

# **INTELLIGENT IP ADDRESS LOCATOR VIA GOOGLE MAP**

A Thesis  
Presented to the  
Faculty of the Computer Science Department  
College Arts and Sciences  
**JOSE RIZAL MEMORIAL STATE UNIVERSITY**  
Main Campus, Dapitan City

In Partial Fulfillment  
of the Requirements for the Degree  
Bachelor of Science in Computer Science

**Jay Marie A. Egoogan**  
**Khimberly R. Ebe**  
**Lea E. Monopollo**

May 2017



Republic of the Philippines  
JOSE RIZAL MEMORIAL STATE UNIVERSITY  
The Premier University in Zamboanga Del Norte  
Main Campus, Dapitan city



## CERTIFICATION

This thesis entitled "**Intelligent IP Address Locator via Google Map**" prepared and submitted by Jay Marie A. Egoogan, Lea E. Monopollo, and Khimberly R. Ebe, in partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science has been reviewed, checked and verified by the undersigned as to grammar, coherence and organization.

**EDENROSE M. SAGAPSAPAN**  
*English Critic*



Republic of the Philippines  
JOSE RIZAL MEMORIAL STATE UNIVERSITY  
The Premier University in Zamboanga Del Norte  
Main Campus, Dapitan city



## CERTIFICATION

This is to certify that **Jay Marie A. Egoogan, Khimberly R. Ebe, and Lea E. Monopollo**, are taking Project Thesis for the degree of Bachelor of Science in Computer Science during the second semester, School Year 2015.

**QUINDHE M. BANQUIAO, Ed.D**  
*Dean, College of Arts and Sciences*

This thesis entitled “**INTELLIGENT IP ADDRESS LOCATOR VIA GOOGLE MAPS**” prepared and submitted by **Jay Marie A. Egoogan, Khimberly R. Ebe, and Lea E. Monopollo**, has been examined and is recommended for final examination on May 2017.

**ARMANDO T. SAGUIN JR., MSIT, MTCNA, CCTT, MTCRE, MTCUNE**  
*Adviser*



Republic of the Philippines  
JOSE RIZAL MEMORIAL STATE UNIVERSITY  
The Premier University in Zamboanga Del Norte  
Main Campus, Dapitan city



## APPROVAL SHEET

This thesis entitled "**INTELLIGENT IP ADDRESS LOCATOR VIA GOOGLE MAPS**" has been prepared by **Jay Marie A. Egoogan, Khimberly R. Ebe, and Lea E. Monpollo**, has been reviewed and is recommended for acceptance and approval.

**EDENROSE M. SAGAPSAPAN**  
English Critic

**ENGR. ANGELITO M. PUNZAL, MSIT**  
Adviser/Instructor

---

### PANEL EXAMINERS

**APPROVED** by the COMMITTEE IN ORAL EXAMINATION on April \_\_\_\_, 2017 with the rating of \_\_\_\_.

**JOHN D. SAGAPSAPAN, MSIT**  
*Member*

**ARMANDO T. SAGUIN Jr., MSIT**  
*Member*

**ED NEIL O. MARATAS, MA MATH**  
*Member*

**ENGR. ANGELITO M. PUNZAL, MSIT**  
*Member*

**ENGR. JOSEPH AURELIUS P. JACINTO, MSIT**  
*Chairman*

**Accepted** and approved in partial fulfillment for the subject Thesis leading to the degree of Bachelor of Science in Computer Science leading to the degree of Bachelor of Science in Computer Science during the second semester, School Year 2017.

**QUINDHE M. BANQUIAO, Ed.D.**  
Dean, College of Arts and Sciences

## **DEDICATION**

*We, the researchers, dedicate this study to our respective parents for the guidance and unquestioned moral and financial support that made us pursue this study.*

*To the College of Arts and Sciences, for providing us instructors who untiringly continue striving and giving us quality of education that we sought for. There is no doubt in mind that without their continued support and counsel we could not have completed this process.*

*To our adviser, for giving us more ideas for making this project a successful one.*

*To our valued friends for their moral support and cooperation. Thanks for inspiring us to go on, no matter what problems may arise in their path in making the study.*

*To our Alma Matter, Jose Rizal Memorial State University, for developing our intellectual aspects, modeling our personality and for making us better individuals.*

*To our Heavenly Father, for the blessings knowledge that He has given us.*

***The Researchers***

## **ACKNOWLEDGEMENT**

We would like to express our gratitude to all who extended their continuing help, support, encouragement and to those who believed in our study to work successfully. To God the father of all, we are thankful for the strength that keeps us standing and for the hope that keeps us believing that this affiliation would be possible and more interesting. We also want to thank our families for being there, for their never ending support, financially, spiritually and morally, for not letting us down during our straggling days and for giving us encouragement and to our dear friends and classmates who gave us the advices and sharing their knowledge, contributes a lot during the development of the system study.

We would like also to thank our adviser, **Prof. Armando T. Saguin Jr.**, who shared his thoughts, remarkable ideas, knowledge, advice and suggestions, and to **Prof. John D. Sagapsapan, Engr. Joseph Aurellious P. Jacinto, and Prof Ed Neil O. Maratas**, for their assistance, and for showing us examples related to the topic.

To our instructor **Engr. Angelito M. Punzal** who supported and guided us to develop our study, without those knowledge and assistance this study would not have been successful. This research project would not have been possible without the support of the people above.

Thank you so much!

***The Researchers***

## **TABLE OF CONTENTS**

### **PRELIMINARIES**

Title Page	i
Certification	ii-iii
Approval Sheet	iv
Abstract	xiii
Dedication	v
Acknowledgement	vi
Table of Contents	vii
List of Tables	xi
List of Figures	xii

### **CHAPTER 1 THE PROBLEM AND ITS SCOPE**

Introduction	1-2
Theoretical/Conceptual Framework	2-6
The Schema of the Study	6-8
Statement of the Problem	9
Significance of the Study	9
Scope and Delimitation	10-11
Operational Definitions of Terms	11-12

### **CHAPTER 2 LITERATURE AND STUDIES**

Literature	13-21
Studies	21-28

## **CHAPTER 3 RESEARCH METHODOLOGY**

Research Methods	29
Research Environment	29-30
Project Development Process	30-31
Data Gathering	31
Requirements Analysis	31
Designing	32-33
System Flowchart	33
Unified Modeling Language	34-35
Class Diagram	35-36

## **CHAPTER 4 RESULTS AND DISCUSSIONS** 53-60

## **CHAPTER 5 SUMMARY, FINDINGS, CONCLUSIONS AND RECOMMENDATIONS**

Summary	61-62
Findings	62-64
Conclusions	64
Recommendations	64
Bibliography	65

## **APPENDICES**

**Appendix A** User's Manual

**Appendix B** Source Code

**Appendix C** Letter to the Panelist

## **LIST OF TABLES**

Table 1.Respondent of the Study	48
Table 2.Calendar of Activities	51
Table 3.Over-all Result on the Factor: Functionality	57
Table 4.Over-all Result on the Factor: Reliability	58
Table 5.Over-all Result on the factor: Usable	59
Table 6.Over-all Result of all Functional, Reliability, and Usability	60

## LIST OF FIGURES

Figure 1. Schema of the study	8
Figure 2. Black Diagram of Security System	14
Figure 3. Program flow chart of the Tracking system	16
Figure 4. Block Diagram and Working of the Proposed System	18
Figure 5. Flow Chart Subroutine sends AT command	20
Figure 6. Architecture of Bus Management and Monitoring System	28
Figure 7. Project Development Life Cycle	31
Figure 8. System Project Design Diagram	32
Figure 9. System Flow Chart	33
Figure 10. Use Case Diagram	35
Figure 11. Class Diagram	36
Figure 12. Object Diagram	37
Figure 13. Sequence Diagram	38
Figure 14. Activity Diagram	39
Figure 15. Data Flow Diagram	40
Figure 18. Entity Relational Diagram	41
Figure 20. Software Quality Factors	47
Figure 21. Parallel Method of System Implementation	50

## **INTELLIGENT IP ADDRESS LOCATOR VIA GOOGLE MAP**

### **Abstract**

This study aimed to develop a “Intelligent IP Address Locator via Google Maps”. The developmental research method was used which has been defined as the systematic study of designing, developing, and evaluating instructional programs, processes, and products that must meet criteria of internal consistency and effectiveness. The main assessment tool was the survey questionnaire which was prepared by the researchers based on ISO 9126 for software quality factor. There were 30 respondents included in this study. The statistical tools used were the frequency count and mean computation. The study revealed that the developed Intelligent IP Address Locator via Google Maps has a feature where the system is a technique designed for solving a problem more quickly when classic methods are too slow, or for finding an approximate solution when classic methods fail to find any exact solution. This is achieved by trading optimality, completeness, accuracy, or precision for speed. It was concluded that the developed Intelligent IP Address Locator via Google Maps was “Acceptable” when measured in terms of functionality, reliability, usability, efficiency, maintainability and portability and meets the requirements and intended functionalities. It is recommended that the old manual procedure of informing the residents on awareness and response information system in Dapitan City be process. Adoption of this system basically means very functional. The outcome of this study will be beneficial to many communities, institutions, organizations and residents. Its advanced feature will be convenient to utilize the systems to the maximum benefits available today.

# **Chapter 1**

## **THE PROBLEM AND ITS SCOPE**

### **Introduction**

Expert system is a branch of computer science is artificial intelligence. Their ability to mimics experts by applying their domain knowledge has led to the construction of a number of medical applications. A brief resume of the structure and the knowledge-based expert system is given. Reasons are given for the failure of these successful programs to be widely implemented. It is to be expected that improvements in other areas of artificial intelligence may make the more widely acceptable to the non-expert. Intelligence is the ability to reach once objectives. A system is more intelligent if it reaches its objectives faster and easier. The intelligence of a system is a properly of its mind. The mind is the functioning of its brain. An intelligent system is a machine with an embedded, internet-connected computer that has the capacity to gather and analyze data and communicate with other systems. Intelligent system also is a system that has that has its own main objective, as well as senses and actuators to reach its experiences. It requires for an intelligent system include security, connectivity, and the ability to adopt according to current data and the capacity for remote monitoring and management.

The national information infrastructure may have profound effects on the lives of every citizen. It promises to deliver to people in their homes and offices a vast array of information in many forms, changing the ways in which business is conducted, offering new educational opportunities, bringing geographically dispersed library resources and entertainment materials to everyone's doorstep. It may connect people, and help them with their jobs and tasks. However, people may need easy and efficient access to its

resources. Today's computers are complex and difficult to use, even for experts. The national information infrastructure may be orders of magnitude more complex than current systems; it could easily become labyrinth of databases and services that is inconvenient for experts and inaccessible to many people.

Intelligent system is anything that contains a functional, computer with internet connectivity. An embedded system may be powerful and capable of complex processing and data analysis, but it is usually specialized for tasks relevant to host machine. As this ongoing trend continues, many foresee a scenario known as the Internet of Things (LoT), in which objects, animals, and people can all be provided with unique identifiers data over a network without requiring human-to-human or human-to-computer interaction.

Artificial intelligence (AI) uses the theoretical and experimental tools of computer science to study the phenomena of intelligent behavior and to construct intelligent systems. AI research has also produced an extensive body of principles, representations, algorithms, and spin-off technologies. Successful applications range from the DART system, which was used in deployment planning to desert shield, to broadly adopt symbolic math packages to a thousand of fielded expert systems. Incorporating AI technology into the next generation of computers can help ensure the nation's information infrastructure flexible and easy to use.

### **Theoretical/Conceptual Framework**

This study is premised on "Location-Based Communication Systems A Look at Intelligent Networking and Privacy Concerns" by Laurie Thomas Lee, Ph.D. on (May 2015). Location-based networks are quickly becoming an integral feature of today's intelligent networking landscape. They are transforming how people and government communicate. Global positioning satellite systems, Wi-Fi, and other such decentralized networks make it possible to track and physically locate users and pair that information

with location-based information from and about others, such as friends, products, and services. Users with Smartphone's, for example, can access local maps, check out nearby restaurant specials, notify friends as to their whereabouts, find others, and get information on local traffic alerts, events and landmarks, all by accessing the various types of networking systems and applications that rely on location tracking. Finally, because intelligent networks by their nature present both advantages and disadvantages, the opportunities of location-based networks as well as specific problems such as location privacy violations are discussed.

Other anchored to this study is "Intelligent System for Tourism" by Steffen Staab, University of Karlsruhe, sst@aifb.uni-karlsruhe.de Hannes Werthner, ITC-irst Research Center, werthner@itc.it on November 2002. The tourism industry has specific features that explain its importance for economic (regional) development and its inclination toward IT systems. Tourism is a leading industry worldwide, representing approximately 11 percent of the worldwide GDP. There will be approximately one billion international arrivals in the year 2010. Furthermore, tourism represents a cross sector (umbrella) industry, including many related economic sectors such as culture, sports, and agriculture, where over 30 different industrial components have been identified that serve travelers. In addition, tourism greatly influences regional development, owing to its SME structure and relatively small entrance barriers. For example, in the European Union, the hotel and restaurant sector accounts for more than 1.3 million enterprises. This is approximately 8.5 percent of the total number of enterprises, and 95.5 percent of these enterprises are small. Also, because tourism is based on mobility, the supply and demand side forms a worldwide network, where production and distribution are based on cooperation. In addition, it is an information-based industry, so the tourism product is a confidence good, where at the moment of decision-making, only information about the

product—not the product itself—is available. The problem with these statistics is that they refer to different meanings and varying definitions of e-business and e-commerce. Some definitions distinguish between the two, while others view them as the same, and all have their own variables and measurement methods. Even more problematic is that the definitions are all transaction- and business oriented. They ignore that the Web is also a medium for creating communities, learning new things, and having fun—things that don't always result in business. The Web also encourages user interaction; users can build their own sites to share their travel experiences. Thus, another traveler—rather than hotel management or a travel agency—might provide the most valuable information about a vacation resort.

Another related to this study is “Intelligent Chat Bot for Banking System” by Mr. Aniket Dole, Mr. Hrushikesh Sansare, Mr. Ritesh Harekar, and Mrs. Sprooha Athalye on (October 2015). This banking bot project is built using artificial algorithms that analyze user's queries and understand user's message. We are going to design system for banks where users can ask any bank related questions like loan, account, policy etc. This application may be developed for web users. There is no specific format the user must follow to ask questions. The built in artificial intelligence system realizes users requirements and provides suitable answers to the user. The purpose of a chat bot system is to simulate a human conversation; the chat bot architecture integrates a language model and computational algorithm to emulate information chat communication between a human user and a computer using natural language. With the improvement of data-mining and machine-learning techniques, better decision-making capabilities, availability of corpora, robust linguistic annotations/processing tools standards like XML and its applications, chat bots have become more practical in daily life applications such as help desk tools, information retrieval tools, automatic telephone answering systems,

advertising, tools to aid in education, business and Ecommerce. In E-commerce, chat bot helps in information retrieval tasks, such as for searching and browsing, as menu based navigation poses difficulties in locating the appropriate information. According to Dr. Wallace, perhaps, the biggest market of chat bot is Entertainment Markets, in which, we can imagine that chat bots can act as a talking book for children and provide foreign language instruction or can be a tutor in Intelligent Tutoring system. The study focused more on user attitudes rather than on chatter bot efficiency. Another tutoring study focused on using ALICE as a course enhancement tools with Social and Political Theory knowledge. Chatter bot development is reasonably well studied ever since the Turing Imitation Game (TIG) was first proposed. Eliza was the first famous chat bot, and ALICE was another milestone. The Loebner Prize and The Chatterbox Challenge are both annual competitions which have their roots in TIG. However, these are typically text only experiments, although some limited visual components are often added. This focus is on; however, whether with the text exchange alone, we can replicate human “behavior”. This study found that most subjects used the system as a search engine rather than as a conversation partner. It was further concluded that their system was unable to function as a stand-alone tutor. Dialog system can adequately carryout the conversations with the user and can log the conversations which can be good source for knowledge acquisition for domain specific topic. Therefore, techniques of knowledge acquisition were rightly used in their system AZ-ALICE chat bot that is an extension in ALICE chatter bot.

Another primised to this study is “GPS Based Low Cost Intelligent Vehicle Tracking System (IVTS)” by Dr. Kamal Jain and Rahul Goel on October 2012. Intelligent Vehicle tracking systems (IVTS) are used for the purpose of tracking and navigation of vehicles. The paper describes the implementation of Global positioning systems (GPS) in IVTS systems. The GPS based low cost intelligent vehicle tracking system can be

successfully designed and applied in the urban environment of a developing country like India. Thus if implemented in a well planned manner may bring significant revolutionary enhancement in the Indian transportation industry. Maps are older than alphabets. It is one of the most primitive natures of human being to know their exact location on the surface of earth and at present Radio Navigation Systems (e.g NAVSTAR GPS (USA), GLONASS (Russia), GALELIO (European Community)) are the most efficient, accurate and reliable technology available to determine the precise position of any point on or above the surface of earth.

To reduce the cost of communication in Intelligent Vehicle Tracking system it is very important to reduce the size of the data packet transmitted from In Vehicle Unit to the Base Station. This can be achieved by removing the fields which are not required in the navigational solution computation from the data packet being broadcasted (Brown, 1992). The reduction in the data packet size leads to increase in communication speed, improved multi-vehicle tracking, better utilization of the communication bandwidth and reduction in communication time. These all add to overall reduction in the communication cost of the IVTS system. The Information Superhighway in India is dynamically heading towards a new dimension in Geometrics Industry. Geometrics is the synergy of geo related sciences, mainly including Surveying (Land/Geodetic), Photogrammetric, Remote Sensing, GPS, GIS. Implementation of GPS in vehicles can certainly bring a revolutionary impact in transportation science in a developing country like India where there is an extremely high urban as well as rural vehicular transition every day.

### **The Schema of the Study**

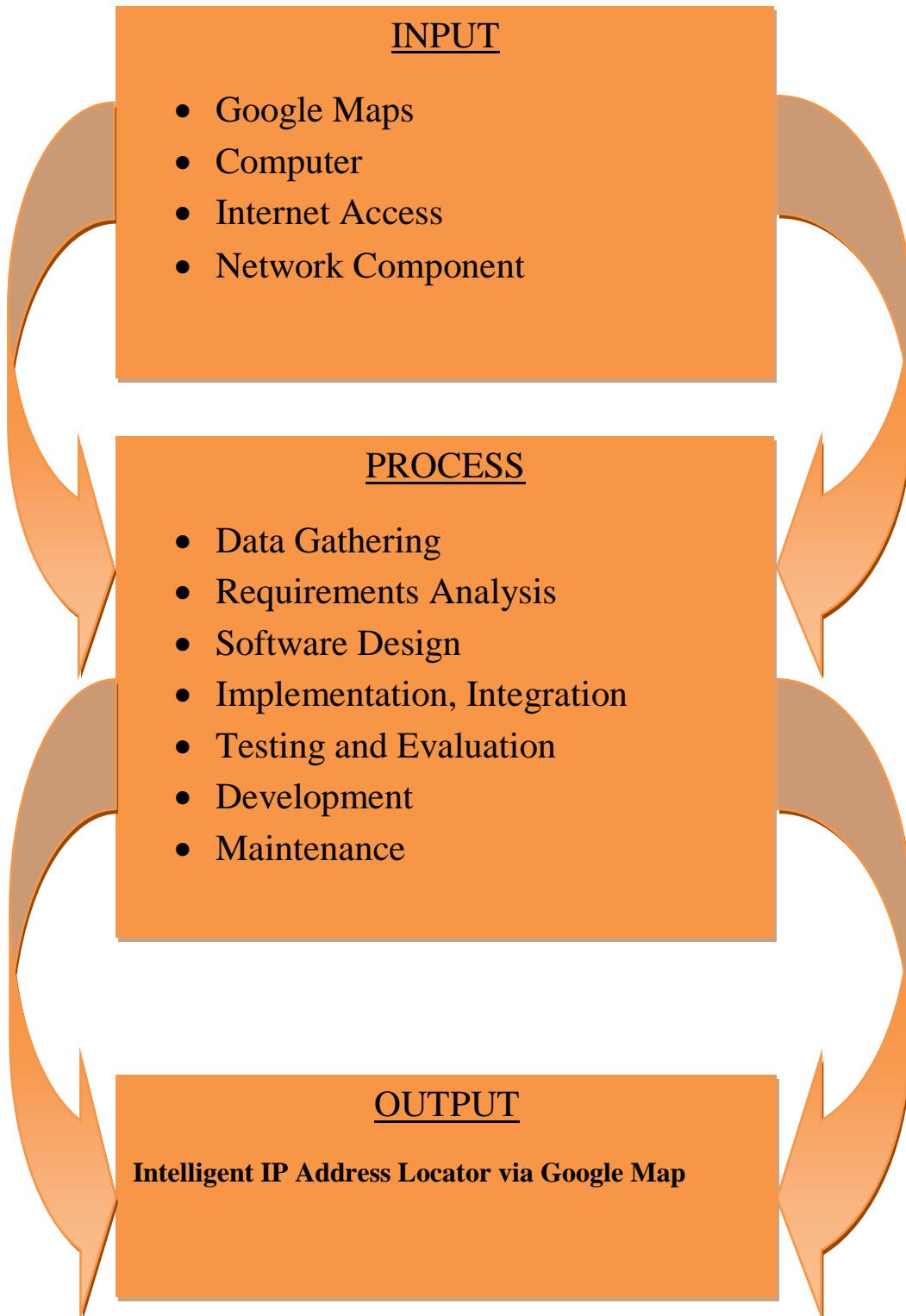
Figure 1 on page 8 presents the schema of the study. Each progress is connected into three boxes, called the IPO chart which means the Input, the Process and the Output.

Each of the boxes holds on how to essentially use of Intelligent IP address Locator via Google Maps.

The first box represents the elements or device to be used to process the data in Intelligent IP Address Locator via Google Maps. It also involves the input which is computer. Computer is a device that accepts information (in form of digitalized) and manipulates it for some result based on a program or sequence of instructions on how the data is to be processed. A program may be invariable and built into the computer (and called logic circuitry as it is on microprocessors) or different programs may be provided to the computer (loaded into its storage and then started by an administrator or user). Google map is a desktop web mapping service developed by Google. Internet Access connects individual computer terminals, computer, mobile devices, and computer networks to the internet, enabling user to access internet services, such as email and the World Wide Web. Network Component is a device such as firewall, sensors (local or remote), switches, guards routers gateways, wireless access points, and network appliances.

The second box is the solution of the relevant problems that the researchers encountered. It presents the process and system development life cycle containing the following: Data Gathering, Input Data, Requirements Analysis, Software Design Implementation and Integration, Testing, Deployment and Maintenance. In addition, the specific process illustrates the way how to track a Pc IP Address and to check a certain person.

The last third box is the output of the study in an “Intelligent IP Address Locator via Google Maps”. The user may not know that they are being checked. The researchers can get name of city from where this IP belongs to.



**Figure 1.** Schema of the study

## **Statement of the Problem**

This study aims to develop software of Intelligent IP Address Locator via Google Maps during school year 2016-2017 that may help a company to record and track its target customers by plotting them on a map. It helps companies take better marketing decisions by checking the geographic spread of their software/product in various cities.

Specifically the study sought answers to the following:

1. What is the current status of Intelligent IP Address Locator via Google Maps?
2. What are the processes that can be designed and developed of Intelligent IP Address Locator via Google Maps?
3. What are the similar or existing system features that can be fused into one composite to come up with the new hybrid system?
4. How can the Intelligent IP Address Locator via Google Maps are measured in terms of:
  - 1.1 Functionality;
  - 1.2 Reliability;
  - 1.3 Usability;
  - 1.4 Efficiency;
  - 1.5 Portability; and
  - 1.6 Maintainability?

