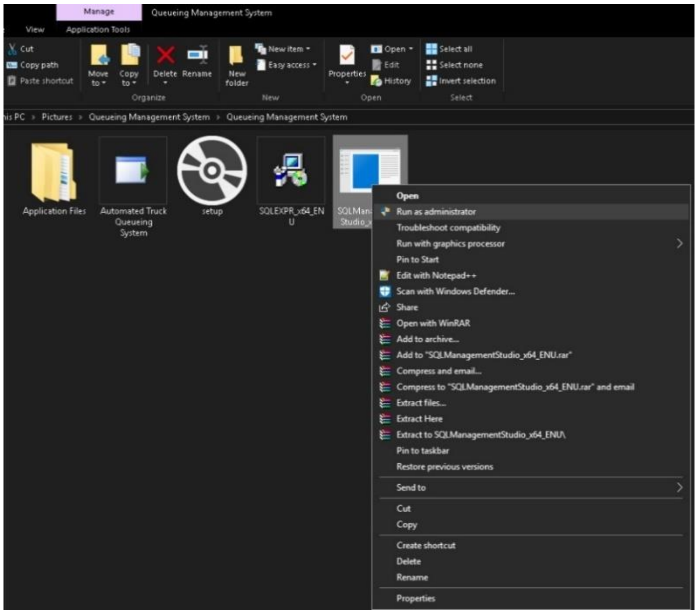
Viewing Product Documentation for SQL Server

Only the components that you use to view and manage the documentation for SQL Server have been installed. By default, the Help Viewer component uses the online library. After installing SQL Server, you can use the Help Library Manager component to download documentation to your local computer. For more information, see Use Microsoft Books Online for SQL Server <http://go.microsoft.com/fwlink/?LinkID=224683>.

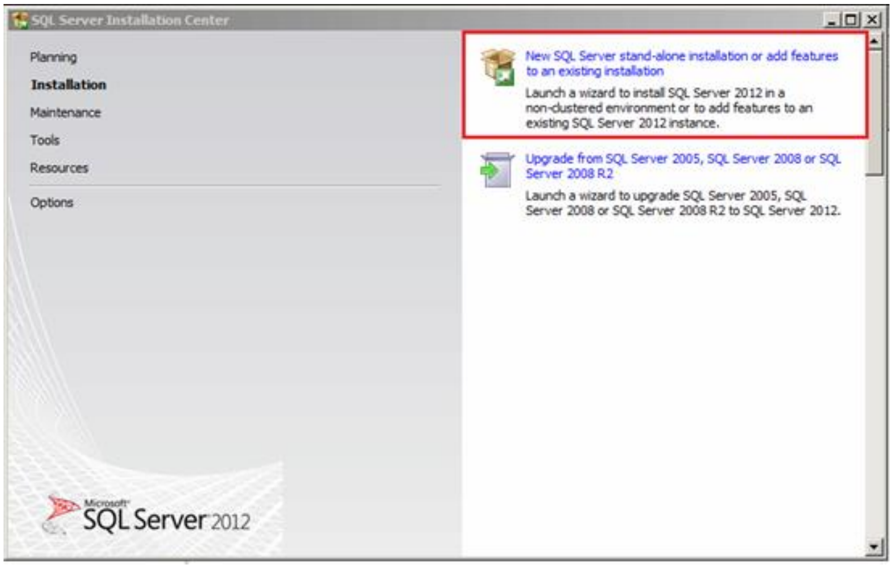
Summary log file has been saved to the following location:

C:\Program Files\Microsoft SQL Server\110\Setup Bootstrap\Log\20150625_050015\Summary_WIN-JL1P3B9F3JP_20150625_050015.txt

Inside the Queueing Management System Folder, Run SQLManagementStudio_x64_ENU.exe as administrator



On the next screen choose the top option to install a new stand-alone installation.



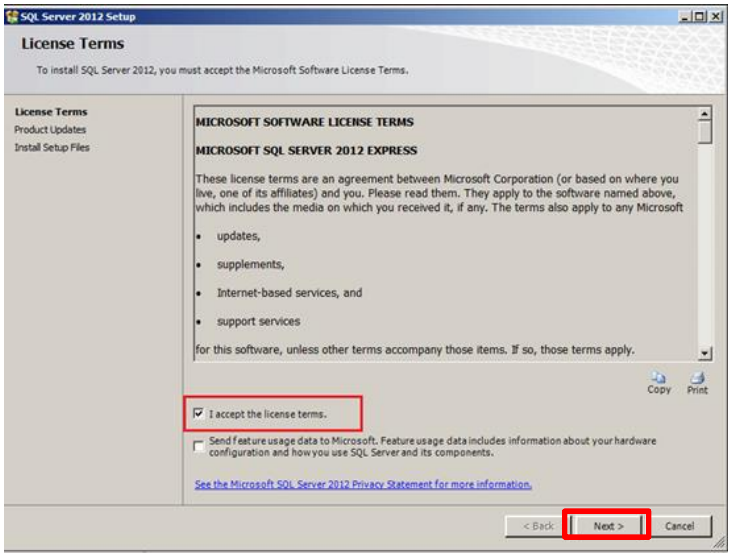
New SQL Server stand-alone installation or add features to an existing installation

Launch a wizard to install SQL Server 2012 in a non-clustered environment or to add features to an existing SQL Server 2012 instance.

Upgrade from SQL Server 2005, SQL Server 2008 or SQL Server 2008 R2

Launch a wizard to upgrade SQL Server 2005, SQL Server 2008 or SQL Server 2008 R2 to SQL Server 2012.

The set up will process the basic operations and you will see the licensing screen where you have to click the check box for “I accept the license terms”, then click next.



SQL Server 2012 Setup

License Terms

To install SQL Server 2012, you must accept the Microsoft Software License Terms.

MICROSOFT SOFTWARE LICENSE TERMS

MICROSOFT SQL SERVER 2012 EXPRESS

These license terms are an agreement between Microsoft Corporation (or based on where you live, one of its affiliates) and you. Please read them. They apply to the software named above, which includes the media on which you received it, if any. The terms also apply to any Microsoft

- updates,
- supplements,
- Internet-based services, and
- support services

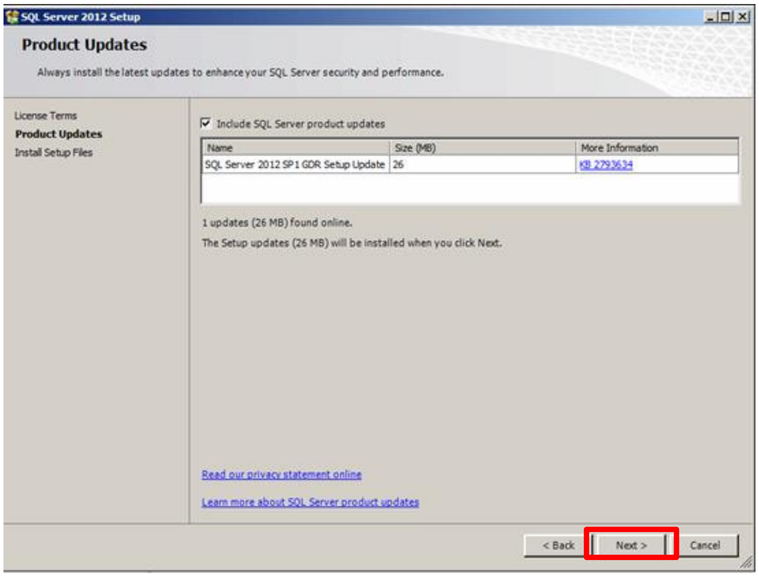
for this software, unless other terms accompany those items. If so, those terms apply.

☑ I accept the license terms.

☐ Send feature usage data to Microsoft. Feature usage data includes information about your hardware configuration and how you use SQL Server and its components.

See the Microsoft SQL Server 2012 Privacy Statement for more information.

Once you accept the license terms, it will scann all the available product updates. Leave it as it is, then click Next.



SQL Server 2012 Setup

Product Updates

Always install the latest updates to enhance your SQL Server security and performance.

☑ Include SQL Server product updates

Name	Size (MB)	More Information
SQL Server 2012 SP1 GDR Setup Update	26	KB 2793634

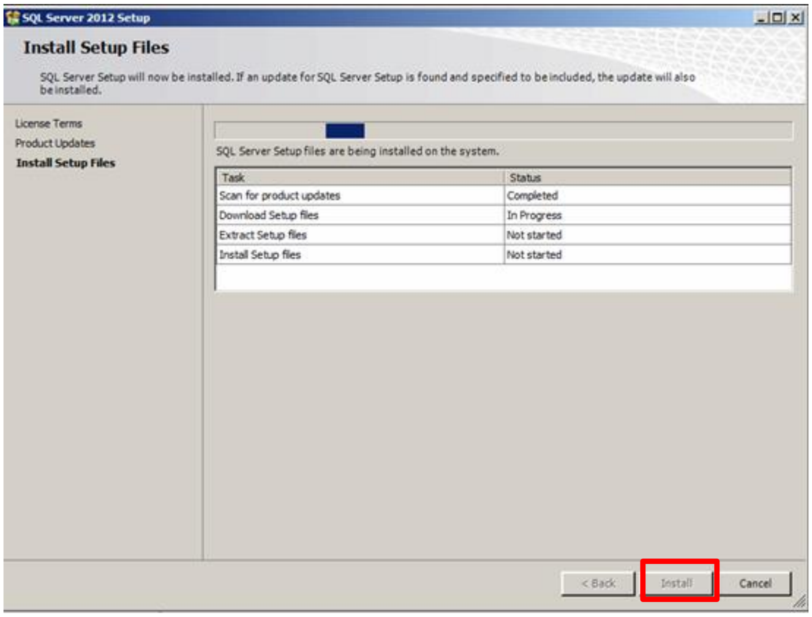
1 updates (26 MB) found online.

The Setup updates (26 MB) will be installed when you click Next.

Read our privacy statement online

Learn more about SQL Server product updates

The next step will Install Setup Files. Once the installation has completed, you will see a screen showing the details of what was completed and if there were any problems occurred.



SQL Server 2012 Setup

Install Setup Files

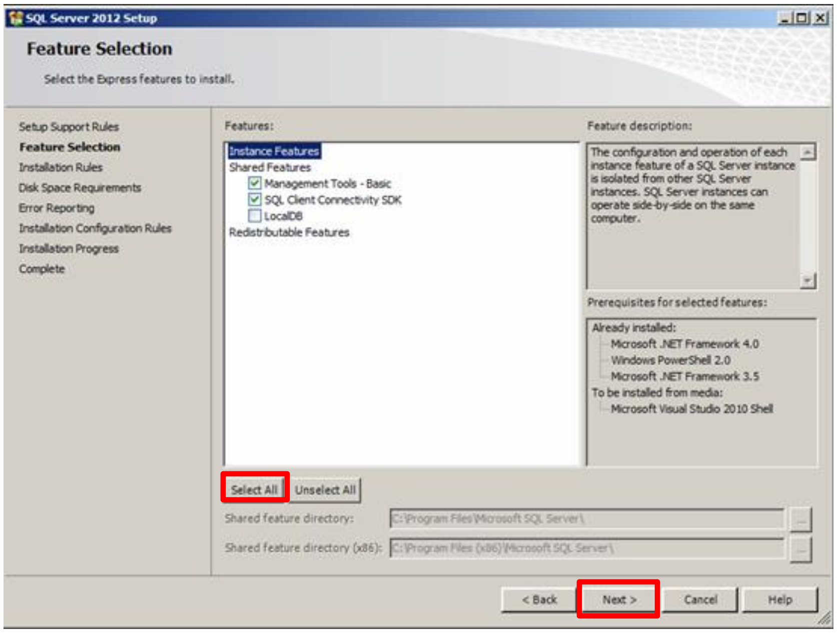
SQL Server Setup will now be installed. If an update for SQL Server Setup is found and specified to be included, the update will also be installed.

