

Table of Contents

Chapter 1 Introduction	11
1.1 Topic of the System.....	11
1.2 Purpose of the Situation	11
1.3 Target User of the System.....	13
1.4 Topic Background.....	13
1.5 Problem Context.....	14
1.6 Description of the Problem Area.....	15
1.7 Rationale behind the System	16
1.8 Project Objectives	18
1.9 Academic Objectives.....	18
1.10 Traceability Matrix between problems and functionalities	19
1.11 Assumptions	19
1.12 Success Criteria	19
1.13 Project Scope	20
1.14 Limiting the Project Scope	20
1.15 Functionalities of the System	21
1.16 Feasibility Analysis	22
1.17 Project Planning.....	23
1.18 Ethical Issue.....	24
Chapter 2 Problem Description.....	25
2.1 Introduction to Problem Area	25
2.2 Problems Identified	25
2.3 Problem Importance and Justification	25
2.4 Challenges in the Project	26

2.5 New Concepts, Theories and Technologies to be learnt.....	26
Chapter 3 Literature Review	28
3 Literature Review.....	28
3.1 Advanced Preliminary	28
3.3 Depth of Secondary Research.....	30
3.5 Research on Similar Systems.....	32
3.5 Quantitative and Qualitative Matrix	34
3.6 Market Value of the Project.....	34
Chapter 4 Research Methods	36
4.1 Primary Research	36
4.1.1 Questionnaires.....	36
4.1.2 Interviews.....	37
4.2 Secondary Research	37
4.2.1 Technical Research	37
4.2.1.1 Programming Language Research	38
4.2.1.1.1 Visual Basic, Java and C++ comparison based on project	38
4.2.1.1.2 Reasons for choosing Visual Basic .NET for Beacon	39
4.2.1.1.3 Packages to be used	40
4.2.2 Methodology: Advanced Waterfall.....	41
4.2.2.1 Description	41
4.2.2.2 Justification	41
4.2.3 Algorithms: Why Particular Algorithm?	42
4.2.3.1 Reasons for choosing Genetic Algorithm for Route Planning.....	42
4.2.3.2 Reasons for choosing Bin Packing Algorithm for Load Planner (CLP).....	43
4.3 Academic Research.....	44
Chapter 5: Research Analysis	45

5.1 Questionnaire Analysis	45
5.1.1 Questionnaire Conclusion.....	45
5.1.2 Recommendation	45
5.2 Interview Analysis	46
5.2.1 Interview Conclusion.....	46
5.2.2 Recommendation	46
5.3 Traceability Matrix for Questionnaire and Functionality Mapping.....	47
Chapter 6: System Design.....	48
6.1 Use Cases	49
6.1.1 Use Case Context Diagram.....	50
6.1.2 Use Case Specification (Design Route Map).....	51
6.1.3 UC-01 Use Case Diagram (Design Route Map).....	52
6.1.4 AD-01 Activity Diagram	53
6.1.5 SD-01 Sequence Diagram (Design Route Map)	54
6.1.6 UC-02 Use Case (Plan Route)	55
6.1.7 UC-02 Plan Route Map Use Case Diagram.....	56
6.1.8 AD-02 Activity Diagram (Plan Route)	57
6.1.9 SD-02 Sequence Diagram (Plan Route)	58
6.1.10 UC-03 Use Case Specification (XML Editor)	59
6.1.11 UC-03 Use Case Diagram (XML Editor)	59
6.1.12 AD-03 Activity Diagram (XML Editor).....	60
6.1.13 SD-03 Sequence Diagram (xml editor).....	61
6.1.14 UC-04 Use Case Specification (Manage Loading).....	62
6.1.15 UC-04 Use Case Diagram (Manage Loading).....	63
6.1.16 AD-04 Activity Diagram (Manage Loading)	64
6.1.17 UC-05 Use Case Specification (View Loading).....	65

6.1.18 UC-05 Use Case Diagram (View Loading)	66
6.1.19 AD-05 Activity Diagram (View Loading).....	67
6.1.20 SD-04 Sequence Diagram (View Loading)	68
6.1.21 UC-06 Use Case Diagram (Change Language)	69
6.1.22 UC-06 Use Case Diagram (Change Language)	70
5.1.23 AD-06 Activity Diagram (Change Language).....	70
6.1.24 SD-05 Sequence Diagram (Change Language)	71
6.2 Class Diagrams	72
6.3 Database Design Diagram.....	73
6.3.1 Data dictionary (TSP)	73
6.3.2Data Dictionary (CLP)	73
6.4 Test Design	73
6.4.1 Test Plan	75
Chapter 7: Implementation	76
7.1 Tools used for Implementation	76
7.1.1 Implementation Plan	76
7.2 Implementing Complex Modules	78
7.2.1 Create Random Population (TSP).....	78
7.2.1.1 Objective	78
7.2.1.2 Description: This module basically covers the following subparts:	79
7.2.1.3 Flow of the module:	79
7.2.2. Crossover (TSP).....	80
7.2.2.1 Objective:	80
7.2.2.3 Flow of the Module:.....	82
7.2.3 Map Editor - GUI Drawing (TSP)	83
7.2.4 Container loading Algorithm – Bin Packing Algorithm.....	85

7.3 Problems (Hardest Task).....	88
7.3.1 Problem 1: (Mapping TSP with GA and approach of solving)	88
7.3.2 Problem 2 (Generating the Graph of Loading)	93
7.4 Technical Quality.....	98
7.4.1 Memory Utilization.....	98
7.4.2 Performance	98
7.4.3 User Interface.....	98
7.4.4 Short-Cut Keys.....	98
7.4.5 Naming Convention.....	98
7.4.6 Comments	99
7.4.7 Validations	99
7.4.8 User Manual.....	101
7.4.9 Technical Manual.....	107
Chapter 8 Testing.....	109
8.1 Test Plan.....	109
8.2 Test Duration.....	110
8.3 Unit Testing	111
8.3.1 Black Box Testing.....	111
8.3.2 White Box Testing	120
8.4 System testing	123
8.4.1 Full System Testing	123
8.5 Compatibility Testing.....	126
8.6 Usability Testing.....	126
8.7 User Acceptance Testing	130
8.7 Testing Summary	131
Chapter 9 Critical Evaluation.....	132

9.1 Introduction.....	132
9.2 Usefulness of the Beacon for Target Users.....	132
Chapter 10 Conclusion.....	133
10.1 Degree of Success	133
10.2 Limitations and Errors	134
10.3 Possible Future Enhancement	134
10.4 Main Computational Challenge	134
10.5 Learning Experiences.....	134
References.....	135
Books / Articles and White Paper References	135
Website Reference	136

Figure List

Figure 1: Travelling Salesman Problem (Weisstein, 1999)	12
Figure 2 : 3 Dimensional Container Loading graph (F. Parre, 1994)	12
Figure 3: Analogy between a numerical GA and biological genetics (Haupt, 2004).....	14
Figure 4: 2D Load packer Software.....	29
Figure 5: Market Value of Container Loading Calculator.....	35
Figure 6 : Advanced Waterfall follows V model.....	41
Figure 7: Bin Packing Example (Gordon, 2011)	43
Figure 8 : Use Case Context Diagram	50
Figure 9: UC-01 Use Case Diagram (Design Route Map)	52
Figure 10: AD-01 Activity Diagram (Design Route Map)	53
Figure 11 : SD-01: Sequence Diagram (Design Route Map)	54
Figure 12: UC-02 : Use Case Diagram (Plan Route).....	56
Figure 13: AD-02 Activity Diagram (Plan Route)	57
Figure 14: SD-02 Sequence Diagram (Plan Route)	58
Figure 15: UC-03 Use Case Diagram (XML Editor).....	59
Figure 16:AD-03 Activity Diagram(xml editor).....	60
Figure 17: SD-03 Sequence Diagram (xml editor)	61
Figure 18: UC-04 Use Case Diagram (Manage Loading)	63
Figure 19: AD-04: Activity Diagram (Manage Loading)	64
Figure 20: UC-05 Use Case Diagram (View Loading)	66
Figure 21: AD-05 Activity Diagram (View Loading)	67
Figure 22: SD-04 Sequence Diagram (View Loading).....	68
Figure 23: UC-06 Use Case Diagram (Change Language)	70
Figure 24: AD-06 Activity Diagram (Change Language)	70
Figure 25: SD-05 Sequence Diagram(Change Language).....	71
Figure 26: Class Diagram (Beacon).....	72
Figure 27: Test Plan PERT chart	75
Figure 28 : Implementation Plan.....	77
Figure 29 : General Scheme of Evolutionary Process	78
Figure 30 Figure of Running Algorithm of Route Map.....	85
Figure 31 : Distance calculation between two co-ordinates using RMS method	89
Figure 32 : logic and code of Mutation in GA.....	92

Figure 33: Initial and default parameters of TSP in GA	93
Figure 34: Picture Box used as a Drawing Surface by Usercontrol	94
Figure 35: Code and logic for Best-Fit Algorithm.....	95
Figure 36 : Code Snippet from Draw Bins Sub procedure to draw graph	96
Figure 37 : Change events calling compute method.....	97
Figure 38 : Error Message because of less number of cities.....	99
Figure 39: Showing the validations used in TSP	100
Figure 40 : testConnectionValid function determines the validity of clink between cities ...	101
Figure 41 : Type of Testing	109

List of Tables

Table 1: Traceability Matrix between Problems and Functionalities	19
Table 2: Core Functionalities Table.....	21
Table 3: Enhanced Functionalities Table.....	21
Table 4: Special Functionalities Table.....	21
Table 5: 2D Load packer, Similar System	33
Table 6: Opti-Map (similar systems)	33
Table 7: Language Suitability Table and comparison	38
Table 8 : Packages to be used	40
Table 9 : Traceability Matrix between Questionnaire, Interviews and Functionalities	47
Table 10 : Traceability Matrix between Requirement Analysis and design	49
Table 11 : Use Case List	51
Table 12: UC-01 Use Case Specification (Design Route Map)	51
Table 13: UC-02 Use Case Diagram (Plan Route)	55
Table 14: UC-03: Use Case Specification (xml editor)	59
Table 15: UC-04 Use Case Specification (Manage Loading)	62
Table 16: UC-05 Use Case Specification (View Loading).....	65
Table 17: UC-06 Use Case Specification (Change Language).....	69
Table 18: Test Plan Sample	74
Table 19 : Table of Tools Used.....	76
Table 20 : Pseudo code of Population Class.....	79
Table 21: Pseudo code for population and bestTour	80
Table 22 : Partially Mapped Crossover Demonstration.....	82
Table 23: Pseudo Code for Links in Link Class (TSP).....	83
Table 24 :Pseudo code for XML based city loading in map.....	84
Table 25:Pseudo Code for new City generation in pictureBox	85
Table 26: Packages used and pseudo code for decreasing order of bins	86
Table 27: Pseudo code for Best Fit Algorithm and decreasing order of bin.....	88
Table 28 : Pseudo code for finding the closet city	89
Table 29 : Fitness Calculation code for each tour.....	90
Table 30: List of methods used in drawing the loading graph.....	97
Table 31 : Use of List and Inheritance for memory utilization.....	98

Table 32: Showing naming convention used in code	98
Table 33: Showing the used of xml comments to describe code	99
Table 34 : User Manual Table	106
Table 35: Technical Manual Table	108
Table 36: Type of testing involved	110
Table 37 : Test Duration Table	110

List of Equations

Equation 1 Equation to create random population.....	79
Equation 2 : Fitness Equation	90

Chapter 1 Introduction

The past two decades is the era of the personal computers and information technology .

The mathematical algorithm is no longer just participating with our world – they shape it. Algorithms are used in calculation, data processing and automated reasoning. From Financial market to small mobile device, everything uses a smart algorithm to reduce the complexity of some problem and make the technology looks wonderful. But there are questions that remain unanswered? How the Information Technology Field will change in the next 10 years. If we will find some algorithmic approach to solve a problem that is best suited and most optimized.

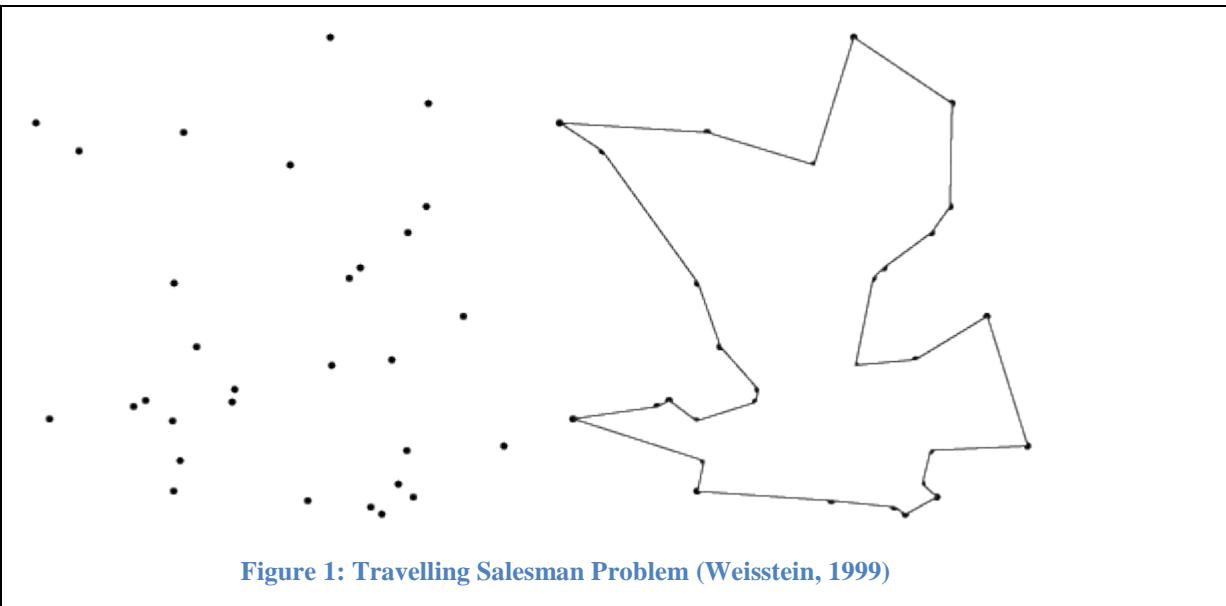
1.1 Topic of the System

The Primary goal of the project “Beacon (Hi-Tech Marine Loading and Path Detection System)” is to resolve the logical and mathematical complexities in the Logistics System. The project provides a new approach of solution to two popularly known NP-hard problems, Travelling Salesman Problem (TSP), Container Loading Problem (CLP). Beacon keeps everything organised, updated, smart and accessible.

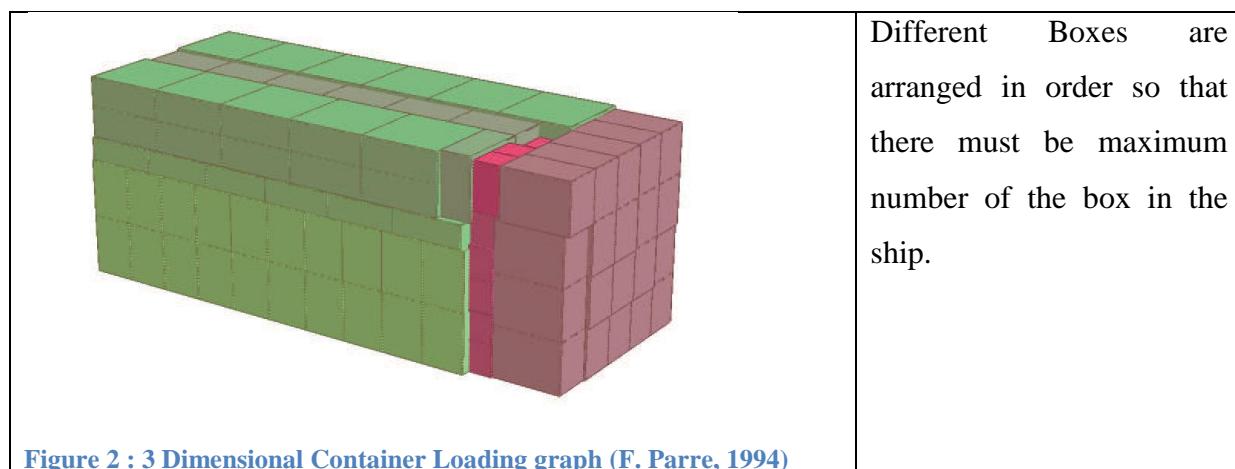
1.2 Purpose of the Situation

The Developer is sensing immense scope and intelligent system name Beacon that uses advanced genetic and greedy approach of solving the problem, a new algorithm is generated to solve the complex and unsolvable problem. The approach results a optimize solution of the scenario.

Every Logistics company that transfer heavy goods materials to different destinations plans the routing of the vehicle (Trucks/Ships) to manage the cost of travel. This cost is very critical to the business process and can save lots of money. Route Planner is basically required here to get the optimized and shortest route. Since the number of stops are uncertain and a very large number of stoppage can cause a very problematic situation to find the best possible solution. This problem is commonly known as Travelling Salesman Problem in Algorithms.



The second algorithm developer is doing is related to marine loading. Large numbers of container are being loaded to ships and other vehicle for supply. It is always a tedious task for loading staff to manage the containers and make a optimize load plan. The load plan should be made so that the maximum number of container gets load without affecting the balance of the ship. The problem arises due to different number of containers get transported with various size of goods materials.



1.3 Target User of the System

Target users are group of those people for whom the system is going to be developed for.

- + **Shipping Companies/SMEs /Courier Company** - Organization who manages a large supply or export/import packages to move on.
- + **Skilled People targeted** – Package planner, Package Loader, Insurance People, Manager, Accounts. The system also targets different worker of same enterprise according to work divided.
- + **General/Typical Users** -Any person who wants to control their data can use the system.
- + **Research peoples and Students** – The advance algorithm and problem solving techniques can be used in academics and research.

1.4 Topic Background

The basic idea of the system is to solve the complexity in the enterprises in route mapping and container loading. Both the Idea is unsolvable due because of being a NP hard problem that cannot be solvable and verifiable in polynomial time. To solve this algorithm in the feasible time, a backtracking approach needs to be applied. Genetic algorithm is one of the various effective ways to approach for solutions of NP-hard and NP-complete problems.

Bin Packing Algorithm is a mathematical way to deal with efficiently fitting Elements into Bins. The hitch is that Bin packing problem is classified as the NP-complete problem. Basically means that there's no way of being guaranteed the best solution without checking every possible solution. The goal of every Bin Packing algorithm is to use the least amount of Bins to hold the required number of Elements. (Dirk, 2009)

Genetic Algorithm is a search heuristic algorithm that is routinely used to generate useful solutions to optimizations and search problems. It generated solutions to optimization problems using techniques inspired by natural evolution, such as inheritance, mutation, selection and crossover. Genetic algorithms are one of the best ways to solve a problem for which little is known. (Bajpai, 2008)

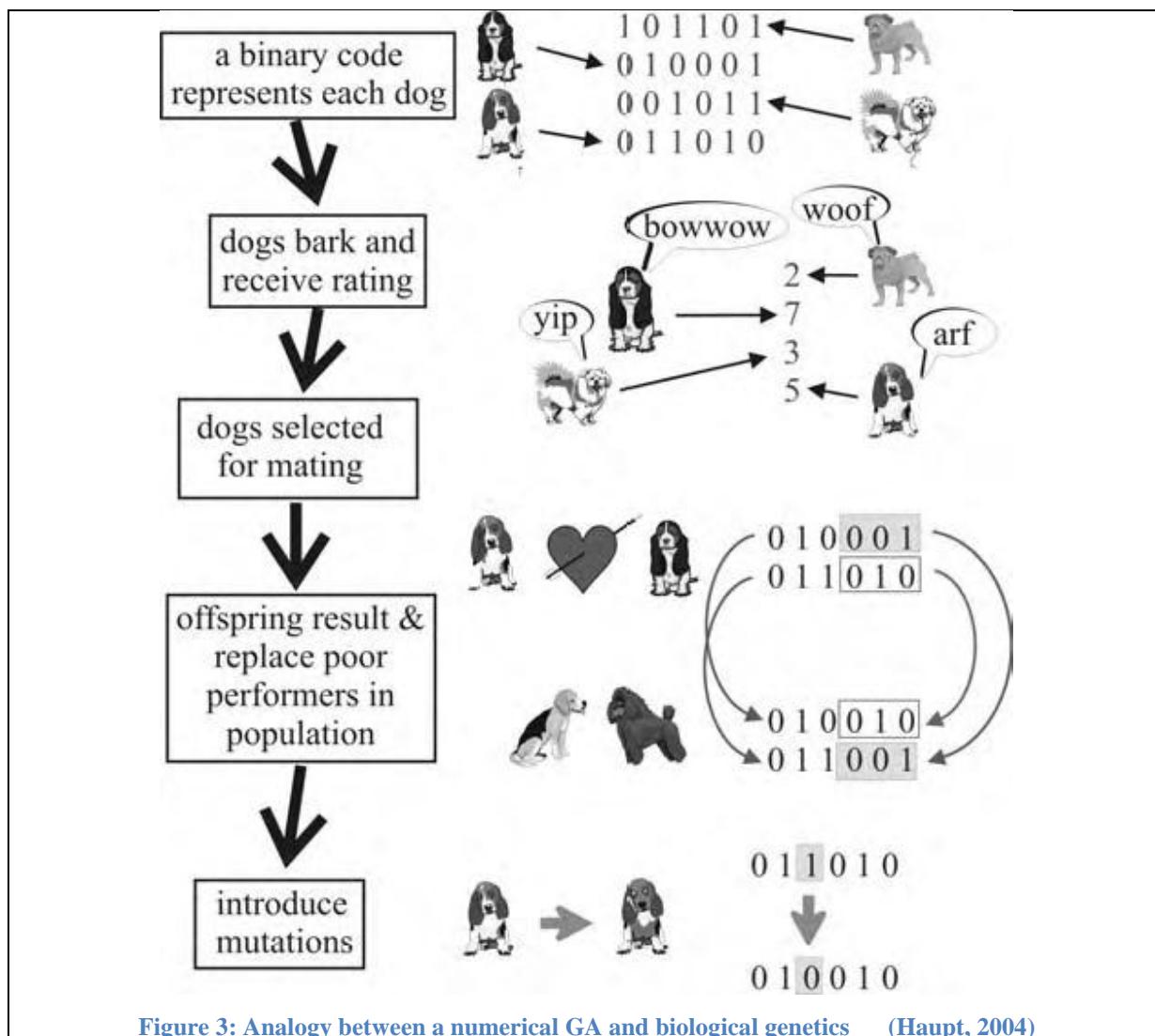


Figure 3: Analogy between a numerical GA and biological genetics (Haupt, 2004)

1.5 Problem Context

In the present world, Information System and calculations plays a very vital role in every place. Every real world problem can be linked as an entity and the corresponding relationships can be defined in between.

Beacon is responsible to solve the complexity of the problems and system of any logistics company. Its solves the mind jolting and tedious tasks automatically using its advanced algorithms and data flow design .Complex problems like TSP (travelling salesman problem) CLP (container loading problem) can't be solved manually or any other way .Beacon keeps everything organised ,updated ,smart and accessible.

1.6 Description of the Problem Area

- ⊕ To enter data of the goods to be transported and organizing the relationships between data is a complex and time-taking tasks. Every goods from nail to big industry parts are needed to be logged into the system with its specifications and related data .What are the materials going to transfer in which container and vehicle.
- ⊕ The information related to the goods, transported like basic price of goods, charge of transportation, value, size, weight, owner, destination, insurance papers etc. This data are needed to be arranged in the normalized manner to solve the data redundancy and availability.
- ⊕ Efficient algorithm for solving the TSP with the facility to import map coordinate or create a weighted graph. To solve the problem with the advanced genetic algorithm is also a tedious and brainy task that includes various constraints like time, amount of goods, traffic etc.
- ⊕ Second most important algorithm need to generate to solve CLP of ships. To generate whole new algorithms is the research part in the project .Also the system is responsible to generate 2D/3D graphs to show the efficient container loading .This features include a high level of challenge and skills.
- ⊕ Implementing the unique ID generation for the transport and the goods is also a challenge, these Ids can be used to track and update the status. Also small mobile app can also be used for the customer of the company to manage their goods.
- ⊕ Features like language support and import of data through excel sheets, interactive reports are also the problem areas which are not clear.
- ⊕ **Setting an example** – Mr Sharma is a famous business man from India. His company exports various things like natural resource (coal, iron ores, etc.), agriculture product, couriers, vehicles and everything from ships and trucks. He needs to cut the expenses from the TSP and CLP problem in the system .Company can saves thousands of dollars if they have efficient algorithms for the best loading and supplying .And also they need the interactive system to deal with the various constraints, data and reports of every transportsations.

1.7 Rationale behind the System

As obvious from the problem context presented above the situation warrants such a system that helps reducing the complexity , solve the TSP , CLP problem that is impossible to solve in any manual way or by supercomputers . In TSP and CLP the problem grows exponentially fast in every turn that makes its unsolvable when the inputs of data is in more in numbers. When the destination in TSP is more than 20 or in CLP .It's very tough to create the optimal solution of the problem using any straight approach.

Setting an example: A company like DHL is known for exports/imports business and they need the system to plan, organize and track the goods. Beacon is perfect in this situation .It's also has small mobile app for users to track the products.

Benefits of the System

The system implemented could bring about significant tangible and intangible benefits. Given below is a list of tangible and intangible benefits expected from the system:

Tangible Benefits

- **Reduced Complexity of Problem-** This system generates the optimal solution that seems impossible to solve manually or technically .TSP and CLP problems are being solved with the most efficient problem solution being designed and implemented by developer.
- **Graphical view of route and container loading –** The system will generate 2D/3D view of the containers to show the best loading graph in CLP and the best route graph in TSP. Its gets easier to load container using graph and route map makes the route clear.
- **Organized Data and accessibility –** The Data of every container is very much organized and validated during entry and can be used to plan a consignment. Also every goods/material will gets easily to track as per the unique id provided .The status can be made available for the customer in the current time.
- **Cost Benefits -** Cost being “the most” important factor for any organization, it's always beneficial to adopt cost-cutting or cost-saving measures. With Beacon, cost benefits come in following ways-

Most efficient route will be produced, results in precious fuel saving hence cost cutting. Best Container loading plan optimizes the overall loading with the ships and also maxims the number of container .Hence this reduces the administration and managerial overhead results into cost benefits

- ❖ **Time Benefits-** with Beacon, the time of container loading reduced efficiently as everything will be decided by the system will be pre-planned. Report generation and task assignments are easier to discuss and also proper planning and scheduling make the most out of work in minimum time. Trucks and ships save time by going through the shortest path provided results in more resource availability. **For Example:** More the trucks reduce the time of travelling; more the loading of trucks will be possible. Also the container loading takes the minimum time in ships.

Intangible Benefits

Increased Employee /Enterprise productivity – With employee being aware about the task and work division among the organization, it results into employee task productivity. For Example – The Work load is very much reduced by the algorithms of TSP /CLP and also the planning tool makes the productivity easier .The Consignment planner can easily arrange every box and goods.

- ❖ **Increased work flexibility and easier to understand and operate** – The work load assigned are easier to plan and manage and keep track of the work progress .Also the Beacon makes it easier to understand the problem statement and to organize and plan the things based on different constraints. **For example** - Desirable production and better planning can be targeted as the vision of the project.
- ❖ **Increased customer satisfaction** - All organizations churn out some services whose final consumer is the customer .With Beacon promising to deliver so much for improving the productivity and uptimes for the businesses they are surely going deliver better results for their end customers thereby promoting customer satisfaction and customer loyalty thereby increasing business for the organizations.
- ❖ **Reduces stress-** – The automated advanced application that works on backtracking can automatically reduce the stress and work overhead of the users.

1.8 Project Objectives

- To learn advanced programming techniques, software engineering principles, project management concepts, advanced algorithms and integration of varied technologies.
- Learn about domain area, i.e. Genetic algorithms and Greedy Approach of solving graphs in algorithms
- To implement the knowledge gained in Objective 1 to develop such a system which leads to goal mentioned above.
- To make efforts to ensure that the system being developed will fulfil all the tangible and intangible benefits mentioned in the previous section.

The Proposed system intends to allow engineers to manage their container loading provided with the unique id of every container. The system manages all the data using an integral optimum methods to organise them as the weight remain same and the overall container loading doesn't affect the balance of the ship or anything .In second stage it provides the route map of the ship. All the input data are verified and validated with unique ID. Custom report generation, language support, data entry through excel/xml files and other are some other enhanced features. The important feature of the system is the graph generation of the container loading in 2D/3D.

The modules for the proposed system have been listed below:

Software Module: This module will be installed on desktop machine of the organization .Different software are divided mainly for data entry, container loading and route map and report generators.

Database Module: This module will save every single consignment for the company with details of final results of TSP and CLP.

1.9 Academic Objectives

- To learn advanced algorithmic approach like greedy algorithms, genetic algorithms and implementation of algorithms.
- To create a research paper and whole new approach to solve the travelling salesman problem and container loading problem.
- Project management concepts like methodologies, scheduling, time estimation, work breakdown techniques and tools etc.

- Implementation of the algorithms derived into visual basic to create a running application while learning the programming background of the technology.
- To make efforts to ensure that the system being developed will fulfil all the tangible and intangible benefits mentioned in the previous section.

1.10 Traceability Matrix between problems and functionalities

Problems	Solution
Map Design	Editor to draw a weighted graph of route
Best Route	Best optimized route generator using Genetic Algorithm
Container Information	Data entry and Validate of each container
Optimized Load Plan	Best solution using Bin Packing Algorithm
View Load	2D /3D view of the final loading
Different Language	Language Support
Easy data entry	Using Excel sheets/Export –Import / XML
Document Generation	Printable document generation as a report

Table 1: Traceability Matrix between Problems and Functionalities

1.11 Assumptions

The successful implementation of the project is much depends on the understanding of the user towards the system. So as this system is developed for a logistics company so it is believed that all the users are expert user and know about it. Since developer is providing a solution based on his own research, so there may be other best possible way to solve the problem. One or more than one user may be responsible to use the system, developer is using work break structure of company. Predefined boxes and map are included in the system to show the usability, however one can change specification according to their needs. Language support is very vast. The system can be installed and ready to be used and developed to work on windows environment. The final output/result of route/load plan can be saved in image or pdf file and can be copied or used elsewhere.

1.12 Success Criteria

Success Criteria depends upon the depth of understanding and experience gained from this project and how efficiently developer solves the problem described above.

1.13 Project Scope

The proposed solution is desktop-based standalone application for solving to algorithmically complex task. The system can be downloaded or used by directly installing into the windows machine. The final output or result produced by using the application can be used as image or word files and can be used later

The features of the application are –

- **Editor for graph Design:** This will be the simple editor to design a weighted graph on the system .It includes toolbox and other specification of node and path in the graph.
- **Route Optimizer Module:** This is particularly the research part in the system .It create the major algorithmic approach that will take data from the editor or predefined map and solve the problem to create the best optimized path.
- **Data entry and Validate Module:** This module can be used to create new specification of boxes (weight, size etc) and the good contained in every box. All the entry must be verifiable cannot exceed the specification.
- **Loading Planner Module:** This module is the main feature and research part in the system that used advanced algorithmic approach to solve the loading plane based on the data entry and validate module. This will result a best optimized load plan for the ship or trucks.
- **Graph generator Module:** The final 2D /3D graph of loading is generated in this module using the load planner module.
- **Language Support Module:** More than 3 languages support are provided in the system. All the name, word will be changes by selecting one option.

1.14 Limiting the Project Scope

- The application is strictly desktop based standalone application include no support from the external source or web. Although the result of route and load can be used as image or word file.
- There is no support for third party application or Google map.
- There is no major database role in the system, it can use some simple excel file or MS Access records.

1.15 Functionalities of the System

CORE FUNCTIONALITIES

#	Functionality	Description
1.	Data entry and Unique ID	The information about each Routes and container need to be logged in the system with unique identification.
2.	New Algorithms for TSP (Research Paper)	Intelligent Genetic Algorithm to determine the shortest and the best path for transportation. Mapping real world Natural Selection process into programming to solve TSP (Refer to Appendices for Research Paper)
3.	Implementation of TSP	Implementation of TSP problem on graphs taking co-ordinates from a real world scenario
4.	Algorithms Design for CLP	Bin Packing Algorithm that solves CLP that provides the optimal container loading solution and ship balance.
5.	Implementation of CLP	Implementation of Container loading for a ship with various box of different size

Table 2: Core Functionalities Table

ENHANCED FUNCTIONALITIES

#	Functionality	Description
1.	Language Support	Different Major Language Support for the system.
2.	XML Data Reader and Writer	Data entry through xml files
3.	Interactive Report and Document Generation	Automatic and downloadable Insurance , Bill , Transport Agenda and other report generation

Table 3: Enhanced Functionalities Table

SPECIAL FUNCTIONALITIES

#	Functionality	Description
1.	2D /3D view of CLP	Loading Plan of containers
2.	Map Editor for TSP	Map Editor for TSP

Table 4: Special Functionalities Table

1.16 Feasibility Analysis

Feasibility studies addresses things like where and how the business will operate. It provides in-depth details about the business to determine if and how it can succeed and serve as a valuable tool for developing a winning business plan.

Schedule Feasibility Report

It consist of assess the work to which the time frame and the completion date for all major activities within the project that meets organizational deadlines and constraints for affecting change.

The main part of schedule feasibility report is-

Gantt chart - Developer has used this chart for the time estimation.

Technical Feasibility

Technical Feasibility mainly involves the hardware and the software requirements of the system regarding all the tasks of the system to be done

Minimum Hardware Requirement

- CPU: 2.0 GHz Pentium IV or above,
- Memory: RAM 512 MB
- Disk Space: 5 GB space (at least)
- Monitor: Any Standard Monitor
- Peripherals: Mouse, Keyboard

Software Requirement

- Operating System: Windows® 7, Windows XP,
- Software Development Tools: Microsoft Visual Studio 2010,
- Project Management Tools: Microsoft Project, Visio 2010,
- Case Tools: Smart Draw, Visual Paradigm 6.0,
- Graphic Design Tools: Adobe Photoshop CS4,
- Database : Microsoft SQL Server 2008,
- Documentation: Microsoft Office 2010.

Operational Feasibility

It may be defined as, the process of assessing the degree to which a proposed system solves business problems or takes advantage of business opportunities. Problems addressed and advantages of this system are provided in the documentation.

Economic Feasibility

The purpose for assessing economic feasibility is mainly to identify the financial benefits and costs associated with the development project. Economic feasibility is referred as cost and benefit analysis for any system.

Once the technical feasibility is established, it is important to consider the monetary factors also. Since it might happen that developing a particular system may be technically possible but it may require huge investments and benefits may be less. Cost Benefits are included under Rationale by the developer.

1.17 Project Planning

Project Management

“ Process of planning, directing and controlling the development of an acceptable system at a minimum cost within a specified time frame.” (Whitten et al, 1994)

Time Management

Project started on date and will be completed on the date specified in the academic requirement. Start and End Date is already specified in the Project Development Plan. After estimating the time, activities were sequenced to create the Gantt chart. Activities can be adjusted to meet the academic dead line through change control process. (For Gantt chart see Appendix Section)

Deployment Plan

V-Mode has five stages these are: Concept and operation, Requirement and Architecture, Detail Design, Implementation, Elaboration Phase, Construction Phase and Transition Phase. Below it describes, how there phases are involving in my project. (Time Division for Deployment Plan is fully explained in 4.2.3.3 Section).

1.18 Ethical Issue

It ensures that the project won't do any harm to the society or anyone in anyways. It contains issues like no animal harm & no copying of data without prior permission etc and to ensure that the developer has filled up an ethical form and it will be provided in documentation.