## **Objectives**

1. To investigate the current status of Intelligent IP Address Locator via Google Map;
2. To develop an Intelligent IP Address Locator via Google Map;
3. To integrate other features into one composite system;
4. To evaluate the effectiveness of Intelligent IP Address Locator via Google Map and to

5. To implement Intelligent IP Address Locator via Google Map

### **Significance of the study**

The result of this study may benefit the following:

**Companies.** That has large numbers of contractors out on the road, such as electrical and plumbing businesses are rapidly implementing IP locator solutions.

**Researchers.** The proposed study may benefit and help the future researchers as their guide. The study is also open in development of this study.

### **Scope and Delimitation of the study**

This study focuses in an Intelligent IP Address Locator via Google Maps. It may consist of some significant modules in order to satisfy the company's needs as well as the customers'. The system may provide a security module which may have an administrators and users. An administrator is fully responsible for monitoring and maintaining the website since they have access on some modules.

The system may only have one administrator who has the authority to add more administrators by filling up the form which is available only in the admin module. This helps a company to record and track its target customers by plotting them on a map. Google Earth is very popular free software that provides maps through satellite images around the world.

### **Operational Definitions of Terms**

The research participant agreement by an individual not competent to give legally valid informed consent to participate in research.

**Government.** Is the system by which a state or community is controlled.

**Online Company.** Main businesses are to use the internet for purposes other than to make sales to the public.

**Artificial Intelligent.** The theory and development of computer system may be able to perform tasks that normally require human intelligence, such as visual perception, speech recognition, decision-making, and translation between languages.

**Google Maps.** Is a Web-based service that provides detailed information about geographical regions and sites around the world.

**API.** Is a set of subroutine definitions, protocols, and tools for building application software.

**Monitoring.** It is used by the operator to monitor the information reflects on the screen through the information which may occur over time.

**Data.** It is information that has been translated into a form that is more convenient to move or process.

**Database.** It acts as a server of the information being collect for ease and speed of search and retrieval. The data are typically organized to model relevant aspects of reality in a way that supports process requiring this information.

**Functionality.** The Intelligent IP Address Locator via Google Maps; is a convenient and flexible system which is well suited for human lifestyle towards reality. Intelligent IP Address Locator via Google Maps makes the task easy, smart, convenient, and practical.

**Reliability.** Intelligent IP Address Locator via Google Maps has the ability to track relevant information consisted by depending sources.

**Usability.** Intelligent IP Address Locator via Google Maps is a software application can be used by specified consumers to achieve quantified objectives with effectiveness, efficiency, and satisfaction in a quantified context of use.

**Efficiency.** Intelligent IP Address Locator via Google Maps is being developed to provide a reliable, secure and efficient method of recording the IP Addresses.

**Portability.** Intelligent IP Address Locator via Google Maps is capable of being transferred from one employer to another.

**Maintainability.** Intelligent IP Address Locator via Google Maps is measures the ease and speed with which a system can be restored to operational status after a failure occurs.

**Hybrid System.** Is a dynamic system that exhibits both continuous and discrete dynamic behavior – a system that can both flow (described by a differential equation) and jump (described by a state machine or automaton).

**Embedded System.** A computer system with a dedicated function within a larger mechanical or electrical system, often with real-time computing constraints. Internet.

**MySql.** an open-source relational database management system (RDBMS).

**Locator Map.** Sometimes referred to simply as a *locator*, is typically a simple map used in cartography to show the location of a particular geographic area within its larger and presumably more familiar context. Depending on the needs of the cartographer, this type of map can be used on its own or as an inset or addition to a larger map.

**Tracking System.** Observing persons or objects on the move and supplying a timely ordered sequence of location data to a model and depicting the motion on a display.

## **Chapter 2**

### **REVIEW OF LITERATURE AND STUDIES**

This chapter reviews the literature and studies that are relevant to the present one.

Different views from different persons were considered in framing up the researchers points of views. The motivation of conducting this study is to solve the problem to this modern embedded and communication technology.

#### **Literatures**

According to ISO/IEC 9126, which is the international standard for the evaluation of software quality, states that the fundamental objective of the ISO/IEC 9126 standard is to address some of the well known human biases that can adversely affect the delivery and perception of a software development project. These biases include changing priorities after the start of a project or not having any clear definitions of "success". By clarifying, then agreeing on the project priorities and subsequently converting abstract priorities (compliance) to measurable values (output data can be validated against schema X with zero intervention), ISO/IEC 9126 tries to develop a common understanding of the project's objectives and goals.

The standard is divided into four parts: quality model; external metrics; internal metrics; and quality in use metrics. The quality model presented in the first part of the standard, ISO/IEC 9126 classifies software quality in a structured set of characteristics and sub-characteristics as follows: a) Functionality – Is a set of attributes that bear on the existence of a set of functions and their specified properties. The functions are those that satisfy stated or implied needs along Suitability, Accuracy, Interoperability and Security; b) Reliability – Is a set of attributes that bear on the capability of software to maintain its level of performance under stated conditions for a stated period of time along Maturity,

Fault tolerance and Recoverability; c) Usability – is a set of attributes that bear on the effort needed for use, and on the individual assessment of such use, by a stated or implied set of users along Understandability, Learnability, Operability and Attractiveness; d) Efficiency – is a set of attributes that bear on the relationship between the level of performance of the software and the amount of resources used, under stated conditions along Time behaviour and Resource utilization; e) Maintainability – is a set of attributes that bear on the effort needed to make specified modifications along Analyzability, Changeability, Stability and Testability; and f) Portability – is a set of attributes that bear on the ability of software to be transferred from one environment to another along Adaptability, Installability, Co-existence and Replaceability.



**Figure 3. ISO 9126 Software Quality Factor**

Each quality sub-characteristic (e.g. adaptability) is further divided into attributes. An attribute is an entity which can be verified or measured in the software product. Attributes are not defined in the standard, as they vary between different software products.

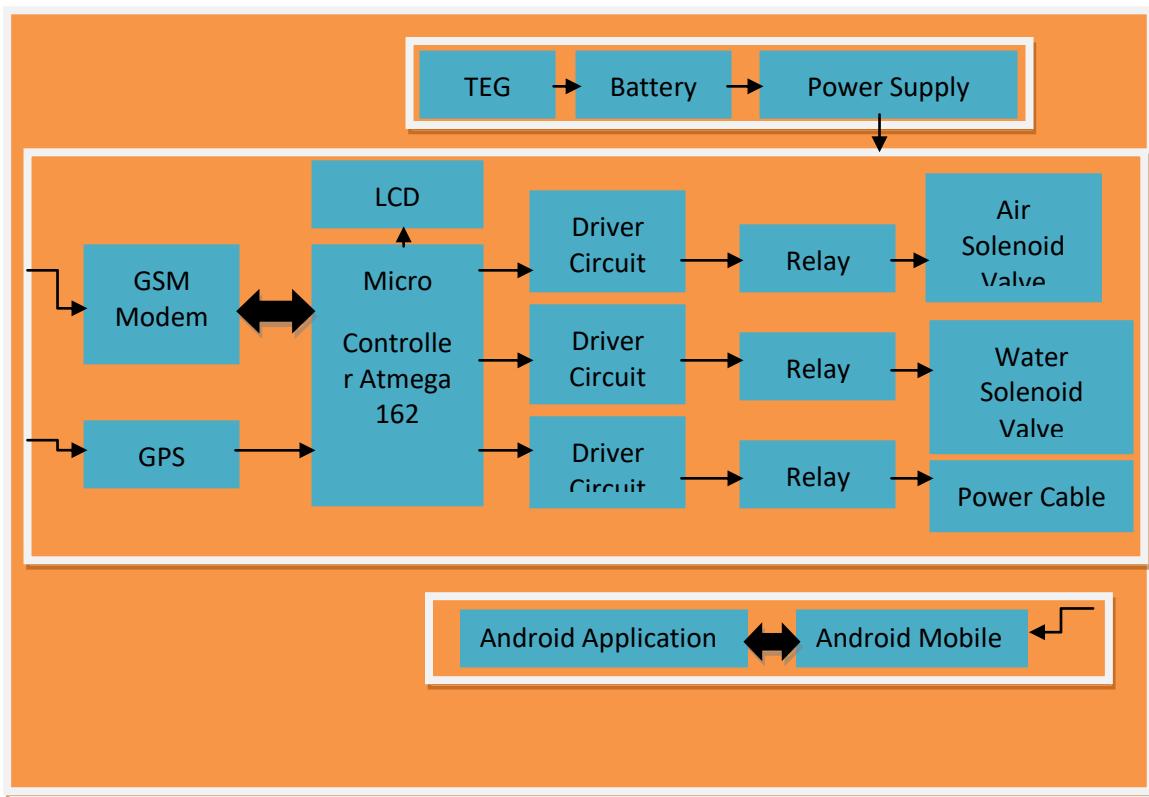
Software product is defined in a broad sense: it encompasses executables, source code, architecture descriptions, and so on. As a result, the notion of user extends to operators as well as to programmers, which are users of components such as software libraries.

Over the past decades, cases of student plagiarism in higher education have increased substantially (Dee & Jacob, 2012). Therefore, clear academic staff guidelines for detecting and tackling plagiarism should be introduced in higher education institutions (HEIs). However, plagiarism is a concept that is not that straightforward and can be interpreted from a myriad of perspectives (Bretag, 2013). Moreover, plagiarism is not a legal term as such given that it does not per se entail a violation of copyright law (Stearns, 1992). Students and staff are often not familiar with the diverse complexities of the description of plagiarism. Hence, students enrolled in a higher education program fall into the plagiarism trap quite regularly. As a result, some acts of plagiarism can be merely due to ignorance.

This related literature is anchored to “An Intelligent Tracking System Based on GSM and GPS Using Smartphone” which was invented by Vigneshwaran.K, Sumithra.S, and Janani.R on May 2015. Vehicle tracking system is a miniature model of Global Positioning System (GPS). GPS is used to find out the position or location of the vehicle around the world. This implementation introduces an Android based tracking and theft prevention system. The software for the microcontroller interfacing with GSM, GPS is written in Embedded C, that code run through the AVR studio. This system mainly approach in four wheelers. Even if it is GPS enabled vehicle, the researcher can only track and monitor the vehicle. Researcher cannot stop the stolen vehicle. The objective is to track the stolen two wheeler and to stop it using AIR SOLENOID and WATER SOLENOID. Automobile theft has increased in the world and also in India. According to

the NCRB data, 8,171 motor vehicles were stolen in Delhi in 2012. It is a serious problem in automobile field. This issue is a motivation to design this system.

The proposed aim is to implement miniature model by using a single chip microcontroller in the vehicle. GPS may be fixed in the vehicle to monitor and to find out location of the vehicle. With the help of the GPS value, the distance can calculated with respect to time. The two wheeler engine is an internal combustion engine. The generated power may be stored in battery used in two wheelers. In this system GPS, GSM is interfaced with atmega162 v microcontroller. A 16x4 LCD display is used to show some message to the user. First the thermal electrical generator is attached with battery. Small amount of power is generated from the silence of the bike. The power supply is given to the system.



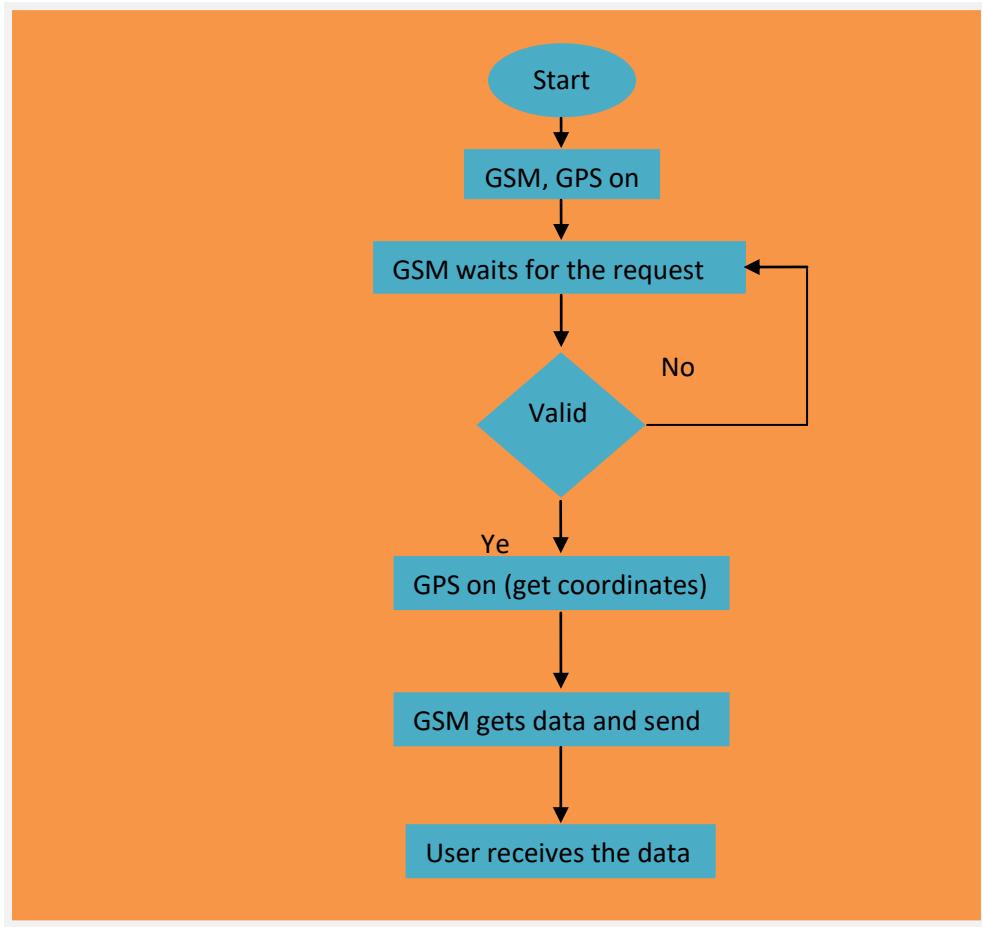
**Figure 2. Black Diagram of Security System**

Other related literature is “Design and Development of GPS-GSM Based Tracking System with Google Map Based Monitoring” by Pankaj Verma, J.S Bhatia on June3, 2013. In this urban life transportation is very common. A lot of miss happenings occur on the road every day .Therefore the need of security and monitoring is developed. To resolve such problems, a system is developed using GPS and GSM technologies and an application is introduced in this research work. Various problems that we face:

1. In critical condition (when vehicle is stolen), one is confused what to do
2. If one has something expensive and he wants to check it regularly
3. To find the shortest path available

All these problems are overcome by the system. This system has Global Positioning System (GPS) which may receive the coordinates from the satellites among other critical information. Tracking system is very important in modern world. This can be useful in soldier monitoring, tracking of the theft vehicle and other various applications. The system is microcontroller based that consists of a global positioning system (GPS) and global system for mobile communication (GSM). This project uses only one GPS device and a two way communication process achieved using a GSM modem. GSM modem, provided with a SIM card uses the same communication process as we are using in regular phone. The mobile number of the user should be included in the software programming in order to receive the location values from the SIM card which we are using in GSM modem. The NMEA protocol consists of set of messages. These messages are ASCII character set. GPS receives data and present it in the form of ASCII comma – delimited message strings. ‘\$’ sign is used at the start of each message. The locations (latitude and longitude) have the format of degrees minutes and decimal minutes. The software protocol consists of the GGA (global positioning system fixed data) and GLL (geographic position latitude/longitude). But in this system we are using

CGA only. The monitoring unit consists of a GSM mobile and a Web Application. The GSM mobile may acquire the position of the vehicle (longitude and latitude) and then by typing those co-ordinates in web application owner of vehicle can get the exact location of the vehicle. The flow chart of the system is given as:



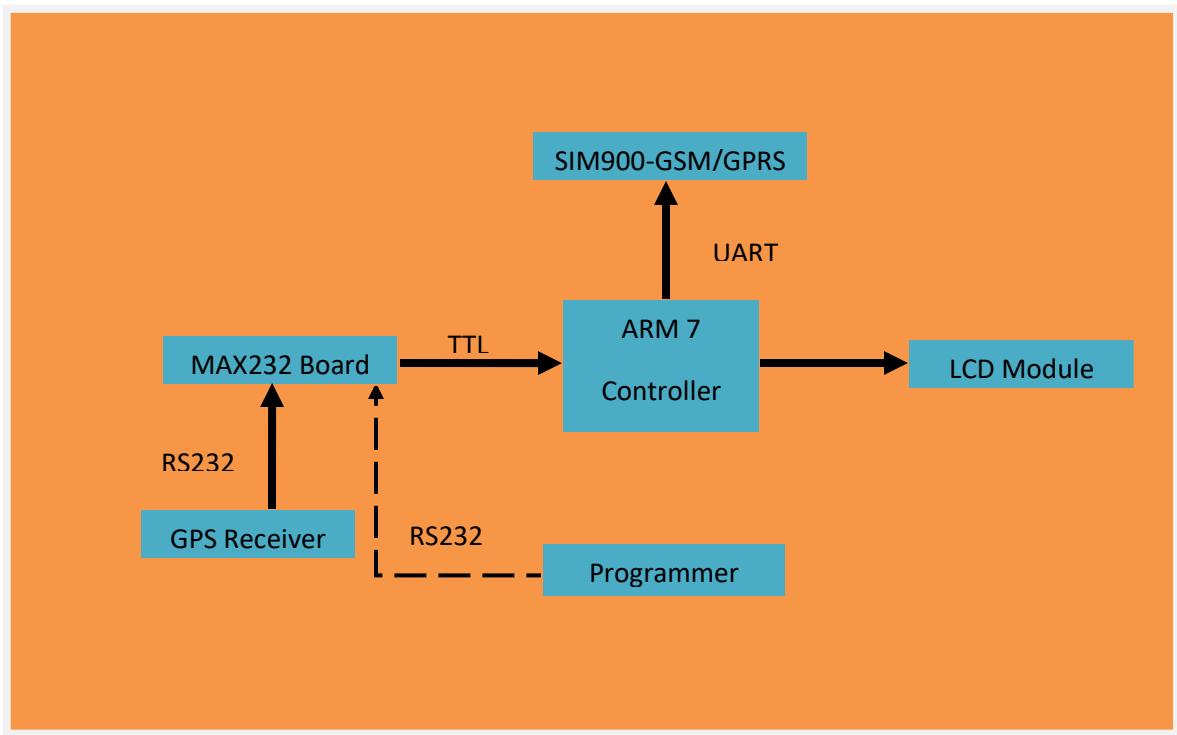
**Figure 3. Program flow chart of the Tracking system**

The web application named as ‘Tracking System’ is shown in figures that represent the complete output of the system. In this system two applications are developed that are linked to each other. First one is used to get the initial position of the

vehicle (starting point) and as system may receive the different co-ordinates (longitude and latitude) switching to the next one may be done to get the distance travelled b/w the two positions.

Another related literature is “Real Time Position Tracking System Using Google Maps API V3” by MihirGrarude, NirmalHaldikar on (September 9, 2014). Vehicle Sensor networks (VSN) are emerging as a new tool for effectively monitoring the physical world, especially in urban areas where a high concentration of vehicles equipped with on board sensors is expected. A vehicle tracking system combines the installation of an electronic device in a vehicle, or fleet of vehicles, with purpose-designed computer software at least at one operational base to enable the owner or a third party to track the vehicle’s location, collecting data in the process from the field and deliver it to the base of operation. Several types of vehicle tracking devices exist. Typically they are classified as “passive” and “active”. “Passive” devices store GPS location, speed, heading and sometimes a trigger event such as key on/off, door open/closed. Once a vehicle returns to a predetermined point, the device is removed and the data is downloaded on a computer for evaluation. Passive systems include auto download type that transfer data via wireless download. “Active” devices also collect the same information but usually transmit the data in real-time via cellular or satellite network to a computer or data-center for evaluation. The sensors report vehicle status information to our server, which can be useful for information processing and for intelligent tracking management. Also, these systems are designed using a proprietary software and hardware. Due to the high cost of these systems, urban transit agencies cannot afford the systems. To overcome such problems we have proposed a system designed using open source software and hardware, keeping the cost of the system to a bare minimum.

GPS tracking of vehicles is not very new, but the system highlights the improvements in end user experience. End user experience is of prime importance, and display of GPS coordinates of a vehicle on a dedicated website enhances it. It allows a fair access to all users. This system can be provided as a premium service to consumer, which may help it, make sustainable. Alternatively releasing its code open-source under GNU-GPL V3 may allow developers to build further on this project by predicting highly localized observational parameters like environmental pollution, local weather and humidity point, etc. Finally this system provides a low cost solution to localized data acquisition which is highly scalable.



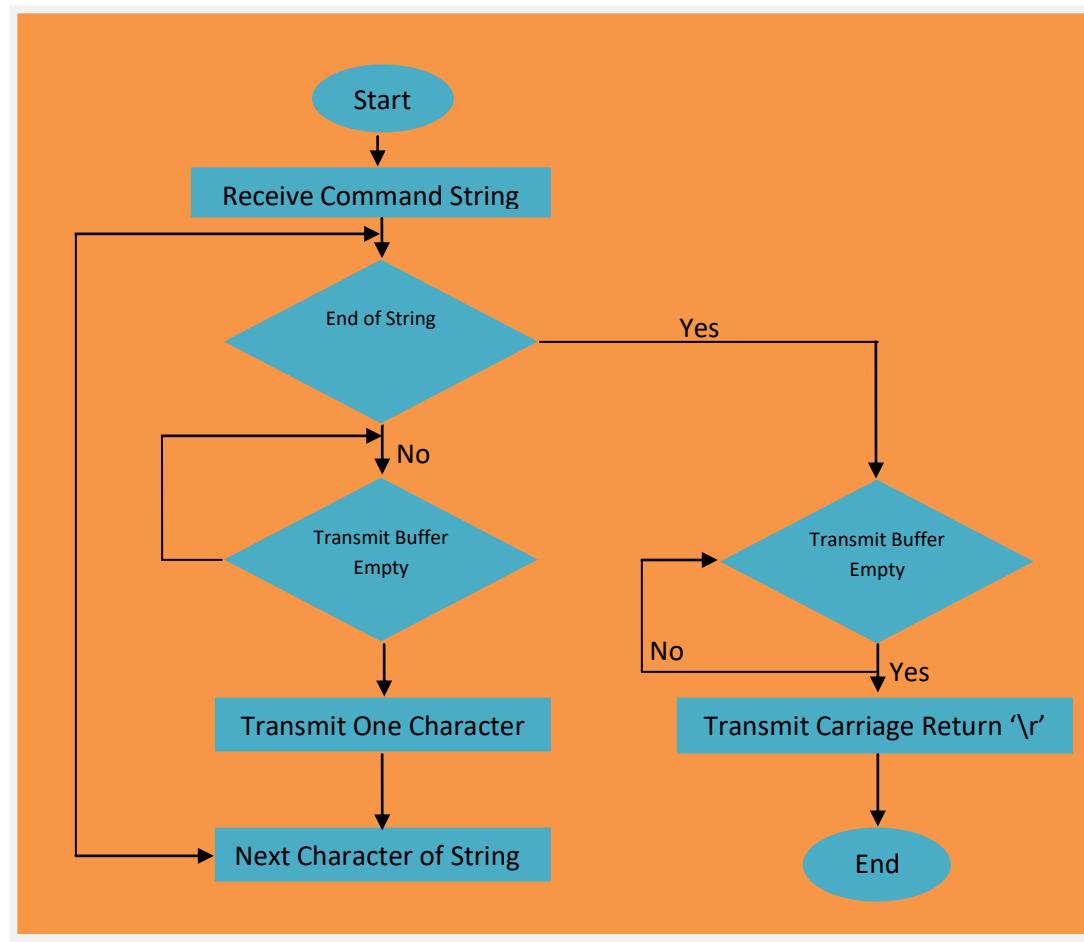
**Figure 4. Block Diagram and Working of the Proposed System**

This literature is related to “Integrating GPS, GSM and Cellular Phone for Location Tracking and Monitoring” by B. P. S. Sahoo and Satyajit Rath on (February 29, 2012). The wide spread of mobiles as handheld devices leads to various innovative

applications that makes use of their ever increasing presence in our daily life. One such application is location-tracking and monitoring. This paper proposes a prototype model for location tracking using Geographical Positioning System (GPS) and Global System for Mobile Communication (GSM) technology. The system displays the object moving path on the monitor and the same information can also be communicated to the user cell phone, on demand of the user by asking the specific information via SMS. This system is very useful for car theft situations, for adolescent drivers being watched and monitored by parents. The result shows that the object is being tracked with a minimal tracking error.