License Terms
Product Updates
Install Setup Files

SQL Server Setup files are being installed on the system.

Task	Status
Scan for product updates	Completed
Download Setup files	In Progress
Extract Setup files	Not started
Install Setup files	Not started

On the next screen is Feature Selection, Click Select All to install additional features, then click Next.



SQL Server 2012 Setup

Feature Selection

Select the Express features to install.

Setup Support Rules
Feature Selection
Installation Rules
Disk Space Requirements
Error Reporting
Installation Configuration Rules
Installation Progress
Complete

Features:

Instance Features
Shared Features
☑ Management Tools - Basic
☑ SQL Client Connectivity SDK
□ LocalDB
Redistributable Features

Feature description:

The configuration and operation of each instance feature of a SQL Server instance is isolated from other SQL Server instances. SQL Server instances can operate side-by-side on the same computer.

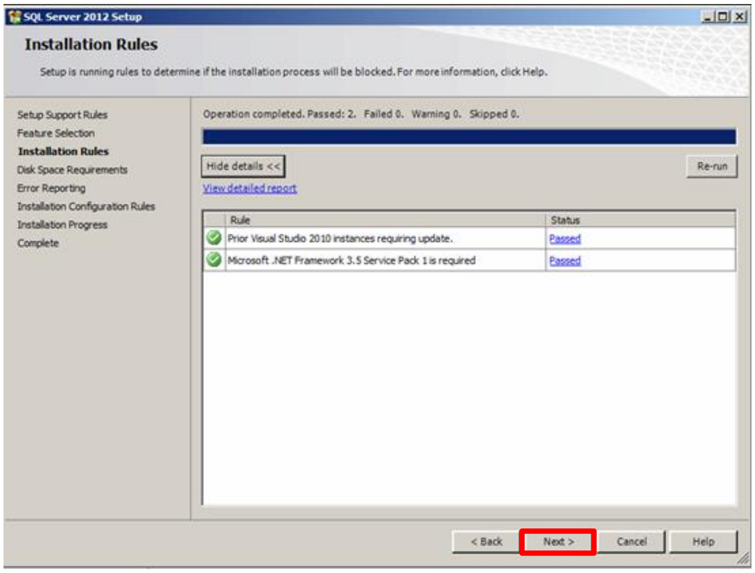
Prerequisites for selected features:

Already installed:
Microsoft .NET Framework 4.0
Windows PowerShell 2.0
Microsoft .NET Framework 3.5
To be installed from media:
Microsoft Visual Studio 2010 Shell

Shared feature directory: C:\Program Files\Microsoft SQL Server\

Shared feature directory (x86): C:\Program Files (x86)\Microsoft SQL Server\

In next step, SQL Server Management Studio setup will check the installation rules, Just leave it as it is, then click on Next.



SQL Server 2012 Setup

Installation Rules

Setup is running rules to determine if the installation process will be blocked. For more information, click Help.

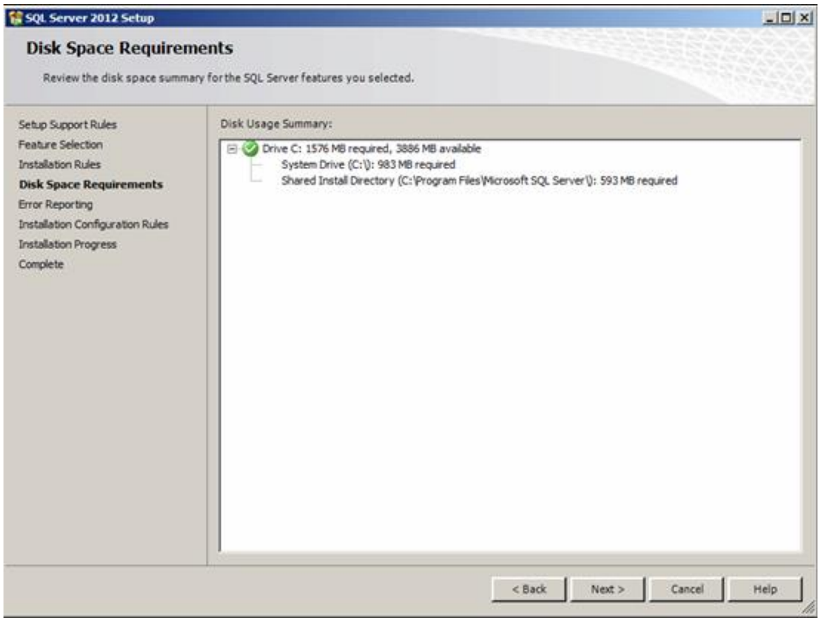
- Setup Support Rules
- Feature Selection
- **Installation Rules**
- Disk Space Requirements
- Error Reporting
- Installation Configuration Rules
- Installation Progress
- Complete

Operation completed. Passed: 2. Failed 0. Warning 0. Skipped 0.

View detailed report

	Rule	Status
✓	Prior Visual Studio 2010 instances requiring update.	Passed
✓	Microsoft .NET Framework 3.5 Service Pack 1 is required	Passed

In this step SQL Server Management Studio setup will verify the disk space, Leave it as it is, then click on Next.



SQL Server 2012 Setup

Disk Space Requirements

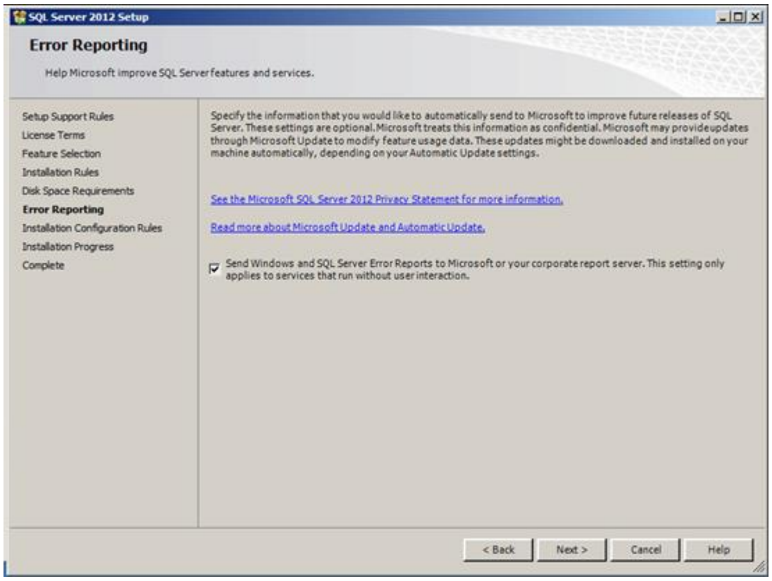
Review the disk space summary for the SQL Server features you selected.

- Setup Support Rules
- Feature Selection
- Installation Rules
- **Disk Space Requirements**
- Error Reporting
- Installation Configuration Rules
- Installation Progress
- Complete

Disk Usage Summary:

- ✓ Drive C: 1576 MB required, 3886 MB available
  - System Drive (C:\): 983 MB required
  - Shared Install Directory (C:\Program Files\Microsoft SQL Server\): 593 MB required

In this Error Reporting screen, you can decide whether or not to send error notification to Microsoft, leave it as it is, then click on Next.



SQL Server 2012 Setup

Error Reporting

Help Microsoft improve SQL Server features and services.

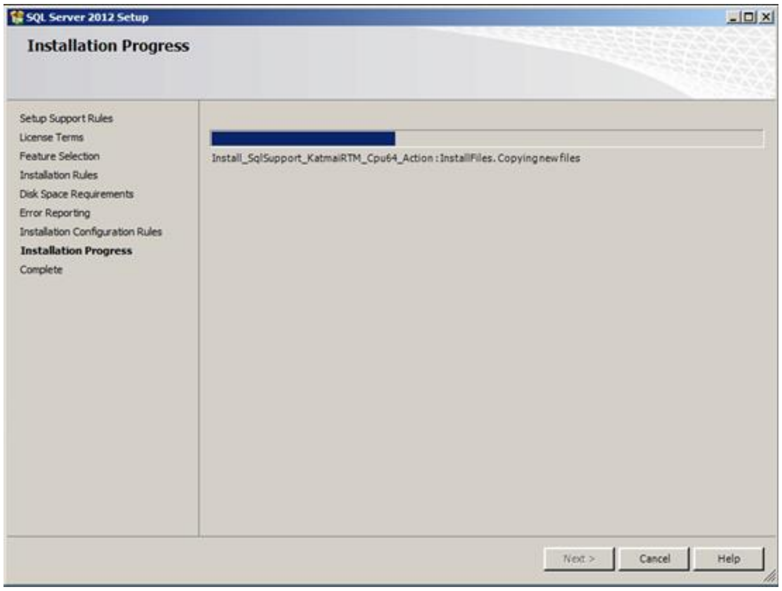
Specify the information that you would like to automatically send to Microsoft to improve future releases of SQL Server. These settings are optional. Microsoft treats this information as confidential. Microsoft may provideupdates through Microsoft Update to modify feature usage data. These updates might be downloaded and installed on your machine automatically, depending on your Automatic Update settings.

See the Microsoft SQL Server 2012 Privacy Statement for more information.

Read more about Microsoft Update and Automatic Update.

☑ Send Windows and SQL Server Error Reports to Microsoft or your corporate report server. This setting only applies to services that run without user interaction.

Once you click on Next, setup checks the instalaltion configuration rules and if it doesn't have errors, setup will continue

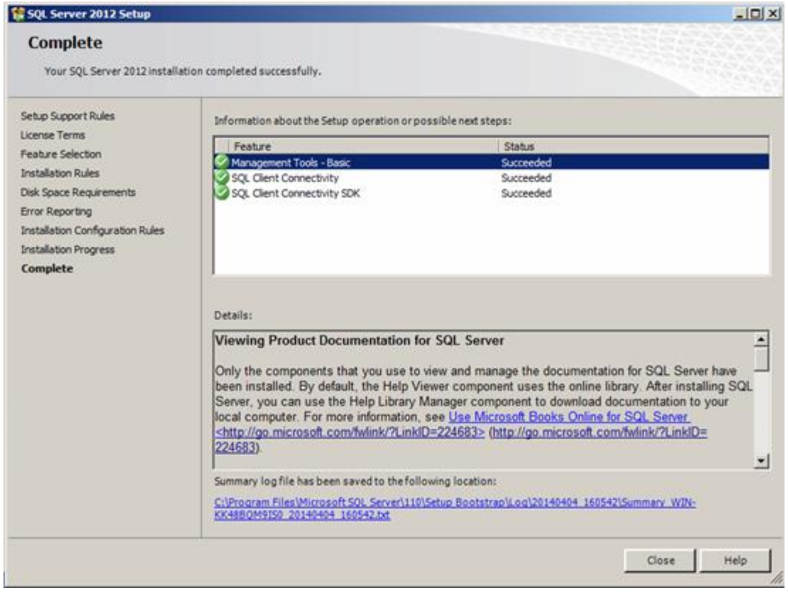


SQL Server 2012 Setup

Installation Progress

Install_SqlSupport_KatmaiRTM_Cpu64_Action : InstallFiles. Copyingnew files

This screen displays the installation status of SQL Server Management Studio along with its features, click on Close to Exit.