Chapter 2 Problem Description

2.1 Introduction to Problem Area

In this era of Information Technology and Computation ,every little work is expected to done smartly and speedily .Sorting is by no means the only computational problem for which algorithms have been developed .Practical Applications of algorithms are ubiquitous and include various problems like Travelling Salesman Problem and Container Loading Problem.

2.2 Problems Identified

Beacon is going to solve many real time problems that companies face on daily basis as –

- **Easy Data Entry** – Co-ordinate of map need to be easily entered. Specifications of Containers are different and must be defined by shape and weight.
- **Route Map Plan** - Best optimized path of travelling with downloadable plan. Easy solvable for large number of problems.
- **Load Plan** – 2D /3D view of optimized container loading for a single consignment.
- **Path Editor** – Easy Path Editor so that destination can be altered accordingly
- **Language Support** – Different Language Support
- **Consignment Planner** – Easy Planner for each consignment and goods on go.

2.3 Problem Importance and Justification

- **Easy Data Entry** – Since the number nodes entered in route planner is huge, it should be easy and portable. Same in Load planner, there are various boxes of different sizes and shapes. Excel file based entry makes data easier to input. Also the graphical user interface are need to be with the design principle of human interaction .The information entered must be validated and crosschecked before getting a final output.
- **Route Map Plan** - Best optimized path of travelling with downloadable plan. There's no solution that is verifiable within the polynomial time to solve this problem because it grows exponentially at every step. In any Logistics company, the problem is same as Travelling Salesman Problem. The possible paths can be predetermined and can be solved to get the best way when the number of destination is high.
- **Load Plan** – 2D /3D view of optimized container loading for an every consignment need to generate. This is the main feature plus the most tough and complex problem of the project. It is very hard to map a large number of uneven containers into an

algorithm to get a 3D model of optimized load. Also the loading include longitudinal balance of the vehicle /ship and the mass must be distributed evenly.

- **Path Editor** – Editor is the graphical feature and challenging to develop. The entire route can be defined manually in the editor by drawing a weighted graph. It includes direction and weight in the graphs same the distance and direction of the road map.
- **Language Support** – More than 1 language support are provided to the software, so that it can be used by different users who differs in their demographic background.

2.4 Challenges in the Project

The development of Beacon calls for numerous challenges to be faced by the developer which includes learning curve of new concepts, learning new development tools, mastering already known tools, domain analysis, new programming languages etc.

Challenges

The development of Beacon (Hi –tech Marine Loading and Path Detection), calls for numerous challenges to be faced by the developer which includes learning curve of new concepts, learning new development tools, mastering already known tools, domain analysis and new programming languages etc.

2.5 New Concepts, Theories and Technologies to be learnt

Concepts: Various new concepts includes in the project mainly in the research part. Genetic algorithms, Greedy Approach, Heuristic approach of algorithms are the new concepts in the algorithm. Concepts include generation of algorithm, mapping it into programming and generation the right user interface. It includes matching of real word problem as an entity to the scientific approach of problem solving followed by defining the relationship between them to get a best optimized solution.

Theories: The theory is very complex to map with the algorithms and solvable approach. To map the real world problem into the binary or any data structure and generating a backtracking approach is real tough task. All the best solution comes after mutation and theory of natural selection in the genetic approach.

Next challenge is to map this solution of the real world problem into the programming language. It includes various graphs theory and complexity of the programming modules. The parameters of data entry and entity are identified by the data gathering and research theories.

Technologies

Programming Language Skill

Visual Basic – The application running on the windows machine is going to be developed using visual basic. Mastering Visual Basic is a significant challenge given the steep learning curve it requires with pre-requisites involving skills of working with graphs, procedures, class, libraries and implementing algorithmic approach in it. (**Refer to 4.2.1 for Programming Language Research**)

Graph Technology - The identification, generation and working on a graph is the tough task especially when it is random and created by some logics. The graph of TSP is created using edit and the final output of the CLP is some 2D or 3D graph .This part is very hard and need constant learning curve.

Chapter 3 Literature Review

3 Literature Review

A literature review is a description of the literature relevant to a particular field or topic. It gives an overview of what has been said, who the key writers are, what are the prevailing theories and hypotheses, what questions are being asked and what methods and methodologies are appropriate and useful. As such, it is not in itself primary research, but rather it reports on other findings. This contains following things-

3.1 Advanced Preliminary

The Idea of this project came into developer mind while he was learning the applications of algorithms and capability of doing smart stuffs. The idea has been derived in the process of learning a blog on genetic algorithm and its application. Developer thought of this idea could be implemented for big and small logistics company that can be a highly cost and time beneficial .Also developer was quite interested in learning the advanced and scientific approach of algorithms that he knows a little bit from course module.

The developer performed an extensive research of the topic & found IEEE research papers that guided the developer towards the logic and approach on algorithms design and implementation

The excerpts from these research papers are given below-

“The Single Container Loading Problem (CLP) is a three-dimensional packing problem in which a large parallelepiped has to be filled with smaller parallelepipeds, available in different sizes and limited quantities, so that empty space is minimized

When speaking about real-world container loading problems space usage is the most important objective, but other issues have to be taken into account, such as cargo stability, multi-drop loads or weight distribution. Among these additional considerations, cargo stability is the most important one.

In this paper we present a new algorithm for the container loading problem that is based on an original heuristic enhanced by a GRASP solution space search strategy. “
(F Pareno and R. Alvarez ,2009, pp.1-3)

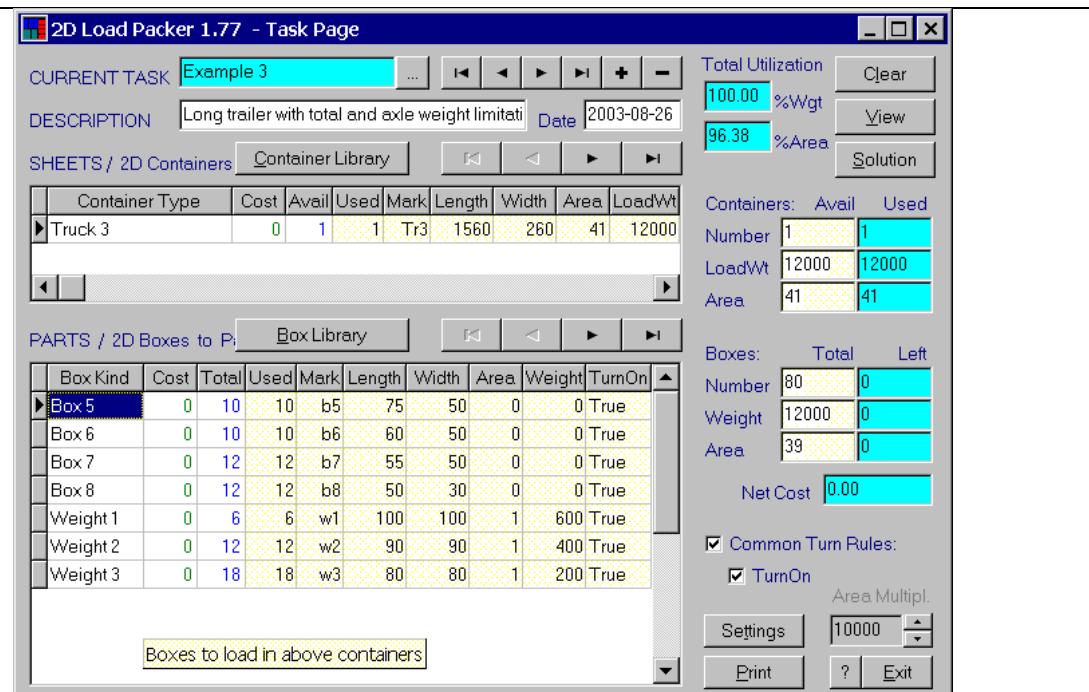
“For solving TSP, numeric, heuristics, genetic, hybrid or other algorithms may be used, as this problem is not easy to solve for large number of cities, the computation is often deployed on multi-processor or clustered hardware”

The problem has to be coded into data structure, which can be handled like a chromosome.

For TSP, the chromosome is set of ordered indexes of cities, through which the traveller goes. For other problems, it could be integer or real number, for difficult tasks, there is idea of using Neural Networks for coding the problem” (Dusan Saiko ,2005 pp. 1-4)

The Developer knew few things before developing the project:

1. **2D load Packer**- It is popular application that is used for creating load plan. It uses some algorithmic approach to solve the problem of loading while taking care of the longitudinal balance of the vehicle in terms of weight. Tasks, Containers and Boxes are entered, edited and manipulated. It generated a load plan in 2D after getting all the validated input from the user.



3.3 Depth of Secondary Research

The developer conducted secondary research from various Books, Websites and Magazines etc. to find out about the development process of proposed system and for this the developer needed to find about following areas-

Domain Research

Topic Background

Algorithms are the key change process in any problem in the market that results into various benefits in today scenario of computation. Particularly genetic approach of solving problem with natural selection is applicable in most of the NP problem and treated as the best approach of solving discovered in most of the cases. Marine Loading and Path Detection is such a system. Kevin Slavin believes the power of algorithms in shaping our world in TED Global Conference 2011

“All the fun is powered by algorithm, we may not realise it but we live in the algorithm world “ (Slavin; 2011)

Conclusion Drawn: Advanced algorithms are a smart changing key to change the way anything computes. It makes the things readable and processed smartly at the same time , that is not possible in any other way .They acquire the sensibility of truth because they repeat over and over again. And they ossify and calcify and they become real.

Travelling Salesman Problem

"If there are n cities a salesman must visit, and the distance between each pair of these cities is given, find the shortest tour where each city is visited exactly once and returning to your starting point." (Anonymous, 2000)

Euler introduced a form of the travelling salesperson problem in 1759, and it was formally named and introduced by the Rand corporation in 1948 (Michalewicz, 1992)

This problem in the route planning of Logistics Company is exactly same as in TSP. To find the best route between different destinations could be best found using genetic algorithm.

Container Loading Problem: a genetic algorithm (GA) for the container-loading problem. The main ideas of the approach are first to generate a set of disjunctive box towers and second to arrange the box towers on the floor of the container according to a given optimization criterion. The loading problem may include different practical constraints. The performance of the GA is demonstrated by a numerical test comparing the GA and several other procedures for the container-loading problem. (Gehring, Borlfeldt 2008, pp 1)

Design Principle: The design principles that need to be studied in order to provide are as follow:

⊕ **Visibility:** Visibility is one of the most important design principles and what it means is that, as and when the user looks on the system screen he/she may feel the possibility for action. The developer will keep an eye on this principal in order to provide better visibility.

Conclusion: The developer will provide colour schemes in a way that users of this system can easily find suitable content.

⊕ **Feedback:** Feedback is the response to the user of the action performed.

Conclusion: The developer will provide appropriate message box and notification to provide feedback.

⊕ **Constraints:** Constraints are some universally accepted conventions which notify some specific actions.

Conclusion: Inclusion of warning messages to show the errors.

⊕ **Consistency:** Consistency means maintaining similar colour schemes, backgrounds, font colour and size etc.

Conclusion: The developer will provide Consistent Colour, Same background, same font size and colour for the whole application

⊕ **Affordances:** Affordance means expected behaviour like cursor of mouse changes into hand symbol on roll-over the link.

Conclusion: The developer will provide features like double clicking etc on editor.

3.4 Programming Language Research

Visual Basic

- Provides very easy GUI creation using its tools, picture box and user control features.
- Provides built in support for data grid view and other custom reporting.
- Easy event handlers, procedures can be designed
- Partial classes and static method can be easily implemented
- Generic library class and Array list can be easily inherited within classes.

3.5 Research on Similar Systems

Contents	Description
Product Name	2D Load Packer
Description	2D Load Packer (2DLP) is the unique space optimizer designed to help you plan quickly and easily the best compact arrangement of a number of different size 2D rectangular objects ("Boxes") within one or more 2D rectangular Container. 2DLP is based on the truly two-dimensional original packing algorithm.
Functionality	<ul style="list-style-type: none">■ The overall load weight limit and truck axle weight limits can be taken into account as the additional constraints or actual optimization factors.■ The program has a facility for specifying the associated cost for each box (part) / container (sheet) item in order to calculate totals and affect the optimization as an additional priority factor. Optimizer goal and other main settings are adjustable.■ The program can optimize over multiple containers (sheets) in multiple sizes at the same time, taking into account overall item set considered, as well as allowed item orientations specified for each item separately or for all together■ The Image Page presents 2D views of the container (sheet) layouts and Area / Weight utilization diagrams. Layout patterns are displayed graphically as the complete adjustable full-colour images. You can observe step-by-step load sequence and print any image view, as well as other task and solution data reports.

	<ul style="list-style-type: none"> ✚ The system supports network data sharing and has some specific DB export / import facilities, allowing a user to exchange selected task / solution data with another 2DLP system.
Limitations (if any)	<p>The program generates only 2D image of the final loading as it takes only one surface to load on.</p> <p>Suited for container of same height only.</p>
User Review	User rated it with 3 stars, but algorithms, speed and designs are really appreciable.
Link -	http://www.astrokettle.com/pr2dlp.html

Table 5: 2D Load packer, Similar System

Conclusion:

- ✚ Container includes the size and amount of goods inside it to the file.
- ✚ 2D application can be generated if the height are the same but it is a limitation
- ✚ Sample Container can be defined and used further.

Contents	Description
Product Name	OptiMap - Fastest Roundtrip Solver
Description	OptiMap is an online application that solves travelling salesman problem and any other cyclic route problem and provides shortest distance route. It is developed using Google API and use latitude/longitude as co-ordinates to solve.
Functionality	<ul style="list-style-type: none"> ✚ Can substitute address for a city making it an intra-urban travelling problem ✚ Can enter the nodes by clicking in Google map interface ✚ Can also enter address manually by using text fields ✚ Can also add text list, all the address at one time ✚ Toogle raw path gives you exact latitude/longitude co-ordinates in the sequence for this route
Limitations	Can solve upto 100 nodes only
User Review	Give 3 out of 10
Link -	http://gebweb.net/optimap/

Table 6: Opti-Map (similar systems)

Conclusion:

- To deal with high number of container it requires connecting with some database.
- Weight distribution can be analysed along the longitudinal axis.
- Loading rules can be imposed using panel permit or something.

3.5 Quantitative and Qualitative Matrix

Features	Beacon	2D Load Packer	OptiMap
Map Editor	✓		✓
TSP(>20)Cities	✓		
Route Graph	✓		✓
Database Support	✓	✓	
XML Reader	✓		
XML Writer	✓		
Container Planner	✓	✓	
Choice of algorithm	✓		
Loading Graph	✓	✓	
Loading Sequence		✓	
Box Size Editor		✓	
Language Support	✓	✓	✓

3.6 Market Value of the Project

There is no product in the market that solves both route complexity problems with travelling salesman problem. The approach of and the solution is still not identifiable as the best possible. Genetic approach is hard to apply but it gives a near to optimized output for larger number of products

Effland from Denmark has rated CLP as five stars in terms of complexity with average budget of \$500 at website (freelance.in).

CONTAINER LOADING CALCULATOR

17 Bids

Bids
17
Avg Bid (USD)
\$468
Project Budget (USD)
\$250-\$750

Freelancer.in, A leading freelancing

Project ID: 380919 Project Type: Fixed

Project Description:
OBJECTIVE
 Creating an application to simulate and compute how to place some goods into some containers so they would be best loaded and optimized.

DESCRIPTION
 Container loading calculator is an application that is intended to compute and to place some goods into container and make their position to be fit / loaded and optimized. User would have to select the kind of cargo along its dimension parameters (width, height, length, weight, and number), each cargo can be packed into some pallet or packed directly into a container. Then the program will display how the goods (cargo or pallet) are positioned within a container. The way they are positioned should be best and fit, it means they must occupy some space if necessary.
 See reference below on the live site example.

REQUIREMENT

- PHP
- Javascript
- Graphic PHP library, for instance GD
- Algorithm to solve, probably graph theory

+ Follow

Project posted by **Efland**

5

SPECIFICATIONS

- User has to select kind of cargo (i.e box, sacks, barrel). We will supply you with our cargo data.
- User is able to add more than 1 cargos, and input with different parameters (width, length, height, weight) for each cargo.
- User is able to select whether to place the cargos directly into container or pre-packed within a pallet first. Pallet parameters (dimension) are also can be changed by user.
- User is able to select whether manually input for the container data or let the application automatically choosing the container data for him. If user selected to manual input, he may add different container (different dimension/parameters). Of course the application should check if the container was too small for the loaded cargos.
- The application displays the final result how the goods are loaded within the container, how much space has been occupied by the goods and available space within container, how many container do we need, and a step by step how we packed the goods. Please note that the way we load the goods should be optimized within a container. We need your expertise in algorithm to solve the problem. This stage is the most important thing in this application.
- For the complete flow, see the reference below on the live site example

Only half of the functionality of this Final year project

REFERENCE
 should end up just like below link
<http://www.searates.com/reference/stuffing>

Skills required:
 .NET , ASP , Graphic Design , Javascript , PHP

Figure 5: Market Value of Container Loading Calculator

Chapter 4 Research Methods

4.1 Primary Research

The chapter focused on primary of the project. Developer has done this in the initial stage primarily for gathering the information from the potential user of the system. The information is critically analysed and documented so that at the end it is user acceptable. Primary researches are done through following techniques.

4.1.1 Questionnaires

Questionnaire (**Refer to appendices for questionnaires**) is a part of data gathering technique in which a series of questions designed to gather specific information. Questionnaires will specially be designed for users who would use the system which will include people who are either staff of the logistics company or on crew of ship. They have few advantages as Questionnaire –

- ⊕ Can be used to collect both qualitative and quantitative data
- ⊕ Can be distributed manually or electronically
- ⊕ Can reach a vast number of people regardless of physical location or geographical dispersion
- ⊕ Can be distributed quickly and cheaply
- ⊕ Can be used when human and financial resources are not available to conduct interviews

Why Questionnaire?

As this application is basically targeting to a specific group of professional who are going to use the proposed system, so they are huge and difficult to contact, so it is not possible to take interview of them because it will be huge time consuming. There are some reasons for using Questionnaires-

- ⊕ The analysis of questionnaire is most efficient than other techniques as we can draw the graphs and analyse the data.
- ⊕ It can reach a vast number of people regardless of physical location or geographical dispersion.
- ⊕ It can be distributed quickly and cheaply.

4.1.2 Interviews

Interview (**Refer to appendices for Interview**) is a part of data gathering technique in which a series of question asked face to face from users of the system, designed to gather detailed information. They have few advantages as Interview –

- Can be used for evaluating information needs as interview will allow us to gather detailed information about functionalities of project
- Can be used for gathering knowledge about perceptions of the staff about the application.

Why Interview?

- It provides us with immediate response which saves the developer's precious time
- It allows participants to express themselves in their own words
- It allows the collection of a large volume of rich data
- It allows discussion, probing and unexpected insights
- It is best for investigating problems

4.2 Secondary Research

Secondary research means finding information from third-party sources such as marketing research, websites, magazine articles, and other sources that is already published or gathered by somebody. Books, journal articles and research paper those are necessary in order to understand the project.

4.2.1 Technical Research

This part is going to research the proposed system technically means what programming language the system is going to use. As the proposed system is an algorithmic application, so only few languages to choose from and these are C++, C#, Visual Basic and Java. To choose the best suitable development language for the proposed system, developer carried out a lot of research on different languages.

4.2.1.1 Programming Language Research

This part is going to research the proposed system about the programming language, the system is going to use. The selection of right platform for development of project is prior requirement of developer. While selecting the programming language the developer takes care of following issues:

- Modules to be developed
- Time of development
- Interface required
- Application programming

Developer had decided to go for Object Oriented Language which supports GUI interface, and can reduce developer's time in designing the interface itself. The best options available with him are:

4.2.1.1.1 Visual Basic, Java and C++ comparison based on project

Comparison	Visual Basic	Java	C++	Python
Learnable	Very Easy	Hard to Easy	Easy to Hard	Easy
Graphic generation and Support	Best suited for Graphics , Reporting Applications	Can be used using Libraries, Need extra effort to learn	Very Hard to implement in C++	Tedious task
OOP feature and Exception handling	Fulfils totally	Fulfill	Fulfill	Fulfils
Error Elimination	Easy	Hard	Hard	Hard
File Stream and Database	Easily compatible with Excel and Java	Hard	Very Hard	Very Hard

Table 7: Language Suitability Table and comparison

4.2.1.1.2 Reasons for choosing Visual Basic .NET for Beacon

- **RAD:** Rapid application development tool, reduce time and efforts in application development. Within a matter of minutes a complete **Graphical User Interface (GUI)** can be produced; thus requiring less programming time and less design time for Beacon
- **Multithreading:** Visual Basic .NET applications can perform multiple tasks simultaneously using multithreading (or free threading), a process in which individual tasks execute on separate threads. Multithreading improves the performance and responsiveness for Beacon. Threading concepts are used when **executing various events** at same time
- **Globalization:** Using dozens of different objects in the **System.Globalization** namespace, developer can customized language difference such as fonts and right to left writing styles, calendars and different patterns for dates, currency , numbers etc. VB.NET is fully Unicode compliant and the advantage is that developer can display virtually all characters in all languages. **Language Change feature** use this concept in Beacon
- **User Control:** provide a means by which custom graphical interfaces can be created and reused. A user control is essentially a component with a visual representation. It can consists of one or more Windows Forms controls, components, or blocks of code that can extend functionality by validating user input, modifying display properties. It can be simply used in the same manner as other controls. **Bin Packing Graph** will be created using this concept
- **Project Wizard :** can be used as a powerful tool which can generates the code automatically, help a lot as less prone to typing error. Using wizard user do not help to enter code line manual but just have to configure the things within the project.
- **OOP Concepts:** Supports object oriented concepts and a class can inherit other class as a base class and can also define its own properties in derived class. **City, Cities ,Tour , Population class** uses inheritance and other oops features in Beacon
- Classes created in VB.NET have public access modifier by default and can be inherited by any other class within the same project.

4.2.1.1.3 Packages to be used

Packages	Justification
Imports System.Globalization	For Culture Related information includes language, country, region, calendars etc.
Imports System.Drawing.Drawing2D	The System.Drawing.Drawing2D namespace provides advanced two-dimensional and vector graphics functionality.
Imports System.Collections.Generic	contains interfaces and classes that define generic collections, which allow users to create strongly typed collections that provide better type safety and performance than non-generic strongly typed collections.
Imports System.Text	contains classes that represent ASCII and Unicode character encodings; abstract base classes for converting blocks of characters to and from blocks of bytes; and a helper class that manipulates and formats String objects without creating intermediate instances of String.
Imports System.ComponentModel	provides classes that are used to implement the run-time and design-time behaviour of components and controls.
Imports System.Data	provides access to classes that represent the ADO.NET architecture. ADO.NET lets you build components that efficiently manage data from multiple data sources.
Imports System.Threading	provides classes and interfaces that enable multithreaded programming
Imports System.IO	contains types that allow reading and writing to files and data streams, and types that provide basic file and directory support.
Imports WindowsApplication1.Beacon	Use classes within the Beacon itself e.g. BinPackingGraph , Usercontrols
Imports System.Windows.Forms	contains classes for creating Windows-based applications that take full advantage of the rich user interface features available in the Microsoft Windows operating system.
Imports System.Data.SqlClient	The System.Data.SqlClient namespace is the .NET Framework Data Provider for SQL Server.

Table 8 : Packages to be used

4.2.2 Methodology: Advanced Waterfall

4.2.2.1 Description

The Advanced waterfall provides an orderly sequence of development steps and helps ensure the adequacy of documentation and design reviews to ensure the quality, reliability, and maintainability of the developed software.

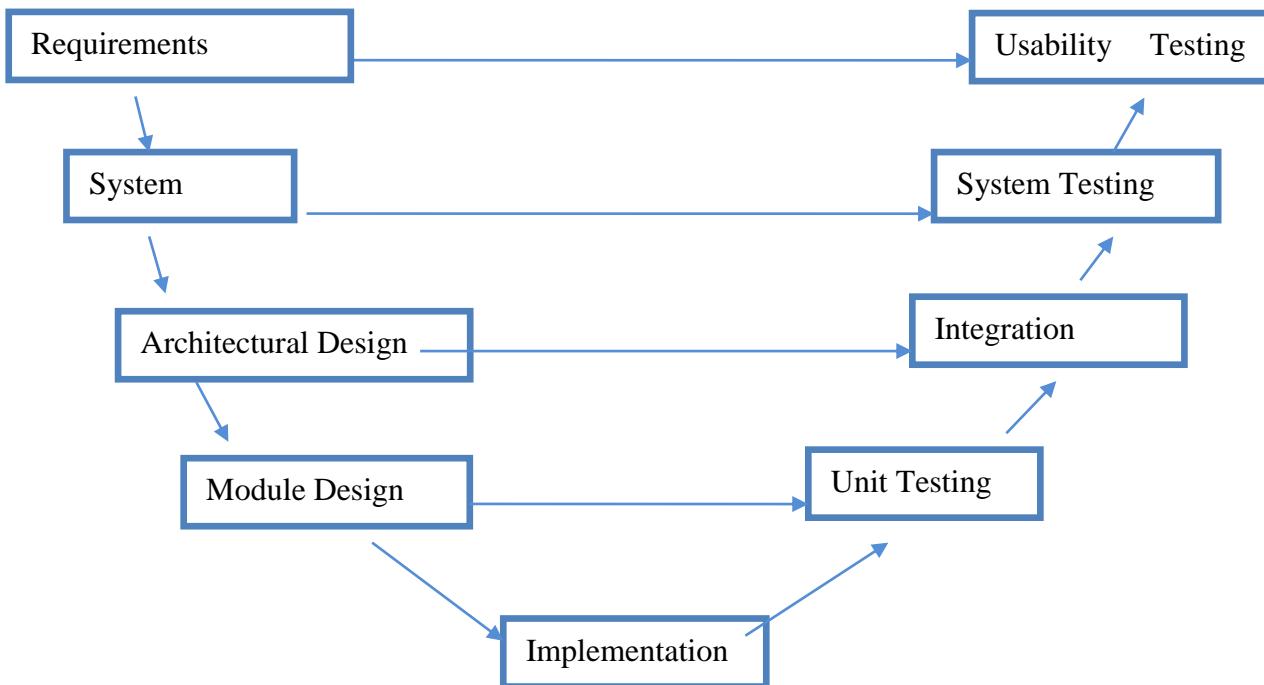


Figure 6 : Advanced Waterfall follows V model

Advanced waterfall methodology follows V model in development stages.

4.2.2.2 Justification

After some deliberation and discussion the developer settled upon using Waterfall Model for the development of this system. The most tempting factor for selection of Waterfall model is-

- **Stable project requirements:** As in our project most of the user requirements are freeze at the time of PSF so it indicates a stable project requirements and Waterfall methodology completely supports a project which has requirements decided in advance.

- **Progress of system is measurable:** After each step it produces the documentation and as the structure of our Final year project we need to submit the documentation after each phase so it will be best suitable.
- **Strict sign-off requirements:** As the developers goal will be to satisfy the user and until the user will be satisfied the developer will be providing the user desired functionalities and proper features so this methodology will be best suitable.
- The emphasis on requirements and design before writing a single line of code ensures minimal wastage of time and effort and reduces the risk of schedule slippage, or of customer expectations not being met.
- In modified waterfall model life cycle phases are permitted to overlap. Because of the phases overlap, a lot of flexibility has been introduced in the modified waterfall model in software engineering. At the same time, a number of tasks can function concurrently, which ensures that the defects in the software are removed in the development stage itself and the overhead cost of making changes to the software before implementation is saved.
- Making changes to the basic design is also possible, as there are a number of phases active at one point of time. In case there are any errors introduced because of the changes made, rectifying them is also easy (Testing can be done). This helps to reduce any oversight issues. The modified waterfall model diagram does not differ from the traditional waterfall model diagram, as to every phase of the model verification and validation step has been added. (Satalkar, 2010)

4.2.3 Algorithms: Why Particular Algorithm?

4.2.3.1 Reasons for choosing Genetic Algorithm for Route Planning

- TSP is known as the NP-Complete problem means it cannot be solved in polynomial time, so we need to use an approximation solution that fits right in. Genetic Algorithm is the successful way to approach to get optimized results.
- Testing every possibility for an N city tour would be $N!$ math additions. A 30 city tour would have to measure the total distance of be $2.65 * 10^{32}$ different tours. Assuming a trillion additions per second, this would take 252,333,390,232,297 years. Adding one more city would cause the time to increase by a factor of 31. Obviously, this is an impossible solution.

- Genetic Algorithms mimic nature and evolution using the principles of Survival of the Fittest. Although it might not find the best solution, it can find a near perfect solution for a 100 city tour in less than a minute.
- Greedy approach can be used in between the Genetic algorithm to map closest cities in the initial population generation

4.2.3.2 Reasons for choosing Bin Packing Algorithm for Load Planner (CLP)

- 2D-Bin-Packing works on pack a given set of 2D-rectangles into unit square bins so that the number of bins is minimised. Even very simple cases of these problem are known to be NP-hard, and hence, it is very likely that no efficient algorithms for them exist
- In the design of such algorithms, a simple shelf technique is used: order the rectangles according to a sorting rule like decreasing width, increasing height, etc., and then greedily pack them one by one in this order over package shelves according to some rule, like **First-Fit**, **Next-Fit**, **Best-Fit**, **Worst-Fit** and so on. This allows a simple code design, a very fast running time, and a relatively good quality guarantee.
- Beacon has an assumption that all the **container are of same width**, only the height of containers gets changed, that fixes the width of the rectangle and makes work easier enough in one dimension.

Tall Containers	Container used
	50x400
	50x300
	50x200
	50x100
	50x50

Figure 7: Bin Packing Example (Gordon, 2011)

4.3 Academic Research

There are varied areas in which research has to be accomplished in order to derive some deductions during the development of the proposed system. This system is going to involve all three kinds of research namely, Primary, Secondary and Academic research. Following areas have been shortlisted which needs to researched:

- ✚ Key Concepts Genetic Algorithms and Greedy approach
- ✚ Concepts of Implementation of Graph theory with data structure
- ✚ Bin Packing algorithm and Greedy approach
- ✚ Database concepts
- ✚ Visual Basic / .NET
- ✚ Software Methodology and Software Engineering
- ✚ Human Computer Interaction Principles
- ✚ Design Pattern

Books:

Algorithms:

1. Clifford Stein, Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest ,2010, *Introduction to Algorithms* ; Massachusetts USA: PHI Learning.
2. David E. Goldberg ,1989, *Genetic Algorithms in search, Optimization & Machine Learning*, 1/e; USA ; Pearson Education
3. Steven Holzner ,2009 *Visual Basic .NET Programming* ;USA; Paraglyph Press

Software Methodology and Software Engineering

1. Kendall and Kendall (2005); *System Analysis and Design*; 4th Ed; New York: Prentice Hall.
2. Jeffry L Whittem, Lonnie Bentley (2005). *System Analysis and Design*; London: Mc Graw-Hill.
3. Shelly Cashman (2002). *System Analysis and design*; 2nd Ed. Sydney: Shelly Cashman Series.
4. Shari Lawerence Pfleeger (2002). *Software Engineering*; 2nd Ed; New Jersy: Pearson Education.

Chapter 5: Research Analysis

5.1 Questionnaire Analysis

Refer to appendices for analysis of questionnaire.

5.1.1 Questionnaire Conclusion

After analysis of questions, the developer concluded on some points which are following –

Most of the user's in India are unaware of the situation and complexity, although they are affected. They have no knowledge of the software and its benefits in terms of time and cost cutting.

The developer has decided to provide metaphors for increasing learning ability. Also the application has scope for lots of design principles to increase usability and interaction.

For data entry purpose, developer is not using any advanced database system but a simple excel file or csv file. Also users will be allowed to enter data directly into the application and defining the parameters (e.g. specification of boxes).

Time complexity is the crucial factor in terms of speed, Most of the companies have more than 100 consignments per day, and so without using an advanced approach of solving, the system would be very slow and result into no benefit.

5.1.2 Recommendation

Graphical tour generation is motivated by user to showcase the real working of algorithms. Also user can stop running algorithms at any moment.

Since the co-ordinates required for working is very small in size, XML file is recommended to store, edit and change the values of co-ordinates of cities

City Entry (XML file entry) module required to create co-ordinates so that it can be further used in the algorithms.

Time is crucial factor to solve such big problem, so only the best approach should be applied (like genetic algorithm) to reduce time complexity.

5.2 Interview Analysis

Refer to appendices for analysis of Interview.

5.2.1 Interview Conclusion

After analysis of Interview, the researcher conclude various points from other interviewee,

Travelling Salesman Problem and Container Loading problem are known as the NP-hard problems and no exact solution is invented yet for this problem. Although real world algorithms like Genetic algorithm, Ant Colony Algorithm, Simulated Annealing are suggested for solving this problem to most optimized state.

Major parameters used in the TSP are the cost calculation, crossover, tournament and mutation factor. Tours are the arrays of cities in a particular order and initial population needs to be created to crossover between them.

5.2.2 Recommendation

Initial Parameters used for solving TSP are population Size, mutation %, Random seeds, No of close cities, odd %.

Validation should be provided to check the links between the routes.

Bin Packing Algorithm is suited to the CLP, Next-Fit, First-Fit, Worst-Fit and Best Fit is some loading plans that can be used to get the optimized results.

Both TSP and CLP are NP –complete problems so only the optimized solution can be achieved, no solution for bigger values can be guaranteed as perfect.

5.3 Traceability Matrix for Questionnaire and Functionality Mapping

RI	FC 1	FC 2	FC 3	FC 4	FC 5	FE 1	FE 2	FE 3	FS 1	FS 2
<u>Questionnaires</u>										
Q1.6					X					
Q1.7		X		X		X				
Q1.8	X									
Q2.6							X			
Q2.7								X		
Q2.8								X		
<u>Interview</u>										
I1		X		X						
I2		X		X						
I4			X		X					
I6									X	
I7										X

Table 9 : Traceability Matrix between Questionnaire, Interviews and Functionalities

Chapter 6: System Design

The aim of design is to produce a model that will provide a seamless transition to the coding phase, i.e. once the requirements are analysed and found to be satisfactory, a design model is created which can be easily implemented.

Selected Methodology: Object-oriented design

In the object-oriented design approach, the system is viewed as collection of objects (i.e. entities). The state is decentralized among the objects and each object manages its own state information. For example, in this project Objects have their own internal data which define their state. Similar objects constitute a class. In other words, each object is a member of some class. Classes may inherit features from super class. Conceptually, objects communicate by message passing.

Why object-oriented design approach?

Unlike function-oriented design methods, in OOD, the basic abstraction is not real-world functions such as sort, display, track, etc., but real-world entities such as cities, tour, population etc.

Function-oriented techniques such as SA/SD group functions together if, as a group, they constitute a higher-level function. On the other hand, object-oriented techniques group functions together on the basis of the data they operate on.

Justification of using these diagrams (object-oriented design Methodology):

Use Cases: Use cases will show who will use the system and what they will do with it. This will specify the system requirements in the context of the user.

Activity Diagrams: Activity diagrams will show the flow of work and information between activities performed by users and system or its parts. They elaborate the use cases.

Sequence Diagrams: Sequence diagrams will show the sequence of interactions between users and system or its parts. This will help in realizing the structural organization of objects that communicate within a system. (**Kennesaw 2001**)

Class Diagrams: UML class diagrams will be used to provide an implementation-independent description of the objects that are used in the system and passed between its components. The design then can be implemented on any platform using any development environment.