Due to the ripeness in the wireless technology, location-tracking of objects and people in indoor or outdoor environments has received ample attention from researchers lately. There are various methods for identifying and tracking user position such as Cricket, Mote Track or GPS. GPS offers a scalable, efficient and cost effective location services that are available to the large public. However, the satellite emitted signals cannot be exploited indoors to effectively determine the location. The aim of this research is to track a user position in both indoor and outdoor environments with a minimal tracking error by incorporating a trans-receiver in the device which to be monitored. The tracking system can also be very useful for Intelligent Transportation System (ITS). For example, it can be used in probing cars to measure real-time traffic data to identify the congesting area. In addition, it can be attached to a vehicle with an anti-theft system to identify its location when it is stolen. Our proposed system offers a real-time tracking system using a client-server approach. Our client can be any device, incorporated with the tracking antenna to identify device location information that is periodically transmitted to the base station via wireless medium. Our base station is a personal computer connected with a GPS receiver, GSM modem and interfacing software module

written in C to filter the location information like altitude, longitude, speed, time and etc. This system has also another facility to monitor the path along which the device moves. From application point of view, this environment tracking scheme can be utilized for various applications such as locating in-demand personnel like doctors or patients with vital sign sensor in hospital environment, Army, Air force, etc. It can also serve as a basis for context-aware application.



**Figure 5. Flow Chart Subroutine sends AT command**

Another literature that premised to this “Design of Intelligent solar Tracker Robot for surveillance” by B.Suchitha Samuel1, J.Mrdula, Geethanjali, Keesara, and Andhra

Pradesh on (October 10, 2013). It is often observed that it is difficult to carry out manual surveillance in various scenarios where either the conditions are hostile for sending in personnel (fires, hostage situation) or the available man power is insufficient to dispatch workforce at our disposal (patrolling large complexes). Also the recharging of robots in war field or surveillance robots in far off locations often becomes a problem. In this situation it becomes imperative to have a platform that can negotiate these difficulties while still remaining economically viable. A mobile microcontroller based surveillance platform is one such option. Solar tracking for robot/vehicles is mostly done using fixed solar panels which is disadvantageous. This paper presents the hardware design and implementation of a system which performs intelligent solar tracking and powering the robot with solar energy that can be used for surveillance and monitoring operations. This paper aims to present a microcontroller based surveillance platform that is effective as a surveillance tool in both civilian and military applications. The main aim of our project is to make the panel to rotate according to the sun's direction from morning to evening automatically so that the panel grabs the solar energy to maximum extent possible throughout the day. Intelligent Solar Tracking System is used to generate power from sunlight and it can be used by storing the generated power. This method of power generation is simple and is taken from natural resource.

The unique feature of the proposed system is that instead of taking the earth as its reference, it takes the sun as a guiding source. Its active sensors constantly monitor the sunlight and rotate the panel towards the direction where the intensity of sunlight is maximum from morning to evening automatically, so that the panel grabs the solar energy to maximum extent possible throughout the day. When there is decrease in intensity of light, this system automatically changes its direction to get maximum intensity of light. The power generated from this process is then stored in a battery. The

controlling device of the whole system is a PIC Microcontroller .The light dependent resistors do the job of sensing the change in the position of the sun which is dealt by the respective change in the solar panel's position by switching on and off the geared motor. The voltage from solar panels will be displayed on LCD by interfacing solar panel to PIC microcontroller through a voltage measuring circuitry.

### **Studies**

This study is promised to “Intelligent Students Tracking system in campus based on RFID and Zigbee” by P.S.Kiran kumar, and Dr. Shankaraiah on (august 8, 2014). With the rapid increase in the number of institutions as well as number of students in every institution, it is very difficult to monitor whether the students are attending the classes as per the time table or not. Even the students are in the campus but not visible in the classes during the class hour. In the existing system, presence of students in the class is being monitored by taking attendance manually in the class.

According to Kumarl & Shankaraiah (2014) paper, they propose an intelligent student tracking system based on RFID and ZigBee wireless network. Kumarl stated that the proposed system reads the RFID tags data through RFID reader, and then sends it to PC node by a custom wireless protocol on the ZigBee. PC node gives corresponding warning or hints by the result of matching master-slave RFID tag information. Here each and every student is assigned with different RFID tags with individual tags numbers and the RFID readers are placed in campus at different locations which identifies the persons coming to that location by reading tag numbers of the students and intimates to the control room about their status.

As the number of institutions as well as the number of students in each institution is increasing it is harder to monitor the presence of students in class during the class hours. In existing system students should be monitored continuously whether they are

attending the classes or not manually in the classes by means of taking attendance but there are many chances of student escaping from different ways. In the proposed paper, an intelligent campus student tracking system may be designed and implemented based on the RFID and ZigBee network which keeps track of where the student moves within the campus. The system reads the RFID tags data through RFID reader, and then sends it to PC node by a custom wireless protocol using ZigBee. PC node gives corresponding warning by verifying with the time table stored in the database and updates the report table appropriately if the student is anywhere else other than classroom during class hours. Here each and every student is assigned with different RFID tags with individual tags numbers and the RFID readers are placed in campus at different locations which identifies the students moving from place to place automatically by reading tag numbers of the students and intimates to the central server about their status.

The proposed system would help to keep track of whether the students are regularly attending the classes or not. The main aim of this paper is to keep track of student's movement within the campus and also to provide a warning/update report table if student is outside the classroom during class hours as per the timetable stored in the database. Several technologies are available using the angle and distance can be measured. In a typical TAG management system many of the TAGs shall be operating in the same neighborhood and therefore forms an interference network, thereby effecting the identification of the location of the TAG. Angle and distance from a reference point along may not give the exact location of the subject. The directions also are equally important to locate an object most accurately.

Another related study is “Location- based Communication Systems a Look at Intelligent Networking and Privacy Concerns” by Laurie Thomas Lee on (September 19,

2012). Location-based networks are quickly becoming an integral feature of today's intelligent networking landscape. They are transforming how people and government communicate. Global positioning satellite systems, Wi-Fi, and other such decentralized networks make it possible to track and physically locate users and pair that information with location-based information from and about others, such as friends, products, and services. This paper examines the nature of location-based systems in the context of intelligent networking, where various technologies, pathways, and information repositories operate in a decentralized environment for the purpose of providing location-specific communication and information. Different types of technologies, routes, and databases are described and discussed. Privacy issues are of particular concern, given that the advanced features of an intelligent network make it easy and economically compelling to track individual locations without the person's knowledge or consent. Other problems include the ability to invade privacy in a manner that is continuous, global, and permanent.

Laurie Thomas Lee (2012) said that location-based networks are quickly becoming an integral feature of today's intelligent networking landscape. They are transforming how people and government communicate. Global positioning satellite systems, Wi-Fi, and other such decentralized networks make it possible to track and physically locate users and pair that information with location-based information from and about others, such as friends, products, and services. Users with Smartphone's, for example, can access local maps, check out nearby restaurant specials, notify friends as to their whereabouts, find others, and get information on local traffic alerts, events and landmarks, all by accessing the various types of networking systems and applications that rely on location tracking. Location-based systems present considerable economic efficiencies and other advantages. Researchers predict phenomenal growth for the

“geolocation” business, with revenues topping \$4 billion by 2015 (With Location-Tracking, 2010). Yet this type of intelligent networking also possesses serious security and privacy concerns, as location may be easily tracked and disclosed without knowledge or permission and with economic and other consequences. In this case, location-based networks are capable of tracking and communicating the whereabouts of a user to other users across different and complex networks and platforms in an instant. Decision-making capability and analysis are also a part of location-based networks, whereby intelligence is added by the contributions and actions of the various users on the networks. As with organizational intelligence (Liebowitz, 2000), knowledge is created and strategically used for a benefit, such as improved organizational performance.

Location-based systems in the context of intelligent networking, where various technologies, pathways, and information repositories operate in a decentralized environment for the purpose of providing location-specific communication and information. These different types of technologies, routes, and databases are described and distinguished as they pertain to location-based networking. The various types of users involved are also identified because of their contributions to the intelligence of location-based networks. Finally, because intelligent networks by their nature present both advantages and disadvantages, the opportunities of location-based networks as well as specific problems such as location privacy violations are discussed.

This study is anchored to “A Smart Location Based Time and Attendance Tracking System Using Android Application” by Shermin Sultana, Asma Enayet and Ishrat Jahan Mouri on (February 2015). Over the years the process of manual attendance has been carried out which is not only time consuming but also provides erroneous result. Automated time and attendance monitoring system provides many benefits to organizations. This reduces the need of pen and paper based manual attendance tracking

system. Following this thought, we have proposed a smart location based time and attendance tracking system which is implemented on android mobile application on Smartphone reducing the need of additional biometric scanner device. The location of an organization has a specific location, which can be determined by the GPS. Each employee's location can be determined by the GPS using smart phone. This location is defined as a key of time and attendance tracking in our paper. Nowadays, attendance monitoring and working hour calculation are very essential for almost every institution or organization. Typically there are two types of attendance system available, Manual and Automated. Manual system involves the use of sheets of paper or books in taking attendance where employees fill out and managers oversee for accuracy. This method could be erroneous because sheets could be lost or damaged. Also the extraction of relevant data and the manual computation of working time is very time consuming. It takes an extra employee to check for the attendance and timing of other employees which includes cost overhead for the organization as well.

On the other hand Sultana (2014) stated that automated time and attendance systems imply the use of electronic tags, barcode badges, magnetic stripe cards, biometrics (hand, fingerprint, or facial), and touch screens in place of paper sheets. In these aforementioned techniques, employees touch or swipe in order to provide their identification and also the entering and leaving time to calculate working hours. The provided informations are recorded and automatically transferred to a computer for processing. Using an automated system for time and attendance monitoring reduces the errors of manual system and conserve optimal amount of time. But these automated systems require heterogeneous devices need to be located in the organization which is costly.

Mouri (2015) and co-researcher are considering the wide popularity of Smartphone's, we introduce the use of Smartphone for this time and attendance tracking purpose. **We have proposed** a location based smart time and attendance tracking system based on the concept of web services which is implemented as an Android mobile application that communicates with the remote server in which the database is located. Internet connectivity (Wi-Fi/3G) is needed for connecting to database residing in the remote server. Any employee crossing the area border with a Smartphone and the running application may be tracked automatically.

Another premised to this study is "GPS and RFID Based Intelligent Bus Management and Monitoring System" by Anuradha Vishwakarma, Agraja Jaiswal, Ashwini Neware, Shruti Ghime, Antara Marathe, and Asst. Prof. Rashmi Deshmukh on (March 7, 2016). Now a day, a bus tracking system has been widely used for fleet (group of buses) management and asset tracking. GPS and RFID Based Intelligent Bus Tracking System is useful, accurate, robust, flexible and economical. In this work, GPS is used for obtaining the location of the bus using Google Map and RFID is used for the identification of bus. RFID card is also used here for transactions related to fuel filling. Therefore, whole information of a particular bus is stored in database at the server side.

According to Deshmukh (2016) an ultimate aim of this system is to develop a flexible, cost effective and user-friendly vehicle tracking system that can cater the needs of owner of transportation companies with minimum technologies backing at the user end. In developing countries, such technologies are very useful for tracking of vehicles. Current tracking processes that are in use, such as barcode scanning software, in situations where hundreds of bus are present, the act of scanning each individual bus to check for proper placement can become a tedious and inefficient use of company time and energy. A successful and complete prototype was designed and implemented over the

course of this project. The prototype properly tracks any number of target vehicles through RFID tag scanning and GPS location checking, entered vehicle information. Individual bus details can be accessed by the user through various menus that are provided during the running of the program. This may allow the prototype to be adapted to many different areas of use, instead of a predetermined work environment.

The Global Positioning System (GPS) is a spacebased navigation system that provides location in all weather conditions. GPS satellites continuously transmit their current time and position. A GPS receiver monitors multiple satellites and solves equations to determine the exact position of the receiver and its deviation from true time. At a minimum, four satellites must be in view of the receiver for it to compute four unknown quantities (three position coordinates and clock deviation from satellite time).A radio-frequency identification system uses tags, or labels attached to the objects to be identified. Two-way radio transmitter-receivers called interrogators or readers send a signal to the tag and read its response. RFID tags can be either passive or active. Passive tags collect energy from a nearby RFID reader's interrogating radio waves.

## Chapter 3

## **RESEARCH METHODOLOGY**

This chapter presents the methodology of the study. It involves the discussion of the research method, research environment, project development process, project design, data flow diagram, system theory and concept, project testing method and procedure, project evaluation procedure and calendar of activities.

### **Method Used**

The researchers used the development research method which is commonly used by the computer science student. It makes the previous study more interesting and useful to all students. The point of the method is to broaden the concept into a higher level so that the designers can improve its usability towards students. This involves evaluating the system (Intelligent IP Address Locator via Google Maps), documentary analysis wherein the data gathered from the internet and other resource documents were carefully analyzed in order to have desired results and answer of the problem.

### **Research Environment**

The researchers conducted the study at Jose Rizal memorial State University – Main Campus. The researchers selected the place where equipment can easily be available when needed for the project and of course with the internet connection for online research of related information. Mostly a place had personal computer or either a laptop and other devices which can be used for data processing in making the project.

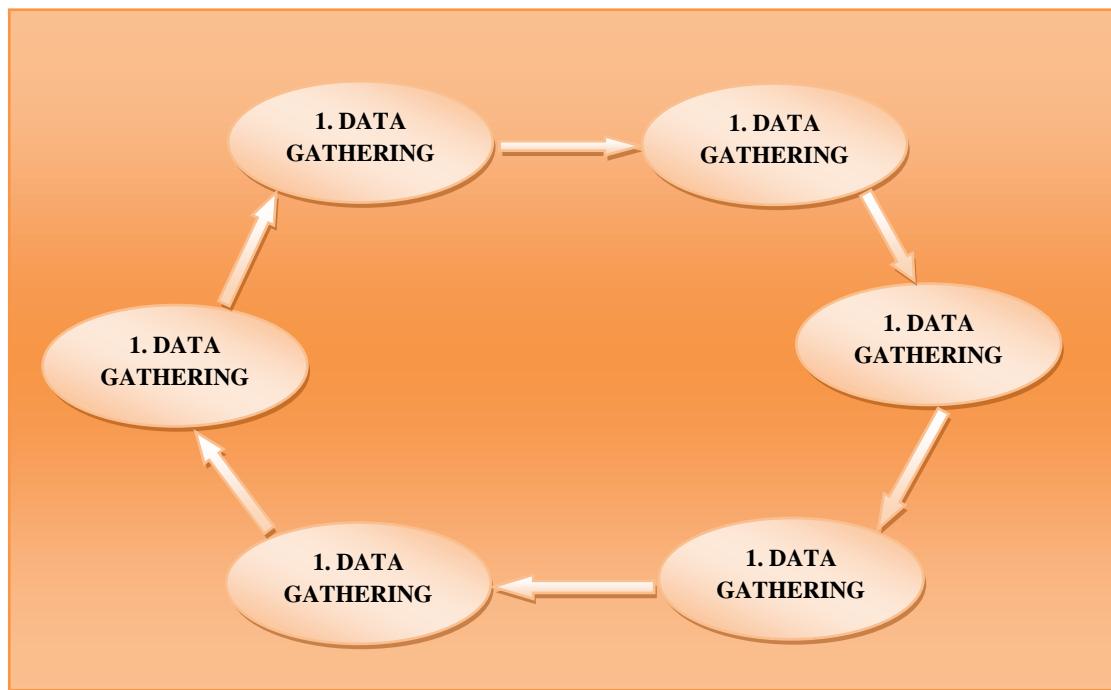
### **Project Development Process**

The project development process is initiated in response to an identified need in the transportation system. It covers a range of activities extending from identification of a project need to a finished set of contract plans, and to construction. According to the

explanation the Project Life Cycle refers to a logical sequence of activities to accomplish the project's goals or objectives.

It consists of events which are necessary to complete a project. While the system Development Life Cycle used to develop and maintain Information System. It focuses on the software engineering phases, process, tools and techniques for building and/or implementing the IT solution. The phases are: Data Gathering, Analysis, Designing, Development, Integration, Testing and Implementation.

Figure 7 on the next page tells about the cycle of the system where the individual process verified every output of the study. It gives information to everyone who uses this study for their proposal method and can help them verify the procedure and process of the system “Intelligent IP Address Locator via Google Map”.



**Figure 7. Project Development Life Cycle**

**Data Gathering.** The researchers conducted a study on how the Intelligent IP Address Locator via Google Maps operates in Jose Rizal Memorial State University – Main Campus, Dapitan City. The researchers may conduct interviews and sought advice from the internet which was used as bases to create the Intelligent IP Address Locator via Google Maps.

**Requirements Analysis.** After gathering all necessary requirements, the researchers with their focus on Intelligent IP Address Locator via Google Maps, researcher made the best to have a better output. The researchers then analyzed all the requirements that can build system software and can help the researchers to interpret all required and the possible output of the proposed study. The Requirement Analysis will be divided into different requirements as follows:

## **1. Hardware Requirements**

### **A. Computer**

Processor – Dual Core

Hard Disk – 50 GB

Memory – 1GB RAM

### **B. Software Requirements**

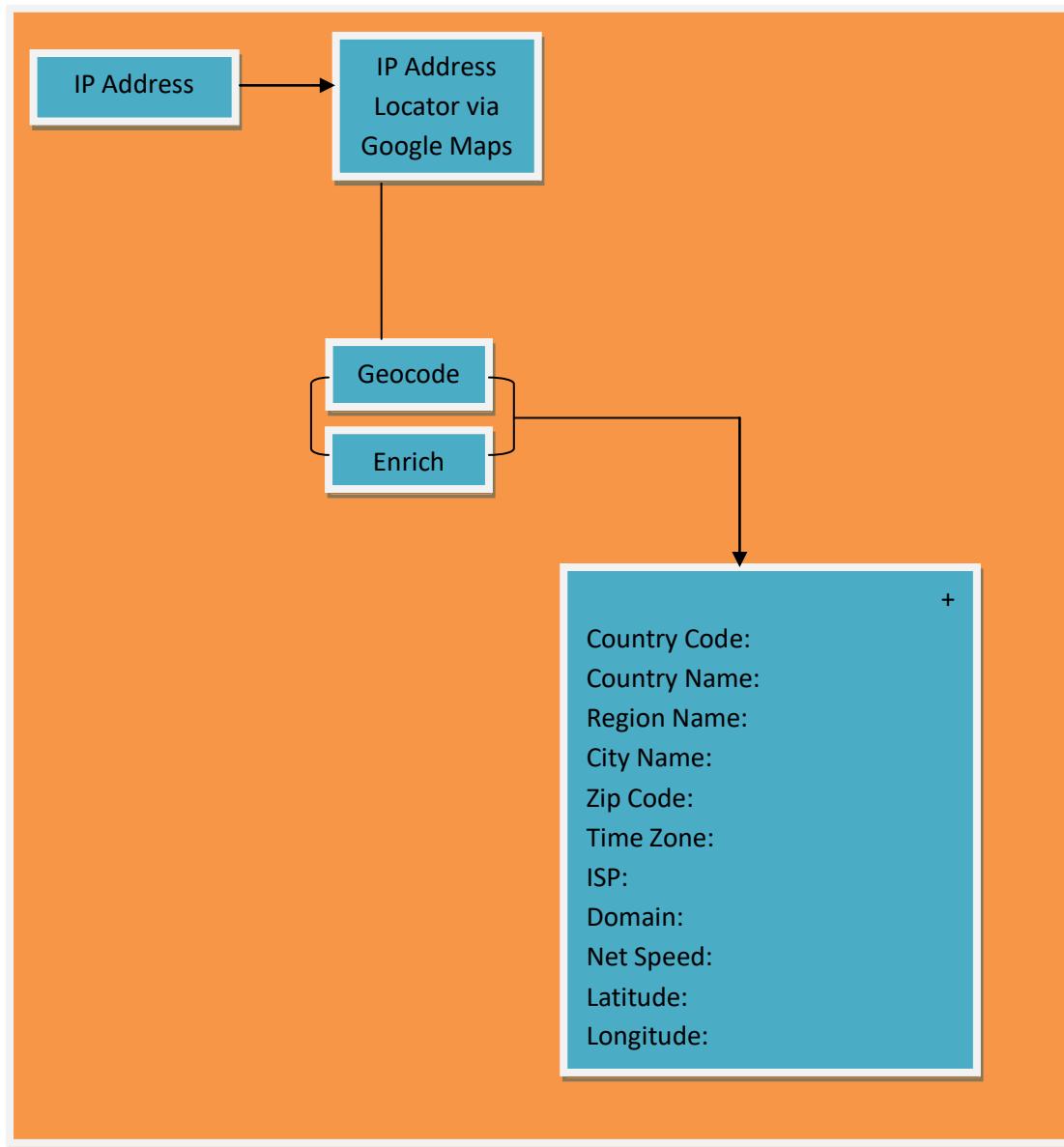
Windows 7

Sql 2008

Visual Studio 2010

**Designing.** The researchers then make a flowchart, techniques, principles and abstract presentation of the study based on the hardware requirement if it is compatible to meets the desired software requirements. Researcher refers to the Information Technology Instructor to have information's, opinions and concept about the proposed study, then

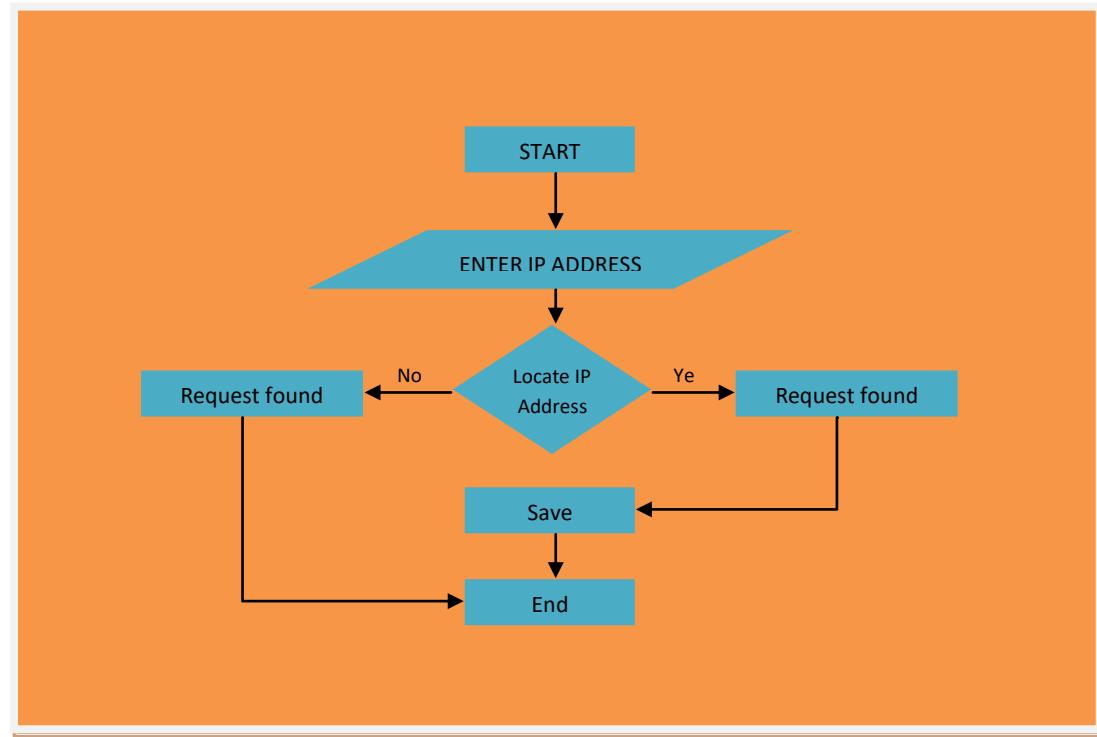
consolidating ideas to ensure that the process of principles design of the output have a functional and unique systematized designed.