Complete

Your SQL Server 2012 installation completed successfully.

Information about the Setup operation or possible next steps:

Feature	Status
Management Tools - Basic	Succeeded
SQL Client Connectivity	Succeeded
SQL Client Connectivity SDK	Succeeded

Details:

Viewing Product Documentation for SQL Server

Only the components that you use to view and manage the documentation for SQL Server have been installed. By default, the Help Viewer component uses the online library. After installing SQL Server, you can use the Help Library Manager component to download documentation to your local computer. For more information, see Use Microsoft Books Online for SQL Server <http://go.microsoft.com/fwlink/?LinkID=224683> (http://go.microsoft.com/fwlink/?LinkID=224683).

Summary log file has been saved to the following location:

C:\Program Files\Microsoft SQL Server\110\Setup Bootstrap\Log\20140404_160542\Summary_WIN-KK48BQM9IS0_20140404_160542.txt

5.2 Install Queueing Management System

Unzip the Queueing Management System.rar file into your desired location.

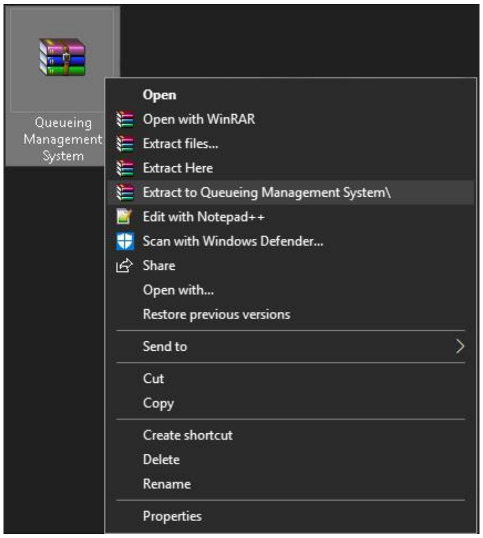


Figure 5-1

Inside the Queueing Management System Folder, Launch setup.exe.

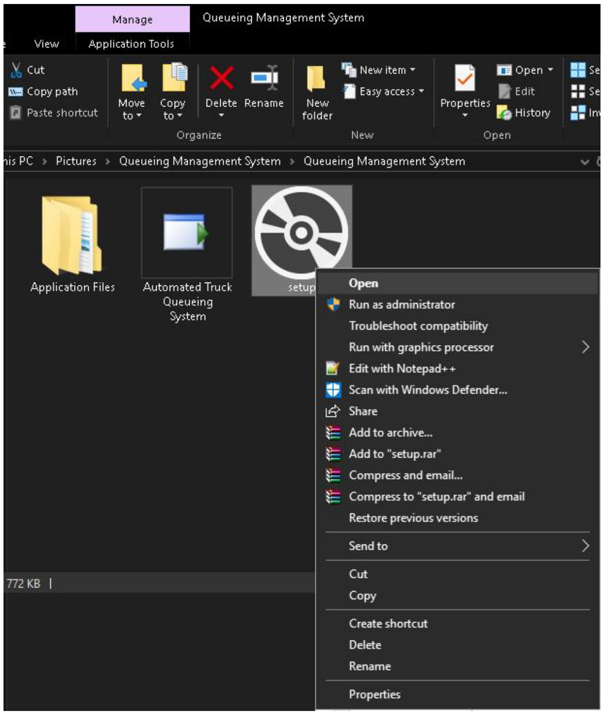
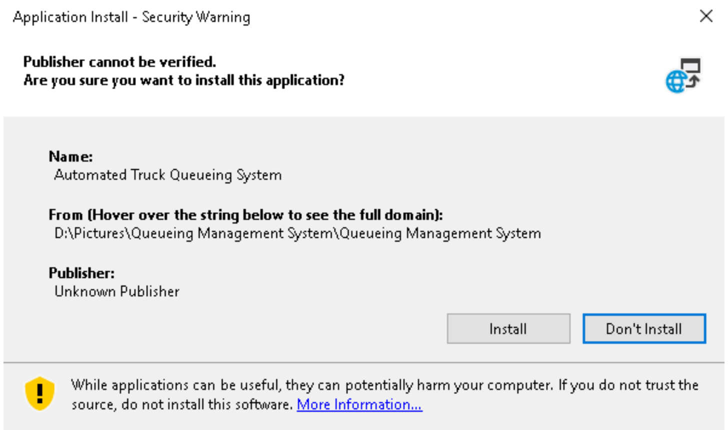


Figure 5-2

If Securiy Warning Prompted, Click Install.



Application Install - Security Warning

Publisher cannot be verified.
Are you sure you want to install this application?

Name:
Automated Truck Queueing System

From (Hover over the string below to see the full domain):
D:\Pictures\Queueing Management System\Queueing Management System

Publisher:
Unknown Publisher

While applications can be useful, they can potentially harm your computer. If you do not trust the source, do not install this software. More Information...

Figure 5-3

Wait until the Login Interface is Launch.

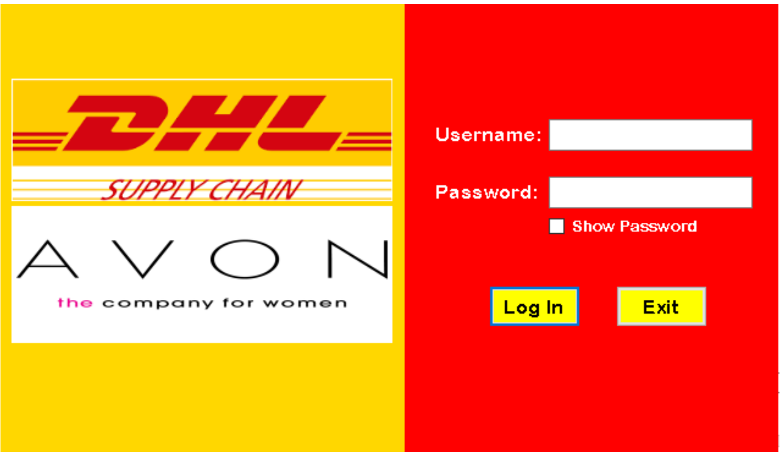


Figure 5-4

Login the Username and Password Given by your administrator.

5.2.1 Configure Admin Settings

Click CTRL+SHIFT+D Keys to launch Admin Authentication Settings.

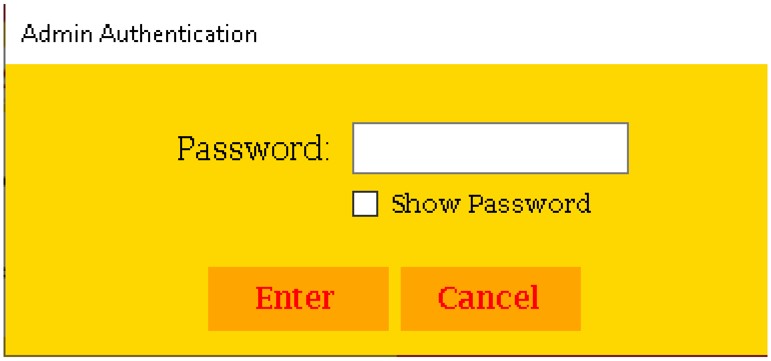


Figure 5-5

In Figure 5-5, the default Password is Blank (“”), so just click Enter.

5.2.1.1 Database Configuration

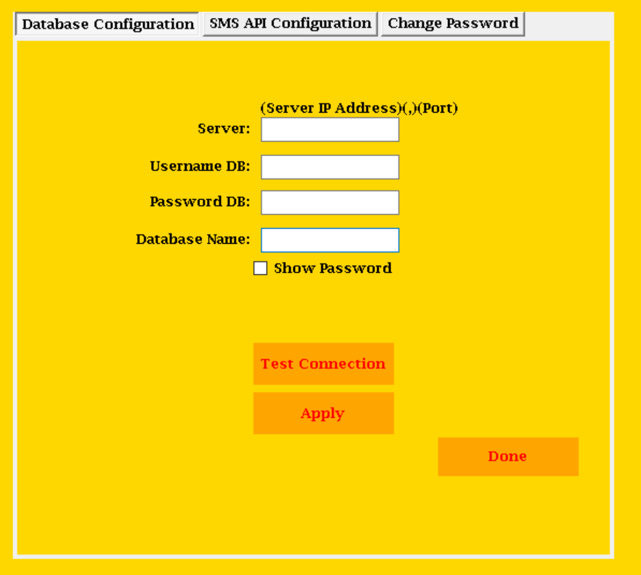
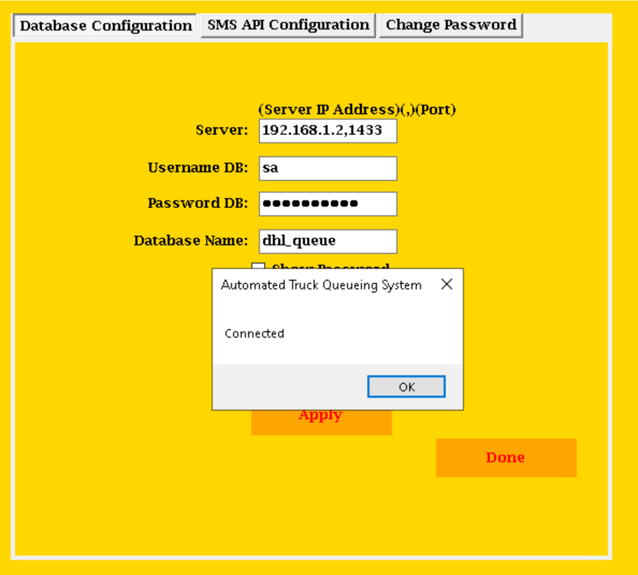


Figure 5-6

Server – Enter Your Servers IP Address + “,” + Servers used
Network Port. (Example: 192.168.1.1,1433)
Note: Please Follow the Desired Format.
Username DB – Enter Servers Database Username
Password DB – Enter Servers Database Password
Database Name – Enter Servers Database Name

Click the Test Connection, and when the Prompt Appears Connected. Click Apply.