Traceability Matrix between Analysis and Design

Requirement Analysis	Design		
	Use Case	Activity Diagram	Sequence Diagram
Design Route Map	UC-01	AD-01	SD-01
Plan Route	UC-02	AD-02	SD-02
XML Connection	UC-03	AD-03	SD-03
Manage Loading	UC-04	AD-04	
View Loading	UC-05	AD-05	SD-04
Change Language	UC-06	AD-06	SD-05
Lock			
Image Download	UC-03		
Save/ View	UC-02	AD-02	

Table 10 : Traceability Matrix between Requirement Analysis and design

6.1 Use Cases

With the help of use case diagrams, the developer wishes to discuss and communicate:

- The scenarios in which Beacon will function as a product.
- The goals that it helps those actors achieve.
- The scope/limitations of Beacon.

The developer's approach:

For the benefit of the reader's the developer has prepared a high-level use-case context diagram with all specifications. After that for each context a full formed use-case is prepared along with any **assumptions, pre-conditions, post-conditions, happy-path, alternate-paths (if any), and exception-pathways**.

Along with the use cases activity diagrams and sequence diagrams are also provided wherever relevant and deemed necessary.

6.1.1 Use Case Context Diagram

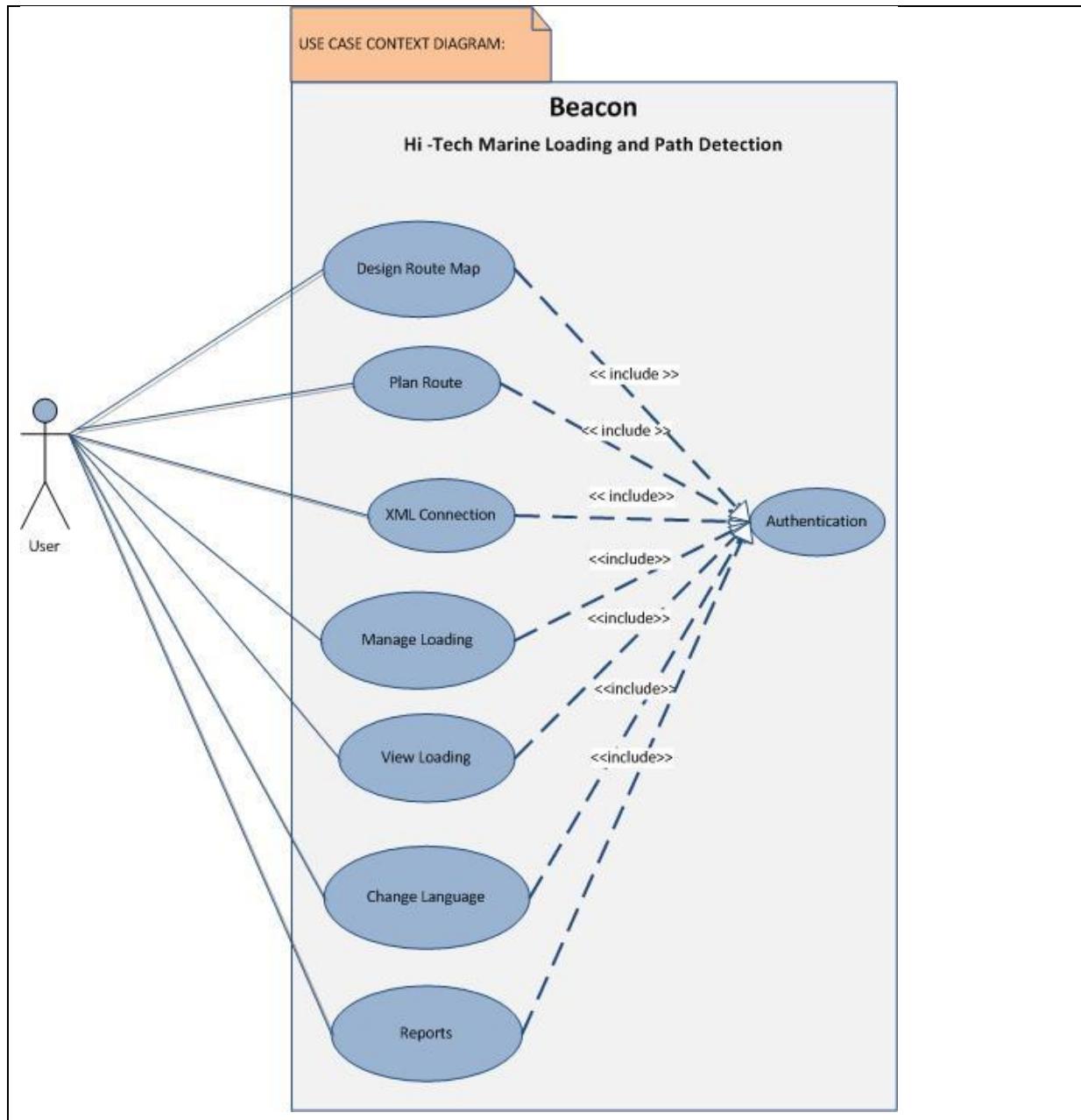


Figure 8 : Use Case Context Diagram

6.1.1.1 Actor List

Actor	Description
User	Any user who wants to use Beacon for finding shortest path, container loading or Consignment management.

6.1.1.2 Use – Case List

ID	Use Case Name	Description
UC-01	Design Route Map	Designer for the route map
UC-02	Plan Route	Algorithms behind the route and customized output
UC-03	XML-Connection	Read /Write to XML file
UC-04	Manage Loading	Get elements and helps in choosing best algorithm
UC-05	View Loading	View Loading plan in the Graph form
UC-06	Change Language	Language Changer
UC-07	Reports	Previously Saved Solution Reports

Table 11 : Use Case List

6.1.2 Use Case Specification (Design Route Map)

Name	Design Route Map	Use Case	ID: UC-01
Description:	Use Case initiates when the user want to design the route in the route map		
Goal:	To be able to create list of cities		
Actor(s):	User		
Assumption(s):	User have the access and knowledge of the application		
Pre-condition(s):	The User has unlocked the application before using		
Post-condition(s):	User can design on picture box/ or import co-ordinates through xml		
Primary (Happy) Path:			
The User select create cities			
<ol style="list-style-type: none"> 1. User can design cities on the map 2. User can import co-ordinates of cities into the map 3. Cities created must be valid entry 4. Further user can clear out the cities from map 5. Use show graph to view axis. 			
Alternate Pathway(s):			
None at the point			
Exception Pathway(s):			
<ol style="list-style-type: none"> 1. User select xml file but null entry is taken. 2. Cities in xml file are out of scope of the map area. 			

Table 12: UC-01 Use Case Specification (Design Route Map)

6.1.3 UC-01 Use Case Diagram (Design Route Map)

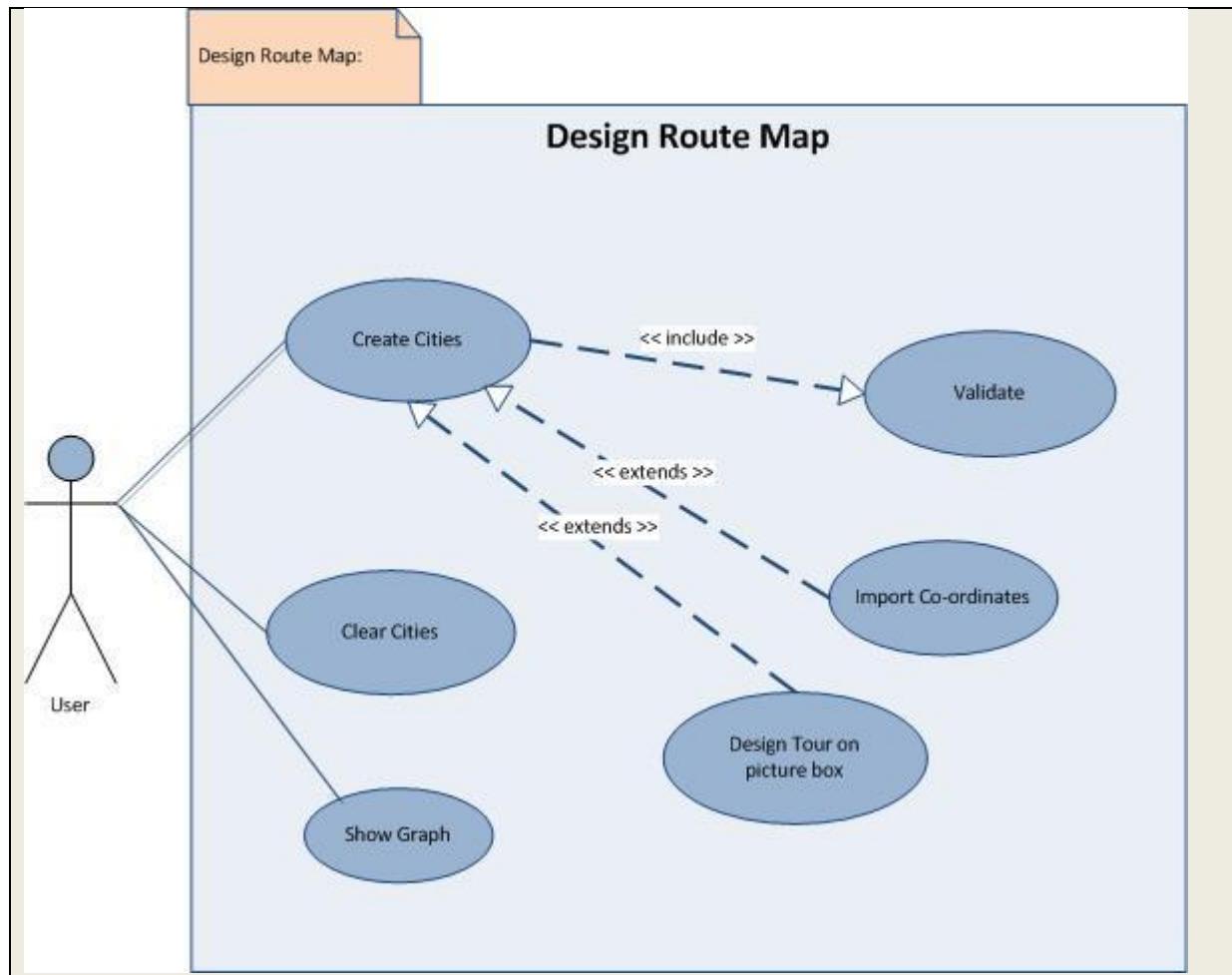


Figure 9: UC-01 Use Case Diagram (Design Route Map)

6.1.4 AD-01 Activity Diagram

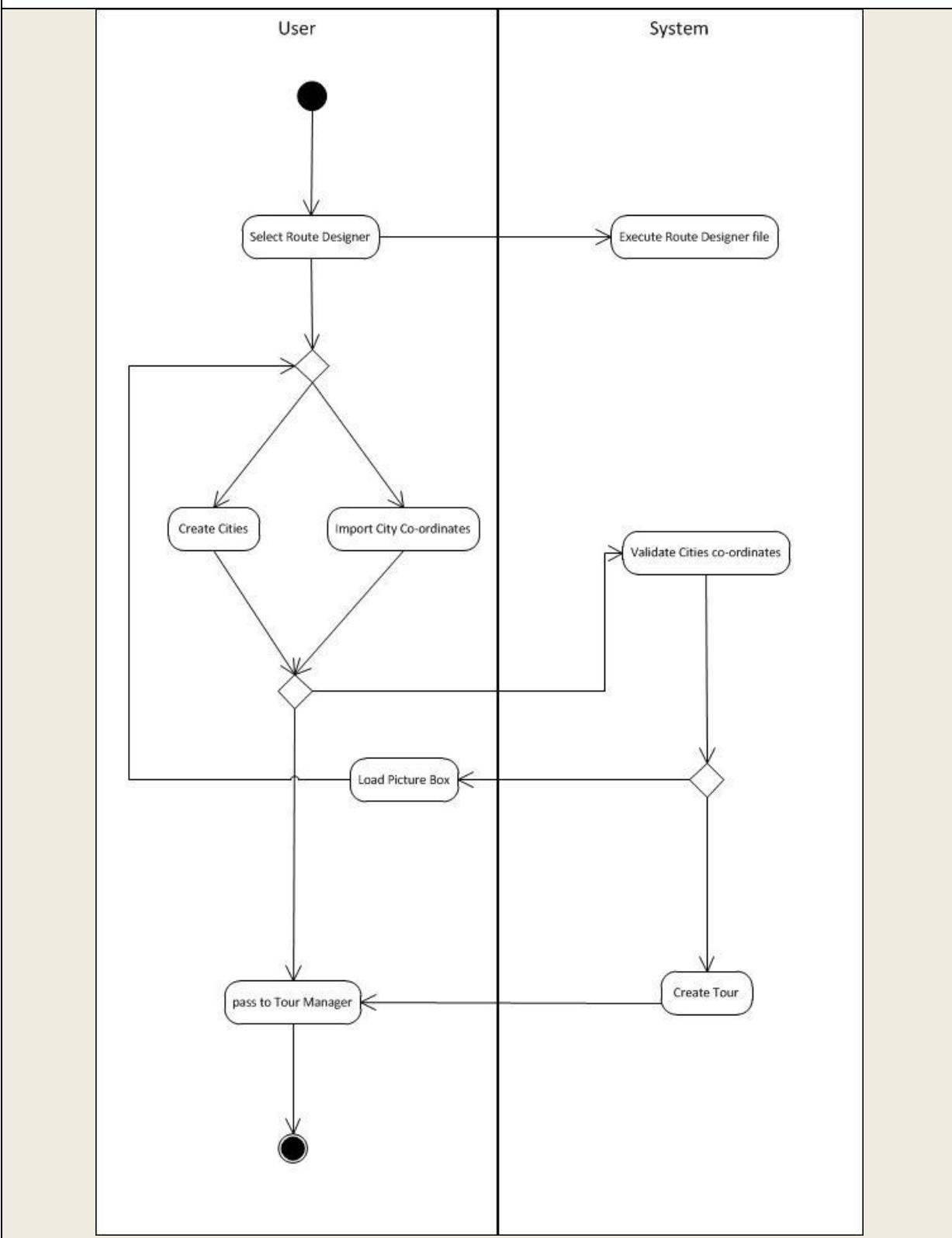


Figure 10: AD-01 Activity Diagram (Design Route Map)

6.1.5 SD-01 Sequence Diagram (Design Route Map)

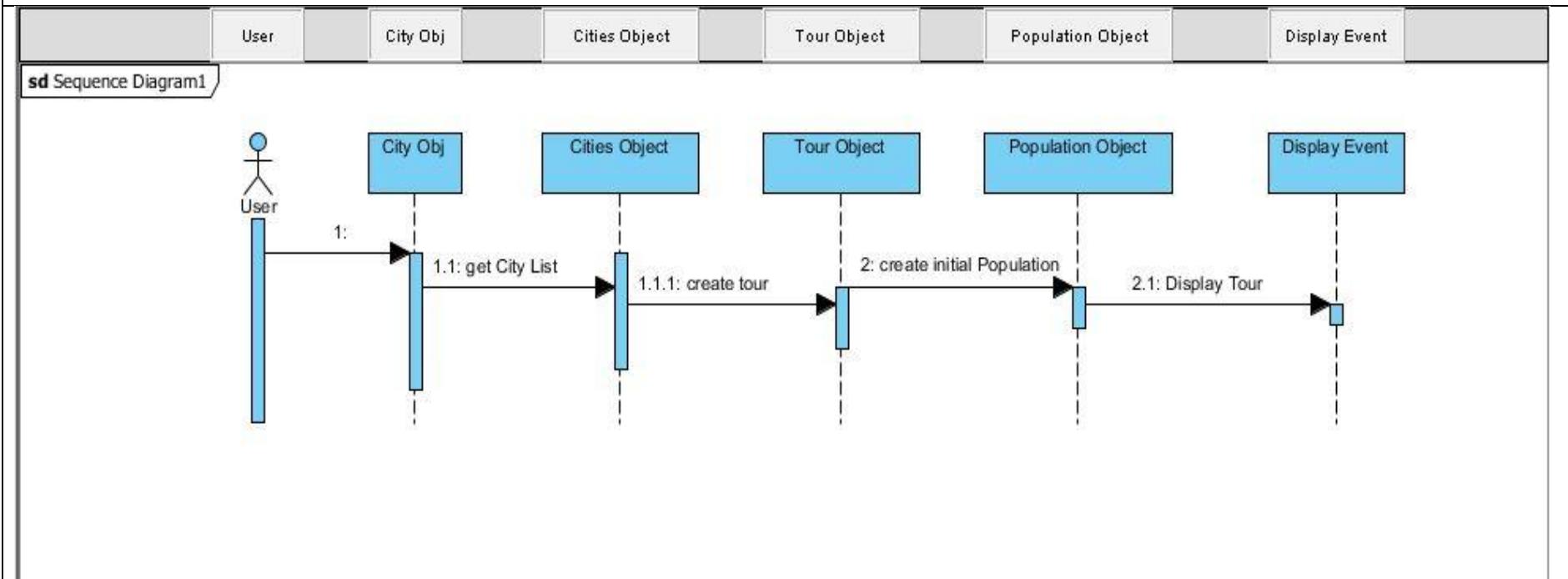


Figure 11 : SD-01: Sequence Diagram (Design Route Map)

6.1.6 UC-02 Use Case (Plan Route)

Name Plan Route Use Case		ID : UC-02		
Description:	Use case is applied when the Design route Use Case is successfully done. It is responsible to start algorithm and play with it.			
Goal:	To use the parameter in the best way to determine best path			
Actor(s):	User			
Assumption(s):	Default values of initial constraint are not null. Algorithm will not stop by own.			
Pre-condition(s):	User have the access of lock and knowledge of the TspForm User has already the cities list in the array as well as in the map.			
Post-condition(s):	Get the tours which have the best fitness			
Primary (Happy) Path:				
The User select Edit Constraints and edit initial parameters like				
<ol style="list-style-type: none"> 1. Population Size, Mutation %, Group Size, Max generation, No-of close cities, Close cities and random seeds. 2. Start button initiates the algorithm and the process start running. 3. Algorithm search for best fitness tour and display it on map. 4. Distance and number of iteration are displayed. 5. Running algorithm can be stopped at any time. 6. After stop, user can edit cities and start algorithm all again. 				
Alternate Pathway(s):				
The Running Algorithm can be stopped at any point and initial parameter and cities list can be updated according to needs.				
<ol style="list-style-type: none"> 1. After rerunning the algorithm, the algorithm takes all the cities from beginning. 2. Xml file and manual city entry can be done. 				
Exception Pathway(s):				
<ol style="list-style-type: none"> 1. Less than 3 cities cannot come in the process the throw error. 2. New Cities and initial parameter entered are wrong through and error message 				

Table 13: UC-02 Use Case Diagram (Plan Route)

6.1.7 UC-02 Plan Route Map Use Case Diagram

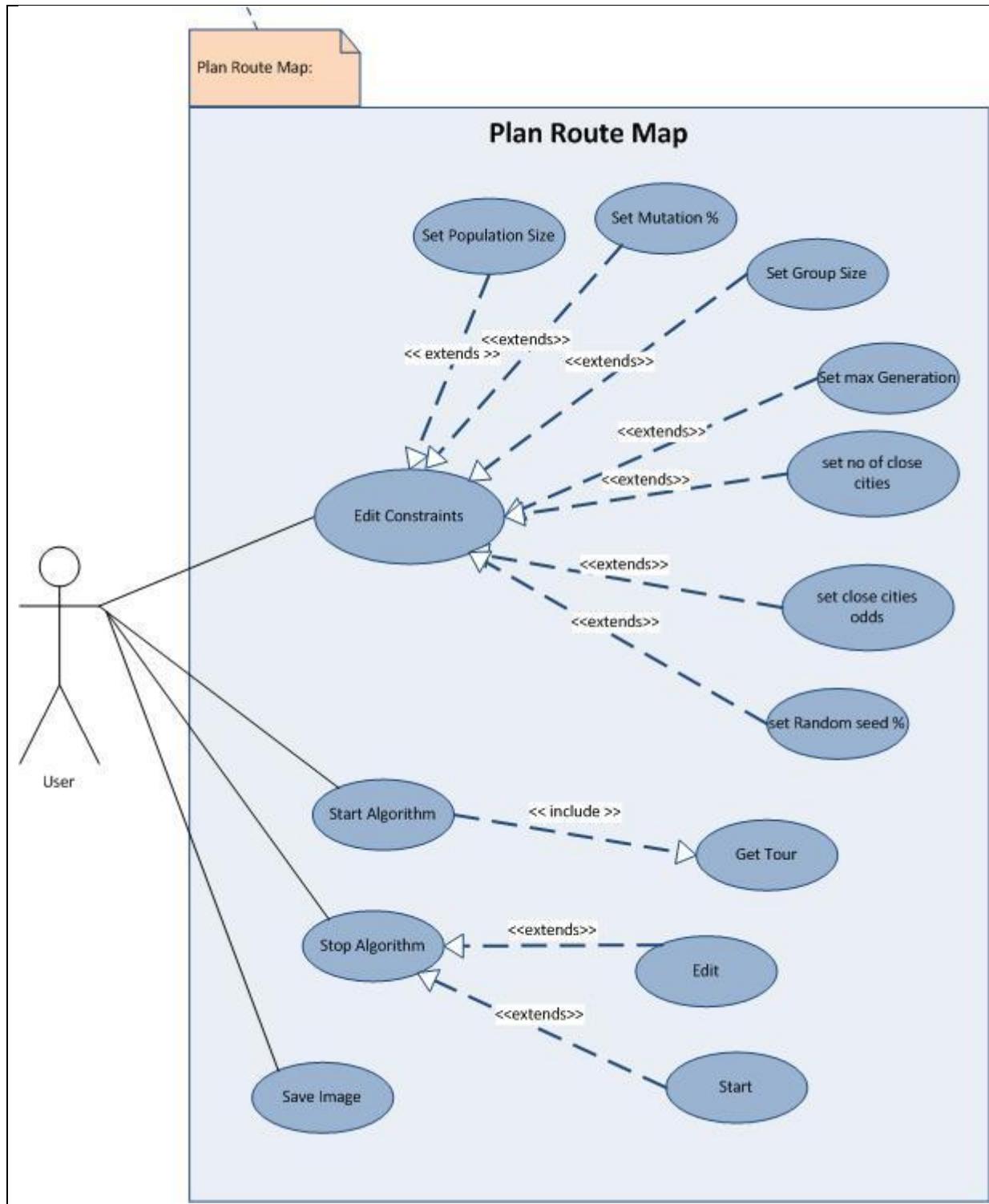


Figure 12: UC-02 : Use Case Diagram (Plan Route)

6.1.8 AD-02 Activity Diagram (Plan Route)

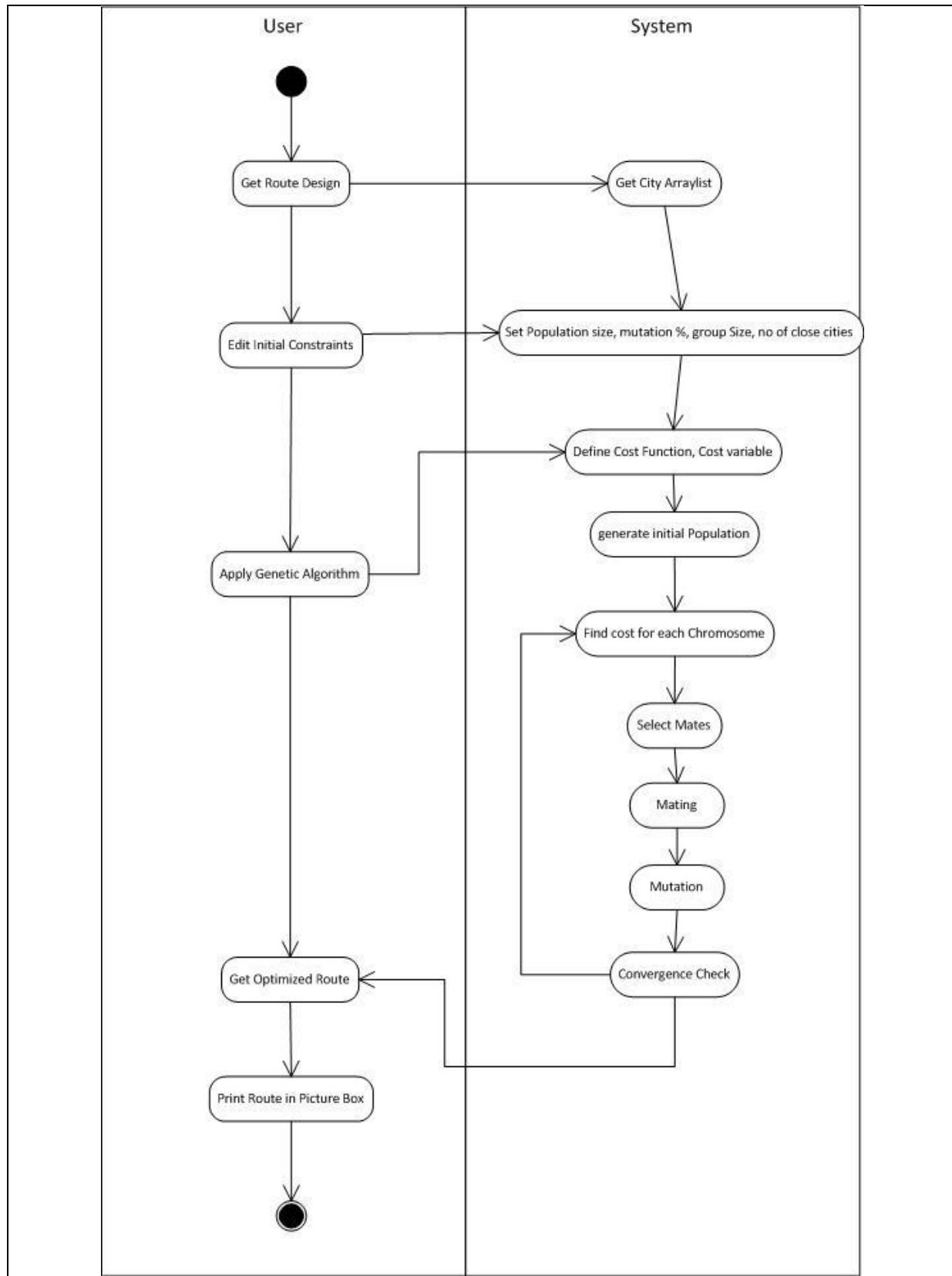


Figure 13: AD-02 Activity Diagram (Plan Route)

6.1.9 SD-02 Sequence Diagram (Plan Route)

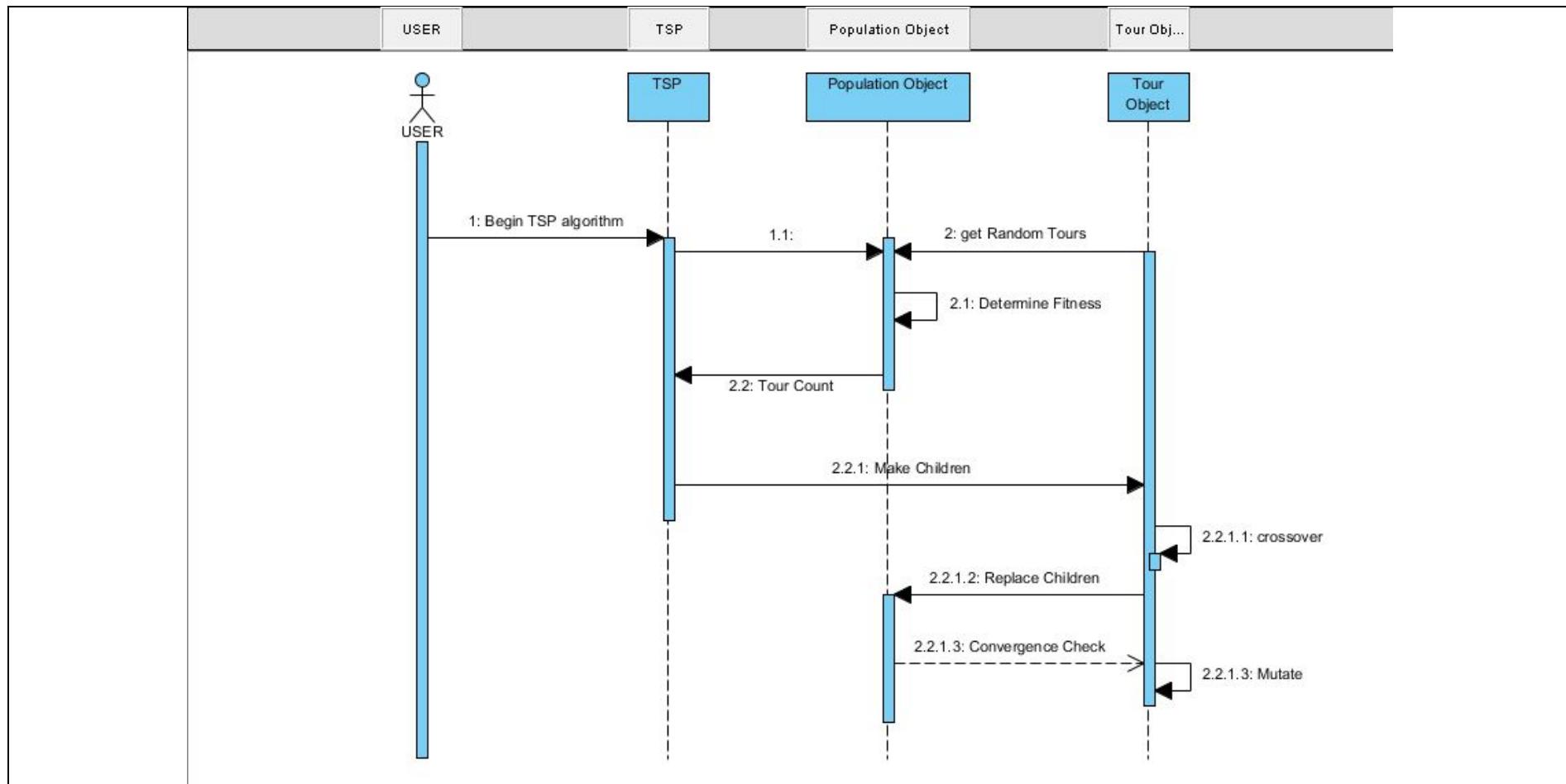


Figure 14: SD-02 Sequence Diagram (Plan Route)

6.1.10 UC-03 Use Case Specification (XML Editor)

Name XML Editor Use Case		ID : UC -03
Description:	Use case initiates when user chooses to read/write from xml file.	
Goal:	To use the Xml connection to read cities for the map.	
Actor(s):	User	
Assumption(s):	Xml file should be valid	
Pre-condition(s):	User have the access of lock and knowledge of the TspForm	
Post-condition(s):	Cities list are updated by reading	
Primary (Happy) Path:	<ol style="list-style-type: none"> 1. User browse the xml file , Load City List , show in map 2. Write co-ordinates to xml file from map 	
Alternate Pathway(s):	<ol style="list-style-type: none"> 1. None at the point 	
Exception Pathway(s):	<ol style="list-style-type: none"> 1. Wrong / Invalid File selection 	

Table 14: UC-03: Use Case Specification (xml editor)

6.1.11 UC-03 Use Case Diagram (XML Editor)

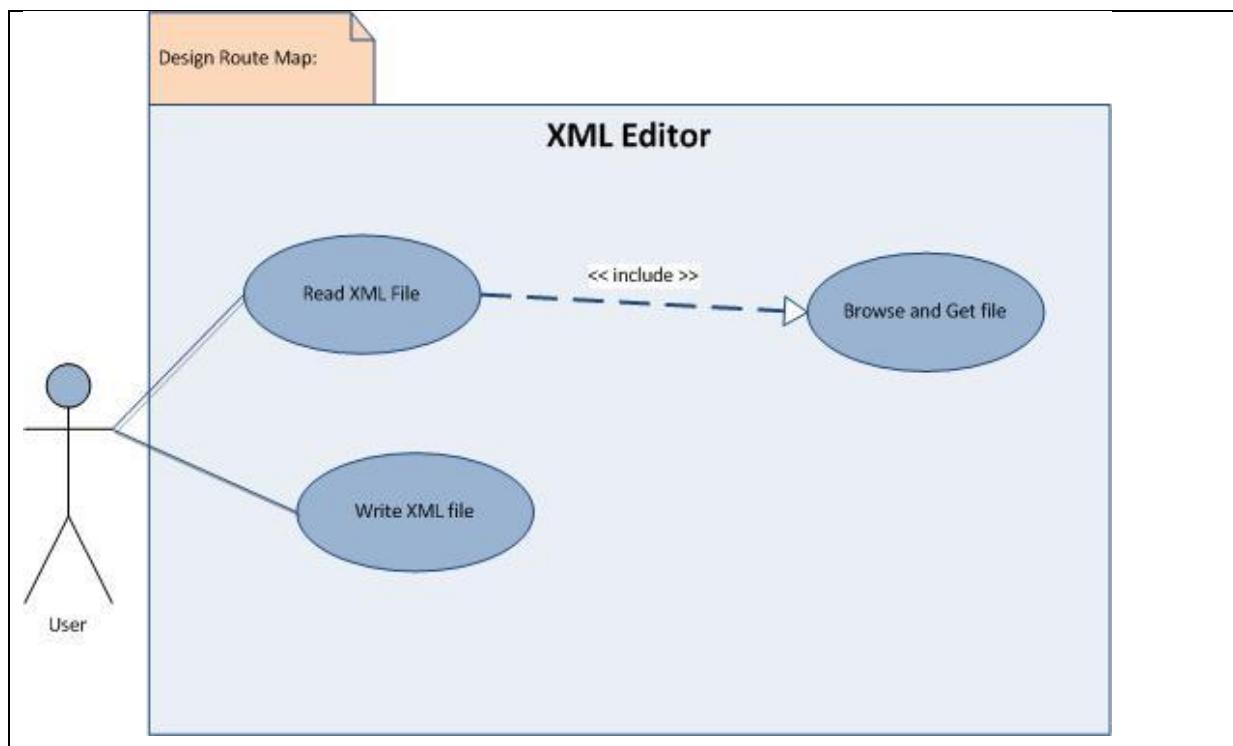


Figure 15: UC-03 Use Case Diagram (XML Editor)

