Freight Management System for Worldwide Relocation Services (Pvt) Ltd

B.A.S.S. Bamunuarachchi
BIT registration number: R040849
Index Number: 0408492

Supervisor: Mr. Aruna Warnakulasuriya

2009/ 2010

This dissertation is submitted in partial fulfillment of the requirement of the Degree of Bachelor of Information Technology(External) of the University of Colombo School of Computing

DECLARATION

"I certify that this dissertation does not incorporate, without acknowledgement, any material previously submitted for a degree or diploma in any university and to the best of my knowledge and belief, it does not contain any material previously published or written by another person or myself except where due reference is made in the text. I also hereby give consent for my dissertation, if accepted, to be made available for photocopying and for interlibrary loans, and for the title and abstract to be made available to outside organizations.

Signature of Candidate:. Date:1/08/2010

Name of Candidate:. B.A.S.S. Bamunuarachchi. . . . . . .

Countersigned by:

Signature of supervisor(s):. Date:1/08/2010

Name(s) of Supervisor(s):. . Mr. Aruna Warnakulasuriya .. . ."

ABSTRACT

Freight forwarding and handling industry plays a vital role in Sri Lankan exports and imports. Worldwide Relocation Services (WRS) is an emerging and very challenging company who involves in this industry. Among the hundreds of highly competitive freight forwarders carrying out their business locally and internationally, WRS stands steadily and wishfully to win the market share. Relocating personal effects all over the world is the main business of WRS which is currently operated by a manual system. This is a great drawback in their aim to provide more efficient and reliable services to customers.

The proposed freight management system was designed to address the issue by automating the manual process. The entire system was divided into five operational modules. Employees at WRS can use the functionalities provided by those modules to perform the operations namely, user administration, registering external parties involved in fright management, freight job handling and basic inventory handling. Management at WRS can use this system to evaluate their progress in the business.

The system was developed according to the three-tier architecture. Concepts in Object Oriented methodology for system design and the rapid application development methodology was used in development life cycle. Unified Modeling Language and its notations were used during the mapping of business objectives to the system architecture. C sharp .NET programming language on top of the Visual Studio 2005 platform was used for the development while the data storage and manipulation was done using MySQL.

The system evaluation was done in compliance with the standard evaluation procedures. System is functioning as desired and all the client requirements were fulfilled. Hence it can be said that this new system can be successfully used at WRS to perform their business activities.

ACKNOWLEDGEMENTS

I would like to express my deepest appreciation to the Director and the academic staff of University of Colombo School of Computing, for the support and facilities provided for the successful completion of the BIT program.

I am grateful to my advisor Mr. Aruna Warnakulasuriya for his invaluable support, guidance and encouragement rendered to me throughout this project.

Ms. Marleena Laxana, the Director of Worldwide Relocation Services (Pvt.) Ltd. happens to be a person for whom I do not have sufficient words to express my gratitude for all the support and encouragement given in fulfilling this project. Many thanks to the staff of Worldwide Relocation Services (Pvt.) Ltd for all the support and the useful discussions which helped me to obtain a better understanding of the freight industry which eventually helped me greatly in achieving the expected outcomes of this project.

Special thanks should be given to my parents for their constant love, support, encouragement and blessing. Finally, I would like to thank my brother and my friends for all the help given to me in various ways for the successful completion this project.

TABLE OF CONTENT

TABLE OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

WWW - World Wide Web

SDLC - System Development Life Cycle

OOD - Object Oriented Design

SD - Structured Design

WRS - Worldwide Relocation Services

FCL – Less Container Load

FCL – Full Container Load

HHG - Household Goods

NVOCC - Non Vessel Operating Common Carrier

RAD - Rapid Application Development

GUI - Graphical User Interface

CHAPTER 1: INTRODUCTION

1.1 INTRODUCTION

Sri Lanka the paradise island in the Indian Ocean has shown a rapid expansion in its economy resulting in a huge growth in the nation's Import Export sectors. Quality finished products are being manufactured and world renowned Sri Lankan products including tea and natural rubber are exported in large scale. Important products such as food items, electronics, automobile and petroleum are among the main imports.

The rapid expansion in international trade has shown the growing need for an efficient supply chain solution. Freight forwarding industry in Sri Lanka plays a vital and active role in this sector.[WWW1]

Freight is defined as goods transported generally for commercial gain, by ship, aircraft, train, van or truck. [WWW2] Supply chain management, Freight forwarding, ocean import export, air import export, warehousing, custom brokerage can be identified as main supportive roles in freight industry.

Freight management is a strategic system that helps to optimize the efficiency of freight and commercial transport by closely coordinating above supportive roles to minimize the freight cost. [Paul, 1998]

There are number of approaches to implement freight management in the transportation of goods, when possible shippers are encouraged to utilize either water or rail systems. Optimizing truck loads by improving the routing and scheduling to avoid empty trucks on return trips is one approach. Organizing regional deliveries is another approach which reduces the number of trips and saves on energy expenses which is also a benefit to the environment.[WWW2] In all the approaches stated above introduction of information technology has greatly improved the core aspects of freight management. So many supportive software which handle different sections of the freight management has been introduced to the market over the past few years. These IT solutions have largely helped to administrate the freight efficiently and effectively in a profitable manner.

In the Sri Lankan context as mentioned before with the rapid expansion of export import industry many freight forwarding companies have mushroomed and there is a big competition to increase their market share. WRS is one of them.

1.2 PROBLEM DOMAIN AND THE NECESSITY OF THE PROJECT

Currently WRS is operating in a low profile. Their business mainly focuses on a specialized area known as Removal & Relocation of personal and household goods (HHG). They undertake shipments to translocate personal effects all over the world. In addition, they work as consultants in this business area. Currently most of their business processes are carried out manually and their data are stored in spreadsheets. These manual processes are very time consuming and they experience a lot of data duplication which leads to inconsistencies in their business. Due to these reasons they are in need of an IT based solution to increase their productivity which consequently helps them to increase their market share. Also with the complete solution tailor made to their specialized area they can monitor the business more efficiently. They can provide an analysis to the shipper the exact cost per freight order of the anticipated freight cost. This will improve the customer satisfaction which is ultimately going to reflect in their business growth.

1.3 OBJECTIVE AND SCOPE OF THE PROPOSED SYSTEM.

Considering the business performing at WRS, following items have been identified as the main objectives of the project and are described below with the scope.

System should be able to register customers, third party organizations that support freight handling at WRS and suppliers of the inventory.
Facilitate the shipment process starting from generation of shipment ID based on the type of shipment after checking the preconditions. This also includes handling necessary document in the flow, identifying the route based on the cost and the availability of cargo carriers.

Handle the packing and loading processes of shipping items. This includes checking the condition of the goods, depending on the LCL and FCL identifying the sizes of containers needed. Depending on the type of goods, system should be able predict appropriate packing materials and packing procedure to the staff.
Automate the process to identify the inventory items to be purchased and issuing items for packing.

In addition to those the system will provide assistance to managers for profit calculation in individual shipments with cost sheets.

As a result of achieving above objectives following deliverable will be produced in the system.

- ✓ Implementation and support for:
  - Shipper information
  - Consignee information
  - Supplier information
  - Vessel and flight information
- ✓ Freight job ID Generation
- ✓ Shipper instruction generation
- ✓ Pick list
- ✓ Load chart
- ✓ Pre alert notices
- ✓ Material cost sheet
- ✓ Profit share calculation
- ✓ Customer invoicing
- ✓ Document management
- ✓ Training and documentation of the system

1.4 CHAPTER OVERVIEW

Structure of the subsequent chapters of this dissertation is described below. Each of these chapters explains the activities performed during the software development life cycle of this project.

Chapter 2: Analysis

Activities carried out in order to identify the functional and non functional requirements will be discussed in this chapter. This will include identification of user domain and different fact gathering techniques that were used in this phase.

It will also briefly explain about the current system. This chapter will also outline the existing similar solutions and the development methodology that was followed in this development life cycle.

Chapter 3: Design

This chapter begins with a discussion of different system development methodologies and a justification of the selected approach. It also covers the activities performed during the design phase of the proposed system. Here the technology used and the overall architecture of the system will be discussed with appropriate diagrams. The design of the user interfaces of the system will also be discussed with some sample screen shots of the system.

Chapter 4: Implementation

The different paradigms of the implementation stage will be discussed in this chapter. This includes details of the hardware and software that were used in the development environment. Source code samples and implementation of the proposed modules will be discussed in this.

Chapter 5: Evaluation

This chapter will give an overall idea about the evaluation process done in the system development life cycle and compare it with the proposed system. This chapter will also describe the lessons learnt during the project.

Chapter 5: Conclusion

This chapter will conclude the dissertation summarizing the project activities carried out in this phase of the project with suggestions for any future modifications.

CHAPTER 2: ANALYSIS

This chapter consists of the activities carried out in the analysis phase. It includes a description of the current system, an outline of existing similar solutions and requirements of the proposed system. It will give a brief description about the user domain and discuss the strategies followed in gathering and analyzing the requirements of the system.

2.1 REQUIREMENT ANALYSIS

In the System development life cycle considerable amount of time was allocated to requirement gathering and analysis. Simply because having a very clear understanding of the proposed project at the beginning is essential for a successful completion. Since this project is to address somewhat complex business scenarios in the freight forwarding industry very broad knowledge of the domain was gathered during the analysis phase. This helped a lot in identifying domain specific terms and the business.

Several fact gathering techniques were used to identify the business requirements.

Through one to one interview, Group interview, facilitated sessions, questionnaires and following people around are some of them.

Out of these techniques the main strategy followed in gathering user requirements for this system was through user interviews. Furthermore, a vast amount of knowledge was obtained by investigating the documents in the current manual system. Several interviews were conducted with the different levels of users. In order to get an overall picture of the manual system WRS management was interviewed. Few people from the operational staff were interviewed. This was helpful in understanding the nonfunctional requirements.

2.2 THE BUISNESS PROCESS AT WRS

2.2.1 Existing Manual system

People are traveling allover the world for different purposes. For study purposes, business requirements, as migrants are some of the examples. Since they stay in those countries for more than a usual tourist, they tend to take their personal effects with them. But the amount of baggage that they can carry onboard is very limited and they need to depend on a reliable and an economical air or sea cargo service.[WWW3]

Worldwide Relocation Services is specialized in this business area and they work as a freight forwarder to provide reliable and very efficient services to thier customers. So that, customers can travel allover the world with a greater peace of mind when it comes to the security of their baggage.

In addition WRS organizes shipments for organizations who needs to export their products in large quantities and to import raw materials for their facilities.

Personal effects shipment business process starts when a shipper contacts the WRS office regarding his or her shipment. For this type of shipment shipper must hold an international airline travel ticket, and visa documents between the origin and the destination airports. Once the shipper places an order with details of items to be shipped WRS as the freight forwarding service establish relationships with carriers of all kinds from air freighters, trucking companies to rail freighters and ocean lines. Then they negotiate the best possible price to move the products along the most economical route by working out various rate structures and choosing the best balances in speed, cost and reliability.

Then WRS provides one or more estimates to the client along with advice when necessary. Considerations that affect the price will range from origin and destination to special requirements such as refrigeration, fumigation, special insurances and transport of potentially hazardous materials, also if there are any special government approval that's need to be taken. Assuming the client accepts the bid the freight is prepared for shipping. WRS then undertakes the responsibility of arranging the transport from port of origin to the destination by an agreed upon date.

WRS takes care of ancillary services that are part of international and shipping business. Insurance and custom documentation, authentication documents from the government such as forest authorization license for wooden items, chemical license for organic products and clearance are some of examples.

As a consolidator WRE provides non vessel operating common carrier (NVOCC) documentation or bill of lading, warehousing risk assessment and management. They also provide method of international payments to the client.

Different shipping methods are used to transport personal effects. Mostly it's door to door shipment. Shipper and the consignee is the same person. These shipments are normally less container load (LCL) and collection of orders is going in one container. Some customers come with a heavy load to transport and they are known as full container load (FCL). Large organizations mainly involved in exports and imports are the potential customers of this type. These types of shipments are categorized under door to port shipments and the freight forwarder in the destination country is responsible for deliver goods.

When WRS accepts the shipping items to be shipped they will be checked at the customer site and will be transported to the warehouse owned by WRS. Then at the warehouse items will be further processed and the packing and loading will be done in to standard size containers according to the international standards.

The containers will transported to the port on agreed date with the shipping liner. After custom approval containers will be loaded to the carriers and transport to the destination port. Necessary documents will be sent with the shipment and the shipper will be informed about the progress of the shipment.

There are several shipping liners, air freighters and trucking agents registered with WRS. WRS will contact these organizations when they receive a shipment.

There is a separate process conducting at the warehouse to purchase and handle packing items. Suppliers who register with WRS will provide these items when the request is made by the warehouse manager. Currently, they maintain their inventory using spreadsheets. There are several employees involved in packing and loading, who works on contract basis.

Currently, the staff at WRS handles all these business processes and other supportive tasks manually. They keep track of the records mainly through spreadsheets. This is very time consuming and prone to errors. WRS has a very loyal set of customers and they tend to do their business regularly with WRS. Since the company does not have dedicated and organized way of customer detail handling customers have to provide personal detail each time when they come to do a shipment. This is a big burden to the customers as well as WRS office staff which will also reduce the customer satisfaction. Since the current system is a manual process office staff needs to handle lots of paper work and they maintain separate files for each shipment. After a customer has placed an order there are lots of internal operations that are being done in order to identify the possibility of shipment while keeping a profit margin. Staff has to go through this files in order to get the required information and it’s time consuming. Data circulation in the organization was not logically arranged. As result of that management cannot get quick decisions resulting in a loss of profit margins.

Since all the calculation and routing decisions are done manually customers have to wait for some period of time to get their pre alert notices and the cost of shipment. In a highly competitive industry like freight forwarding this type of situations are not tolerable and is a great disadvantage for them when gaining the market share. Due to the above stated reasons management of the WRS has realized that if an automated system is available they will be able to carry out these activities very easily.

2.2.2 Some documents involved in the current system.

WRS possess several business document templates that are circulating in the business flow. Currently they manually complete these documents on demand.

Some of the documents involved in the current system are listed below.

- Shipment instruction sheet
- Pre alert notice.
- Bill of lading.
- Material cost sheet
- Load charts
- Pick list
- Profit share Calculator
- Invoice form

2.3 THE OPERATIONAL ENVIRONMENT

When considering the operational environment related to the business process carried out at WRS several user roles can be identified. They are,

- Customer
- Sales Officer
- Forman
- Inventory Clerk
- Supplier
- Manager
- Administrator

2.4 FUNCTIONAL REQUIREMENTS

The functional requirements are the statements of services. They state what the system should provide, how the system should react and how to behave in particular situations.

After detailed analysis of the proposed system functional requirements described in table 2.1 have been identified and they are categorized based on the criticality and the dependencies.

Req. No	Description	Criticality	Dependencies	Technical issues
1	Ability to register shippers and suppliers	High		
2	Ability to enter details of shipping lines and air lines	High		
3	Ability to generate Freight job ID	High		
4	Ability to enter shipper instructions	High	1,3	
5	Ability to handle shipping items	High	3,4	

6	Ability to generate cost sheets of the goods for customer purpose	High	1,3,4,5	
7	Ability to generate cost sheets of the goods for insurance purpose	High	3,4,5	
8	Ability to generate pre-alert notice	High	3	
9	Ability to generate shipment advice	High	3	
10	Ability to create invoices	High	3	
11	Ability to perform profit share calculation	High	1,3,5,6,7,9	
12	Ability to keep track of external documents	High	3,4	
13	Ability to generate cost per job ID.	High	3	
14	Ability to create packing chart	High	3,4	
15	Ability to analyze conditions of the goods	High	4	
16	Ability to handle loading and vessel detail	High		
17	Ability to register Inventory location	Low		
18	Ability to register inventory items	High		
19	Ability to identify reorder level	Low	18	

Table 2. 1 Functional requirements of the system

2.4 NON FUNCTIONAL REQUIREMENTS

Non functional requirements of the system can be categorized into two types.

- Qualities of the system
- Constraints of the system

Qualities of the system are the features that the users expect from the system. These could be further divided as run time qualities, which specify how well the functional requirements are satisfied, and development time qualities which are driven by the goals, of the developer or development organisation. Runtime qualities is described in table 2.2 while the development time qualities illustrated in the table 2.3.

Constraints of the system are the limitations which apply when the system is being developed and used.

2.4.1 Run-time Qualities.

Usability	The system should allow users to operate it with minimum training. The system should tolerate errors made by the users.
Supportability	The system should allow users to configure the necessary software.
Availability/Reliability	The system should be operational 24 hours. The mean time to repair should be less than 24 hours.
Quality of Service Requirements	All forms should load within 5-60 seconds at any load condition.
Security	User should protect form the malicious programs.
Scalability	-

Table 2. 2 Runtime qualities of the system

2.4.2 Development time Qualities.

Localizability	N/A
Modifiability	Should be able to be modified, include new functions.
Evolvability	The system should evolve into a complete system at the end of all phases.
Composability	Should consist of service layer components, implementing the business logic.
Reusability	Should consist of reusable service layer functions.

Table 2. 3 Development time qualities of the system

2.5 SIMILAR SOFTWARE SOLUTIONS IN THE FREIGHT INDUSTRY

1) Logisuite.

Logisuite provides a complete set of logistic software solution to the freight industry. This vender provides a customized set of software which can support different section of the freight industry like Inventory Control, Supply-Chain Management, Third-Party Logistics (3PL), Freight Forwarding Software, Ocean Import, Air Import, Domestics

Freight, Pickup & Delivery and Purchase Order Management. Companies who are specialized in each of these areas can purchase the most suitable software solution to their business for an affordable price.[WWW4] Among the available software solutions "Logisuite Enterprise" provides the complete set of functionalities. Logisuite solutions design to run on web browsers.

2) Cargo WIZ

Cargo WIZ is a truck and container loading software. It is specialized in packing and loading. Venders of this product allow the customers to use a trial version free of charge for a short period of time and purchase the software when they are satisfied. This software possesses a nice and advanced feature which is 3D visualization of the packing containers.[WWW5]

3) Parcel Perfect

Parcel Perfect is freight management software specially designed for the smooth running of a courier company. It provides variety of features related to the freight industry. Interactive collection of data using text messages, providing flexible rating structure of all the type of freight jobs, Consolidation of waybills onto trip-sheets for delivery, Invoice generation and printing are some of the key functionalities supported by this software.

This software uses an SQL database engine and can be deployed in a local server. This system can be deployed on a Microsoft windows platform or in Linux. The maximum number of workstations that can be attached to the system is being determined by the hardware specification of the server. All workstations must run Windows. Versions supported are Windows™ 98SE, 2000 Professional and XP Professional.

This system is available for a customized pricing system and fully maintained and supported for a monthly fee.

CHAPTER 3: DESIGN

3.1 SYSTEM DEVELOPMENT METHODOLOGIES

A methodology can be simply defined as a set of procedures that one follows from the beginning to the completion of the software development process. A variety of such frameworks have involved over the years, including the waterfall model (the original SDLC method), rapid application development (RAD), joint application development (JAD), prototyping and the spiral model, each with its own recognized strengths and weaknesses.[WWW6] One system development methodology is not suitable for all projects. Each of the available methodologies are best suitable based on various technical organizational, project and team considerations. Documentation is crucial regardless of the type of the model chosen or devised for any application and is usually done in parallel with the development process.

3.1.1 Waterfall model

The waterfall model is a sequential development process, in which development is seen as flowing steadily downwards (like a waterfall) through the phases of requirements analysis, design, implementation, testing (validation), integration, and maintenance.

Emphasis is on planning, time schedules, target dates, budgets and implementation of an entire system at a time.[WWW6]

This model is inflexible, slow, costly and cumbersome due to significant structure and tight controls.

3.1.2 Prototyping

Prototyping is not a standalone development methodology. But rather an approach of handling selected portions of larger, more traditional development methodologies like Incremental, Spiral, or Rapid Application Development. It attempts to reduce inherent project risk by breaking a project into smaller segments and providing more ease-of-change during the development process. User is involved throughout the process, which increases the likelihood for user acceptance of the final implementation.

3.1.3 Spiral

Focus is on risk assessment and on minimizing project risk by breaking a project into smaller segments. Each cycle involves a progression through the same sequence of steps.

3.1.4 Rapid application development

Key objective is to develop and deliver a high quality system at a relatively low investment cost. It aims to produce high quality systems quickly, primarily through the use of iterative Prototyping (at any stage of development), active user involvement, and computerized development tools.[WWW6] These tools may include Graphical User Interface (GUI) builders, Computer Aided Software Engineering (CASE) tools. Active user involvement is imperative. Iteratively produces production software as opposed to a throwaway prototype.

3.2 MAJOR SOFTWARE DESIGN METHODOLOGIES

A methodology can be simply defined as a set of procedures that one follows from the beginning to the completion of the software development process. Some of the popular design methodologies are described below.

3.2.1Top-Down/ Bottom-Up Design

Top-down design directs designers to start with a top-level description of a system and then refine this view step by step. With each refinement, the system is decomposed into lower-level and smaller modules. Top-down decomposition requires identifying the major higher-level system requirements and functions, and then breaking them down in successive steps until function-specific modules can be designed. Thus, top-down design is a level-oriented design approach.

Even though this is an iterative process, the decisions made at the upper-levels will have a significant effect on subsequent decomposition at the lower-levels. As a result, there is a possibility that decisions made at the upper level will result in an untenable or awkward or inefficient decomposition at the lower-levels. To overcome this, there has to be a significant amount of backtracking where the decisions of the higher levels have to be re-evaluated and then re-structured accordingly. This is in contrast to the bottom-up approach where the lowest modules or the basic features are determined first, and then the additional modules or features are added.

Bottom-up design is also an iterative process, and can also result in significant backtracking if the primitives are not properly constructed. The benefit of the bottom-up design is that it permits the assessment of the sub-modules during the system development process. Whereas in the top-down design, performance evaluation can be done only when the complete system is integrated.

3.2.2 Structured Design

It is a data flow-oriented design approach. In the recent past it was probably the most popular methodology of software design. It is easy to use and there is an evaluation criterion that can serve as a guide in the software design. The main notational scheme that SD uses is the data flow diagram (DFD). SD is conceptually dependent on three rationales which are composition and refinement of the design, separation of issues into abstraction and implementation and evaluation of the results. From the compositional rationale, SD views systems from two perspectives: as the flow of data and the transformations that data flow undergo through a system. Since data flows and transform actions are the only characteristics depicted in the DFDs, the element of time is not present. Thus, the designer can just focus on the transformations of the data flows through a system. This is in contrast to the top-down design where decisions made at the top-level will affect decomposition at lower-levels. In addition, the interdependence of these data flows and transformations will result in the identification and organization of modules required in the building of the system model.

3.2.3 Object Oriented Design

Object Oriented Design (OOD) provides a mechanism that encompasses three important concepts in software design: modularity, abstraction, and encapsulation. OOD models the problem in terms of its objects and the operations performed on them.

OOD decomposes the system into modules where each module in the system denotes an object or class of objects from the problem space [Booch, 1999]. Objects encapsulate data attributes, which can be data structures or just attributes, and operations, which are procedures. A class provides template for objects that share a set of common structure and behavior [Jacobson, 1999]. The derivation of subclasses from a class is called inheritance. The ability of any objects to respond to the same message and of each object to implement it appropriately is called polymorphism. Relationships describe the dependencies between classes and objects.

3.3 ALTERNATIVE SOLUTION EVALUATION

In software development it is important to have an appropriate architecture for the system. The evaluation of an appropriate architecture is not only governed by the functional requirements but also by the quality attributes to a large extent.

However, knowing this it is still a non trivial task to discern between architecture candidates. There is usually more than one quality attributes involved in a system.

Pros and cons of each of the analyzed solutions during the feasibility study are given in the table 3.1.

Feasibility factor	Candidate Solutions			
	Web applications	Client Server applications	Standalone desktop applications	Mobile applications.
Operational feasibility	Staff at WRS virtually uninvolved in the system upgrades and it's done on the ASP's schedule.	Internal staff has full control of the environment. But the current staff lacks IT competency. Must hire skilled staff.	Highly feasible as it can be implemented in the current environment.	It's not technically feasible to implement all the requirements but can be used as an auxiliary service to get quick responses from external parties.
Security	Open to large number of users in the Internet community thus exposed to more security risks.	Number of uses is limited. Can fully control the usage of the system by introducing own security	Have a total control over the standalone applications and protects it from various vulnerabilities.	Security specific to mobile client must be implemented

		policies.	Security can be granted individually.	
Connectivity	Significantly depends on Internet connectivity. Poor connectivity has a great effect on system performance.	Speed and the performance totally depend on the network infrastructure used in the company.	Desktop applications are standalone in nature and hence do not face any hindrances resulting from Internet connectivity.	It is feasible as many ISP's provide this facility.
Maintenance & Upgrade	Web based applications need to be installed only once in the server hence updating the application is very easy	Most of the core components are installed in the server. Implementation of thin client helps in easy updates.	Applications are to be installed on each computer. Updating the application is cumbersome. Needs to be done on every single computer.	Must require special hardware. Size of the application is a major concern.
Cost Factor	System is typically offered on a subscription or usage basis. Development and its maintenance involve higher costs and have no control over them. It will cost approximately Rs 10000/= for web hosting.	In control of upgrades and cut ties with service providers. Purchasing of hardware can be decided based on cost and performance. Maintenance cost and the consultancy fee is high. Setting up a network will cost more than Rs200,000.	It's highly feasible because the existing hardware components can be used. Minimum networking and consultancy fee	Purchasing specialized hardware and software will add an extra cost to the system.
System Access.	Software is accessed from anywhere with a standard browser and an internet connection.	Software is accessible through a local network creating reasonable quick response	Each user must have a PC with minimum hardware requirements to access the system. Provide	Improve system interaction. Possible to update customers about their shipments

		and autonomy from the internet	fastest accessibility compared to other systems.	regularly hence improve the customer satisfaction.
Acceptance	Rejected	Rejected	Accepted	Rejected
Reason	Requirements Identified in this development life cycle mostly relates to company internal operations and system response time is a major concern.	High cost of implementation and Maintenance	Technically and economically most feasible solution. System response time is very high. Immune to most of the security threats.	Not suitable to the functional requirements. High cost of implementation

Table 3. 1 Alternative solution analysis

3.4 JUSTIFICATION OF THE SELECTED APPROACH

Considering the strengths and weaknesses of the above mentioned methodologies Rapid application development methodology was selected as the development framework for this project and the object oriented design was used as the design methodology. RAD allows the usage of several techniques like prototyping, incremental development of selected modules in short period of time and creation of reusable components. This improves the user interactivity and reduces the project risk.

This application is developed using the .Net platform, using C sharp as the programming language. C sharp has been introduced as a brand new object-oriented language.

Today, many developers use Unified Modeling Language (UML) as a standard for specifying, visualizing, constructing, and documenting the artifacts of software systems. Usage of Object-oriented software development allows using UML as a standard notation for modeling application.

Three-tier architecture was used to improve the reusability and this helps to perform future upgrades easily. Data of the system resides in a relational database and the DBMS work as the interface to the application. Application consists of data manipulation layer, business logic layer and the presentation layer.

3.5 DESIGN OF THE SYSTEM

Functional and non functional requirements identified at the requirement analysis phase have been used as the input for the design. The Object Oriented methodologies such as use case diagrams, sequence diagrams were used to model the main functionalities and the scope of the system. UML notations were used to visualize the system design.

Use case diagram of the generalized system is given in figure 3.1.

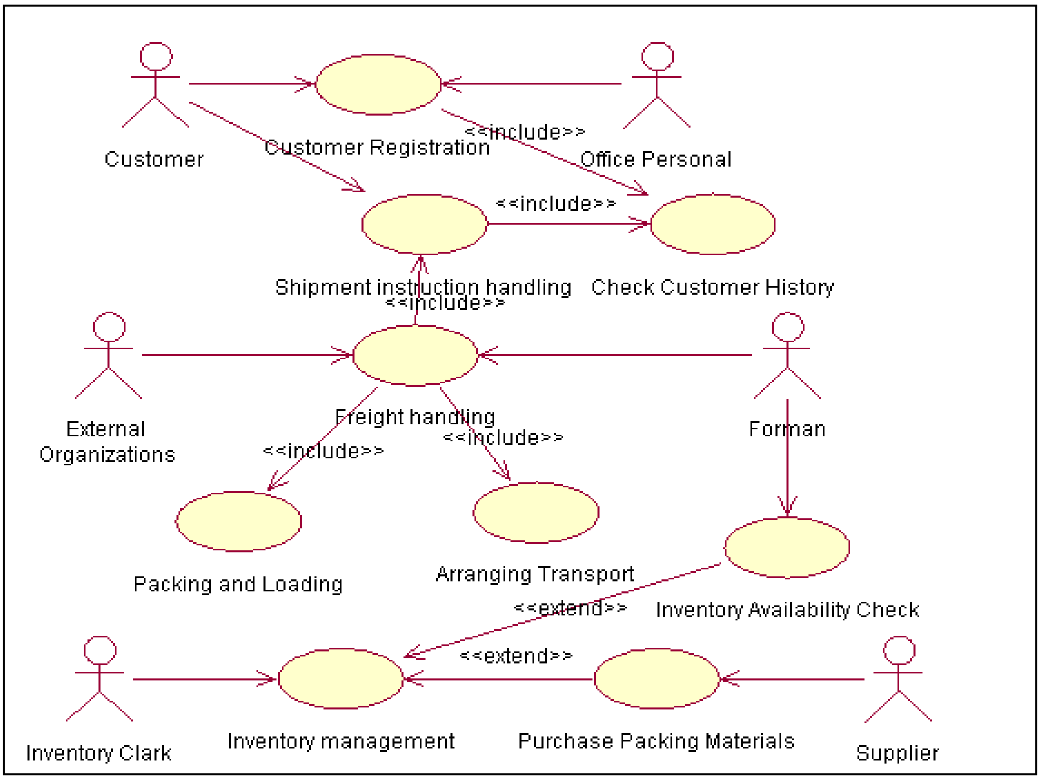


Figure: 3. 1 Main use case diagram of the Freight management system

3.5.1 Registration of sub system

Figure 3.2 shows the use case diagram for the registering process. This is the process where different users of the system with different business prospective will be registered. Actor is defined as a customer who can be specialized into shippers, shipping line, supplier because they are having common attributes with a specialized business role.

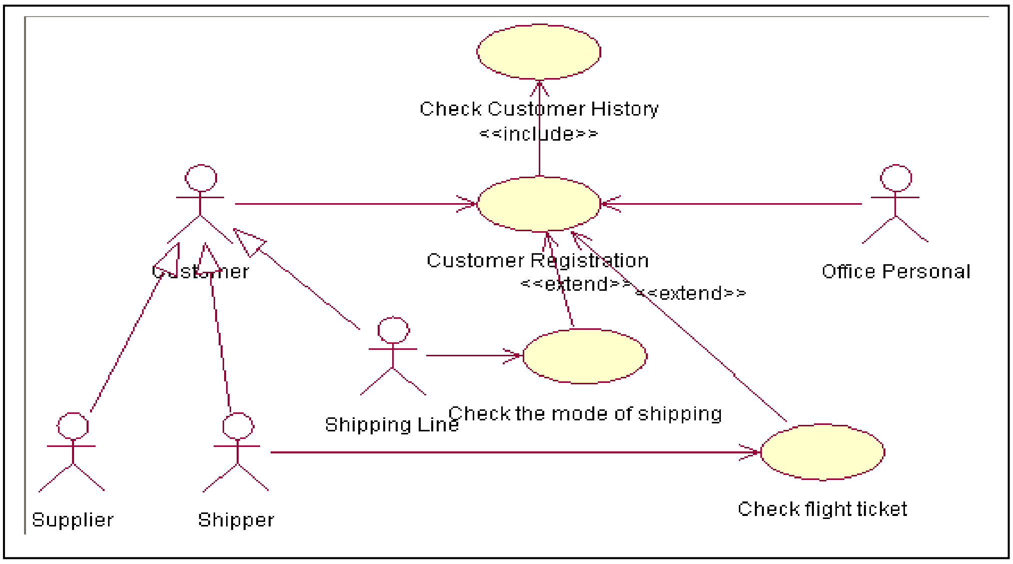


Figure: 3. 2 Use case diagram of the registration system

Use Case	Register in WRS system
Actors	Customer (specialized as shippers, Shipping liner, Supplier), office personal
Activity	Register shippers, Shipping lines, Suppliers
Preconditions	
1)Should not be already registered in WRS	
Flow of Events	
1. Check if the Customer is already registered in WRS, if not proceed 2. Complete all the relevant details and validate. If validation fails allow reenter or reject 3. Provide a generated ID	
Post Condition	
Customer will have a valid registration number which can be used in all future references	

Figure 3.3 shows the activities carried out during the customer registration.