(Server IP Address)(,)(Port)

Server: 192.168.1.2,1433

Username DB: sa

Password DB: ●●●●●●●●●●●

Database Name: dhl_queue

□ Show Password

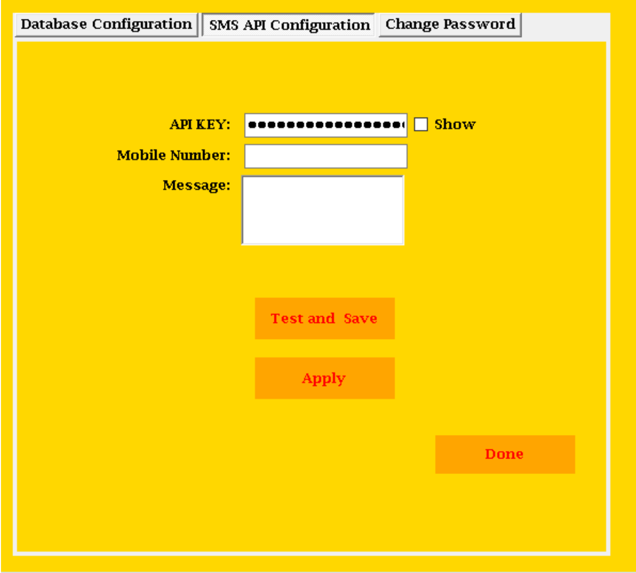
Automated Truck Queueing System

Connected

Figure 5-7

Click Done, and the system reboots and ready to login.

5.2.1.2 SMS API Configuration



API KEY: ●●●●●●●●●●●●●●●●● □ Show

Mobile Number:

Message:

Figure 5-8

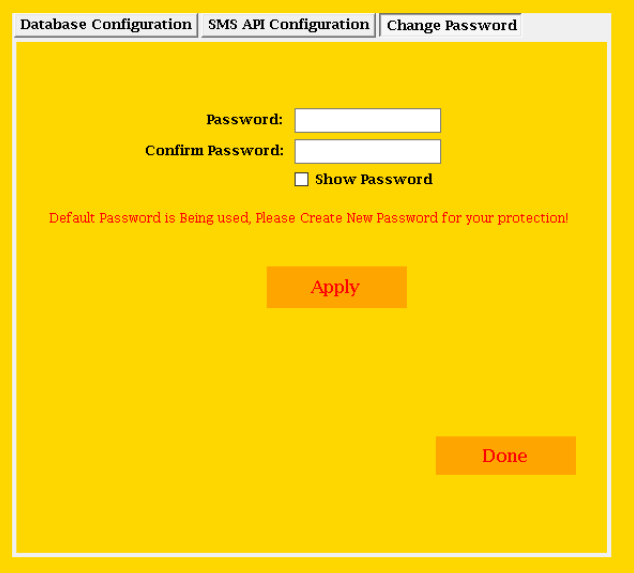
API KEY – Enter Your API Key
For Test Only:
Mobile Number – Enter Your Mobile Number for Testing
Message – Enter any Message for you to Receive for Testing

Click Test and Save, and when the Prompt Appears "Test Message Has been Sent! API code has been saved", the system reboots and ready to use.

Or, Click Apply if you don't want to test your API Key.

Then, Click Done.

5.2.1.3 Admin Settings Password Configuration



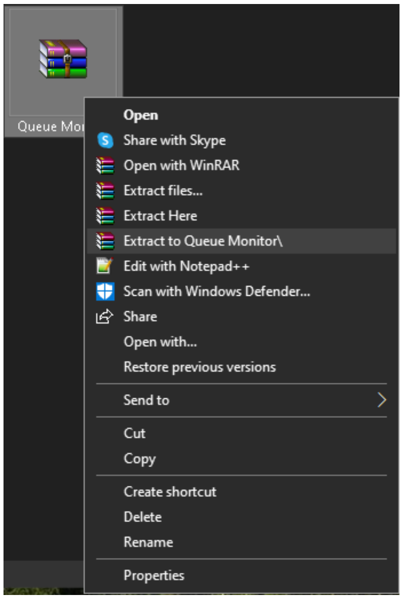
Default Password is Being used, Please Create New Password for your protection!

Figure 5-9

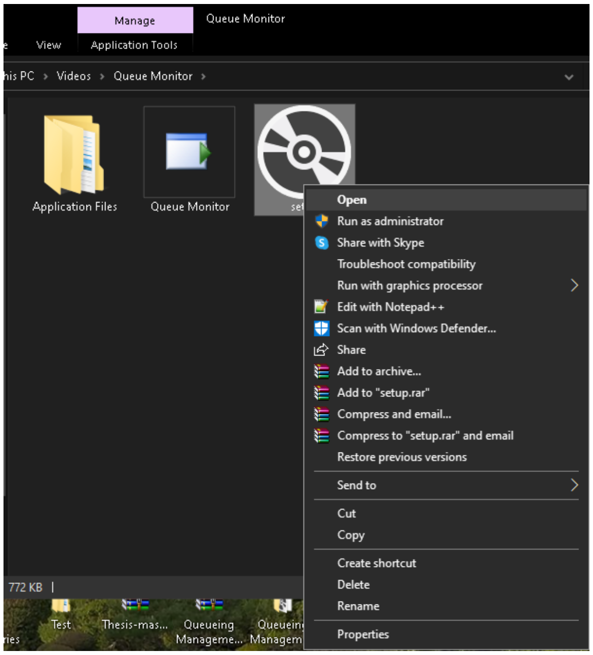
We can now assign New password and click Apply. Where as the system login will restart and can use the new Entered Password.

5.3 Install Queueing Monitor System

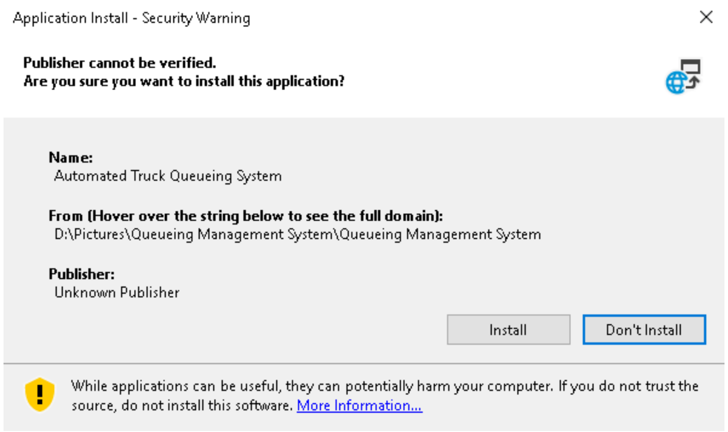
Unzip the Queue Monitor.rar file into your desired location.



Inside the Queue Monitor Folder, Launch setup.exe.



If Securiy Warning Prompted, Click Install.



Application Install - Security Warning

Publisher cannot be verified.
Are you sure you want to install this application?

Name:
Automated Truck Queueing System

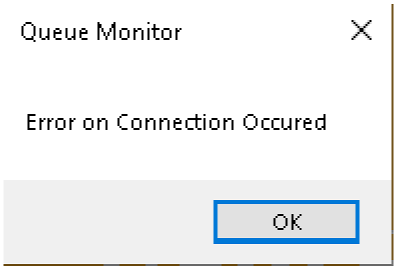
From (Hover over the string below to see the full domain):
D:\Pictures\Queueing Management System\Queueing Management System

Publisher:
Unknown Publisher

While applications can be useful, they can potentially harm your computer. If you do not trust the source, do not install this software. More Information...

Wait until the Login Interface is Launch.

5.3.1 Configure Queue Monitor Admin Settings



Queue Monitor

Error on Connection Occured

Figure 5-10

When the Figure 5-13 prompt, Just Click OK, It’s normal for new Installed Queue Monitor because its not connected on its Database.

Or click CTRL+SHIFT+D Keys to launch Admin Authentication Settings.

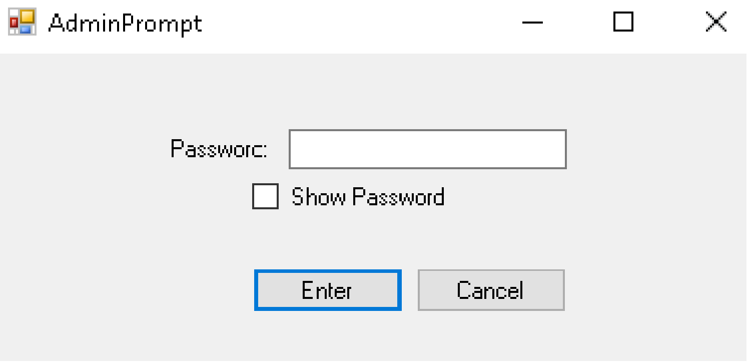


Figure 5-11

In Figure 5-14, the default Password is Blank (“”), so just click Enter.

5.3.1.1 Database Configuration

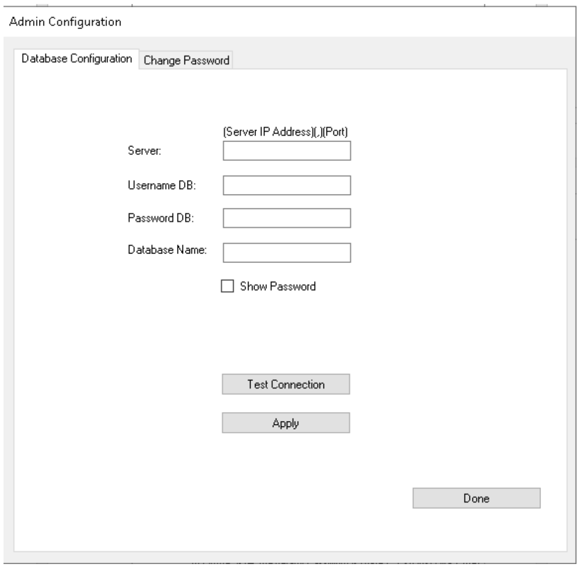
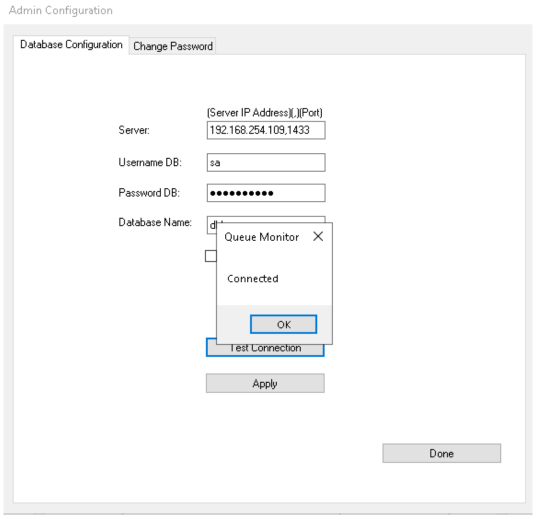


Figure 5-12

Server – Enter Your Servers IP Address + “,” + Servers used
Network Port. (Example: 192.168.1.1,1433)
Note: Please Follow the Desired Format.
Username DB – Enter Servers Database Username

Password DB – Enter Servers Database Password
Database Name – Enter Servers Database Name

Click the Test Connection, and when the Prompt Appears Connected. Click Apply.