**Figure 8. System Project Design Diagram**

Figure 8 illustrates the System Project Design Diagram for the study as well as the concept of the system.

## A. System Flow Chart



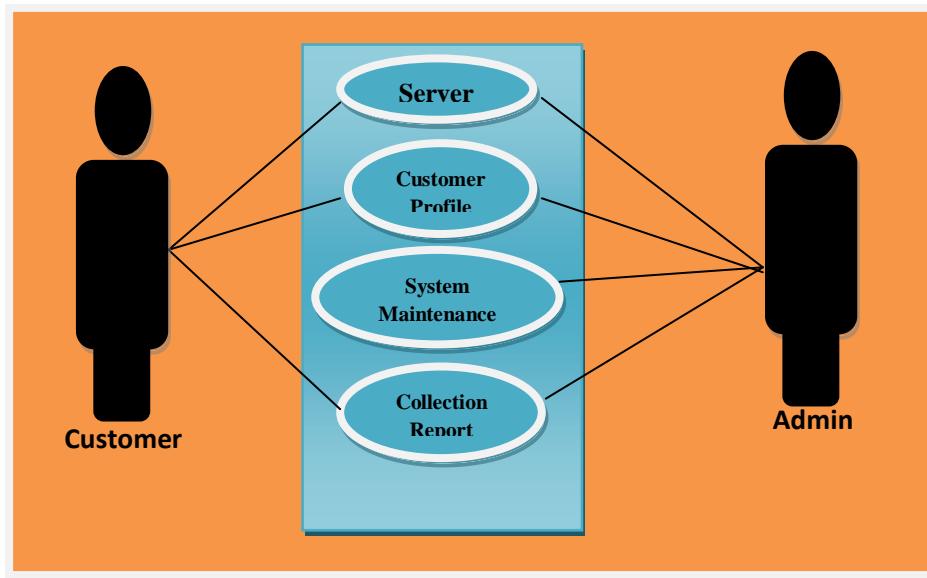
This Figure 9 explains the flow of the program that the System flowcharts are a way of displaying how data flows in a system and how decisions are made to control events. To illustrate this, symbols are used. They are connected together to show what happens to data and where it goes.

## B. Unified Modelling Language

UML is a standard language for specifying, visualizing, conducting, documenting, communicating, and business modeling. This is a standard notation for the modeling of real-world objects as a first step in developing an object-oriented design methodology. This is also a robust notation that can express information gathered throughout a project's lifecycle and can be used as an effective data modeling tool as well as an object modeling tool.

A use case is initiated by an actor to perform a required task. An actor is a person, organization, or external system that plays a role in one or more interactions with your system. Most of the time, an actor is a person, but that doesn't have to be the case. Sometimes, an actor can be another computer system—for example, when an external system requires a roll-up report processing, one might view that system as an actor, and the generation of that roll-up report as use case.

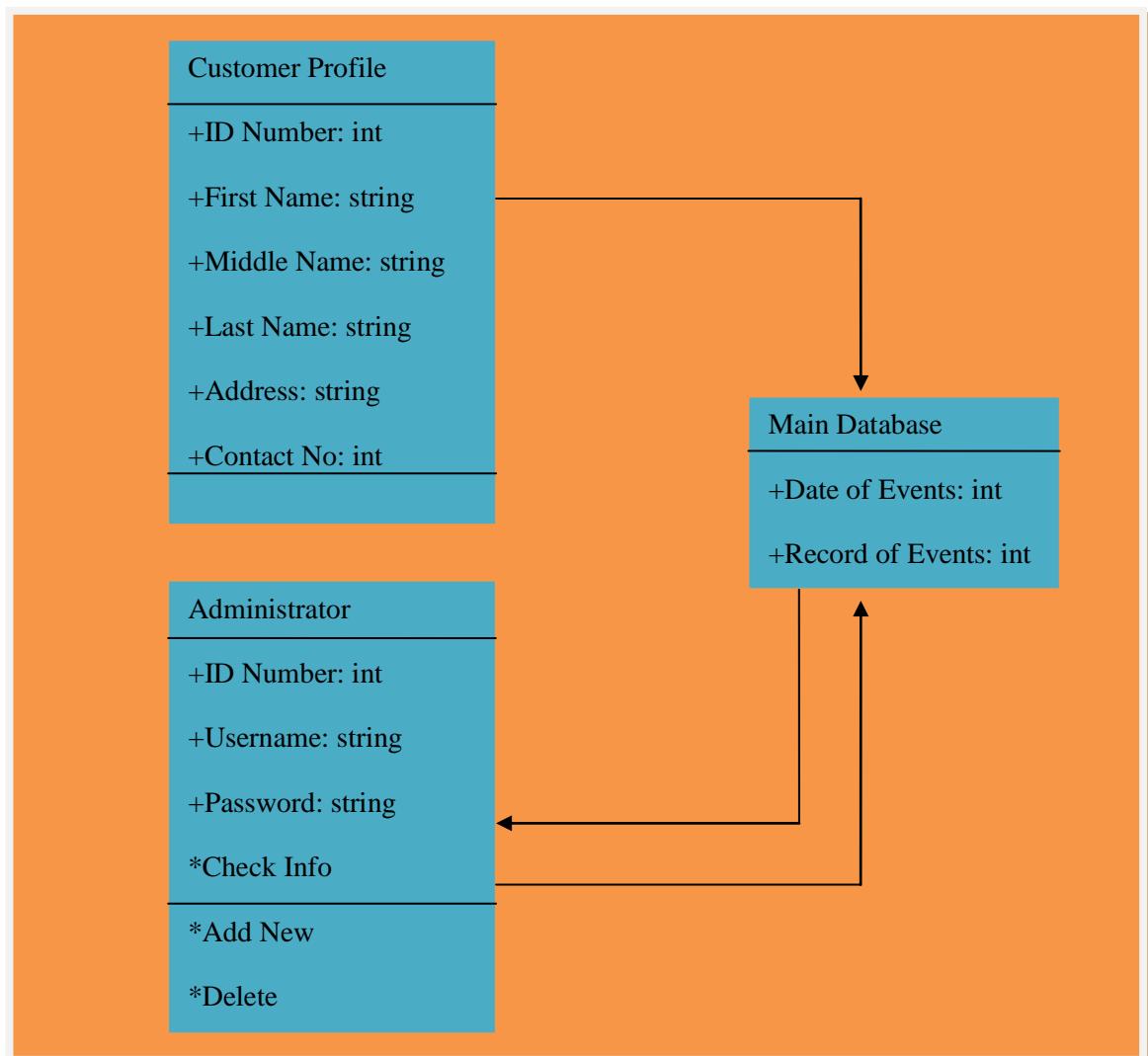
Figure 10. Is the Use Case Diagram of the Intelligent IP Address Locator via Google Maps that has two actors: the Admin and the Customers. The Admin has four use cases the Server, Customers Profile, System Maintenance, and Collection Report. While the Customers have three use cases, the Server, Customers Profile, and Collection Report, these three use cases are included so that customers can see the report if they have been connected and it prevents error and misunderstanding. System Maintenance is no longer included to the use cases of the customer because only the Admin may do. System maintenance is really important so that the proposed system can last longer and also it can prevent error.



**Figure 10. Use Case Diagram**

## UML-Class Diagram

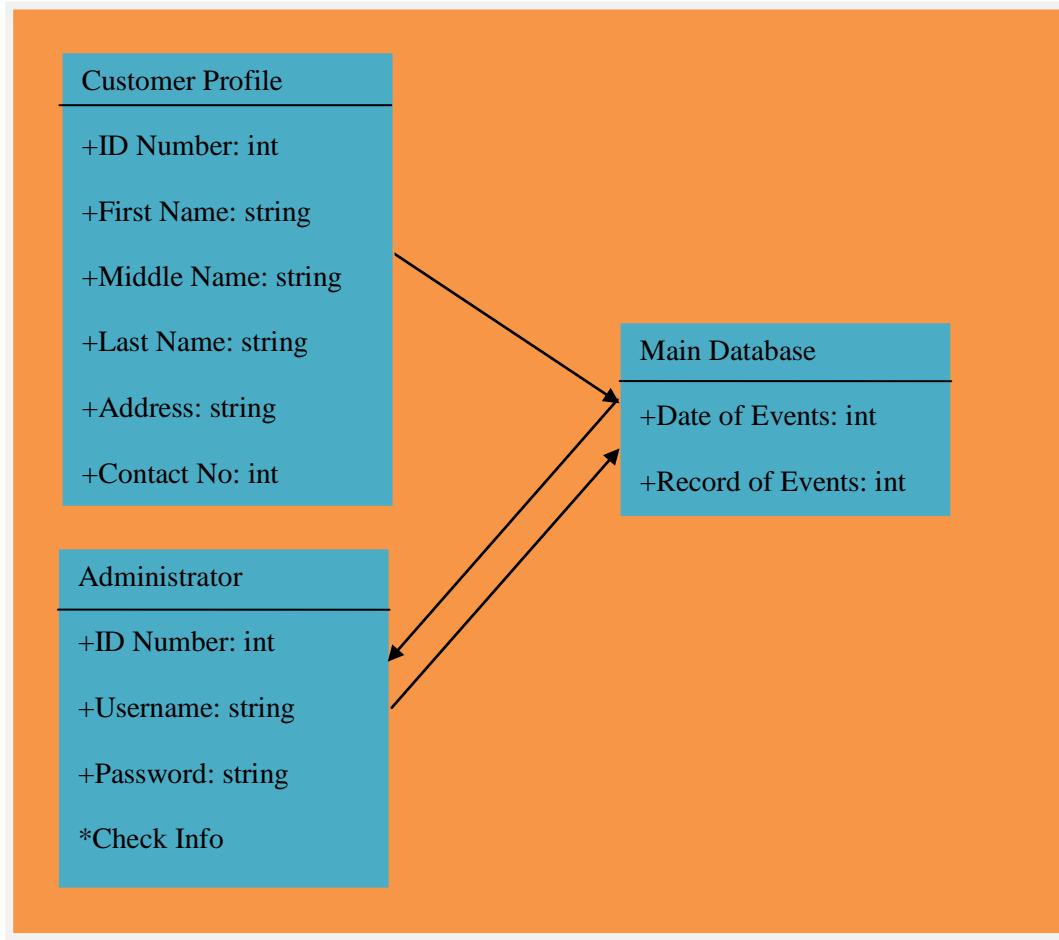
Figure 11 shows the Class Diagram of the system. It demonstrates the interrelationship, the operators, the attributes and the classes of the Administrator. The customer information includes the ID number, First Name, Middle Name, Last Name, Address, Contact No. Then the main database may record and also the dates of the events were it happened and the types of the events whether it is general and non-general event. And then the Administrator serve as to view, updates, and records all the information after tracing the identities of the customer if they can be trusted.



**Figure 11. Class Diagram**

## UML-Object Diagram

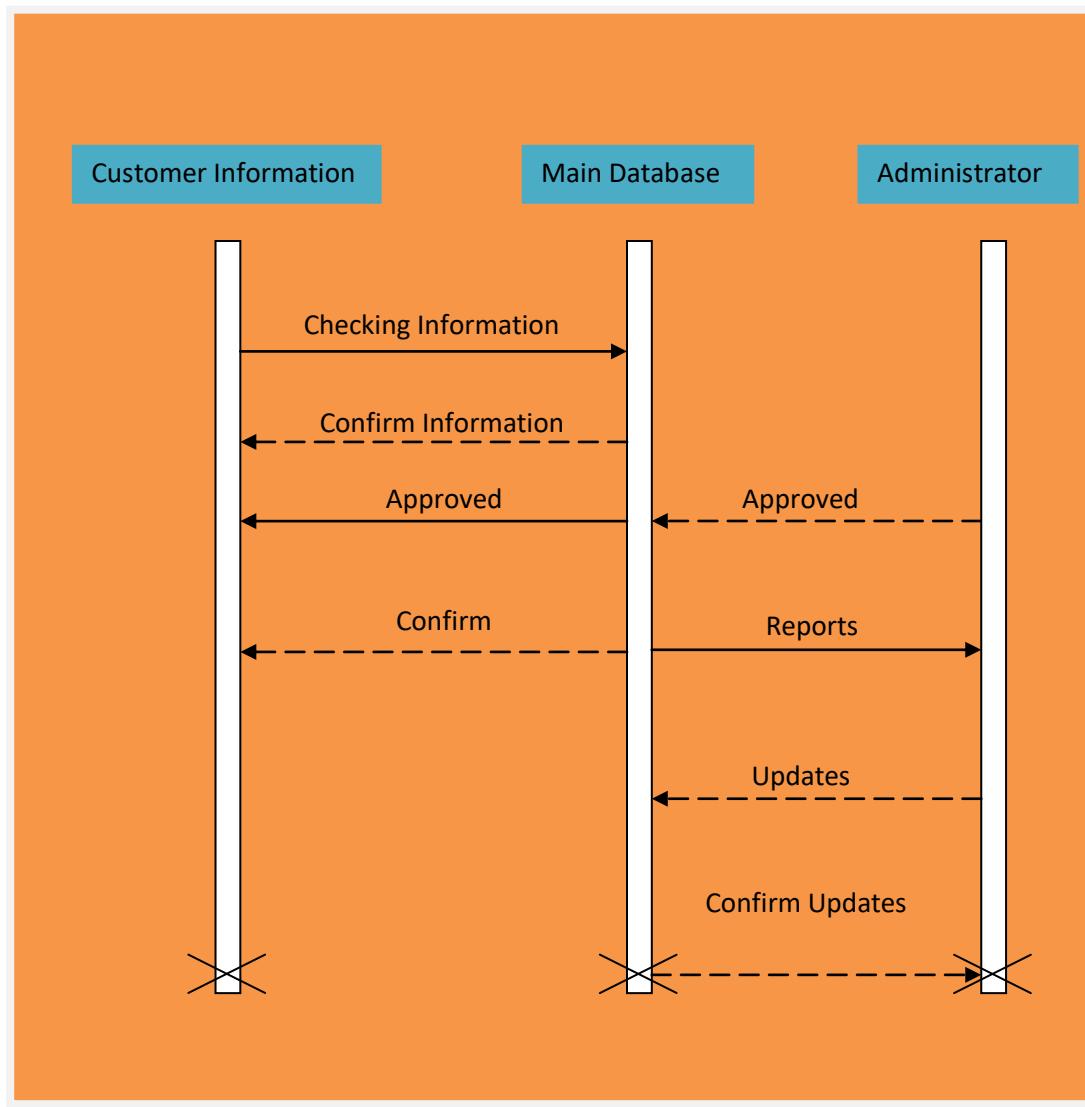
Figure 12 shows the Object Diagram of the system. It demonstrates the interrelationship, the operators, the attributes and the classes of the Administrator. The customer information includes the ID number, First Name, Middle Name, Last Name, Address, Contact No. Then the main database will record and also the dates of the events were it happened and the types of the events whether it is general and non-general event. And then the Administrator serves as to view, updates, and records all the information after tracing the identities of the customer if they can be trusted.



**Figure 12. Object Diagram**

## UML-Sequence Diagram

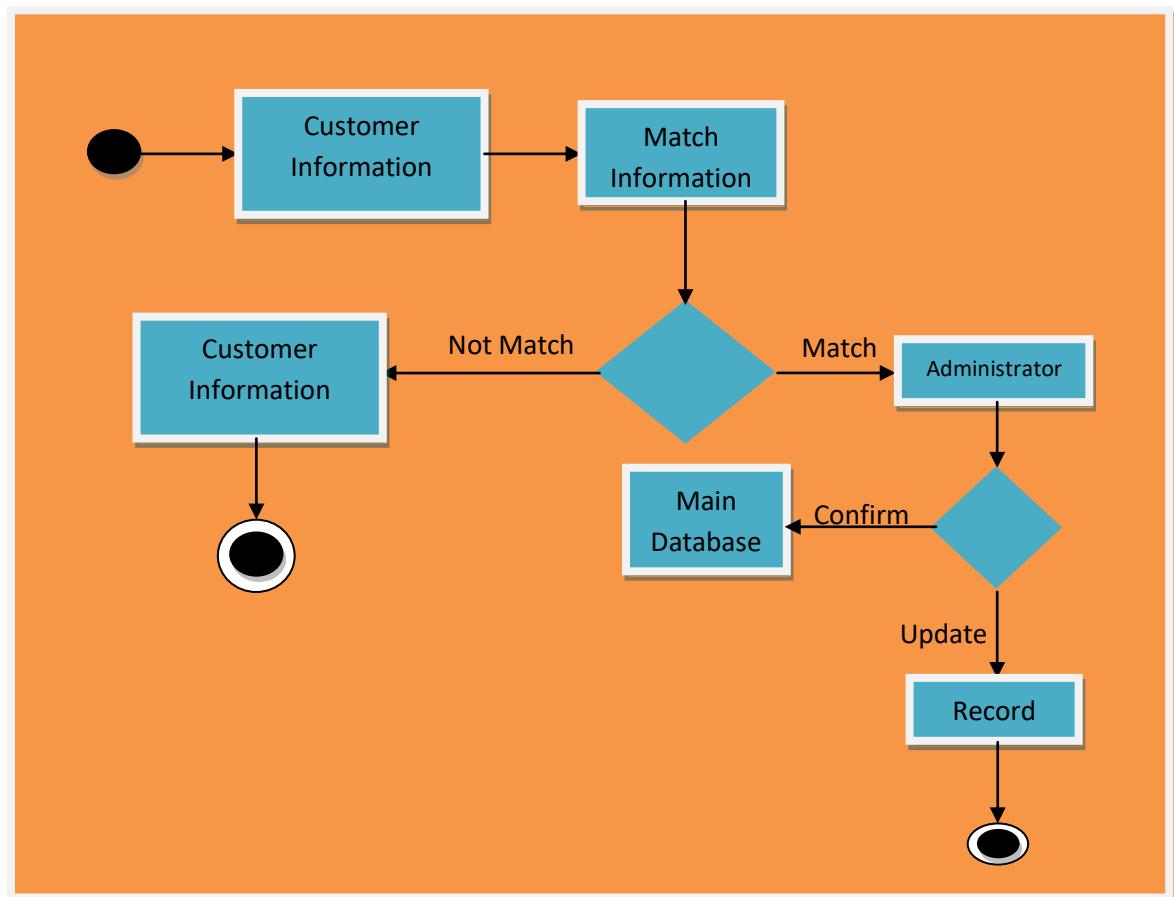
Figure 13 shows the diagram of the system. In customer information, the process conformation of information from the main database that may occur, then you can see the types of the events that may occur. From the main database to the administrator you can find the confirmed updates.



**Figure 13. Sequence Diagram**

## UML-Activity Diagram

Activity diagrams at figure 14 describe the work flow behavior of a system it is also another important diagram in UML to describe dynamic aspects of the system. Activity diagram is basically a flow chart that presents the flow from one activity to another activity. The activity can be described as an operation system. So the control flow was drawn from one operation to another. This flow can be sequential, branched or concurrent. Activity diagrams deal with all type of flow control by using different elements like fork, join etc.

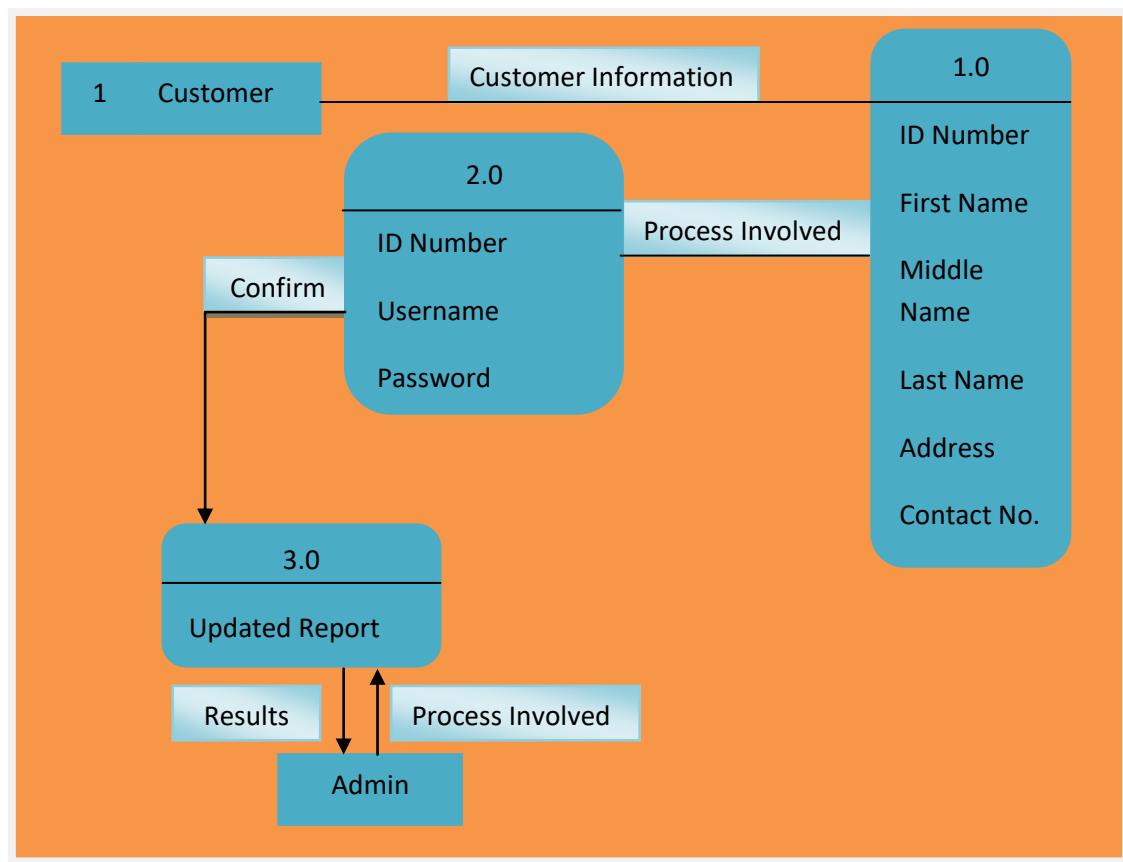


**Figure 14. Activity Diagram**

## Context Data Flow Diagram

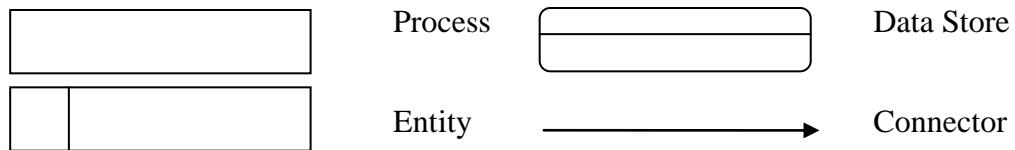
A data flow diagram (DFD) of the scope of organizational system that shows the system boundaries, external entities that interact with system and the major information flows between the entities and the system. Software engineering and systems engineering are diagrams that define the boundary between the system, or part of a system, and its environment, showing the entities that interact with it. This diagram is high level view of a system. It is similar to a block diagram.