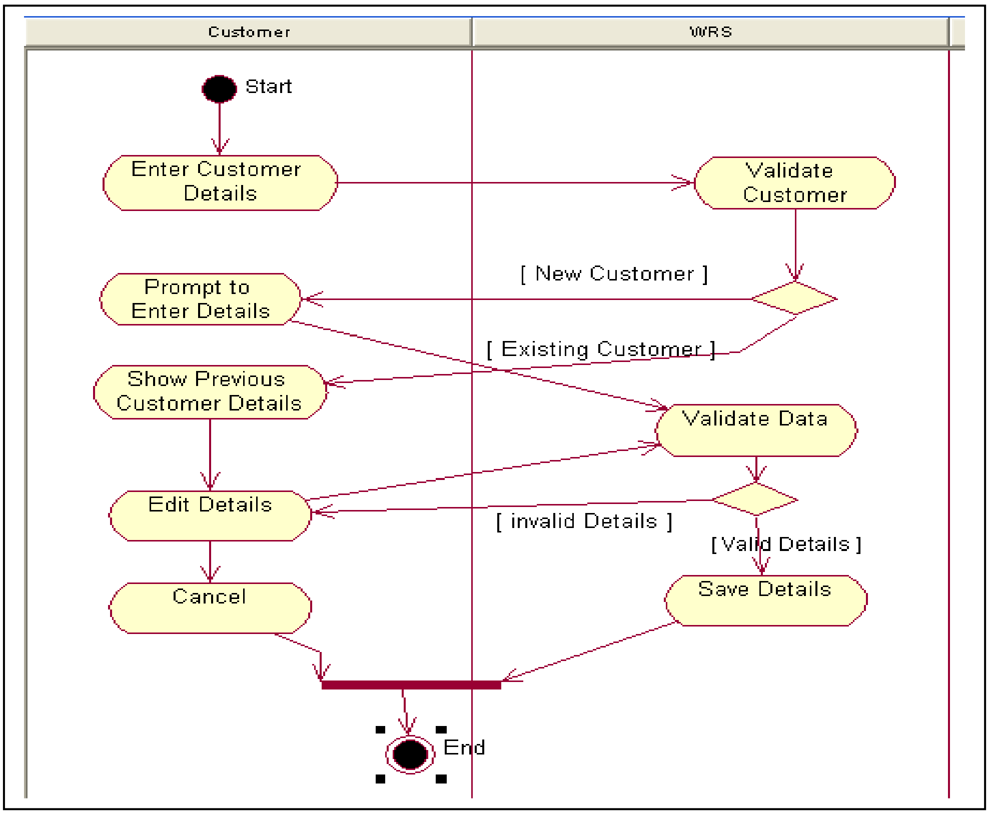


Figure: 3. 3 Activity diagram of the registration system

3.5.2 Generation of Shipment instruction.

Figure3.4 illustrates the user interactivity with the system when customer places an order of shipment.

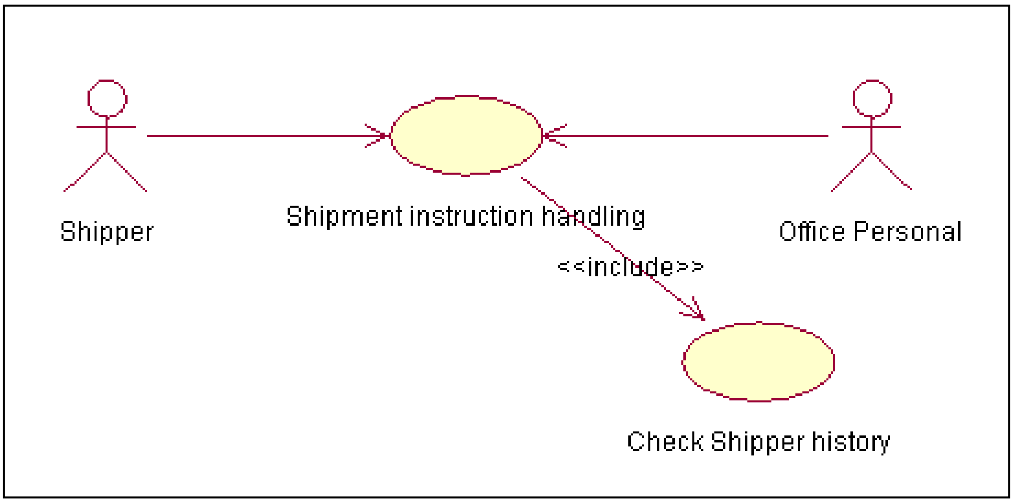


Figure: 3. 4 Use case diagram of the Shipment instruction

Description for shipment instruction use case.

Use Case	Generation of Shipment instruction
Actors	Customer (specialized as shippers, Shipping liner), office personal

Activity	Generate shipment instruction
Preconditions	
1) Shippers should be WRS registered customers. 2) Shippers should possess a valid air ticket.	
Flow of Events	
1. Check if the shipper is already registered in WRS, if not proceed prompt to enter shipper detail 2. Fill all the relevant details and validate. If validation fails allow reenter or reject. 3. Add the items to be shipped with the relevant details. 4. Provide a generated ID for shipment.	
Post Condition	
Customer should be notified the acceptance or rejection of the shipping order.	

Activities performed during the Shipment registration is given in the figure 3.5.

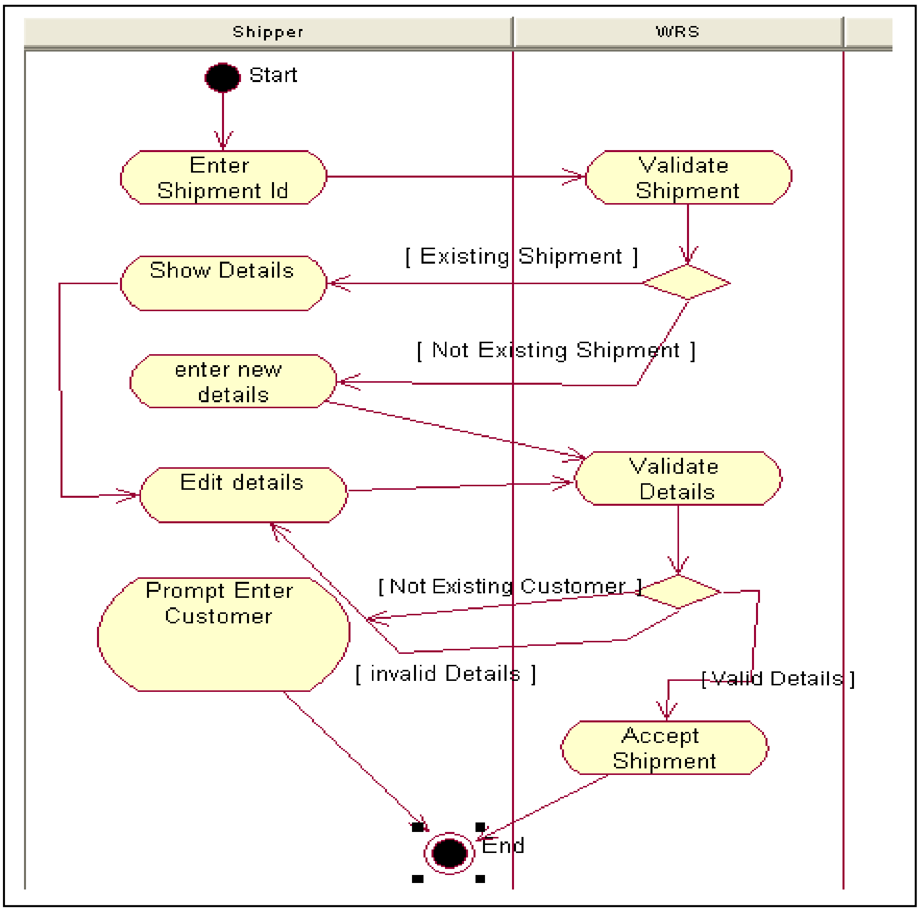


Figure: 3. 5 Activity diagram of the Shipment instruction

Class diagram of the Shipment instruction handling is given in the figure 3.6.

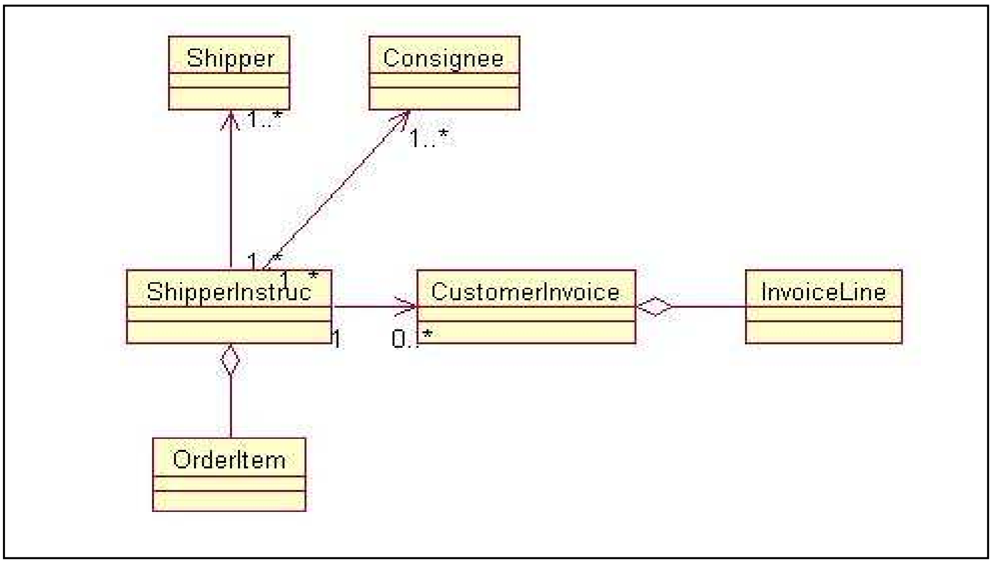


Figure: 3. 6 Class diagram of the Shipment instruction

3.5.3 Freight handling

Pictorial view of the freight handling is shown in figure 3.7 and the activities carried out are given in figure 3.8. This is the most complex functional requirement to handle in the system. This involves lot of internal documentation and checks.

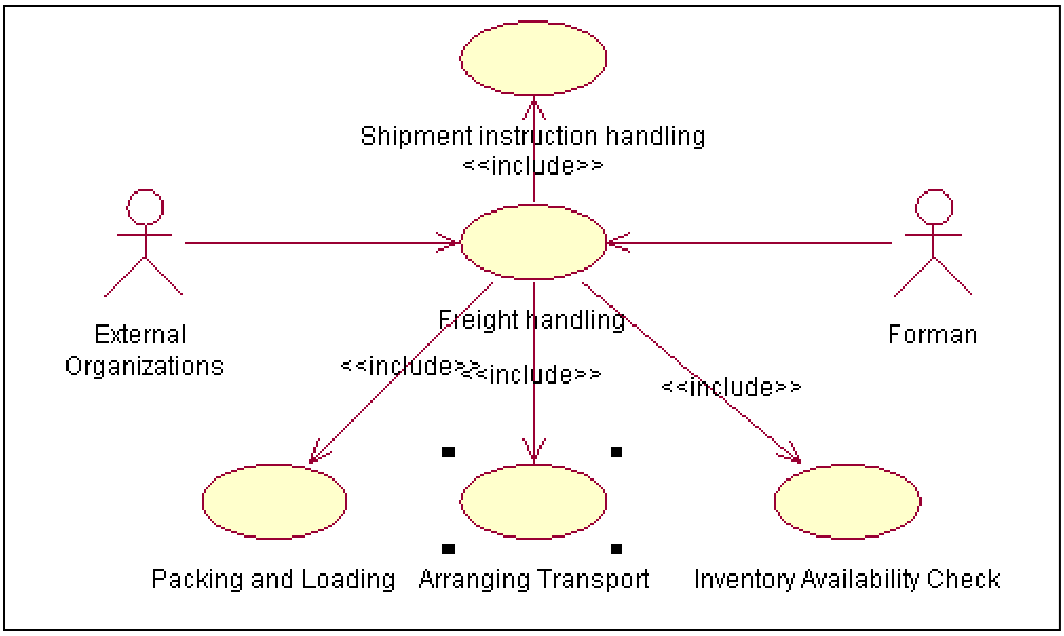


Figure: 3. 7 Use case diagram of the Freight handling

Description for Freight handling use case.

Use Case	Freight handling
Actors	Customer (specialized as shippers, Shipping line), Forman

Activity	Assign most economical route/shipping line, Create packing list, Create cost per job id. Generate pre alert notice
Preconditions	
1) There should be an authorized shipment order. 2) Shipping line should be available 3) Packing items must be available	
Flow of Events	
1. Check if the valid shipping instructions are available 2. Analyze the conditions of the goods. 3. Create packing chart 4. Check availability of container 5. Check availability of vessels 6. If all the above conditions satisfied generate pre alert notice 7. Upon customer agreement generate invoice 8. Perform the shipment	
Post Condition: Items will be shipped.	

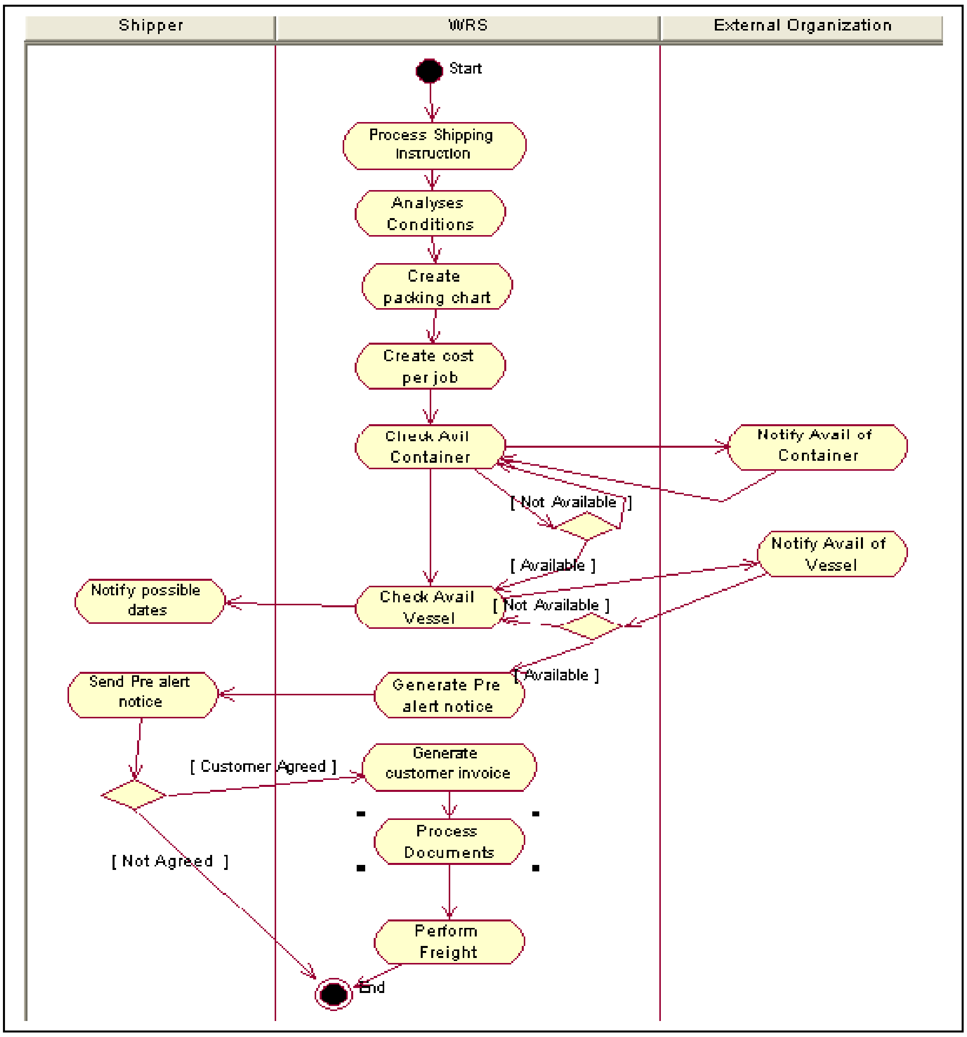


Figure: 3. 8 Activity diagram of the Freight handling

Figure 3.9 shows the classes that are involved in the freight handling and the relationships.

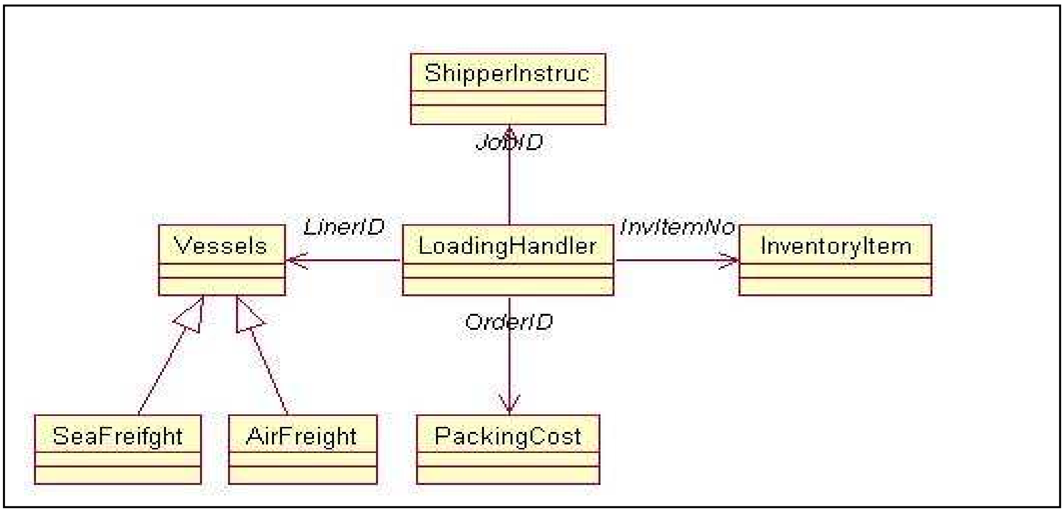


Figure: 3. 9 Class diagram of the Freight handling

3.5.4 Basic inventory control

Use case diagram of the basic inventory control system is given in figure 3.10

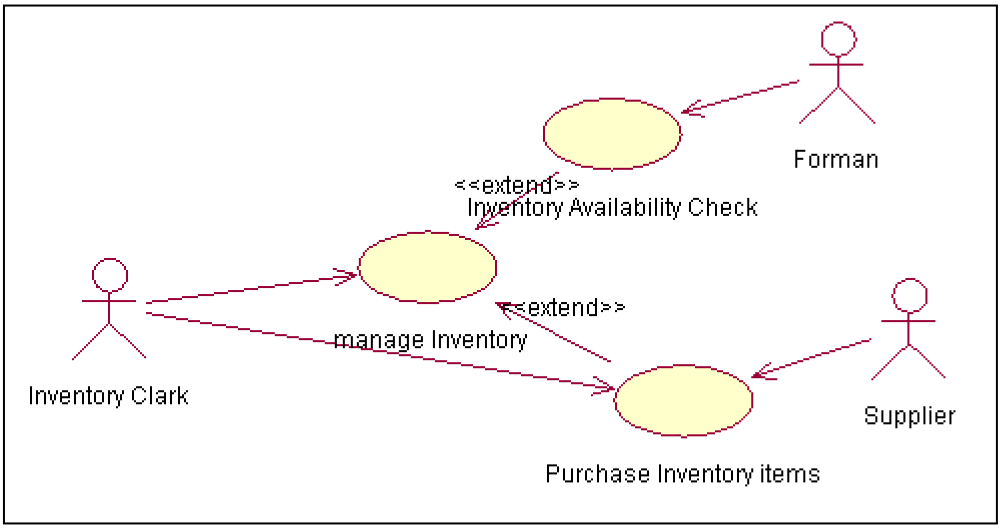


Figure: 3. 10 Use case diagram of the in inventory control

Description for Inventory handling use case.

Use Case	Manage Inventory
Actors	Inventory Clark, Supplier, Forman
Activity	Forman will request for packaging items. Inventory clerk must issue the packing items and purchase new items if there is a shortage.
Preconditions	
1. Supplier should be registered 2. Request of packing items with quantity must be available.	
Flow of Events	
1. Issue material to the shipment orders.	

2. Check possible shortage of items and create purchase orders.
Post Condition

Figure 3.11 shows the Class diagram of the inventory handling sub system.

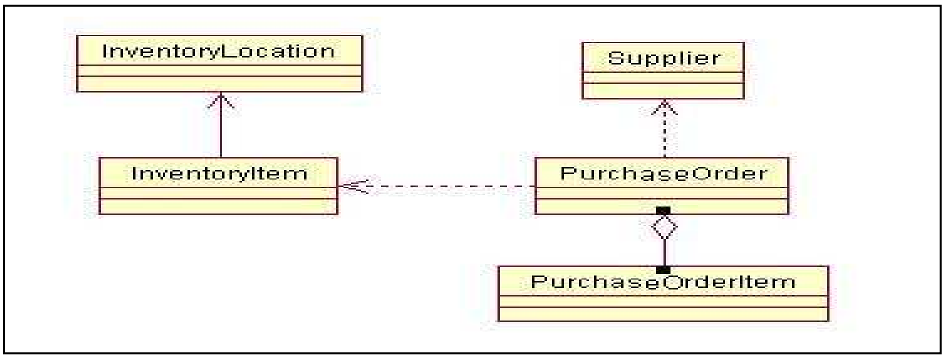


Figure: 3. 11 Class diagram of the inventory control

3.6 INTERFACE DESIGN

3.6.1 User Interfaces

The importance of good user interface design can be the difference between the success and failure of a system. Especially for a generic software poorly designed user interface can be the cause of a failed marketing campaign and loss of market share.

When it comes to the tailor made off the shelf software like this system still the user-friendly interfaces place a major role in its success.

The only way the end users see the system is through the interfaces. They judge the system based on how easily they can get familiarized to the system and carry out their work. Therefore many factors need to be considered when developing user interfaces. When the users of the system are considered, all the users are not the same. There are many differences in terms of experience in problem domain, knowledge of the system and frequency of using the system.

Also when it comes to customized software like this, specially management expects a uniqueness of the system so that they can show these systems as their assets. Therefore the above stated differences should be addressed when designing the user interfaces.

3.6.2 Requirements of the user interfaces

A well designed user interface is a successful outcome of a collaborative work done by programmers, analysts and graphic designers. Having Graphical User Interfaces(GUI), is an essential requirement in the current day interface designing approach. When it comes to the usability of interface five major quality attributes must be considered.

- How easy it is for users to accomplish basic tasks the first time they encounter the interface.
- Ones the users have learnt the design, how quickly can they perform tasks.
- When users return to the design after some period without using it, how easily can they reestablish the proficiency.
- How many errors do users make, how severe are these errors and how easily can they recover from those errors.
- How pleasant is it to use the design.

The language used in the system should be familiar to the user, rather than in system-oriented terms. Usage of better error messages is a must, so that error prone situations can be minimized. When traversing through a series of windows to perform a task, users need not remember information from one window to another. Relevant help and documentation may be necessary. It should list down the required steps to be carried out very clearly and should not be too complex.

3.6.3 Design Strategy

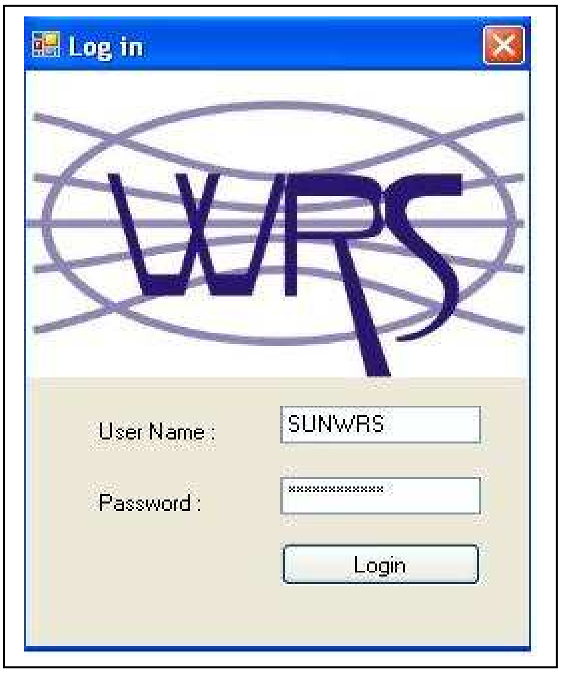
Presentation logic layer implementation was focused on developing user friendly interfaces. To reduce the error prone situations error messages were used appropriately. Buttons with self explanatory terms were used to interact with the system. Technical or system oriented words were not used. As much as possible options were provided through combo boxes, without asking the user to type them manually. These combo boxes were populated only with relevant data at the run time.
Some of the eye pleasing standard colors were used so that the user can work with the system for a long time without any discomfort.
Whenever there is an existence of master-detail functionality windows were designed to show these relationships using master-detail tab form windows.

Reports generated from this system are also a deliverable of this system. Management and other technical staff heavily depend on these reports. Relevant logos and report headers were used in both dynamically generated and the persistent reports.

Some of documents like shipment instructions generated through this system are direct deliverables to the end users. Usage of MDI window concept helps the user to easily interact with multiple windows at once.

3.6.4 Sample Screen shots

Some of the sample screen shots are provided below to have a look and feel of the Freight management system. Rest of the user interfaces is included in the Appendix C under user document section.



User Name : SUNWRS

Password : ************

Figure: 3. 12 Login form of the system

When the user loads the application within a few milliseconds login form shown in figure 3.12 is displayed. User can login to the system by providing valid username and password. For invalid username and password appropriate error messages will be provided.

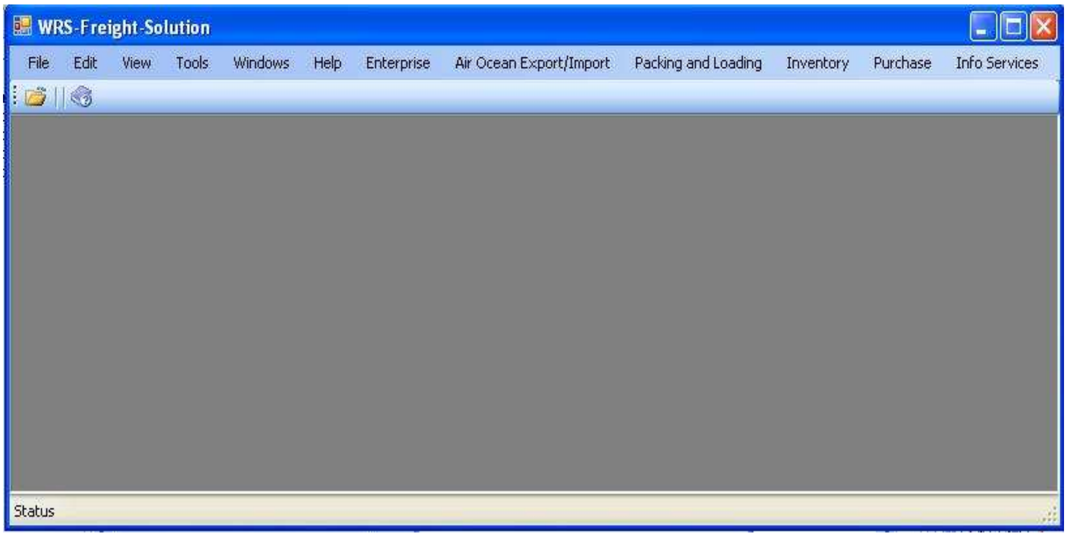
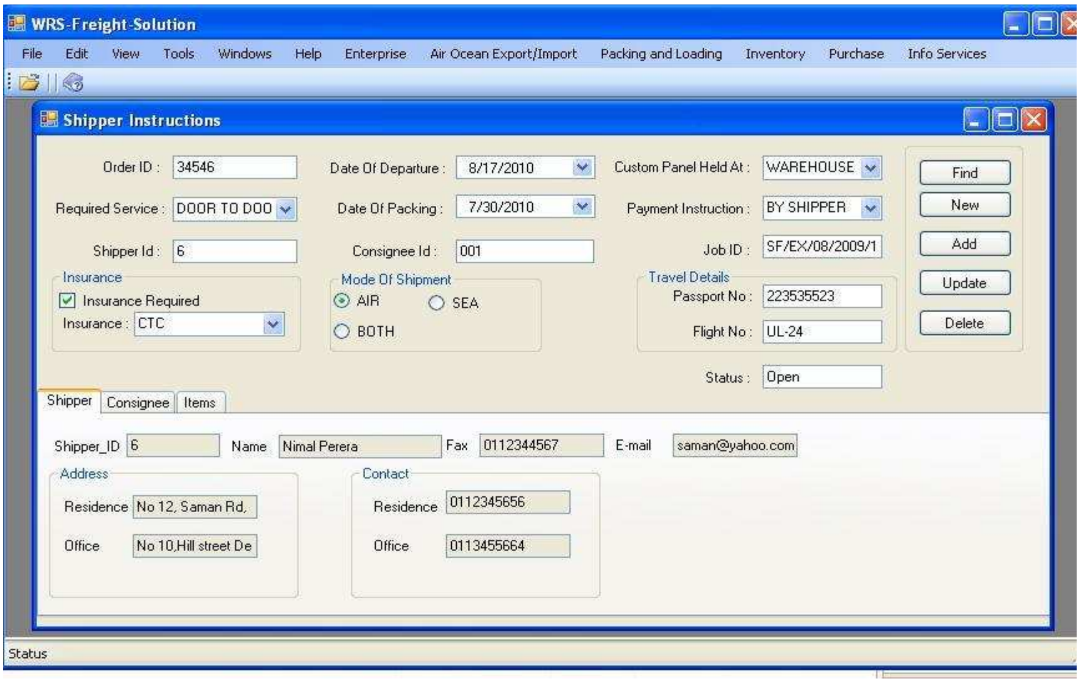


Figure: 3. 13 WRS-MDI form window

Upon successful login into the system, Main menu form shown in figure 3.13 will be displayed to the user. Menu Items are arranged according to the main system functionality. User can select any of the windows from the dropdown list provided that the user is authorized to access them. Forms which are not accessible to the user will be grey out. This application provides role based user authentication.



Order ID : 34546

Date Of Departure : 8/17/2010

Custom Panel Held At : WAREHOUSE

Required Service : DOOR TO DOO

Date Of Packing : 7/30/2010

Payment Instruction : BY SHIPPER

Shipper Id : 6

Consignee Id : 001

Job ID : SF/EX/08/2009/1

Insurance

☑ Insurance Required

Insurance : CTC

Mode Of Shipment

● AIR ○ SEA

○ BOTH

Travel Details

Passport No : 223535523

Flight No : UL-24

Status : Open

Shipper_ID 6 Name Nimal Perera Fax 0112344567 E-mail saman@yahoo.com

Address

Residence No 12, Saman Rd,

Office No 10,Hill street De

Contact

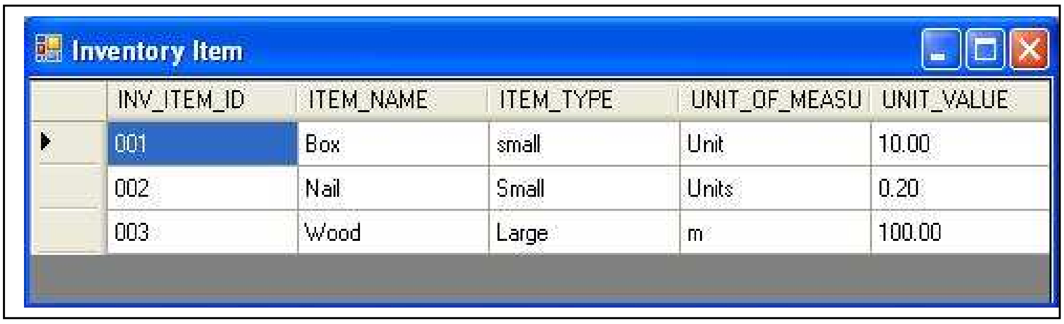
Residence 0112345656

Office 0113455664

Figure: 3. 14 Shipper Instructions window

Figure 3.14 illustrates the shipper instruction user interface. This is a master-detail form window. Details of the freight job are given in the header section. First tab shows the Shipper details attached to this job and the second tab is the Consignee Detail. Third tab is for the items to be shipped. Appearance of this tab is shown in the user document section.

Sample table of the Overview-Inventory Items is given in the figure 3.15 below. Overview tables in the system are read only and if the user needs to edit or delete the record, should click right mouse button on top of the row and select "Show Detail" in the context menu. A user who has permission to view these records can do so and users with edit rights to this table can only modify the Inventory Item details.