(Server IP Address)(,)(Port)
Server: 192.168.254.109,1433
Username DB: sa
Password DB: •••••••••••
Database Name: db

Queue Monitor

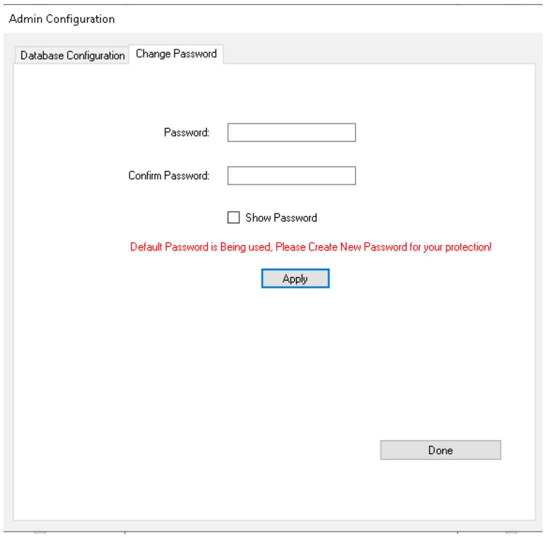
Connected

Figure 5-13

Click Done, and the system reboots.

5.4.1.2 Admin Settings Password Configuration

Figure 5-14



Admin Configuration

Password:

Confirm Password:

□ Show Password

Default Password is Being used, Please Create New Password for your protection!

We can now assign New password and click Apply. Where as the system login will restart and can use the new Entered Password.

6 User Interface and Functions

6.1 Login

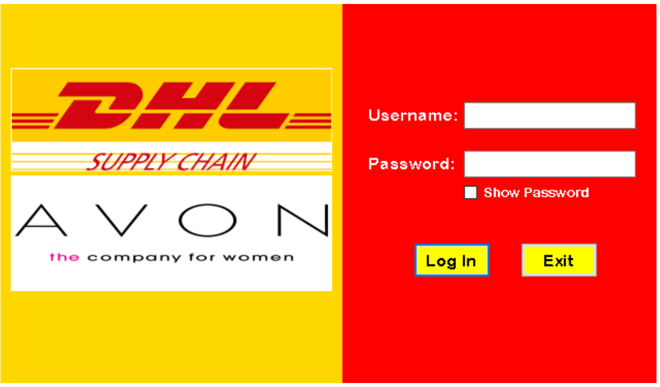
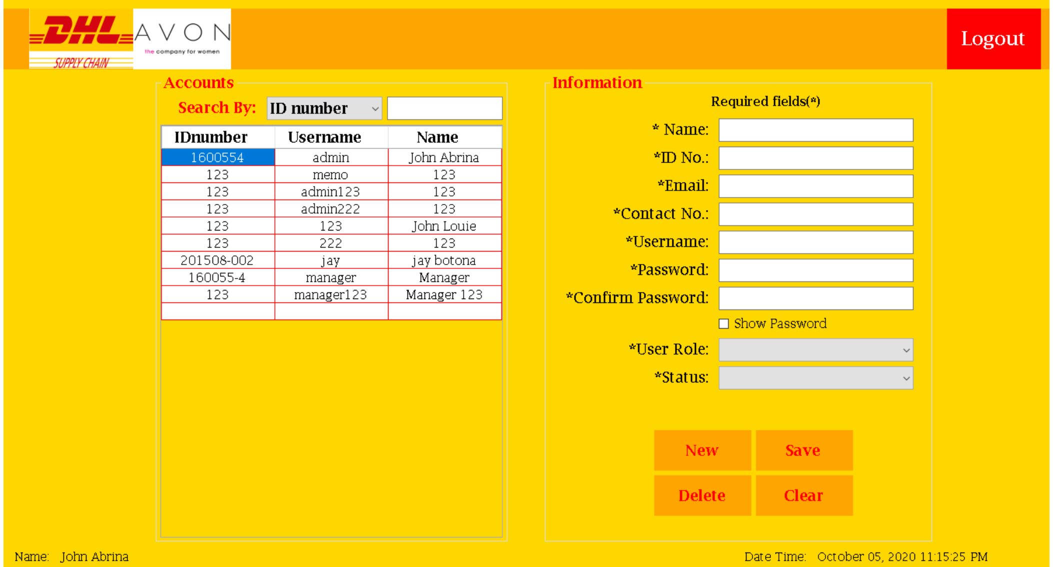


Figure 6-1

6.1.1 Enter Username
6.1.2 Enter Password
6.1.3 Tick Show Password Check Box to show Password Characters if needed.
6.1.4 Click Log In to continue.
6.1.5 Click Exit to close the application.

6.2 Admin Account Interface



Search By: ID number

IDnumber	Username	Name
1600554	admin	John Abrina
123	memo	123
123	admin123	123
123	admin222	123
123	123	John Louie
123	222	123
201508-002	jay	jay botona
160055-4	manager	Manager
123	manager123	Manager 123

Required fields(*)

* Name:
*ID No.:
*Email:
*Contact No.:
*Username:
*Password:
*Confirm Password:

□ Show Password

*User Role:
*Status:

Name: John Abrina

Date Time: October 05, 2020 11:15:25 PM

Figure 6-2

Method 1: Add New User Account:

1.1 Click New
1.2 Fill Up the ff:
  1.2.1 Name
  1.2.2 ID No.
  1.2.3 Email
  1.2.4 Contact No.
  1.2.5 Username
  1.2.6 Password
  1.2.7 Confirm Password

Tick Show Password Check Box to show Password Characters if needed.

1.3 Choose from User Role Dropdown Menu
1.4 Choose from Status Dropdown Menu
1.5 Click Save
1.6 When prompt appears says “Are you sure you want to Create this New Accounts”, Click Yes.

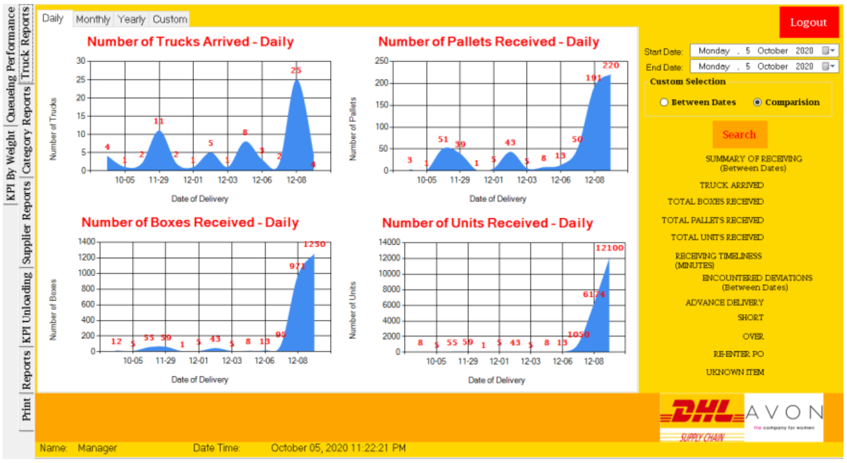
Method 2: Edit User Account:

2.1. Highlight your selected row
2.2. Edit you desired data on required filed
2.3. Click Save

Method 3: Delete User Account:

3.1. Highlight your selected row
3.2. Click Delete

6.3 Manager Account Interface



Start Date: Monday , 5 October 2020
End Date: Monday , 5 October 2020
Custom Selection: ○ Between Dates ● Comparision

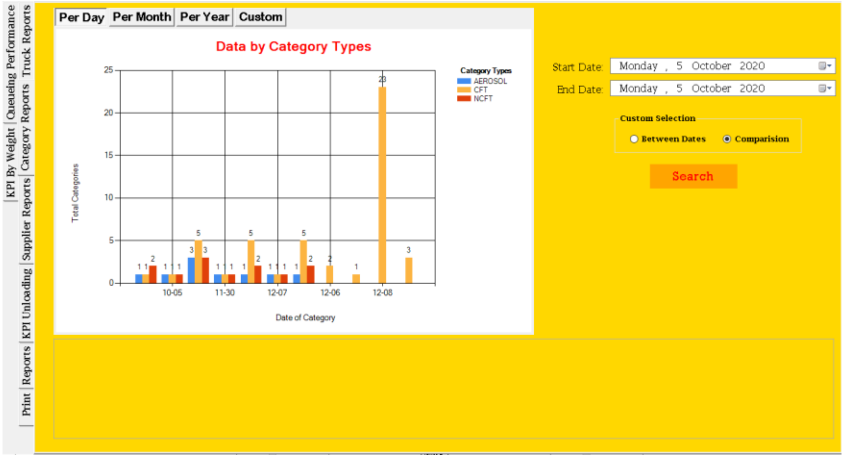
Name: Manager
Date Time: October 05, 2020 11:22:21 PM

Figure 6-3
(Truck Reports)

6.3.1. Displays Daily Truck Reports
6.3.1.1 Number of Trucks Arrived – Daily, Monthly, Yearly
6.3.1.2 Number of Boxes Received – Daily, Monthly, Yearly
6.3.1.3 Number of Pallets Received – Daily, Monthly, Yearly
6.3.1.4 Number of Units Received – Daily, Monthly, Yearly
6.3.2. Display Daily Truck Reports Custom Selection
6.3.2.1 Start Date Selection
6.3.2.2 End Date Selection
6.3.2.3 Between Dates or Comparison Selection
6.3.3. Display Summary of Receiving (Between Dates)
6.3.3.1 Truck Arrived
6.3.3.2 Total Boxes Received

6.3.3.3 Total Pallets Received
6.3.3.4 Total Units Received
6.3.3.5 Receiving Timeliness in Minutes

6.3.4. Display Encounted Deviations (Between Dates)
6.3.4.1 Advance Delivery
6.3.4.2 Short
6.3.4.3 Over
6.3.4.4 Re-Enter Purchase Order
6.3.4.5 Unknown Item



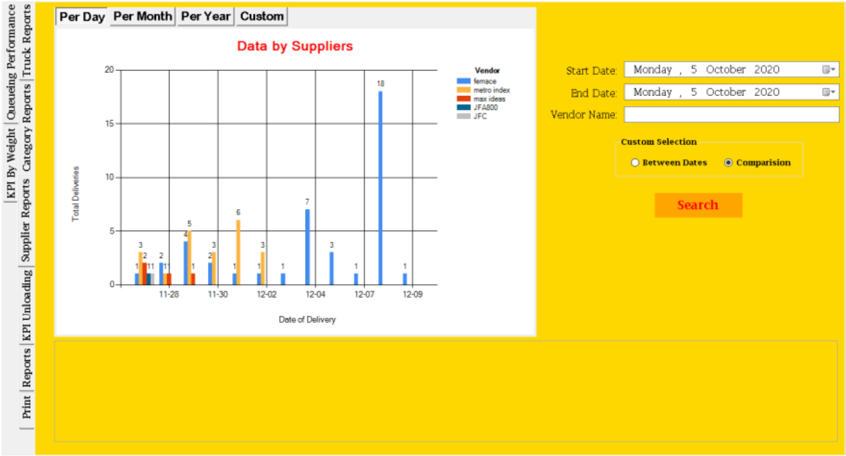
Start Date: Monday , 5 October 2020
End Date: Monday , 5 October 2020