Figure 15 shows the process of the system. First step is customer information, second step the customer connected the main database and the third step, and the administrator will automatically complete updates.



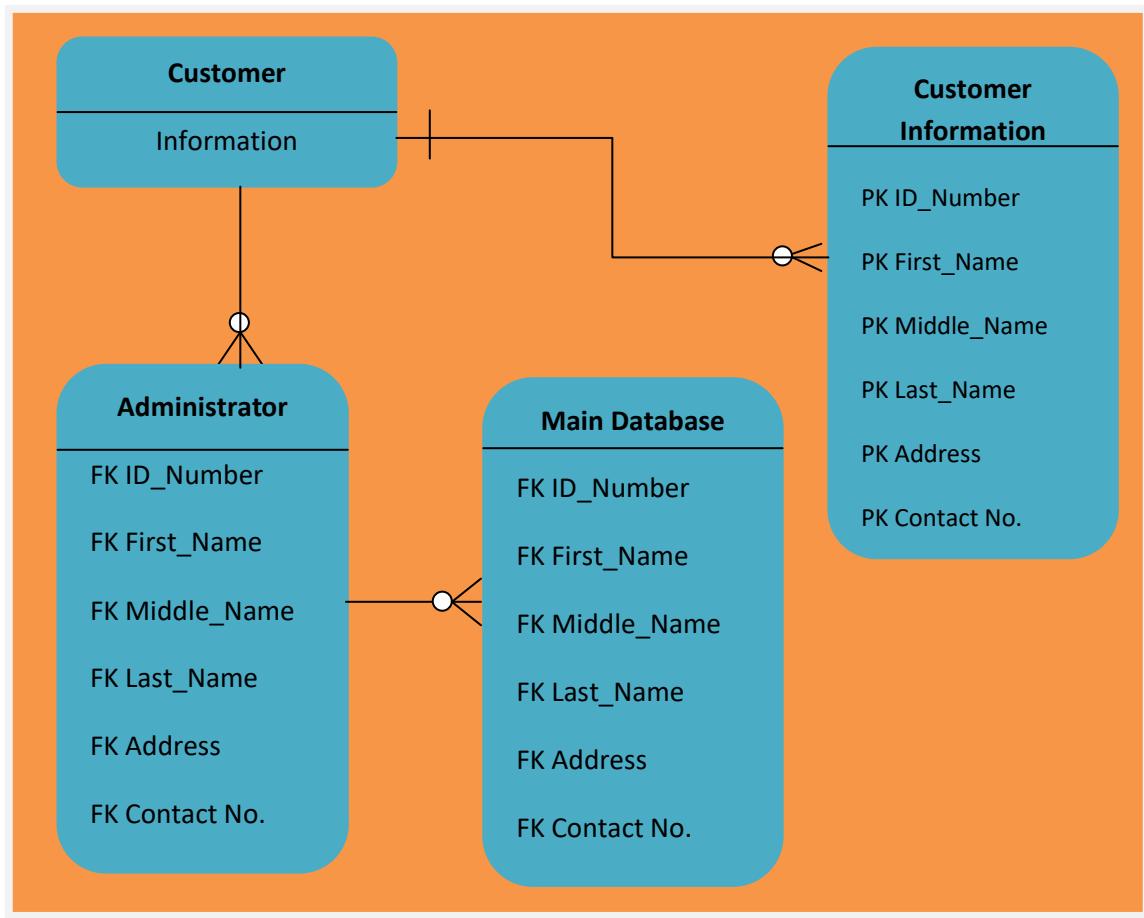
**Figure 15. Data Flow Diagram**

### Legend:



### A. Entity Relational Diagram

Figure 18 shows the components and the processes of the system on how it is being connected to each other and how this system function based on its features. The first entity is the customer information or profile. The second entity may be the customer's username and password being confirmed. The third entity is the administrator may confirm all updates from the customer.



**Figure 18. Entity Relational Diagram**

### **Legend:**



**Coding.** The researchers made the actual program based on the technique and principle of programming. Researchers set commands of instructions that the user could easily interpret to the software and has an application and a systematized software output that can execute through any computer.

The following are the method and techniques of the system:

**Looping** – used to provide additional time which in turn, enhances instruction and assessment.

**Function** – a piece of code which takes one more input in the form of parameter and does some processing and returns a value. It is a name given to a block of code that can be executed whenever needed.

**Arrays** – a systematic arrangement of objects, objects, usually in rows and columns. It is a special variable, which can hold more than one value at a time.

**Query** – used to connect from visual basic to MS Access/mysql. It is a command to save, edit, delete, and search data from the database.

**Testing and Evaluation.** In this process, the researchers with the concern of the user of the software test and evaluate the program to identify the correctness, completeness, security and quality of the developed computer software. The researchers used the process of investigation approach, “the process of questioning a product to evaluate it”, where the “questions” are things the tester tries to do with the product, and the product answer with the behavior in reaction to the probing test. During the testing, the users also considered the common quality attributes of the software that include

functionality, Reliability, Usability, Efficiency, Portability, and Maintainability of the “Intelligent IP Address Locator via Google Maps”.

### **Operational and Testing Procedure**

The researchers followed the instruction as a minimum requirements based on the study. The study passed through critic evaluation, software evaluation hardware minimum requirements and software minimum requirements until it meets the specific output of the proposed study.

### **Software Testing Procedure**

The result of it may determine whether or not the researchers continue the study.

1. Set-up the system and check if it is properly working.
2. Load all the necessary data needed for the operation, then check the default values in setting and maintenance and make some correction if necessary.
3. Try at least 2 users, using “Intelligent IP Address Locator via Google Maps” let them use the system whether it may work the way they want it to happen.
4. The actual operation begins with supervision of the researchers. Testing method may be applied in a computerized process for at least 1 month.
5. After 3 hours the researchers may conduct an evaluation about the system software.

### **Software Evaluation Procedure**

The system software can only be met by means of through test and evaluation. The system developers also believed that all software, either system or application, cannot be implemented without proper testing and evaluation before its operation. The process could help the researchers in attaining the finest output. The respondents of the system study evaluated it carefully, the selected respondents included IT professionals such as five (5) IT instructors, five (5) non IT instructors, and twenty (20) students that

are considered professional evaluators, as well as end users. Each respondent rated the study according to the criteria created by the researchers.

**Functionality.** A set of attributes that bear on the existence of a set of functions and their specified properties. The functions are those that satisfy or implied need.

<b>Mean Ranges</b>	<b>Descriptive Rating</b>
4.21 – 5.00	Very Much Functional
3.41 – 4.20	Much Functional
2.61 – 3.40	Moderately Functional
1.81 – 2.60	Less Functional
1.00 – 1.80	Not Functional

**Reliability.** The ability of a system to perform its required functions under stated conditions whenever required functions under stated conditions, having a long mean time between failures.

<b>Mean Ranges</b>	<b>Descriptive Rating</b>
4.21 – 5.00	Very Much Reliable
3.41 – 4.20	Much Reliable
2.61 – 3.40	Moderately Reliable
1.81 – 2.60	Less Reliable
1.00 – 1.80	Not Reliable

**Usability.** A set of attributes that bear on the efforts needed for use, and on the individual assessment of such use, by a stated or implied set of users.

<b>Mean Ranges</b>	<b>Descriptive Rating</b>
4.21 – 5.00	Very Much Usable
3.41 – 4.20	Much Usable
2.61 – 3.40	Moderately Usable

1.81 – 2.60	Less Usable
1.00 – 1.80	Not Usable

**Efficiency.** A set of attributes that avoid wasting time in doing something or in producing a desired result.

Mean Ranges	Descriptive Rating
4.21 – 5.00	Very Much Effecient
3.41 – 4.20	MuchEffecient
2.61 – 3.40	Moderately Efficient
1.81 – 2.60	Less Effecient
1.00 – 1.80	Not Effecient

**Portability.** A set of attributes that can be transferred from one machine or system to another.

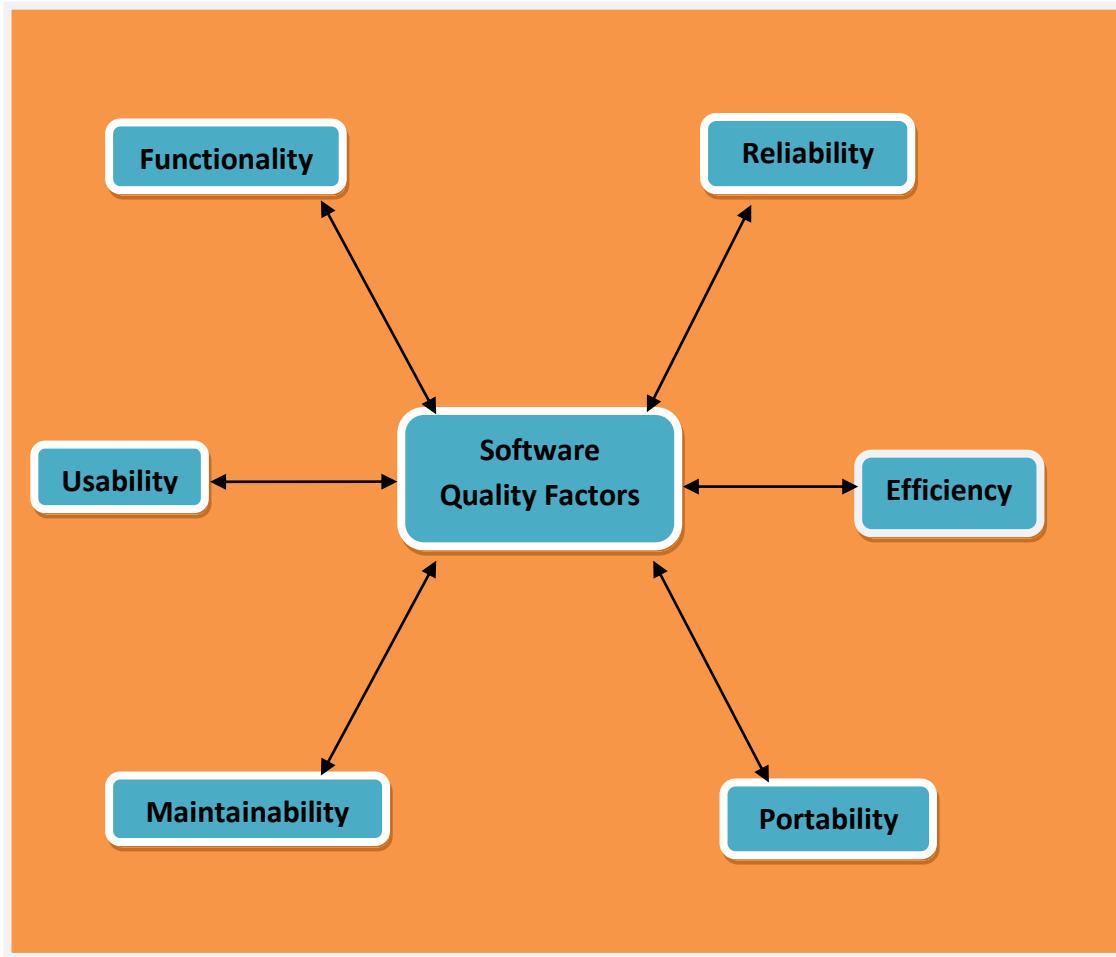
4.21 – 5.00	Very Much Portable
3.41 – 4.20	Much Portable
2.61 – 3.40	Moderately Portable
1.81 – 2.60	Less Portable
1.00 – 1.80	Not Portable

**Maintainability.** A set of attributes that measures the ease and speed with which a system can be restored to operational status after a failure occurs.

4.21 – 5.00	Very Much Portable
3.41 – 4.20	Much Portable
2.61 – 3.40	Moderately Portable
1.81 – 2.60	Less Portable
1.00 – 1.80	Not Portable

## Software Quality Factors

A software quality factor is a non-functional requirement for a software program which is not called up by the customer's contract, but nevertheless is a desirable requirement which enhances the quality of the software program.



**Figure 20. Software Quality Factors**

#### **Respondent of the study**

There were thirty (30) respondents of the study and were classified into four (4) groups. The first group comprises of the instructors (IT and Non-IT), the second group comprises of the students (IT and Non-IT), and the third group comprises of the office of the director for copyrights and office staff. The respondents were given questionnaires for

them to provide information intended for the study. There was one (1) item each of the system evaluation criterion namely functional, reliability and efficiency.

**Table 1. Respondent of the Study**

RESPONDENTS	FREQUENCY	PERCENTAGE
<b>Non IT Instructor</b>	5	33.33%
<b>Students (IT)</b>	10	53.33%
<b>Random Students</b>	10	6.67%
<b>IT Instructors</b>	5	6.6%
<b>TOTAL</b>	30	100%

Table 1 shows that there were five (5) non IT instructor or 33.33% and five (5) or 6.6% IT instructors. There is also ten (10) or 53.33% for the students which were ITs and ten (10) or 6.67% random students.

### **Research Instrument**

This study utilized the researcher-made questionnaire as instrument in data collection. The principal data-gathering tool was a set of official documents obtained from various groups of IT Students, IT Instructors, Random Students and Non IT Instructors team showing the effectiveness of the Intelligent IP Address Locator via Google Map in terms of system functionality, reliability and usability.

### **Statistical Tools and Treatment**

The mean was used to measure the effectiveness of the Intelligent IP Address Locator via Google Maps in terms of functionality, reliability and usability. It uses the solving formula which shows the following range of values.

$$\text{weighted mean} = \frac{\sum f x}{N}$$

Where;

- f - the frequency in the weight score
- x - the weights or score
- N - the total number of respondents

### **Figure 21. Formula on Weighted Mean**

The respondents were chosen with the application of the combination of Quota Sampling Method and Convenience Sampling Method. Quota Sampling Method is the gathering of respondents from a group. It requires the individual respondents that are chosen out of a specific group. The convenience Sampling Method is the selection of respondents within the group because of their availability and easy access. The advantages of this method of sampling are the availability and quickness with which data are gathered. The disadvantages are the risk that the sample might not represent the true answer to the survey as a whole, and it might be biased.

The researchers applied the two methods of selecting respondents. They determined the following groups: IT Students, Random Students, IT Instructors and Non IT Instructors of JRMSU Dapitan;

**Implementation.** The idea is to run the old system alongside the new system for a while. With this approach, the outputs or behavior of each system can be compared to make sure that the new system is performing as expected. A method used by the researchers in eliciting some ideas is from **Parallel Adoption Process**. According to this method, it cannot be represented without paying attention to the steps before the actual

conversion, namely the construction of a conversion scenario and the identification and testing of all the requirements of a system in order to reduce risk, the old and new system run simultaneously for some period of time after which, if the criteria for the new system are met, the old system is disabled. The process requires careful planning and control and a significant investment in labor hours. Some systems are incapable of being introduced in pieces as it is too reliant on the whole system. Implementation of this system is not highly successful without any idea or source from the environment of the researchers.

With the realization of a technical specification or algorithm as a program, software components, or other system is done through programming and development. It gives benefits to the users involvement and management support in designing and operating the system “Intelligent IP Address Locator via Google Map”. The system implementation has its several positive results. First the users are heavily involved in system design; it offers opportunities to mold the system according to their priorities and business requirements and more opportunities to control the outcome. Second, they are more likely to react positively to the change process. Incorporating user knowledge and expertise leads to the better solutions.

**Documentation.** The researchers made documentation after data gathering, requirement analysis, designing, coding, testing and evaluation of the publication and realization. The researchers kept the data from the beginning of the creation of the system project. The procedures in the project were kept as a referral in times of upgrading the software. Those files would help to recall all the input modules which are very important.

### **Calendar of Activities**

The researchers followed on the ruling of the calendar of activities in which the flow has minimum and maximum activities of the study. It indicated weeks and months

to take the study completed and showed how the study goes on with the corresponding activities taken.

**Table 2. Calendar of Activities**

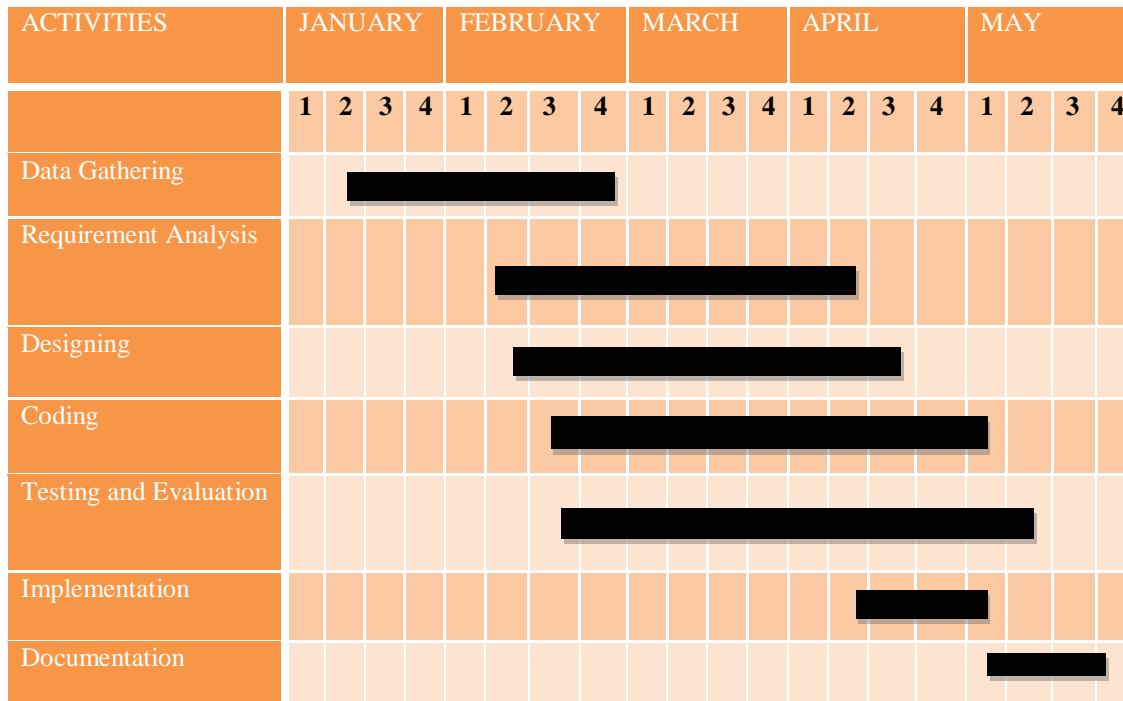


Table 2 shows the calendar of activities and the undertakings of the researchers from the start of the study until the proposed study was completely finished/ the first task of the project was Data Gathering and conducted on the 2<sup>nd</sup> week of January. It took almost seven (7) weeks before some necessary data were gathered and consolidated. While on the progress of data gathering, the researchers were doing a Requirement Analysis on software and hardware requirements which started on the 2<sup>nd</sup> week of February, it showed almost nine (9) weeks of analyzing the software and hardware minimum requirements needed for the proposed study. Followed by designing, coding, testing and evaluation. Maintenance shall follow in order to determine its correctness, functionality and to finalize the full implementation of the system on the start of the 2<sup>nd</sup> week of April. A full documentation of the study conducted from the start to last day of implementation.

The researchers made some correction, consultation and verification about the data gathered if it is applicable to the study which had took almost days on documentation process. The researchers had passed through a hectic and time conscious to make the study possible with specific output.

## **CHAPTER 4**

### **RESULTS AND DISCUSSIONS**

This chapter presents the answers of the problems presented upon developing the proposed study. It answers sequentially the questions shown on the statement of the problem.

#### **1. What is the current status or of Intelligent IP Address Locator via Google Maps?**

At the moment, most of the IP address locators that are being used in our society still are very poor in tracing people who are involved to wrong doing in our society like in social media. People nowadays are very fond of social media. And some of the online company is being attached to social media too. They were having online business transactions to people who are interested of their product. And that's the time when some of the scammers are involving in social media or even in our society. Intelligent IP Address Locator via Google Maps is a simple solution to this problem, which also gives you the added benefit of being able to drive your customers to a mobile web page. By using the service you can now set aside your worries.

#### **2. What are the process involved in the design and development of Intelligent**

##### **IP Address Locator via Google Maps?**

In phases of this study Intelligent IP Address Locator via Google Map was conducted by the researchers. The system is to be designed to retrieve IP address information of particular PC/laptops and then reverse engineer them into city coordinates. These city coordinates are then mapped onto a Google map in order to graphically map them on a Google map. This provides a robust and efficient PC/laptop tracking system for companies as well as cyber crime that allows for instant IP retrieval and reverse IP

looking that PC address to plot it on a Google map. This system uses DOTNET framework coupled with sql backend in order to achieve this functionality.

**3. What are the similar or existing system features that can be fused into one composite to come up with the new hybrid system?**

The other existing system which can be fused into one composite system is AN INTELLIGENT TACKING SYSTEM BASED ON GSM AND GPS USING SMARTPHONE'S used to find out the position or location of the vehicle around the world. This implementation introduces an Android based tracking and theft prevention system. The DESIGN AND DEVELOPMENT OF GPS-GSM BASED TRACKNG SYSTEM WITH GOOGLE MAP BASED which may receive the coordinates from the satellites among other critical information. Tracking system is very important in modern world. This can be useful in soldier monitoring, tracking of the theft vehicle and various other applications. REAL TIME POSITION TRACKING SYSTEM USING GOOGLE MAPSAPI V3 is emerging as a new tool for effectively monitoring the physical world, especially in urban areas where a high concentration of vehicles equipped with on board sensors is expected.

The researchers made a way to improve the performance of the Intelligent IP Address Locator via Google Maps by improving the added features from other systems which are very useful to everyone and also to improve the existing concept to make it more useful and are functional. The researchers added some features to improve the system's performance.

**4. How can the Intelligent IP Address Locator via Google Maps are measured in terms offunctionality, reliability, usability, efficiency, portability, and maintainability?**

## Evaluation Process

The findings that proved the functionality, reliability, usability, efficiency, portability and maintainability of the Intelligent IP Address Locator via Google Map are presented here.

The result of this evaluation determined the need of creating the project, Intelligent IP Address Locator via Google Map. The means may be interpreted as to the ranges plotted below.

The respective respondents are composed of five (5) Information Technology individuals such as five (5) I.T instructors and twenty (20) I.T students and random students, they were considered as a professional evaluators as well as end users. Each respondents rate the study according to the criterion created by the researcher. The results as follows:

### 4.1 Functionality

A set of attributes that bear on the existence of a set of functions and their specified properties. The functions are those that satisfy or implied set of users' needs. . A given software component or system does not typically function in isolation.

**Table 3. Over-all Result on the Factor: Functionality**

FUNCTIONALITY	Mean	Description
1. Intended use of the software	<b>4.53</b>	Very Much Functional
2. Data manipulation	<b>4.3</b>	Very Much Functional
3. Compliance of end-user needs	<b>4.3</b>	Very Much Functional
4. Security of system data	<b>4.3</b>	Very Much Functional
5. Compatibility of other system	<b>4.4</b>	Very Much Functional
6. Speed in data processing	<b>4.5</b>	Very Much Functional
7. Fitness of its intended use	<b>4.4</b>	Very Much Functional
8. Minimization of its run-time error	<b>4.5</b>	Very Much Functional
9. Detection of error	<b>4.5</b>	Very Much Functional
10. Data storage	<b>4.6</b>	Very Much Functional
<b>Average Mean</b>	<b>4.43</b>	<b>Very Much Functional</b>

Legend: 4.20-5.0 Very Much Functional      3.40-4.19 Much Functional      2.60-3.39 Moderately Functional  
 1.80-2.60 Less Functional      1.0-1.80 Not Functional

Functionality is the sum or any aspect of what a product, such as a software application or computing device, can do for a user. Table 3 shows the descriptive rating of functionality which is composed of four criteria to be rated as much functional namely; the Intended use of the software got some mean of 4.53 or Very Much Functional. The second criterion of Data manipulation got some mean of 4.3 or Very Much Functional. The third criterion is the Compliance of the end-user needs got some mean of 4.3 or Very Much Functional. The fourth criterion is the Security of the system data got some mean of 4.3 or Very Much Functional. The fifth criterion is the Compatibility of other system which has a mean of 4.4 or Very Much Functional. The sixth criterion is the Speed in data processing has got a mean of 4.5 or Very Much Functional. The seventh criterion is the Fitness of its intended use got some mean of 4.4 or Very Much Functional. In the eighth criterion is the Minimization of its run-time error that got some mean of 4.5 or Very Much Functional. In the ninth criterion is the Detection of error which got some mean of 4.5 or Very Functional. The last criterion Data storage got a mean of 4.6 or Very Functional.