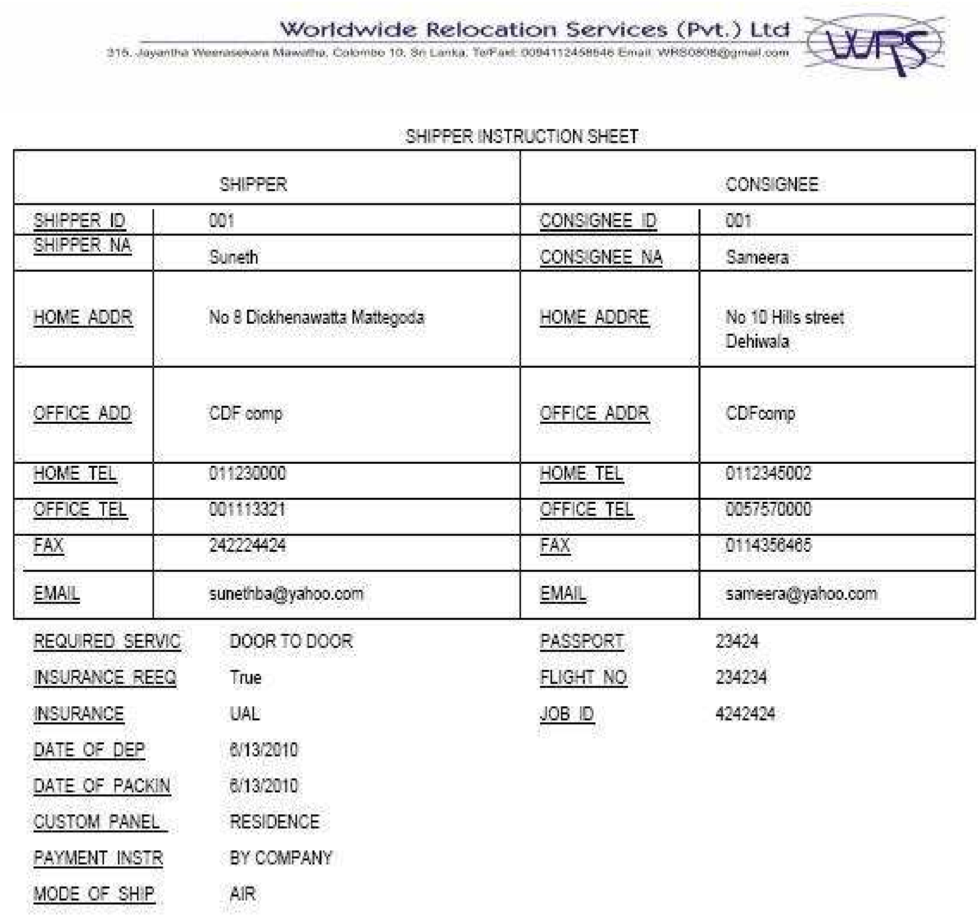
Inventory Item

	INV_ITEM_ID	ITEM_NAME	ITEM_TYPE	UNIT_OF_MEASU	UNIT_VALUE
▶	001	Box	small	Unit	10.00
	002	Nail	Small	Units	0.20
	003	Wood	Large	m	100.00

Figure: 3. 15 Overview-Inventory Item Window

Figure 3.16 is sample report generated through the system. It shows the details of a valid shipment instruction Sheet.

These documents are exported to a particular location in the local hard drive and user can later use these reports.



Worldwide Relocation Services (Pvt.) Ltd

315, Jayantha Weerasekara Mawatha, Colombo 10, Sri Lanka. Te/Fax: 0094112458546 Email: WRS0808@gmail.com

SHIPPER INSTRUCTION SHEET

SHIPPER		CONSIGNEE	
SHIPPER ID	001	CONSIGNEE ID	001
SHIPPER NA	Suneth	CONSIGNEE NA	Sameera
HOME ADDR	No 8 Dickhenawatta Mattegoda	HOME ADDRE	No 10 Hills street Dehiwala
OFFICE ADD	CDF comp	OFFICE ADDR	CDFcomp
HOME TEL	011230000	HOME TEL	0112345002
OFFICE TEL	001113321	OFFICE TEL	0057570000
FAX	242224424	FAX	0114356465
EMAIL	sunethba@yahoo.com	EMAIL	sameera@yahoo.com

REQUIRED SERVIC	DOOR TO DOOR	PASSPORT	23424
INSURANCE REEQ	True	FLIGHT NO	234234
INSURANCE	UAL	JOB ID	4242424
DATE OF DEP	6/13/2010		
DATE OF PACKIN	6/13/2010		
CUSTOM PANEL	RESIDENCE		
PAYMENT INSTR	BY COMPANY		
MODE OF SHIP	AIR		

Figure: 3. 16 Shipment instruction report

CHAPTER 4: IMPLEMENTATION

4.1 INTRODUCTION

The objective of this chapter is to give the reader an understanding about the aspects of the implementation state. The following will be focused in the chapter:

- Hardware/Software used in the implementation environment
- Physical implementation of the three-tier Architecture
- Modularized implementation
- Source code
- Creation of standard deployment project

4.2 DEVELOPMENT ENVIRONMENT

Hardware requirements of the system are given in below.

Development Environment	Testing Environment
Intel Pentium IV(or equivalent) 2.4 MHz processor or higher	Intel Pentium IV(or equivalent) 2.4 MHz processor or higher
512 MB RAM	Minimum 128 MB RAM
Hard disk with free space of at least 2GB to install the Visual .NET platform and other database software	Suitable Graphic Card
CD ROM Drive	

Table 4. 1 Hardware Requirements

Software requirements are given in below table,

Development Environment	Testing Environment
Windows XP Service Pack 2	Windows XP Service Pack 2
.NET Frame work 2.0	My SQL Server 4.1
Microsoft Visual Studio 2005	.NET Frame work 2.0

My SQL Server 4.1	
MyODBC 3.51.connector	
Adobe Photoshop	

Table 4. 2 Software Requirements

4.3 INTRODUCTION TO SOFTWARE USED IN DEVELOPMENT

WRS Freight Management System was developed using the C# programming language and the very popular Visual Studio .NET Development Environment.

My SQL database was used as the database and the My SQL Server used to access the Database. Each software component was installed separately and configured appropriately to setup the development environment.

4.3.1 Microsoft Visual Studio

Microsoft Visual Studio 2005 provides a complete set of development tools to develop data focus client applications. It provides a choice to the user to select a suitable programming language out of Visual Basic, C#, C++ and J#. Visual Studio also facilitates the sharing of tools among these languages. It helps in developing mixed language solutions. The IDE programmers helps to create run, test and debug C# programs, thereby reducing the time it takes to produce a working system.

This platform provides support for building Console applications, desktop applications, ASP.NET web applications, XML web services and mobile applications. Visual Studio includes a code editor supporting the intelliSense as well as code refactoring.

4.3.2 C# Programming Language

Out of the languages provided by Visual Studio IDE, C# has been selected as the programming language for this project. C# was developed by Microsoft within the .NET initiative. C# is standardized by ECMA International as the ECMA-334 standard and by ISO/IEC as the ISO/IEC 23270 standard.

C# is an event driven fully object oriented, visual programming language. Further C# has roots in C, C++ and Java, adapting the best features of each and adding new features on its own. Several C# features aid in the construction of robust and durable applications.

Garbage Collection automatically reclaims memory occupied by unused objects Exception handling provides a structured and extensible approach to error detection and recovery.

The type safe design of the language makes it impossible to read from not initialized variables, to index arrays beyond their bounds, or to perform unchecked type casts.

Standard C# libraries that come with the IDE provides a valuable support to the developer to perform standard functionality. Therefore, user can focus on implementing business requirement identified at the analysis stage without implementing any housekeeping tasks.

4.3.3 SQL

SQL stands for Structured Quarry Language. SQL is a standard computer language for accessing and manipulating databases. SQL statements are used to insert, select, update and delete data in the database. SQL works with most of the relational databases. Most of the SQL databases also have their own proprietary extensions in addition to the SQL standards. [Ryan K. Stephens].

4.3.4 MySQL database server

The MySQL database server is the most popular open source database management system. Currently, this system is developed, distributed and supported my Sun Microsystems, Inc.

MySQL is primarily RDBMS and therefore initially operates using command line. Users of MySQL need to download GUI tools from different vendors. Recently the official MySQL workbench was released and now the users can experience the user-friendly interface it provides. This has become the database of choice for the new generation application built on the LAMP stack.(Linux, Apache, MySQL, PHP/Python) In addition, OLE DB and ODBC providers exist for MySQL data connection in the Microsoft environment. A MySQL .NET native provider is also available, which allows the native MySQL to .NET access without the need of OLE DB. MySQL is most commonly used for Web applications and for embedded applications. It has become a popular alternative to proprietary database systems because of its speed and reliability. [WWW7]

4.3.5 Crystal Reports

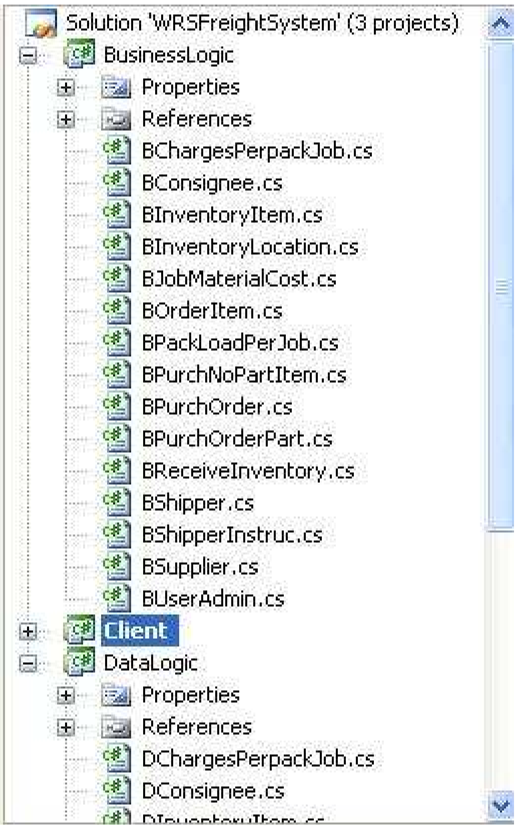
The product was originally created by Crystal Services inc. for their accounting software. The crystal reports allow users to graphically design data connections and report layouts. Users can select and link tables from a wide variety of data sources, Including Microsoft Excel Spreadsheets, Oracal database. It supports the data retrieval from Open Source databases like MySQL.

Both fields and formulae have a wide variety of formatting options available, which can be applied absolutely or conditionally. Crystal Reports has been part of Visual Studio since 1993, and is now the standard for reporting in Visual Studio 2005.[WWW8]

Reports generated in this system are developed using Crystal Reports.

4.4 THREE TIER ARCHITECTURE IN IMPLEMENTATION

System was developed according to the three tier architecture



```
Solution 'WRSFreightSystem' (3 projects)
  BusinessLogic
    Properties
    References
    BChargesPerpackJob.cs
    BConsignee.cs
    BInventoryItem.cs
    BInventoryLocation.cs
    BJobMaterialCost.cs
    BOrderItem.cs
    BPackLoadPerJob.cs
    BPurchNoPartItem.cs
    BPurchOrder.cs
    BPurchOrderPart.cs
    BReceiveInventory.cs
    BShipper.cs
    BShipperInstruc.cs
    BSupplier.cs
    BUserAdmin.cs
  Client
  DataLogic
    Properties
    References
    DChargesPerpackJob.cs
    DConsignee.cs
    DInventoryItem.cs
```

Figure: 4. 1 Organization of system components

Distribution of the System Logic among the three layers namely Client tier, Business tier and Database tier helped the implementation clear and the system more understandable. This will be advantageous in future upgrades of the system. For each entity identified in the system there is a logical class in each layer. For each persistent Entity there is a table in the database. Code to fetch the data in this table is done by the relevant Class in the Data layer. Sample Data accessing code is given below in steps. Connection to the database will be established using the following code.

```
using BusinessLogic;
using System.Data;

public class DShipper
    {

        private System.Data.Odbc.OdbcConnection OdbcCon;
        private System.Data.Odbc.OdbcCommand OdbcCom;


        private void OpenCnn()
            {
            // initialise connection
                   String cnnStr = "DRIVER={MySQL ODBC 3.51
            Driver};SERVER=" + "" + ";PORT=" + "" +
            ";DATABASE=" + "WRSFreight" + ";UID=" + "root" +
            ";PWD=" + "root" + ";OPTION=3";
                  OdbcCon = new
                  System.Data.Odbc.OdbcConnection(cnnStr);

                  // open connection
                  OdbcCon.Open();
            }
        private void CloseCnn()
            {
            // close connection
            OdbcCon.Close();
            }
      }
```

Once the connection is established existing data will be fetched using the following code and the data will be sent to calling object.

```
public DataSet Find(String argStr)
    {
      DataSet ds = null;

      try
      {
        OpenCnn();
        String selectStr = "select * from " + thisTable
      + " where SHIPPER_ID = '" + argStr + "'";
        OdbcCom = new
System.Data.Odbc.OdbcCommand(selectStr, OdbcCon);
        da = new
System.Data.Odbc.OdbcDataAdapter(selectStr, OdbcCon);
        OdbcDR = OdbcCom.ExecuteReader();
        ds = new DataSet();
        da.Fill(ds, thisTable);
        CloseCnn();
      }
      catch (Exception e)
      {
        String Str = e.Message;
      }
      return ds;
    }
```

For each Entity there is set of predefined attributes and the basics add, update, delete and quarry methods are done on those attributes. To Support those operations there is a reusable Attribute list and Variable passing method in each operation that was designed as follows.

```
//Shipper table Attribute List
    private const String thisTable = "tblShipper";
    private const String shipper_Id = "SHIPPER_ID";
    private const String shipper_name = "SHIPPER_NAME ";
    private const String home_address = "HOME_ADDRESS";
    private const String office_address = "OFFICE_ADDRESS";
    private const String home_tel = "HOME_TEL";
    private const String office_tel = "OFFICE_TEL";
    private const String fax = "FAX";
    private const String email = "EMAIL";
private String BuildAddString(BusinessLogic.BShipper shipper)
        {
            // these are the constants of the filed names as
            // set in the top of this module.
            strTable = "Insert into " + thisTable;
            strFields = " (" + shipper_Id +
            "," + shipper_name +
            "," + home_address +
            "," + office_address +
            "," + home_tel +
            "," + office_tel +
            "," + fax +
            "," + email + ")";

            //these are the attributes of the
            //Shipper business object.
            strValues = " Values ( '" + shipper.Shipper_id +
            "' , '" + shipper.Shipper_Name +
            "' , '" + shipper.S_Residence_Add +
            "' , '" + shipper.S_Office_Add +
            "' , '" + shipper.S_Residence_Tel +
            "' , '" + shipper.S_Office_Tel +
            "' , '" + shipper.S_Fax+
            "' , '" + shipper.S_Email + "' )";

            insertStr = strTable + strFields + strValues;
            return insertStr;

        }
```

Following code line is to add a new record to the System.

```
public void Add(BusinessLogic.BShipper shipper)
        {

            String str = BuildAddString(shipper);

            OpenCnn();

            //Open command option
            OdbcCom = new System.Data.Odbc.OdbcCommand(str,
            OdbcCon);

            //execute connection
            OdbcCom.ExecuteNonQuery();

            // close connection
            CloseCnn();

        }
```

Logic written in the Business module to perform the basic operations is very simple and straightforward. Since there is no complex logic in these methods any module that needs these basic operations can reuse them. This strategy facilitates the other developers to do future modifications without much complexity. Development strategy used here is the implementation of low coupling and high cohesiveness in the system.
Logic to add a new record in the Business layer is given below.

```
public class BShipper
    {
        private DataLogic.DShipper ShipperData;

        public BShipper()
            {
                  ShipperData = new DataLogic.DShipper(this);
            }

public void Add()
        {
            ShipperData.Add(this);
        }
```

Client tier is designed to distribute the data fetched through the Business logic.
Fetching existing data, add new records, update and delete is initiated by the objects in the client tier. Most of the time code was written inside try catch blocks. By this way System will not crash if there is an exception during the run time.
Sample code is given below.

```
public partial class frmCShipper : Form
    {
        BusinessLogic.BShipper shipper;

        private void btnFind_Click(object sender, EventArgs e)
            {
            String shipperID = txtShipperid.Text.ToString();
            FindRecords(shipperID);

            }
```

```
//Fetch Existing data
    private void FindRecords(String Shipperid)
    {
      try
      {
        shipper = new BusinessLogic.BShipper();
        DataSet ds = shipper.Find(Shipperid);
//Add the records in the dataset to the client Objects
        foreach (DataRow rows in ds.Tables[0].Rows)
          {
             txtShipperid.Text = 
rows["SHIPPER_ID"].ToString();
             txtName.Text = 
rows["SHIPPER_NAME"].ToString();
             txtResidenceAdd.Text = 
rows["HOME_ADDRESS"].ToString();
             txtOfficeAdd.Text = 
rows["OFFICE_ADDRESS"].ToString();
             txtResidenceTel.Text = 
rows["HOME_TEL"].ToString();
             txtOfficeTel.Text = 
rows["OFFICE_TEL"].ToString();
             txtFax.Text = rows["FAX"].ToString();
             txtEmail.Text = rows["EMAIL"].ToString();

          }


      }
      catch (Exception err)
      {
        MessageBox.Show(err.Message.ToString());
      }


    }
```

Attributes in the classes are not directly accessible to the outside world. They can only access through the public get and set methods. Also if the user tried to save the records without valid data, appropriate error messages will be produced.

```
         private String shipper_id;
         private String shipper_name;
         private String s_residence_add;
         private String s_office_add;
         private String s_residence_tel;

       public String S_Residence_Add
         {
             get
             {
                 return this.s_residence_add;
             }
             set
             {
                 this.s_residence_add = value;
                 if (this.s_residence_add == "")
                 {
                     throw new Exception("Please provide
residence address...");
                 }

             }
         }
```

Report Generation module,

This module handles process related to the report generation. The process includes generation of different types of reports such as Shipment instruction generation, Pre alert notices, Invoices and also some reports that help in management decisions.

Profit share for the job, Total registered Customers are among the managing reports.

Crystal Repots was used as the Reporting tool. Facility to preview the reports during the development greatly helped to reduce the time in report structuring.

Selected sections of the code that does report generation are given below.

```
private void btnShipmnetInstructions_Click(object sender,
EventArgs e)
        {
            cryRpt = new ReportDocument();
            cryRpt.Load("E:\\BIT-
PROJECT\\WRSFreightSystem\\WRSFreightSystem\\Client\\Shippers
Instruction.rpt");
            crystalReportViewer1.ReportSource = cryRpt;
            crystalReportViewer1.Refresh();
        }
```

For future references and printing purposes system generated reports will be exported as pdf documents.

```
private void btnExptShipIns_Click(object sender, EventArgs e)
        {
            try
            {
                ExportOptions CrExportOptions;
                DiskFileDestinationOptions
CrDiskFileDestinationOptions = new
DiskFileDestinationOptions();
                PdfRtfWordFormatOptions CrFormatTypeOptions =
new PdfRtfWordFormatOptions();
                CrDiskFileDestinationOptions.DiskFileName =
"E:\\BIT-
PROJECT\\WRSFreightSystem\\WRSFreightSystem\\RoprtArchive\\Ship
perInstructionsRpt.pdf";
                CrExportOptions = cryRpt.ExportOptions;
                {
                    CrExportOptions.ExportDestinationType =
ExportDestinationType.DiskFile;
                    CrExportOptions.ExportFormatType =
ExportFormatType.PortableDocFormat;
                    CrExportOptions.DestinationOptions =
CrDiskFileDestinationOptions;
                    CrExportOptions.FormatOptions =
CrFormatTypeOptions;
                }
                cryRpt.Export();
            }
            catch (Exception ex)
            {
                MessageBox.Show(ex.ToString());
            }
        }
```

CHAPTER 5: EVALUATION

5.1 INTRODUCTION

The Objective of this chapter is to give the reader an understanding about the different testing strategies used to test the developed WRS Freight Management system against the proposed system. The problems revealed in the testing process and the lessons learnt will be presented in this chapter.

5.2 SOFTWARE TESTING

Testing represents the ultimate review of specification, design and implementation. This is one of the most important phases in the system development life cycle. Both validation (Whether expectation of the client has met or not) and verification (whether the program confirms to its specification or not) has to be done. This process will uncover the errors and ensure that defined input test data will produce expected results. Any software product contains a testing phase in its development life cycle. During the testing phase software product is given a set of inputs and the outputs are observed. There are several categorizations for the types of testing. Ultimate goal of all these testing strategies is to produce error free software to the interested parties.

One of the broad categorization of the testing is,

- Black Box testing
- White Box testing

5.2.1 Black box Testing

It takes an external perspective of the test object to derive test cases. These tests can be functional or non-functional, though usually functional. The test designer selects valid and invalid input and determines the correct output. There is no knowledge of the test object's internal structure.

5.2.2 White Box Testing

White box testing uses an internal perspective of the system to design test cases based on the internal structure. It requires programming skills to identify all paths through the software. The tester chooses test case inputs to exercise paths through the code and determines the appropriate outputs. Since the tests are based on the actual implementation, if the implementation changes, the tests probably may need to change.

5.3 TESTING STRATEGY

In Object Oriented Software Engineering the test cases are built according to the use cases. As the analysis and the design progress through the life cycle, testing plans and procedures are created in order to realize all the possible flows in the system.

Usually large systems are built out of small components and each should be tested separately. Sometimes customers will purchase only subset of components provided by the vendor. In such a situation, the system components should not be interdependent. After successful testing of the components, the system can be tested as a single unit. Testing should be carried out incrementally and parallel to the implementation.

Testing can be described in five stages. Testing process of this system was carried out in five stages. As described testing was carried out starting from smallest unit incrementally and spirally.

Test plan consisted of three steps as component testing, integration testing and acceptance testing. Each phase is described in the figure 5.1 given below.

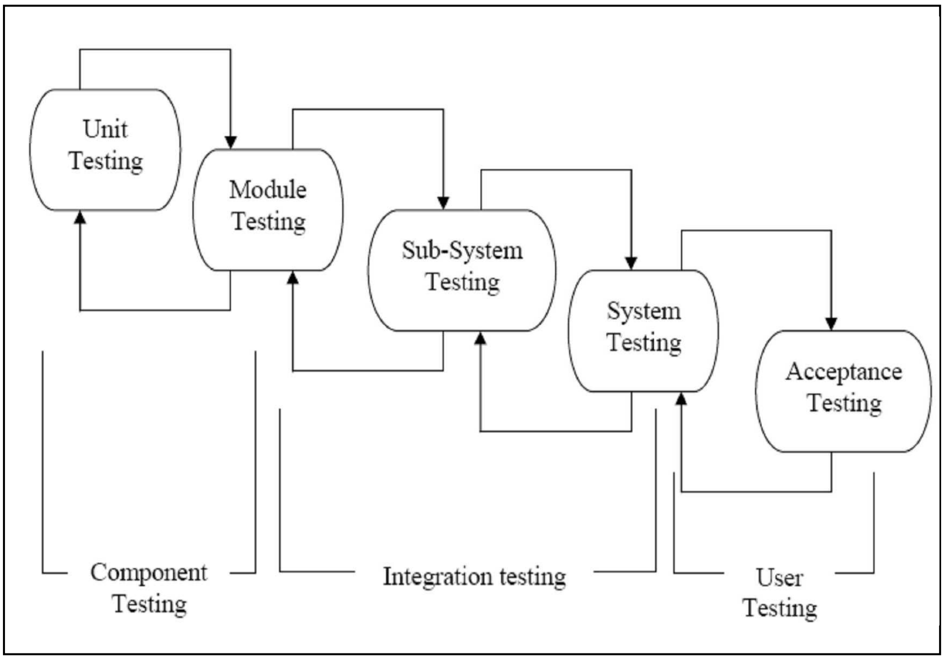


Figure: 5. 1 General Testing process of the system

5.3.1 Component testing

Individual components were tested in order to ensure that they operate correctly. Each component is tested separately. White box testing or black box testing can be used to test the individual components. In this system test scenarios such as user authentication, basic operations like add, update and delete was tested under this section. Loose coupling provided by this system greatly helped in testing each component separately. Client side validation was also done in this step. Majority of errors were found under this section and were corrected.

5.3.2 Integration testing

When individual error corrections for each module were finished successfully integration testing of the system was carried out. It was tested for the functionality that flows across those modules and observed how the input in one component behaves in another component. Some incompatibilities in the data types and in the lengths used in client windows were identified and were corrected accordingly. New relations were made among the data tables to ensure the data integrity. During the functional testing input data with both correct and anomalous were used. There were several issues in user management during the authentication of users to perform task based on their specific role. They were identified and corrected.

5.3.3 User Acceptance testing

One key source of information about problems and the impact of the newly implemented system are the end users. After the integration testing in the system was done, the regression testing was carried out. Subsequently, the system was installed in the client environment and user acceptance testing was done at the client site. Company employees have been used as the real time testers of the system. The aim of this was to make sure the system functions as expected before the system put into real production environment.

User feedback was gathered based on functional and non functional requirements. Following strategies used as the fact gathering techniques from the end users.

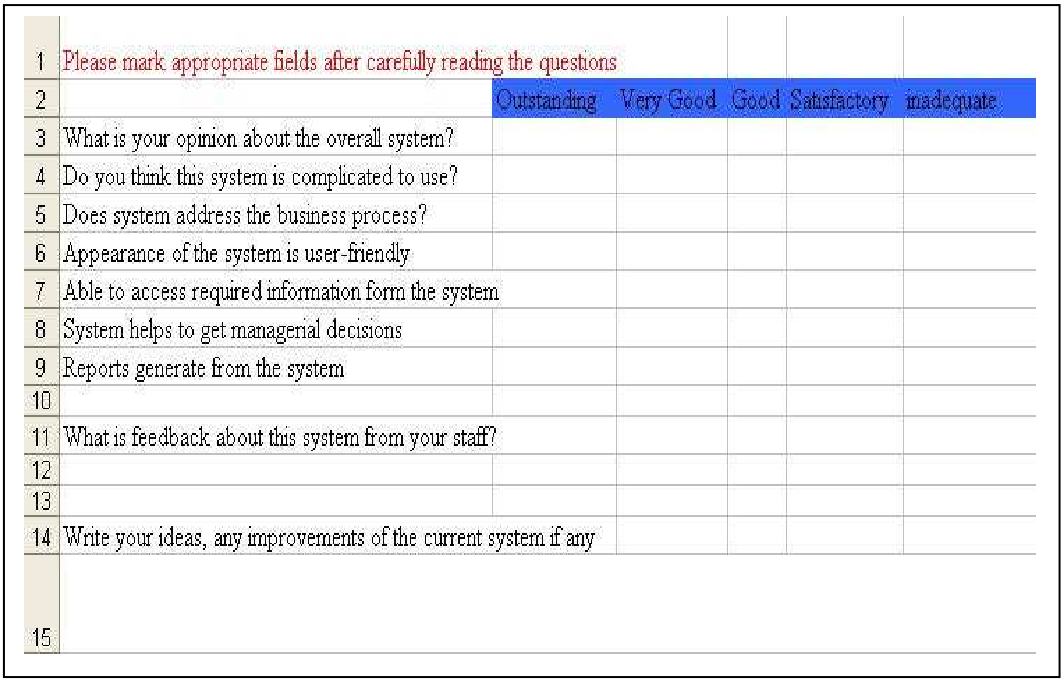
- User workshops
- Site visits
- Mailing actions
- Questionnaires
- Phone calls

5.3.4 User feedback analysis

After demonstration of basic flows in the system and the functionality that the system provides, users were asked to test the system. Feedback from the customer was taken to consideration and some interface modifications were done.

User feedback was gathered from two separate user communities, namely Managers and Operational staff. Two different sets of questionnaire were given to them.

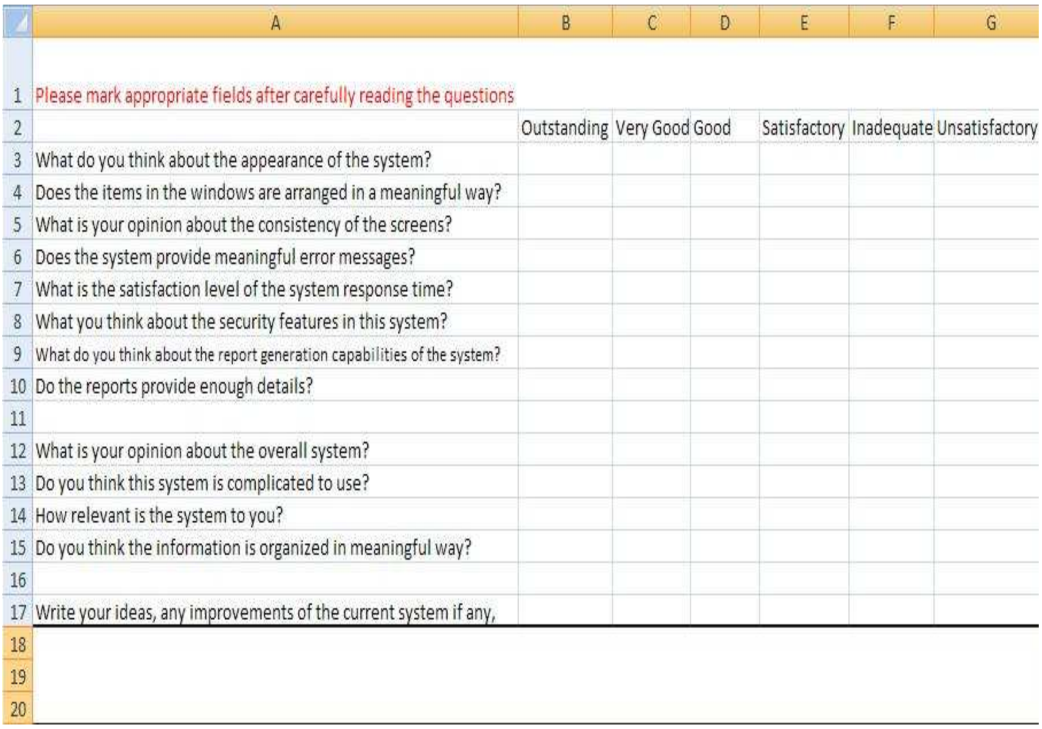
The managerial user feedback was gathered using the questionnaire given in figure 5.2 below.



1	Please mark appropriate fields after carefully reading the questions					
2		Outstanding	Very Good	Good	Satisfactory	inadequate
3	What is your opinion about the overall system?					
4	Do you think this system is complicated to use?					
5	Does system address the business process?					
6	Appearance of the system is user-friendly					
7	Able to access required information form the system					
8	System helps to get managerial decisions					
9	Reports generate from the system					
10						
11	What is feedback about this system from your staff?					
12						
13						
14	Write your ideas, any improvements of the current system if any					
15						

Figure: 5. 2 managerial user feedback sheet

The Feedback from the operational staff was gathered using the questionnaire given in figure5.3.



	A	B	C	D	E	F	G
1	Please mark appropriate fields after carefully reading the questions						
2		Outstanding	Very Good	Good	Satisfactory	Inadequate	Unsatisfactory
3	What do you think about the appearance of the system?						
4	Does the items in the windows are arranged in a meaningful way?						
5	What is your opinion about the consistency of the screens?						
6	Does the system provide meaningful error messages?						
7	What is the satisfaction level of the system response time?						
8	What you think about the security features in this system?						
9	What do you think about the report generation capabilities of the system?						
10	Do the reports provide enough details?						
11							
12	What is your opinion about the overall system?						
13	Do you think this system is complicated to use?						
14	How relevant is the system to you?						
15	Do you think the information is organized in meaningful way?						
16							
17	Write your ideas, any improvements of the current system if any,						
18							
19							
20							

Figure: 5. 3 End user feedback sheet

Graphical representation of their feedback analysis for the functional and non functional requirements are shown in the figure 5.4

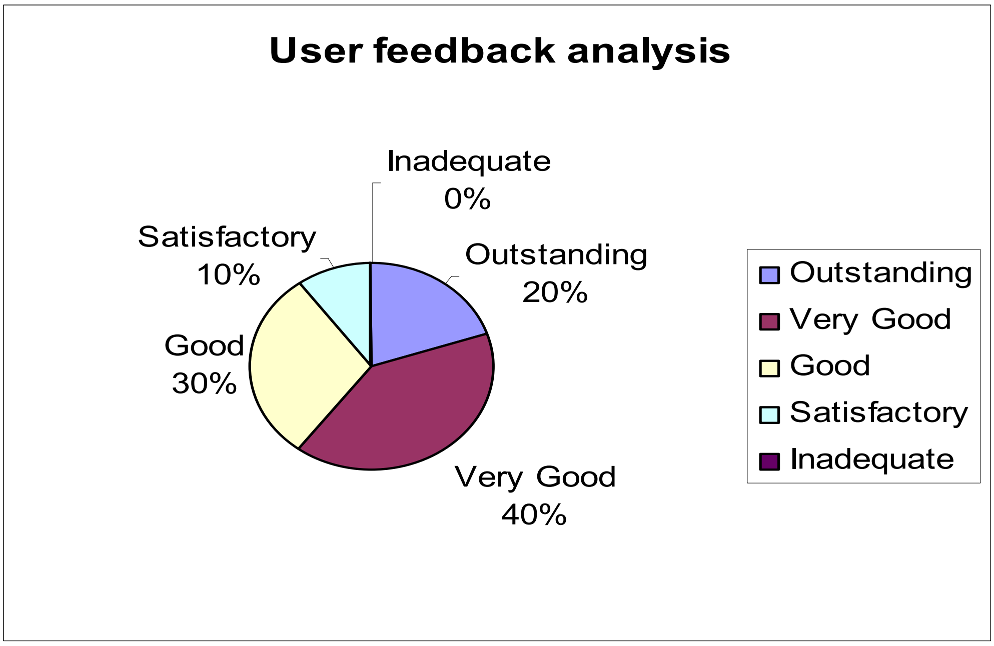


Figure: 5. 4 End user feedback analysis results.