Custom Selection
○ Between Dates ● Comparision

Figure 6-4
(Category Reports)

6.3.5. Displays Data by Category Types - Daily, Monthly, Yearly

6.3.6. Displays Data by Category Types Custom Selection
6.3.6.1 Start Date Selection
6.3.6.2 End Date Selection
6.3.6.3 Between Dates or Comparison Selection



Start Date: Monday , 5 October 2020

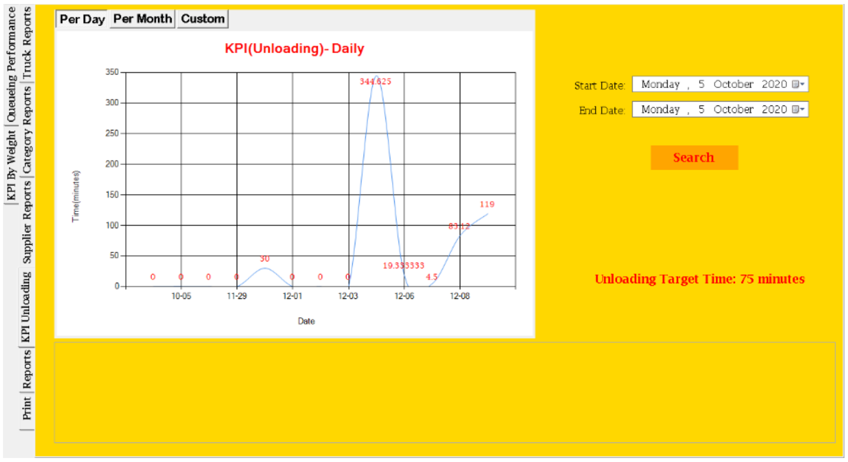
End Date: Monday , 5 October 2020

Vendor Name:

○ Between Dates ● Comparision

Figure 6-5
(Supplier Reports)

6.3.7. Displays Data by Supplier - Daily, Monthly, Yearly
6.3.8. Displays Data by Supplier Custom Selection
6.3.8.1 Start Date Selection
6.3.8.2 End Date Selection
6.3.8.3 Between Dates or Comparison Selection



Start Date: Monday , 5 October 2020

End Date: Monday , 5 October 2020

Unloading Target Time: 75 minutes

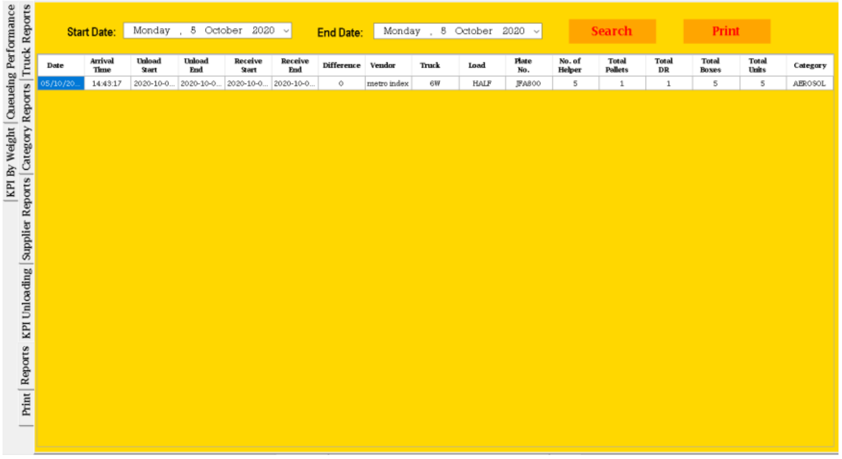
Figure 6-6
(KPI Unloading)

6.3.9 Displays Average KPI(Key Performance Indicator) – Daily, Monthly, Yearly

6.3.10 Displays Average KPI(Key Performance Indicator) Custom Selection

6.3.10.1 Start Date Selection

6.3.10.2 End Date Selection



Start Date: Monday , 5 October 2020

End Date: Monday , 5 October 2020

Date	Arrival Time	Unload Start	Unload End	Receive Start	Receive End	Difference	Vendor	Truck	Load	Plate No.	No. of Helper	Total Pallets	Total DR	Total Boxes	Total Units	Category
05/10/20...	14:43:17	2020-10-0...	2020-10-0...	2020-10-0...	2020-10-0...	0	metro index	6W	HALF	JFA800	5	1	1	5	5	AEROSOL

Figure 6-7
(Reports)

6.3.11 Display Reports Custom Selection

6.3.11.1 Start Date

6.3.11.2 End Date

6.3.12 Display Table Reports

6.3.12.1 Date Arrived

6.3.12.2 Arrival Time

6.3.12.3 Unload Date & Time Start

6.3.12.4 Unload Date & Time End

6.3.12.5 Receive Date & Time Start

6.3.12.6 Receive Date & Time End

6.3.12.7 Unload to Receive Time Difference

6.3.12.8 Vender Name

6.3.12.9 Truck Type

6.3.12.10 Truck Load

6.3.12.11 Truck Plate No.

6.3.12.12 No. of Helper

6.3.12.13 Total Pallets Used

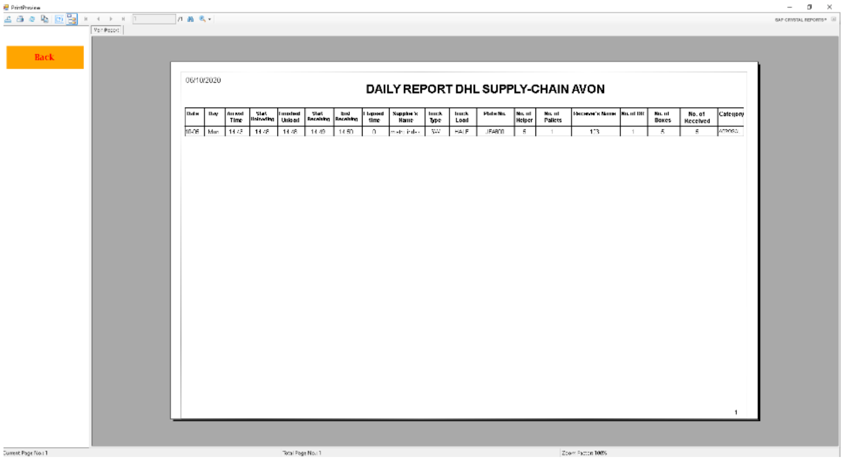
6.3.12.14 Total Delivery Receipts

6.3.12.15 Total Boxes

6.3.12.16 Total Units

6.3.12.17 Delivery Category

6.3.13 Print Reports



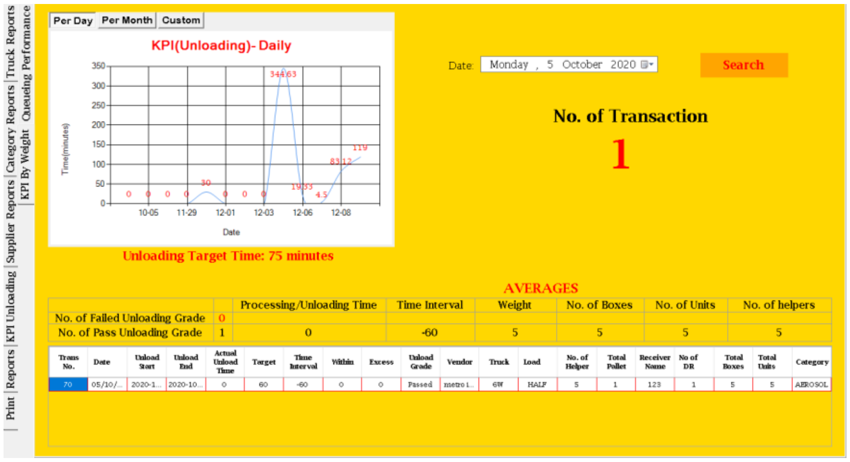
06/10/2020

DAILY REPORT DHL SUPPLY-CHAIN AVON

Date	Day	Arrived Time	Start Unloading	Finished Unload	Start Receiving	End Receiving	Elapsed time	Supplier's Name	Truck Type	Truck Load	Plate No.	No. of Helper	No. of Pallets	Receiver's Name	No. of DR	No. of Boxes	No. of Received	Category
10-05	Mon	14:45	14:46	14:48	14:49	14:50	0	metro index	6W	HALF	JFA800	5	1	123	1	5	5	AEROSOL

1

Figure 6-8
(Print)



Unloading Target Time: 75 minutes

No. of Transaction

1

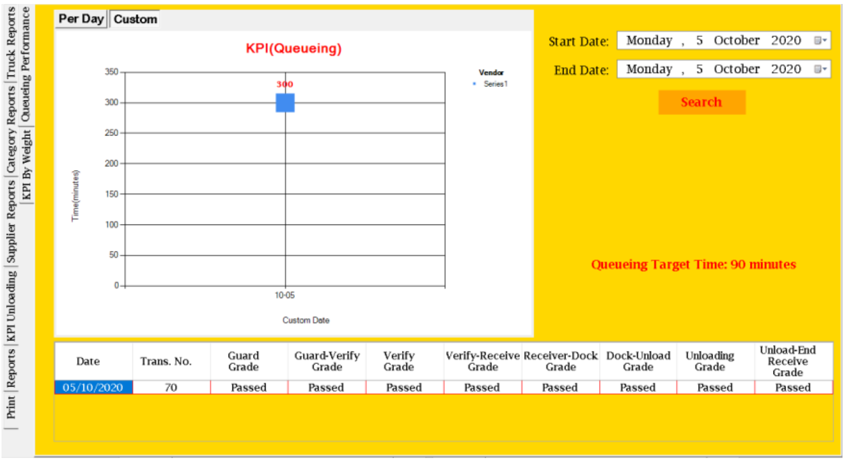
AVERAGES

		Processing/Unloading Time	Time Interval	Weight	No. of Boxes	No. of Units	No. of helpers
No. of Failed Unloading Grade	0						
No. of Pass Unloading Grade	1	0	-60	5	5	5	5

Trans No.	Date	Unload Start	Unload End	Actual Unload Time	Target	Time Interval	Within	Excess	Unload Grade	Vendor	Truck	Load	No. of Helper	Total Pallet	Receiver Name	No of DR	Total Boxes	Total Units	Category
70	05/10/...	2020-1...	2020-10...	0	60	-60	0	0	Passed	metro i...	6W	HALF	5	1	123	1	5	5	AEROSOL

Figure 6-9
(KPI by Weight)

6.3.14 Date Selection of Daily Averages
6.3.15 Display No. of Failed Unloading Grade
6.3.16 Display No. of Pass Unloading Grade
6.3.17 Averages Table
    6.3.17.1 Average Processing/Unloading Time
    6.3.17.2 Average Time Interval
    6.3.17.3 Average Weight
    6.3.17.4 Average No. of Boxes
    6.3.17.5 Average No. of Units
    6.3.17.6 Average No. of Helpers