These findings supported by the study of Kumarl &Shankaraiah (2014) entitled “Intellegence Student Tracking System based on RFID and Zigbee”. This paper proposes an intelligent student tracking system based on RFID and ZigBee wireless network. The findings of the study showed that even the students are in the campus but not visible in the classes during class hour but the presence of students in the class is being monitored by taking attendance manually in the class.

The computed mean of the criteria “security of system data” is 4.3 and described as “very much functional”. The respondent of the system evaluated it carefully and surely based on the system’s functions manipulated by them.

Therefore, the researchers proposed system is very much functional in data security.

#### **4.2 Reliability**

It refers to the ability of the Intelligent IP Address Locator via Google Maps to consistently perform its intended or required function or mission on demand and without degradation or failure.

**Table 4. Over-all Result on the Factor: Reliability**

RELIABILITY	Mean	Description
1. Error tolerance	<b>4.4</b>	Very Much Reliable
2. Ease in data recovery	<b>4.4</b>	Very Much Reliable
3. Program security	<b>4.5</b>	Very Much Reliable
4. Data security	<b>4.5</b>	Very Much Reliable
5. Creation of back-up system	<b>4.5</b>	Very Much Reliable
6. Accuracy of data capture	<b>4.1</b>	Much Reliable
7. Accuracy of Results	<b>4.3</b>	Very Much Reliable
8. Data storage volume	<b>4.3</b>	Very Much Reliable
9. Understand ability of output	<b>4.3</b>	Very Much Reliable
10. Completeness of the system	<b>4.5</b>	Very Much Reliable
<b>Average</b>	<b>4.38</b>	<b>Very Much Reliable</b>

Legend: 4.20-5.0 Very Much Reliable    3.40-4.19 Much Reliable    2.60-3.39 Moderately Reliable  
 1.80-2.60 Less Reliable    1.0-1.80 Not Reliable

Reliability is the sum or any aspect of what a product, such as a software application or computing device, can do for a user. Table 3 shows the descriptive rating of reliability which is composed of four criteria to be rated very much functional namely; the Error tolerance got some mean of 4.4 or Very Much Reliable. The second criterion of Ease in data recovery got some mean of 4.5 or Very Much Reliable. The third criterion is the Program security got some mean of 4.5 or Very Much Reliable. The fourth criterion is the Data security got some mean of 4.5 or Very Much Reliable. The fifth criterion is the Creation of back-up system which has a mean of 4.5 or Very Much Reliable. The sixth criterion is the Accuracy of data captured has got a mean of 4.1 or Much Reliable. The seventh criterion is the Accuracy of result got some mean of 4.3 or Very Much Reliable. In the eighth criterion is the Data storage volume that got some mean of 4.3 or

Very Much Reliable. In the ninth criterion is the Understand ability of output which got some mean of 4.5 or Very Much Reliable. The last criterion Completeness of the system got a mean of 4.5 or Very Much Reliable.

These findings supported the study of Kumar (2014) fitted one of the criteria of the recent findings which is “accuracy of results”. Based on the study propose an intelligent student tracking system based on RFID and ZigBee wireless network. The proposed system reads the RFID tags data through RFID reader, and then sends it to PC node by a custom wireless protocol on the ZigBee. PC node gives corresponding warning or hints by the result of matching master-slave RFID tag information.

Generally, the computed mean of the criteria “accuracy of results” is 4.3 and which is described as “very much reliable. This system produced real time tracks and accurate location results which the user can rely on the determined location and status of the vehicle.

### 4.3 Usability

It pertains to the user’s total satisfaction received from using the proposed Intelligent IP Address Locator via Google Maps.

**Table 5: Over-all Result on the factor: Usable**

USABILITY	Mean	Description
1. User friendly program	<b>4.6</b>	Very Much Usable
2. Quick driven program	<b>4.3</b>	Very Much Usable
3. Simple manipulation features	<b>4.2</b>	Very Much Usable
4. Wrong key input errors detection	<b>4.2</b>	Very Much Usable
5. Wrong time input errors detection	<b>4.3</b>	Very Much Usable
6. Data storage	<b>4.2</b>	Very Much Usable
7. Data retrieval	<b>4.4</b>	Very Much Usable
8. Data edit/correction	<b>4.4</b>	Very Much Usable
9. Tolerable difficulty level	<b>4.4</b>	Very Much Usable
10. Production of data output	<b>4.3</b>	Very Much Usable
<b>Average</b>	<b>4.33</b>	<b>Very Much Usable</b>

Legend: 4.20-5.0 Very Much Usable  
 3.40-4.19 Much Usable  
 2.60-3.39 Moderately Usable  
 1.80-2.60 Less Usable  
 1.0-1.80 Not Usable

Usability is the sum or any aspect of what a product, such as a software application or computing device, can do for a user. Table 5 shows the descriptive rating of reliability which is composed of four criteria to be rated very much functional namely; the User friendly program got some mean of 4.6 or Very Much Usable. The second criterion of Quick driving program got some mean of 4.3 or Very Much Usable. The third criterion is the Simple manipulation features got some mean of 4.2 or Very Much Usable. The fourth criterion is the Wrong key input errors detection got some mean of 4.2 or Very Much Usable the fifth criterion is the Wrong time input errors detection which has a mean of 4.3 or Very Much Usable. The sixth criterion is the Data storage has got a mean of 4.2 or Very Much Usable. The seventh criterion is the Data retrieval got some mean of 4.4 or Very Much Usable. In the eighth criterion is the Data edit/correction that got some mean of 4.4 or Very Much Usable In the ninth criterion is the Tolerable difficulty level which got some mean of 4.4 or Very Much Usable The last criterion Production of data output got a mean of 4.3 or Very Much Usable.

The findings supported the study of by Lee (2012) and fitted one of the criteria of the present findings “simple manipulation” features because it uses simple keys that will ease the user in system manipulation. This criterion has garnered the mean of 4.2 which described as very much usable.

Therefore the researchers proposed system is very much usable in terms of features.

#### **4.4 Efficiency**

Refers to Intelligent IP Address Locator via Google Maps of what is actually produced or performed with what can be achieved in contrast with tracking and monitoring location.

**Table 6: Over-all Result on the factor: Efficiency**

EFFICIENCY	Mean	Description
1. Support on minimum facilities	<b>4.5</b>	Very Much Efficient
2. Support on minimum requirements	<b>4.4</b>	Very Much Efficient
3. Provision of configurable automation	<b>4.4</b>	Very Much Efficient
4. Support on business workflow process	<b>4.4</b>	Very Much Efficient
5. Support on number of user's	<b>4.4</b>	Very Much Efficient
6. Speed of navigation and production of outputs	<b>4.3</b>	Very Much Efficient
7. Speed of data capture and retrieval	<b>4.5</b>	Very Much Efficient
8. Hardware utilization	<b>4.4</b>	Very Much Efficient
9. Support on interfacing with other devices	<b>4.5</b>	Very Much Efficient
10. Compatibility with interfaced devices	<b>4.6</b>	Very Much Efficient
<b>Average</b>	<b>4.44</b>	<b>Very Much Efficient</b>

Legend: 4.20-5.0 Very Much Efficient      3.40-4.19 Much Efficient      2.60-3.39 Moderately Efficient 1.80-2.60 Less Efficient      1.0-1.80 Not Efficient

Efficiency is the sum or any aspect of what a product, such as a software application or computing device, can do for a user. Table 6 shows the descriptive rating of reliability which is composed of four criteria to be rated very much functional namely; the Support on minimum facilities got some mean of 4.5 or Very Much Efficient. The second criterion of Support on minimum requirements got some mean of 4.4 or Very Much Efficient. The third criterion is the Provision of configurable automation got some mean of 4.4 or Very Much Efficient. The fourth criterion is the Support on business workflow process got some mean of 4.4 or Very Much Efficient the fifth criterion is the Support on number of user's which has a mean of 4.4 or Very Much Efficient. The sixth criterion is the Speed of navigation and production of outputs got a mean of 4.3 or Very Much Efficient. The seventh criterion is the Speed of data capture and retrieval got some mean of 4.5 or Very Much Efficient. In the eighth criterion is the Hardware utilization that got some mean of 4.4 or Very Much Usable In the ninth criterion is the Support on interfacing with other devices which got some mean of 4.5 or Very Much Efficient. The last criterion Compatibility with interfaced devices got a mean of 4.6 or Very Much Efficient.

These findings supported the study of Moura (2015) it reduces the need of pen and paper based manual attendance tracking system. Following this thought, we have proposed a smart location based time and attendance tracking system which is implemented on android mobile application on Smartphone reducing the need of additional biometric scanner device. This study fitted one of the criteria of the present findings the “ provision of configurable automation”, because after all the deliberation of what is the best thing to do to come up with better plans, this related study is able to finish and able to do the final realization. The same ways have done with the present study.

The computed mean of the criteria “provision of configurable automation” is 4.4 which described as “very much efficient” to the user. The respondents of the system evaluated carefully based on the effectiveness of the system and the configuration made by the user.

#### **4.5 Potability**

It refers to the Intelligent IP Address Locator via Google Maps of what is actually produced or performed with what can be achieved in contrast with the traditional monitoring. This quality factors is also measured through adaptability which characterizes the ability of the system to change into new environments.

**Table 7: Over-all Result on the factor: Portability**

PORTABILITY	Mean	Description
1. Modification of the system	<b>4.5</b>	Very Much Portable
2. Adaptability to other environment	<b>4.4</b>	Very Much Portable
3. Adaptability to other applications	<b>4.4</b>	Very Much Portable
4. Flexibility to other settings	<b>4.4</b>	Very Much Portable
5. Support in any form of network communication	<b>4.6</b>	Very Much Portable
6. Adaptability to new version of system requirements	<b>4.3</b>	Very Much Portable
7. System supports on maximum hardware requirements	<b>4.4</b>	Very Much Portable
8. User capability/capacity	<b>4.5</b>	Very Much Portable
9. Its intended application	<b>4.5</b>	Very Much Portable
10. Its intended design	<b>4.6</b>	Very Much Portable
<b>Average</b>	<b>4.46</b>	<b>Very Much Portable</b>

Legend: 4.20-5.0 Very Much Portable    3.40-4.19 Much Portable    2.60-3.39 Moderately Portable  
 1.80-2.60 Less Portable    1.0-1.80 Not Portable

Portability is the sum or any aspect of what a product, such as a software application or computing device, can do for a user. Table 7 shows the descriptive rating of reliability which is composed of four criteria to be rated very much functional namely; the Modification of the system got some mean of 4.6 or Very Much Portable. The second criterion of Adaptability to other environment got some mean of 4.3 or Very Much Portable. The third criterion is the Adaptability to other applications got some mean of 4.2 or Very Much Portable. The fourth criterion is the Flexibility to other settings got some mean of 4.2 or Very Much Portable the fifth criterion is the Support in any form of network communication which has a mean of 4.3 or Very Much Portable. The sixth criterion is the Adaptability to new version of system requirements has got a mean of 4.2 or Very Much Portable. The seventh criterion is the System supports on maximum hardware requirements got some mean of 4.4 or Very Much Portable. In the eighth criterion is the User capability/capacity that got some mean of 4.4 or Very Much Portable. In the ninth criterion is the Its intended application which got some mean of 4.4 or Very Much Portable. The last criterion, its intended design got a mean of 4.3 or Very Much Portable.

In these findings supported the study of Deshmukh (2016) entitled “GPS and RFID Based Intelligent Bus Management and Monitoring System” fitted one of the criteria of the present findings “its intended application” that a bus tracking system has been widely used for fleet management and asset tracking. GPS and RFID Based Intelligent Bus Tracking System are useful, accurate, robust, flexible and economical. In this work, GPS is used for obtaining the location of the bus using Google Map and RFID is used for the identification of bus.

The criteria “its intended application” of the software garnered the mean of 4.5 and described as very much portable. The respondents of the system evaluated carefully based on the effectiveness of the system and the configuration made by the user.

#### **4.6 Maintainability**

Pertains to characteristic of the system and its installation which determine the probability that its failure can be restored to its normal operable state within a given time frame using the prescribed practices and procedures. One of the ability of maintainability is the ability to verify a system's testability.

**Table 8: Over-all Result on the factor: Maintainability**

MAINTAINABILITY	Mean	Description
1. Modification of the system software	<b>4.6</b>	Very Much Maintainable
2. Change of software capabilities	<b>4.5</b>	Very Much Maintainable
3. Increase program capabilities	<b>4.3</b>	Very Much Maintainable
4. Improving performance	<b>4.3</b>	Very Much Maintainable
5. Correction of program defects	<b>4.4</b>	Very Much Maintainable
6. Accessibility for maintenance	<b>4.4</b>	Very Much Maintainable
7. Configuration of system operation	<b>4.4</b>	Very Much Maintainable
8. Flexibility for system modification	<b>4.5</b>	Very Much Maintainable
9. Compliance of concurrent system requirements	<b>4.3</b>	Very Much Maintainable
10. Advance feature for recent technology	<b>4.6</b>	Very Much Maintainable
<b>Average</b>	<b>4.43</b>	<b>Very Much Maintainable</b>

Legend: 4.20-5.0 Very Much Maintainable      3.40-4.19 Much Maintainable      2.60-3.39 Moderately Maintainable  
1.80-2.60 Less Maintainable      1.0-1.80 Not Maintainable

Maintainability is the sum or any aspect of what a product, such as a software application or computing device, can do for a user. Table 8 shows the descriptive rating of reliability which is composed of four criteria to be rated very much functional namely; the Modification of the system software got some mean of 4.6 or Very Much Maintainable. The second criterion of Change of software capabilities got some mean of 4.3 or Very Much Maintainable. The third criterion is the Increase program capabilities got some mean of 4.2 or Very Much Maintainable. The fourth criterion is the

Improving performances got some mean of 4.2 or Very Much Maintainable, the fifth criterion is the Correction of program defects which has a mean of 4.3 or Very Much Maintainable. The sixth criterion is the Accessibility for maintenance has got a mean of 4.2 or Very Much Maintainable. The seventh criterion is the Configuration of system operation got some mean of 4.4 or Very Much Maintainable. In the eighth criterion is the Flexibility for system modification that got some mean of 4.4 or Very Much Maintainable. In the ninth criterion is the Compliance of concurrent system requirements which got some mean of 4.4 or Very Much Maintainable. The last criterion Advance feature for recent technology got a mean of 4.3 or Very Much Maintainable.

These finding supported the study of Sultana (2014) fitted one of the criteria of the present findings “advance feature for recent technology”. We have proposed a location based smart time and attendance tracking system based on the concept of web services which is implemented as an Android mobile application that communicates with the remote server in which the database is located. Internet connectivity (Wi-Fi/3G) is needed for connecting to database residing in the remote server. Any employee crossing the area border with a Smartphone and the running application may be tracked automatically.

The criteria “advance feature for recent technology” of the software garnered the mean of 4.6 and described as very much portable. The respondents of the system evaluated carefully based on the effectiveness of the system and the configuration made by the user.

### **Summary of the Ratings of Software Quality Factors**

Based on the results obtained from the evaluators of the Intelligent IP Address Locator via Google Maps, it was concluded that the system is acceptable and it meets all

its requirements like functionality, reliability, usability, efficiency, portability and maintainability of the system in building good quality software.

**Table 6: Over-all Result of all Functional, Reliability, and Usability:**

Criteria	Mean	Descriptive rating
1. Functionality	4.4	Very Much Functional
2. Reliability	4.38	Very Much Reliable
3. Usability	4.33	Very Much Usable
4. Efficiency	4.44	Very Much Efficient
5. Portability	4.46	Very Much Portable
6. Maintainability	4.43	Very Much Maintainable
<b>Total of Over-all Mean</b>	<b>4.41</b>	<b>Acceptable</b>

The result of the evaluation of the system is acceptable because it has a total quality to operate the system. It is real-time monitoring solution providing visibility and traceability of the whole process of the system and it is designed to make the system functional, reliable and usable.

To solve the over-all mean, first add the three mean ( $4.4 + 4.38 + 4.33 + 4.44 + 4.46 + 4.43 =$ ), next divide the total of the six mean to the number of criteria used ( $26.44/6 =$ ). And the answer of the total of over-all mean is 4.41.

## **CHAPTER 5**

### **SUMMARY, FINDINGS, CONCLUSIONS AND RECOMMENDATIONS**

This chapter presents the summary of the study that has been presented and evaluated. It also covers the presentation of the findings out if the given data and the conclusion are based on the findings.

#### **Summary**

In this chapter the researchers sought to study “intelligent IP Address Locator via Google Map” and the sub problem in the statement of the problem. This was accomplished through a process called the project development cycle which includes Data Gathering, Requirements and Analysis, Designing, Coding, and Testing & Evaluation. This system can give people efficient information.

Answers to the following sub problems were being sought. Below are the questions that include:

1. What is the current status or of Intelligent IP Address Locator via Google Maps?
2. What are the components involved in the design and development of Intelligent IP Address Locator via Google Maps?
3. What are the similar or existing system features that can be fused into one composite to come up with the new hybrid system?
4. How can the Intelligent IP Address Locator via Google Maps be measured in terms of functionality, reliability, usability, efficiency, portability and maintainability?

The method used in the study was constructive research method employing documentary analysis wherein the data gathered from books, internet research and other

resource document were carefully examined and analyzed in order to produce the desired result and accurate output of the study.

## **Findings**

Summary of findings is the last part of the thesis or dissertation. It is where the findings or the result of the thesis study is written. This part is the heart of the system “Intelligent IP Address Locator vi Google Map” with which the value of the thesis may stand or fall on the validity and quality of the thesis findings. Once it is completed the literature review, created a research design and formulated hypotheses from the research questions to test, the rest of the thesis project centers on making this thesis findings be done.

The following findings were revealed in the level of effectiveness of the system according to its problem from the previous chapter. After all the necessary data of the study were treated accordingly, the following findings were exposed:

1. Researchers found that most of the IP address locator that are being used in our society still are very poor in tracing people who are involved to wrong doing in our society like in social media. People nowadays, are very fund of social media. And some of the online company is being attached to social media too. They were having online business transaction to the people who are interested of their product. And that's the time when some of the scammers emerging in social media or even in our society. Intelligent IP Address Locator via Google Maps is a simple solution to this problem, which also gives you the added benefit of being able to drive your customers to a mobile web page. By using the service you can now set aside your worries.
2. Researchers formulated the Intelligent IP address Locator via Google Mapsshow the process of the system explains the flow of the program that

the System flowcharts are a way of displaying how data flows in a system and how decisions are made to control events. The process, environment, and blue print of the system in which the user's know the flow of the system to make the process faster, easier and more usable.

3. The researchers compared the features of similar and existing systems from the proposed system and the features of the similar and existing system may be fused to the proposed system which is the Intelligent IP Address Locator via Google Maps to know if the system is more functional, reliable, usable and faster than the existing systems.
4. The proposed system entitled “Intelligent IP Address Locator via Google Maps” may help to maintain the stability of the process of their monitoring system.

4.1 The system software bears on the existence of a set of functions and their specified properties. The functions are those that satisfy stated or implied needs and were proven to be very much functional with mean average of 4.4.

4.2 The software quality in terms of reliability which determines the ability of a system to perform its required functions under the stated conditions whenever required – having a long time between failures much reliable was proven to be very much reliable with the mean average of 4.38.

4.3 The software quality in terms of usability that offer its interfaces in a user friendly and elegant way. The usability is the degree to which software can be used by specified consumers to achieve and satisfies has proven very much usability with the average mean of 4.33.

4.4 The software quality in terms of efficiency that bears on the relationship between the levels of performance, capability of the system and produce a specific outcome effectively was proven to be very much efficient with the average mean of 4.44.

4.5 The software quality in terms of portability that should run on as much various platforms. Portability is about adapting software that can run on different platforms and being compatible which proven very much portable with the average mean of 4.46.

4.6 The software quality in terms of maintainability the ability to adapt to changes, improve over time, correct any bugs and be proactively fixed through prevention maintenance which proven very much maintainable with the average mean of 4.43.

4.7 The overall mean average rating of the software quality factors which was interpreted as 4.41 is very much acceptable. The system works fast, quick and accessible to the user.

## **Conclusion**

The researchers concluded that the “INTELLIGENT IP ADDRESS LOCATOR VIA GOOGLE MAP” is found to be functional, less time consuming in giving information to the residents in anywhere via online. Thus, the purpose of the system is to complete the old manual procedure of informing the residents about awareness and response information system.

## **Recommendation**

This phase is used for measuring the degree to which the recommendations are followed by the students. Tracking generates various graphical models for displaying the statistical deviation in behavior of the user inputs and variations from their ideal situation.

Moreover, its further scope includes reports in the form data table, or any other relevant videos to evaluate themselves and trigger their learning interests. This section is based on an adaptive model wherein tracking will ensure the continuous evaluation of student's interests and any huge deviation will be accounted to the students with its advantages and disadvantages.

The proposed system analyzes on multiple factors. The inclusion of various factors enhances its ability to generate more accurate career options. The system suggested is also based online rather than on paper. Adaptive learning ensures that the questionnaire is updated per each user which also tracks his progress in the area. The inclusion of machine learning also adds to increasing the accuracy of the system by studying each user's behavior. The tracking feature of the system makes sure that user is making progress in the chosen field and checks if student's interest is changing over the period.

The researchers would like to recommend that the old manual procedure of informing the residents on awareness and response information system in Dapitan City be process. Adoption of this system basically means very functional.