6.1.12 AD-03 Activity Diagram (XML Editor)

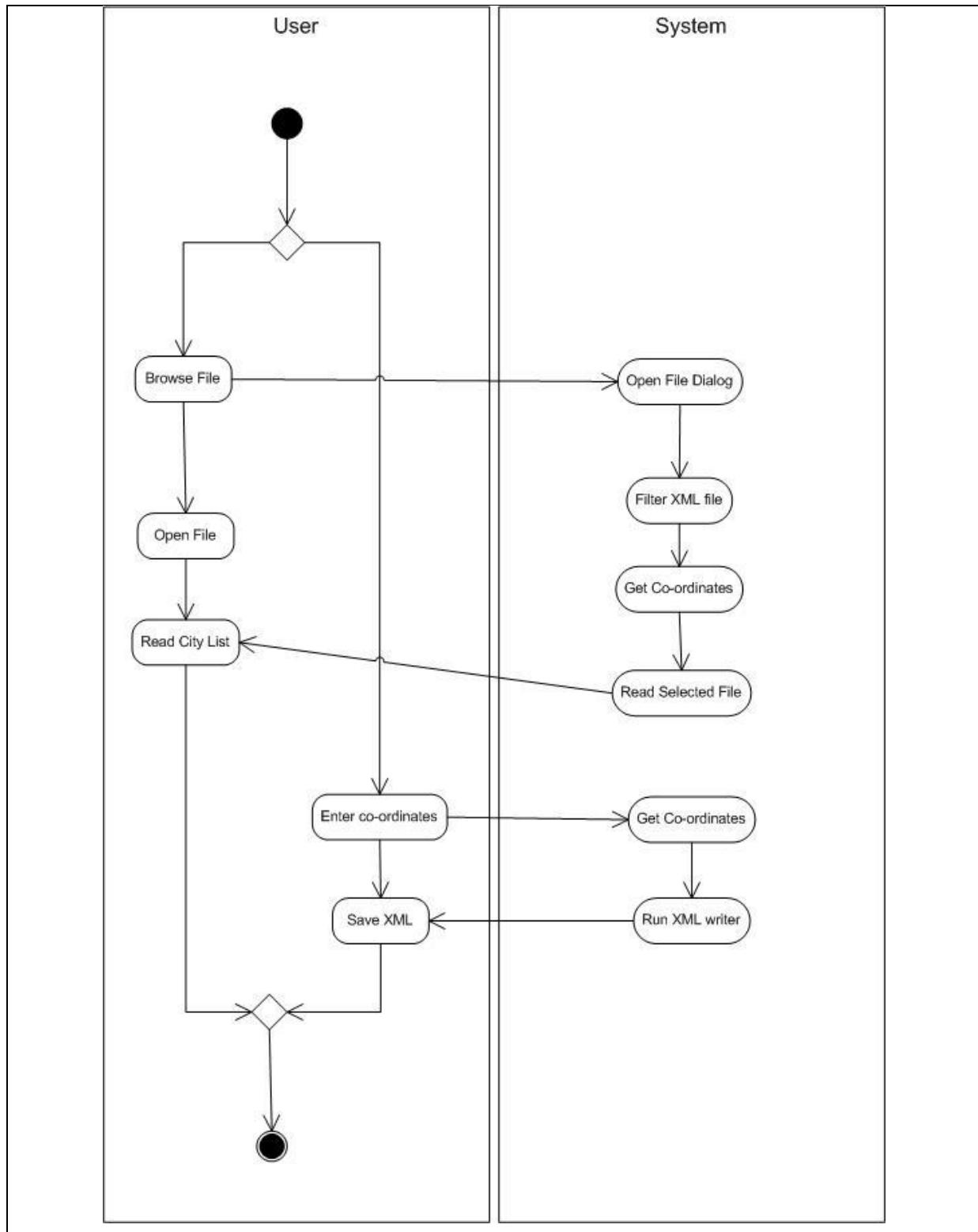


Figure 16:AD-03 Activity Diagram(xml editor)

6.1.13 SD-03 Sequence Diagram (xml editor)

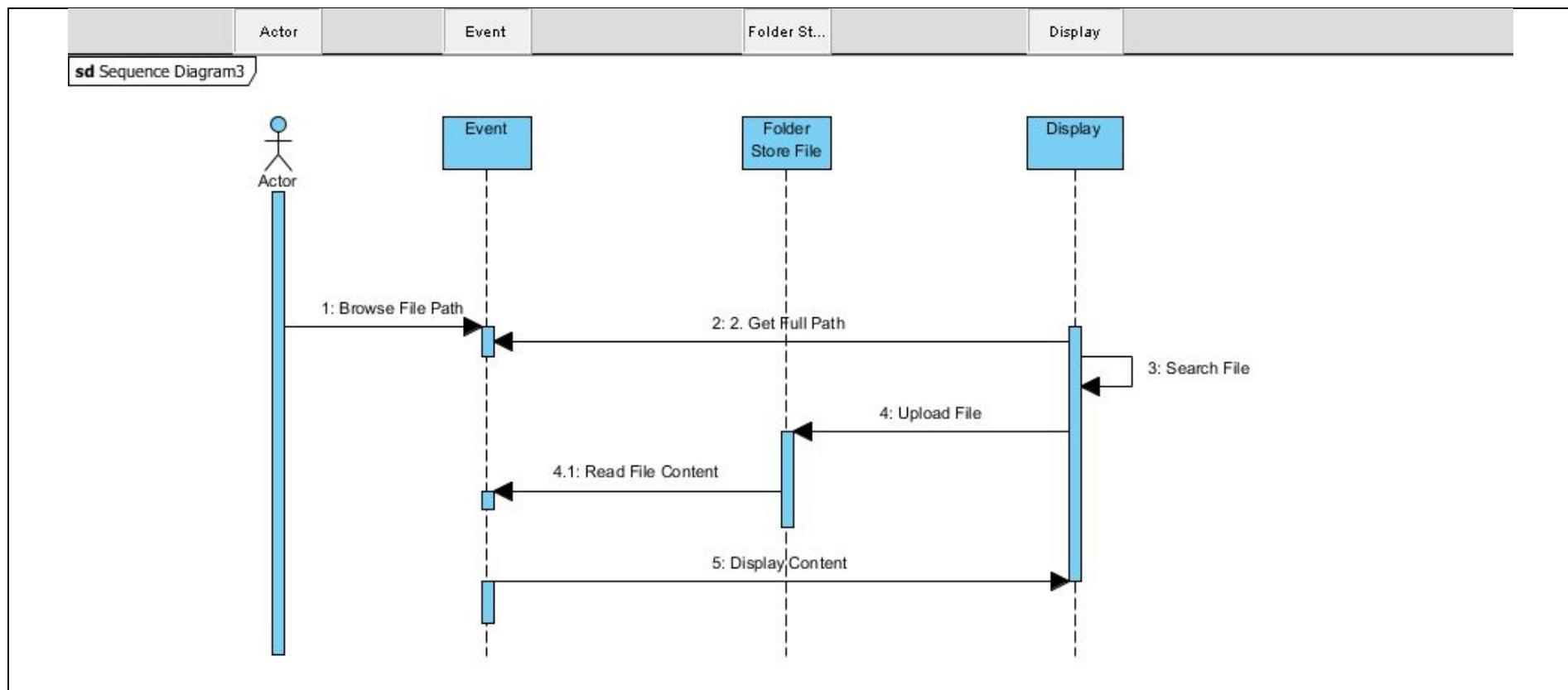


Figure 17: SD-03 Sequence Diagram (xml editor)

6.1.14 UC-04 Use Case Specification (Manage Loading)

Name Manage loading Use Case		ID: UC-04
Description:	Use Case initiates in the ClpForm to solve loading problem	
Goal:	To manage the loading problem by setting elements, algorithm, Ship length and order of loading.	
Actor(s):	User	
Assumption(s):	<ul style="list-style-type: none"> 1. ClpForm should be executed 2. Elements are not null. 3. Default Algorithm is Next-Fit Algorithm. 4. Default Ship Length is 80 ft 5. All containers have same length. 	
Pre-condition(s):	User have the access of lock and knowledge of the ClpForm	
Post-condition(s):	<ul style="list-style-type: none"> 1. Ordering of container comes in the bin array in order 2. Initiates View loading Use Case 	
Primary (Happy) Path:	<p>In ClpForm, user choose the elements text box</p> <ol style="list-style-type: none"> 1. Update the container length in the element multi text field. 2. Choose the Ship Length. 3. Select the Algorithm for fitting of containers. 4. Choose the order. 5. Initiates View Loading. 	
Alternate Pathway(s):	None at the point	
Exception Pathway(s):	<p>User choose the elements text box</p> <ol style="list-style-type: none"> 1. Container length are not entered in the proper format with space in between 2. Ship length is increased to a very high value. 3. Containers lengths are of very high (infeasible) length. 	

Table 15: UC-04 Use Case Specification (Manage Loading)

6.1.15 UC-04 Use Case Diagram (Manage Loading)

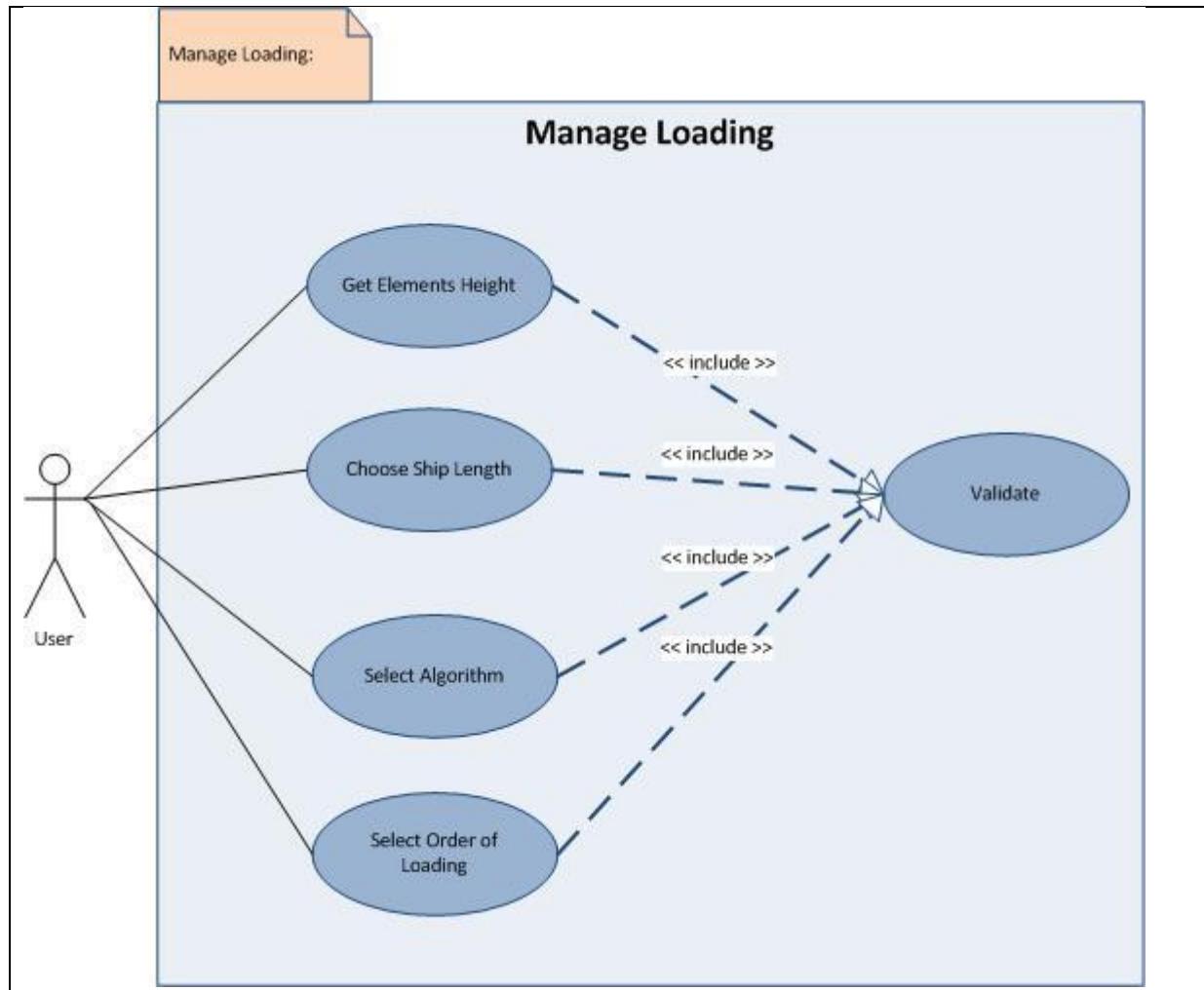


Figure 18: UC-04 Use Case Diagram (Manage Loading)

6.1.16 AD-04 Activity Diagram (Manage Loading)

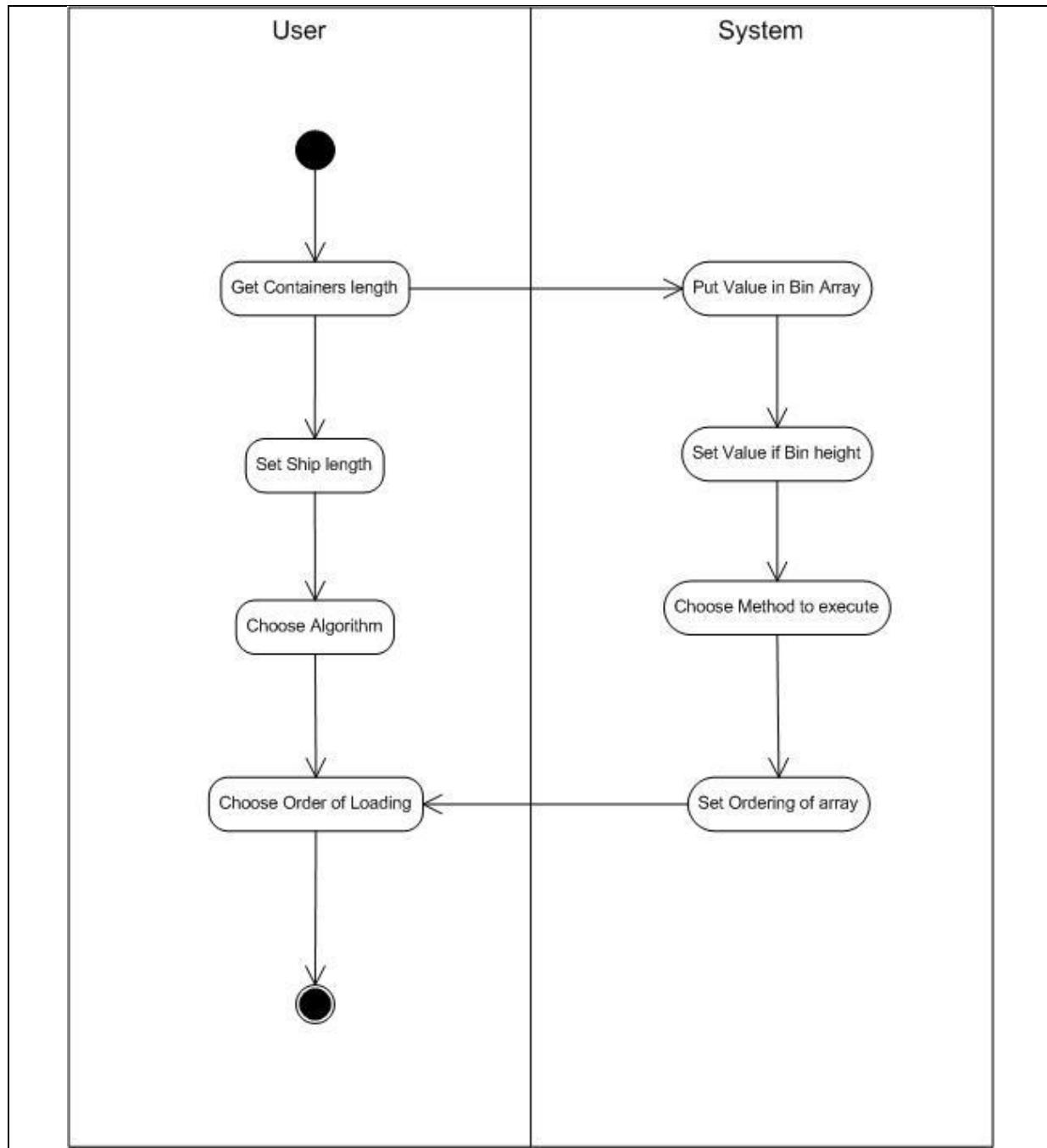


Figure 19: AD-04: Activity Diagram (Manage Loading)

6.1.17 UC-05 Use Case Specification (View Loading)

Name : View Loading Use Case		ID: UC-05		
Description:	Use Case initiates automatically when manage loading is done			
Goal:	To View the loading Plan of container and reflect changes in the loading in every change of elements, algorithm chosen, ship height and order of loading.			
Actor(s):	User			
Assumption(s):	Every change refreshes the picture box and update graph. Loading Graph initiates with default initialization. Height of the Graph is divided into percentage scale.			
Pre-condition(s):	1. User have the access of lock and knowledge of the ClpForm 2. Elements, Ship length ,order and algorithm are selected			
Post-condition(s):	1. Elements can be updated at any time 2. Ship length is divided into four parts 0%, 25%, 50%, 75%, 100%.			
Primary (Happy) Path:				
User loads the ClpForm 1. Default Loading Graph is shown. 2. Changes reflect with every change in elements, ship length, algorithm and order				
Alternate Pathway(s):				
None at this point				
Exception Pathway(s):				
User load the ClpForm 1. Enter wrong /invalid elements length.				

Table 16: UC-05 Use Case Specification (View Loading)

6.1.18 UC-05 Use Case Diagram (View Loading)

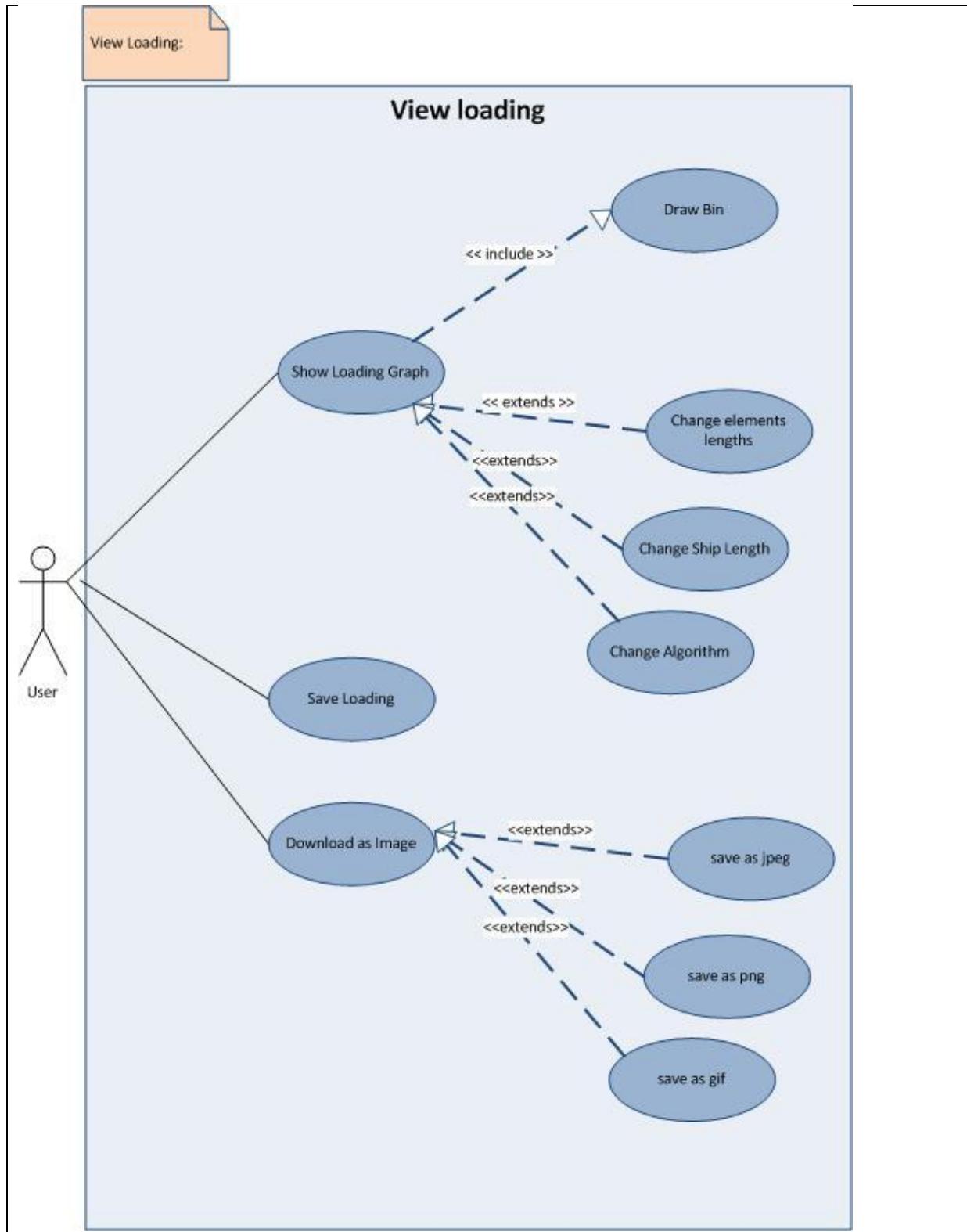


Figure 20: UC-05 Use Case Diagram (View Loading)

6.1.19 AD-05 Activity Diagram (View Loading)

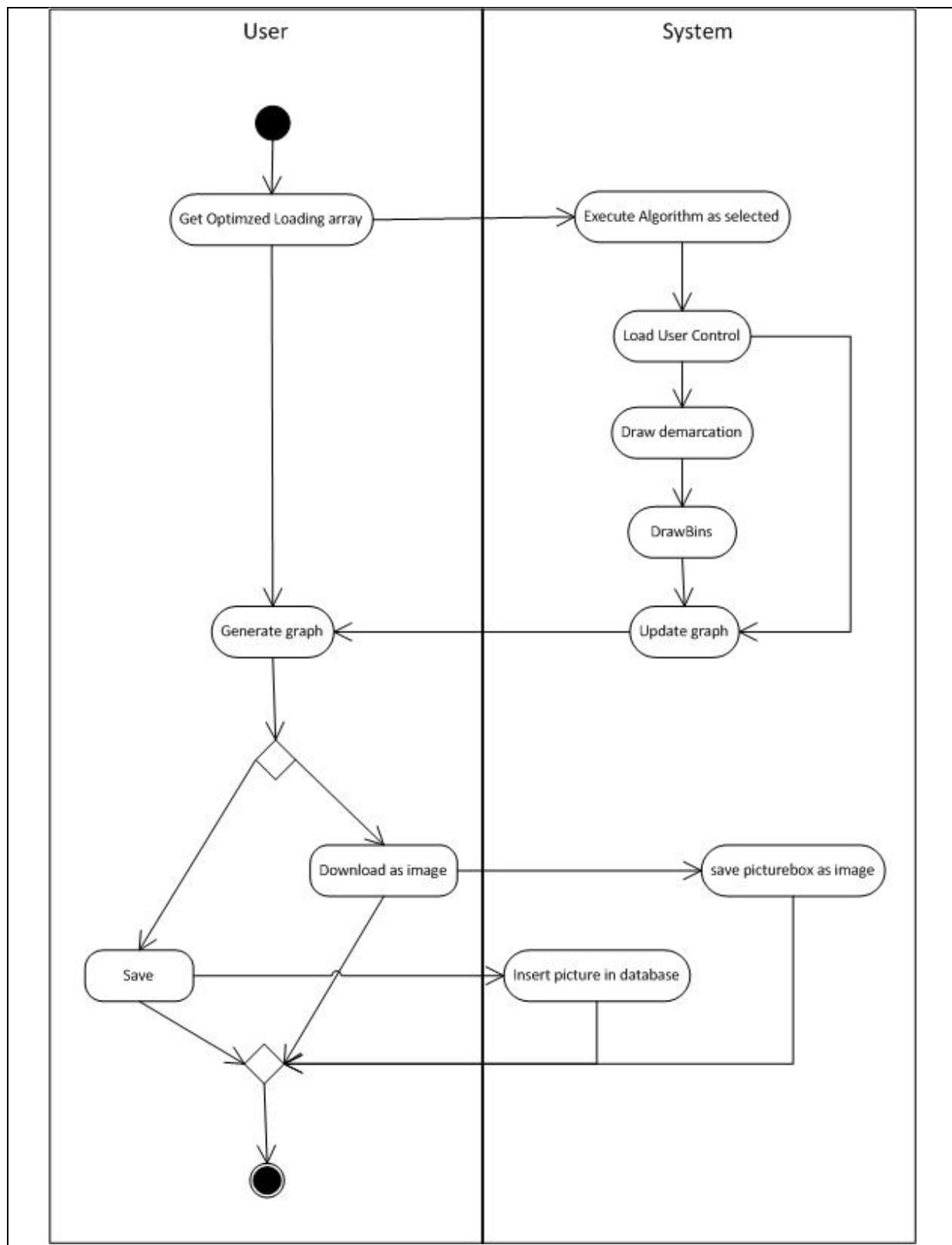


Figure 21: AD-05 Activity Diagram (View Loading)

6.1.20 SD-04 Sequence Diagram (View Loading)

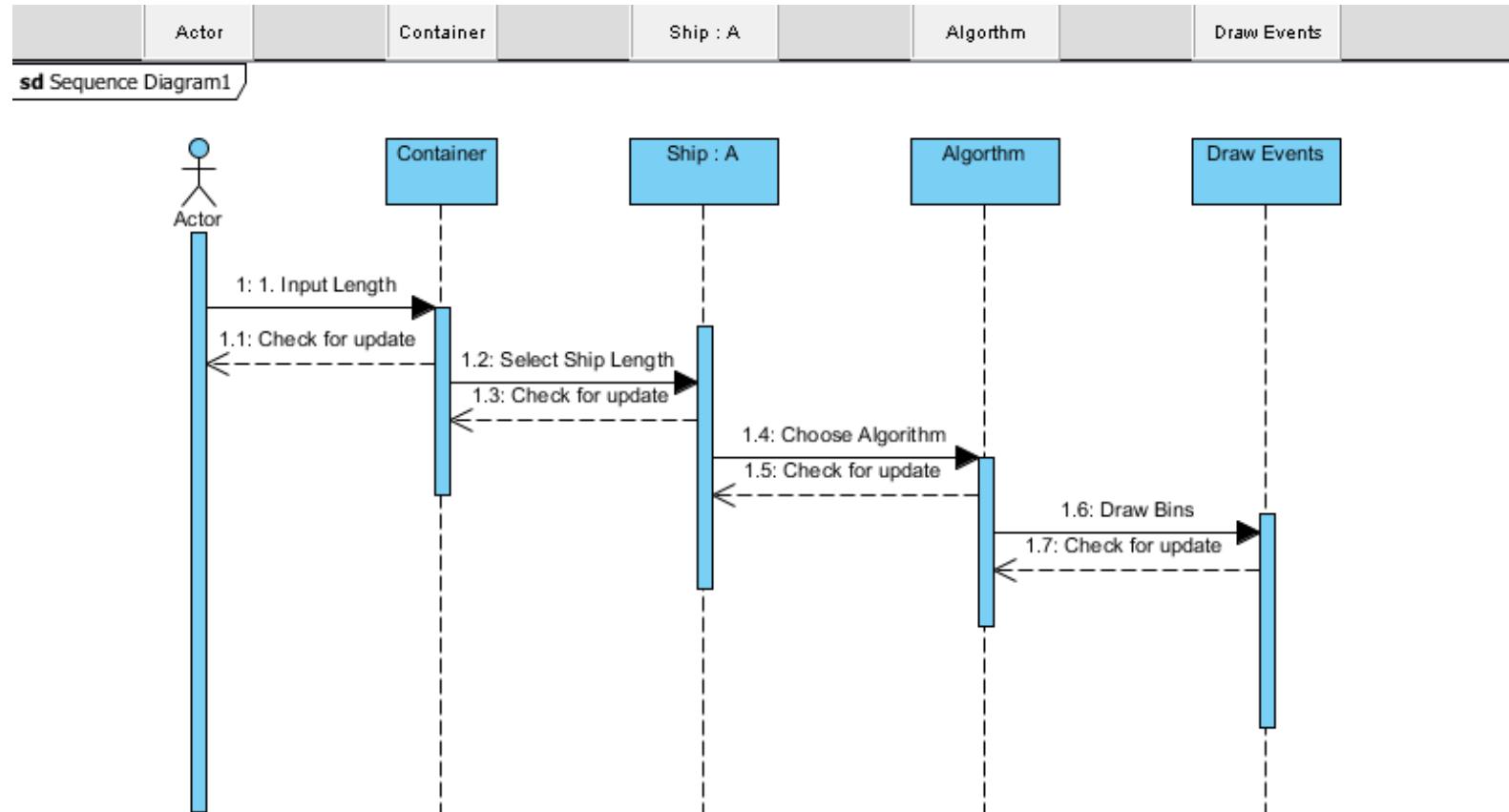


Figure 22: SD-04 Sequence Diagram (View Loading)

6.1.21 UC-06 Use Case Diagram (Change Language)

Name : Use Case Change Language		ID : UC-06
Description:		To Change the Language
Goal:		To Choose Language that user requires
Actor(s):		User
Assumption(s):		<ul style="list-style-type: none"> 1. User knows the path of the Menu to change language 2. English is the default language. 3. No third party tools are used
Pre-condition(s):		User have the access of lock and knowledge of the ClpForm
Post-condition(s):		Option for English ,French and Hindi
Primary (Happy) Path:		
<p>User navigates to the Help menu bar</p> <ol style="list-style-type: none"> 1. Move to Change Language 2. Select Language of the Choice 3. Apply Changes 		
Alternate Pathway(s):		
<ol style="list-style-type: none"> 1. Shortcut press – Alt+L 2. Choose language 		
Exception Pathway(s):		
None at the point		

Table 17: UC-06 Use Case Specification (Change Language)

6.1.22 UC-06 Use Case Diagram (Change Language)

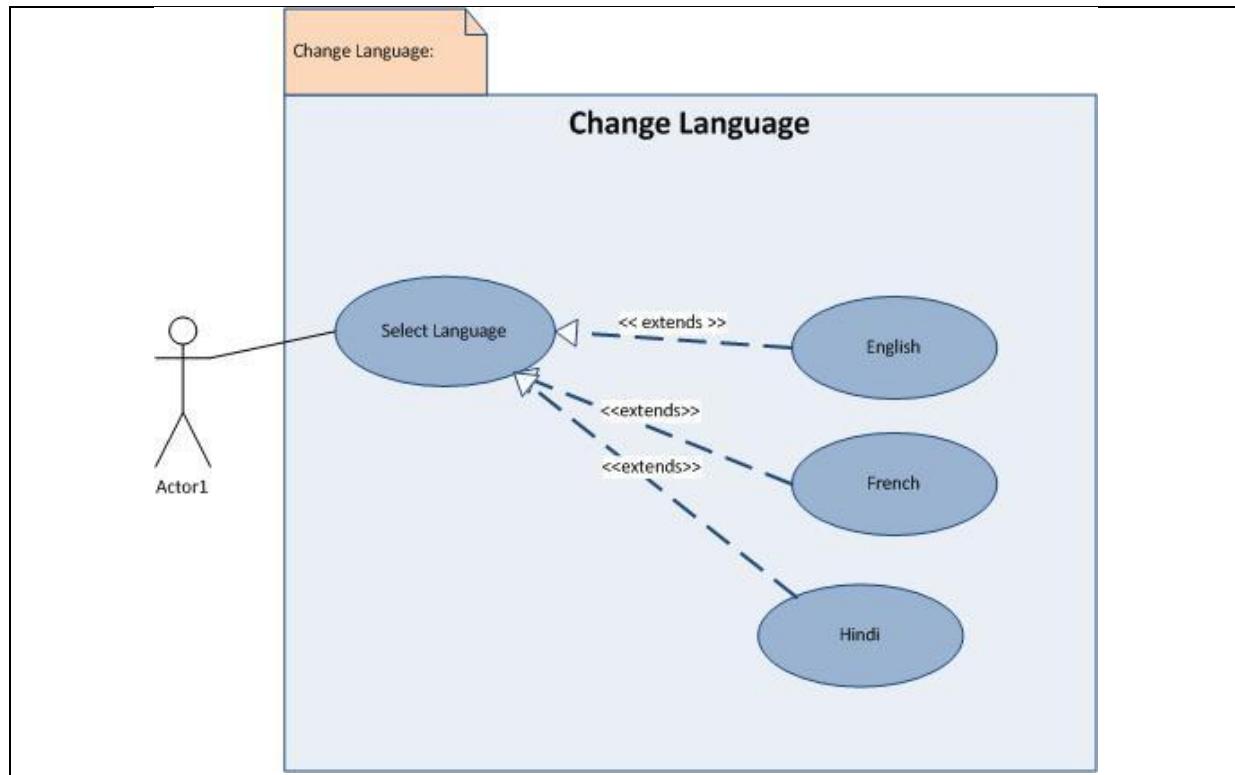


Figure 23: UC-06 Use Case Diagram (Change Language)

5.1.23 AD-06 Activity Diagram (Change Language)

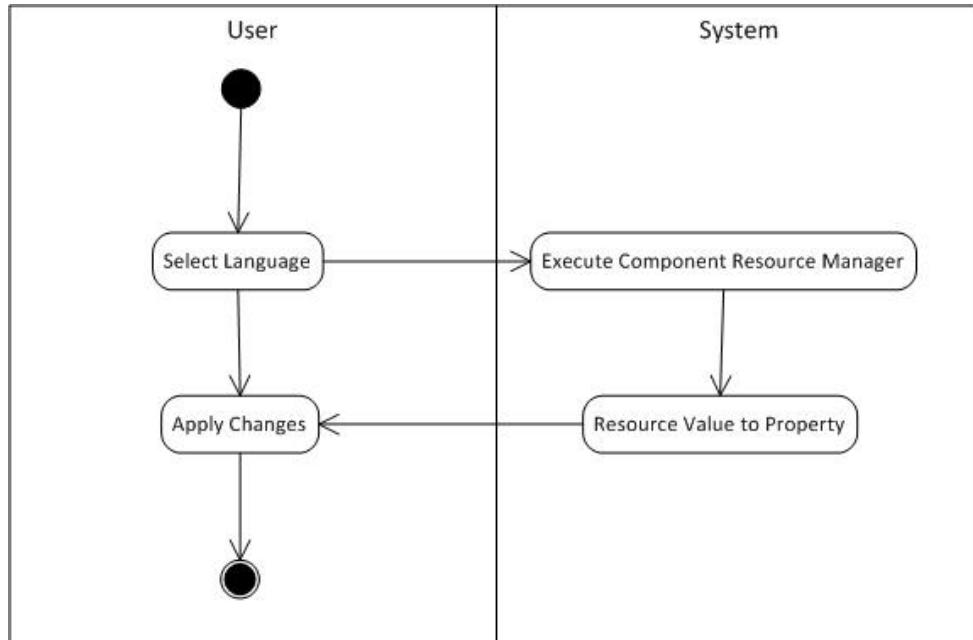


Figure 24: AD-06 Activity Diagram (Change Language)

6.1.24 SD-05 Sequence Diagram (Change Language)

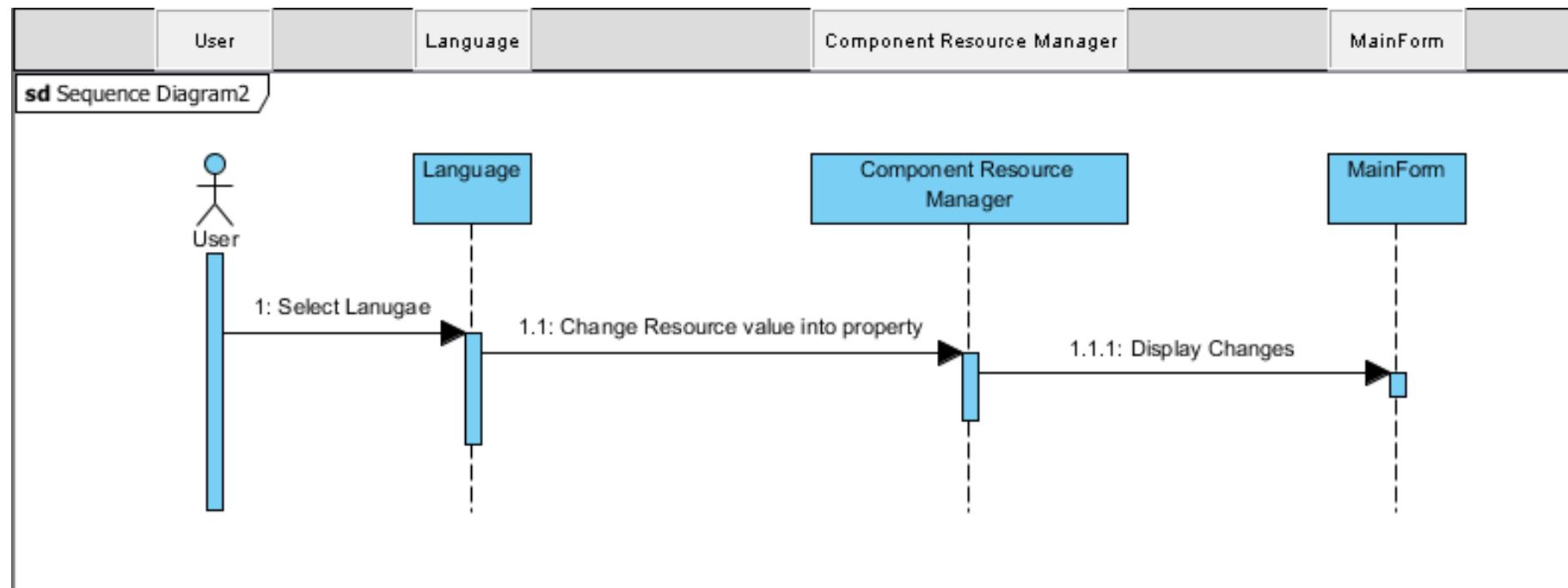


Figure 25: SD-05 Sequence Diagram(Change Language)