Most of the current employees were very satisfied about the system and their main concern was the adaptation to the automated system from the existing manual system. User manual of the system was modified according to the user feedback. Detailed description of the user documentation is provided under appendix C.

5.4 DEFECTS IDENTIFICATION

Defects testing was done to find the areas where the system does not conform to its specification. The test cases were amid at identifying the errors in the system. When the defects are found, they must be analyzed and removed. This is called debugging. The functions identified in the analysis and the design phases were used to prepare the test cases and test data. Then the system was tested against the test data. Results obtained were compared with the test cases designed. This confirms the usability of the system.

5.5 FUNCTIONAL TESTING

Functional testing was carried out to find errors in the functional modules. This test was done to test whether all the functional requirements have been implemented. The sequential flow between the interface screens must also be tested to ensure that they perform as described in the design chapter.

During the functional flow testing, correct implementation of the industry specific standards were also observed. The result of these tests showed that the functionality had been implemented correctly.

5.6 REGRESSION TESTING

Regression testing was done in both the development environment and the client site to ensure that there are no faults in the system due to hardware problems. Response time of the system during the executions of various operations was observed.

5.7 OTHER TYPES OF TESTING

5.7.1Usability testing

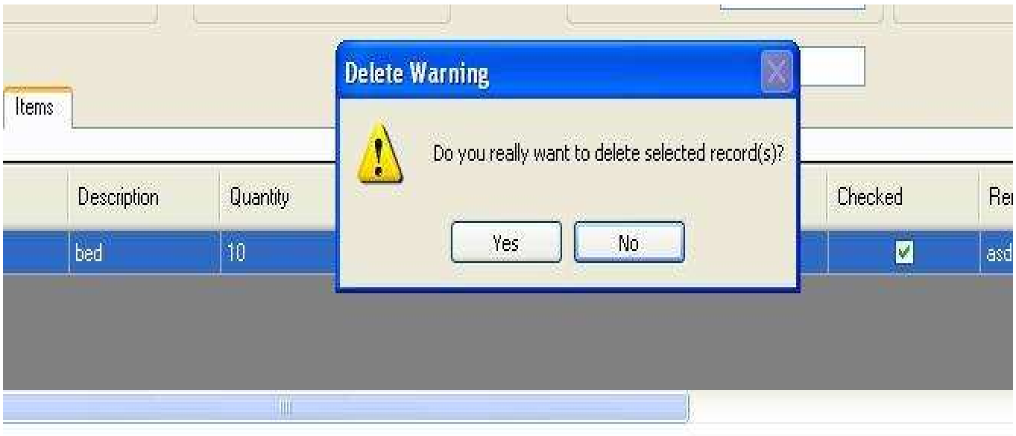
Some screen resolution issues in the client environment were different from the development environment. These were corrected after user feedback.

5.7.2 Error handling

The system was tested for the possible errors that can occur during the user inputs. The ability of the system in handling these situations and providing the appropriate error messages were examined afterwards. For an example in case of an invalid user login, the response of the system was examined.

5.7.3 Warnings and confirmation messages

System must provide meaningful warnings and confirmation messages. It is always important to handle the situations that the system cannot rollback an action performed by the user. For example deleting records must be always confirmed by the user. System was tested against these type of scenarios. Following figure 5.5 is an example of the confirmation message that appears when user tries to delete a record.



Do you really want to delete selected record(s)?

Figure: 5. 5 Warning message

5.8 TEST CASES AND PLANS

Test cases were generated considering the execution of each line of code. They were generated for both functional and nonfunctional requirements. Functional test cases were generated based on the functional requirements identified during the analysis phase. Exceptions that can occur in every functional flow was also considered in test case generation. User interface testing, load and stress testing, security and access control testing was done using nonfunctional test cases. Existing real world data from the manual system and randomly generated data was used as the test data to the system. A sample set of use cases was used to evaluate the login process given in the table 5.1 below.

Please refer to the Appendix E for the rest of the test data and the results that were obtained.

Test cases for the User Login

Test Case No	Test	Expected Result	Status
1	Login without providing details	Show login error message	Ok
2	Login with incorrect credentials	Show login error message	Ok
3	Login with correct credentials	Allow user login to the system	Ok

Table 5. 1 Test case for User Login

5.9 EVALUATION

At the end of the project life cycle it is very important to go through the activities performed in each phase. This is to identify the achievements and the problems encountered. It is also important to discuss about the project in terms of reliability, usability, efficiency and maintainability.

The objective of this system is to automate the business process carried out at WRS. The requirements of the proposed system were described in detail in the second chapter. Final outcome of the requirement analysis phase was fully implemented in this system.

It was decided to implement the system in .NET platform and the MySQL database was used as the data storage. Crystal reporting was used as the reporting tool of this system. The system was developed using the three-tier architecture. Standard development methodologies and the Object Oriented concepts were used in the system. After finishing the development, testing was performed according to the different testing protocols as described in the system testing section of this chapter. Finally, the system was installed at the client environment and the user acceptance testing was performed. Some of the non functional aspects of the system that has been tested are elaborated below.

5.9.1 Usability

System was adhered to the interface design standards. Sufficient amount of error handling was done to ensure the data integrity in the system. Users are guided with meaningful error and warning messages as and when necessary. User documentation was done with sufficient amount of details. Report generation process is very straightforward and easy to understand. User-friendly color themes were used in the system and constancy of the basic operations were achieved.

5.9.2 Reliability of the system

The system was developed in a very reliable and common platform. Therefore, the codes used in the implementation adhered to the coding standards and was formatted well. Solid architecture used during the development will be an advantage in future modifications. NET C# and MySQL configuration are powerful and efficient and hence data reliability was achieved.

5.9.3 Efficiency

System is deployable in standard personal computers without high-end hardware requirements. Response time of the system was minimized. Users can perform their work in a short period of time. Implementation of master-detail window concept minimized the amount of information user needs to remember from one window to another.

5.9.4 Maintainability

The system has a well defined set of classes. Loose Coupling and High Cohesiveness was maintained in those classes. Classes have required methods to operate on data. The classes and methods have very meaningful names. So the system can be updated easily. The code was written clearly and sufficient amount of comments were made. So the code can be easily understood by another user and change without much effort.

5.10 CRITICAL APPRAISAL

This project was initiated to develop an user-friendly system that is capable of handling the business processes carried out at WRS. Although it was initially planned to handle all the import, export and the purchasing flows which were conducted by WRS, later it was decided to limit the functionality. After the feasibility study it was decided to implement only the export handling section. Later on with the motive from the client it was decided to implement some of the sections in purchasing department since some of the activities performed in that process is a direct input to the export management.

The largest portion of the project schedule time was devoted to the system analysis phase. Interviewing different user categories, analyzing the existing documents in the manual system, and real time observations were among the different facts gathering techniques. All the gathered data were documented and the requirements were revised with the help of management at WRS.

Design of the system was done according to the Object Oriented concepts and three-tier Architecture. Some of the theories of Service Oriented Architecture were also kept in mind in order to implement smooth functional flow. Design was visualized as set of Entities and class diagrams using the Rational Rose tool. It made the implementation of the system more understandable. Sufficient amount of testing was done in order to make sure the system functioned as expected.

There were some deviations from the initially planned time schedules during the requirement analysis phase and the design phase due to geographical and communication barriers. Some extra time was spent during the implementation phase in order to meet the schedule. However, as a whole implementation of the system was successful.

Remaining facilities that has not been implemented by this phase of system development life cycle can be considered as future work and modifications to the system.

5.11 LESSONS LEARNT

Vast amount of knowledge and experience was gathered during the project. Some of the lessons learnt during the project are given below.

- ➢ The way the real world freight management process is carried out was understood.
- ➢ The mapping of the business process into the software development process, beginning from the project proposal to final system implementation and testing was learnt.
- ➢ During the Analysis phase and the feasibility study communication skills were improved and negotiation with people in different user categories was learnt.
- ➢ It was a great opportunity to work in a modern development platform like Visual Studio .NET and the well known MySQL database. Since there were some connectivity requirements in the database and the system vast amount of knowledge was acquired on how to integrate different systems to work as a unit.
- ➢ Large amount of technical knowledge was acquired in relation to C# programming language.
- ➢ As a summary, learnt how to analyze, design and implement a system within the given constraints.

CHAPTER 6: CONCLUSION

6.1 INTRODUCTION

The objective of this chapter is to summarize the system objectives and how they were achieved. Furthermore, the deviations from the original project plan and the unavoidable causes which affected the overall work are discussed. Finally the future enhancements that can be done to upgrade this system are outlined.

6.2 CONCLUSION

Motivation to computerize the freight management business activities at World Wide Relocation Services laid the path to develop this system. It was supposed to achieve two main categories of objectives. One was to develop an user-friendly system that can handle and guide the staff at WRS to perform their routine business activities, and the other was to provide an automated mechanism for the management at WRS to carryout their business management processes.

As mentioned in the previous chapter almost all the functionalities that expected by the system was implemented. Therefore by using this system users can carry out their activities efficiently and effectively. This system provides two types of reports. One is dynamically generated reports that helps in the operational activities. Management reports helps managers to get managerial decisions based on them. As a result management can easily monitor the business in a Birds' eye view. By considering all these facts it can be concluded that this system will have a positive impact on the business growth of WRS.

6.3 UNAVOIDABLE CAUSES AND EFFECTS

During the fact gathering phase initially, obtaining a clear idea of what is expected from the system was problematic. This was mainly due to inexperience in this business domain. Nevertheless by working with the staff at WRS and by interviewing them sufficient amount of knowledge on the business flow was obtained.

During the implementation phase there was some difficulty in identifying a good connector that can connect .NET platform with the database. Sufficient amount of time was spent on knowledge searching for this and finally it was achieved.

The next issue occurred during the deployment of the system in the client environment. Several software components were needed to install in the client machine in order to run and administrate the system. Sufficient amount of dick space was obtained after discussing with the client.

Purchasing module of the system was a requirement identified at the latter stage of the development. Therefore some of the functionalities in the purchasing module were not implemented during this time frame.

6.4 SUGGESTIONS TO IMPROVE THE SYSTEM

The current version of the system provides sufficient amount of facilities to perform the main business flow at WRS. Following items have been identified as possible future enhancements that can be easily integrated with the current system.

- Improve the purchasing functionality that has already been implemented in the system.
- Special module to handle the inventory at warehouses.
- Facilitate interaction with the parties at the receiving end of the goods by integrating a web module to the system.
- Automation of the import flow.
- A web based module that user can easily interact with the system and place their orders on their own.
- Provide a tracking mechanism for the freight goods, which route in the supply chain.
- Implementation of a human resource module.

REFERENCES

[Paul, 1998] Paul Bugden 1998. Freight Forwarding and Goods in Transit, Sweet & Maxwell Publishing Ltd. ISBN 978-0421609402.

[Booch, 1999] Booch Grady, Rumbaugh James, Jacobson Ivar. 1999. The Unified Modeling Language User Guide. Copyright 1999 by Pearson Education, Inc. (Law price Edition) ISBN -81-7808-5.

[Jacobson, 1999] Jacobson Ivar, Booch Grady, Rumbaugh James.1999. The Unified Software Development process. Addison Wesley Longman Inc 1999. (Law price Edition)

[Sommerville, 2000] Sommerville, Ian 2000. Software Engineering. Addison Wesley Publishing Ltd.

[Ryan K. Stephens] Ryan K. Stephens, R. R. Teach Yourself SQL in 21 Days, Second Edition. Macmillan Computer Publishing.

[WWW1] – Sri Lanka Freight Forwarders association. http://slffa.com/history/ [05/02/2010]

[WWW2] – http://ezinearticles.com/[22/03/2010]

[WWW3] – http://www.australia-migration.com/[02/04/2010]

[WWW4] – http://www.logisuite.com/[03/04/2010]

[WWW5] - http://www.softtruck.com/[03/04/2010]

[WWW6] – http://www.cms.gov/ [07/04/2010]

[WWW7] – http://www.mysql,com/[27/06/2010]

[WWW8] - http://www.msdn.microsoft.com/[04/07/2010]

APPENDIX A: DESIGN DOCUMENTATION

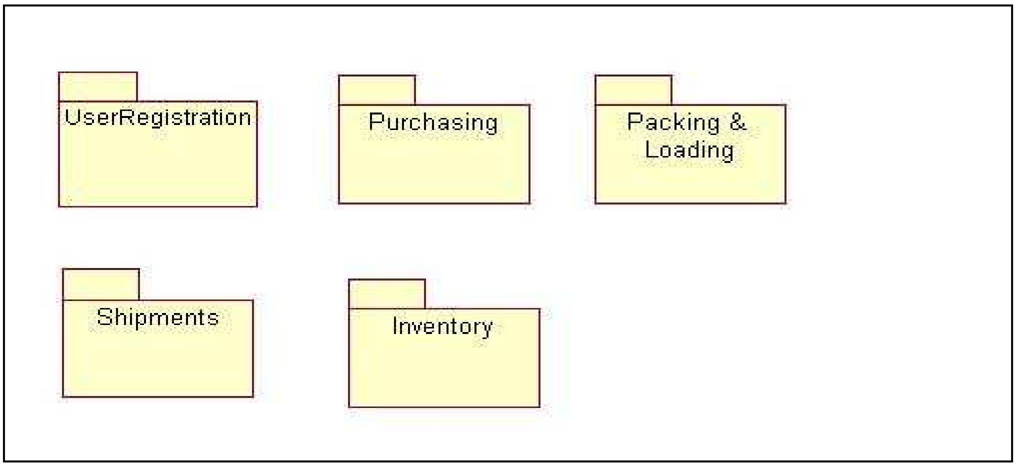


Figure A.1 Package view of the system

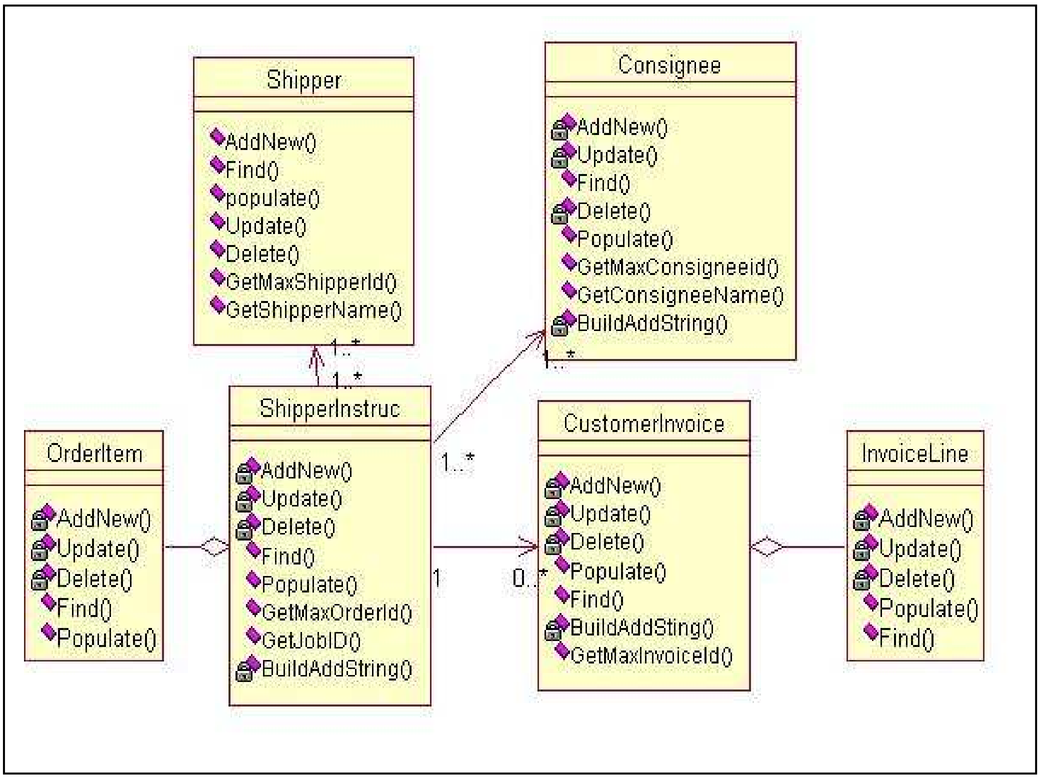


Figure A.2 Class diagram of the shipment instruction handling showing the operations of each class.

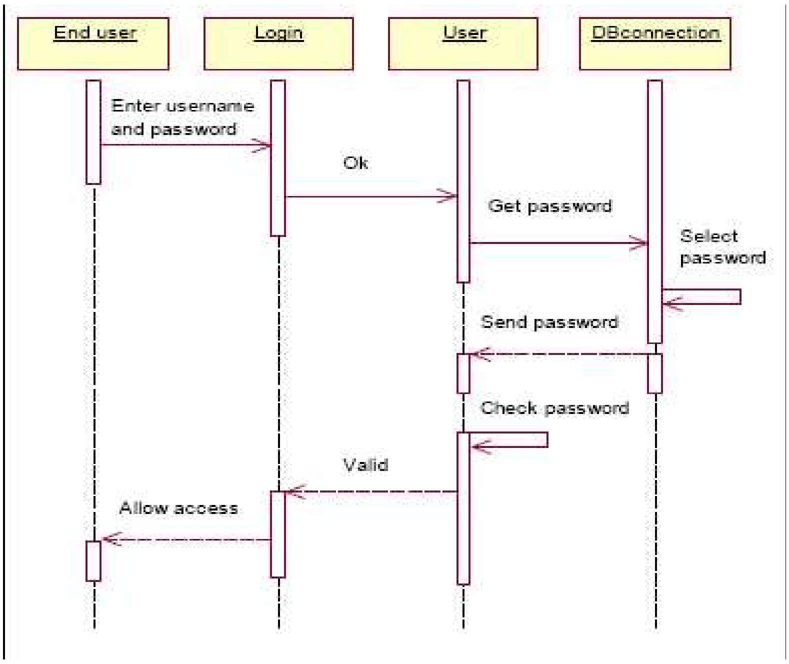


Figure A.3 Sequence diagram of user login

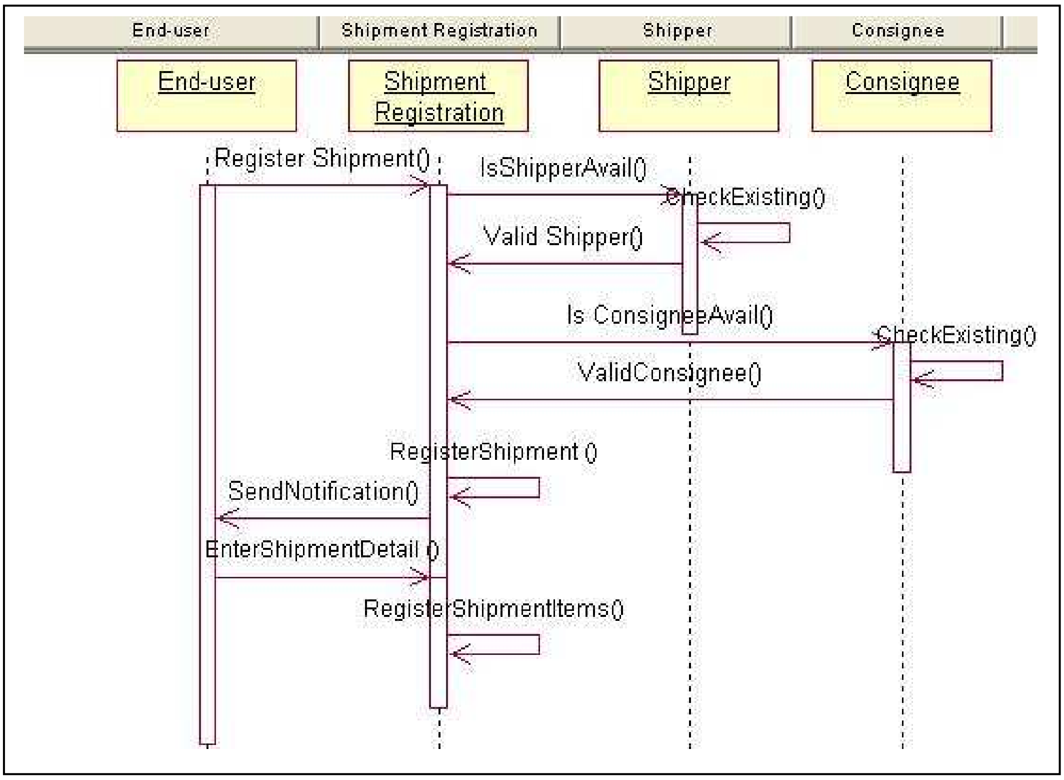


Figure A.4 Sequence diagram of shipment instruction generation

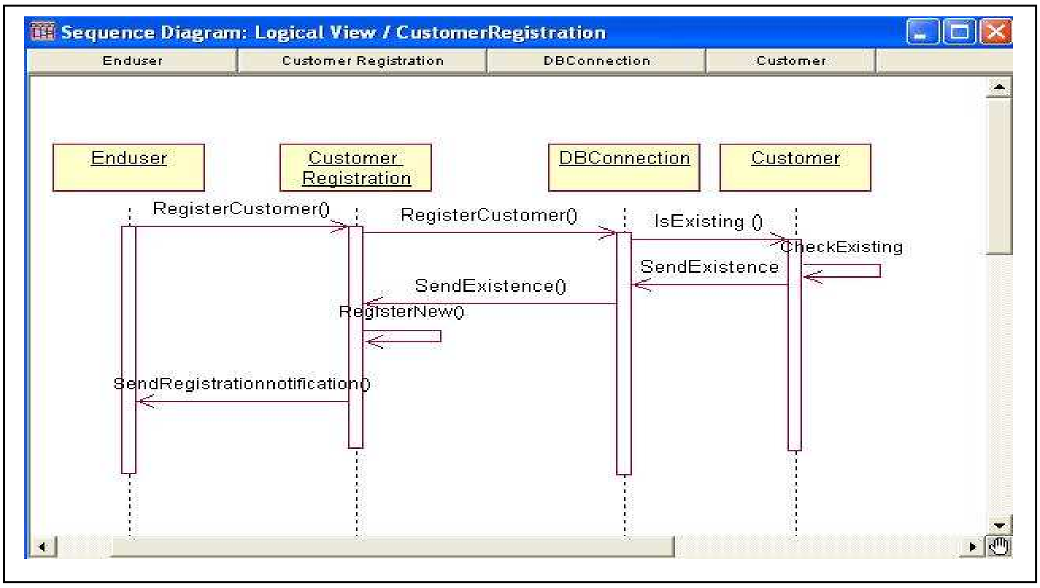


Figure A.5 Sequence diagram of customer registration.

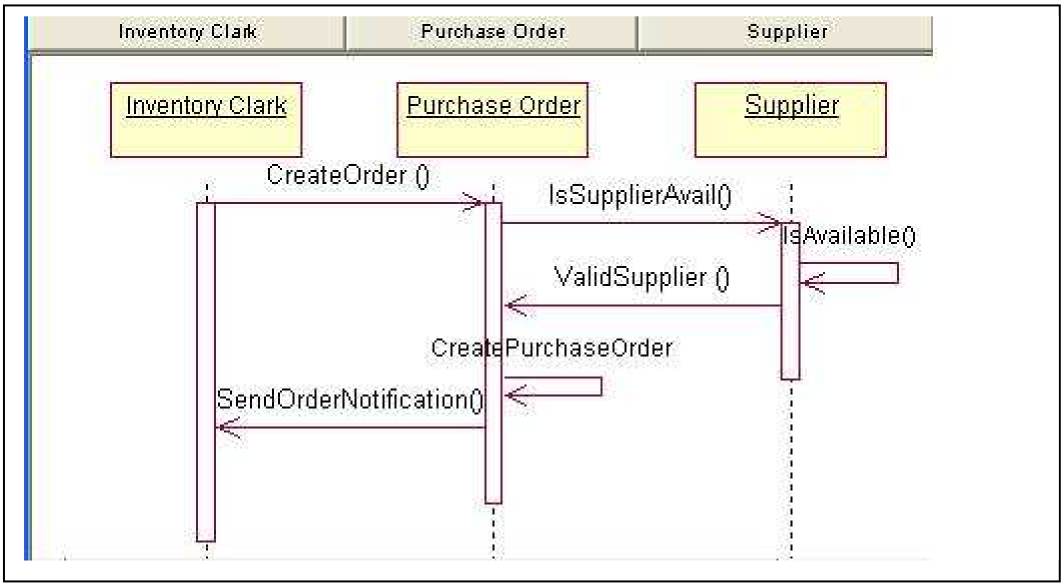


Figure A.6 Sequence diagram for purchasing process.

Database Design Structure

Table Name: tblUser

Description: User Details

Key	Name	Description	Type	Length	Allow Null
PK	USER_ID	User Id	char	12	No
	PASSWD	Password	char	12	No

Table Name: tblRole

Description: User Roles

Key	Name	Description	Type	Length	Allow Null
PK	ROLE_ID	User Id	char	10	
	PERMISSIONSET	Allowed permissions	char	100	
	ROLE_NAME	User Role name	char	20	

Table Name: tblUserRole

Description: Assign user to the Role

Key	Name	Description	Type	Length	Allow Null
PK/FK	ROLE_ID	User Id	char	10	No
PK/FK	USER_ID	Allowed permissions	char	100	No
	VALID_FROM_DATE	User Role name	char	20	
	VALID_TO_DATE				

Table Name: tblShipper

Description: Shipper Detail

Key	Name	Description	Type	Length	Allow Null
PK	SHIPPER_ID	Shipper Id	char	10	No
	SHIPPER_NAME	Shipper Name	char	200	No
	HOME_ADDRESS	Home Address	char	200	
	OFFICE_ADDRESS	Office Address	char	200	
	HOME_TEL	Residence contact Number	INT	10	
	OFFICE_TEL	Office contact number	INT	10	
	FAX	Fax	INT	10	
	EMAIL	E-mail	char	100	

Table Name: tblConsignee

Description: Consignee Detail

Key	Name	Description	Type	Length	Allow Null
PK	CONSIGNEE_ID	Consignee Id	char	10	No

	CONSIGNEE _NAME	Consignee Name	char	200	No
	HOME_ADDRESS	Home Address	char	200	
	OFFICE_ADDRESS	Office Address	char	200	
	HOME_TEL	Residence contact Number	INT	10	
	OFFICE_TEL	Office contact number	INT	10	
	FAX	Fax	INT	10	
	EMAIL	E-mail	char	100	

Table Name: tblsupplier

Description: Supplier Detail

Key	Name	Description	Type	Length	Allow Null
PK	SUPPLIER_ID	Consignee Id	char	10	No
	SUPPLIER _NAME	Consignee Name	char	200	No
	OFFICE_ADDRESS	Office Address	char	200	
	OFFICE_TEL	Office contact number	INT	10	
	FAX	Fax	INT	10	
	EMAIL	E-mail	char	100	

Table Name: tblsupplier

Description: Supplier Detail

Key	Name	Description	Type	Length	Allow Null
PK	SUPPLIER_ID	Consignee Id	char	10	No
	SUPPLIER _NAME	Supplier Name	char	200	No
	OFFICE_ADDRESS	Supplier Address	char	200	
	OFFICE_TEL	Supplier number	INT	10	
	FAX	Fax	INT	10	
	EMAIL	E-mail	char	100	

Table Name: tblShipperInstruc

Description: Shipper Instruction Information

Key	Name	Description	Type	Length	Allow Null
PK	ORDER_ID	Order Id	char	25	No
	REQUIRED_SERVICE	Required Service	char	10	
	DATE_OF_DEP	Date Of Departure	date	200	
	DATE_OF_PACKING	Date Of Packing	date	10	
	CUSTOM_PANEL_HELD	Customer Panel held Location	INT	10	
	PAYMENT_INSTRUCTION	Payment instructions	char	100	
	INSURANCE_REEQUIRED	Insurance Required	Boolean		
	INSURANCE	Insurance	char	10	
	MODE_OF_SHIP	Mode of Shipment	char	10	
	PASSPORT_NO	Passport Number	char	10	
	FLIGHT_NO	Flight Number	char	10	
FK	SHIPPER_ID	Shipper Id	char	10	No
FK	CONSIGNEE_ID	Consignee Id	char	10	No

Table Name: tblOrderItem

Description: Order Items

Key	Name	Description	Type	Length	Allow Null
PK	ITEM_NO	Item Number	char	25	No
	DESCRIPTION	Description	char	10	No
	QUANTITY	Quantity	date	200	No
	UNIT_VALUE	Unit Value	date	10	No
	TOTAL_AMOUNT	Total Amount	INT	10	
	CONDITION_AT_ORIGIN	Condition at Origin	char	100	
	CONDITION_AT_DEST	Condition at Destination	char		
	PACKAGE_ID	Package Id	char	10	
	CHECKS	Checks	Boolean	10	

	REMARKS	Remarks	char	200	
FK	ORDER_ID	Order Id	char	10	No

Table Name: tblPurchaseOrder

Description: Purchase Order Information

Key	Name	Description	Type	Length	Allow Null
PK	PURCH_ORD_NO	Purchase Order Id	char	25	No
	SUPPLIER	Supplier Name	char	10	No
	DUE_DATE	Due Date	date	200	No
	TOTAL_AMOUNT	Total Amount	date	10	No
	AUTHORIZER	Authorizer	INT	10	

Table Name: tblPurchaseItem

Description: Purchase Order Item Information

Key	Name	Description	Type	Length	Allow Null
PK/FK	PURCH_ORD_NO	Order Id	char	25	No
PK	LINE_NO	Line No	char	10	No
PK	ITEM_NO	Item No	date	10	No
	QUANTITY	Quantity	date	10	No
	UNIT_VALUE	Unit value	double	10	
	UNIT_OF_MEASURE	Unit of measure	char	10	
	DUE_DATE	Due Date	date		
	TOTAL_AMOUNT	Total Amount	double		

Table Name: tblReceiveInventory

Description: Receive Purchase Order Information

Key	Name	Description	Type	Length	Allow Null
PK/FK	PURCH_ORD_NO	Item Number	char	10	No
PK/FK	INV_ID	Description	char	5	No
PK	SUPPLIER_ID	Quantity	date	20	No
	AUTHORIZER	Unit Value	date	20	No
	CHECKED	Total Amount	Boolean		
	RECEIVE_DATE		date		

Table Name: tblInventorylocation

Description: Inventory Location Detail

Key	Name	Description	Type	Length	Allow Null
PK/FK	INVENTORY_ID	Inventory Id	char	10	No
PK/FK	INV_TYPE	Type of Inventory	char	5	No
PK	ADDRESS	Address	date	20	No
	CAPACITY	Capacity	date	20	No
	CONTACT_NO	Contact No	Boolean		
	SPECIAL_COMMENTS	Special comments	date		

Table Name: tblInventoryItem

Description: Inventory Items Detail

Key	Name	Description	Type	Length	Allow Null
PK	INV_ITEM_ID	Inventory Item Id	char	10	No
	ITEM_NAME	Item Name	char	5	
	ITEM_TYPE	Item Type	char	20	
	UNIT_OF_MEASURE	Unit of Measure	char	20	
	UNIT_VALUE	Unit Value	double	(10,2)	
	STORAGE_CONDITION	Storage conditions	char	100	