Start Date: Monday , 5 October 2020

End Date: Monday , 5 October 2020

Queueing Target Time: 90 minutes

Date	Trans. No.	Guard Grade	Guard-Verify Grade	Verify Grade	Verify-Receive Grade	Receiver-Dock Grade	Dock-Unload Grade	Unloading Grade	Unload-End Receive Grade
05/10/2020	70	Passed	Passed	Passed	Passed	Passed	Passed	Passed	Passed

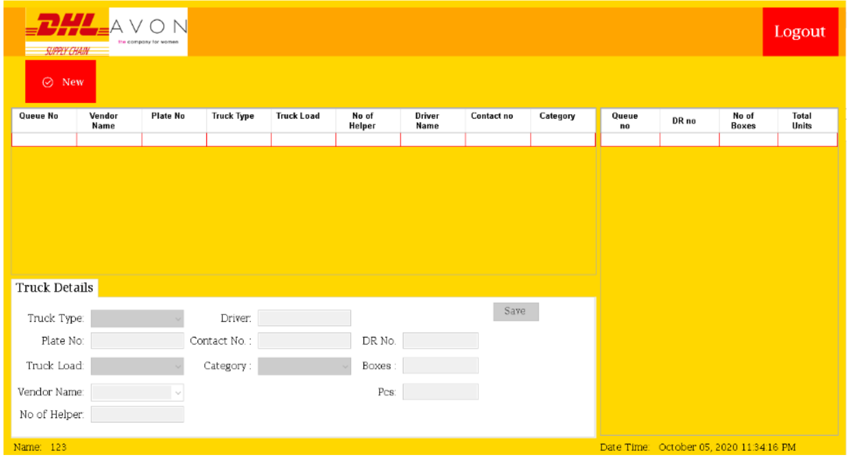
Figure 6-10
(Queueing Performance)

6.3.18 Start Date Selection
6.3.19 End Date Selection
6.3.20 Queueing Target Time Table
    6.3.20.1 Queue Date
    6.3.20.2 Transaction No.
    6.3.20.3 Guard Grade
    6.3.20.4 Guard-Verify Grade
    6.3.20.5 Verify Grade
    6.3.20.6 Verify-Receive Grade
    6.3.20.4 Receive-Dock Grade
    6.3.20.5 Dock-Unlock Grade

6.3.20.6 Unloading Grade

6.3.20.7 Unload-End Receive Grade

6.4 Security Guard Account Interface



Queue No	Vendor Name	Plate No	Truck Type	Truck Load	No of Helper	Driver Name	Contact no	Category

Queue no	DR no	No of Boxes	Total Units

Truck Details

Truck Type:
Plate No:
Truck Load:
Vendor Name:
No of Helper:
Driver:
Contact No. :
Category :
DR No.
Boxes :
Pcs:

Name: 123

Date Time: October 05, 2020 11:34:16 PM

Figure 6-11

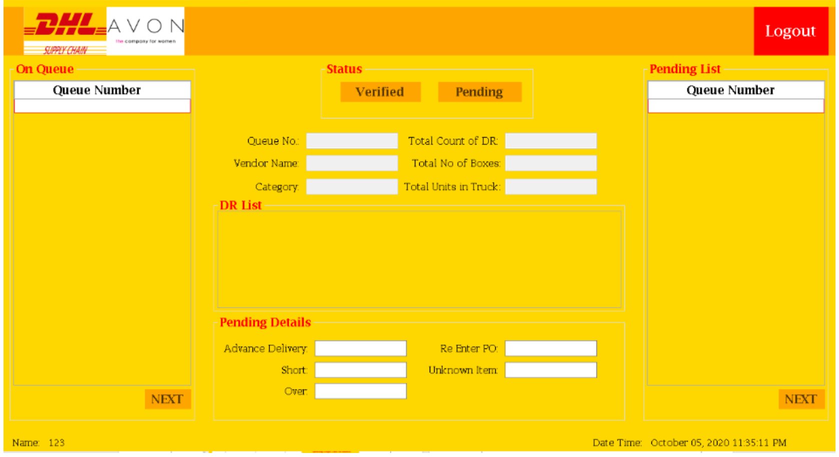
Method 1: Add New Truck

1. Click New
2. Select Truck Type
3. Enter Plate No.
4. Select Truck Load
5. Select or Enter Vendor Name(if not in the list)
6. Enter No. of Helper
7. Enter Driver Name
8. Enter Driver or Helper Contact No.
9. Select Delivery Category
10. Click Save
11. Enter Delivery Receipt No.
12. Enter Number of Boxes indicated in each Delivery Receipt
13. Enter Number of Pcs. Indicadted in each Delivery Receipt
14. Click Add and Repeat 11-13, if there’s more Delivery Receipt
15. Click Done

Method 2: Edit Truck Details

1. Click on the Table row selection
2. Edit on Text box under Truck Details Field
3. Click Save
4. Click Done

6.5 Receiving Associate Accont Interface



Name: 123

Date Time: October 05, 2020 11:35:11 PM

Figure 6-12

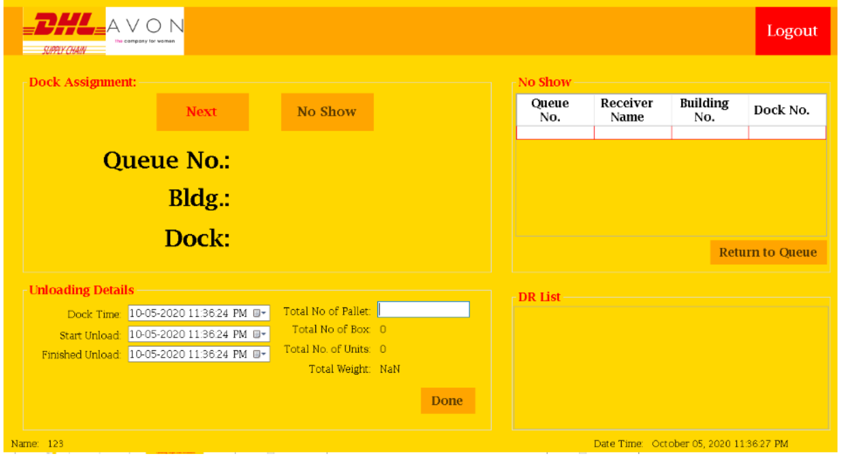
Method 1: Verify On Queue Numbers

1. Click Next under On Queue Table
2. If Details are all Correct, Click Verified
3. If Details are incorrect, Enter the Pending Details.
    - 3.1. Advance Delivery
    - 3.2. Short
    - 3.3. Over
    - 3.4. Re Enter Purchase Order
    - 3.5. Unknown Item

Method 2: Verify Pending List Number

1. Click Next under Pending List Table
2. If Details are now all Correct, Click Verified

6.6 Receiver Account Interface



Dock Assignment:

Queue No.:

Bldg.:

Dock:

No Show

Queue No.	Receiver Name	Building No.	Dock No.

Unloading Details

Dock Time: 10-05-2020 11:36:24 PM

Start Unload: 10-05-2020 11:36:24 PM

Finished Unload: 10-05-2020 11:36:24 PM

Total No of Pallet:

Total No of Box: 0

Total No. of Units: 0

Total Weight: NaN

DR List

Name: 123

Date Time: October 05, 2020 11:36:27 PM

Figure 6-13

Method 1: Verify Received Items

1. Click Next
2. Go to Designated Blgd. No. And Dock No.
3. If Truck Didn’t appear at Dock, Click No Show
4. Select Dock Date and Time
5. Select Start Unload Date and Time
6. Select DR No. from DR List Table Row
7. Enter No. of Box, No. of Units, Total Weight per DR No.
8. Enter Total No. of Pallet Used
9. Click Done

Method 2: Return to Queue

1. If the Truck now reported at the office, Click Return to Queue under No Show table

6.7 Queueing Monitoring Interface

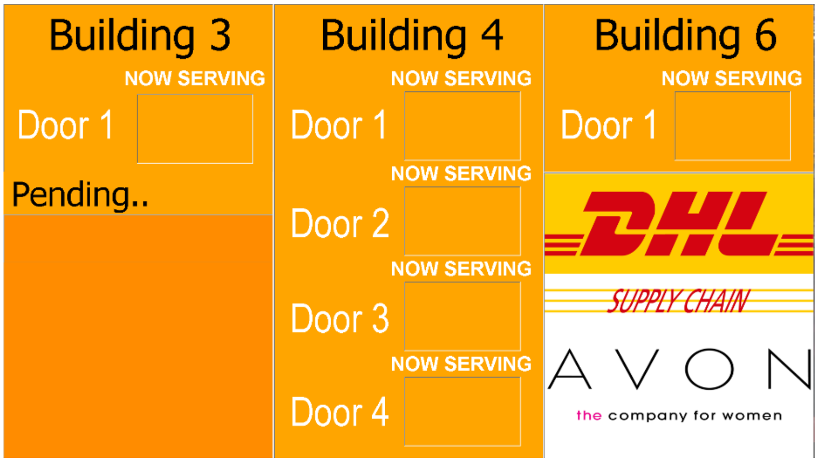
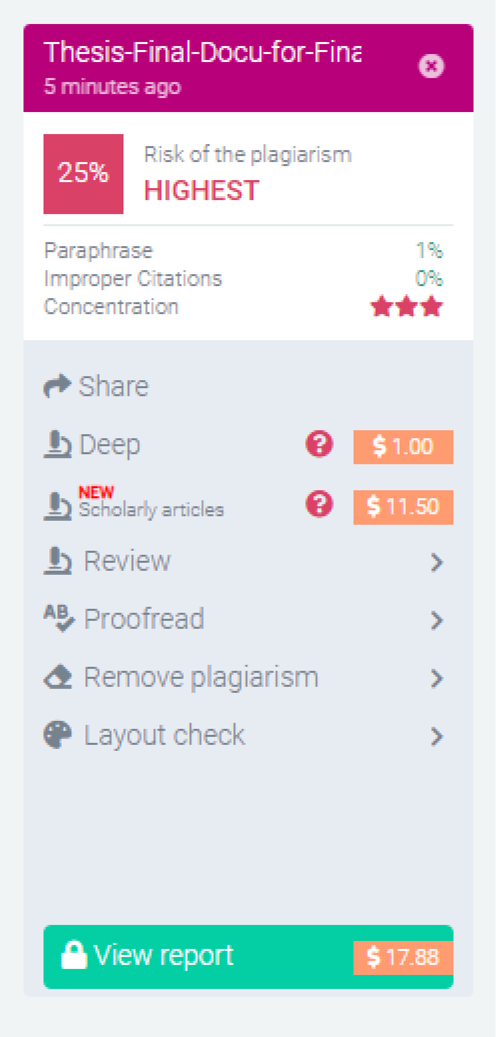


Figure 6-14

PLAGRAMME REPORT



Thesis-Final-Docu-for-Fina
5 minutes ago

25% Risk of the plagiarism
HIGHEST

Paraphrase 1%
Improper Citations 0%
Concentration ★★★

ACM JOURNAL ARTICLE PROGRAM

A queue management system is used to control waiting lines. The queue management is a vital part of the logistic companies daily activities in receiving in delivery of local and imported goods to be stored in the warehouse that will distributed to stores nationwide. An effective queue management system will helps in overall productivity, can reduce cost and thus improve profitability.