6.2 Class Diagrams

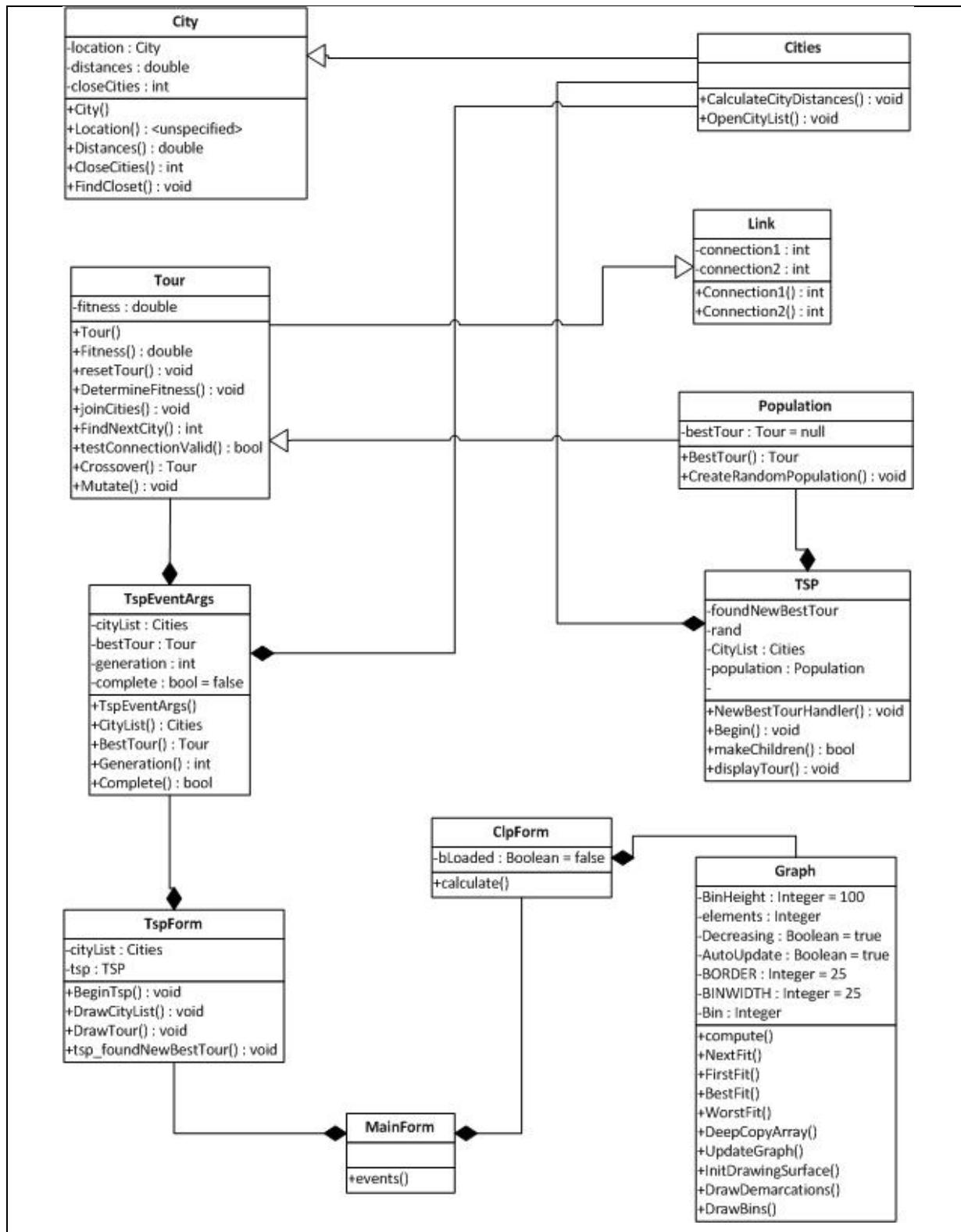


Figure 26: Class Diagram (Beacon)

6.3 Database Design Diagram

6.3.1 Data dictionary (TSP)

Table No : 1		
Table Name : TSP		
Description : Saves the final image of TSP into the database		
Attributes	Data type	Description
Tsp_id	Int(identity,100)	Primary key saves unique data with identity.
Image	Image	Image type saves image of picture box
Date	Date-Time	Date –Time

6.3.2 Data Dictionary (CLP)

Table No : 2		
Table Name : CLP		
Description : Saves the final image of CLP into the database.		
Attributes	Data type	Description
Tsp_id	Int(identity,100)	Primary key saves unique data with identity .
Image	Image	Image type saves image of picturebox
Date	Date-Time	Date –Time

6.4 Test Design

Today, software testing has become very systematic and standard testing techniques are available. Testing activity has also become all-encompassing in the sense that test cases are being developed right from the requirements specification stage. The developer has decided to design the test cases in design phase. The following diagram represents the format of unit testing that will be used to test.

Project Title	Name of the project		Date	Test conducting date	
Test Case Name	Name of the test case		Test ID	Id of the test case	
Conducted By	The name of the person who is performing the testing				
Description	Description of the test case.				
MODULE EXECUTION					
No	Steps (Inputs to Module)	Result Expected	Outputs from Module	Result (Pass/Fail)	Corrective Actions
Serial No.	Steps to be executed by tester	The developer's expectation's from the testing	Actual output provided by System	Whether the test fails or pass	Actions to be taken if actual output doesn't meet expectations.
Conclusion: Whether after corrective actions module works fine or not?					

Table 18: Test Plan Sample

The developer has decided to provide following test-cases:-

1. **Login-** It is for user authentication through username & password.
2. **Design Route map-** Basically a picture box used to design the tour
3. **Plan Route-** Algorithm behind the route
4. **Get Container Details-** Container details in the simples using XML
5. **Manage Loading –**Algorithms behind the loading
6. **View Loading –** final Report generation of loading
7. **Change Language –**Language Changer

6.4.1 Test Plan

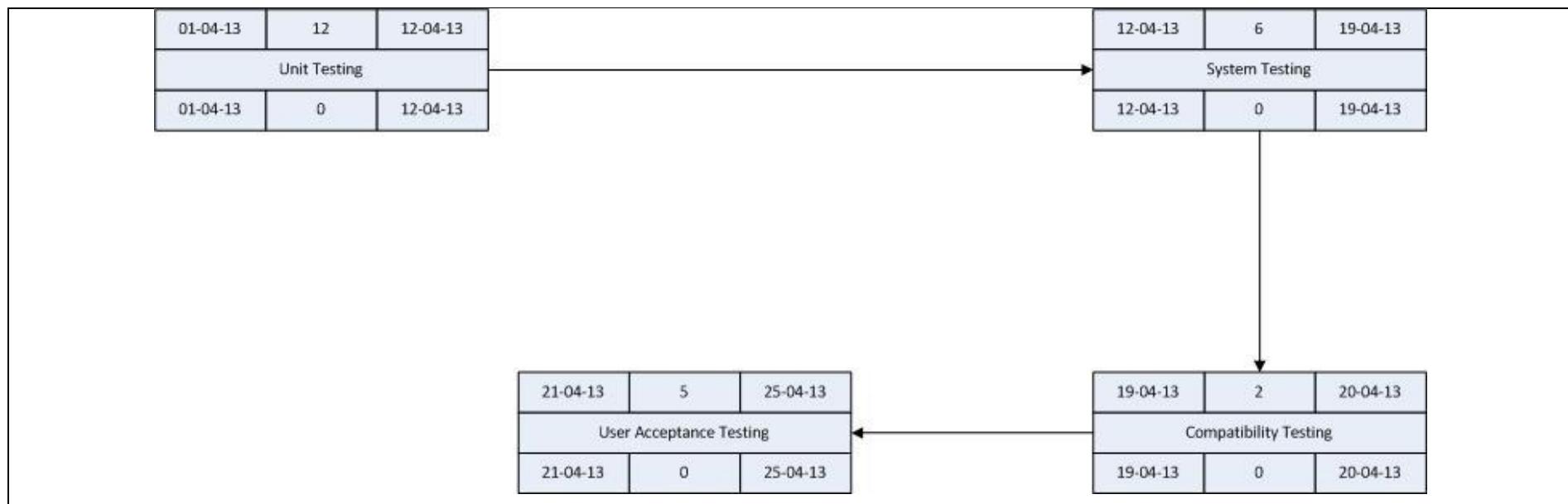


Figure 27: Test Plan PERT chart

Task ID	A1	A3	A3	A4
Task Details	Unit Testing	System Testing	Compatibility Testing	Usability Testing

Chapter 7: Implementation

This chapter describes the core, enhanced and special functionalities of the Beacon. Design are clearly explains these implementation through designs.

7.1 Tools used for Implementation

No.	Case Tools/ Development Tool/ Others	Purpose
1	Microsoft Visio 2007	To draw all the UML diagrams.
2.	Microsoft Project	To draw Project Gantt chart
3.	Visual Paradigm UML	To draw UML Diagrams
4.	Microsoft Visual Studio 2010	Used as a primary IDE for functionality, testing.
5.	SQL Server Management Studio	Used as the backend database.
6.	Windows 7 Home Premium	Used as the primary OS to publish software and testing.
7.	.NET Framework version 4	Framework of development
8.	Microsoft Word 2010	To document the report.
9.	Microsoft PowerPoint 2010	To prepare presentation slides.
10.	MS Paint	For image and logo creation.

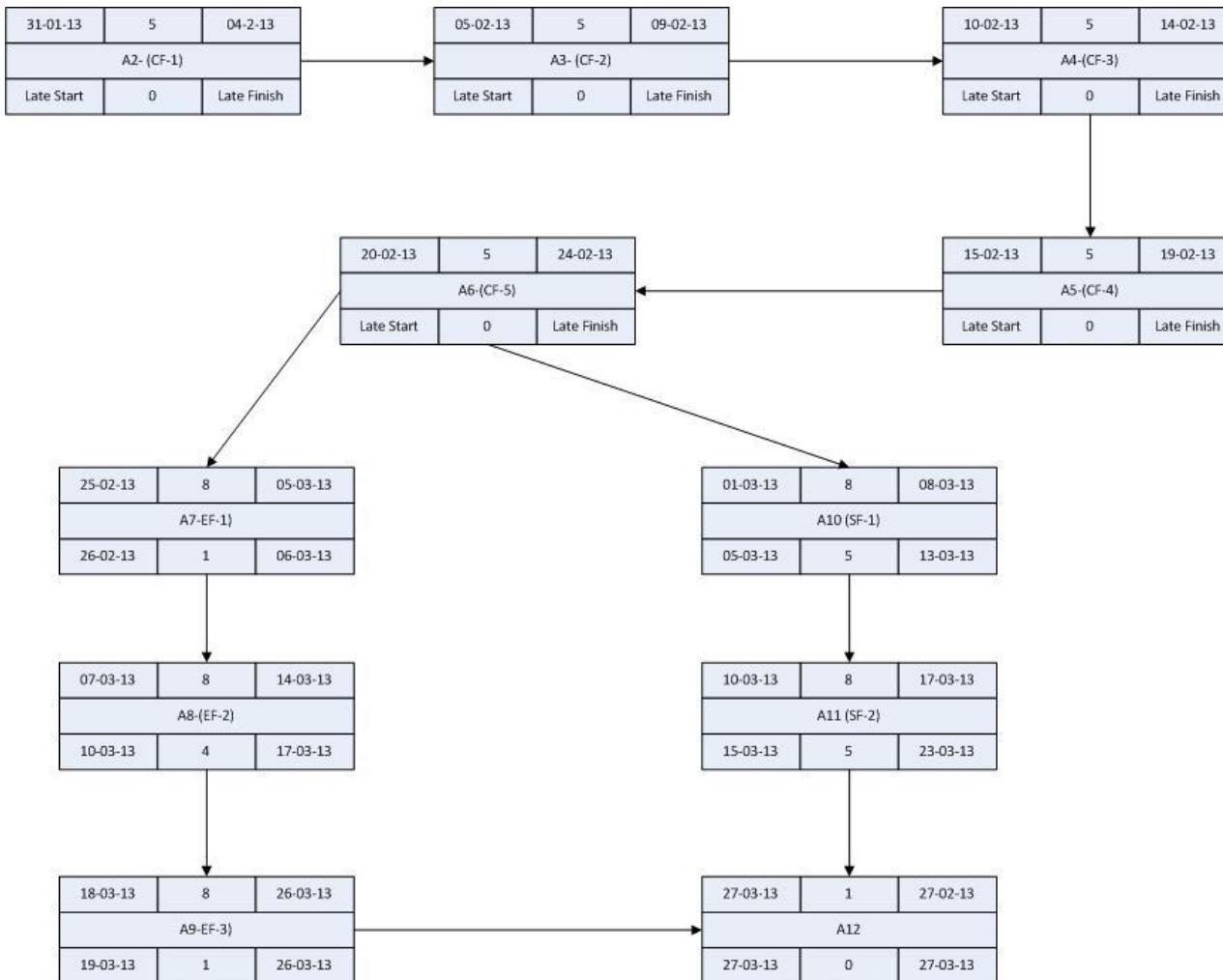
Table 19 : Table of Tools Used

7.1.1 Implementation Plan

Task ID	A1	A2	A3	A4	A5	A6
Task Details	Implementation Plan	CF-1	CF-2	CF-3	CF-4	CF-5
Task ID	A7	A8	A9	A10	A11	A12
Task Details	EF-1	EF-2	EF-3	SF-1	SF-2	Handover

Abbreviations

CF-Core Functionality	EF-Enhanced Functionality	SF-Special Functionality
-----------------------	---------------------------	--------------------------


Figure 28 : Implementation Plan

7.2 Implementing Complex Modules

The following sections describe how each module was implemented. In addition to that, each part explains the problems encountered and how they were tackled by the developer. Implementation phase of the entire system was divided as follows for easier and better implementation and maintenance.

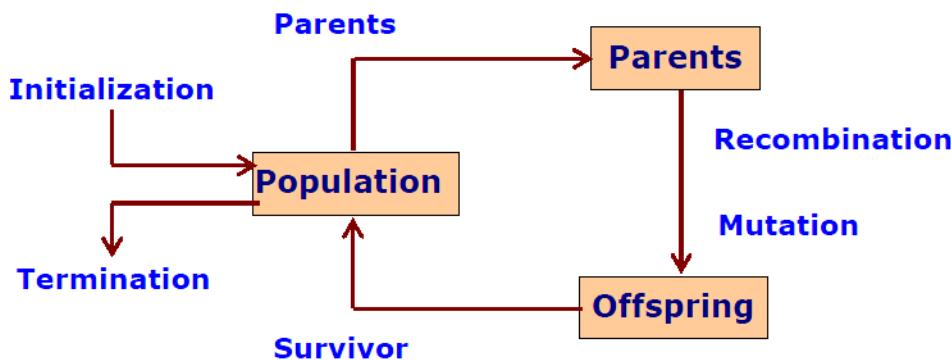


Fig. General Scheme of Evolutionary process

Figure 29 : General Scheme of Evolutionary Process

7.2.1 Create Random Population (TSP)

Initialization of population is probably the last of application issues to be solved, thus is not also easy to decide. In TSP, the population is randomly initialized and mixed, but the initialization of population could be more sophisticated task to do and think about. The initial population is base for all further population growth and development, so if the population is initialized incorrectly, e.g. not enough randomized for TSP, then the whole computation is going in wrong way.

7.2.1.1 Objective

GA is applied to the certain number of population that includes random cost and constraints to deal with. Mating and crossover is done to the specific chromosomes based on their cost. Tours are the individuals here.

- Create the initial set of random tours of the same cities.
- Choosing a starting point of the tour.
- Prefer choosing a city, the logic is not up to now, but it should be near enough to be chosen.
- Connect the last two cities

7.2.1.2 Description: This module basically covers the following subparts:

To begin a GA, we define an initial population of N_{pop} chromosomes. A matrix represents the population with each row in the matrix being a $1 * N_{var}$ array (chromosome) of continuous values.

$$pop = rand(N_{pop}, N_{var})$$

Equation 1 Equation to create random population

If initial look at this problem with $N = 13$ cities with the fixed starting and ending points, there are a total of $\frac{13!}{2} = 3.1135 * 10^9$ possible combinations can be generated. In the Implementation, developer has provided indefinite number of cities to select for and this results in a very large number of possible and unexpected tours.

7.2.1.3 Flow of the module:

Step 1: Population class inherits the Generic list of Tour that is strongly typed list of objects can be used to sort, search and manipulate. Tour consists the link to the cities in particular order and sub function to use the tour for crossover. Best Available Tour is null during the initiation of population.

Pseudo code	Packages
Step1. START	
Step2. CREATE Class population	<code>using System;</code>
Step 3. Inherit arrayList of tour	<code>using System.Collections;</code>
Step 4. SAVE private bestTour as NULL	<code>using System.Collections.Generic;</code>
Step 5 GET/SET bestTour to public function	<code>using System.Text;</code>
Step 6. Generate Random Population	
Step 7. UPDATE bestTour	
Step 8. STOP	

Table 20 : Pseudo code of Population Class

Step 2: Random generation of population is done by using three parameters, population size, city list and chance to choose closet city. New tour object is created by counting the total number of cities in the tour array. After this starting and ending point of tour is randomly selected (refer to TSP definition 3.3.1) A random tour is generated using random method and closet city parameters and last tour is first to create a cyclic map.

Then the function determines the Tour Fitness by using tour object. Fitness is the total distance of the tour. Add tour into the population array. If the Tour found has the best fitness, save it as best Tour

Pseudo code	External Class and function used
<p>Step 1. START</p> <p>Step 2. CREATE object of tour of length of city list</p> <p>Step 3. Randomly select first city</p> <p>Step 4. FOR EACH city</p> <p>Step 4.1. DO</p> <p>Step 4.2. IF new next city , ADD in array</p> <p>Step 4.3 ELSE ADD random next city</p> <p>Step 4.4 .END WHILE when next city is last city</p> <p>Step 4.5 END FOR</p> <p>Step 5 Connect last two cities using</p> <p>Step 6 Determine tour fitness</p> <p>Step 7. ADD tour in the Population</p> <p>Step 8. IF tour fitness is smaller than best tour fitness</p> <p>Step 9. REPLACE best tour with tour</p> <p>Step 10. STOP</p>	<p>Object</p> <p>Tour object is created of size city list</p> <p>Functions</p> <p>Tour.nextCity</p> <p>Tour.firstCity</p> <p>Tour.determineFitness</p> <p>CityList.CloseCities</p> <p>Rand.next</p>

Table 21: Pseudo code for population and bestTour

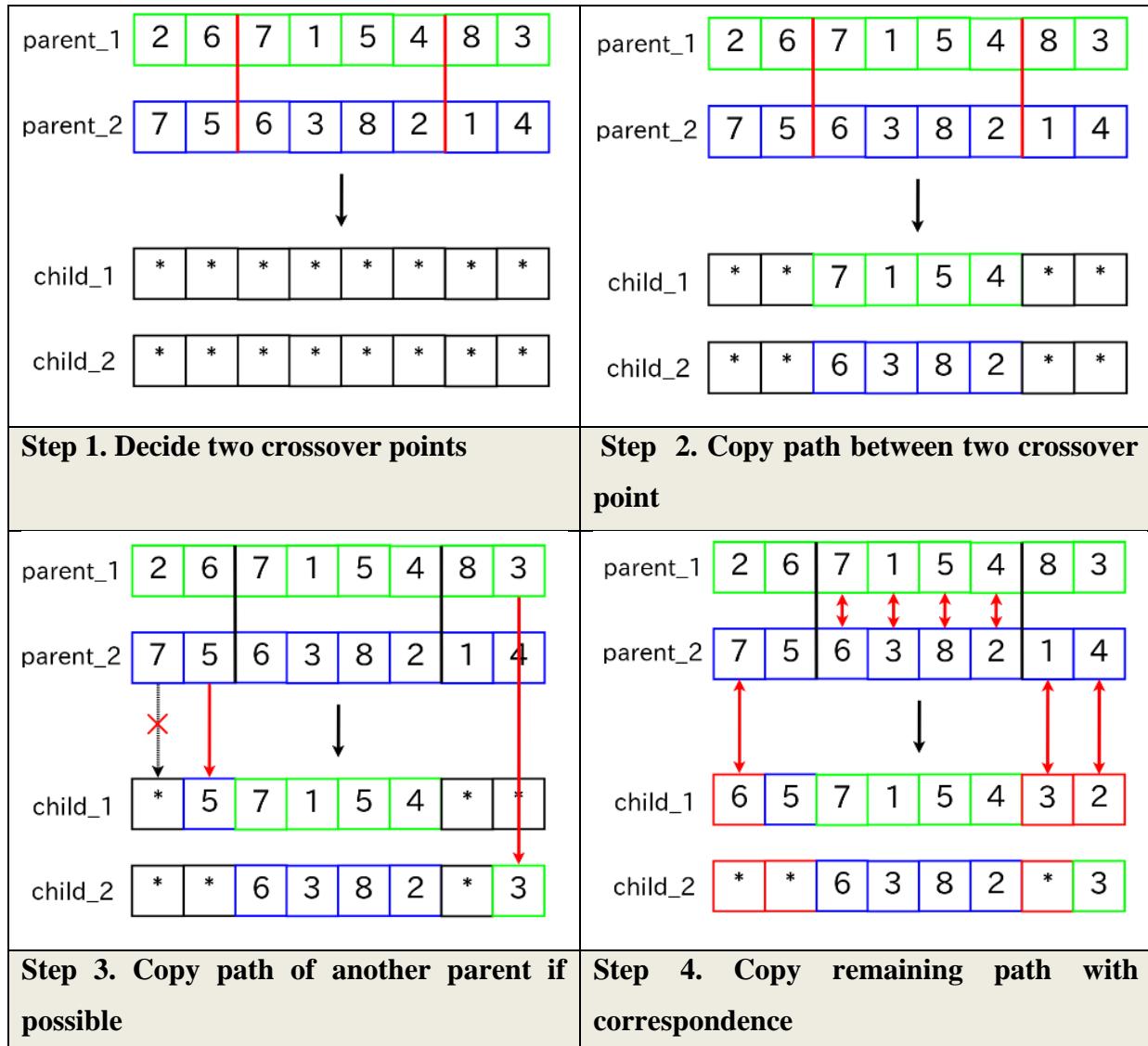
7.2.2. Crossover (TSP)

Crossover comes after the generation of initial population the GA flowchart. Crossover is done to the parents to create new offspring of better and hybrid cost.

7.2.2.1 Objective:

- Perform the crossover operation on two parent tours to create new child tour
- Total of two children will be created so twice execution of the function
- Count links of the cities using iteration
- Take all the common links from both parents and put them in child, this will inherit parent's traits into child tours.
- Randomly assign remaining links as parent's link cause multiple disconnected loops in the tour

7.2.2.2 Description: The crossover technique used here is much similar as PMX (partially mapped Crossover (**Goldberg, 1985**). A better crossover should introduce new edges to the children and common edges must be inherited. Common edges preserve parent's traits and new edges provide diversity and escape from local optima.



<p>The diagram shows the PMX process. Parent 1 (green border) has cities [2, 6, 7, 1, 5, 4, 8, 3]. Parent 2 (blue border) has cities [7, 5, 6, 3, 8, 2, 1, 4]. A vertical line separates cities 7 and 8. Red arrows indicate crossover points at cities 6, 1, 5, and 4. Child 1 (red border) receives cities 6, 5, 7, 1, 5, 4 from Parent 1 and 3, 2 from Parent 2. Child 2 (red border) receives cities 4, 7, 6, 3, 8, 2 from Parent 2 and 5, 3 from Parent 1.</p>	<p>The result of PMX is shown in two columns. The left column is labeled "Step 5. Copy remaining path with correspondance" and the right column is labeled "Step 6. Result of PMX".</p> <table border="1"> <tr> <td>parent_1 [2, 6, 7, 1, 5, 4, 8, 3]</td><td>parent_1 [2, 6, 7, 1, 5, 4, 8, 3]</td></tr> <tr> <td>parent_2 [7, 5, 6, 3, 8, 2, 1, 4]</td><td>parent_2 [7, 5, 6, 3, 8, 2, 1, 4]</td></tr> <tr> <td>child_1 [6, 5, 7, 1, 5, 4, 3, 2]</td><td>child_1 [6, 5, 7, 1, 5, 4, 3, 2]</td></tr> <tr> <td>child_2 [4, 7, 6, 3, 8, 2, 5, 3]</td><td>child_2 [4, 7, 6, 3, 8, 2, 5, 3]</td></tr> </table>	parent_1 [2, 6, 7, 1, 5, 4, 8, 3]	parent_1 [2, 6, 7, 1, 5, 4, 8, 3]	parent_2 [7, 5, 6, 3, 8, 2, 1, 4]	parent_2 [7, 5, 6, 3, 8, 2, 1, 4]	child_1 [6, 5, 7, 1, 5, 4, 3, 2]	child_1 [6, 5, 7, 1, 5, 4, 3, 2]	child_2 [4, 7, 6, 3, 8, 2, 5, 3]	child_2 [4, 7, 6, 3, 8, 2, 5, 3]
parent_1 [2, 6, 7, 1, 5, 4, 8, 3]	parent_1 [2, 6, 7, 1, 5, 4, 8, 3]								
parent_2 [7, 5, 6, 3, 8, 2, 1, 4]	parent_2 [7, 5, 6, 3, 8, 2, 1, 4]								
child_1 [6, 5, 7, 1, 5, 4, 3, 2]	child_1 [6, 5, 7, 1, 5, 4, 3, 2]								
child_2 [4, 7, 6, 3, 8, 2, 5, 3]	child_2 [4, 7, 6, 3, 8, 2, 5, 3]								

Table 22 : Partially Mapped Crossover Demonstration

7.2.2.3 Flow of the Module:

Step 1: The Shared crossover modules in the Tour have three parameters parent1, parent2 and city list. Basically the module is mathematical selection and manipulation of the link of the city. Tour object is created of size city list to maintain new tour. Calculation of the city link is done through iteration of city list array.

Step 2: Parent's traits are saved as the link that are common to both the parents is also saved in child. To implement this, developer has used iteration of city and then if connection of links of parents are same, save it for next city. Since there are 2 parents and 2 connections, a total of $2^2 = 4$ checking will happen. Two offspring are their so, it require to execute this step twice.

Step 3: Since parents would cause multiple disconnected loops in the tour. To fill those loops random links are connected to the loops to join cities. Next city will be random city until the flow finds a new city to link to.

Pseudo code
Step 1. START
Step 2. Take parameters as Parent1 , Parent2 and city list
Step 3. CREATE object of Tour
Step 4. FOR EACH city
Step 4.1. SAVE cityusage

Step 4.2. END FOR
Step 5. FOR EACH city
Step 6. IF parent1 connection 1 is EQUALS parent2 connection 1
Step 7. join cities in child connection to next city
Step 8. REPEAT step 6 7 times again
Step 9. FOR EACH city
STEP 9.1. WHILE cityusage is smaller than 2
STEP 9.2. next city EQUALS random city
STEP 10. RETURN child
STEP 11. STOP

Table 23: Pseudo Code for Links in Link Class (TSP)

7.2.3 Map Editor - GUI Drawing (TSP)

Map Editor Module is the used to showcase the route and the working of algorithm at runtime.

7.2.3.1 Objective:

- To Display the cities in the Picture Box of the graph
- Display the routes within the city and algorithm is finding the solution
- Enable to create cities at the mouse click at any co-ordinates.
- Load co-ordinates from XML file
- Can be saved in the image format to be saved in the database

7.2.3.3 Description: Map editor is basically a picture box in the system that works on various events. Graphical properties of Picture Box are used to create the map editor. It consists of several mouse down event, zooming property, saving property , custom graph creator and other small features.

7.2.3.4 Flow of the Module:

Step 1: The Map editor start with the loading of picture box. When mouse down events is initiated, the map looks for drawCityList function, City list are created by taking the co-ordinates of picture box in consideration. Also city list can be shown in map by loading it through XML file.

Pseudo Code	Libraries
Step 1. START	using System;
Step 2. CREATE cityimage EQUALS picturebox	using System.Collections.Generic;
Step 3. DECLARE graphics for cityImage	using System.ComponentModel;
Step 4. FOR EACH city in citylisy	using System.Data;
Step 4.1. DRAW small rectangle	using System.Drawing;
Step 4.2 END FOR	using System.Text;
Step 5. RETURN cityimage	using System.Windows.Forms;
Step6. UPDATE city count	using System.Threading;
Step 7. STOP	using System.IO;
XML file based loading	using System.Globalization;
Step 1. START	
Step 2. TRIGGER event	
Step 3. OPEN file Dialog	
Step 4. Filter.xml file	
Step 5. IF GET File name	
Step 6. OPEN city list from file	
Step 7. ELSE	
Step 8. PRINT error file not found	
Step 9. END IF	
Step 10. STOP	

Table 24 :Pseudo code for XML based city loading in map

Step 2: After loading the city list, next task is to apply algorithm to generate best possible tour. Developers approaches to show the working of algorithm with the TSP Events. Every New tour in the process is reflected into the map with total distance and routes.

Pseudo code
Step 1 .START
Step 2. GET last best fitness value
Step 3. GET last iteration value
Step 4. next city EQUALS best tour connection1
Step 5. FOR EACH city in city list
Step 5.1. DRAW rectangle for city

Step 5.2. DRAW the line connecting city
 Step 5.4. IF last city EQUALS NOT best city connection 1
 Step 5.5. next city EQUALS best tour connection 1
 Step 5.6. ELSE next city EQUALS best tour connection 2
 Step 5.7. END FOR
 Step 5.8. RETURN city image
 Step 6. STOP

Table 25:Pseudo Code for new City generation in pictureBox

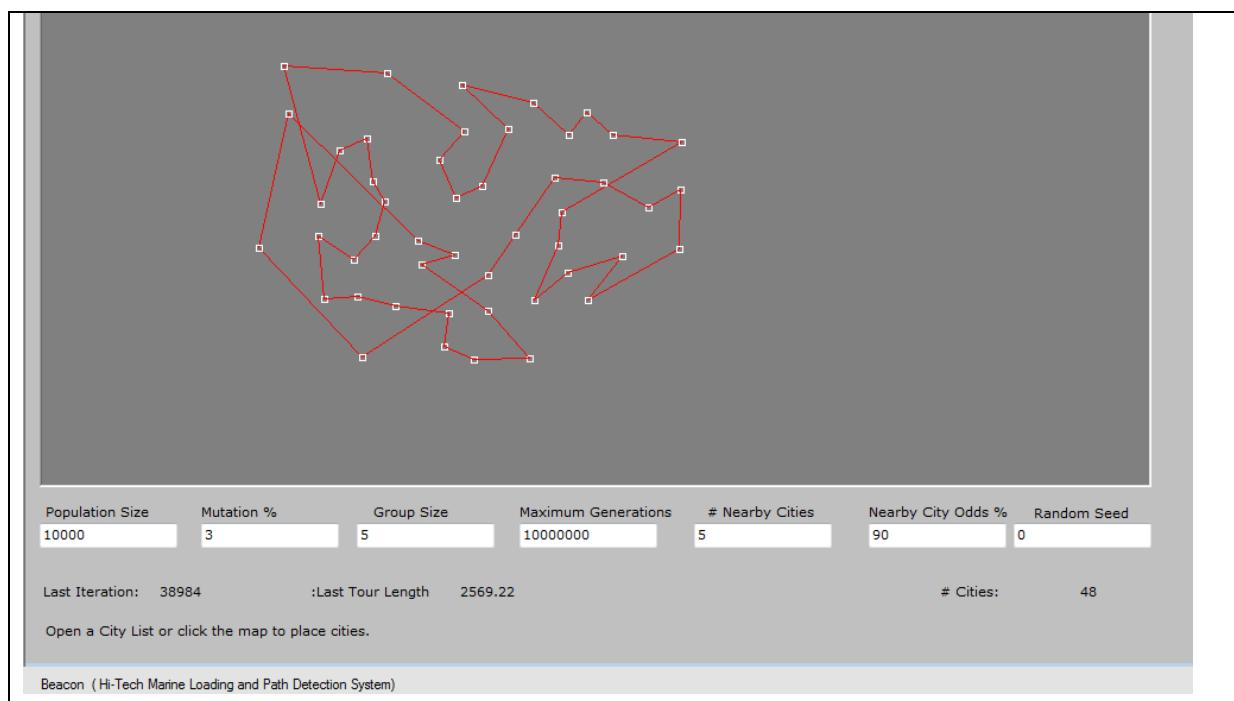


Figure 30 Figure of Running Algorithm of Route Map

7.2.4 Container loading Algorithm – Bin Packing Algorithm

Bin Packing is a mathematical way to deal with efficiently fitting Elements into Bins. The hitch is that Bin packing problem is classified as the NP-complete problem. Basically means that there's no way of being guaranteed the best solution without checking every possible solution

7.2.4.1 Objective

- To take the elements as the length of the container
- Apply approach of solving through first fit, next fit, best fit, and worst fit.
- Bin height can be increased and decreased during the execution

7.2.4.2 Description

The goal of every Bin Packing algorithm is to use the least amount of Bins to hold the required number of Elements

- **Bin:** The fixed size container that can hold the Elements
- **Bin Height:** The specified amount that each Bin can hold.
- **Element:** An item that is to be placed in a Bin having a certain Element Height
- **Element Height:** The amount of Bin space the Element will take up if placed in that Bin.

7.2.4.4 Flow of the Module:

Step 1: The first step includes getting the elements from the field and the choice of algorithm to choose for when the event triggered. The decreasing check box apply this algorithm in the decreasing order of array.

Libraries	Pseudo code
<pre>using System; using System.Collections.Generic; using System.ComponentModel; using System.Data; using System.Drawing; using System.Text; using System.Windows.Forms; using System.Threading; using System.IO; using System.Globalization; using System.Data.SqlClient;</pre>	<p>Step 1: START</p> <p>Step 2. IF decreasing equals TRUE</p> <p>Step 2.1. SORT elements</p> <p>Step 2.2. REVERSE array</p> <p>Step 2.3. END IF</p> <p>Step 3 STOP</p>

Table 26: Packages used and pseudo code for decreasing order of bins

Step 2: Next-Fit Algorithm - This algorithm starts at the beginning of the Elements array and steps through each one. Once Bin 1 is full, it moves on and starts placing elements into Bin 2, never looking back to see if an Element in the future may fit inside Bin 1.

Step 3: First-Fit Algorithm: It steps through the Elements sticking them into the first Bin it can, if there aren't any Bins that it will fit into, a new Bin is added. It is more efficient than Next Fit Algorithm.

Step 4: Worst-Fit Algorithm: It comes with some extra processing though (on small data sets it doesn't really matter). The only difference between the two algorithms is that Worst

Fit picks the Bin with the most amount of free space (or creates a new Bin if no existing one can fit the Element) instead of just picking the first Bin available.

Step 5: Best-Fit Algorithm: Instead of picking a Bin with the most amount of free space, this algorithm picks the Bin with the least amount of free space in it that can still hold the current Element. The results you obtain by using this algorithm are not always the same as the Worst Fit, sometimes it is slightly better, other times it is not. It depends on the nature of the data supplied.