Table Name: tblPackLoadPerJob

Description: Packing Cost information

Key	Name	Description	Type	Length	Allow Null
PK/FK	ORDER_ID	Order Id	char	10	No
FK	SHIPPER_ID	Sipper Id	char	5	
PK	JOB_ID	Job Id	char	50	
	SHIPPER_NAME	Shipper Name	char	200	
	TOTAL_VALUE	Total coat per Job for packing	double	(10,2)	
	START_DATE	Storage conditions	date		
	DUE_DATE	Due Date	date		
	END_DATE		date		

Table Name: tblMaterialCostPerJob

Description: Materrial Cost information

Key	Name	Description	Type	Length	Allow Null
PK/FK	ORDER_ID	Order Id	char	10	No
PK	MATERIAL_ID	Material Id	char	5	
	MAT_TYPE	Material Type	char	50	
	MAT_ISSUES	Material issue for packing	double	200	
	MAT_USAGE	Material used for packing	double	(10,2)	
	BALANCE	Balance Material	double	(10,2)	
	COST_PER_UNIT	Cost per Unit	double	(10,2)	
	TOTAL_COST	Total Cost	double	(10,2)	

Table Name: tblChargesPerJob

Description: Charges per Job Information

Key	Name	Description	Type	Length	Allow Null
PK/FK	ORDER_ID	Order Id	char	10	No
PK	CHARGE_LINE	Charge line id	char	5	
	CHARGE_TYPE	Charge Type	char	50	
	COMMENTS	Comments	double	200	
	CHARGED_DATE	Unit Value	double	(10,2)	
	TOTAL_COST	Storage conditions	double	(10,2)	

Table Name: tblInvoice

Description: Invoice Detail

Key	Name	Description	Type	Length	Allow Null
PK	INVOICE_ID	Inventory Item Id	char	10	No
FK	SHIPPER_ID	Item Name	char	5	
	INVOICE_DATE	Item Type	date	50	
	SHIPPER_NAME	Unit of Measure	char	200	
	JOB_ID	Unit Value	char	(10,2)	
FK	ORDER_ID	Storage conditions	char	(10,2)	
	CONTAINER_NO	Container number	char	(10,2)	
	VESSEL_NAME	Vessel name	char	(10,2)	

	FLIGHT_DETAIL	Flight Detail	char		
	EDT		date		
	ETA		date		
	TOTAL_AMOUNT	Total Amount	double	(10,2)	
	TAX	Tax	double	(10,2)	
	TOT_INVOICE_AMT	Total Invoice amount	double	(10,2)	

Table Name: tblInvoicCharges

Description: Invoice Charges Detail

Key	Name	Description	Type	Length	Allow Null
PK/FK	LINE_NO	Line No	char	10	No
PK	CHARGE_TYPE	Charge type	char	10	
	CHARGE_AMOUNT_LOC	Amount in local currency	char	(10,2)	
	CHARGE_AMOUNT_FRN	Amount in local currency	double	(10,2)	
	CHARGE_DATE	Charge date	date		
	COMMENTS	comments	double	200	
	INVOICE_ID	Invoice id	char	10	

Table Name: tblCostCharges

Description: Actual Cost Details

Key	Name	Description	Type	Length	Allow Null
PK/FK	LINE_NO	Line No	char	10	No
PK	CHARGE_TYPE	Charge type	char	10	
	CHARGE_AMOUNT_LOC	Amount in local currency	char	(10,2)	
	CHARGE_AMOUNT_FRN	Amount in local currency	double	(10,2)	
	CHARGE_DATE	Charge date	date		
	COMMENTS	comments	double	200	
	INVOICE_ID	Invoice id	char	10	

Relationships among the tables

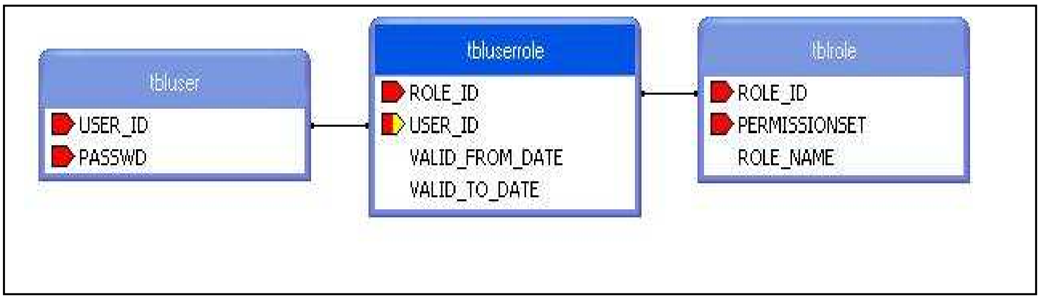


Figure A.1 Entities involved in User Login

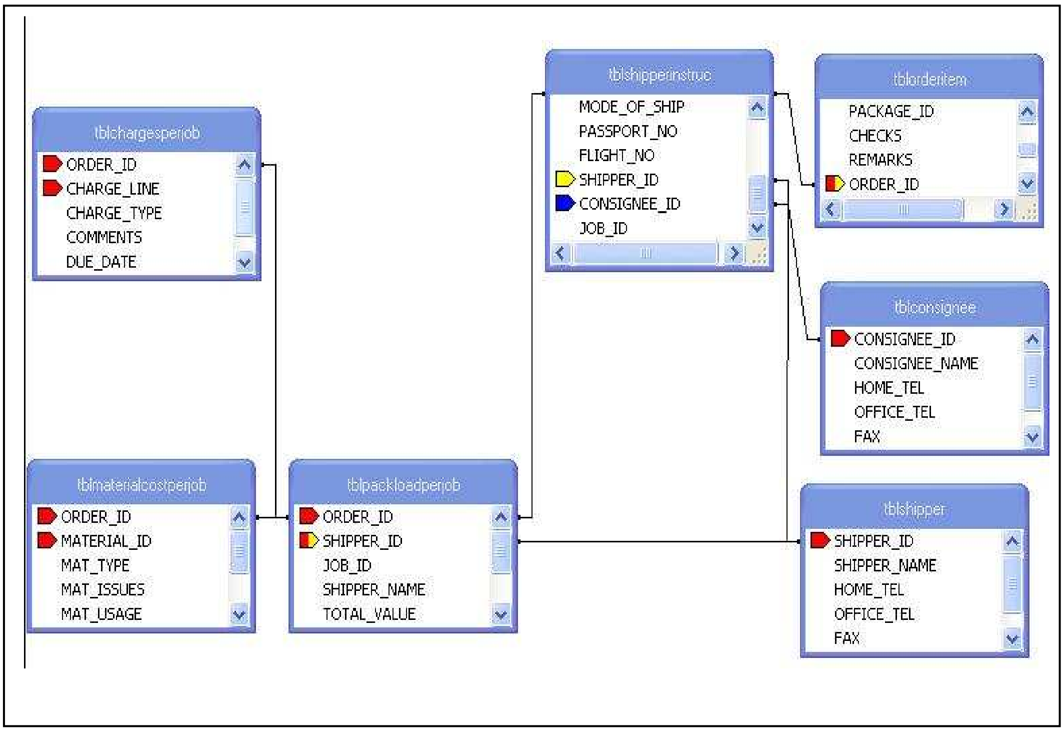


Figure A.2 Entities involved in Freight Job handling

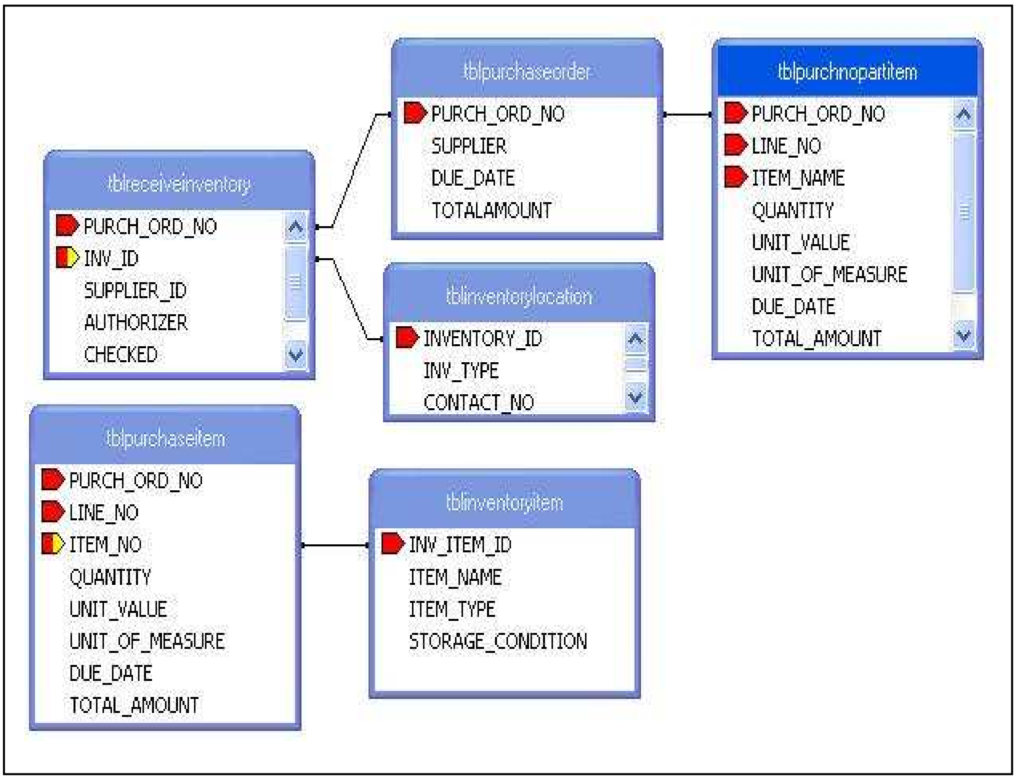


Figure A.3 Entities involved in Purchasing

APPENDIX B: SYSTEM DOCUMENTATION

System must satisfy following requirements to install this system

a) Hardware requirements,

   Intel Pentium IV(or equivalent) 2.4 MHz processor or higher

   Minimum 128 MB RAM

   CD ROM

b) Software requirements

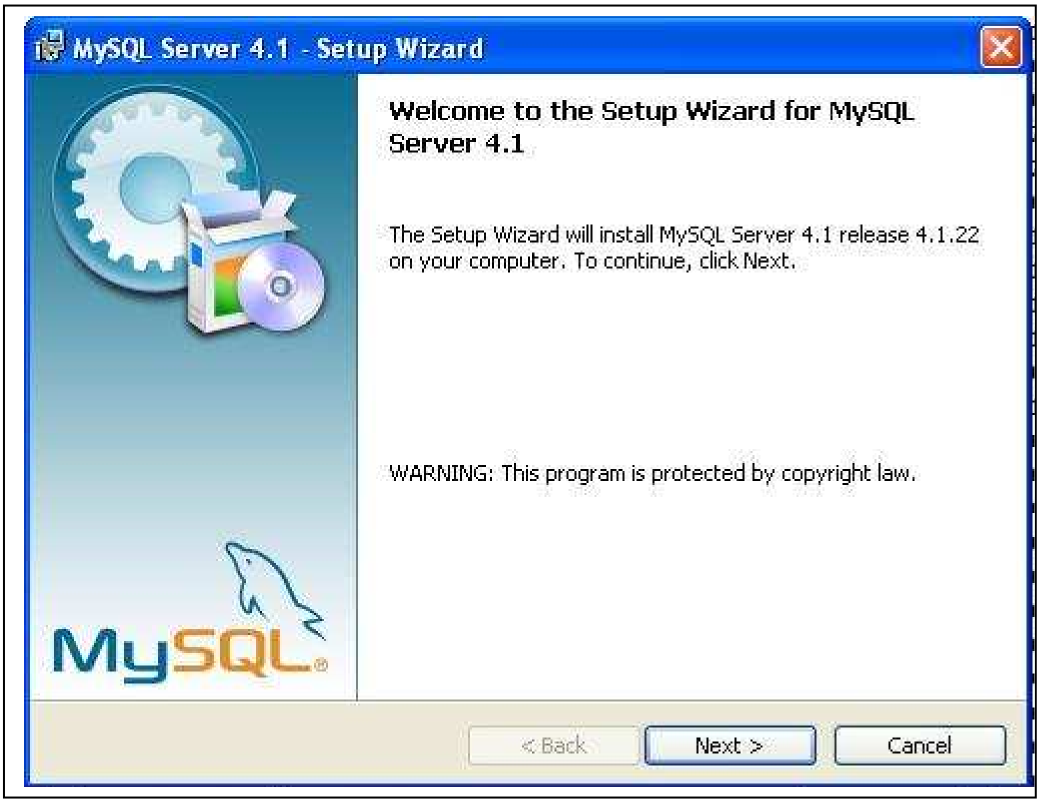
   Windows XP Service Pack 2

   My SQL Server 4.1

   .NET Frame work 2.0

1) Installing MySQL server

   a) Please find the MySQL installer provided with the installation CD.
   b) Double-click on the installer.
   You will see an installation wizard as shown below.



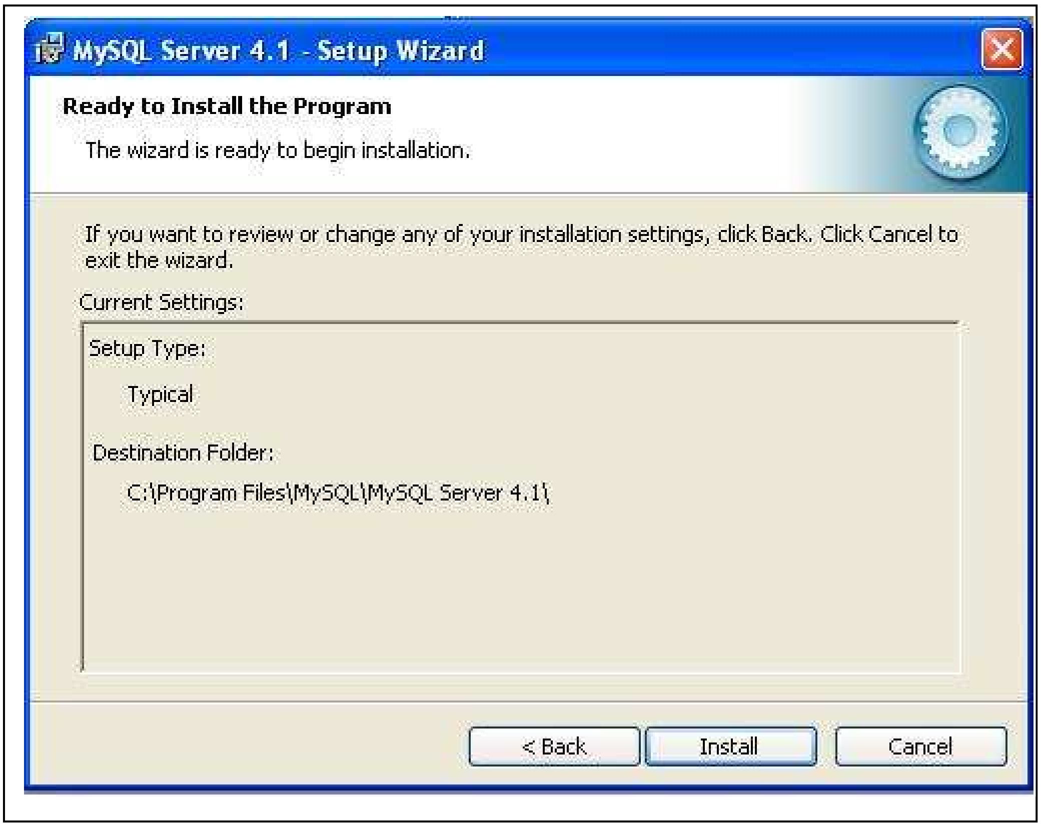
Welcome to the Setup Wizard for MySQL Server 4.1

The Setup Wizard will install MySQL Server 4.1 release 4.1.22 on your computer. To continue, click Next.

WARNING: This program is protected by copyright law.

Figure B.1 MySQL setup Wizard

c) Select “Typical” and press next(If you want to install the server to a different location select custom and in the next step give location)
d) Then click "Install" and let it do its thing. Press Next, and then press Finish on the next screen.



Ready to Install the Program

The wizard is ready to begin installation.

If you want to review or change any of your installation settings, click Back. Click Cancel to exit the wizard.

Current Settings:

Setup Type:

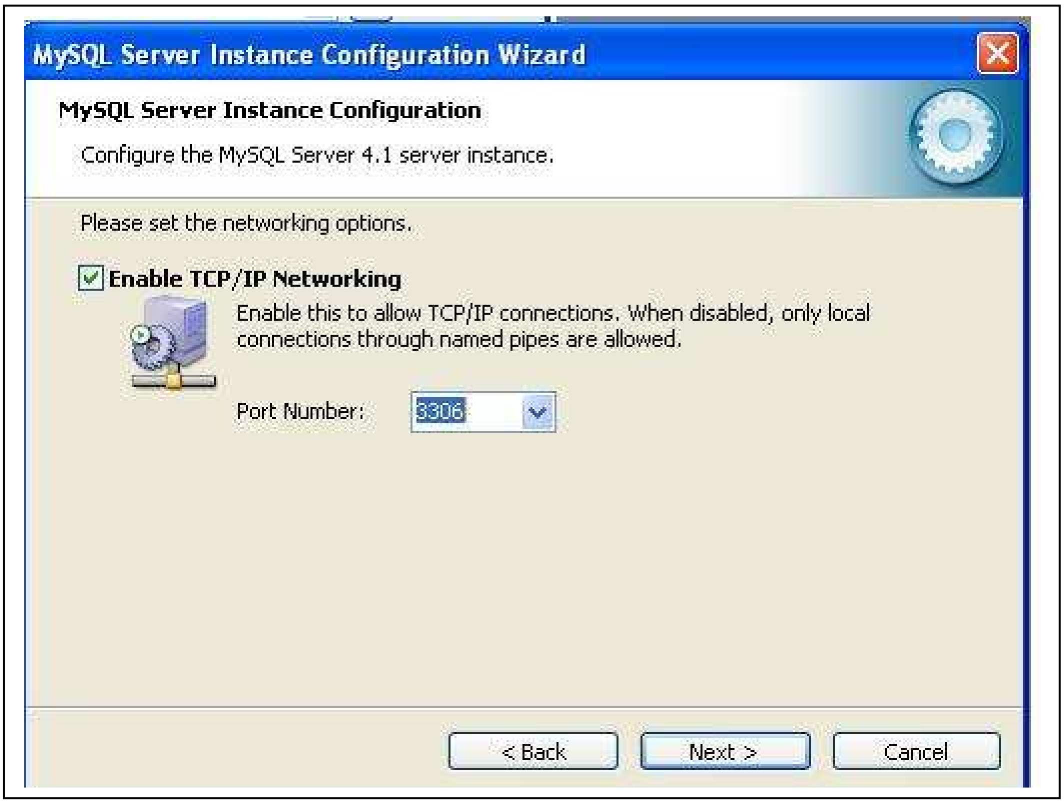
Typical

Destination Folder:

C:\Program Files\MySQL\MySQL Server 4.1\

Figure B.2 installation path of the MySQL

e) The MySQL Server Instance Configuration Wizard will pop up. Click "Next". Choose "Detailed Configuration" and click "Next". Answer the questions according to computer type. The instructions are fairly straight forward.
For the port leave the defullt port as it is



MySQL Server Instance Configuration Wizard

MySQL Server Instance Configuration

Configure the MySQL Server 4.1 server instance.

Please set the networking options.

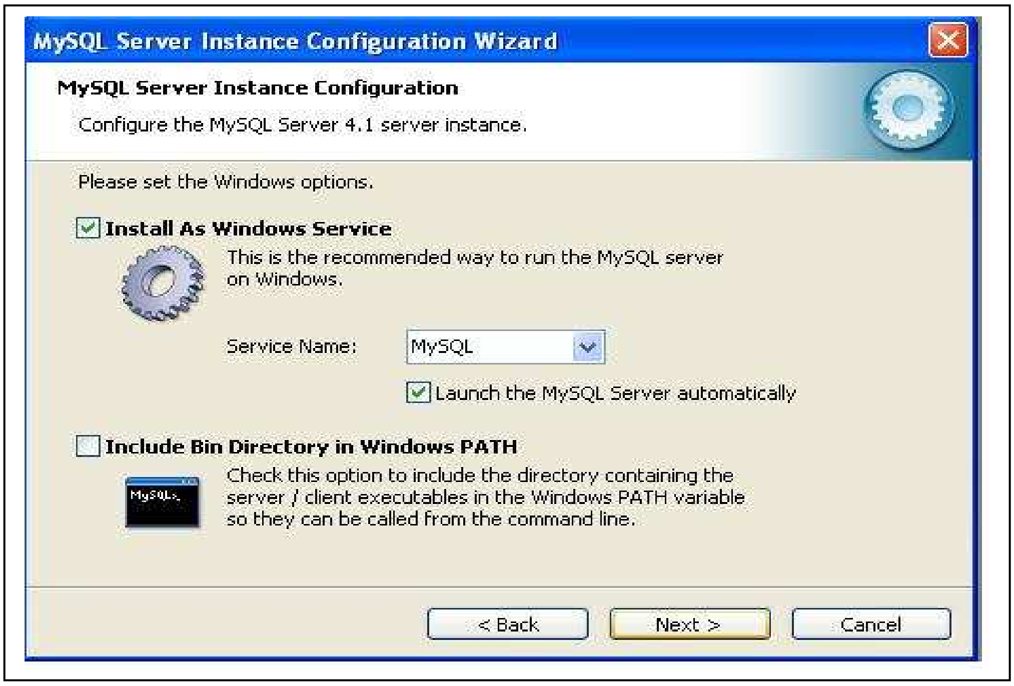
☑ Enable TCP/IP Networking

Enable this to allow TCP/IP connections. When disabled, only local connections through named pipes are allowed.

Port Number: 3306

Figure B.3 Database port used by MySQL

f)On the Windows Options Page, it may want to un-check the "Launch MySQL Server automatically" option, and check the Include Bin Directory in Windows PATH so that the MySQL server can be used from the windows command line. It is easiest to Run MySQL as a Windows service.



MySQL Server Instance Configuration Wizard

MySQL Server Instance Configuration

Configure the MySQL Server 4.1 server instance.

Please set the Windows options.

☑ Install As Windows Service

This is the recommended way to run the MySQL server on Windows.

Service Name: MySQL

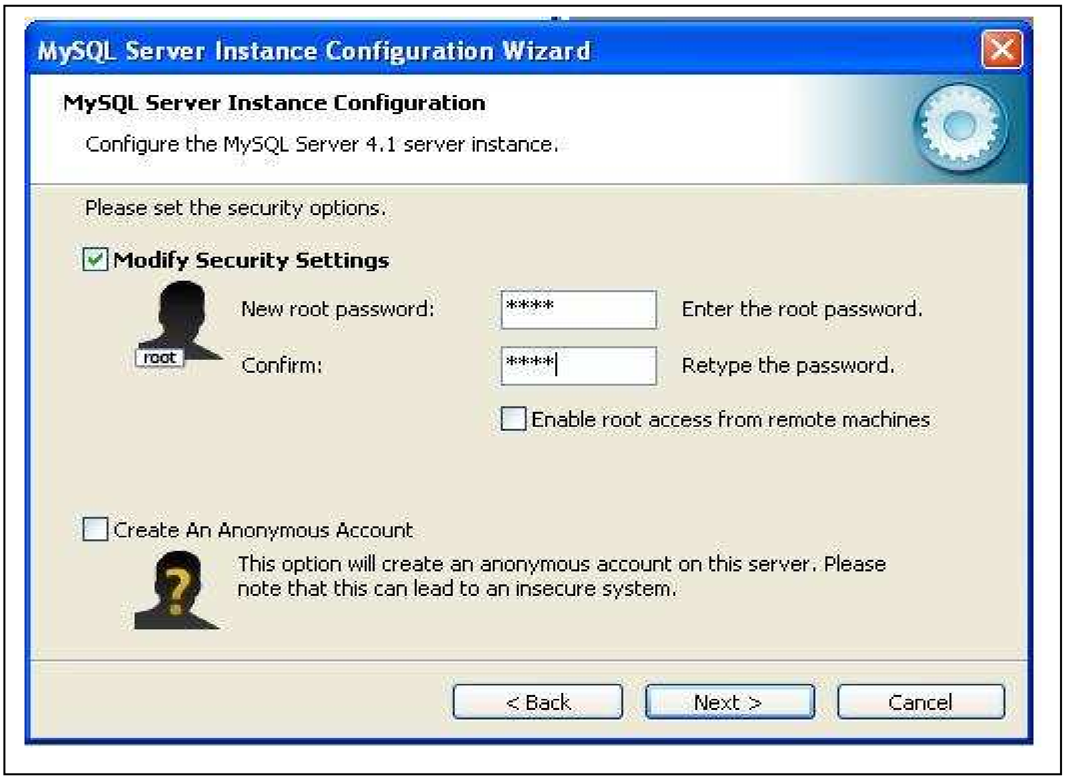
☑ Launch the MySQL Server automatically

□ Include Bin Directory in Windows PATH

Check this option to include the directory containing the server / client executables in the Windows PATH variable so they can be called from the command line.

Figure B.4 MySQL configuration detail

g) On the next screen, pick a root password. This is the main "superuser" password for the server. At the initial installation type the user name “root” and password as “root”. It can be changed later as required.



MySQL Server Instance Configuration Wizard

MySQL Server Instance Configuration

Configure the MySQL Server 4.1 server instance.

Please set the security options.

☑ Modify Security Settings

New root password: **** Enter the root password.

Confirm: **** Retype the password.

☐ Enable root access from remote machines

☐ Create An Anonymous Account

This option will create an anonymous account on this server. Please note that this can lead to an insecure system.

Figure B.5 Database User security handling window

h) Finally, click the "Execute" button to configure the MySQL server:

i) you may need to restart the machine for correct paths to be setup in the system.

2) Installing MySQL Administrator

a) Please find the MySQL Administrator installer provided with the installation CD. (mysql-essential-5.1win32.msi)

b) Run the installer choosing the "Complete" installation type.
c) Can be installed at anywhere (Usually at the “Program Files”).
d) The installation is straight forward.

3) Populating the database
a) Login to MySQL Administrator with the user name = “root” and password = “root”.
b) Select “Restore”.
c) Click “Open backup file”, browse and select “WRSFreightDatabase.sql” in the CD.
d) Click “Start Restore”.
e) When the database population process is over click “Ok”.

4) My SQL connect installation.

This step is a pre requirement to run the system. To connect the system developed on .NET environment this connector is use to connect the system with the MYSQL database

a) Please find the connector in the Database Software folder provided with the installation CD. (mysql-connector-odbc-3.51.27-win32.msi)

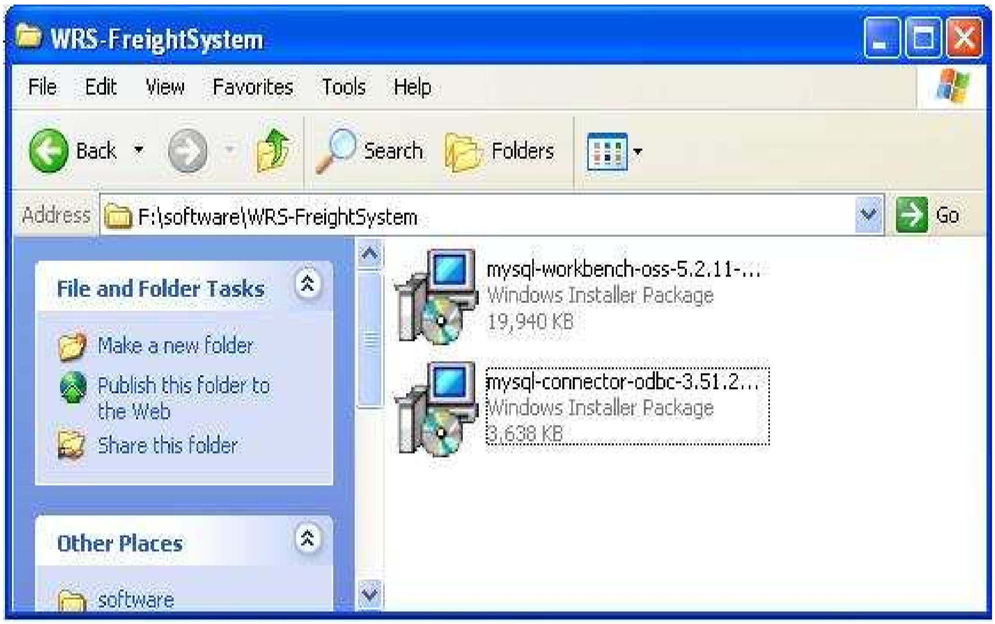
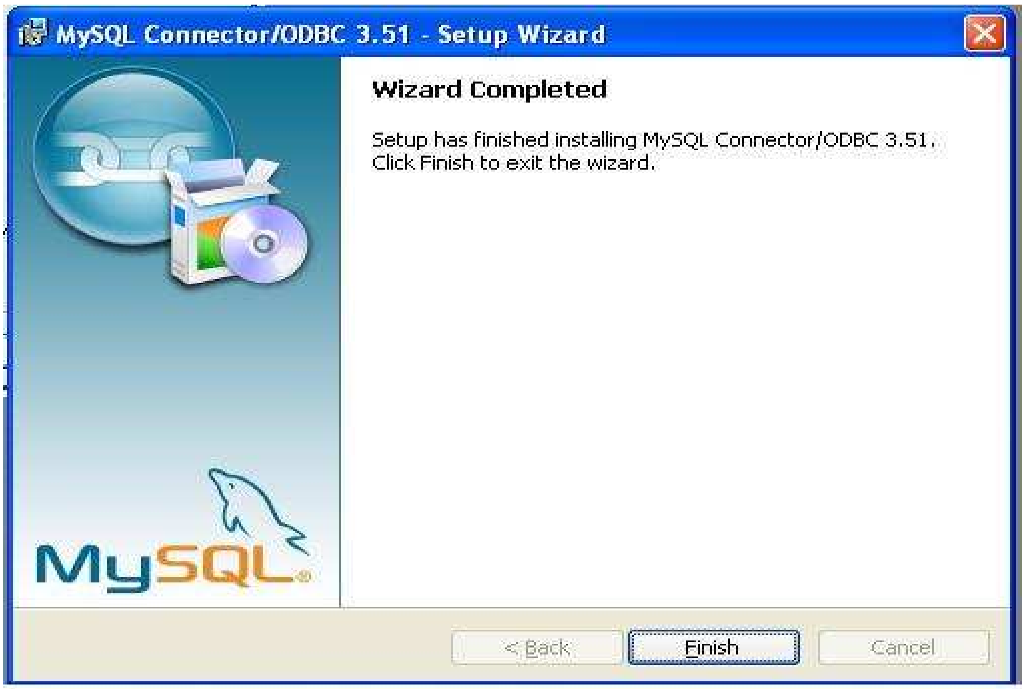

WRS-FreightSystem
File
Edit
View
Favorites
Tools
Help
Back
Search
Folders
Address
F:\software\WRS-FreightSystem
Go
File and Folder Tasks
Make a new folder
Publish this folder to the Web
Share this folder
Other Places
software
mysql-workbench-oss-5.2.11-...
Windows Installer Package
19,940 KB
mysql-connector-odbc-3.51.2...
Windows Installer Package
3,638 KB

Figure B.6 Installer location of MySQL connect

b) Run the setup file and Press "next"

c) When the installation finishes close the window by clicking finish button.



Wizard Completed

Setup has finished installing MySQL Connector/ODBC 3.51.
Click Finish to exit the wizard.

Figure B.7 Connector installation

d) Setup the Environment Variables in the PC that you have installed the MQSQL and Connecter services to access your system
e) Please go to Start/Control Panel/ Performance and maintenance/ Administrative tools. You see an Icon similar as shown in the Figure below.