# **APPENDICES**

## APPENDIX A

### Bibliography

## Internet Source

### Literature

Vigneshwaran, Sumithra & Janani (2015). An Intelligent Tracking System Based on GSM and GPS Using Smartphones. College of Engineering and Technology, Tiruchirappalli, Tamilnadu, India1&3 Associate Professor, Department of ECE, Pavendar Bharathidasan College of Engineering and Technology, Tiruchirappalli, Tamilnadu, IndiaVol. 4, Issue 5, May 2015.[https://www.ijareeie.com/upload/2015/may/16\\_AN-1.pdf](https://www.ijareeie.com/upload/2015/may/16_AN-1.pdf)

Bhatia, J. & Verma, P. (2013).Design and Development of GPS-GSM Based Tracking System with Google Map Based Monitoring. Centre for Development of Advanced Computing, Mohali, Punjab, IndiaVol.3, No.3, June 2013.[https://www.academia.edu/22721340/GSM\\_BASED\\_TRACKING\\_SYSTEM\\_WITH\\_GOOGLE\\_MAP\\_BASED\\_MONITORINGDESIGN\\_AND\\_DEVELOPMENT\\_OF\\_GPS-](https://www.academia.edu/22721340/GSM_BASED_TRACKING_SYSTEM_WITH_GOOGLE_MAP_BASED_MONITORINGDESIGN_AND_DEVELOPMENT_OF_GPS-)

MihirGarude, NirmalHaldikar (2014).Department of Electronics Engineering, Datta Meghe College of Engineering, Airoli. Volume 4, Issue 9, September 2014  
<http://www.ijsrp.org/research-paper-0914/ijsrp-p3362.pdf>

B. P. S. Sahoo and Satyajit Rath (2012).Integrating GPS, GSM and Cellular Phone for Location Tracking and MonitoringCSIR-Institute of Minerals & Materials Technology, Bhubaneswar, IndiaProceedings of the International Conference on Geospatial Technologies and Applications, Geomatrix'12 February 26 – 29, 2012, IIT Bombay, Mumbai, India<https://arxiv.org/ftp/arxiv/papers/1307/1307.3147.pdf>

B.Suchitha Samuel, J.Mrudula (2013).Design of Intelligent solar Tracker Robot for surveillancePost Graduate Scholar, Department of Electronics & Communication Engineering, Geethanjali College of Engineering, Keesara, Andhra Pradesh, India1 Associate Professor, Department of Electronics& Communication Engineering, Geethanjali College of Engineering, Keesara, Andhra Pradesh, IndiaVol. 2, Issue 10, October 2013<https://www.ijareeie.com/upload/2013/october/8UDesign.pdf>

### Local Studies

P.S.Kiran Kumar, Dr. Shankaraiah (2014). INTELLIGENT STUDENTS TRACKING SYSTEM IN CAMPUS BASED ON RFID AND ZIGBEE (Assistant Professor in E & E Engineering, VVIET, Mysore, Karnataka, India)(Professor, Dept. of E & C Engineering, SJCE, Mysore, Karnataka, India)Volume 5, Issue 8, August (2014), pp. 117-126

<http://www.iaeme.com/MasterAdmin/uploadfolder/INTELLIGENT%20STUDENTS%20TRACKING%20SYSTEM%20IN%20CAMPUS%20BASED%20ON%20RFID%20AND%20ZIGBEE/INTELLIGENT%20STUDENTS%20TRACKING%20SYSTEM%20IN%20CAMPUS%20BASED%20ON%20RFID%20AND%20ZIGBEE.pdf>

Laurie Thomas Lee, Ph.D. (2015). Location- based Communication Systems a Look at Intelligent Networking and Privacy Concerns College of Journalism and Mass Communications University of Nebraska-Lincoln<http://www.globalmediajournal.com/open-access/locationbased-communication-systemsa-look-at-intelligent-networking-and-privacy-concerns.php?aid=35306>

Shermin Sultana, Asma Enayet and Ishrat Jahan Mouri (2015). A Smart Location Based Time and Attendance Tracking System Using Android ApplicationDepartment of Computer Science and Engineering, Stamford University Bangladesh, Dhaka, BangladeshVol. 5, No.1, February 2015<http://airccse.org/journal/ijcsein/papers/5115ijcsein01.pdf>

Komal Satish Agarwal, Kranti Dive (2014). GPS and RFID Based Intelligent Bus Management and Monitoring SystemInternational Journal of Engineering Research & TechnologyVol. 3 - Issue 7 (July - 2014)<http://www.ijert.org/view-pdf/10385/rfid-based-intelligent-bus-management-and-monitoring-system>

## **APPENDIX B**

### **Users Manual**

## USER'S MANUAL

### *USER'S MANUAL*

- ❖ Using the Online system “Intelligent IP Address Locator via Google Map”.

### **STEPS:**

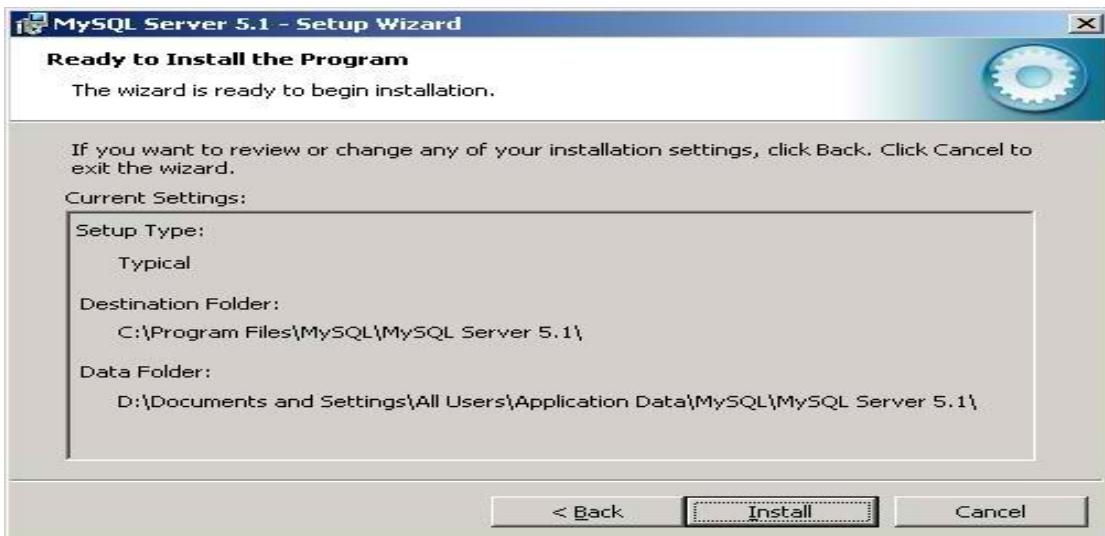
- Welcome to the Setup Wizard for MySQL Server 5.1. Click Next to continue.



- Select a setup type - Typical, Complete, and custom. Select Typical and click next.



- Ready to install MySQL.



- Installation in progress.



- More information about MySQL Enterprise Subscription.



- Setup Wizard Completed.

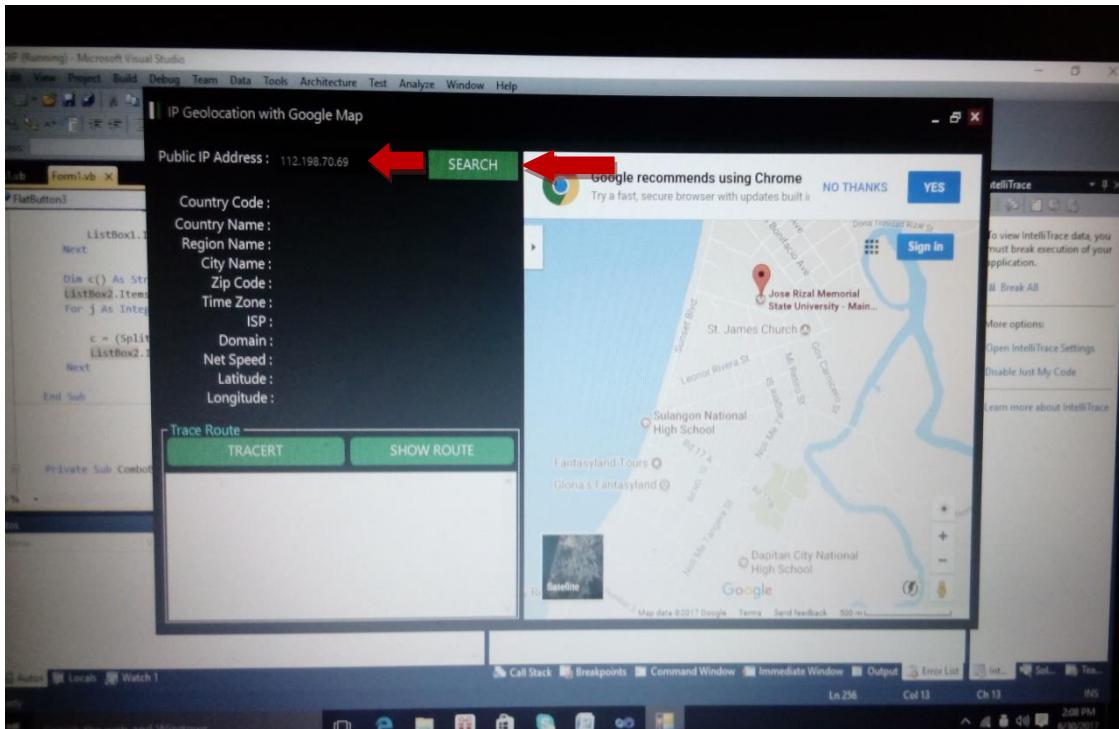


(Intelligent IP Address Locator via Google Maps)

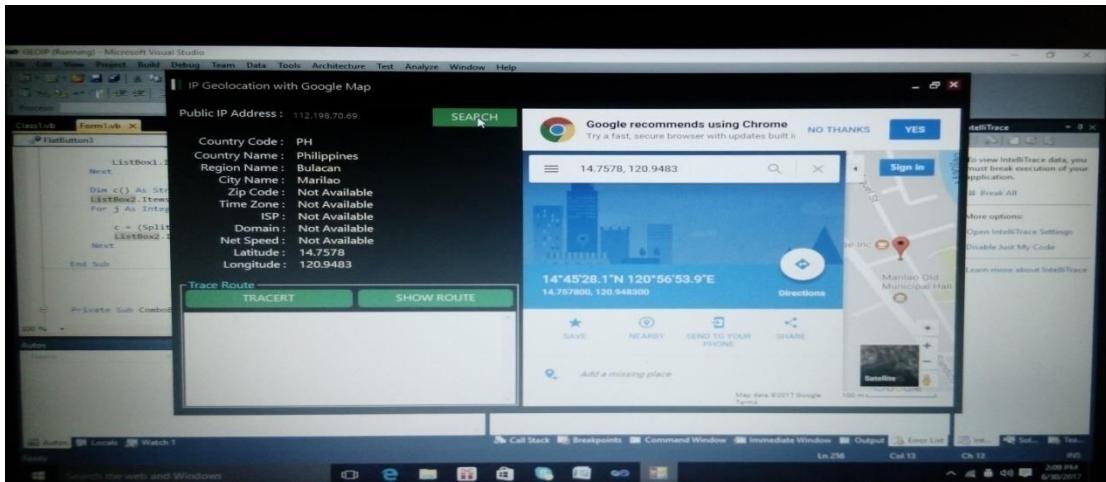
### Main menu of the system.

For Admin. The admin can view the menu bars like home, maps, IP address, and Location. Admin can search a location a certain person through IP Address. At first set-up the internet and once it done open the system.

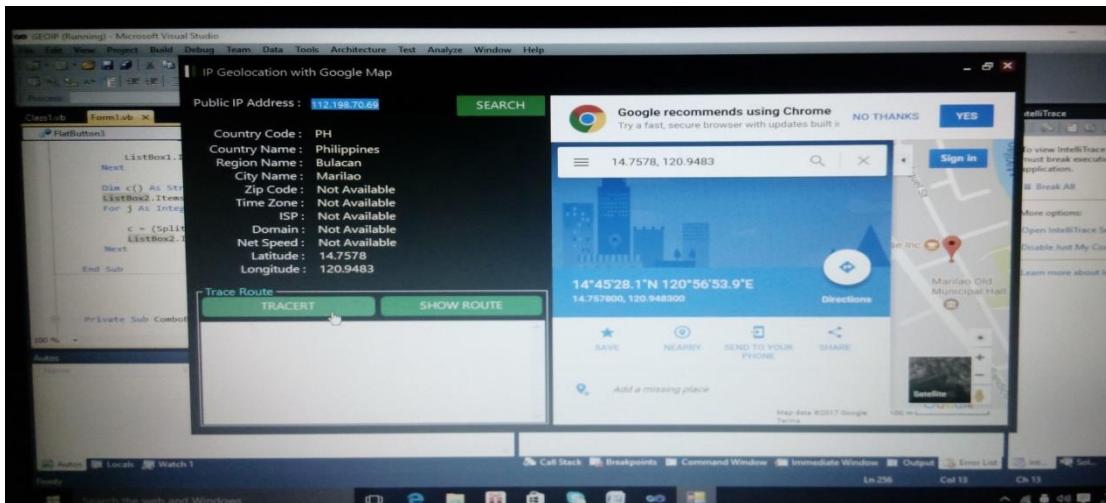
- Input IP Address and click Search.



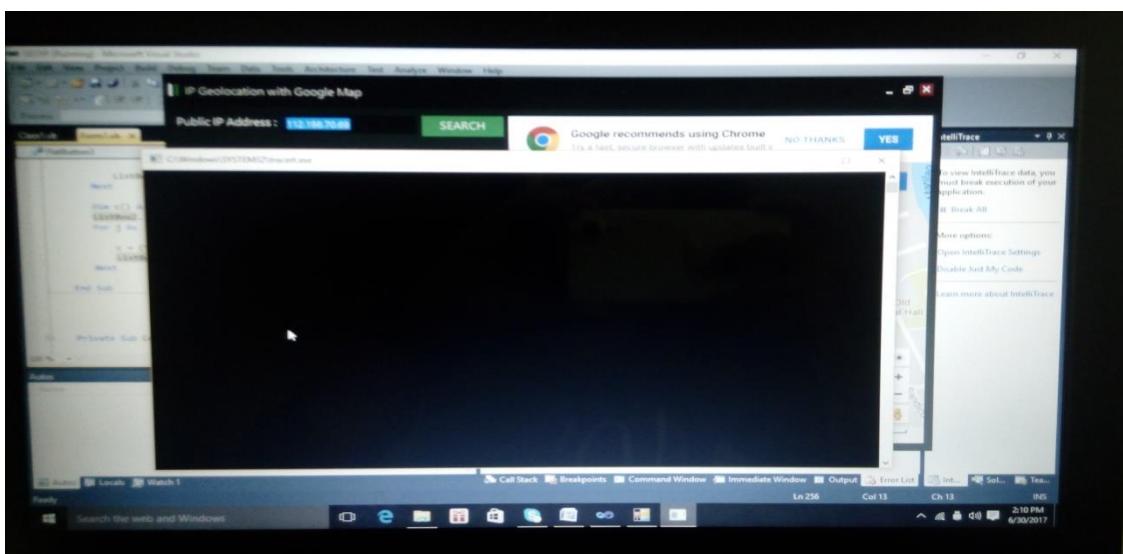
- Then you see the Location of a certain IP Address.



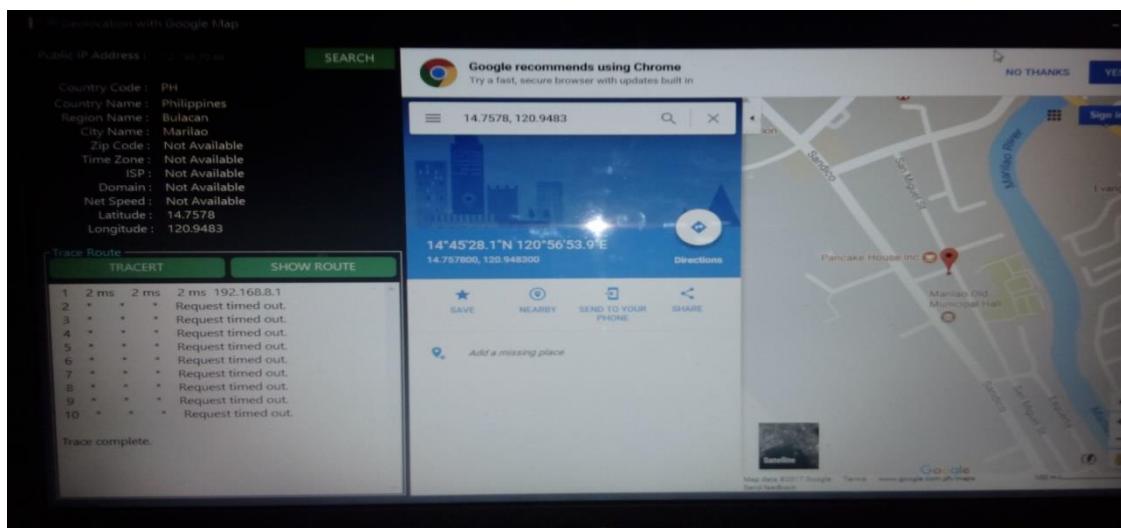
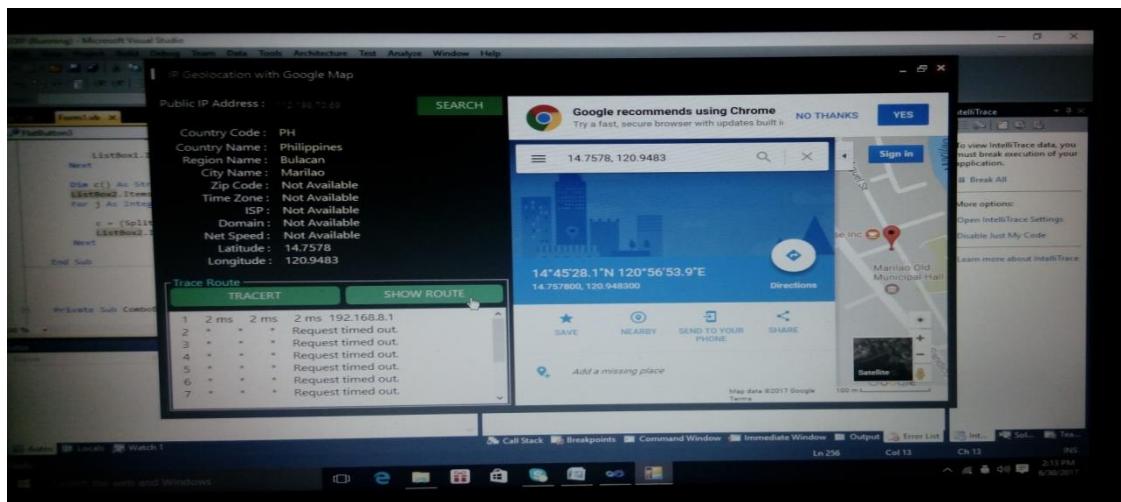
- Click the Tracert button.



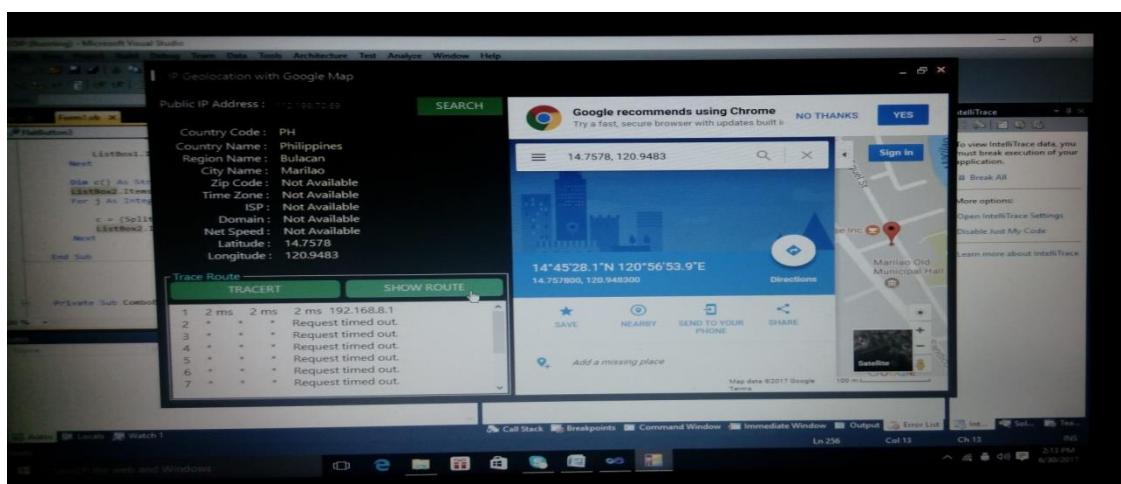
- The cmd prompt will pop up.



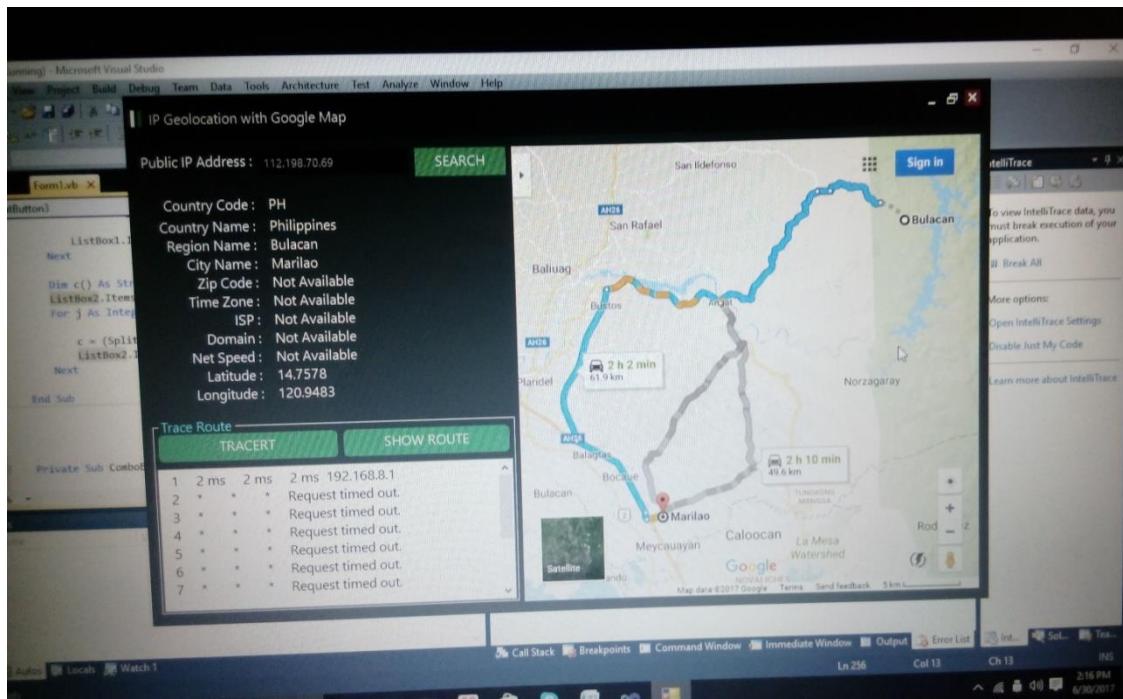
- The IP address was connected to this different connection that was converted IPs.



- After that click the button show route.



- And all show the route of the certain public IP address.