Pseudo Code
Step 1. START
Step 2. IF elements equals NULL
Step 2.1. EXIT
Step 3. Declare elements copy, Bin number, Bin Element, Bin count as integer
Step 4. Declare best bin , bin amount ,i , j ,k as integer
Step 5. COPY of array
Step 6. SORT array in descending order
Step 7. FOR EACH element
Step 7.1 best bin and best bin amount EQUALS -1
Step 7.1.1 FOR EACH bin number
Step 7.1.2 bin element =bin(j)
Step 7.1.3 COUNT amount in bin
Step 7.1.4 END FOR
Step 7.2 FIND the most full bin
Step 7.3 RETURN best bin
Step 7.4 END FOR
Step 8. IF best bin equals -1 THEN
Step 9. CREATE new bin
Step 10. Initialise first element to new bin
Step 11. ELSE
Step 12 Place element in the best bin
Step 12 END FOR
Step 13 REMOVE unused elements

Step 14 STOP

```
//Make a copy of the array incase we need to sort it
DeepCopyArray(Elements, ElementsCopy);

//Sort in descending order if needed
if (this.Decreasing == true)
{
    Array.Sort(ElementsCopy);
    Array.Reverse(ElementsCopy);
}
```

Table 27: Pseudo code for Best Fit Algorithm and decreasing order of bin

7.3 Problems (Hardest Task)

7.3.1 Problem 1: (Mapping TSP with GA and approach of solving)

For solving the route planner problem the developer had to do the following task on the single action

1. Identifying the cost function for the TSP
2. Find the closet city with minimum distance logic
3. Code the chromosome (as in Genetic Algorithm) of the TSP using data structure.
4. Define fitness function that decide the overall rank of a tour
5. Generate Genetic Algorithm engine for population control and filtering
6. Mating algorithm that do crossover between two parents and generate new offspring.
7. Implement Mutation to reallocate the population from any local extrema
8. Apply right initial parameters.

(Ray, Bandyopadhyay, & Pal, 2004)

Solution

1. **Cost Function:** The cost function id determined by the public sub procedures in Cities class. It is basically derived from the Pythagoras Theorem of calculation that is

$$\text{Distance} = \sqrt{(x - x_1)^2 + (y - y_1)^2}$$

where x, y is next city's location and x_1, y_1 is current city location.

Equation 2: Distance Calculation

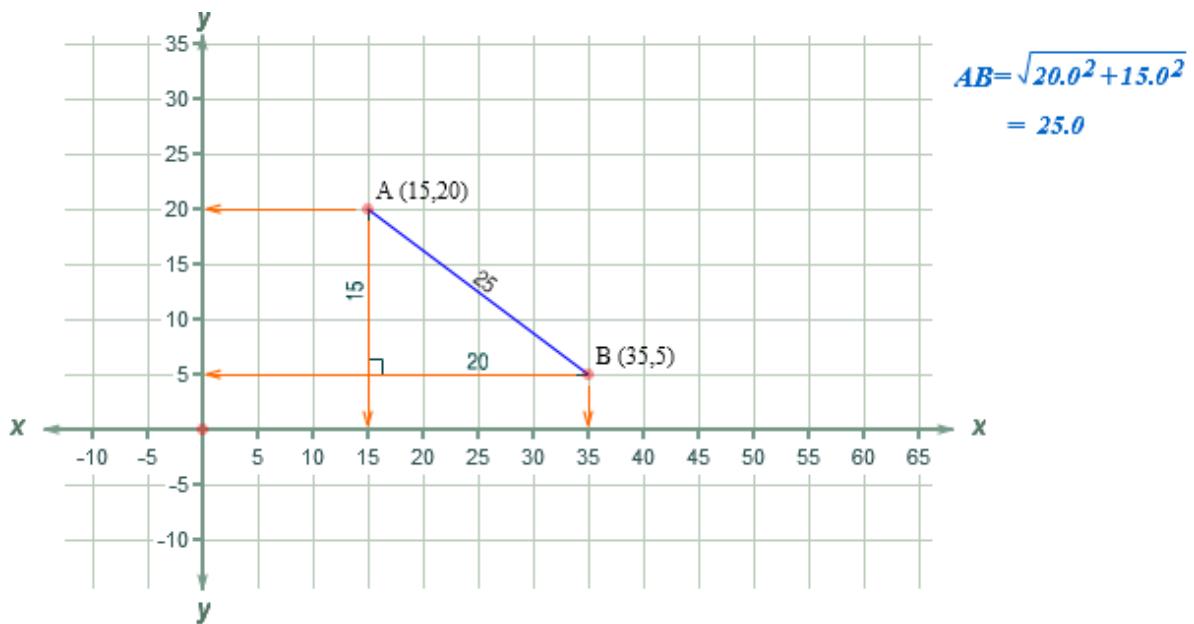


Figure 31 : Distance calculation between two co-ordinates using RMS method

Source: <http://www.mathopenref.com/coorddist.html>

The City distances are calculated for all the cities and further the distance are added to the array list for processing.

- 2. Closet city with minimum distance logic:** When creating the initial population of tours, this is a greater chance that a nearby city will be chosen for a link. This is the number of cities that will be considered close

```

1.START
2. FOR EACH i
    2.1 FOR EACH cityNum
        2.1.1 IF distance of cityNim is smaller then shortest distance THEN
            2.1.2 shortest distance EQUALS distance of cityNum
            2.1.3 shortest city is cityNum
        2.1.4 END IF
    2.2 END FOR
    2.3 ADD shortest city of closecities array
2.4 END FOR
    
```

Table 28 : Pseudo code for finding the closet city

The final array include the array of the shortest distance to the next city of the all the distances of that city to all other cities. The final values stored in the process are the cities that have the shortest distance.

3. Map the Chromosomes of TSP for GA

The chromosome is set of ordered indexes of cities, through which the traveller goes. The TSP is coded into data structure, which can be handed like a chromosome. The array index is our route order, the contents of each array elements is a city number. Hence, if developer wrote the following pseudo code:

A[] = { 5 ,64, 23 ,8 , 32 }

It would mean that the visit of city 5 is the starting point followed by city 64, 23, 8, 32 and so on. This is how ‘chromosomes’ or trial solutions is encoded in genetic algorithm. In the code, tour class represents the order of cities linked together and object of the tour class is used to find fitness and other processing in GA.

4. Fitness Function

The fitness is calculated for each chromosome that further mates to create new offspring. Fitness is the criteria of ranking the tours. TSP determines fitness as the total distance of a Tour, The higher the distance, lower is the fitness.

$$Fitness = \frac{1}{Total\ Distance}$$

Equation 2 : Fitness Equation

Root Mean Square (RMS) value is used to determine fitness from the cost function.

```
Public Sub DetermineFitness(ByVal cities As Cities)
    Fitness = 0
    Dim lastCity As Integer = 0
    Dim nextCity As Integer = Me(0).Connection1

    For Each link As Link In Me
        Fitness += cities(lastCity).Distances(nextCity)

        If lastCity <> Me(nextCity).Connection1 Then
            lastCity = nextCity
            nextCity = Me(nextCity).Connection1
        Else
            lastCity = nextCity
            nextCity = Me(nextCity).Connection2
        End If
    Next
End Sub
```

Table 29 : Fitness Calculation code for each tour

5. Genetic Algorithm Engine for Population control and Filtering

Select genetic algorithm engine care about the population, its growth, filtering, selecting and sorting individuals and random mutations of chromosomes. It also handles all the computation process and optionally enables multi-threading processing of the problem.

Description of Population Generation (Refer to 7.2.1.2)

Pseudo code of Population Generation (Refer to 7.2.1.3)

6. Mating Algorithm Crossover

Mating algorithm is very important way, how to create offspring (child) from the parent chromosomes. The task of the mating is to create new offspring, which has characteristics from both parent and improves the quality returned by fitness function. Developer recognises this task as the most difficult and most important as it decides, how well and how fast will the population improve.

Description of Crossover in GA (Refer to 7.2.2.2)

Pseudo code of Crossover in GA (Refer to 7.2.2.3)

7. Implement Random Mutation

Enable the mating algorithm for moving the population from local extrema. It could happen that the computation is in such a state locked in non-optimal position and needs external (random impulse to break the disability apart a start again moving ahead. The critical factor is that if there is too much of random process into computation, the result will never be the most optimal.

Description of Mutation in GA

Random mutations are executed side by side of mating algorithm (crossover) for moving the population from local extrema. It could happen that the computation is in such a state locked in non-optimal position and needs external (random) impulse to break the disability apart an start again moving ahead. On the other hand, if there is too much of random process into computation, the result will never be the most optimal.

Mutation type is Random mutation, developer is changing one of the links in this tour randomly, based on initial mutation % , more links can be changed.

Code of Mutation in GA

```

Public Sub Mutate(ByVal rand As Random)
    Dim cityNumber As Integer = rand.[Next](Me.Count)
    Dim link As Link = Me(cityNumber)
    Dim tmpCityNumber As Integer

    ' Find which 2 cities connect to cityNumber, and then connect them directly
    If Me(link.Connection1).Connection1 = cityNumber Then
        ' Conn 1 on Conn 1 link points back to us.
        If Me(link.Connection2).Connection1 = cityNumber Then
            ' Conn 1 on Conn 2 link points back to us.
            tmpCityNumber = link.Connection2
            Me(link.Connection2).Connection1 = link.Connection1
            Me(link.Connection1).Connection1 = tmpCityNumber
        Else
            ' Conn 2 on Conn 2 link points back to us.
            tmpCityNumber = link.Connection2
            Me(link.Connection2).Connection2 = link.Connection1
            Me(link.Connection1).Connection1 = tmpCityNumber
        End If
    Else
        ' Conn 2 on Conn 1 link points back to us.
        If Me(link.Connection2).Connection1 = cityNumber Then
            ' Conn 1 on Conn 2 link points back to us.
            tmpCityNumber = link.Connection2
            Me(link.Connection2).Connection1 = link.Connection1
            Me(link.Connection1).Connection2 = tmpCityNumber
        Else
            ' Conn 2 on Conn 2 link points back to us.
            tmpCityNumber = link.Connection2
            Me(link.Connection2).Connection2 = link.Connection1
            Me(link.Connection1).Connection2 = tmpCityNumber
        End If
    End If

    Dim replaceCityNumber As Integer = -1
    Do
        replaceCityNumber = rand.[Next](Me.Count)
    Loop While replaceCityNumber = cityNumber
    Dim replaceLink As Link = Me(replaceCityNumber)

    Do
        replaceCityNumber = rand.[Next](Me.Count)
    Loop While replaceCityNumber = cityNumber
    Dim replaceLink As Link = Me(replaceCityNumber)

    ' Now we have to reinsert that city back into the tour at a random location
    tmpCityNumber = replaceLink.Connection2
    link.Connection2 = replaceLink.Connection2
    link.Connection1 = replaceCityNumber
    replaceLink.Connection2 = cityNumber

    If Me(tmpCityNumber).Connection1 = replaceCityNumber Then
        Me(tmpCityNumber).Connection1 = cityNumber
    Else
        Me(tmpCityNumber).Connection2 = cityNumber
    End If

```

 Mutate
 Sub Procedure

Random City Location

8. Apply the right initial parameters

The initial parameter includes parameters like population size, size, mutation ratio, population growth. These parameters are relevant to used algorithm, and can influence the computation process a lot.

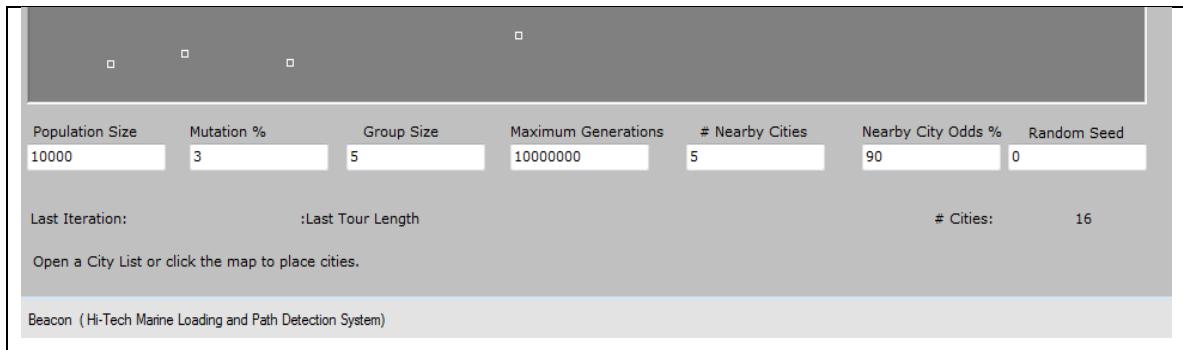


Figure 33: Initial and default parameters of TSP in GA

7.3.2 Problem 2 (Generating the Graph of Loading)

The main problem is this to generate the graph on the change event of elements and algorithm selected. The graph of loading is made and control using user control feature of Visual Basic by which custom graphical interfaces can be created and reused. User control is creates by combining controls and components into a user control container and further properties of graph and custom functionality of changes graph are added to extend the functionality of binpackingGraph (Anonymous, 2003)

For Solving the Container Loading problem, Developer have followed the following steps –

1. Create BinPackingGraph as user control
2. Create Algorithm as Next-Fit, First-Fit, Worst –Fit, Best Fit.
3. Create Drawing methods for BinPacking Graph
4. Select Events to start functions

Solutions:

1. **BinPackingGraph as User Control:** The binPackingGraph2 used in the system basically consists of a picture box named pbDrawingSurface. provide a means by which custom graphical interfaces can be created and reused. A user control is essentially a component with a visual representation. As such, it might consist of one

or more Windows Forms controls, components, or blocks of code that can extend functionality by validating user input, modifying display properties, or performing other tasks required by the Developer.

```
#region Component Designer generated code

/// <summary>
/// Required method for Designer support - do not modify
/// the contents of this method with the code editor.
/// </summary>
private void InitializeComponent()
{
    this.pbDrawingSurface = new System.Windows.Forms.PictureBox();
    ((System.ComponentModel.ISupportInitialize)(this.pbDrawingSurface)).BeginInit();
    this.SuspendLayout();
    //
    // pbDrawingSurface
    //
    this.pbDrawingSurface.BorderStyle = System.Windows.Forms.BorderStyle.FixedSingle;
    this.pbDrawingSurface.Dock = System.Windows.Forms.DockStyle.Fill;
    this.pbDrawingSurface.Location = new System.Drawing.Point(0, 0);
    this.pbDrawingSurface.Name = "pbDrawingSurface";
    this.pbDrawingSurface.Size = new System.Drawing.Size(150, 150);
    this.pbDrawingSurface.TabIndex = 0;
    this.pbDrawingSurface.TabStop = false;
    this.pbDrawingSurface.Resize += new System.EventHandler(this.pbDrawingSurface_Resize);

    this.pbDrawingSurface.Paint += new System.Windows.Forms.PaintEventHandler(this.pbDrawingSurface_Paint);
    //
    // BinPackingGraph
    //
    this.AutoScaleDimensions = new System.Drawing.SizeF(6F, 13F);
    this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;
    this.Controls.Add(this.pbDrawingSurface);
    this.Name = "BinPackingGraph";
    this.Load += new System.EventHandler(this.BinPackingGraph_Load);
    this.Disposed += new System.EventHandler(this.BinPackingGraph_Disposed);
    ((System.ComponentModel.ISupportInitialize)(this.pbDrawingSurface)).EndInit();
    this.ResumeLayout(false);
}

#endregion

private System.Windows.Forms.PictureBox pbDrawingSurface;
```



Figure 34:Picture Box used as a Drawing Surface by Usercontrol

2. Algorithms: Four different algorithms have been implemented to get the most suitable loading plan.

(Refer to 7.2.4.4 for Description and Pseudocode)

Following is the code for Best –Fit Algorithms

```

Private Sub BestFit()
    'Checks to make sure everything is initialized
    If Elements Is Nothing Then Exit Sub

    Dim ElementsCopy(Elements.GetUpperBound(0)) As Integer
    ReDim Bins(0)
    'Bin Number we are on, Bin Element we are on, Amount placed in the current Bin
    Dim BinNumber, BinElement, BinCount As Integer
    Dim BestBin, BestBinAmount As Integer
    Dim i, j, k As Integer

    'Make a copy of the array incase we need to sort it
    DeepCopyArray(Elements, ElementsCopy)

    'Sort in descending order if needed
    If Me.Decreasing = True Then
        Array.Sort(ElementsCopy)
        Array.Reverse(ElementsCopy)
    End If

    'Declare the first element in the first Bin
    ReDim Bins(0)(0)

    'Loop through each Element and place in a Bin
    For i = 0 To ElementsCopy.GetUpperBound(0)
        BestBin = -1
        BestBinAmount = -1

        For j = 0 To BinNumber
            BinElement = Bins(j).GetUpperBound(0)

            'Count the amount placed in this Bin
            BinCount = 0
            For k = 0 To BinElement
                BinCount += Bins(j)(k)
            Next

            'Find the most full Bin that can hold this Element
            If BestBinAmount < BinCount AndAlso BinCount + ElementsCopy(i) <= Me.BinHeight Then
                BestBinAmount = BinCount
                BestBin = j
            End If
        Next

        If BestBin = -1 Then
            'There wasn't room for the Element in any existing Bin
            'Create a new Bin
            ReDim Preserve Bins(BinNumber + 1)
            BinNumber += 1

            'Initialize first element of new bin
            ReDim Bins(BinNumber)(1)
            BinElement = 0

            Bins(BinNumber)(BinElement) = ElementsCopy(i)
        Else
            'There's room for this Element in an existing Bin
            'Place Element in "Best Bin"
            BinElement = Bins(BestBin).GetUpperBound(0)
            ReDim Preserve Bins(BestBin)(BinElement + 1)
            Bins(BestBin)(BinElement) = ElementsCopy(i)
        End If
    Next

    'All Elements have been place, now we go back and remove unused Elements
    For i = 0 To BinNumber
        For j = 0 To Bins(i).GetUpperBound(0)
            If Bins(i)(j) = 0 Then
                ReDim Preserve Bins(i)(j - 1)
            End If
        Next
    Next

    GC.Collect()

```

Sort in Descending
order

Iteration

If Bin is Full

ReDim Preserve
Statement

Placing of
Elements

Figure 35: Code and logic for Best-Fit Algorithm

3. Drawings Methods

```
'Draws the Bins
If i < Bins.GetUpperBound(0) + 1 Then
    'Draws the gradient
    TotalBinHeight = 0
    For j = 0 To Bins(i).GetUpperBound(0)
        TotalBinHeight += Bins(i)(j)
    Next

    TotalPixelBinHeight = CType(TotalBinHeight
        / Me.BinHeight * BinPixelHeight, Integer)

    If TotalPixelBinHeight > 0 Then
        'Draws the Bin gradient
        DrawingBinBrush = New LinearGradientBrush(New Rectangle(X1 + 1, Me.Height - BORDER - TotalPixelBinHeight, BIN_WIDTH - 1, TotalPixelBinHeight), Me.BinColor1, Me.BinColor2, LinearGradientMode.ForwardDiagonal)
        DrawingBinBrush.WrapMode = WrapMode.TileFlipXY
        g.FillRectangle(DrawingBinBrush, New Rectangle(X1 + 1, Me.Height - BORDER - TotalPixelBinHeight, BIN_WIDTH - 1, TotalPixelBinHeight))

        Dim LastY As Integer
        For j = 0 To Bins(i).GetUpperBound(0)
            BinValue += Bins(i)(j)
            Y1 = CType(BinValue / Me.BinHeight * BinPixelHeight, Integer)
            Y1 = Me.Height - Y1 - BORDER
            Y2 = Y1

            If j = 0 Then LastY = Me.Height - BORDER

            'Draws the horizontal lines
            g.DrawLine(DrawingPen, X1, Y1, X2, Y2)

            'Draws the Element value
            Dim TextSize As SizeF = g.MeasureString(Bins(i)(j).ToString, DrawingFont)
            g.DrawString(Bins(i)(j).ToString, DrawingFont, Me.DrawingBinTextBrush, CType(X1 + (BIN_WIDTH / 2) - (TextSize.Width / 2), Integer), CType(Y1 + ((LastY - Y1) / 2) - (TextSize.Height / 2), Integer))
            LastY = Y1
        Next

        'Draws the Bin Number
        g.DrawString((i + 1).ToString, DrawingFont, DrawingTextBrush, CType(X1 + (BIN_WIDTH / 2) - (g.MeasureString((i + 1).ToString, DrawingFont).Width / 2), Integer), Me.Height - BORDER)
    End If
End If
```

Bin Gradient

Bin Draw

Figure 36 : Code Snippet from Draw Bins Sub procedure to draw graph

Drawing methods basically consists of four methods in total

Methods	Use
Private Sub InitDrawingSurface ()	To Create Drawing Surface using graphics libraries
Private Function DrawDemarcations () As Integer	Returns how wide the text was on the left hand side of the graph as well as draws the Demarcations
Private Sub DrawBins()	To actually Draw Bins
Private Sub UpdateGraph()	To call DrawBin() ,DrawDemarcations and InitDrawing Surface in case of new graph

Table 30: List of methods used in drawing the loading graph

4. Events

Developers have provided five events to changing to update graph and re run the algorithm. As elements length, ship length, algorithm and order , all are necessary part of execution, so updateGraph() method is reused in every call .

```

Private Sub cbDecreasing_CheckedChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles cbDecreasing.CheckedChanged
    Calculate()
End Sub

Private Sub nudBinHeight_ValueChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles nudBinHeight.ValueChanged
    Calculate()
End Sub

Private Sub txtElements_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles txtElements.TextChanged
    Calculate()
End Sub

Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
    bLoaded = True
    Calculate()
End Sub

```

Figure 37 : Change events calling compute method

**Events of Value Changes and other
in the application ,that makes
Algorithm re-run**

7.4 Technical Quality

7.4.1 Memory Utilization

<pre>Class Population Inherits List(Of Tour) ... </pre>	<pre>Imports System.Collections Imports System.Collections.Generic</pre>
---	--

Table 31 : Use of List and Inheritance for memory utilization

List represents a strongly typed list of objects that can be accessed by index. Provides methods to search, sort, and manipulate lists. For utilizing less memory the developer has used ArrayList as it utilizes the memory as required(Dynamically) instead of array that uses fix amount of memory whether that is usable or not.

7.4.2 Performance

The application performance is measured in terms of the speed, and time taken to solve the problem, generally termed as the time complexity of algorithm. For upto 100 cities in the tour , application takes less than 30 seconds to approach to the best possible solution in TSP. In loading the output is instant solves and generated graph under a second.

7.4.3 User Interface

- ✚ User Interface is based on the Human Computer Interaction and Usability Concepts.
- ✚ Metaphors are used to increase the usability and understanding.
- ✚ Graph are made is very decent way that show each path and route.
- ✚ Zoom capability, Co-ordinates can be used to view it in proper way.

7.4.4 Short-Cut Keys

For Help F1 , Exit – Alt+f4 key and other set of keys are provided to use so that Beacon can be used as frequently and effectively as user want.

7.4.5 Naming Convention

Naming Convention for Class	Naming Convention for Variables
<pre>Public Class Cities Inherits List(Of City)</pre>	<pre>Private bmpGraph As Bitmap Private DrawingTextBrush As New SolidBrush(Me.ForeColor)</pre>
Starts with Capital letter	Bmp for bitmap graph

Table 32: Showing naming convention used in code

7.4.6 Comments

Relevant Comments are provided for better understanding of codes and flow within the code

```
''' <summary>
    ''' Create the initial set of random tours.
    ''' </summary>
    ''' <param name="populationSize">Number of tours to create.</param>
    ''' <param name="cityList">The list of cities in this tour.</param>
    ''' <param name="rand">Random number generator. We pass around the same random
number generator, so that results between runs are consistent.</param>
    ''' <param name="chanceToUseCloseCity">The odds (out of 100) that a city that
is known to be close will be used in any given link.</param>
```

Table 33: Showing the used of xml comments to describe code

7.4.7 Validations

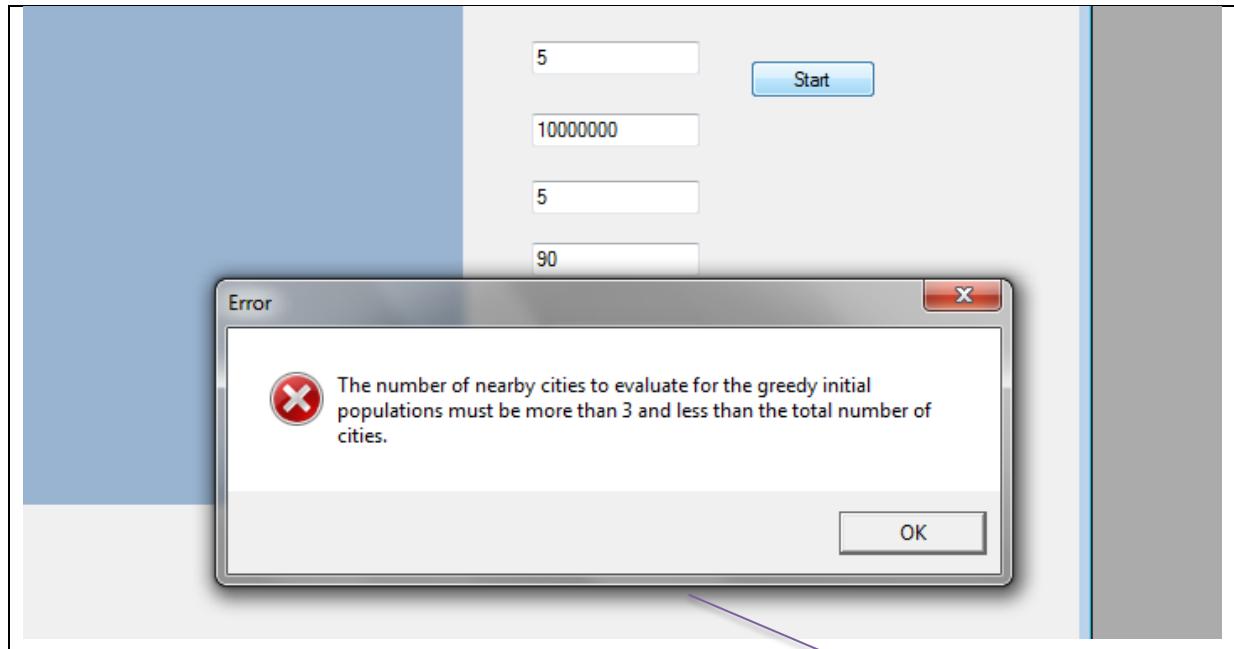
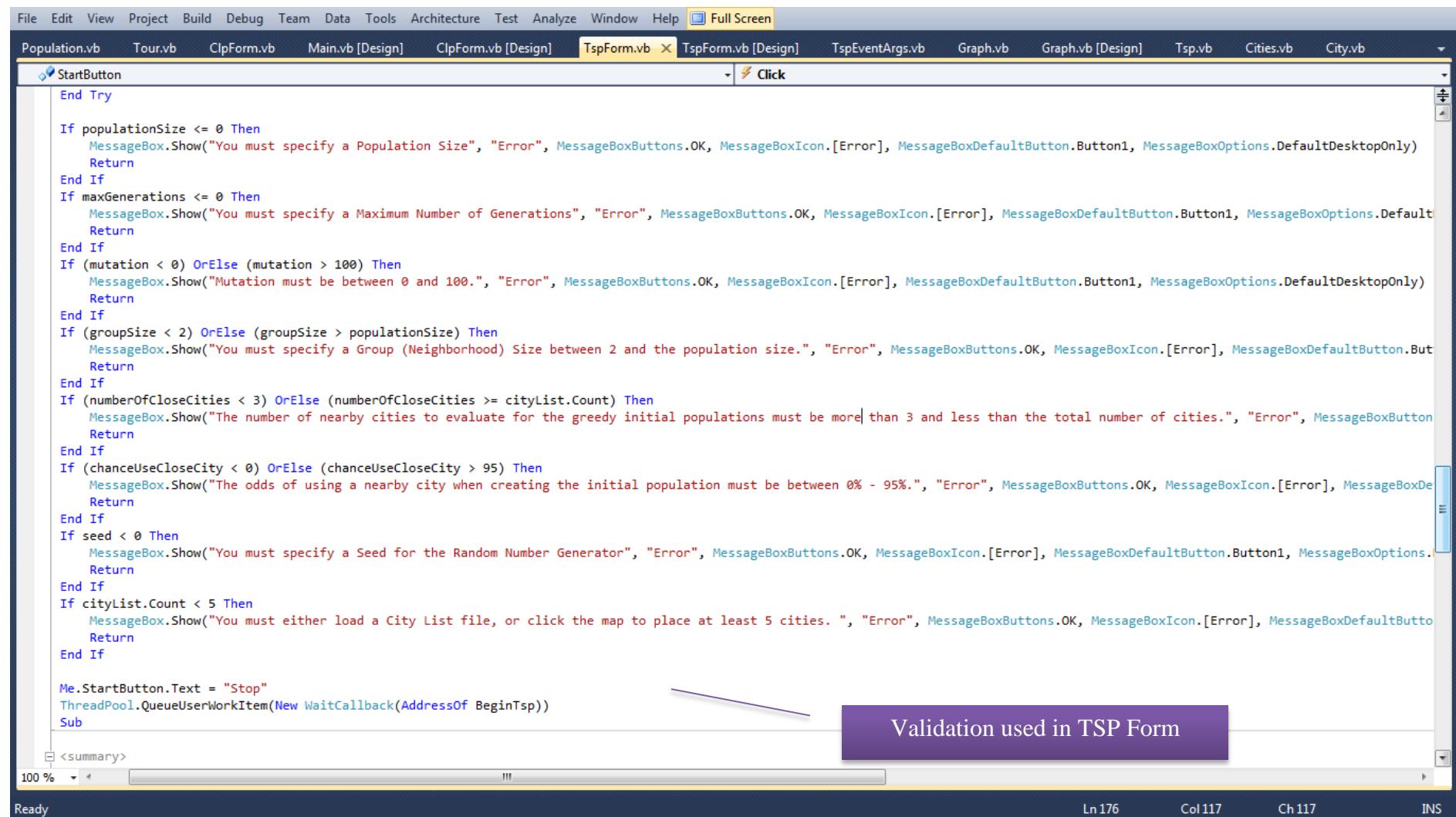


Figure 38 : Error Message because of less number of cities

Error Message ,

If less than 3 cities



The screenshot shows the Microsoft Visual Studio IDE with the TspForm.vb [Design] tab selected in the toolbar. The code editor displays a Try...End Try block containing validation logic. A callout bubble points to the validation code with the text "Validation used in TSP Form".

```
End Try

If populationSize <= 0 Then
    MessageBox.Show("You must specify a Population Size", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error, MessageBoxDefaultButton.Button1, MessageBoxOptions.DefaultDesktopOnly)
    Return
End If
If maxGenerations <= 0 Then
    MessageBox.Show("You must specify a Maximum Number of Generations", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error, MessageBoxDefaultButton.Button1, MessageBoxOptions.DefaultDesktopOnly)
    Return
End If
If (mutation < 0) OrElse (mutation > 100) Then
    MessageBox.Show("Mutation must be between 0 and 100.", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error, MessageBoxDefaultButton.Button1, MessageBoxOptions.DefaultDesktopOnly)
    Return
End If
If (groupSize < 2) OrElse (groupSize > populationSize) Then
    MessageBox.Show("You must specify a Group (Neighborhood) Size between 2 and the population size.", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error, MessageBoxDefaultButton.Button1, MessageBoxOptions.DefaultDesktopOnly)
    Return
End If
If (numberOfCloseCities < 3) OrElse (numberOfCloseCities >= cityList.Count) Then
    MessageBox.Show("The number of nearby cities to evaluate for the greedy initial populations must be more than 3 and less than the total number of cities.", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error, MessageBoxDefaultButton.Button1, MessageBoxOptions.DefaultDesktopOnly)
    Return
End If
If (chanceUseCloseCity < 0) OrElse (chanceUseCloseCity > 95) Then
    MessageBox.Show("The odds of using a nearby city when creating the initial population must be between 0% - 95%.", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error, MessageBoxDefaultButton.Button1, MessageBoxOptions.DefaultDesktopOnly)
    Return
End If
If seed < 0 Then
    MessageBox.Show("You must specify a Seed for the Random Number Generator", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error, MessageBoxDefaultButton.Button1, MessageBoxOptions.DefaultDesktopOnly)
    Return
End If
If cityList.Count < 5 Then
    MessageBox.Show("You must either load a City List file, or click the map to place at least 5 cities. ", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error, MessageBoxDefaultButton.Button1, MessageBoxOptions.DefaultDesktopOnly)
    Return
End If

Me.StartButton.Text = "Stop"
ThreadPool.QueueUserWorkItem(New WaitCallback(AddressOf BeginTsp))
Sub
```

Validation used in TSP Form

```

''' <returns>True if the connection can be made.</returns>
Private Shared Function testConnectionValid(ByVal tour As Tour, ByVal cityList As Cities, ByVal cityUsage As Integer(),
                                         ByVal city1 As Integer, ByVal city2 As Integer) As Boolean
    ' Quick check to see if cities already connected or if either already has 2 links
    If (city1 = city2) OrElse (cityUsage(city1) = 2) OrElse (cityUsage(city2) = 2) Then
        Return False
    End If

    ' A quick check to save CPU. If haven't been to either city, connection must be valid.
    If (cityUsage(city1) = 0) OrElse (cityUsage(city2) = 0) Then
        Return True
    End If

    ' Have to see if the cities are connected by going in each direction.
    For direction As Integer = 0 To 1
        Dim lastCity As Integer = city1
        Dim currentCity As Integer
        If direction = 0 Then
            ' on first pass, use the first connection
            currentCity = tour(city1).Connection1
        Else
            ' on second pass, use the other connection
            currentCity = tour(city1).Connection2
        End If
        Dim tourLength As Integer = 0
        While (currentCity <> -1) AndAlso (currentCity <> city2) AndAlso (tourLength < cityList.Count - 2)
            tourLength += 1
            ' figure out if the next city in the list is [0] or [1]
            If lastCity <> tour(currentCity).Connection1 Then
                lastCity = currentCity
                currentCity = tour(currentCity).Connection1
            Else
                lastCity = currentCity
                currentCity = tour(currentCity).Connection2
            End If
        End While

        ' if cities are connected, but it goes through every city in the list, then OK to join.
        If tourLength >= cityList.Count - 2 Then
            Return True
        End If

        ' if the cities are connected without going through all the cities, it is NOT OK to join.
        If currentCity = city2 Then
            Return False
        End If
    Next

    ' if cities weren't connected going in either direction, we are OK to join them
    Return True
End Function