The proposed system will help the customer service company in storing data of queue, giving an queue status update thru text message, monitor that will display the who’s trucker is on queue and a dashboard that will show the analytical and graphical reports of queues.

For a company time is money. Any process that saves time and space is considered vital in many operations. Time is very important,that need to be manage efficiently. Wastage of time in a queue is a big problem that needs to be resolved by the proposed system.

To overcome this problem the researchers proposed a new way of handling queue, an Automated Truck Queue

Management System. A system that helps service provider to manage customer in efficient way. The system can ease the customer flow management which is useful for manager of the service provider and analytics provided by a queue management system allows to identify key areas that are in need of improvement.

According to Jhala and Bhathawala (2017), Smart Queue Management System for Banks, addressing the problem of long queues of customers faced by banks at peak hours followed by slacked periods where there were no customer entries. The study sought utilize the Short Messaging Service (SMS) intervention to optimize capacity utilization while saving the customers from having to queue by notifying them when they were up for service. According to Silva (2016) An Integrated Queue Management System is using a numbered tickets, served by ticket dispensers, are probably the simplest existing technology for managing waiting lines. A staff member operating a queued service can simply call the next ticket aloud and register the last called number. According to Farayibi (2016) Investigating the Application of Queue Theory in the Nigerian Banking System. The study focused on GTBank and Ecobank and employed the multi-server queuing model to analyze queue characteristics in the banks and the performance measures (waiting and operational cost) to determine optimal service levels.

By implementing the Data Analytics and SMS Notification for Effective Truck Queue Management System in DHL Supply Chain it will ease the process of queueing. With the help of the new technology the encoding and storing of data becomes easy. The use of sms the queue number and the status of queueing will automatically send via text message. The monitoring of aueues will display on the monitor.Data analytics can analyze the delivery history and improve the current operation that will minimize logistical costs. Applying business intelligence, this system also can develop a performance metrics and KPIs, and create computer models to predict delivery issues. As queuing management systems collects data, it should be put to good use. Reporting capabilities will contribute to better visibility and control over future logistics outcomes.

CONCLUSION

Based on the information gathered the researchers therefore conclude that it is necessary to use the proposed system it is more efficient than the existing system. Based on the time study that the researcher conducted between current and proposed system it show that the proposed system is 37% more efficient than the current system.

RECOMMENDATION

Based on the findings of the study researcher would like to make the following recommendations for future improvements:

- Future research might apply an android version of receivers panel for easy assigning of docks.
- Future research should apply the Integration of Avon System to Truck queue management system to lessen the encoded data.
- Future research should include online truck booking.

GRAMMARIAN CERTIFICATE

CV OF EACH MEMBER

John Louie A. Abrina

17 Diego Silang St. Fortunata Village Brgy. San Isidro Sucat Parañaque City

Contact No.: (639) 222442232

E-mail: johnlouieabrina01@gmail.com

EDUCATIONAL BACKGROUND:

TERTIARY	Bachelor of Science in Computer Science
	Taguig City University
	2016 - Present
	Bachelor of Science in Information Technology
	Adamson University
	2010 – 2011
SECONDARY	St. Andrew’s School of Parañaque City
	2006 – 2010
PRIMARY	St. Andrew’s School of Parañaque City
	2000 – 2006

CO-CURRICULAR ACTIVITIES & AFFILIATIONS:

CICT Technolympics

Networking Division Head Committee
October 29,2019
Taguig City University

9th IT Skills Olympics
Computer Networking Category

September 9, 2019
University of Makati

8th IT Skills Olympics
Computer Networking Category

September 28, 2018
University of Makati

HONORS AND AWARDS RECEIVED:

System's Fair 2019
Best System 2nd Runner Up
(Community Management with Map Integration and Document Request System)

December 3, 2019
Taguig City University

9th IT Skills Olympics
1st Runner Up – Computer Networking Category

September 9, 2019
University of Makati

System's Fair 2018
2nd Runner Up – Assemble/Disassemble Category
Taguig City University

System's Fair 2017

2nd Runner Up – Assemble/Disassemble Category
Taguig City University

TRAINING / SEMINARS ATTENDED:

Mikrotik Programming

March 22, 2019
Taguig City University

Python Programming

February 23, 2018
Taguig City University

DOST-MIRDC Campus Talk

February 23, 2018
Taguig City University

All Out Connect – Learning Beyond the Fundamentals of Information Technology

February 10, 2018
Polytechnic University of the Philippines - Taguig

Cultivating I.T. Skills Towards Global Competitiveness

December 7, 2017
Taguig City University

Discover Open Source to a New World of Learning – Linux Day 2017

August 30, 2017
Taguig City University

Cyber Security Awareness Seminar

April 27, 2017
Parañaque City Hall

Survive and Thrive: Re-Discovering Your Career Passion

February 19, 2017
Taguig City University

Virtual World

November 22, 2016
Taguig City University

Game & Development

November 22, 2016
Taguig City University

Foss & Community

November 22, 2016
Taguig City University

QUALIFICATIONS:

- Hardworking, detail-oriented and scrutinous in work
- Excellent creativity skills
- Willing to learn
- Communicates both in Filipino and English
- Has background on web development using HTML, CSS and JavaScript
- Has basic knowledge in Java and C programming
- Has done PC assembly/disassembly and networking
- Database management using MySQL and MsSQL
- Proficient in Microsoft Office (Word, PowerPoint, Excel, Access)
- Familiarity with VMWare Workstation, Visual Basic and CMD
- Video/photo/audio editing (Power Director, Adobe Photoshop, Paint)

PERSONAL INFORMATION:

Date of Birth	: October 19, 1993
Place of Birth	: San Dionisio, Parañaque City
Age	: 26
Civil Status	: Single
Citizenship	: Filipino

Gender : Male

Religion : Roman Catholic

WORK EXPERIENCE:

February 2010 – February 2020	3J'S-NET INTERNET CAFÉ Brgy. San Isidro Sucat Parañaque City Manager / Server Administrator
January 2020 – February 2020	IBEX GLOBAL BF Parañaque IT Intern
June 6,2018 – July 21, 2018	JUS COPHIL TRADING INC. Brgy. San Dionisio Sucat Parañaque City Intern

CHARACTER REFERENCES:

(*Available upon request)*

I hereby certify that the facts contained in this resume are true and complete to the best of my knowledge and belief.

Jay-Alou B. Botona

#Purok 4, Bagong Tanyag, Taguig City

Contact No.: (639) 474462578

E-mail: botonajalo@gmail.com

EDUCATIONAL BACKGROUND:

TERTIARY	Bachelor of Science in Computer Science Taguig City University 2016 - Present
SECONDARY	SIGNAL VILLAGE NATIONAL HIGH SCHOOL 2002 – 2006
PRIMARY	FOURTH ESTATE ELEMENTARY SCHOOL 1996 – 2002

QUALIFICATIONS:

- Excellent in time management skills
- Strong interpersonal and communication skills
- Analytical with excellent creative problem-solving skills
- Extremely organized and detail oriented
- Android Development
- Joomla
- Website Development
- Visual Studio

PERSONAL INFORMATION:

Date of Birth	: September 24, 1989
Place of Birth	: Surigao Del Norte
Age	: 31
Civil Status	: Single
Citizenship	: Filipino
Gender	: Male
Religion	: Roman Catholic

WORK EXPERIENCE:

August 2015 – Present DHL SUPPLY CHAIN PHILS.
Team Leader

CHARACTER REFERENCES:

MELANIE ABARRO

Supervisor

09985854521

JERUSALEM CAMACHO

QA INSPECTOR

09984237547

I hereby certify that the facts contained in this resume are true and complete to the best of my knowledge and belief.

Dianne T. Evangelista

#64 BRAVO St. Zone 3 Central Signal Taguig City

Contact No.: (639) 151970902

Email: Evangelistadiannetan19@gmail.com

EDUCATIONAL BACKGROUND:

TERTIARY	Bachelor of Science in Computer Science
	Taguig City University
	2014 - Present
SECONDARY	SIGNAL VILLAGE NATIONAL HIGH SCHOOL
	2010 – 2014
PRIMARY	EM’S SIGNAL VILLAGE ELEMENTARY SCHOOL
	2004 – 2010

ORGANIZATION:

- **COMPUTER SCIENCE SOCIETY(CSS)**
  Public Information Officer 2019-2020
  Head Committee 2018-2019
  Event Organizer 2017-2018

TRAINING / SEMINARS ATTENDED:

Filipino IDEAS Expo

September 29, 2019
Taguig City University

AGILE PROJECT AND INTELLIGENT AUTOMATION

October 20, 2018
Taguig City University

DOST-MIRDC Campus Talk

February 23, 2018
Taguig City University

All Out Connect – Learning Beyond the Fundamentals of Information Technology

February 10, 2018
Polytechnic University of the Philippines - Taguig

ACHIEVING BUSINESS OBJECTIVES THROUGH DIGITAL TRANSFORMATION

September 9, 2017
Taguig City University

QUALIFICATIONS:

- Knowledgeable in Microsoft Office
- Willing to try new things and am interested in improving efficiency on assigned tasks.

PERSONAL INFORMATION:

Date of Birth	: September 19, 1997
Place of Birth	: Pasig City
Age	: 23
Civil Status	: Single
Citizenship	: Filipino
Gender	: Female
Religion	: Roman Catholic

WORK EXPERIENCE:

March 2019 – January 2020	Jollibee 31st Cor 2nd Avenue BGC Stop Over Pavillion fort bonifacio Service Crew
June 6,2018 – August 1, 2018	On The Job Training RMRB, Camp Bagong Diwa Bicutan Taguig City 300 Hours
April 2017 – September 2017	Jollibee 32nd Street Rizal Drive BGC Taguig Service Crew
June 2016 – May 2016	HILAS FTI Factory worker (chopper)

CHARACTER REFERENCES:

KEVIN FLORES
09269928698

Cherielyn laurio
Service crew
09464089380

I hereby certify that the facts contained in this resume are true and complete to the best of my knowledge and belief.

ALEXIS F. MAHINAY

#25 E-A Reyes St. Road New Lower Bicutan Taguig City

Contact No.: (639) 302987251

Email: alexmahinay023@gmail.com

EDUCATIONAL BACKGROUND:

TERTIARY	Bachelor of Science in Computer Science Taguig City University 2014 - Present
SECONDARY	Taguig National Highschool 2014 – 2015
PRIMARY	RP. Cruz Sr. Elementary School 2011 – 2012

QUALIFICATIONS:

- Knowledgeable in Microsoft Office
- Willing to try new things and am interested in improving efficiency on assigned tasks.

PERSONAL INFORMATION:

Date of Birth	: December 23, 1997
Place of Birth	: Taguig City
Age	: 21
Civil Status	: Single

Citizenship : Filipino

Gender : Male

Religion : Roman Catholic

CHARACTER REFERENCES:

(Available upon request)

I hereby certify that the facts contained in this resume are true and complete to the best of my knowledge and belief.