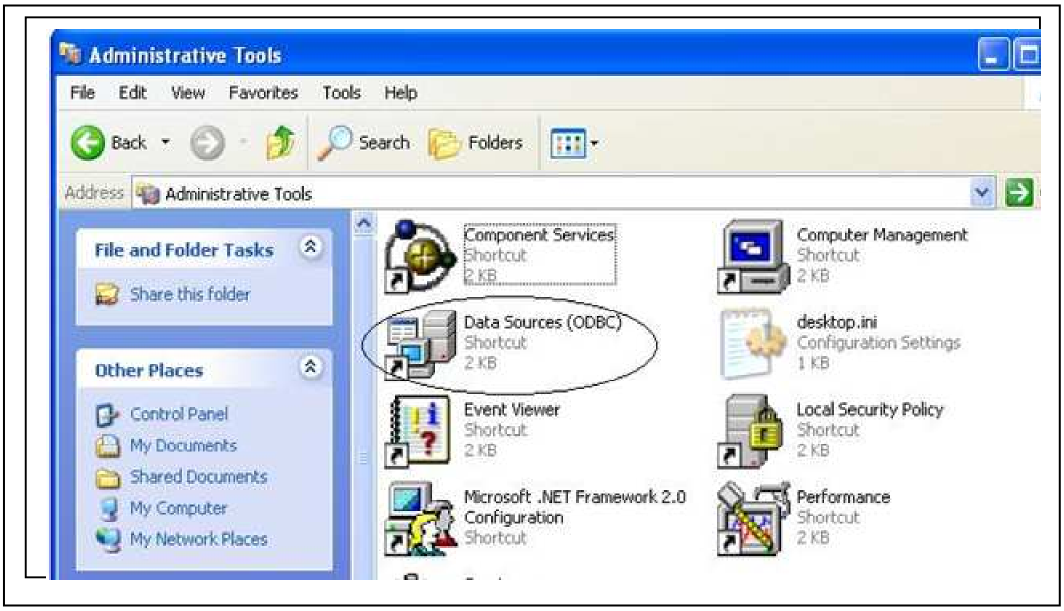
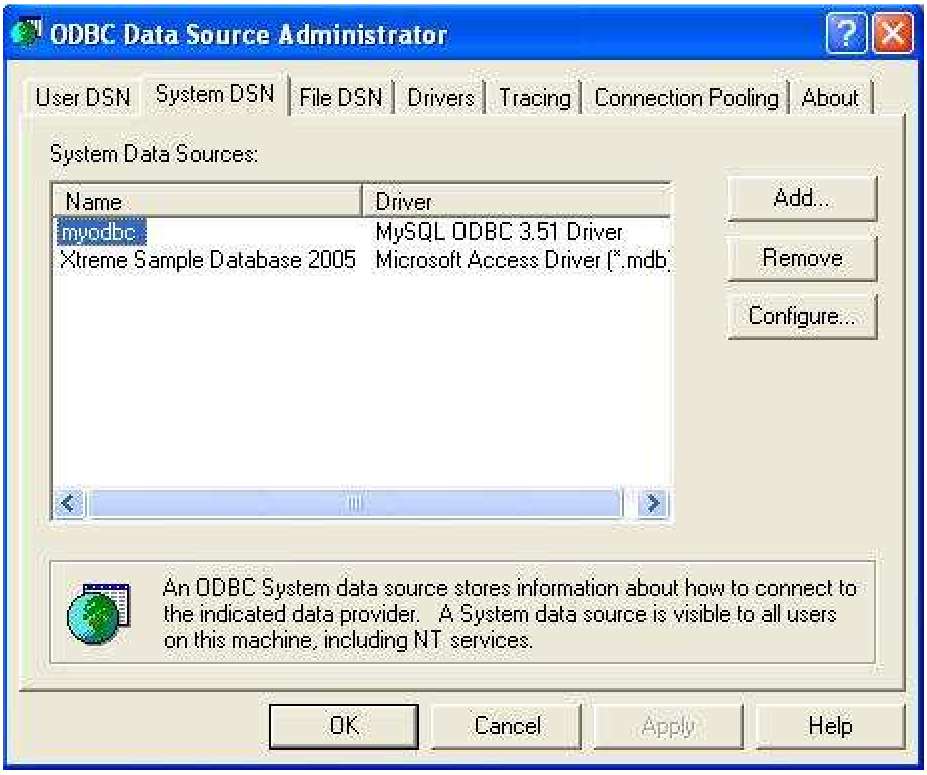


Figure B.7 ODBC Data connector

f) In the ODBC Data Administrator select the system DSN tab and "Add new entry"

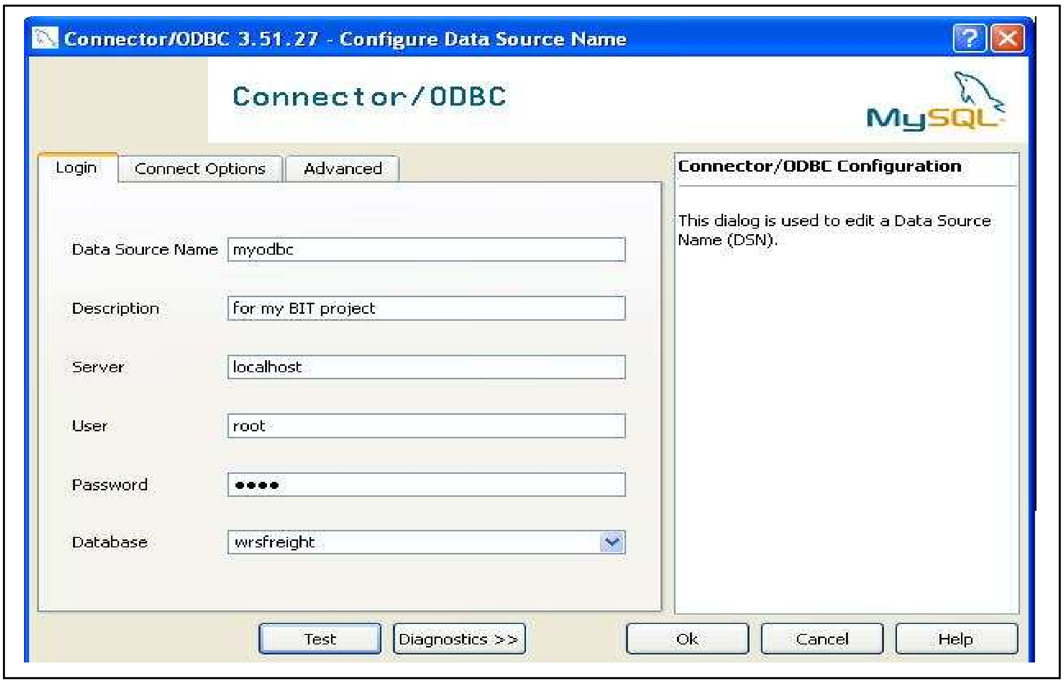


Name	Driver
myodbc	MySQL ODBC 3.51 Driver
Xtreme Sample Database 2005	Microsoft Access Driver (*.mdb

An ODBC System data source stores information about how to connect to the indicated data provider. A System data source is visible to all users on this machine, including NT services.

Figure B.7 ODBC System DSN

G) Edit the details. Give the Database "wrsfreight" as given below Figure.


Connector/ODBC 3.51.27 - Configure Data Source Name

Connector/ODBC Configuration

This dialog is used to edit a Data Source Name (DSN).

Data Source Name: myodbc

Description: for my BIT project

Server: localhost

User: root

Password: ••••

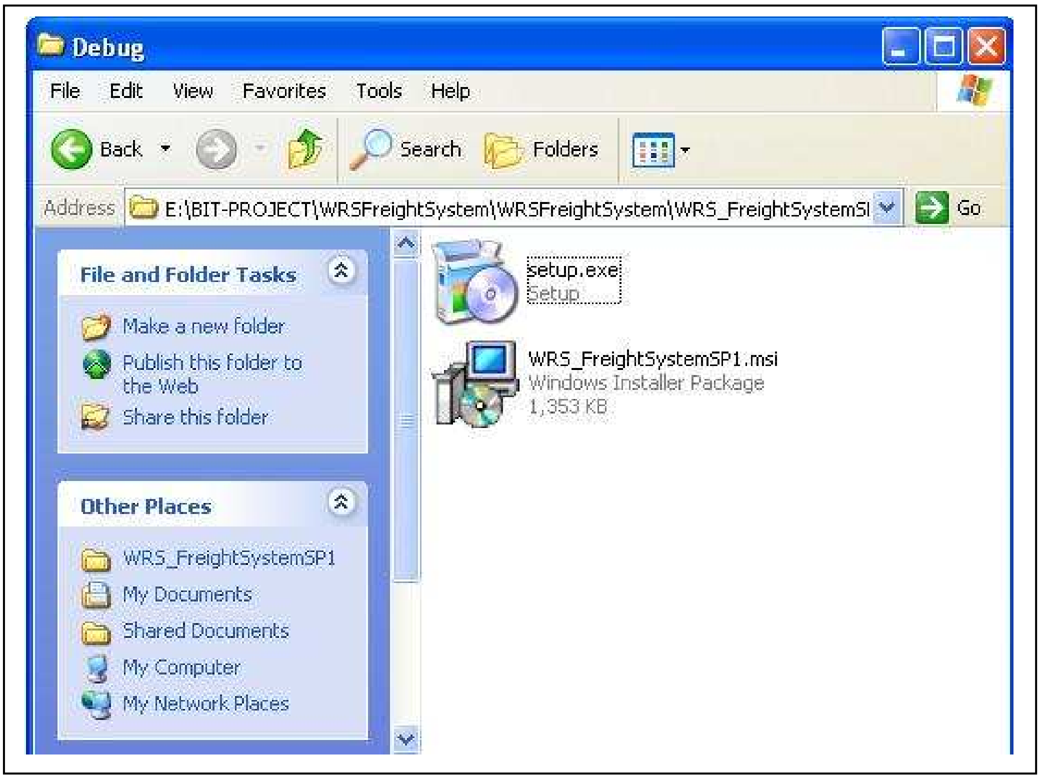
Database: wrsfreight

Figure B.8 Connector information screen

Above given steps will correctly connect your database to the System.

4) WRS Freight Management software installation.

a) Please find the WRS Freight Management software folder provided with the installation CD


Debug

Address: E:\BIT-PROJECT\WRSFreightSystem\WRSFreightSystem\WRS_FreightSystemSI

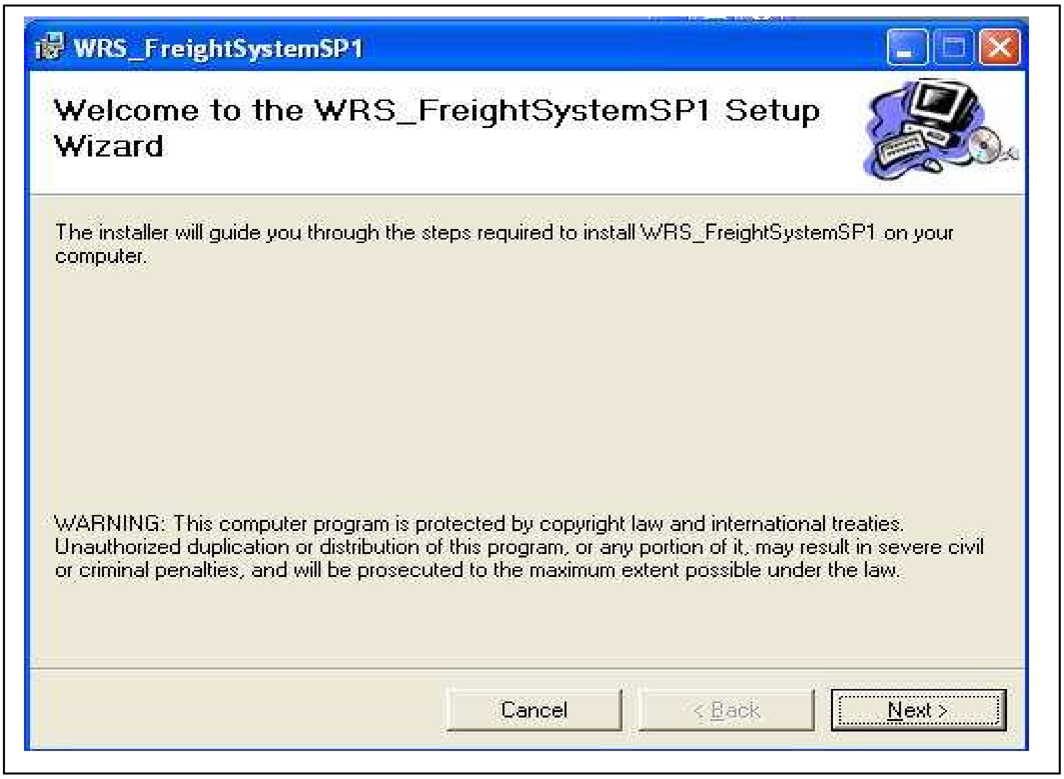
setup.exe
Setup

WRS_FreightSystemSP1.msi
Windows Installer Package
1,353 KB

Figure B.9 WRS Freight Management software installer

b) Double click on the setup icon

c) You will prompt for an installation wizard. Follow the steps as given in below figures.



WRS_FreightSystemSP1

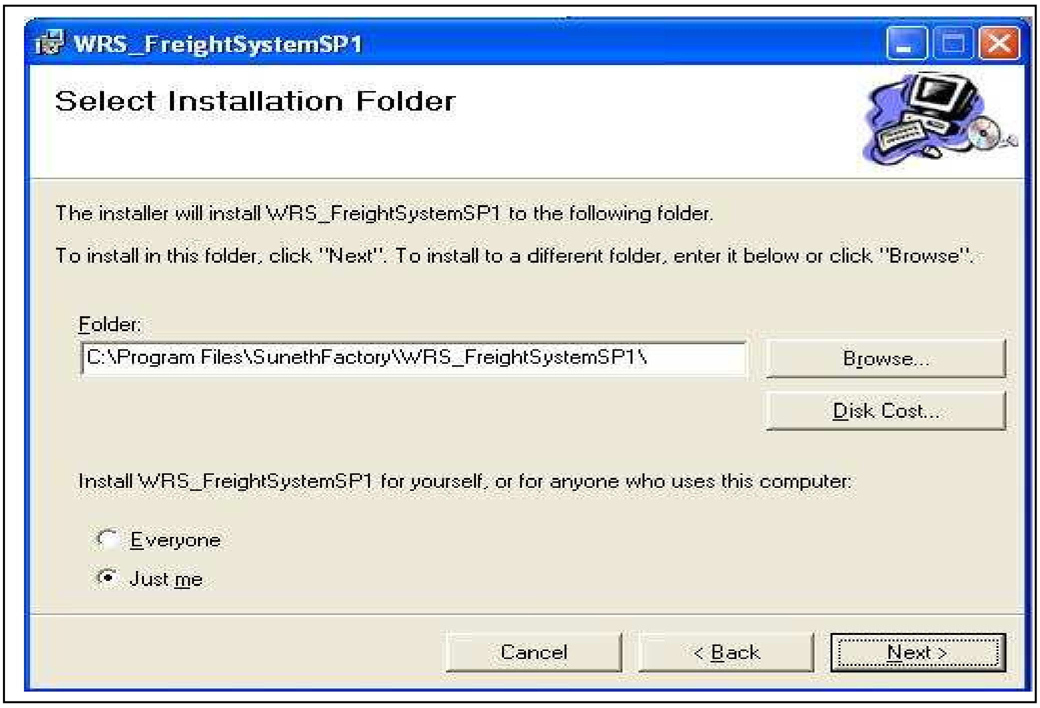
Welcome to the WRS_FreightSystemSP1 Setup Wizard

The installer will guide you through the steps required to install WRS_FreightSystemSP1 on your computer.

WARNING: This computer program is protected by copyright law and international treaties. Unauthorized duplication or distribution of this program, or any portion of it, may result in severe civil or criminal penalties, and will be prosecuted to the maximum extent possible under the law.

Figure B. 10 WRS Freight Management software installer

d) Please select the folder path that you need to install this software.



WRS_FreightSystemSP1

Select Installation Folder

The installer will install WRS_FreightSystemSP1 to the following folder.

To install in this folder, click "Next". To install to a different folder, enter it below or click "Browse".

Folder: C:\Program Files\SunethFactory\WRS_FreightSystemSP1\

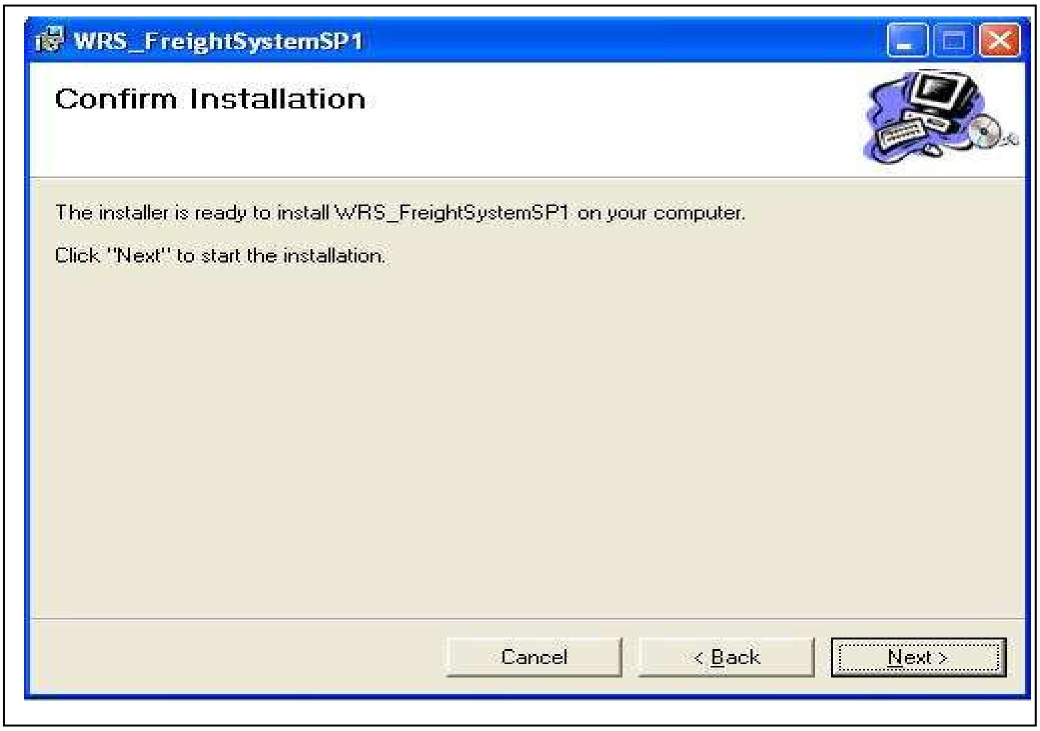
Install WRS_FreightSystemSP1 for yourself, or for anyone who uses this computer:

○ Everyone

● Just me

Figure B. 11 WRS Freight Management software installer path

e) In the Next dialog Click “Next”. This will install the system to your computer.


WRS_FreightSystemSP1

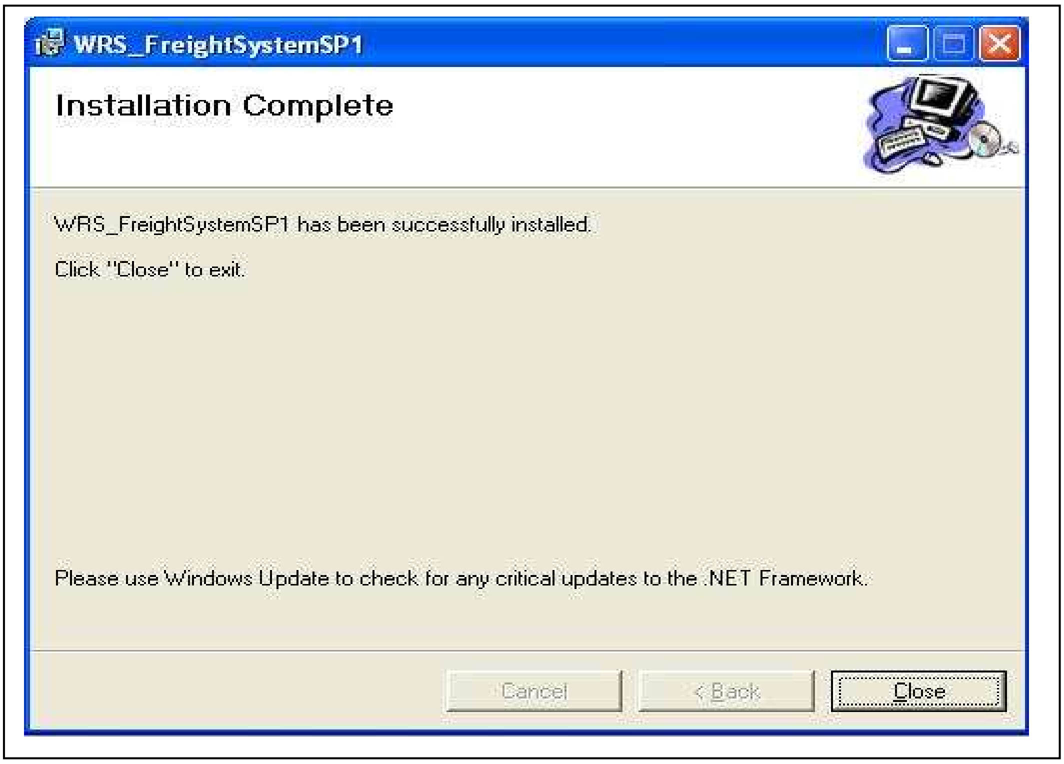
Confirm Installation

The installer is ready to install WRS_FreightSystemSP1 on your computer.

Click "Next" to start the installation.

Figure B. 12 WRS Freight Management software installer path

f) After installation successfully completed Click “Close”


WRS_FreightSystemSP1

Installation Complete

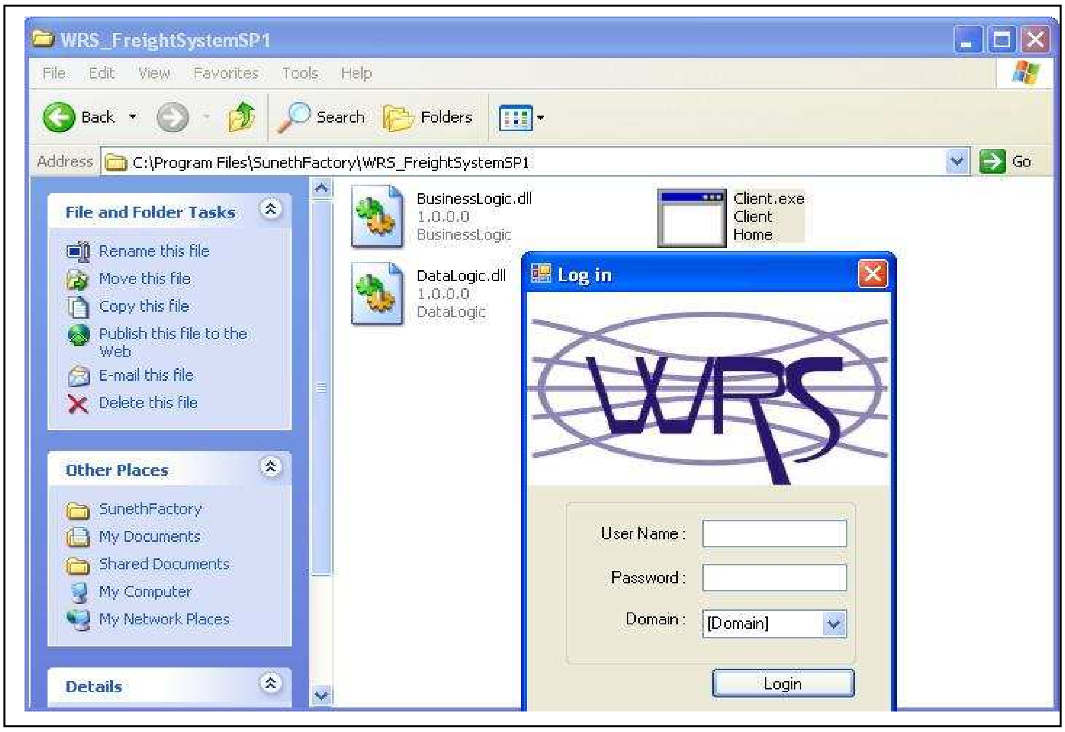
WRS_FreightSystemSP1 has been successfully installed.

Click "Close" to exit.

Please use Windows Update to check for any critical updates to the .NET Framework.

Figure B. 12 WRS Freight Management software installer path

g) Now you can load the system from the Place where you have install the system.



User Name :

Password :

Domain : [Domain]

Figure B. 12 Exsiccation of WRS Freight Management software.

APPENDIX C: USER DOCUMENTATION

WRSFreightSystem Designed and implemented to perform the Basic Business flows at "World Wide Relocation Services" Household goods Relocation Company.
User Administration, External Entity registration, Shipment handling process and purchasing the inventory items to the warehouse are the main functional flows in this system.
Step by step guidelines to perform these activities are given below.

- Start the system

Double Click on the WRSFreightSystem icon in the desktop. If the desktop icon is not visible go to the folder that the program installed and run the .exe file in that folder.
System will be Loaded in a few seconds and Login Screen will prompt to the user to enter the login detail.

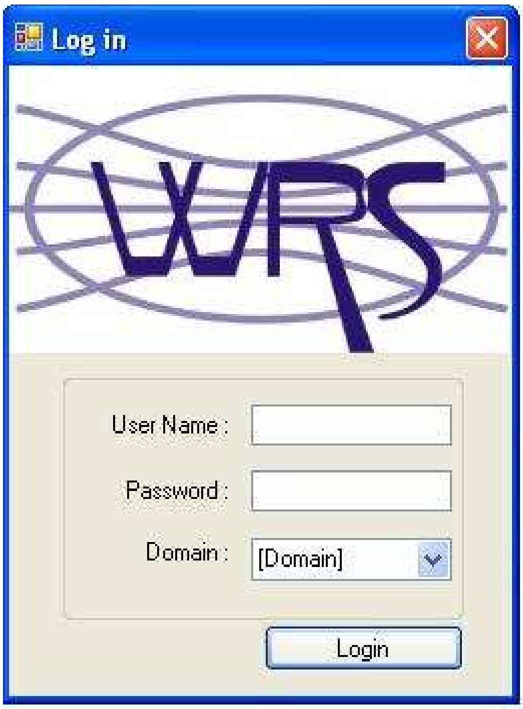


Figure C. 1 User Login Screen

Enter Correct User name and password in the login interface. Select the correct domain and press Login Button.

Successful login to the system will guide to the Main Container Window given below.

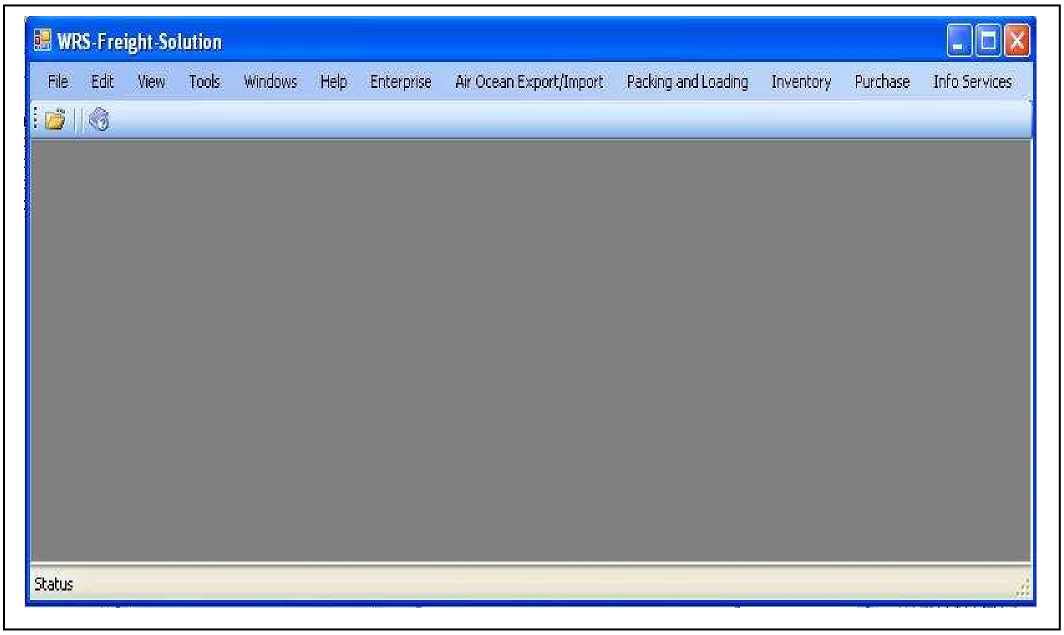

WRS-Freight-Solution
File Edit View Tools Windows Help Enterprise Air Ocean Export/Import Packing and Loading Inventory Purchase Info Services
Status

Figure C. 2 Main form Window

Windows accessibility and the operations connected to each window depends on the user domain that of the current user logged on to the system.

- Shipper Registration

To register new users and to update the existing users,

Enterprise/ Shipper

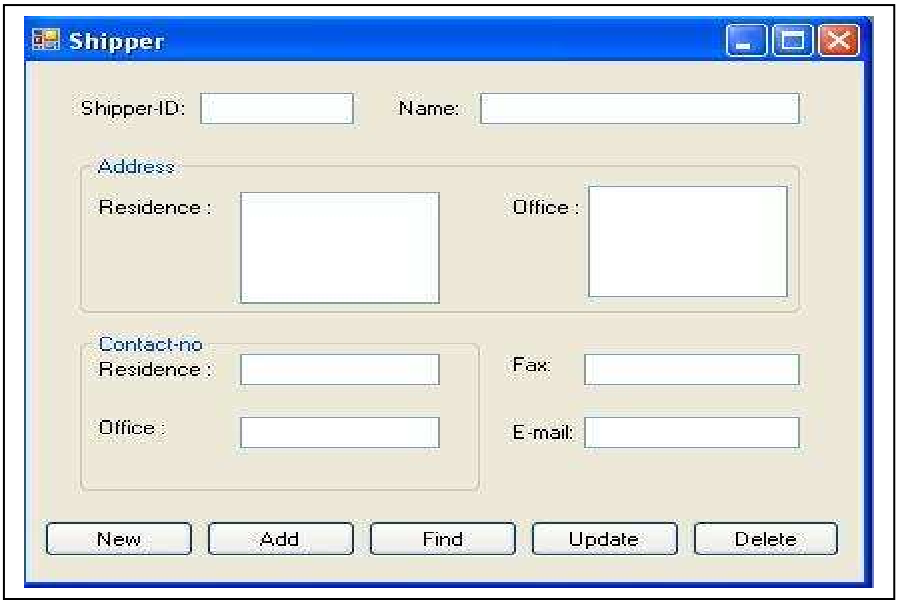
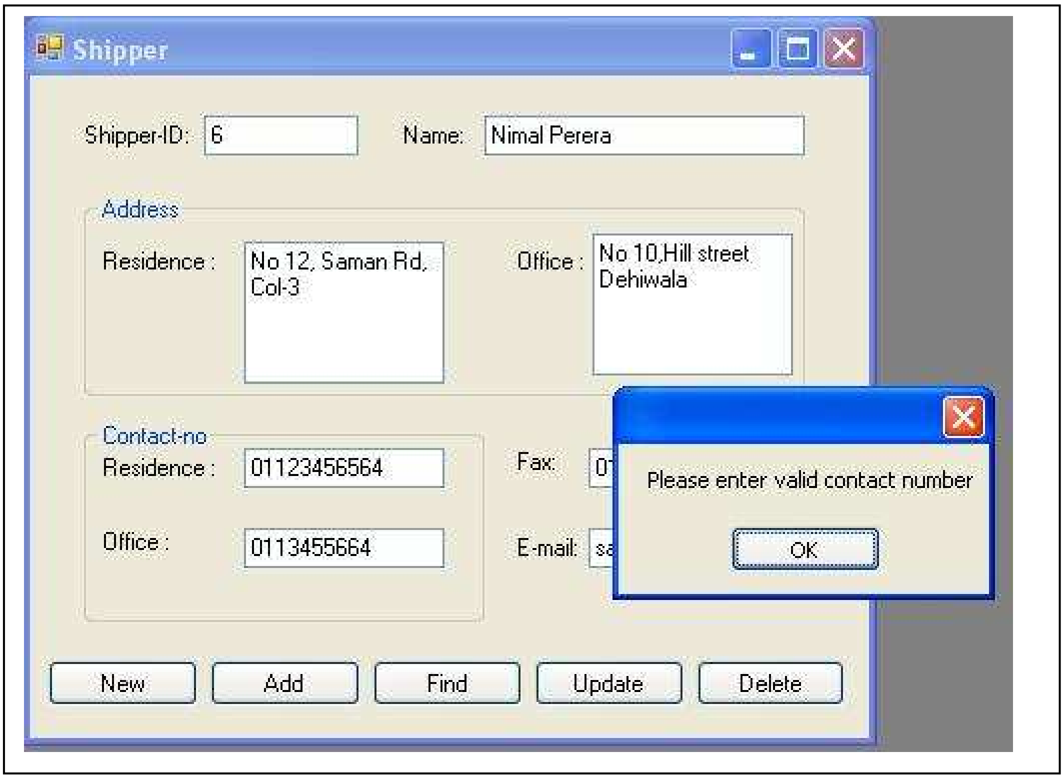


Figure C. 3 Shipper form window

When the user press the new button system will generate shipper id, which will be shown in the shipper ID text box. User is allowed to enter valid details into form. If the user tries to enter invalid contact number (Correct format should be 10digit number) an error message will prompt to enter correct number.