```

Test Connection
Valid Checks only
the valid
connections

Another Checking

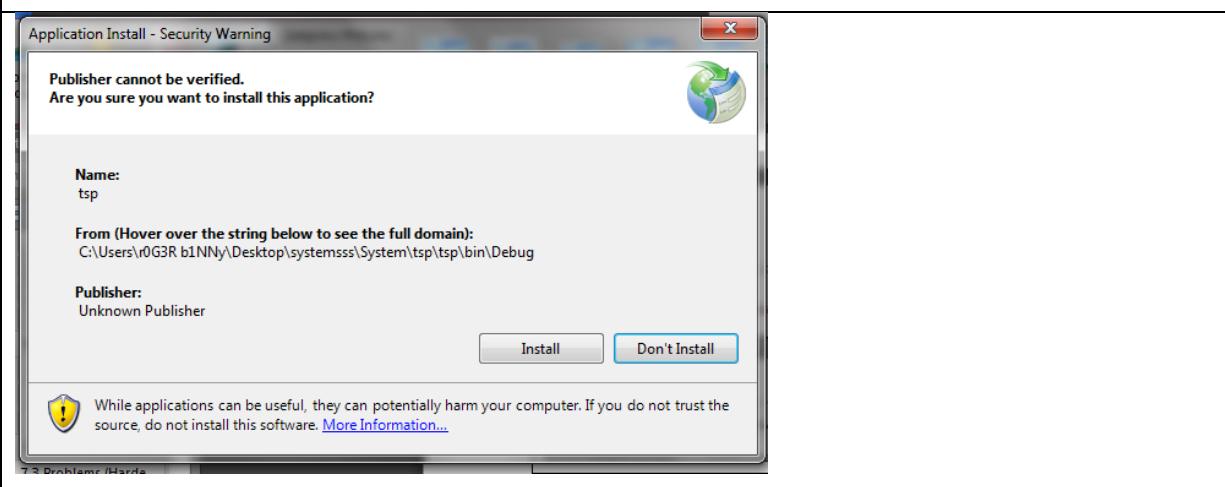
Figure 40 : testConnectionValid function determines the validity of link between cities

7.4.8 User Manual

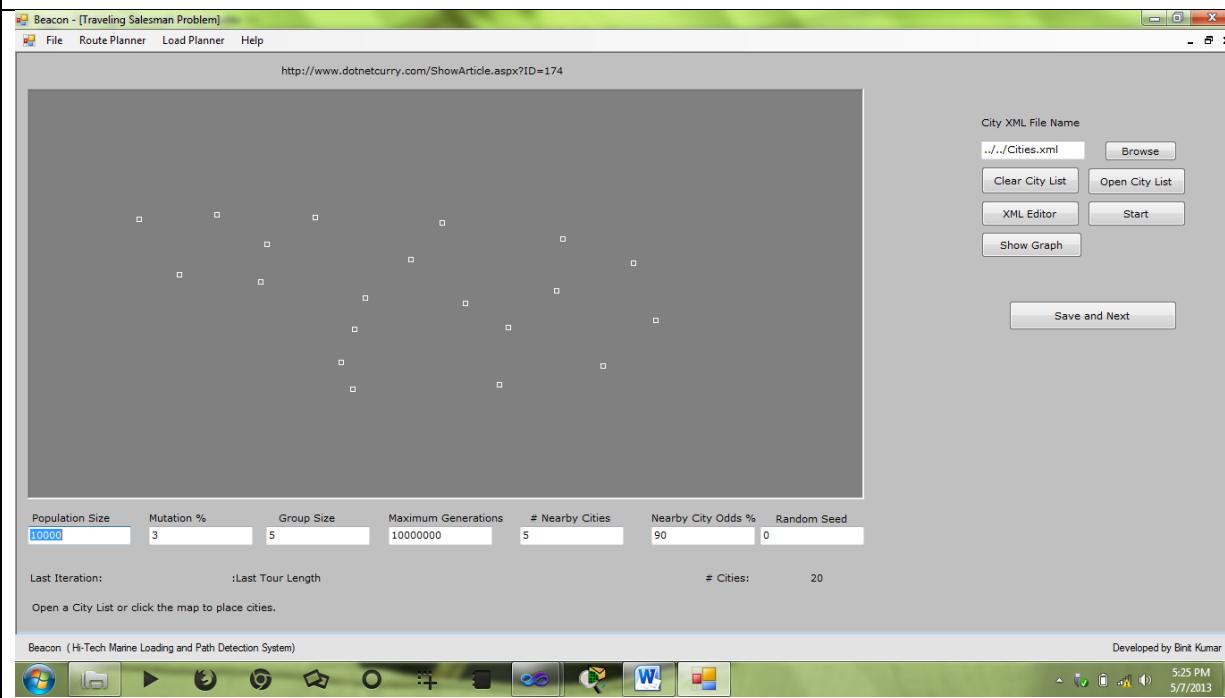
Welcome to the User Manual of Beacon (Hi –Tech Marine Loading and Path Detection System). Run this windows based application in any windows platform by downloading or getting it by other ways. Screen shots are provided below to understand the use of application



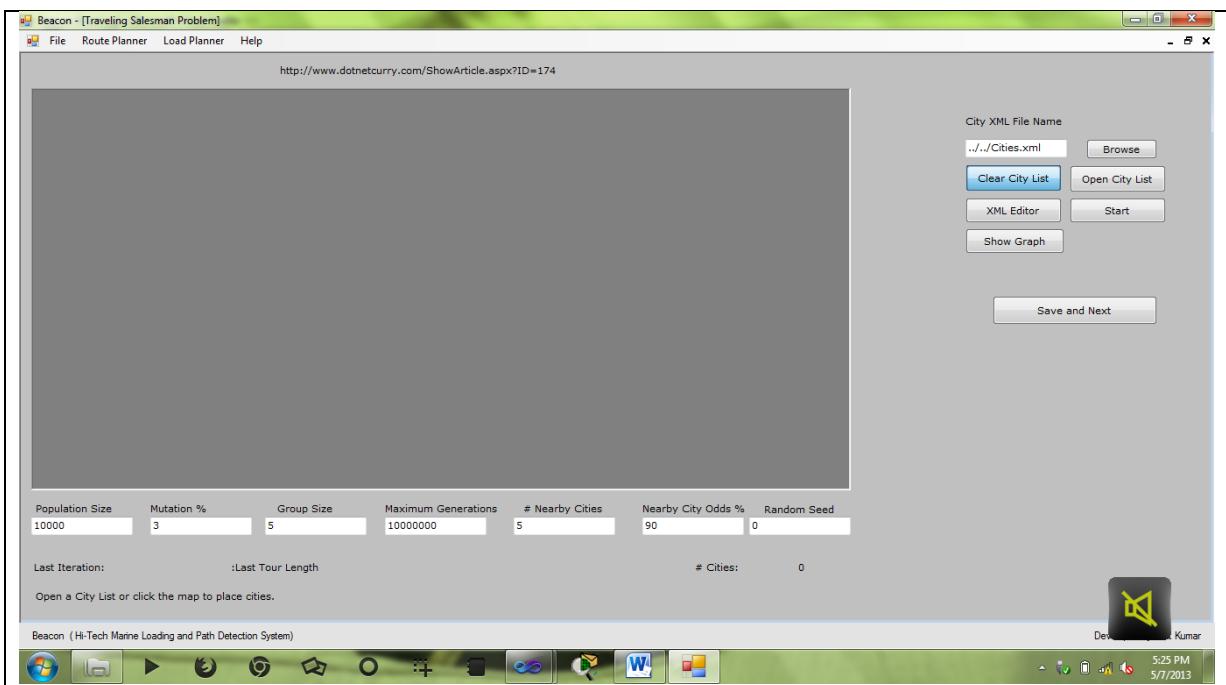
Step 1: Click on the Setup file to start the application



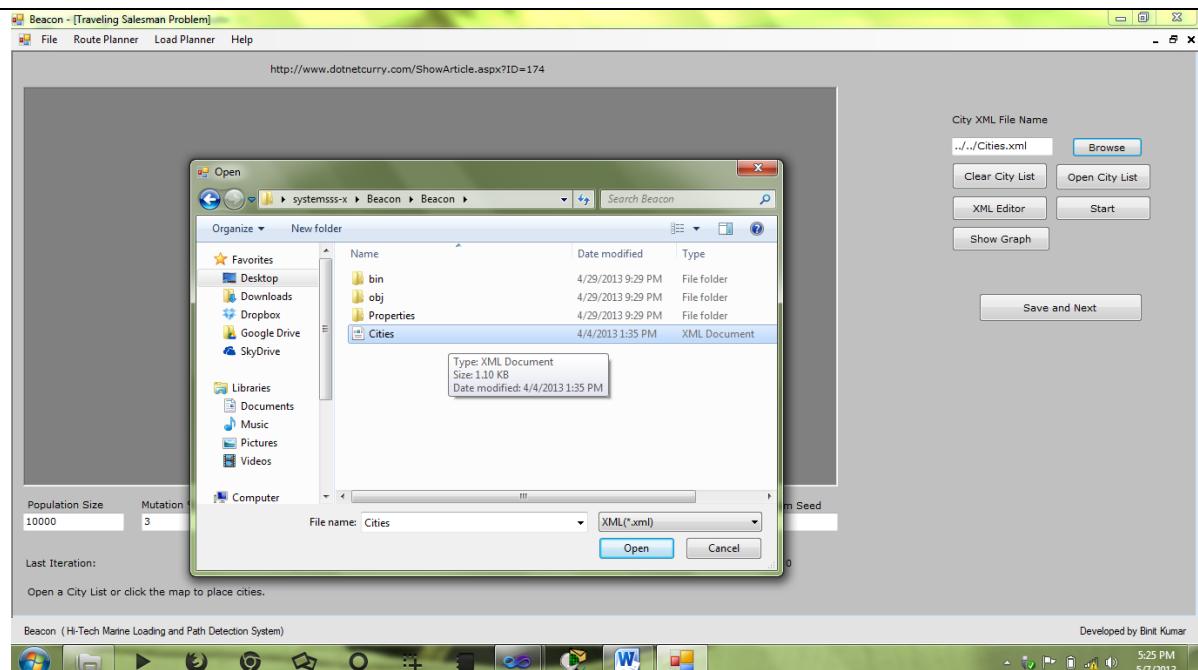
Step 2: Install The Beacon



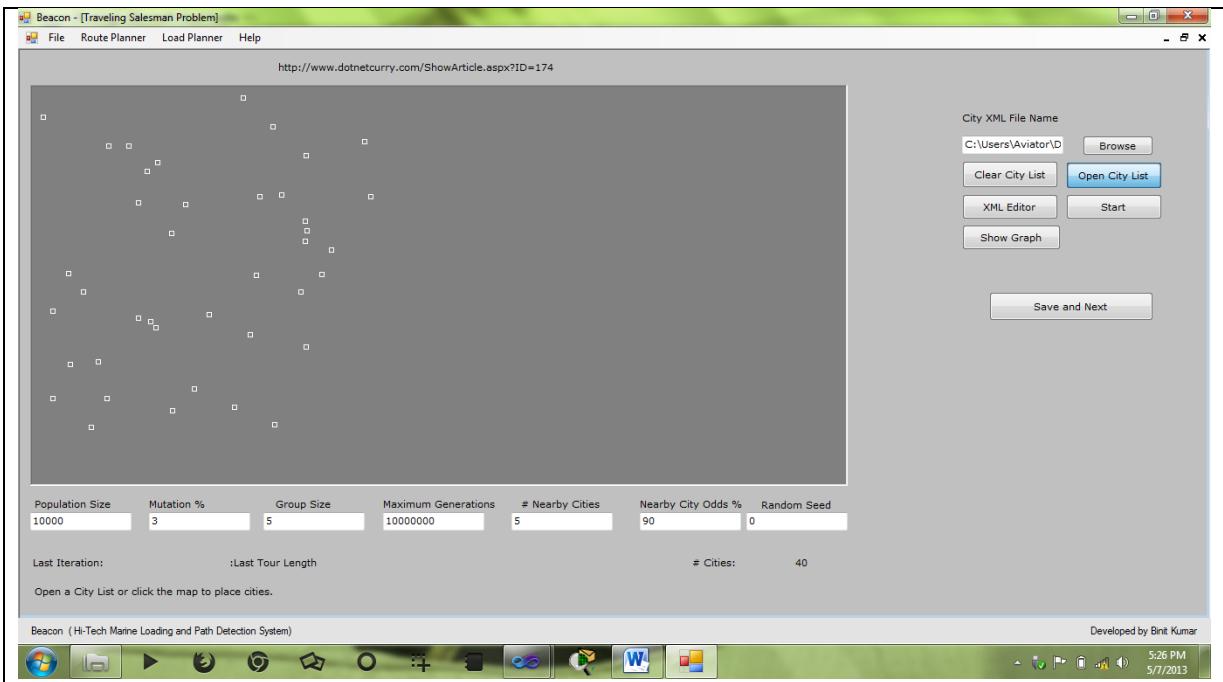
Step3: Mouse Down to create cities



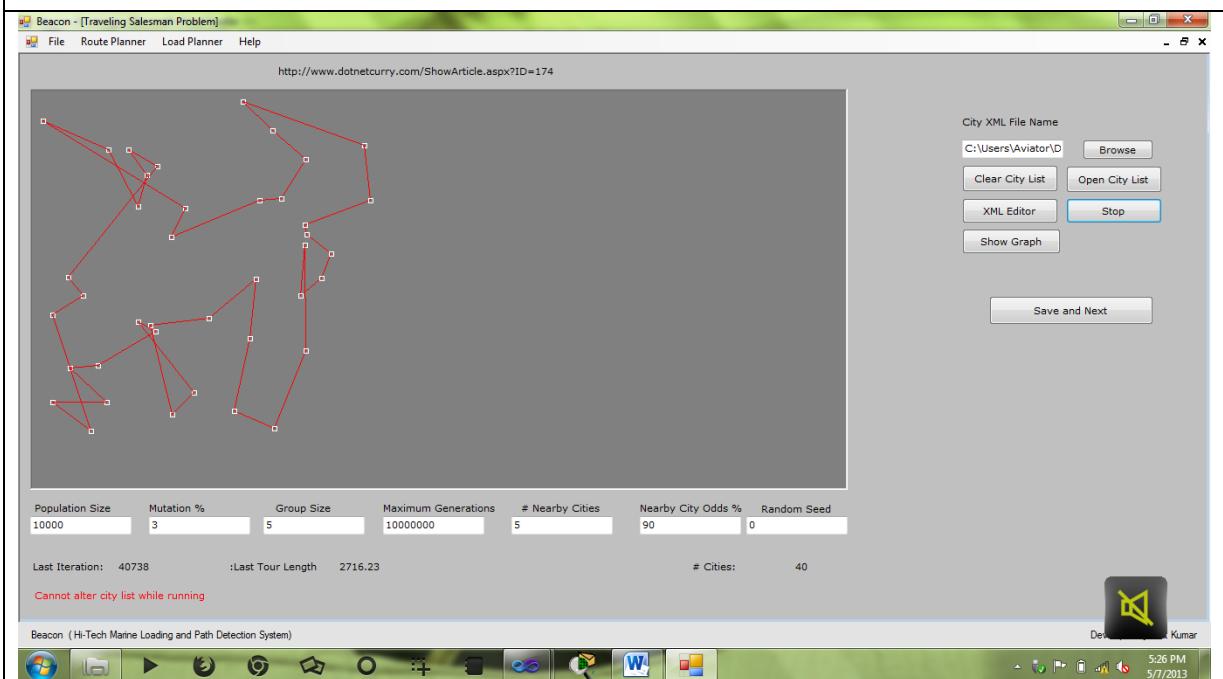
Step4: Press Clear City list to clear city



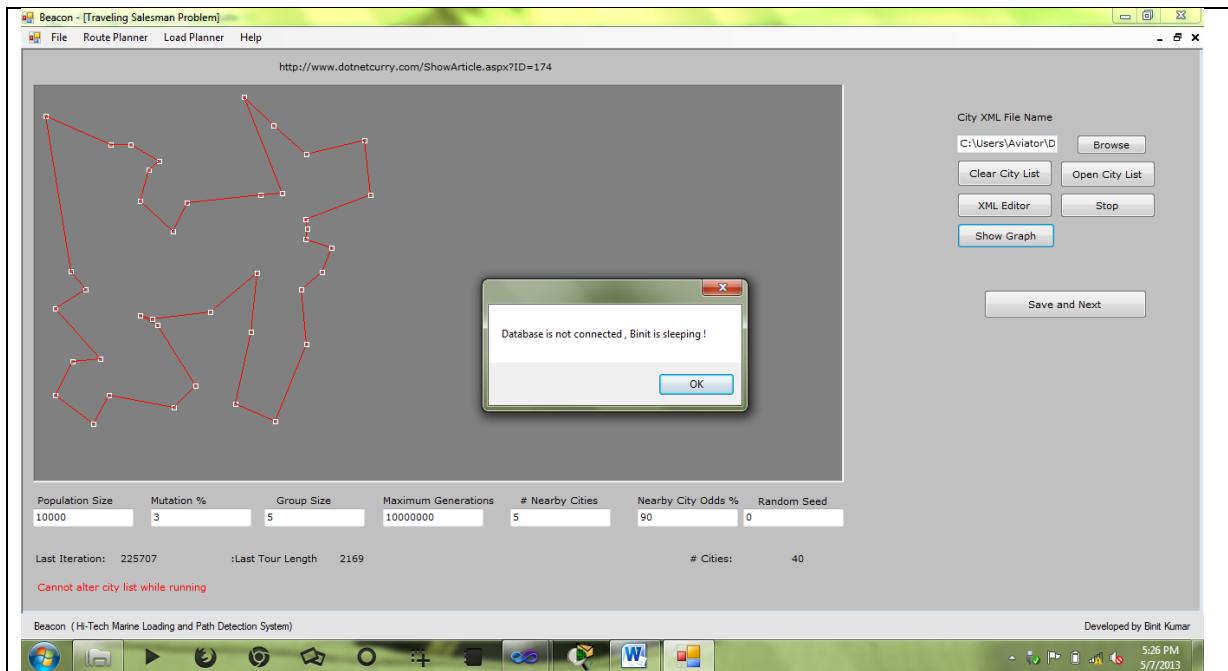
Step 5: Browse City by xml file



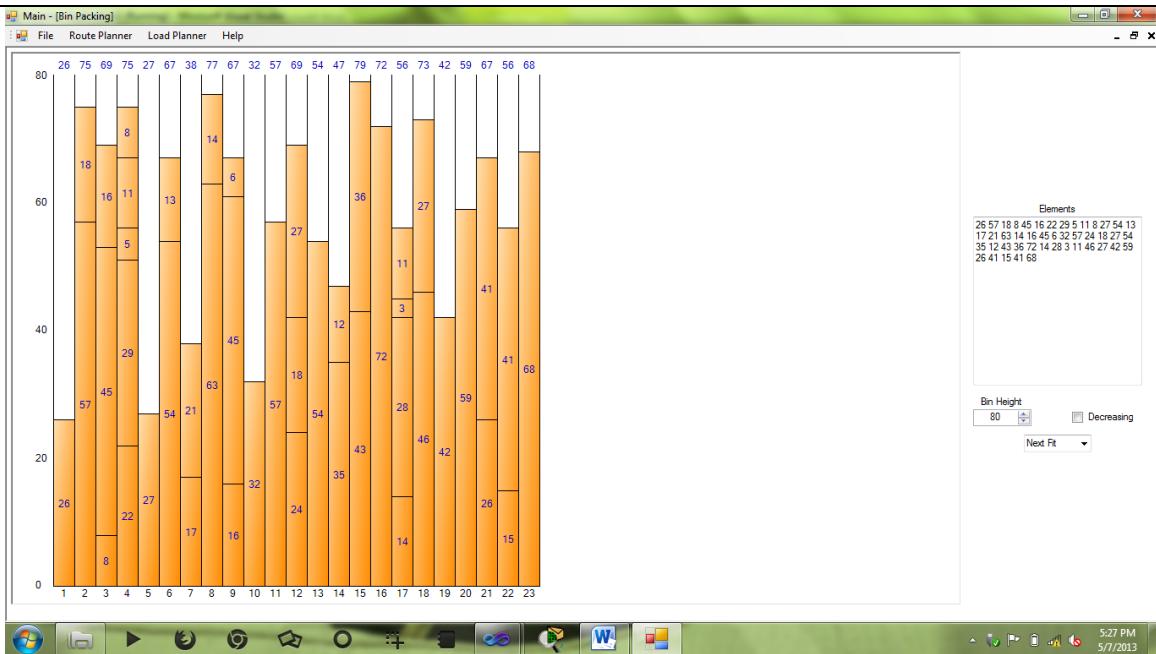
Step 6: Click on Open City List to open xml file



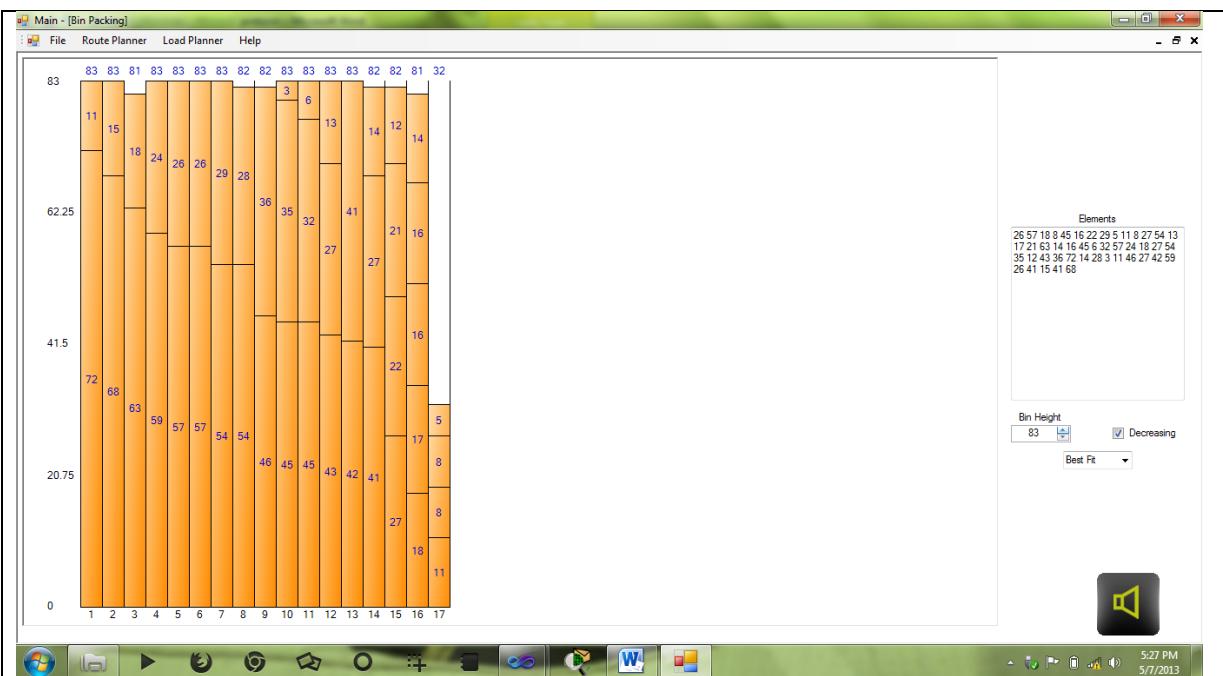
Step 7 : Press Start Button to start the Tsp Algorithm



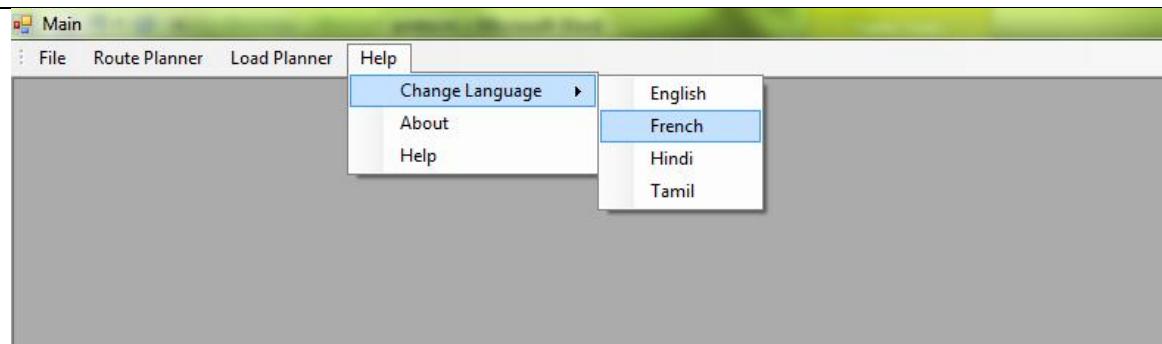
Step 8 : Error Message when Database is not Connected for Saving



Step9 : Default Elements and Graph at the press



Step 10: Best Fit Algorithm is chosen with Decreasing order

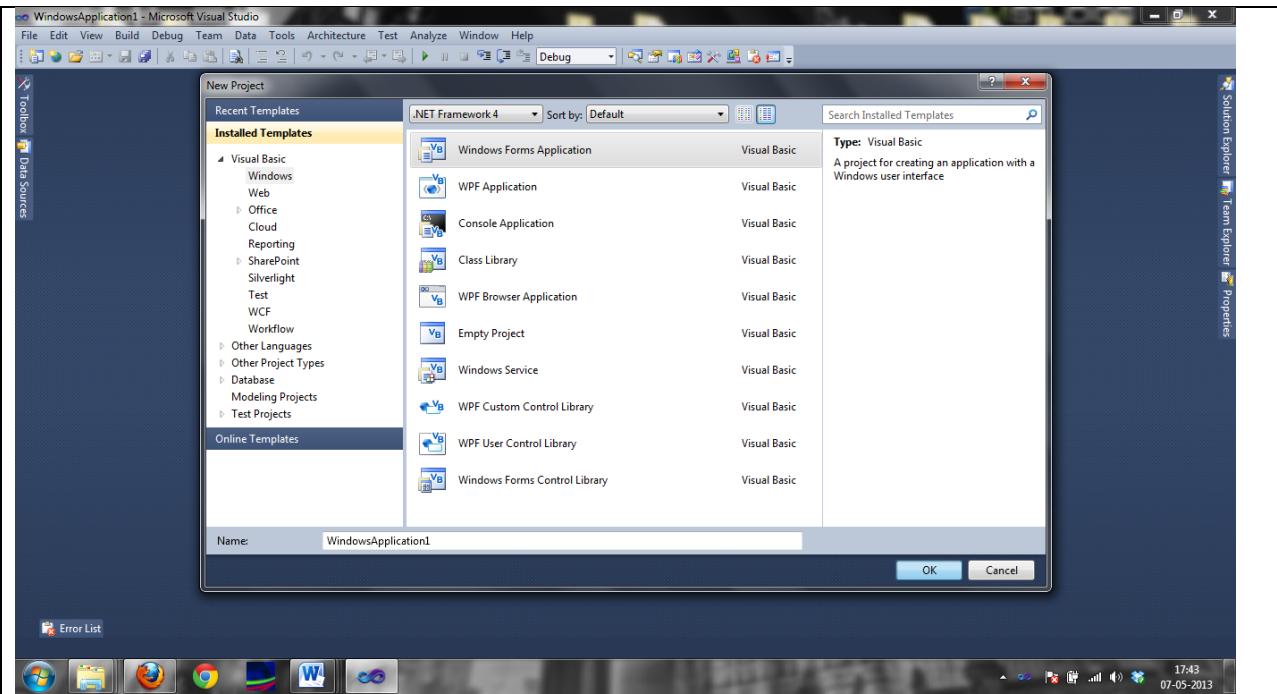


Step 11 : Change Language feature in the menu

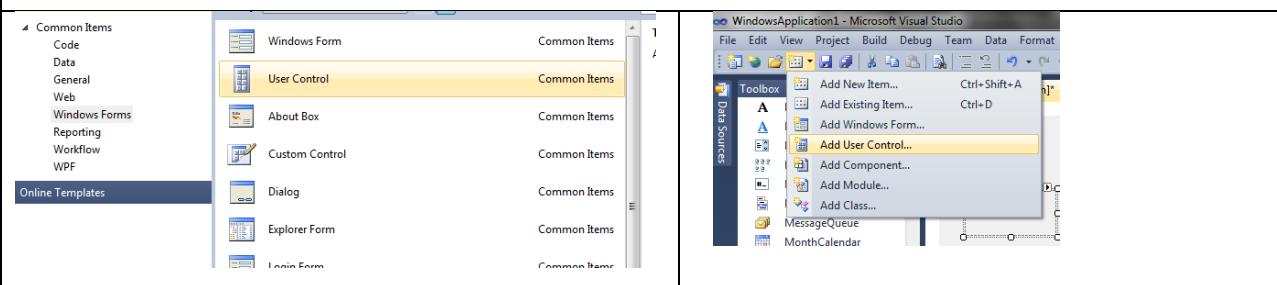
Step 12 : Download TSP using and by selecting format of download

Table 34 : User Manual Table

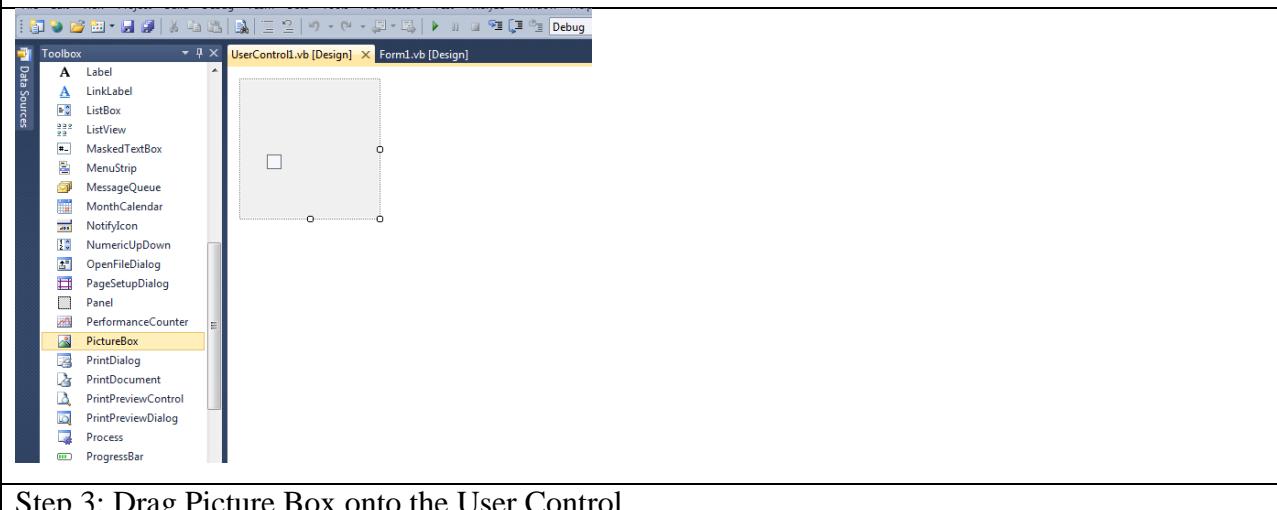
7.4.9 Technical Manual



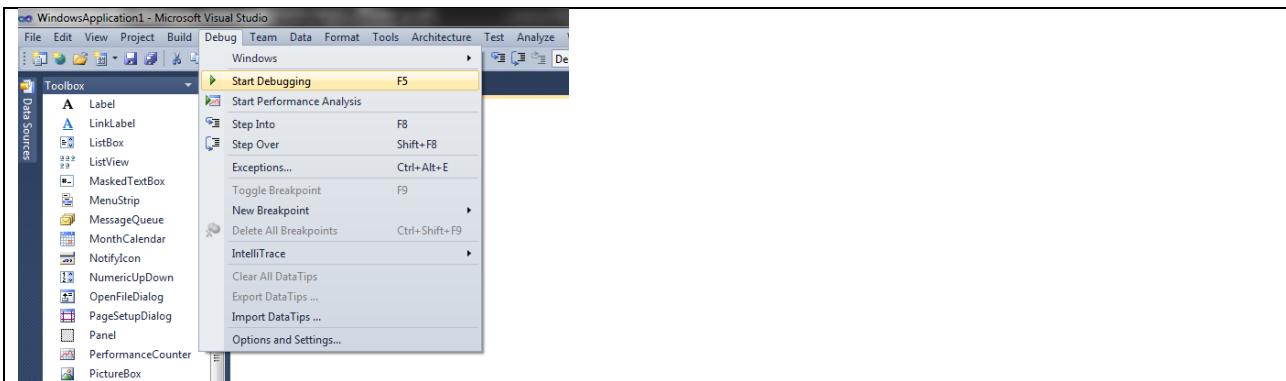
Step 1 : Go to file → New Project → Choose Visual Basic Windows → .NET Framework



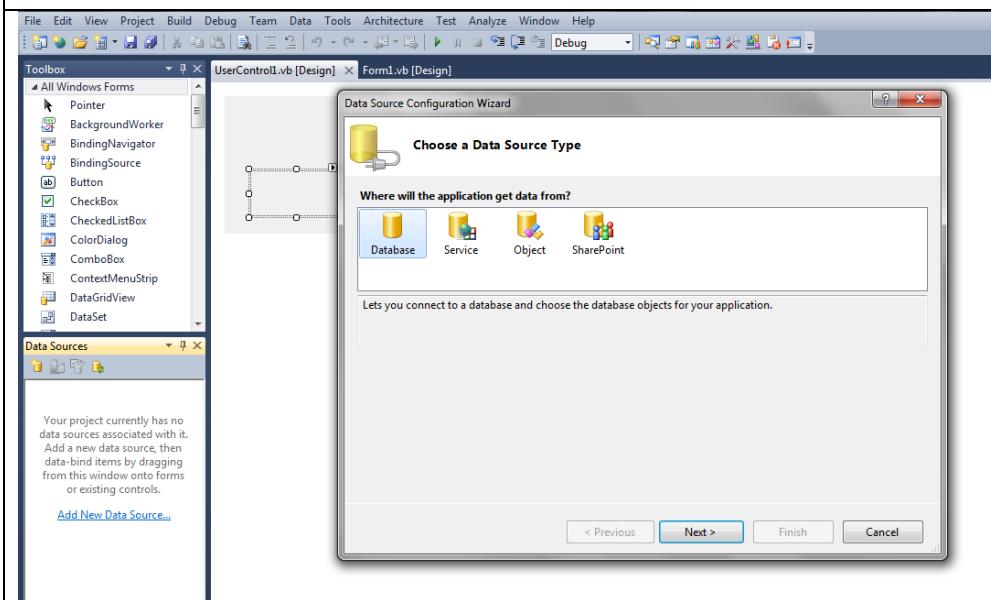
Step 2. Find User Control in the new menu



Step 3: Drag Picture Box onto the User Control



Step4 : Go to Debug and then Start Debugging to debug or press shortcut F5.



Step 5 : Data Source Configuration Wizard to connect to any database

Table 35: Technical Manual Table

Chapter 8 Testing

"... we have as many testers as we have developers. And testers spend all their time testing, and developers spend half their time testing. We're more of a testing, a quality software organization than we're a software organization." – Bill Gates

(Information Week 2002)

Testing is the process of exercising software with the intent of finding and correcting errors. The objective of the testing is to uncover different classes of errors and to do so with a minimum amount of time and effort. In order to provide highly acceptable and error free system, the system should have to face the testing procedure and evaluation of each and every module and functionality.

Testing is the process of executing a program with the intent of finding errors. Once the system is developed, testing will be performed according to test plan. The result of the testing is used for enhancement and correction of the system in the next iteration.

The highlights of this chapter include:

- The different types of testing done.
- The duration taken to test.
- The person(s) involved.
- The reported errors.
- The measures taken.

8.1 Test Plan

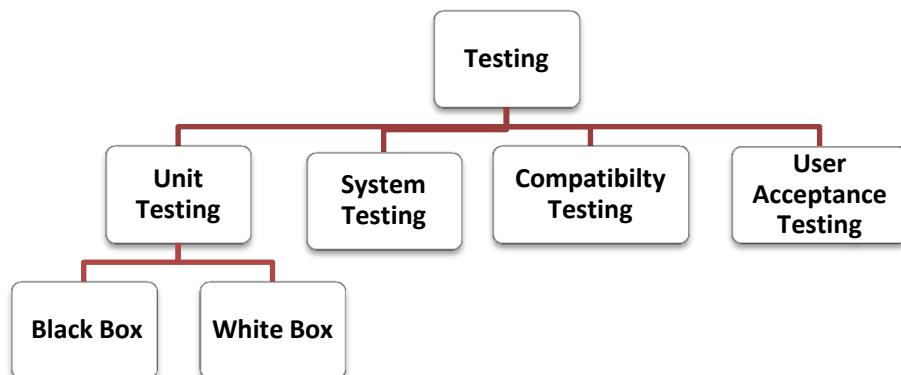


Figure 41 : Type of Testing

No.	Method	Details
1.	Unit Testing	In this testing activity the developer tested each functionality separately in an isolated manner from the rest of the application
2.	System Testing	Here the developer performed testing on the complete, integrated system. All modules were combined and tested together to evaluate the system's compliance with the specified requirements.
3.	Compatibility Testing	Developer has tested the application in different windows platform.
4.	User Acceptance Testing	The system was given to a user and he was asked to test the system to check whether the system fulfills the requirement.
Assumptions: The Complete Testing process is performed by Developer.		

Table 36: Type of testing involved

8.2 Test Duration

Test Strategy	Start Date	End Date
Unit Testing	01 April 2013	12 th April 2013
System Testing	12 th April 2013	19th April 2013
Compatibility Testing	22 April 2013	23 rd April 2013
User Acceptance Testing	23th April 2013	25 th April 2013

Table 37 : Test Duration Table

8.3 Unit Testing

In order to test each small part of the developed system individual test cases were developed. Unit testing was very helpful as it helped the developer to test individual units of source code.

Two Flavours of unit testing was basically conducted, namely black box (also called functional testing) and white box (also called logical testing).

8.3.1 Black Box Testing

Project Title	Beacon 1.0	Testing Date	01-04-13
Test Name	Beacon Design Route Map		
Test ID	Beacon-UT-BlackBox-1		
Conducted By	Binit Kumar		
Description	Validates the Design and Process of Designing		

Beacon-UT1 -Test Case #1		
Function to Test:	City Design on picture Box	
Execution Steps:	On the TspForm picture box mouse down event Or On the TspForm multiple mouse down event Or On the TspForm mouse down when graph checkbox is on	
Expected Result	Actual Result	Status(Pass/Fail)
Small Rectangle of white colour should appear on the map	Rectangles (Cities) are created in the map	Pass
Remarks	Nil	

Beacon-UT1 -Test Case #2		
Function to Test:	Xml file loading in picture box	
Execution Steps:	On browsing xml file Or On opening XML file containing the city co-ordinates Or On reading wrong XML file	
Expected Result	Actual Result	Status(Pass/Fail)
Small Rectangle of white colour should appear on the map	Rectangles are created as per the co-ordinates given in the xml file	Pass
Remarks	The developer has omitted the error that comes , when wrong xml file gets load and no cities are created, only file that contains proper co-ordinates in (x, y) fashion can be read	

Beacon-UT1 -Test Case #3		
Function to Test:	Clear and graph	
Execution Steps:	On clearing the city Or On showing co-ordinate graph to the user when graph checkbox is pressed	
Expected Result	Actual Result	Status(Pass/Fail)
City in the map are cleared on the clear press, and co-ordinate Graph is created on graph check	Cities are cleared and graph is also coming in the proper place	Pass
Remarks	Nil	

Conclusion	
Errors Detected:	1
Measures Taken:	The code was modified as required and the expected result was achieved.

Project Title	Beacon 1.0	Testing Date	02-04-13
Test Name	Beacon Plan Route		
Test ID	Beacon-UT-BlackBox-2		
Conducted By	Binit Kumar		
Description	Check the planning and execution of TSP by using route design, and also check for proper route publishing on the map		

Beacon-UT2 -Test Case #1		
Function to Test:	Initial Constraints default value and edit	
Execution Steps:	On Changing population size textbox value Or On Changing mutation % textbox value Or On Changing group size textbox value Or On Changing max generation textbox value Or On Changing no of close cities textbox value Or On Changing random seed textbox value	
Expected Result	Actual Result	Status(Pass/Fail)
Initial values should be loaded in the variable before begin the tsp() function	Algorithm is showing different output when the initial values are changed	Pass
Remarks	Initial values like population size, mutation ,close cities are kept default but output can vary vastly and positively When values are updated as per the scenario. Values are checked by using message box during testing.	

Beacon-UT2 -Test Case #2		
Function to Test:	Beginning algorithms	
Execution Steps:	cost function, cost variables and fitness function check And Generating initial population in population class And Saving Cost of chromosomes (tours) in array And Crossover in the Tour class between the tours And Mutation to move the execution for any local extrema And Convergence check	
Expected Result	Actual Result	Status(Pass/Fail)
Algorithm provides the shorter path with every iteration of the algorithm	Successful shortest distance are computed with every iteration but some time <i>index was out of range</i> error is coming	Fail
Remarks	Developer has solved the error as error cannot be non-negative and also it can't be smaller than the actual size of collection.	

Beacon-UT2 -Test Case #3		
Function to Test:	Print route in map with every updated shortest tour	
Execution Steps:	On new tour discovery with shortest path in the iteration	
Expected Result	Actual Result	Status(Pass/Fail)
Tour in the map should be updated with every new shortest path finding.	New tours are updated in the picture box, but color chosen is not suitable as per HCI	Pass
Remarks	Colors are changes according to HCI, fast execution of Algorithm and updated tours are changing within seconds,	

Conclusion	
Errors Detected:	1
Measures Taken:	The code was modified as required and the expected result was achieved.

Project Title	Beacon 1.0	Testing Date	03-04-13
Test Name	Beacon XML read/ write		
Test ID	Beacon-UT-BlackBox-3		
Conducted By	Binit Kumar		
Description	Checks the read/ write activity of the application into the xml files.		

Beacon-UT3 -Test Case #1		
Function to Test:	Reading of city xml file	
Execution Steps:	On Reading of city xml file to load the city list in the graph	
Expected Result	Actual Result	Status(Pass/Fail)
Successful reading of the city.xml file in the array list	City are saved in the array from the chosen files	Pass
Remarks	Nil	

Beacon-UT3 -Test Case #2		
Function to Test:	Writing the co-ordinates of city in the file	
Execution Steps:	On using xml writer to save the city into the xml file	
Expected Result	Actual Result	Status(Pass/Fail)
Successful writing in the file	Successful format of saving.	Pass
Remarks	Nil	

Conclusion		
Errors Detected:	0	
Measures Taken:	The Xml editor is updated with the interface for proper usabilty	

Project Title	Beacon 1.0	Testing Date	05-04-13
Test Name	Beacon Manage Container Loading		
Test ID	Beacon-UT-BlackBox-4		
Conducted By	Binit Kumar		
Description	Checks the initial values and loading of the containers		

Beacon-UT4 -Test Case #1		
Function to Test:	Get Elements length in the multi valued textbox	
Execution Steps:	On getting the length in the text box ClpForm Or On getting more than one value in the text box ClpForm Or On having text in wrong order in the text box in ClpForm	
Expected Result	Actual Result	Status(Pass/Fail)
Successful reading of the text values and array should be updated with every new value	Array are not getting updated when new values are provided and error coming index out of range	Fail
Remarks	Redim of the Bin () () is used to change the array size, Redim is a exclusive feature in the VB and used very effectively to re define new bin for updated elements.	

Beacon-UT4 -Test Case #2		
Function to Test:	Set the ship length (Box Height)	
Execution Steps:	On incrementing /decrementing the default length of ship i.e. 80 ft	
Expected Result	Actual Result	Status(Pass/Fail)
Ship length is changing and algorithm is taking the new parameter to solve	Ship length is updated as bin height And compute() function is called in The selectindex event	Pass
Remarks	Nil	

Beacon-UT4 -Test Case #3		
Function to Test:	Next Fit, First Fit, Best Fit, Worst Fit	
Execution Steps:	On Choosing Next fit, Or on Choosing First Fit, Or on Choosing Best Fit, Or on Choosing Worst Fit.	
Expected Result	Actual Result	Status(Pass/Fail)
Next Fit – when bin 1 gets full go for bin2 and never checks back First Fit– Checks all the elements and place it on the first bin it can be Best Fit- Choose the bin with least amount of free space Worst Fit- Choose the bin with most amount of free space	The outputs are coming in the proper Order, All modules working correctly Next Fit, Best Fit, Worst Fit, Best Fit.	Pass
Remarks	Developer have used ReDim Preserve statement many Times, the better approach is to solve would be to make each Bin Array (the array that holds the Elements) as big as the Bin Height right at the declaration and then just chop off the unused (0) elements after the algorithm is done.	

Beacon-UT4 -Test Case #4		
Function to Test:	Decreasing order	
Execution Steps:	Before applying the algorithm to the elements sort it into decreasing order of array	
Expected Result	Actual Result	Status(Pass/Fail)
More Efficient packing is done and large elements are placed in the first	More better packing is coming with All the algorithm	Pass
Remarks	Nil	

Conclusion	
Errors Detected:	1
Measures Taken:	The code was modified as required and the expected result was achieved.

Project Title	Beacon 1.0	Testing Date	08-04-13
Test Name	Beacon View Loading		
Test ID	Beacon-UT-BlackBox-5		
Conducted By	Binit Kumar		
Description	Graph generation of the bins and box during the change event.		

Beacon-UT5 -Test Case #1		
Function to Test:	Draw Demarcation , how wide the graph is on the left side of the graph	
Execution Steps:	Draw 100% , 75%, 50%, 25% ,0% And If there decimals, width 2 will be longer Else width 1 will be longer Return max width for use	
Expected Result	Actual Result	Status(Pass/Fail)
Left hand side vertical demarcation is shown	Line is getting updated and changed When the elements are shown with Higher elements length	Pass
Remarks	Nil	

Beacon-UT5 -Test Case #2		
Function to Test:	Draw Ships and Bins	
Execution Steps:	Draw the vertical line that makes up bins Draw the gradient Draw the Bin gradient Draw the horizontal line with value Draw the element values Draw the Bin Number Draw the Bin Count	
Expected Result	Actual Result	Status(Pass/Fail)
All the Bins are shown with Gradient, Number, Values, Count and perfectly placed lines.	Wrong graph is coming when the New algorithm is chosen, Horizontal line is not coming Bin Count is not coming properly	Fail
Remarks	<pre>TotalPixelBinHeight = CType(TotalBinHeight \ Me.BinHeight * BinPixelHeight, Integer)</pre> Total Pixel Bin Height was getting a wrong value	

Beacon-UT5 -Test Case #3		
Function to Test:	Initialize Drawing Surface	
Execution Steps:	Create new bmpgraph as bitmap Draw graphics Set Size of bmp Graph same as picturebox Refresh picture box during update graph call	
Expected Result	Actual Result	Status(Pass/Fail)
User control is well created and loaded during execution	Bitmap graph and graphics are working correctly	Pass
Remarks	Nil	

Conclusion	
Errors Detected:	1
Measures Taken:	The code was modified as required and the expected result was achieved.

Project Title	Beacon 1.0	Testing Date	09-04-13
Test Name	Beacon Change Language		
Test ID	Beacon-UT-BlackBox-6		
Conducted By	Binit Kumar		
Description	Language Selected and applied values		

Beacon-UT6 -Test Case #1		
Function to Test:	Change Langue Function	
Execution Steps:	Set Language Property Call Component resource Manager	
Expected Result	Actual Result	Status(Pass/Fail)
Show the Language Selected	Showing the language Selected	Pass
Remarks	Nil	

Conclusion	
Errors Detected:	0
Measures Taken:	Working fine , mdi child forms are also updated

8.3.2 White Box Testing

Project Title	Beacon 1.0	Testing Date	10-04-13
Test Name	Beacon TSP Event Handler		
Test ID	Beacon-UT-White Box -1		
Conducted By	Binit Kumar		
Description	TSP Route Map and Algorithm with Event handler		

M-Admin-UT7 -Test Case #1		
Function to Test:	Actual Result	Status(Pass/Fail)
Every Tour should be mapped in the picture box with every iteration of the TSP and best fittest value	Tours are not coming in the picture box as the algorithm is finding the best suitable value	Fail
Code Errors Identified:	After reviewing the code, the developer have seen that when TspEventArgs class have all the getter and setter value that picture box required and it also inherited the EventArgs class from library System.Drawing. But the error is that after beginning the Algorithm , no Add Handler is added to the function to update the tour diagram.	
Tested Code:		
<pre>tsp = New Tsp() ' AddHandler Obj.Ev_Event, AddressOf EventHandler AddHandler tsp.foundNewBestTour, AddressOf tsp_foundNewBestTour ' tsp.foundNewBestTour += New Tsp.NewBestTourEventHandler(tsp_foundNewBestTour) tsp.Begin(populationSize, maxGenerations, groupSize, mutation, seed, chanceUseCloseCity, _ cityList)</pre>		
Rectified Code:		
<pre>tsp = New Tsp() ' AddHandler Obj.Ev_Event, AddressOf EventHandler AddHandler tsp.foundNewBestTour, AddressOf tsp_foundNewBestTour ' tsp.foundNewBestTour += New Tsp.NewBestTourEventHandler(tsp_foundNewBestTour) tsp.Begin(populationSize, maxGenerations, groupSize, mutation, seed, chanceUseCloseCity, _ cityList) AddHandler tsp.foundNewBestTour, AddressOf tsp_foundNewBestTour ' tsp.foundNewBestTour -= New Tsp.NewBestTourEventHandler(tsp_foundNewBestTour) tsp = Nothing</pre>		

Project Title	Beacon 1.0	Testing Date	11-04-13
Test Name	First –Fit Algorithm		
Test ID	Beacon-UT-White Box 2		
Conducted By	Binit Kumar		
Description	First Fit Algorithm		

M-Admin-UT8 -Test Case #1		
Function to Test:	First –Fit Algorithm :	
Execution Steps:	On the ClpForm , First-Fit Algorithm can be selected to use And Select Decreasing to use	
Expected Result	Actual Result	Status(Pass/Fail)
It steps through the Elements sticking them into the first Bin it can, if there aren't any Bins that it will fit into, a new Bin is added	First Fit Algorithm is running but the arrays are not getting sorted in between at the tick of decreasing checkbox	Fail
Code Errors Identified:	When Decreasing is True Array.sort and Array .Reverse syntax in not working	
Tested Code: Following code is test and no sorting is happening here	<pre> Dim ElementsCopy(Elements.GetUpperBound(0)) As Integer ReDim Bins(0) 'Bin Number we are on, Bin Element we are on, Amount placed in the cur Dim BinNumber, BinElement, BinCount As Integer Dim i, j, k As Integer 'Sort in descending order if needed If Me.Decreasing = True Then Array.Sort(Elements) Array.Reverse(Elements) End If 'Declare the first element in the first Bin ReDim Bins(0)(0) 'Loop through each Element and place in a Bin For i = 0 To ElementsCopy.GetUpperBound(0) n:= BinCount As Boolean = False For j = 0 To BinCount - 1 If Bins(j) <= ElementsCopy(i) Then Bins(j) = ElementsCopy(i) n = True Exit For End If Next j If Not n Then ReDim Preserve Bins(0)(BinCount) BinCount = BinCount + 1 Bins(BinCount) = ElementsCopy(i) End If Next i </pre>	

Rectified Code:

DeepCopyArray(Elements, ElementsCopy) function is used to create a new copy incase the module is not working

```
Dim ElementsCopy(Elements.GetUpperBound(0)) As Integer
ReDim Bins(0)
'Bin Number we are on, Bin Element we are on, Amount placed in the current
Dim BinNumber, BinElement, BinCount As Integer
Dim i, j, k As Integer

'Make a copy of the array incase we need to sort it
| DeepCopyArray(Elements, ElementsCopy)

'Sort in descending order if needed
If Me.Decreasing = True Then
    Array.Sort(ElementsCopy)
    Array.Reverse(ElementsCopy)
End If

'Declare the first element in the first Bin
ReDim Bins(0)(0)

'Loop through each Element and place in a Bin
For i = 0 To ElementsCopy.GetUpperBound(0)
    Dim bPlaced As Boolean = False

    'Loops through each Bin to find the first available spot
    For j = 0 To BinNumber
        BinElement = Bins(j).GetUpperBound(0)

        'Count the amount placed in this Bin
        BinCount = 0
        For k = 0 To BinElement
            BinCount += Bins(j)(k)

Private Sub DeepCopyArray(ByVal ArrayStart() As Integer, ByVal ArrayEnd() As Integer)
    Dim i As Integer

    For i = 0 To ArrayStart.GetUpperBound(0)
        ArrayEnd(i) = ArrayStart(i)
    Next
End Sub
```

8.4 System testing

System testing was finally conducted on the complete, integrated system to evaluate the system's compliance with its set requirements. The purpose of system testing is to validate an application's accuracy and completeness in performing the functions as designed. System testing simulates real life scenarios in a test environment. (Robda vispe 2004)

System testing is deemed complete when actual results and expected results are either in line or differences are explainable or acceptable based on client input. (Beizer, 1984, p. 24-25)

8.4.1 Full System Testing

Project Title	Beacon 1.0	Testing Date	15-04-13
Test Name	System Testing		
Test ID	Beacon-ST		
Conducted By	Binit Kumar		
Description	Full System Test		

Beacon-ST1 -Test Case #1		
To Test:	Design Route Map	
Expected Result	Actual Result	Status(Pass/Fail)
Mouse Down to Create Cities	Cities are created	Pass
Show Graph Button to show external Graph in map	Graph is displayed	Pass
Browse File to open dialog box and filer xml	Browse dialog is opening	Pass
Load xml file based co-ordinates in map and array	Cities are loading through xml	Pass
Validate Cities	Validation is running	Pass
Clear Button to clear the map	All Clear	Pass

Beacon-ST1 -Test Case #2			
To Test:	Plan Route		
Expected Result		Actual Result	Status(Pass/Fail)
Default Constraints are loading		Default values	Pass
Edit the default values		Editable	Pass
Start Algorithm		Successful Beginning of TSP	Pass
Stop Algorithm at any time		Stop	Pass
Start when validated cities		Check of Validation is complete	Pass
Save Image of map in jpeg, png and gif format		Successful save of image	Pass
Zooming of pictureBox		Successful	Pass
Saving into the Database		Successful	Pass

Beacon-ST1 -Test Case #3			
To Test:	XML Editor		
Expected Result		Actual Result	Status(Pass/Fail)
Xml Reading of co-ordinates		Reading	Pass
Xml file writing		Writing	Pass

Beacon-ST1 -Test Case #4			
To Test:	Manage Loading		
Expected Result		Actual Result	Status(Pass/Fail)
Default load length of containers in the element multivalued textbox		Loading	Pass
Ship Height value		Default 80	Pass
Select Algorithm		Successful flow	Pass
Decreasing Order		Selected	Pass

Beacon-ST1 -Test Case #5			
To Test:	View Loading		
Expected Result		Actual Result	Status(Pass/Fail)
Initialization of Graph		Successful	Pass
DrawDemarcations		Showing percentage wise	Pass
Draw Bins by the selected elements , algorithm ,order and ship length		Successful	Pass
First-Fit Algorithm		Working as required	Pass
Next-Fit Algorithm		Working as required	Pass
Worst-Fit Algorithm		Working as required	Pass
Best –Fit Algorithm		Working as required	Pass
Update Graph on every event changes		New Graph is loaded everytime	Pass

Beacon-ST1 -Test Case #6			
To Test:	Change Language		
Expected Result		Actual Result	Status(Pass/Fail)
Language are Selected		All three can be chosen	Pass
Applying Changes		Successful conversion	Pass

Beacon-ST1 -Test Case #7			
To Test:	Reports		
Expected Result		Actual Result	Status(Pass/Fail)
View of data from database for TSP		Successful loading	Pass
View of data from database for CLP		Successful loading	Pass

Conclusion

The developer concludes that the system is working fine and all Project Specification Requirements have been fulfilled with some minor and acceptable changes made.

8.5 Compatibility Testing

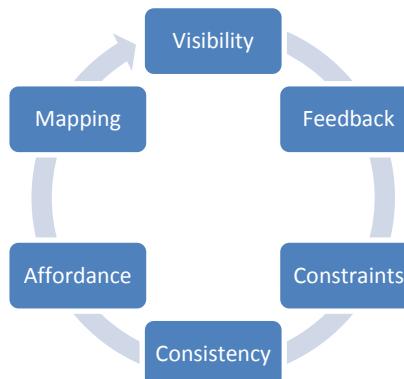
Compatibility testing was conducted on the application to evaluate the application's compatibility with its computing environment. It was done to make sure that the Beacon application which was till now being tested on a Microsoft Visual Studio would work as expected in a real life environment on a real Application.

Project Title	Beacon 1.0	Testing Date	16-04-13
Test Name	Compatibility Testing		
Test ID	Beacon-CT		
Conducted By	Binit Kumar		
Description	Compatibility with Windows XP, Windows Vista, Windows 7 and Windows 8		

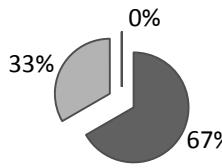
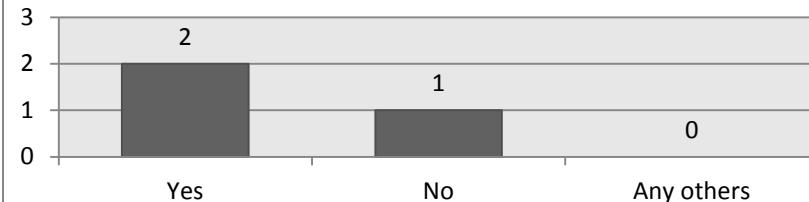
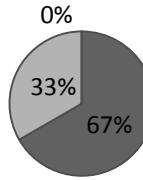
Beacon-CT1 -Test Case #7	
Test Performed	Result
Beacon was run on Windows XP	As Expected
Beacon was run on Windows Vista	As Expected
Beacon was run on Windows 7	As Expected
Beacon was run on Windows 8	As Expected
Beacon was run on Windows Mobile	As Expected
Beacon was run on Windows Server	Successful
Beacon was run on Windows Ubuntu	By installing Wine, Successful running

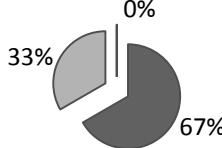
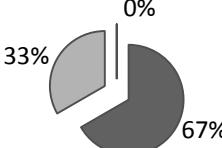
8.6 Usability Testing

The system was given to a user and he was asked to test the system to check whether the system fulfils the requirement. (Refer to appendices for Usability questionnaire)



Question 1	Do you get proper feedback messages for actions like running status of algorithm, values, connection etc.? <input type="checkbox"/> Yes <input type="checkbox"/> No <input type="checkbox"/> Any other, Specify _____								
Justification	This will help to achieve feedback principle in which user gets feedback for every action								
Response	<table border="1"> <thead> <tr> <th>Response</th> <th>Percentage</th> </tr> </thead> <tbody> <tr> <td>Yes</td> <td>67%</td> </tr> <tr> <td>No</td> <td>33%</td> </tr> <tr> <td>Any others</td> <td>0%</td> </tr> </tbody> </table>	Response	Percentage	Yes	67%	No	33%	Any others	0%
Response	Percentage								
Yes	67%								
No	33%								
Any others	0%								
Analysis	As per the user consensus (67% users are in favor), the developer has concluded that feedback principle is achieved.								
Question 2	Whenever you provide any wrong input, do you get proper error messages (validations)? <input type="checkbox"/> Yes <input type="checkbox"/> No <input type="checkbox"/> Any other, Specify _____								
Justification	This will help in understanding that input validations are in proper format that user can understand.								
Response	<table border="1"> <thead> <tr> <th>Response</th> <th>Count</th> </tr> </thead> <tbody> <tr> <td>Yes</td> <td>2</td> </tr> <tr> <td>No</td> <td>1</td> </tr> <tr> <td>Any others</td> <td>0</td> </tr> </tbody> </table>	Response	Count	Yes	2	No	1	Any others	0
Response	Count								
Yes	2								
No	1								
Any others	0								
Analysis	As per the user consensus (67% users are in favor), the developer has concluded that validations are successfully implemented and constraints principle is fulfilled.								
Question 3	Do you find color, background & actions of this application, consistent? <input type="checkbox"/> Yes <input type="checkbox"/> No <input type="checkbox"/> Any other, Specify _____								
Justification	This will help to achieve consistency principle in which user finds consistency throughout the application								

Response	 ■ Yes ■ No ■ Any others
Analysis	As per the user consensus (67% users are in favor), the developer has concluded that consistency principal is achieved.
Question 4	<p>Do you get expected behaviors of every action like whenever you zoom perfect zooming happens?</p> <input type="checkbox"/> Yes <input type="checkbox"/> No <input type="checkbox"/> Any other, Specify _____
Justification	This will help to achieve affordance principle in which user finds expected behavior of actions
Response	 ■ User Responses
Analysis	As per the user consensus (67% users are in favor), the developer has concluded that affordance principal is achieved.
Question 5	<p>Do you find this system fulfilling the basic functionality (Route Finder, Load Planner) of a social network with advance location concepts GA algorithm and maps and graphs</p> <input type="checkbox"/> Yes <input type="checkbox"/> No <input type="checkbox"/> Any other, Specify _____
Justification	This will help in verifying that system covers all the functionality needed.
Response	 ■ Yes ■ No ■ Any others
Analysis	As per the user consensus (67% users are in favor), the developer has concluded that System is fulfilling the customer's need.
Question 6	Are you able to perform all the activities of this application like (route

	calculator, best container planner, language change, database connection, xml reader/writer) without asking to anybody, only with the help of user manual? <input type="checkbox"/> Yes <input type="checkbox"/> No <input type="checkbox"/> Any other, Specify _____
Justification	This will help in verifying that user can work independently without any third-party help
Response	 ■ Yes ■ No ■ Any others
Analysis	As per the user consensus (67% users are in favor), the developer has concluded that user manual guidelines are sufficient to work on the system.
Question 8	Are you able to read the text and find the controls in proper order & sequence like textboxes and menu options? <input type="checkbox"/> Yes <input type="checkbox"/> No <input type="checkbox"/> Any other, Specify _____
Justification	This will help to achieve visibility & mapping principle in which user is able to find the controls
Response	 ■ Yes ■ No ■ Any others
Analysis	As per the user consensus (67% users are in favor), the developer has concluded that visibility & mapping principal is fulfilled in the system.

8.7 User Acceptance Testing

User acceptance testing is usually the last stage of the testing process. The question is who is the user? Here the user means the real business users, who will have to operate the system. A developer cannot perform this type of testing he cannot understand the business needs of the system that it must deliver to an organization where it is deployed

The developer contacted Mr. Tanveer Ahmad, a business man of Logistics Chain from Ranchi, to check the features of Beacon and suitability with the scenario. They have tested both the modules and every features according to their needs. The developer also asked Ms. Richa Gupta to test the system for her use in the store for bin packing.

Project Title	Beacon 1.0	Testing Date	16-04-13
Test Name	Beacon User Acceptance Testing		
Test ID	Beacon-UAT		
Conducted By	Mr. Tanveer Ahamad , Ms. Richa Gupta		
Description	Testing in the real business environment		

Beacon-UAT -1 -Test Case #1	Mr. Tanveer Ahmad	Ms. Richa Gupta		
	Status	Excellence %	Status	Excellence %
Route Map and Designing	Yes	85%	Yes	100%
Xml Reading and Writing	Yes	100%	Yes	98%
GA algorithm for TSP	Yes	100%	Yes	100%
Packing Planning	Yes	95%	Yes	80%
Efficiency of Packing and Loading	Yes	100%	Yes	100%
Database Entry	Yes	100%	Yes	100%
Reports	Yes	100%	Yes	85%

Sign Off			
No	Name	Date	Signature
1	Mr.Tanveer Ahmand	1 st May 2013	
2	Ms. Richa Gupta	5 th May 2013	

8.7 Testing Summary

Testing **Beacon** has been the most tedious of all phases. The developer has invested the maximum effort and time in the testing process. During the testing process a lot errors were identified and subsequently resolved.

According to developers point of view **unit testing** has been the most efficient in terms of errors identified and corrected. The unit testing involved testing the application using black box and white box methods; a large number of black box test cases were produced and the eventual tests were performed.

In the documentation above only those test case descriptions were provided which the developer assumed would be of reading interest to any reader. The white box testing phase was very lengthy and nerve cracking session for the developer as hundreds of lines of codes had to be reviewed and tested numerous times before the final desired outcome was achieved.

Compatibility testing turned out to be a difficult task for the developer. A large number of phones with even larger set of platform which changes drastically each day had put the developer in a fix.

Chapter 9 Critical Evaluation

9.1 Introduction

Critical Evaluation of a project is very important. After completion of the various phases of project development it is of utmost important that the developer does the critical evaluation of the complete project module wise.

So, on the completion of the project it was evaluated for the benefits it gives to the targeted end users. How it solves problems of the organization which is not having any such system.

9.2 Usefulness of the Beacon for Target Users

Reduced Complexity of Problem-.TSP and CLP problems are being solved with the most efficient problem solution being designed and implemented by developer.

Graphical view of route and container loading – The system will generate 2D/3D view of the containers to show the best loading graph in CLP and the best route graph in TSP.

Cost Benefits - Most efficient route will be produced, results in precious fuel saving hence cost cutting. Best Container loading plan optimizes the overall loading with the ships and also maximizes the number of container .Hence this reduces the administration and managerial overhead results into cost benefits

Time Benefits- with Beacon, and ships save time by going through the shortest path provided results in more resource availability. For Example: More the trucks reduce the time of travelling; more the loading of trucks will be possible. Also the container loading takes the minimum time in ships.

Increased work flexibility Beacon makes it easier to understand the problem statement and to organize and plan the things based on different constraints.

Approach of Solving: Since the algorithms used in Beacon are very advanced and hard to understand, these approaches can be used in various places like, circuit board, neural networks, packing problem, conveyor belt problem etc. The system can also be used for studying about Genetic algorithm for any researcher or programmer.

Chapter 10 Conclusion

This has been no less than a journey. A year round of work has finally culminated in a beautiful and successful project. Right through the project phase from initiation to completion, a large trove of knowledge and hands on experience has been gained by the developer. The knowledge and experience are surely going to drive and guide the developer to be a more better IT professional and towards a bright future.

At the outset the developer wished to discuss few such things which have till now remained in the back burner, like does the system solve the problem it was supposed to? What limitations have yet remain untouched? Are there any bugs in the final system? What is the future of the software? What are the possible future enhancements? What if the developer was asked to restart the development from scratch, what would be the changes the developer would like to see? What posed as the main computational challenge for the developer and over all discuss his yearlong learning experience.

The developer wishes to answer all these and more one piece at time:

10.1 Degree of Success

How are we supposed to measure the degree of success of newly developed software? What is going to be the criteria? According to the developer if the software solves the problem it set out to do, fulfilling all functional requirement then it can be said to be a success.

Now what had the developer in mind when he proposed the system? The vision for Beacon was that:

- A tool which would solves the algorithmic and logical complexity
- A Container Loader tool
- A Fun and nice graph generation for route map and container loading

Basically the developer can say this project successfully met all the requirement stated in the requirement specification and as well as in the project specification form because most of the proposed features and functions have been implemented which as well as including some of the special features. This system correctly implements and fulfils the scope of the project and solved the problem specified.

10.2 Limitations and Errors

Developer admits the fact that every system has some limitations and some functionality have been left. Similar is the case with the proposed system. Developer is not able to implement some of the functionality due to time constraints. The list of functions that could have been implemented but were not included in the final system, due to the lack of time, are listed below

- ✚ Support on Linux and mobile based operating System
- ✚ Synchronisation between application and web module
- ✚ Management software with these features

10.3 Possible Future Enhancement

Each and every system has many enhancements to do to make system better for the users because no system is perfect. Some of the identified future enhancements are:

- ✚ Cloud Based Software (SaaS) is in the mind of developer
- ✚ Cloud based software can be accessed with profile management by different users
- ✚ Module for Mobiles

10.4 Main Computational Challenge

The main computational challenge comes while developing this system is:

- ✚ Mapping the Genetic Algorithm approach to the Travelling Salesman Problem
- ✚ Generating the Graph of the Container Loading Problem
- ✚ Crossover and Data Structure mapping for Tours.

10.5 Learning Experiences

- ✚ Time Management – The most crucial part of any project
- ✚ Tools and Techniques – Microsoft visual Studio as a particular tool with VB
- ✚ Better Knowledge of Algorithm – Algorithm is all fun that rules the world
- ✚ High Motivation – Increased confidence and Positive attitude
- ✚ Project Management- Managerial Skill while handling projects.

References

Books / Articles and White Paper References

Reference Detail
Bajpai, 2008, <i>Genetic Algorithm – an Approach to Solve Global Optimization Problems</i> . Indian Journal of Computer Science and Engineering, 1-3.
Beizer, Boris, 1984, Software System Testing and Quality Assurance, 1st Edition, New York, Van Nostrand Reinhold
Burnstein, 2002, Practical software testing: A Process-oriented approach, Springer-Verlag, USA
B. Chazelle, 1983, <i>The bottom-left bin packing heuristic: An efficient implementation</i> . IEEE Transactions on Computers, 32:697--707,
Dirk, 2009, <i>Multi-Objective Container Loading using an Evolutionary Algorithm</i> . Crawley, University of Western Australia.
Haupt, 2004, Practical Genetic Algorithms, 2 nd Edition, <i>The Binary Genetic Algorithm</i> (p. 28), John Wiley & Sons, Inc, New Jersey
Kathy, S 2006, <i>Information Technology Project Management</i> , Thompson Course Technology, USA
Parre, 1994, <i>A maximal space Algorithm for container loading problem</i> , University of Valencia , Spain
Petitpierre, Claude, 2006, <i>Software Engineering: The Implementation Phase</i> , 1 st Edition, Paris, Communication and Information Sciences
Pilone, Dan, 2006, <i>UML 2.0 Pocket Reference</i> , 1 st Edition, O'Reilly Media, New York
Pressman, R,S, 2004, <i>Software Engineering-A Practitioner's Approach</i> , 5 th Edition, Singapore, McGraw-Hill
Ray, Bandyopadhyay, Pal, 2004, <i>New Operators of Genetic Algorithm for Travelling Salesman Problem</i> , Indian Statistical Institute, Kolkata

Website Reference

Reference Detail
Anand Ramdeo, 2011, White Box Testing. Retrieved 16 st April 2011, from http://www.testinggeek.com/white-box-testing
Anonymous. (2000). Travelling Salesman Problem, Retrieved September 20 th Sep 2012, from Travelling Salesman Problem: http://travellingsalesmanproblem.com/
Developer Works, 2001, <i>UML Basics: The class diagram</i> , Retrieved 18 th February 2011, from http://www.ibm.com/developerworks/rational/library/content/RationalEdge/sep04/bell/
Gordon, 2011, <i>CodeInComplete</i> . Retrieved December 20, 2013, from http://codeincomplete.com/posts/2011/5/7/bin_packing/example/
Heidi Smith, 2001, <i>Bin Packing Algorithm</i> , Retrieved March 20, 2013, from http://users.cs.cf.ac.uk/C.L.Mumford/heidi/BinPacking.html
Information Week, 2002, <i>Q&A: Bill Gates on Trustworthy Computing</i> . Retrieved 14 th April 2011, from http://www.informationweek.com/news/6502378
Kennesaw, 2001, <i>Types of UML Diagrams</i> , Retrieved 16 th February 2011, from http://atlas.kennesaw.edu/~dbraun/csis4650/A&D/UML_tutorial/diagrams.htm
Robdavispe, 2001, What is System Testing, Retrieved 26 th April 2011, from http://www.robdavispe.com/free2/software-qa-testing-test-tester-2047.html
Slavin, 2011, TED Global 2011. Retrieved September 2012, from http://www.ted.com/talks/kevin_slavin_how_algorithms_shape_our_world.html
Vanderburg, 2002, <i>Glen Vanderburg: Quotations on Software Design</i> , Retrieved 14 th February 2011, from http://www.vanderburg.org/Misc/Quotes/soft-quotes.html
Weisstein, 1999, "Traveling Salesman Problem.". Retrieved on 29 th August 2012, from http://mathworld.wolfram.com/TravelingSalesmanProblem.html