Shipper-ID: 6
Name: Nimal Perera

Address
Residence : No 12, Saman Rd,
Col-3
Office : No 10,Hill street
Dehiwala

Contact-no
Residence : 01123456564
Office : 0113455664
Fax: 01
E-mail: sa

Please enter valid contact number

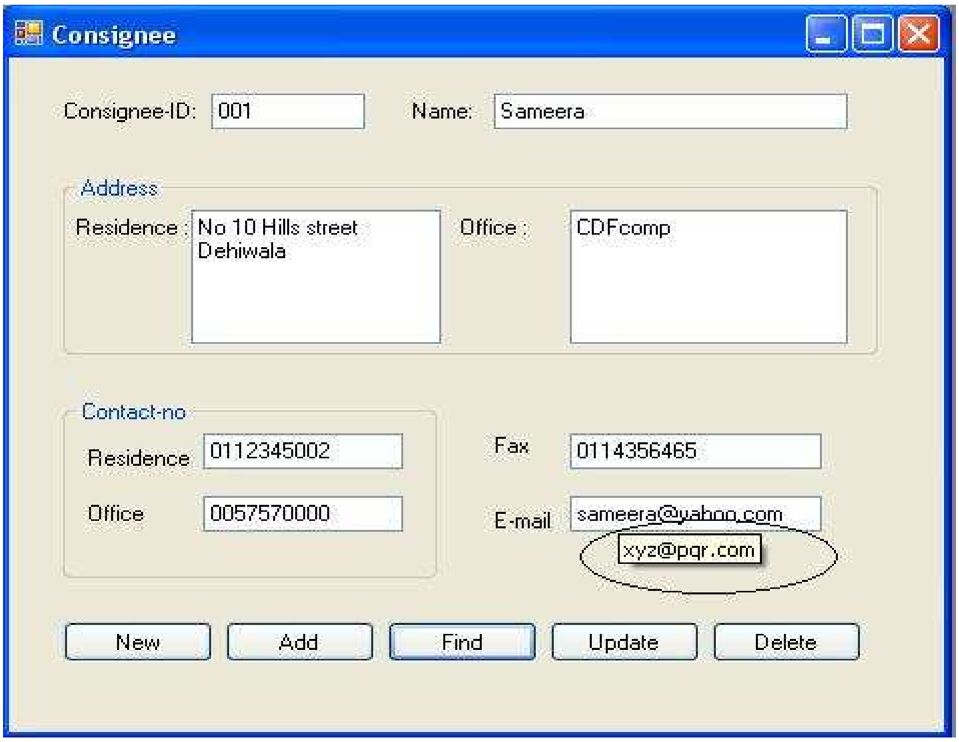
Figure C. 4 Error Message

Authorized Users can perform the basic operations in this window (Add, Update, Delete and Find operations)

- Consignee Registration

Window with a similar appearance and the controls to enter consignee details will be Shown in this interface.
Enterprise/Consignee

Other than the basic operations, when entering details for the special text boxes that need some standard formatting tool tips will be shown. Then the user can avoid entering invalid data to the system. This facility is provided in other windows as well.



Consignee

Consignee-ID: 001

Name: Sameera

Address

Residence : No 10 Hills street
Dehiwala

Office : CDFcomp

Contact-no

Residence 0112345002

Office 0057570000

Fax 0114356465

E-mail sameera@yahoo.com

xyz@pqr.com

Figure C. 5 Tool Tip

- Supplier Registration

Registration of supplier in the WRS is done using the window shown below.

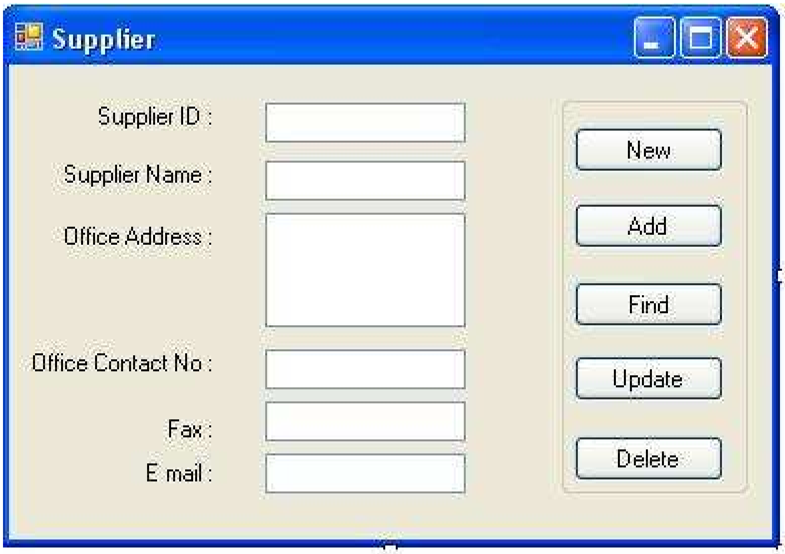


Figure C. 6 Supplier Detail form window

- Shipment handling

This is the main business flow in this system which involves very large number of interrelated data.
Step to follow in shipment handling is categorized under “Air &Ocean Export/Import ”

Shown below is how the steps in shipment handling are physically arranged in a MDI window

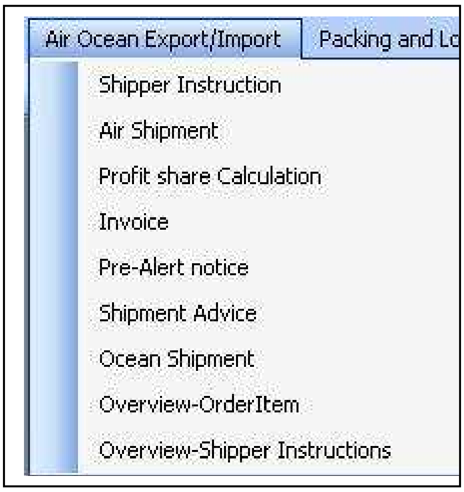
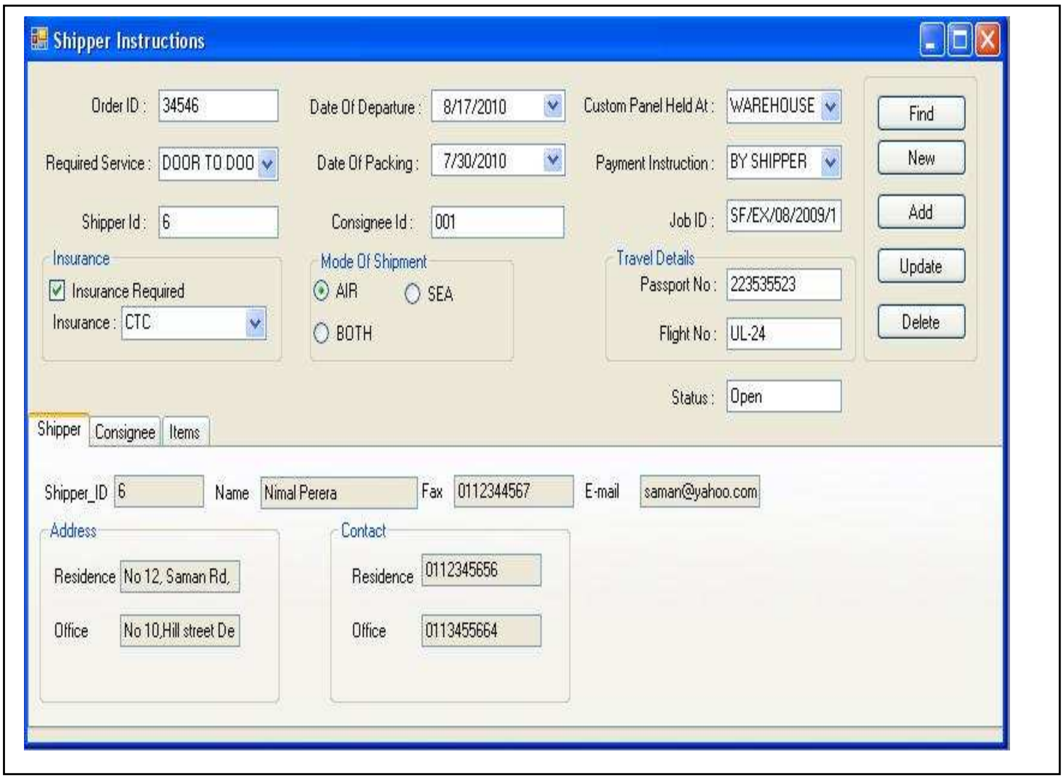


Figure C. 7 Shipment handling navigator

This shipment process results when the user enters the details to the “Shipment Instructions” master detail window.

User is allowed to enter order Items of the particular shipment only after saving the header record.
The form window consists of three tabs,namely Shipper,Consignee and Order Items.
Shipper and Consignee are read only tabs and they are populating when the user tabs out in Shipper Id text box and Consignee text box in the header.
Third tab is to enter the shipment Order items. This tab consist of a table. (Data Grid)
To enter data to the table first user must press “Add” button. Then the new row will appear to enter the detail.
Line number is auto generated and the columns Condition at “Origin” and the “Condition at the Destination “are combo boxes. User can select a condition out of several predefined conditions.
After entering the data user has to press the Update button to save the data.
For existing data also after modification user needs to press “Update” button.
To delete the records user needs to select the record to be deleted and press Delete.
There will be a confirmation message before deleting. If the user press “Yes” the record will be deleted.



Order ID : 34546

Required Service : DOOR TO DOO

Shipper Id : 6

Date Of Departure : 8/17/2010

Date Of Packing : 7/30/2010

Consignee Id : 001

Custom Panel Held At : WAREHOUSE

Payment Instruction : BY SHIPPER

Job ID : SF/EX/08/2009/1

Insurance

☑ Insurance Required

Insurance : CTC

Mode Of Shipment

● AIR ○ SEA

○ BOTH

Travel Details

Passport No : 223535523

Flight No : UL-24

Status : Open

Shipper_ID 6 Name Nimal Perera Fax 0112344567 E-mail saman@yahoo.com

Address

Residence No 12, Saman Rd,

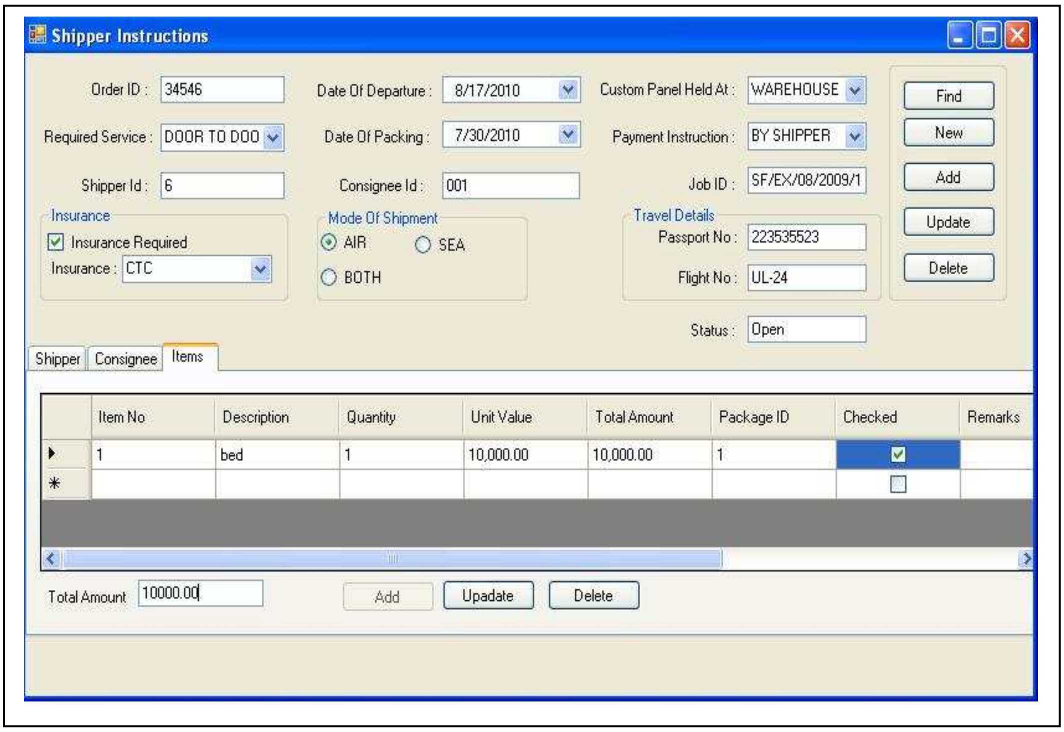
Office No 10,Hill street De

Contact

Residence 0112345656

Office 0113455664

Figure C. 8 Shipment instruction form window



Order ID : 34546

Required Service : DOOR TO DOO

Shipper Id : 6

Date Of Departure : 8/17/2010

Date Of Packing : 7/30/2010

Consignee Id : 001

Custom Panel Held At : WAREHOUSE

Payment Instruction : BY SHIPPER

Job ID : SF/EX/08/2009/1

Insurance

☑ Insurance Required

Insurance : CTC

Mode Of Shipment

● AIR ○ SEA

○ BOTH

Travel Details

Passport No : 223535523

Flight No : UL-24

Status : Open

	Item No	Description	Quantity	Unit Value	Total Amount	Package ID	Checked	Remarks
▶	1	bed	1	10,000.00	10,000.00	1	☑	
*							☐	

Total Amount 10000.00

Figure C. 9 Master detail form window

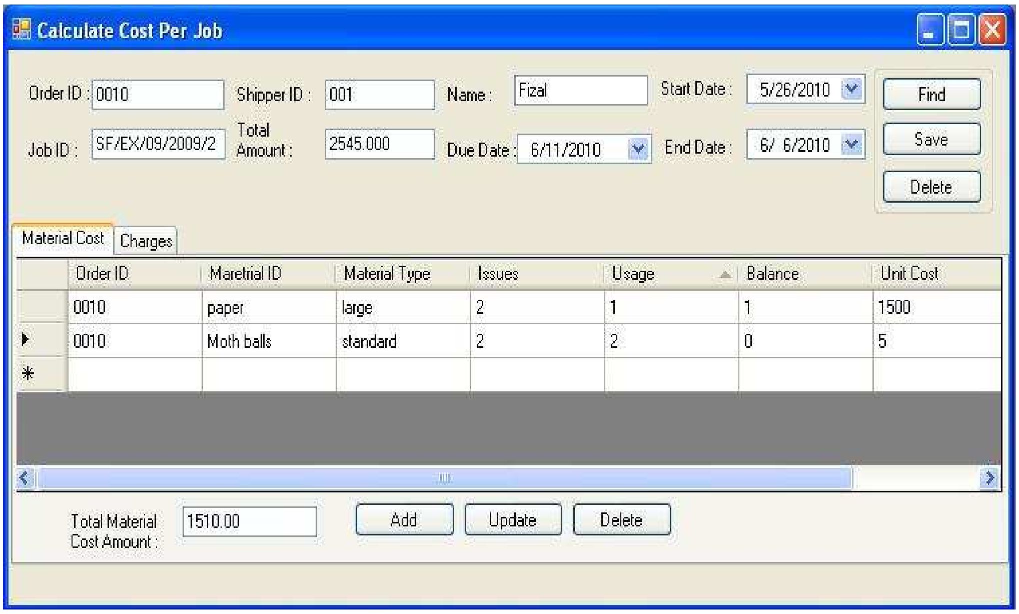
Observe that the Status of the Shipment order is in “Open” state. Shipment orders can be deleted only if the order is in “Open state”. If not user can not delete the record. It can only send to the “Canceled” state if the shipment is canceled.
States of the Shipment order are given below.

- Open- Initial stage of the shipment order. After saving a new record the order is in this state.
- Accepted- Shipper accepted the price quotation sent by WRS and the shipment is ready to lift from the original location.
- Received- When the shipment items are brought to the warehouse own by WRS.
- Packed- Items are packed and ready to ship.
- Shipped- when the shipment is shipped using a Air freight or Sea freight.
- Completed- The consignee received the goods at the destination location.

After successfully entering the details to this window user can generate a Shipment instruction sheet that can be sent the customer. Structure of this report is given in the report section of this documentation.

- Packing and loading

This set of windows is used by the operational staff of the packing and loading department.



Calculate Cost Per Job

Order ID : 0010
Shipper ID : 001
Name : Fizal
Start Date : 5/26/2010
Job ID : SF/EX/09/2009/2
Total Amount : 2545.000
Due Date : 6/11/2010
End Date : 6/ 6/2010

Order ID	Maretrial ID	Material Type	Issues	Usage	Balance	Unit Cost
0010	paper	large	2	1	1	1500
0010	Moth balls	standard	2	2	0	5

Total Material Cost Amount : 1510.00

Figure C. 10 Cost calculator form window

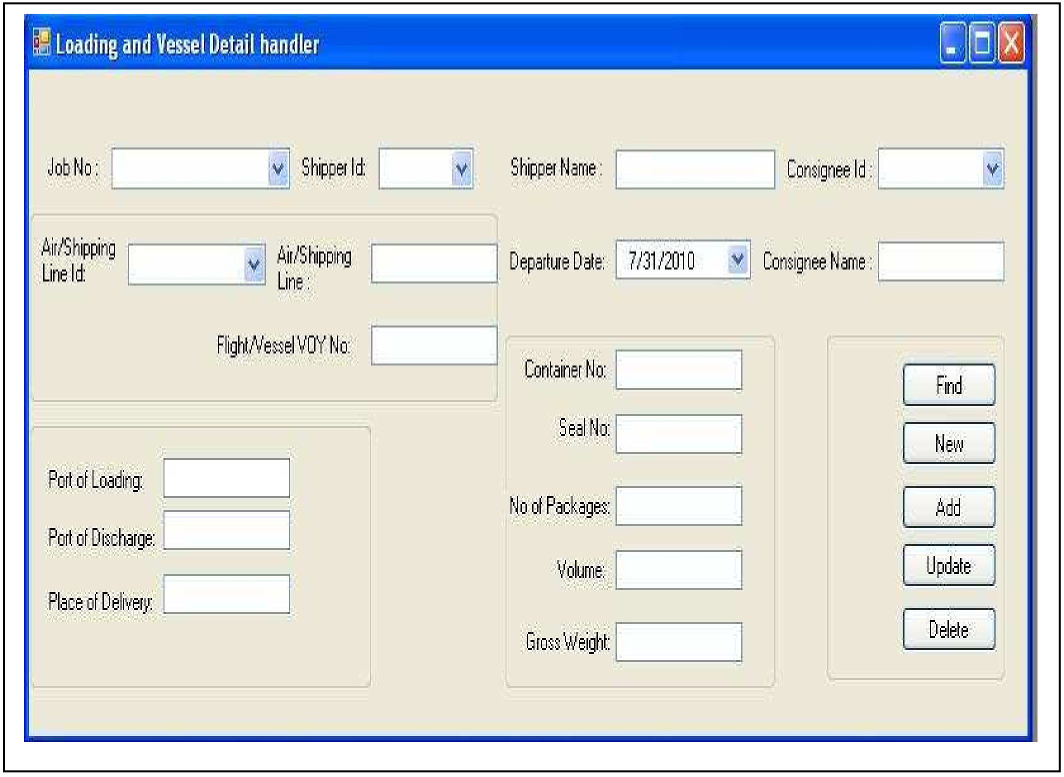
When the goods to ship arrive at the warehouse Inventory is issued by the inventory clerk to package the goods. Packing window header details and the items to be used at

the packing will be entered. After the packing is done remaining packing material will be entered. Total cost for the material will be shown in the Material tab.
Charges tab of this “Calculate Cost per Job” window is used to enter the labor cost and other non material expenses involved in the packing job.
Total cost per packing job is at the header Detail section of the window.

- Loading and Vessel detail handler

This form window is used to enter the vessel detail, Container detail and the route detail of the Shipment.

Pre Alert notification to the shipper is made using the detail of this form.



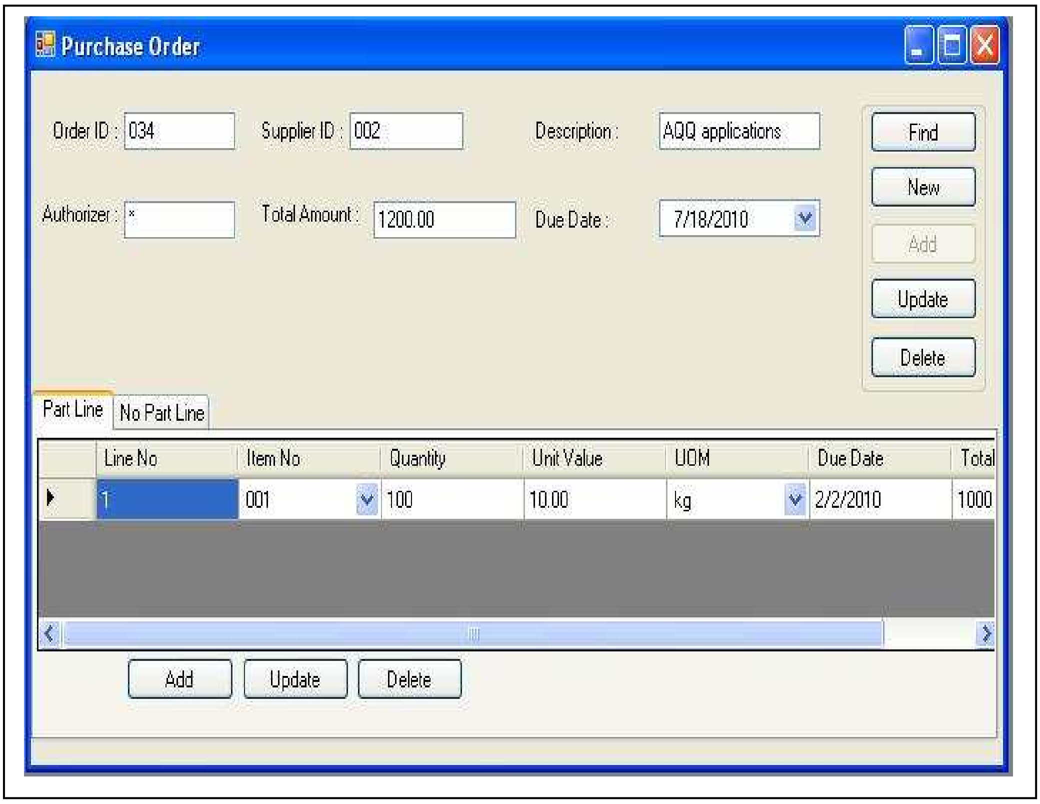
Departure Date: 7/31/2010

Figure C. 11 Loading and vessel detail handler form window

- Purchase Order

Master Detail form Window given below is used to purchase inventory material to the inventory.

After entering the header details user needs to save the record before entering item detail. There are two tabs in this window. One is to enter predefined standard Inventory items and other tab is to enter miscellaneous items.



Order ID : 034

Supplier ID : 002

Description : AQQ applications

Authorizer : *

Total Amount : 1200.00

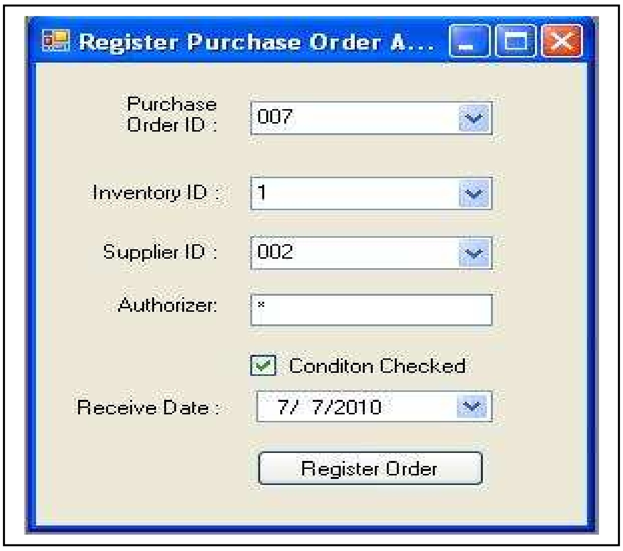
Due Date : 7/18/2010

Line No	Item No	Quantity	Unit Value	UOM	Due Date	Total
1	001	100	10.00	kg	2/2/2010	1000

Figure C. 12 purchase order from window

Register Purchase Order Arrival

Window shown below is used to register received purchase order to the inventory.



Purchase Order ID : 007

Inventory ID : 1

Supplier ID : 002

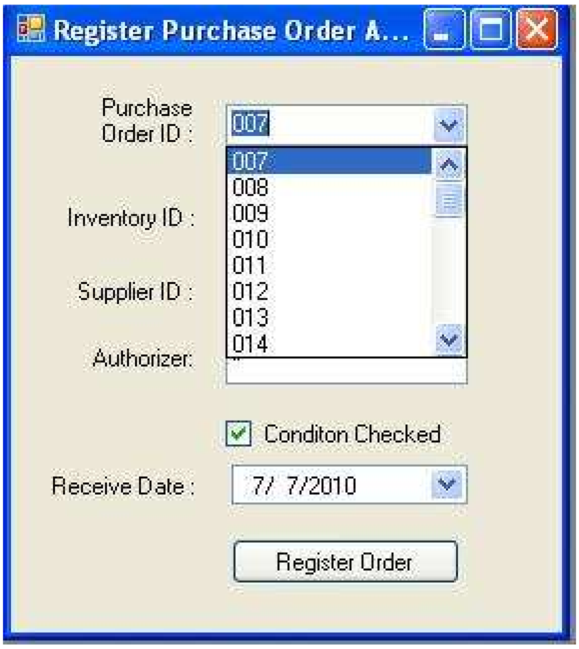
Authorizer: *

☑ Conditon Checked

Receive Date : 7/ 7/2010

Figure C. 13 Register purchase order arrival window

This is an operational window and the job of this window is to assign the purchased items to the inventory. In this window purchase order ID combo box can filter and show only the remaining purchase orders to be registered.



Purchase Order ID : 007

007
008
009
010
011
012
013
014

Inventory ID :

Supplier ID :

Authorizer:

☑ Conditon Checked

Receive Date : 7/ 7/2010

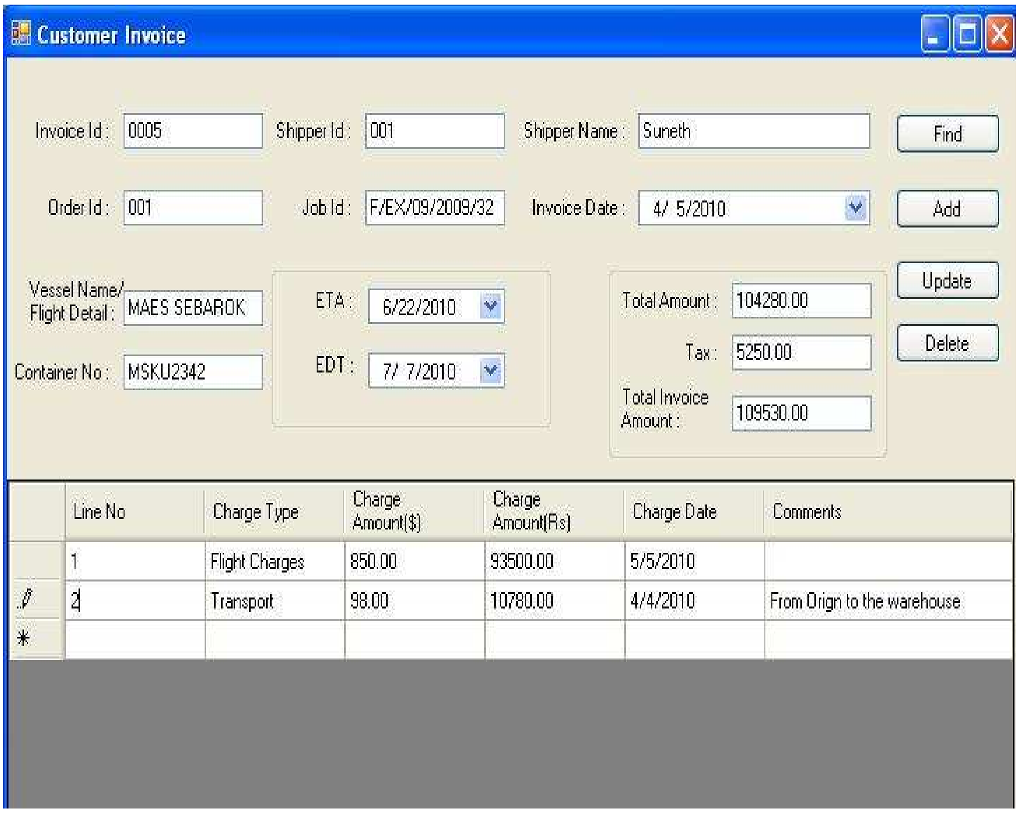
Figure C. 1 Drop down list with orders to receive

- Customer Invoicing

Form window used for the customer invoicing is given below. All the charges involved in this shipment handling process is given in the table format.

In customer invoicing currency rates other than the Sri Lankan Rupees has to be considered. Currently, system does not support automatic currency conversion.

Currency rates must be calculated manually and entered into the system.



Invoice Id : 0005
Shipper Id : 001
Shipper Name : Suneth
Order Id : 001
Job Id : F/EX/09/2009/32
Invoice Date : 4/ 5/2010
Vessel Name/ Flight Detail : MAES SEBAROK
Container No : MSKU2342
ETA : 6/22/2010
EDT : 7/ 7/2010
Total Amount : 104280.00
Tax : 5250.00
Total Invoice Amount : 109530.00

Line No	Charge Type	Charge Amount($)	Charge Amount(Rs)	Charge Date	Comments
1	Flight Charges	850.00	93500.00	5/5/2010	
2	Transport	98.00	10780.00	4/4/2010	From Orign to the warehouse

Figure C. 14 Customer Invoice form window

- Report Generation

This section is to describe the various reports generated by the System.

This system facilitates some dynamic report generation based on the parameters provided by the user.

This mechanism is used to send several documents in the shipment handling flow to the customers

Mangers can obtain some summarize reports of the current business to get management decisions User can either view the reports generated from the system through default report viewer in the system. These reports can be seen in pdf format after exporting the generated report to a specific report archive folder in the system.

Some of the generated reports are,

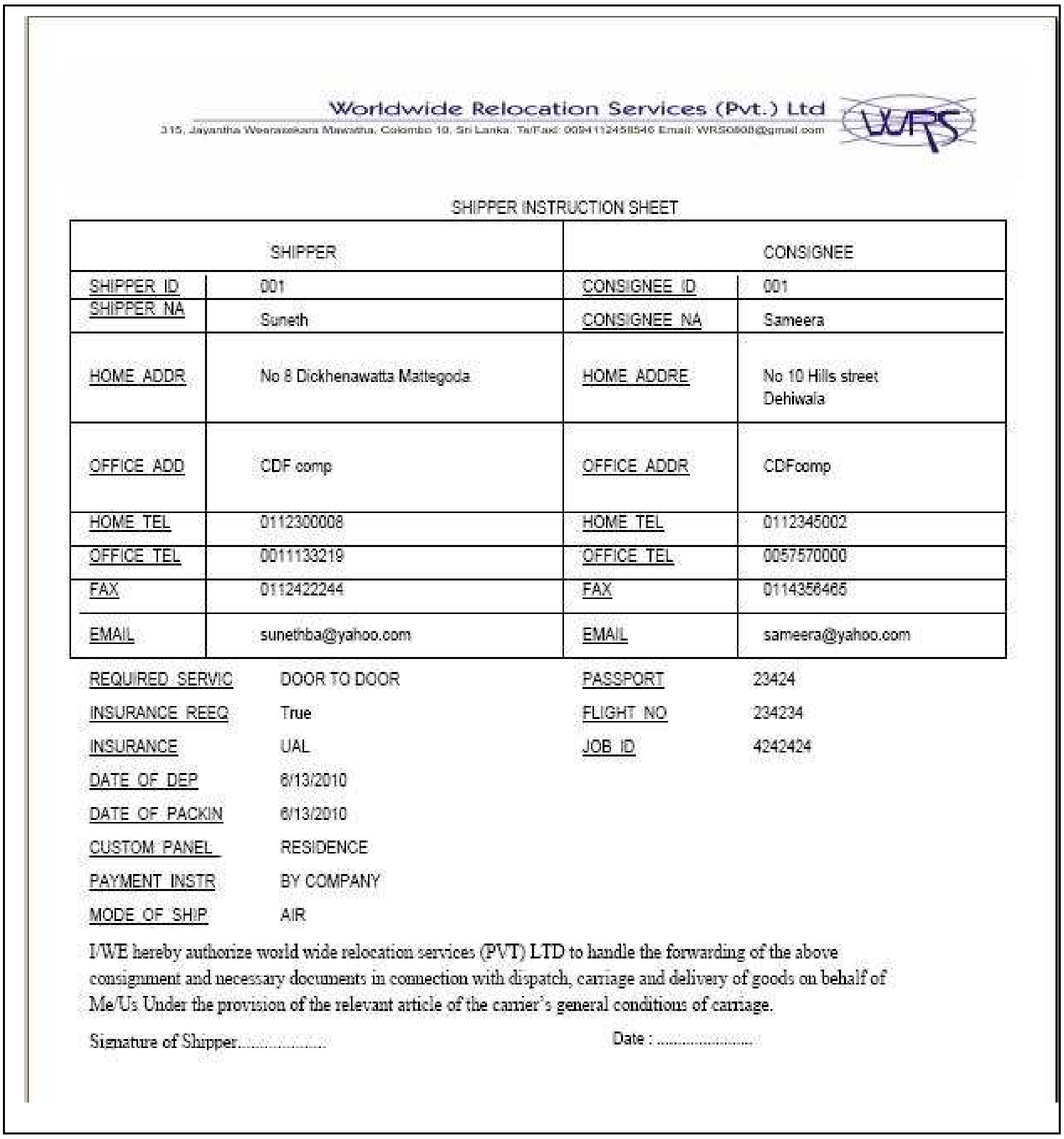
- Shipment instructions
- Pre alert notice
- Material Cost sheets
- Load charts
- Invoices

Reports for the Management purposes are,

- Total Customer base
- Available Suppliers

- Profit share calculator per each job

Report links are located at,
Info Services/
Sample Shipment instruction sheet for a job is given below.



Worldwide Relocation Services (Pvt.) Ltd
315, Jayantha Weerasekara Mawatha, Colombo 10, Sri Lanka. Te/Fax: 0094112458546 Email: WRS0808@gmail.com

SHIPPER INSTRUCTION SHEET

SHIPPER		CONSIGNEE	
SHIPPER ID	001	CONSIGNEE ID	001
SHIPPER NA	Suneth	CONSIGNEE NA	Sameera
HOME ADDR	No 8 Dickhenawatta Mattegoda	HOME ADDRE	No 10 Hills street Dehiwala
OFFICE ADD	CDF comp	OFFICE ADDR	CDFcomp
HOME TEL	0112300008	HOME TEL	0112345002
OFFICE TEL	0011133219	OFFICE TEL	0057570000
FAX	0112422244	FAX	0114356465
EMAIL	sunethba@yahoo.com	EMAIL	sameera@yahoo.com

REQUIRED SERVIC	DOOR TO DOOR	PASSPORT	23424
INSURANCE REEQ	True	FLIGHT NO	234234
INSURANCE	UAL	JOB ID	4242424
DATE OF DEP	6/13/2010		
DATE OF PACKIN	6/13/2010		
CUSTOM PANEL	RESIDENCE		
PAYMENT INSTR	BY COMPANY		
MODE OF SHIP	AIR		

I/WE hereby authorize world wide relocation services (PVT) LTD to handle the forwarding of the above consignment and necessary documents in connection with dispatch, carriage and delivery of goods on behalf of Me/Us Under the provision of the relevant article of the carrier's general conditions of carriage.

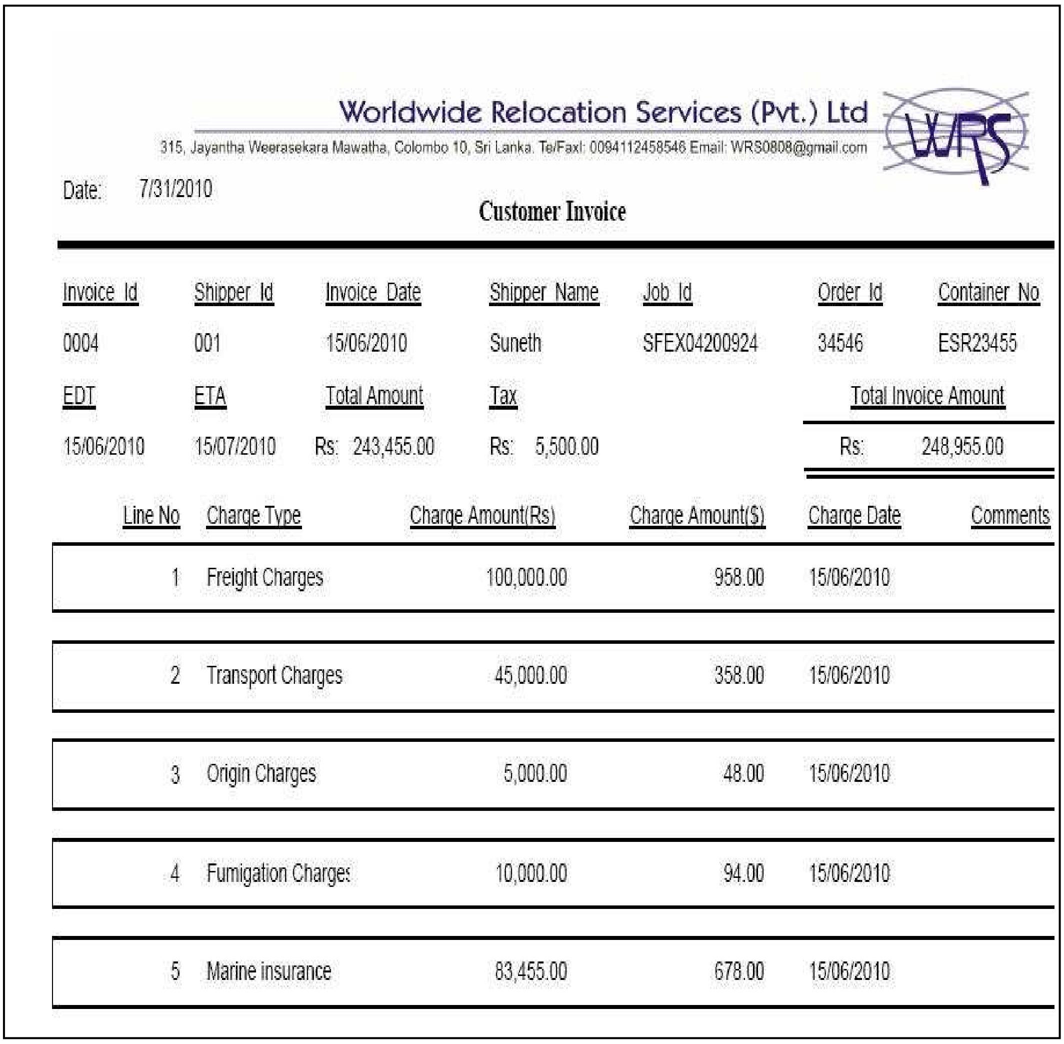
Signature of Shipper.................... Date : ....

Figure C. 15 Shipment instructions sheet

- Customer Invoice

After the shipment is ready to be shipped Customer will be invoiced.

The invoice is shown below



Worldwide Relocation Services (Pvt.) Ltd

315, Jayantha Weerasekara Mawatha, Colombo 10, Sri Lanka. Te/Fax!: 0094112458546 Email: WRS0808@gmail.com

Date: 7/31/2010

Customer Invoice

Invoice Id	Shipper Id	Invoice Date	Shipper Name	Job Id	Order Id	Container No
0004	001	15/06/2010	Suneth	SFEX04200924	34546	ESR23455

EDT	ETA	Total Amount	Tax	Total Invoice Amount
15/06/2010	15/07/2010	Rs: 243,455.00	Rs: 5,500.00	Rs: 248,955.00

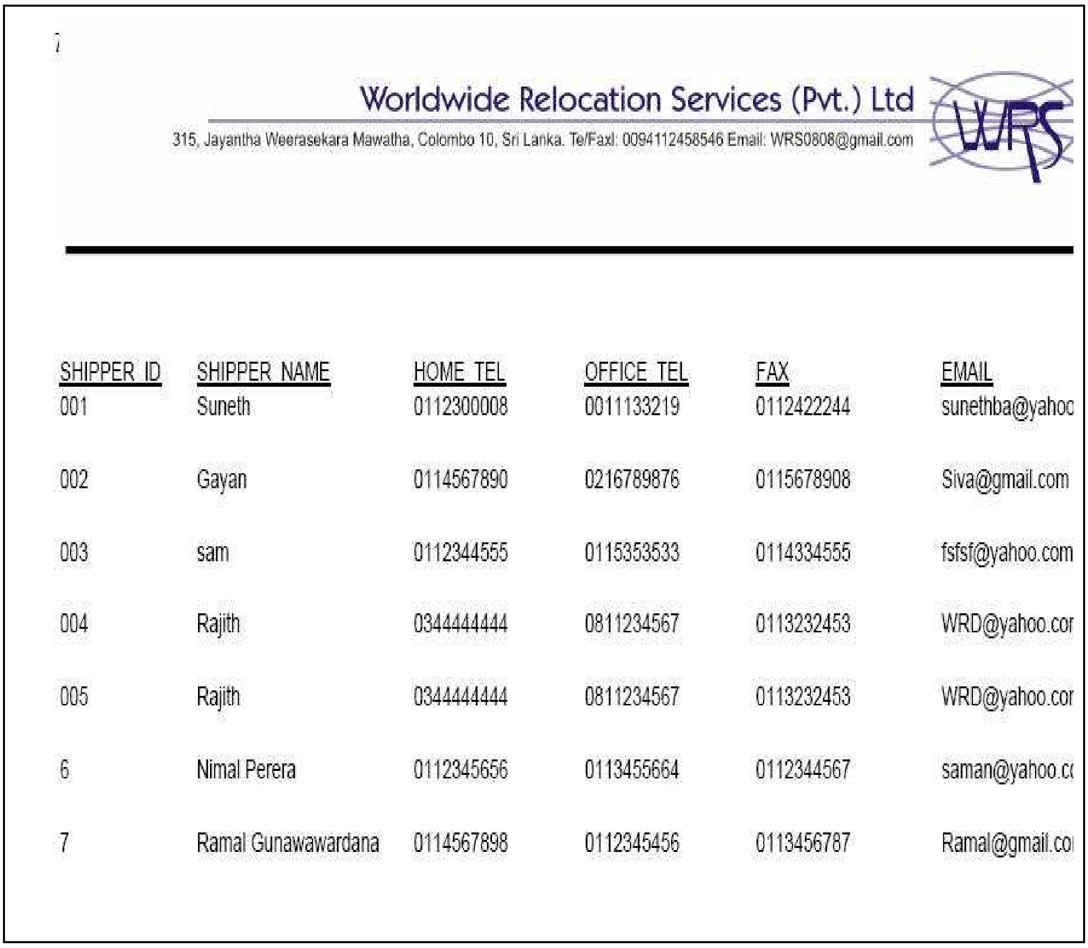
Line No	Charge Type	Charge Amount(Rs)	Charge Amount($)	Charge Date	Comments
1	Freight Charges	100,000.00	958.00	15/06/2010	
2	Transport Charges	45,000.00	358.00	15/06/2010	
3	Origin Charges	5,000.00	48.00	15/06/2010	
4	Fumigation Charges	10,000.00	94.00	15/06/2010	
5	Marine insurance	83,455.00	678.00	15/06/2010	

Figure C. 1 Customer Invoice

- MANAGEMENT REPORTS

Following Reports will be generated for the Managerial purposes.

- Summary of Customers Registered with the system



Worldwide Relocation Services (Pvt.) Ltd

315, Jayantha Weerasekara Mawatha, Colombo 10, Sri Lanka. Te/Faxl: 0094112458546 Email: WRS0808@gmail.com

SHIPPER ID	SHIPPER NAME	HOME TEL	OFFICE TEL	FAX	EMAIL
001	Suneth	0112300008	0011133219	0112422244	sunethba@yahoo
002	Gayan	0114567890	0216789876	0115678908	Siva@gmail.com
003	sam	0112344555	0115353533	0114334555	fsfsf@yahoo.com
004	Rajith	0344444444	0811234567	0113232453	WRD@yahoo.cor
005	Rajith	0344444444	0811234567	0113232453	WRD@yahoo.cor
6	Nimal Perera	0112345656	0113455664	0112344567	saman@yahoo.c
7	Ramal Gunawawardana	0114567898	0112345456	0113456787	Ramal@gmail.co

Figure C. 16 Shipper Detail Summery

- Profit share calculator per Job
- Summary of Suppliers registered with the system

APPENDIX D: CODE LISTING

Consignee Client logic:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
using BusinessLogic;
namespace Client
{
    public partial class frmCConsignee : Form
    {
        BusinessLogic.BConsignee consignee;
        public frmCConsignee()
        {
            InitializeComponent();
        }

        public frmCConsignee(String consigneeid)
        {
            InitializeComponent();
            FindRecords(consigneeid);
            btnAdd.Enabled = false;
            btnFind.Enabled = false;
        }
        // add new record
        private void btnAdd_Click(object sender, EventArgs e)
        {
             try
             {
                 consignee = new BusinessLogic.BConsignee();
                 consignee.Consignee_id = 
txtConsigneeid.Text.ToString();
                 consignee.Consignee_Name = 
txtConsigneeName.Text.ToString();
                 consignee.C_Residence_Add = 
txtResidenceAdd.Text.ToString();
                 consignee.C_Office_Add = 
txtOfficeAdd.Text.ToString();
                 consignee.C_Residence_Tel = 
txtResidenceTel.Text.ToString();
                 consignee.C_Office_Tel = 
txtOfficeTel.Text.ToString();
                 consignee.C_Fax = txtFax.Text.ToString();
                 consignee.C_Email = txtEmail.Text.ToString();
                 consignee.Add();
                 MessageBox.Show("Going from hear");
             }
             catch (Exception err)
             {
                 MessageBox.Show(err.Message.ToString());
             }


        }
        //find exsisting record
        private void btnFind_Click(object sender, EventArgs e)
        {
            try
            {
                String consigneeID = txtConsigneeid.Text.ToString();
                consignee = new BusinessLogic.BConsignee ();
                DataSet ds = consignee.Find(consigneeID);
                DataRow row;
                row = ds.Tables[0].Rows[0];

                //via looping
                foreach (DataRow rows in ds.Tables[0].Rows)
                {
                    txtConsigneeid.Text =
rows["CONSIGNEE_ID"].ToString();
                    txtConsigneeName.Text =
rows["CONSIGNEE_NAME"].ToString();
                    txtResidenceAdd.Text =
rows["HOME_ADDRESS"].ToString();
                    txtOfficeAdd.Text =
rows["OFFICE_ADDRESS"].ToString();
                    txtResidenceTel.Text =
rows["HOME_TEL"].ToString();
                    txtOfficeTel.Text =
rows["OFFICE_TEL"].ToString();
                    txtFax.Text = rows["FAX"].ToString();
                    txtEmail.Text = rows["EMAIL"].ToString();
                }


            }
            catch (Exception err)
            {
                MessageBox.Show(err.Message.ToString());
            }
        }
        //update record
        private void btnUpdate_Click(object sender, EventArgs e)
        {
            try
            {
                consignee = new BusinessLogic.BConsignee();
                consignee.Consignee_id =
txtConsigneeid.Text.ToString();
                consignee.Consignee_Name =
txtConsigneeName.Text.ToString();
                consignee.C_Residence_Add =
txtResidenceAdd.Text.ToString();
                consignee.C_Office_Add =
txtOfficeAdd.Text.ToString();
                consignee.C_Residence_Tel =
txtResidenceTel.Text.ToString();
                consignee.C_Office_Tel =
txtOfficeTel.Text.ToString();
                consignee.C_Fax = txtFax.Text.ToString();
                consignee.C_Email = txtEmail.Text.ToString();
                consignee.Update();

                MessageBox.Show("Data updated ...");
            }
            catch (Exception err)
            {
                MessageBox.Show(err.Message.ToString());
            }
        }
        //delete record
        private void btnDelete_Click(object sender, EventArgs e)
        {
            consignee = new BusinessLogic.BConsignee();
            consignee.Consignee_id = txtConsigneeid.Text.ToString();
            consignee.Delete();

        }

        //find record logic
        private void FindRecords(String consigneeid)
        {
            try
            {
                consignee = new BusinessLogic.BConsignee();
                DataSet ds = consignee.Find(consigneeid);
                DataRow row;
                row = ds.Tables[0].Rows[0];

                foreach (DataRow rows in ds.Tables[0].Rows)
                {
                    txtConsigneeid.Text =
rows["CONSIGNEE_ID"].ToString();
                    txtConsigneeName.Text =
rows["CONSIGNEE_NAME"].ToString();
                    txtResidenceAdd.Text =
rows["HOME_ADDRESS"].ToString();
                    txtOfficeAdd.Text =
rows["OFFICE_ADDRESS"].ToString();
                    txtResidenceTel.Text =
rows["HOME_TEL"].ToString();
                    txtOfficeTel.Text =
rows["OFFICE_TEL"].ToString();
                    txtFax.Text = rows["FAX"].ToString();
                    txtEmail.Text = rows["EMAIL"].ToString();
                }


            }
            catch (Exception err)
            {
                MessageBox.Show(err.Message.ToString());
            }


        }

        //clear the content
        private void RecursiveClearControls(Control.ControlCollection
cc)
        {
            foreach (Control ctrl in cc)
            {
                if (ctrl is TextBox)
                {
                    TextBox tb = ctrl as TextBox;

                    if (tb != null)
                        tb.Clear();
                }
                else if (ctrl is ComboBox)
                {

                    ComboBox tb = ctrl as ComboBox;
                    if (tb != null)
                        tb.Text = "";
                }

                else
                {
                    RecursiveClearControls(ctrl.Controls);
                }
            }

        }
        //intialize the window
        private void btnNew_Click(object sender, EventArgs e)
        {
            int numC;
            RecursiveClearControls(this.Controls);
            consignee = new BusinessLogic.BConsignee();
            numC = consignee.Get_Max_Consignee_id();
            txtConsigneeid.Text = numC.ToString();

        }
    }
}
```

Consignee Bussiness logic

```
public class BConsignee
    {

        private DataLogic.DConsignee ConsigneeData;
        private String consignee_id;
        private String consignee_name;
        private String c_residence_add;
        private String c_office_add;
        private String c_residence_tel;
        private String c_office_tel;
        private String c_fax;
        private String c_email;
        public String Shipper_id;

        public BConsignee()
        {
            ConsigneeData = new DataLogic.DConsignee(this);
        }

        public String Consignee_id
        {
            get
            {
                return this.consignee_id;
            }
            set
            {
                try
                {
                    this.consignee_id = value;
                    if (this.consignee_id == "")
                    {
                        throw new Exception("Please provide ID ...");
                    }
                }
                catch (Exception e)
                {
                    throw new Exception(e.Message.ToString());
                }
            }
        }

        public String Consignee_Name
        {
            get
            {
                return this.consignee_name;
            }
            set
            {
                this.consignee_name = value;
                if (this.consignee_name == "")
                {
                    throw new Exception("Please provide consignee
name ...");
                }

            }
        }
public void Add()
        {
            ConsigneeData.Add(this);
        }

        public DataSet Find(String str)
        {
            DataSet data = null;

            data = ConsigneeData.Find(str);

            return data;

        }
}
}
```

APPENDIX E: TEST CASES

- User Login Functionality

Test Case No	Test	Expected Result	Status
1	Login without providing details	Show login error message	Ok
1	Login with incorrect credentials	Show login error message	Ok
2	Login with correct credentials	Allow user login to the system	Ok

- Handling Entity details in Shipper, Consignee and Supplier Forms

Test Case No	Test	Expected Result	Status
1	User press new Button	ID of the new entity Object that is going to save must auto generated and show in the filed	Ok
1	Try to tab out without providing valid Contact number	Show error message near the Contact number	Ok
2	Saving a record without correct details in the other fields	Show error message with appropriate missing detail	Ok
3	Saving the record with correct detail	Persistence data must be enter into the database	Ok
4	User quarry for a recode by Providing correct ID	Record must be populated with correct results	Ok
5	Update the record with invalid data	Show error message with appropriate missing detail	Ok
6	Update the record with valid data	Save record successfully	Ok
7	Delete record	Confirmation message to delete the record If yes Delete Else No	Ok
8	Unauthorized deletion	Show error message with that user does not have	

		privileges to delete	
	Unauthorized Update	Show error message with that user does not have privileges to Update	Ok
	Authorized Update	Save the record	Ok
	Authorized Delete	Delete the record	Ok

- Shipment Instruction Detail Handling

Test Case No	Test	Expected Result	Status
1	User press new Button	Generate new shipment Order ID	Ok
2	Saving a record without correct details in the header fields	Show error message with appropriate missing detail	Ok
3	Enable Disable Insurance Type	Insurance name filed must enable or disable	Ok
4	Saving the header record with correct detail	Persistence data must be entered into the database	Ok
5	Try to enter Order Item without saving the header detail	Disable the buttons in item tab	Ok
6	Try to enter Order Item after saving the header detail	Enable Add button	Ok
7	User presses add button	New line will be added to the item table and display auto generate line ID	Ok
8	Press “Update” button in the item table	Save record	Ok
9	User Update item details	Save the record	
10	User Delete item Line	Pop up warning message	Ok
11	Confirm the delete	Delete the record	Ok
12	Quarry for the record with incorrect Order ID	Popup information message” No records found”	Ok
13	Quarry for the record with correct Order ID	Populate data	Ok
14	User tab out in the Shipper ID	Populate shipper data in Shipper tab	Ok

15	User tab out in the Consignee ID	Populate Consignee data in Consignee tab	Ok
16	Delete recode in Open state With items attached to	Not allow to delete	Ok
17	Delete recode in Open state and no items attached to	Delete record	Ok
18	Delete recode in Other states	Not allow to delete	Ok
19	Quarry for the record with correct Order ID	Populate data	Ok
20	Send the order to “Accepted” state by unauthorized user	Not allow to do that	Ok
21	Send the order to “Accepted” state by authorized user	Perform state transition	Ok
22	Send the order to “Received” state by unauthorized user	Not allow to do that	Ok
23	Send the order to “Received” state by authorized user	Perform state transition	Ok
24	Send the order to “Packed” state by unauthorized user	Not allow to do that	Ok
25	Send the order to “Packed” state by authorized user	Perform state transition	Ok
26	Send the order to “Shipped” state by unauthorized user	Not allow to do that	Ok
27	Send the order to “Shipped” state by authorized user	Perform state transition	Ok
28	Send the order to “Completed” state by unauthorized user	Not allow to do that	Ok
29	Send the order to “Completed” state by authorized user	Perform state transition	Ok

- Purchase Order Handling

Test Case No	Test	Expected Result	Status
1	User press new Button	Generate new Purchase Order ID	Ok
2	Saving a record without correct details in the header fields	Show error message with appropriate missing detail	Ok
3	Saving the header record with correct detail	Persistence data must be entered	Ok

		into the database	
4	Try to enter purchase order Item or no part item without saving the header detail	Disable the buttons in part item tab and no part item tab	Ok
5	Try to enter purchase order Item after saving the header detail	Enable Add button	Ok
6	Try to enter purchase order no part item after saving the header detail	Enable Add button	Ok
7	Try to enter purchase order part item after saving the header detail	Enable Add button	Ok
8	User presses add button in part tab	New line will be added to the part item table and display auto generate line ID	Ok
9	User presses add button in no part tab	New line will be added to the no part item table and display auto generate line ID	Ok
10	Press "Update" button in the part item table	Save record	Ok
11	Press "Update" button in the no part item table	Save record	Ok
12	User Update part item details	Save the record	Ok
13	User Update part item details	Update the Header total Amount	Ok
14	User Update no part item details	Save the record	Ok
15	User Update no part item details	Update the Header total Amount	Ok
16	User Delete part item Line	Pop up warning message	Ok
17	User Delete no part item Line	Pop up warning message	Ok
18	Confirm the delete	Delete the record	Ok
19	Quarry for the record with incorrect Purchase Order ID	Popup information message" No records found"	Ok
20	Quarry for the record with correct Purchase Order ID	Populate data	Ok
21	Delete recode in Open state With items attached to	Not allow to delete	Ok

- Register Purchase Order Arrival

Test Case No	Test	Expected Result	Status
1	User Select the combo box for Order ID	List the Orders to be registered	Ok
2	User Select the combo box for Inventory ID	List the available Inventory location ID's	Ok
3	User Select the combo box for Supplier ID	List the Suppliers	Ok
4	Saving a record without correct details in the fields	Show error message with appropriate missing detail	Ok
5	Press "Register Order" button	Persistence data must be entered into the database	Ok

- Handle Inventory Location Detail

Test Case No	Test	Expected Result	Status
1	User press new Button	ID of the new Inventory Object that is going to save must auto generated and show in the filed	Ok
2	Try to tab out without providing valid Contact number	Show error message near the Contact number	Ok
3	Saving a record without correct details in the other fields	Show error message with appropriate missing detail	Ok
4	Saving the record with correct detail	Persistence data must be enter into the database	Ok
5	User quarry for a recode by Providing correct ID	Record must be populated with correct inventory details	Ok

6	Update the record with invalid data	Show error message with appropriate missing detail	Ok
7	Update the record with valid data	Save record successfully	Ok
8	Delete record	Confirmation message to delete the record If yes Delete Else No	Ok
9	Unauthorized deletion	Show error message with that user does not have privileges to delete	
10	Unauthorized Update	Show error message with that user does not have privileges to Update	Ok
11	Authorized Update	Save the record	Ok
12	Authorized Delete	Delete the record	Ok

- Packing and Lording

Test Case No	Test	Expected Result	Status
1	User tabs out in Order ID	Populate with relevant data	Ok
2	Saving a record without correct details in the header fields	Show error message with appropriate missing detail	Ok
3	Saving the header record with correct detail	Persistence data must be entered into the database	Ok
4	Try to enter Material Item that is not in the inventory	Show error message and Not Allow to enter	Ok
5	Enter Unit Value and tab out	Populate the total amount for that Item	Ok
6	Try to enter Charge line without saving header detail	Show error message and Not Allow to enter	Ok
7	Try to enter Material line without saving header detail	Show error message and Not Allow to enter	Ok

8	Update the Material Line	Update the total Cost in the header and in the Material tab	Ok
9	Update the Charge Line	Update the total Cost in the header and in the Charge tab	Ok

- Invoice Detail Handling

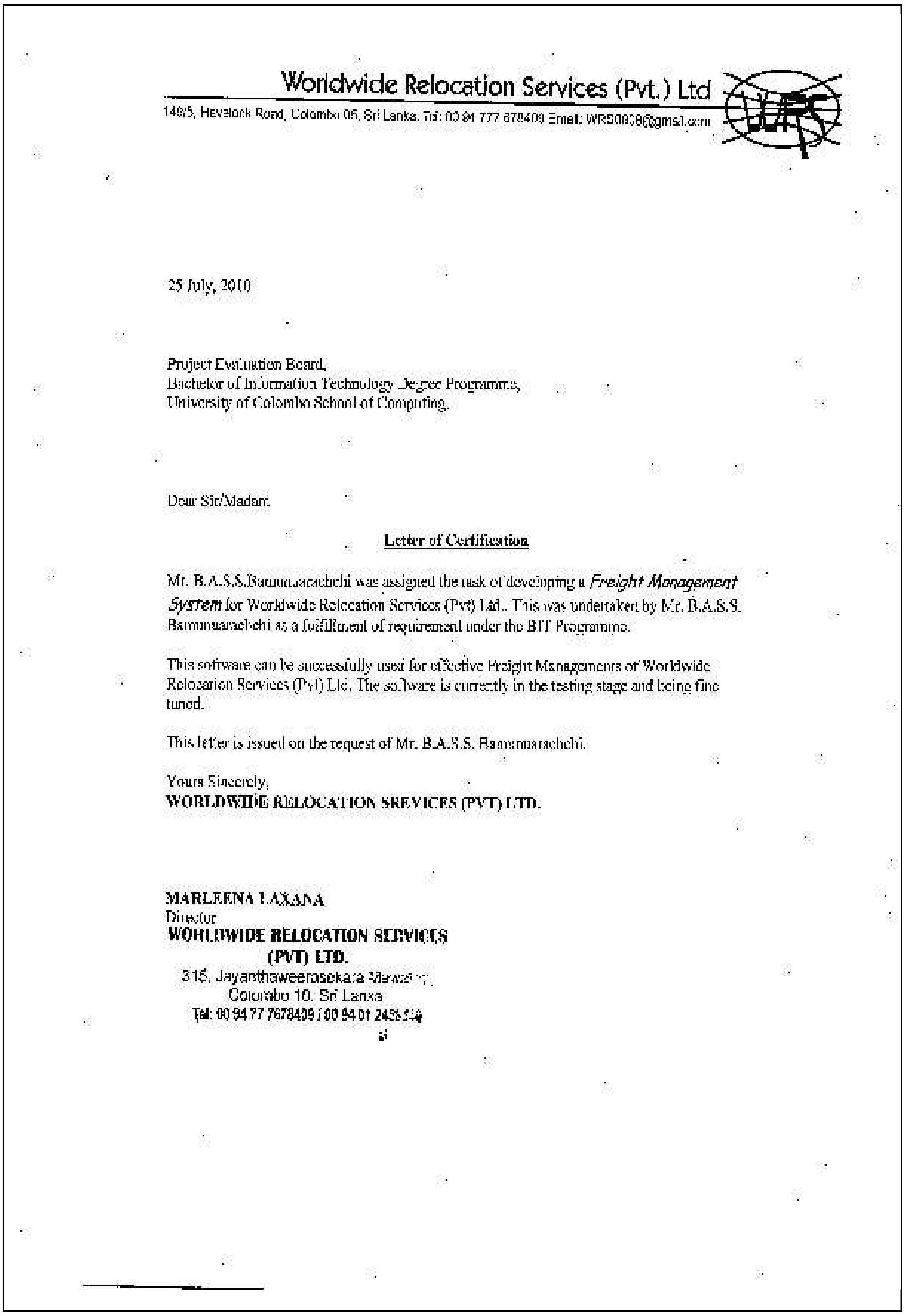
Test Case No	Test	Expected Result	Status
1	User press new Button	ID of the new Invoice Object that is going to save must auto generated and show in the filed	Ok
2	Saving a record without correct details in the fields	Show error message with appropriate missing detail	Ok
3	Saving the record with correct detail	Persistence data must be enter into the database	Ok
4	Enter Details of the Changes incur	User should allow to save the records one by one	Ok
5	Update the record with invalid data	Show error message with appropriate missing detail	Ok
6	Update the record with valid data	Save record successfully	Ok
7	Delete record	Confirmation message to delete the record If yes Delete Else No	Ok
8	Unauthorized deletion	Show error message with that user does not have privileges to delete	
9	Unauthorized Update	Show error message with that user does not have privileges to	Ok

		Update	
10	Authorized Update	Save the record	Ok
11	Authorized Delete	Delete the record	Ok

- Reports Generation

Test Case No	Test	Expected Result	Status
1	User Generate Shipper Instruction	Shipper instructions should be generated for the Given Job	Ok
2	User Generate Pre Alert notice	Pre Alert notice should be generated for the Given Job	Ok
3	User Generate Supplier Details	Report should generate with all the suppliers	Ok
4	User Generate Shipper Details	Report should generate with all the shipper details	Ok
5	User Generate Consignee Details	Report should generate with all the consignee details	Ok
6	User Generate Inventory Details	Report should generate with all the inventory location details	Ok
7	User Generate profit share per job	Report should generate a balance sheet with detail	Ok
8	Unauthorized report generation	Should prevent with relevant error messages	
9	Report Export to pdf format	All the report must convert to pdf format	Ok

APPENDIX F: CLIENT CERTIFICATE



Worldwide Relocation Services (Pvt.) Ltd

149/5, Havelock Road, Colombo 05, Sri Lanka. Tel: 00 94 777 678409 Email: WRS0808@gmail.com

25 July, 2010

Project Evaluation Board,
Bachelor of Information Technology Degree Programme,
University of Colombo School of Computing.

Dear Sir/Madam.

Letter of Certification

Mr. B.A.S.S.Bammuarachchi was assigned the task of developing a *Freight Management System* for Worldwide Relocation Services (Pvt) Ltd.. This was undertaken by Mr. B.A.S.S. Bammuarachchi as a fulfillment of requirement under the BIT Programme.

This software can be successfully used for effective Freight Managements of Worldwide Relocation Services (Pvt) Ltd. The software is correctly in the testing stage and being fine tuned.

This letter is issued on the request of Mr. B.A.S.S. Bammuarachchi.

Yours Sincerely,
WORLDWIDE RELOCATION SREVICES (PVT) LTD.

MARLEENA LAXANA
Director

WORLDWIDE RELOCATION SERVICES
(PVT) LTD.
315, Jayanthaweerasekara Mawatha,
Colombo 10. Sri Lanka
Tel: 00 94 77 7678409 / 00 94 01 2456 [illegible]