## **APPENDIX C**

### **Source Code**

```

Imports System.Xml
Imports System.Text
Imports System.Net
Imports System.Net.NetworkInformation
Imports System.Globalization

PublicClassForm1
Dim WithEvents pingSender AsNewPing()
Dim PINGERROR AsBoolean = False

PrivateConst InternetExplorerRootKeyAsString = "Software\Microsoft\Internet
Explorer"

PrivateConst BrowserEmulationKeyAsString = InternetExplorerRootKey +
"\Main\FeatureControl\FEATURE_BROWSER_EMULATION"

PublicSub SUBPING()
Try
    pingSender.SendAsync(TextBoxTRACEIP.Text, Nothing)
Catch ex AsException
    MsgBox(ex.Message)
EndTry

EndSub

PublicSub TRAZADO()
Dim IP AsIPAddress
Try
    IP = Dns.GetHostEntry(TextBoxTRACEIP.Text).AddressList(0)
Catch ex AsException
    MsgBox(ex.Message)
EndTry
Dim pingOptions AsNewPingOptions()
    pingOptions.Ttl = 1
Dim maxHops AsInteger = 30
Dim stopWatch AsNewStopwatch()

'FlatButton2.Enabled = False

For i AsInteger = 1 To maxHops

    stopWatch.Reset()
    stopWatch.Start()

Dim pingReply AsPingReply = pingSender.Send(IP, 5000, NewByte(31) {}, pingOptions)
    stopWatch.[Stop]()
Try

If PingReply.Address IsNotNothingThen

    TextBoxTRACE.Text = TextBoxTRACE.Text & i &"..." &
stopWatch.ElapsedMilliseconds.ToString &" ms. "& pingReply.Address.ToString &
vbCrLf
        ListBox1.Items.Add(pingReply.Address.ToString)
Else

    TextBoxTRACE.Text = TextBoxTRACE.Text &"Success"& vbCrLf
Exit For

```

```

EndIf

Catch ex AsException
    MsgBox(ex.Message)
EndTry

If pingReply.Status = IPStatus.Success Then
    TextBoxTRACE.Text = TextBoxTRACE.Text &"Trace Complete"& vbCrLf

Exit For
EndIf
    pingOptions.Ttl += 1
Next
    FlatButton2.Enabled = True
EndSub

PrivateSub geo()
Dim xmldoc AsNewXmlDocument
Dim xmlnode AsXmlNodeList
Dim i AsInteger

'xmldoc.Load("http://api.ip2location.com/?ip=" + FlatTextBox1.Text +
"&key=demo&package=WS24&format=xml")
xmldoc.Load("http://www.geoplugin.net/xml.gp?ip=" + FlatTextBox1.Text)
xmlnode = xmldoc.GetElementsByTagName("geoPlugin")

For i = 0 To xmlnode.Count - 1
    xmlnode(i).ChildNodes.Item(0).InnerText.Trim()

    lbl11.Text = xmlnode(i).ChildNodes.Item(7).InnerText.Trim()
    lbl12.Text = xmlnode(i).ChildNodes.Item(8).InnerText.Trim()
    lbl13.Text = xmlnode(i).ChildNodes.Item(4).InnerText.Trim()
    lbl14.Text = xmlnode(i).ChildNodes.Item(3).InnerText.Trim()
    lbl15.Text = "Not
Available'">xmlnode(i).ChildNodes.Item(6).InnerText.Trim()
    lbl16.Text = "Not Available"
    xmlnode(i).ChildNodes.Item(7).InnerText.Trim()
    lbl17.Text = "Not
Available'">xmlnode(i).ChildNodes.Item(8).InnerText.Trim()
    lbl18.Text = "Not
Available'">xmlnode(i).ChildNodes.Item(9).InnerText.Trim()
    lbl19.Text = "Not
Available'">xmlnode(i).ChildNodes.Item(10).InnerText.Trim()
    lbl110.Text = xmlnode(i).ChildNodes.Item(10).InnerText.Trim()
    lbl111.Text = xmlnode(i).ChildNodes.Item(11).InnerText.Trim()

Next
EndSub

PrivateSub FlatButton1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs)
EndSub

PrivateSub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles MyBase.Load

```

```

Try
Dim queryAddress As New StringBuilder

queryAddress.Append("https://www.google.com.ph/maps/place/Jose+Rizal+Memorial+Stat
e+University+-+
+Main+Campus/@8.6516083,123.4220903,15.25z/data=!4m!3m4!1s0x0:0xfc69ea2ff39b5bfd!
8m2!3d8.6559168!4d123.4226257?hl=en")
' queryAddress.Append("https://www.google.com/maps/dir/Current+Location/43.12345,-
76.12345")
    WebBrowser1.Navigate(queryAddress.ToString())
Catch ex As Exception
MessageBox.Show("Unable to Retrive Data")
EndTry
EndSub

Private Sub FlatButton1_Click_1(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles FlatButton1.Click
geo()

Dim lat As String = lbl10.Text
Dim lon As String = lbl11.Text

Try
Dim queryAddress As New StringBuilder

    queryAddress.Append("http://maps.google.com/maps?q=")
If lbl10.Text <> String.Empty Then
        queryAddress.Append(lat + "," & "+")
EndIf
If lbl11.Text <> String.Empty Then
        queryAddress.Append(lon)
EndIf

    WebBrowser1.Navigate(queryAddress.ToString())
Catch ex As Exception
MessageBox.Show("Unable to Retrive Data")
EndTry
EndSub

Private Sub FlatButton2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles FlatButton2.Click
FlatButton2.Enabled = False

'ListBox1.Items.Clear()
'PINGERROR = False
'TextBoxTRACE.Clear()
'TextBoxTRACEIP.Text = FlatTextBox1.Text

'TextBoxTRACEIP.SelectAll()

'If TextBoxTRACEIP.Text.Length <> 0 Then
'    SUBPING()
'    pingSender.SendAsyncCancel()
'    If PINGERROR = False Then
'        Try
'            TRAZADO()
'        Catch ex As Exception
'            MsgBox(ex.Message)
'        EndTry
'    EndIf
'EndIf
EndSub

```

```

        End Try

    Else
        TextBoxTRACE.Text = "Cant Perform Right Now"
    End If

'Else
'    MessageBox.Show("Pls. Fill up the fields")
'End If

If ListBox1.Items.Count <> 0 Then
    ListBox3.Items.Clear()

For i AsInteger = 1 To ListBox2.Items.Count - 1

If ListBox2.Items.Item(i) <>"timed"Then

    FlatTextBox1.Text = ""
    FlatTextBox1.Text = ListBox2.Items.Item(i)
    geo()
    ListBox3.Items.Add(lbl10.Text &, "& lbl11.Text)

EndIf

Next

' WebBrowser1.Navigate("https://www.google.com/maps/dir/ 8.655907,
123.423357/8.58944, 123.34139/14.08776, 120.97098/" & lbl10.Text & ", " &
lbl11.Text)

Dim query AsNewStringBuilder
'http://maps.google.com/?saddr=34.052222,-118.243611&daddr=37.322778,-122.031944

        query.Append("https://www.google.com.ph/maps/dir/")
'query.Append("http://maps.google.com/?saddr=")

For j AsInteger = 0 To ListBox3.Items.Count - 1

        query.Append(ListBox3.Items.Item(j) &"/")
' query.Append(j & "&daddr=")

Next
'query.Append("@12.4589865,120.2933587,7z/")
        WebBrowser1.Navigate(query.ToString())
EndIf

FlatButton2.Enabled = True

EndSub

PrivateSub FlatButton3_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles FlatButton3.Click
Dim oProcess AsNewProcess()
'Dim oStartInfo As New ProcessStartInfo("tracert.exe", "-d -h 10 117.92.174.86")
Dim oStartInfo AsNewProcessStartInfo("tracert.exe", "-d -h 10 "&
FlatTextBox1.Text)
oStartInfo.UseShellExecute = False
oStartInfo.RedirectStandardOutput = True
oProcess.StartInfo = oStartInfo

```

```

oProcess.Start()

Dim sOutput AsString
Using oStreamReader As System.IO.StreamReader = oProcess.StandardOutput
    sOutput = oStreamReader.ReadToEnd()
EndUsing
    TextBoxTRACE.Text = sOutput


Dim newlist AsList(OfString) = TextBoxTRACE.Lines.ToList()

    newlist.RemoveAt(newlist.Count - 1)
    newlist.RemoveAt(0)
    newlist.RemoveAt(0)
    newlist.RemoveAt(0)
    TextBoxTRACE.Lines = newlist.ToArray()

Dim b AsString() = Split(TextBoxTRACE.Text, vbNewLine)

For i AsInteger = 0 To b.Length - 1
    ListBox1.Items.Add(b(i))
Next

Dim c() AsString
    ListBox2.Items.Clear()
For j AsInteger = 0 To ListBox1.Items.Count - 3

    c = (Split(ListBox1.Items.Item(j), " "))
    ListBox2.Items.Add(c(c.Length - 2))
Next

EndSub

PrivateSub ComboBox1_SelectedIndexChanged(ByVal sender As System.Object, ByVal e As System.EventArgs)
EndSub

PrivateSub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
Dim urls AsString
    urls = WebBrowser1.Url.ToString()
    ComboBox1.Text = urls
EndSub
EndClass

Imports System.Drawing.Drawing2D, System.ComponentModel, System.Windows.Forms

''' <summary>
''' Flat UI Theme
''' Creator: iSynthesis (HF)
''' Version: 1.0.4
''' Date Created: 17/06/2013
''' Date Changed: 26/06/2013
''' UID: 374648
''' For any bugs / errors, PM me.

```

```

''' </summary>
''' <remarks></remarks>

ModuleHelpers

#Region" Variables"
Friend G AsGraphics, B AsBitmap
Friend _FlatColor AsColor = Color.FromArgb(35, 168, 109)
Friend NearSF AsNewStringFormat() With {.Alignment = StringAlignment.Near,
.LineAlignment = StringAlignment.Near}
Friend CenterSF AsNewStringFormat() With {.Alignment = StringAlignment.Center,
.LineAlignment = StringAlignment.Center}
#EndRegion

#Region" Functions"

PublicFunction RoundRec(ByVal Rectangle AsRectangle, ByVal Curve AsInteger)
AsGraphicsPath
Dim P As GraphicsPath = New GraphicsPath()
Dim ArcRectangleWidth AsInteger = Curve * 2
    P.AddArc(New Rectangle(Rectangle.X, Rectangle.Y, ArcRectangleWidth,
ArcRectangleWidth), -180, 90)
    P.AddArc(New Rectangle(Rectangle.Width - ArcRectangleWidth + Rectangle.X,
Rectangle.Y, ArcRectangleWidth, ArcRectangleWidth), -90, 90)
    P.AddArc(New Rectangle(Rectangle.Width - ArcRectangleWidth + Rectangle.X,
Rectangle.Height - ArcRectangleWidth + Rectangle.Y, ArcRectangleWidth,
ArcRectangleWidth), 0, 90)
    P.AddArc(New Rectangle(Rectangle.X, Rectangle.Height - ArcRectangleWidth +
Rectangle.Y, ArcRectangleWidth, ArcRectangleWidth), 90, 90)
    P.AddLine(New Point(Rectangle.X, Rectangle.Height - ArcRectangleWidth +
Rectangle.Y), New Point(Rectangle.X, Curve + Rectangle.Y))
Return P
EndFunction

PublicFunction RoundRect(ByVal x!, ByVal y!, ByVal w!, ByVal h!, Optional ByVal r!
= 0.3, Optional ByVal TL AsBoolean = True, Optional ByVal TR AsBoolean = True,
Optional ByVal BR AsBoolean = True, Optional ByVal BL AsBoolean = True)
AsGraphicsPath
Dim d! = Math.Min(w, h) * r, xw = x + w, yh = y + h
    RoundRect = New GraphicsPath

    With RoundRect
        If TL Then .AddArc(x, y, d, d, 180, 90) Else .AddLine(x, y, x, y)
        If TR Then .AddArc(xw - d, y, d, d, 270, 90) Else .AddLine(xw, y, xw, y)
        If BR Then .AddArc(xw - d, yh - d, d, d, 0, 90) Else .AddLine(xw, yh, xw, yh)
        If BL Then .AddArc(x, yh - d, d, d, 90, 90) Else .AddLine(x, yh, x, yh)

        .CloseFigure()
    EndWith
EndFunction

'-- Credit: AeonHack
PublicFunction DrawArrow(ByVal x AsInteger, ByVal y AsInteger, ByVal flip
AsBoolean) AsGraphicsPath
Dim GP AsNew GraphicsPath()

Dim W AsInteger = 12
Dim H AsInteger = 6

If flip Then
    GP.AddLine(x + 1, y, x + W + 1, y)
    GP.AddLine(x + W, y, x + H, y + H - 1)

```

```

Else
    GP.AddLine(x, y + H, x + W, y + H)
    GP.AddLine(x + W, y + H, x + H, y)
EndIf

    GP.CloseFigure()
Return GP
EndFunction

#EndRegion

EndModule

#Region" Mouse States"

EnumMouseStateAsByte
    None = 0
    Over = 1
    Down = 2
    Block = 3
EndEnum

#EndRegion

ClassFormSkin : InheritsContainerControl

#Region" Variables"

Private W, H AsInteger
Private Cap AsBoolean = False
Private _HeaderMaximize AsBoolean = False
Private MousePoint AsNewPoint(0, 0)
Private MoveHeight = 50

#EndRegion

#Region" Properties"

#Region" Colors"

<Category("Colors")>
PublicProperty HeaderColor() AsColor
Get
Return _HeaderColor
EndGet
Set(ByVal value As Color)
    _HeaderColor = value
EndSet
EndProperty
<Category("Colors")>
PublicProperty BaseColor() AsColor
Get
Return _BaseColor
EndGet
Set(ByVal value As Color)
    _BaseColor = value
EndSet
EndProperty
<Category("Colors")>
PublicProperty BorderColor() AsColor
Get
Return _BorderColor
EndGet
Set(ByVal value As Color)
    _BorderColor = value
EndSet
EndProperty

```

```

EndGet
Set(ByVal value As Color)
    _BorderColor = value
EndSet
EndProperty
<Category("Colors")>
PublicProperty FlatColor() AsColor
Get
Return _FlatColor
EndGet
Set(ByVal value As Color)
    _FlatColor = value
EndSet
EndProperty

#EndRegion

#Region" Options"

<Category("Options")> _
PublicProperty HeaderMaximize AsBoolean
Get
Return _HeaderMaximize
EndGet
Set(ByVal value AsBoolean)
    _HeaderMaximize = value
EndSet
EndProperty

#EndRegion

ProtectedOverridesSub OnMouseDown(ByVal e As MouseEventArgs)
 MyBase.OnMouseDown(e)
 If e.Button = Windows.Forms.MouseButtons.Left AndNew Rectangle(0, 0, Width,
 MoveHeight).Contains(e.Location) Then
     Cap = True
     MousePoint = e.Location
 EndIf
EndSub

PrivateSub FormSkin_MouseDoubleClick(ByVal sender AsObject, ByVal e
 AsEventArgs) HandlesMe.MouseDoubleClick
 If HeaderMaximize Then
     If e.Button = Windows.Forms.MouseButtons.Left AndNew Rectangle(0, 0, Width,
 MoveHeight).Contains(e.Location) Then
         If FindForm.WindowState = FormWindowState.Normal Then
             FindForm.WindowState = FormWindowState.Maximized :
             FindForm.Refresh()
         ElseIf FindForm.WindowState = FormWindowState.Maximized Then
             FindForm.WindowState = FormWindowState.Normal :
             FindForm.Refresh()
         EndIf
     EndIf
     EndIf
 EndSub

ProtectedOverridesSub OnMouseUp(ByVal e As MouseEventArgs)
 MyBase.OnMouseUp(e) : Cap = False
EndSub

ProtectedOverridesSub OnMouseMove(ByVal e As MouseEventArgs)
 MyBase.OnMouseMove(e)

```

```

If Cap Then
    Parent.Location = MousePosition - MousePoint
EndIf
EndSub

ProtectedOverridesSub OnCreateControl()
MyBase.OnCreateControl()
    ParentForm.FormBorderStyle = FormBorderStyle.None
    ParentForm.AllowTransparency = False
    ParentForm.TransparencyKey = Color.Fuchsia
    ParentForm.FindForm.StartPosition = FormStartPosition.CenterScreen
    Dock = DockStyle.Fill
    Invalidate()
EndSub

#EndRegion

#Region" Colors"
#Region" Dark Colors"

Private _HeaderColor AsColor = Color.FromArgb(45, 47, 49)
Private _BaseColor AsColor = Color.FromArgb(60, 70, 73)
Private _BorderColor AsColor = Color.FromArgb(53, 58, 60)
Private TextColor AsColor = Color.FromArgb(234, 234, 234)

#EndRegion

#Region" Light Colors"

Private _HeaderLight AsColor = Color.FromArgb(171, 171, 172)
Private _BaseLight AsColor = Color.FromArgb(196, 199, 200)
Public TextLight AsColor = Color.FromArgb(45, 47, 49)

#EndRegion

#EndRegion

SubNew()
    SetStyle(ControlStyles.AllPaintingInWmPaint Or ControlStyles.UserPaint Or
    -
        ControlStyles.ResizeRedraw Or
    ControlStyles.OptimizedDoubleBuffer, True)
    DoubleBuffered = True
    BackColor = Color.White
    Font = New Font("Segoe UI", 12)
EndSub

ProtectedOverridesSub OnPaint(ByVal e AsPaintEventArgs)
    B = New Bitmap(Width, Height) : G = Graphics.FromImage(B)
    W = Width : H = Height

Dim Base AsNew Rectangle(0, 0, W, H), Header AsNew Rectangle(0, 0, W, 50)

With G
    .SmoothingMode = 2
    .PixelOffsetMode = 2
    .TextRenderingHint = 5
    .Clear(BackColor)

'-- Base
    .FillRectangle(New SolidBrush(_BaseColor), Base)

```

```

'-- Header
    .FillRectangle(New SolidBrush(_HeaderColor), Header)

'-- Logo
    .FillRectangle(New SolidBrush(Color.FromArgb(243, 243, 243)), New
    Rectangle(8, 16, 4, 18))
    .FillRectangle(New SolidBrush(_FlatColor), 16, 16, 4, 18)
    .DrawString(Text, Font, New SolidBrush(TextColor), New Rectangle(26,
    15, W, H), NearSF)

'-- Border
    .DrawRectangle(New Pen(_BorderColor), Base)
EndWith

 MyBase.OnPaint(e)
    G.Dispose()
    e.Graphics.InterpolationMode = 7
    e.Graphics.DrawImageUnscaled(B, 0, 0)
    B.Dispose()
EndSub
EndClass

ClassFlatClose : InheritsControl

#Region" Variables"

Private State AsMouseState = MouseState.None
Private x AsInteger

#EndRegion

#Region" Properties"

#Region" Mouse States"

ProtectedOverridesSub OnMouseEnter(ByVal e AsEventArgs)
MyBase.OnMouseEnter(e)
    State = MouseState.Over : Invalidate()
EndSub
ProtectedOverridesSub OnMouseDown(ByVal e AsMouseEventArgs)
MyBase.OnMouseDown(e)
    State = MouseState.Down : Invalidate()
EndSub
ProtectedOverridesSub OnMouseLeave(ByVal e AsEventArgs)
MyBase.OnMouseLeave(e)
    State = MouseState.None : Invalidate()
EndSub
ProtectedOverridesSub OnMouseUp(ByVal e AsMouseEventArgs)
MyBase.OnMouseUp(e)
    State = MouseState.Over : Invalidate()
EndSub
ProtectedOverridesSub OnMouseMove(ByVal e AsMouseEventArgs)
MyBase.OnMouseMove(e)
    x = e.X : Invalidate()
EndSub

ProtectedOverridesSub OnClick(ByVal e AsEventArgs)
MyBase.OnClick(e)
Environment.Exit(0)
EndSub

```

```

#EndRegion

ProtectedOverridesSub OnResize(ByVal e AsEventArgs)
 MyBase.OnResize(e)
    Size = New Size(18, 18)
EndSub

#Region" Colors"

<Category("Colors")> _
PublicProperty BaseColor AsColor
Get
Return _BaseColor
EndGet
Set(ByVal value As Color)
    _BaseColor = value
EndSet
EndProperty

<Category("Colors")> _
PublicProperty TextColor AsColor
Get
Return _TextColor
EndGet
Set(ByVal value As Color)
    _TextColor = value
EndSet
EndProperty

#EndRegion

#EndRegion

#Region" Colors"

Private _BaseColor AsColor = Color.FromArgb(168, 35, 35)
Private _TextColor AsColor = Color.FromArgb(243, 243, 243)

#EndRegion

SubNew()
    SetStyle(ControlStyles.AllPaintingInWmPaint Or ControlStyles.UserPaint Or
    -
        ControlStyles.ResizeRedraw Or
    ControlStyles.OptimizedDoubleBuffer, True)
        DoubleBuffered = True
        BackColor = Color.White
        Size = New Size(18, 18)
        Anchor = AnchorStyles.Top Or AnchorStyles.Right
        Font = New Font("Marlett", 10)
EndSub

ProtectedOverridesSub OnPaint(ByVal e AsPaintEventArgs)
Dim B AsNew Bitmap(Width, Height)
Dim G As Graphics = Graphics.FromImage(B)

Dim Base AsNew Rectangle(0, 0, Width, Height)

With G
    .SmoothingMode = 2
    .PixelOffsetMode = 2
    .TextRenderingHint = 5

```

```

        .Clear(BackColor)

'-- Base
        .FillRectangle(New SolidBrush(_BaseColor), Base)

'-- X
        .DrawString("r", Font, New SolidBrush(TextColor), New Rectangle(0, 0,
Width, Height), CenterSF)

'-- Hover/down
SelectCase State
Case MouseState.Over
        .FillRectangle(New SolidBrush(Color.FromArgb(30,
Color.White)), Base)
Case MouseState.Down
        .FillRectangle(New SolidBrush(Color.FromArgb(30,
Color.Black)), Base)
EndSelect
EndWith

 MyBase.OnPaint(e)
    G.Dispose()
    e.Graphics.InterpolationMode = 7
    e.Graphics.DrawImageUnscaled(B, 0, 0)
    B.Dispose()
EndSub
EndClass

ClassFlatMax : InheritsControl

#Region" Variables"

Private State AsMouseState = MouseState.None
Private x AsInteger

#EndRegion

#Region" Properties"

#Region" Mouse States"

ProtectedOverridesSub OnMouseEnter(ByVal e AsEventArgs)
MyBase.OnMouseEnter(e)
    State = MouseState.Over : Invalidate()
EndSub
ProtectedOverridesSub OnMouseDown(ByVal e AsMouseEventArgs)
MyBase.OnMouseDown(e)
    State = MouseState.Down : Invalidate()
EndSub
ProtectedOverridesSub OnMouseLeave(ByVal e AsEventArgs)
MyBase.OnMouseLeave(e)
    State = MouseState.None : Invalidate()
EndSub
ProtectedOverridesSub OnMouseUp(ByVal e AsMouseEventArgs)
MyBase.OnMouseUp(e)
    State = MouseState.Over : Invalidate()
EndSub
ProtectedOverridesSub OnMouseMove(ByVal e AsMouseEventArgs)
MyBase.OnMouseMove(e)
    x = e.X : Invalidate()
EndSub

```

```

ProtectedOverridesSub OnClick(ByVal e AsEventArgs)
 MyBase.OnClick(e)
 SelectCase FindForm.WindowState
 Case FormWindowState.Maximized
     FindForm.WindowState = FormWindowState.Normal
 Case FormWindowState.Normal
     FindForm.WindowState = FormWindowState.Maximized
 EndSelect
 EndSub

#EndRegion

#Region" Colors"

<Category("Colors")> _
PublicProperty BaseColor AsColor
Get
Return _BaseColor
EndGet
Set(ByVal value As Color)
    _BaseColor = value
EndSet
EndProperty

<Category("Colors")> _
PublicProperty TextColor AsColor
Get
Return _TextColor
EndGet
Set(ByVal value As Color)
    _TextColor = value
EndSet
EndProperty

#EndRegion

ProtectedOverridesSub OnResize(ByVal e AsEventArgs)
 MyBase.OnResize(e)
     Size = New Size(18, 18)
EndSub

#EndRegion

#Region" Colors"

Private _BaseColor AsColor = Color.FromArgb(45, 47, 49)
Private _TextColor AsColor = Color.FromArgb(243, 243, 243)

#EndRegion

SubNew()
    SetStyle(ControlStyles.AllPaintingInWmPaint Or ControlStyles.UserPaint Or
    -
        ControlStyles.ResizeRedraw Or
ControlStyles.OptimizedDoubleBuffer, True)
    DoubleBuffered = True
    BackColor = Color.White
    Size = New Size(18, 18)
    Anchor = AnchorStyles.Top Or AnchorStyles.Right
    Font = New Font("Marlett", 12)
EndSub

```

```

ProtectedOverridesSub OnPaint(ByVal e AsPaintEventArgs)
Dim B AsNew Bitmap(Width, Height)
Dim G As Graphics = Graphics.FromImage(B)

Dim Base AsNew Rectangle(0, 0, Width, Height)

With G
    .SmoothingMode = 2
    .PixelOffsetMode = 2
    .TextRenderingHint = 5
    .Clear(BackColor)

    '-- Base
    .FillRectangle(New SolidBrush(_BaseColor), Base)

    '-- Maximize
    If FindForm.WindowState = FormWindowState.Maximized Then
        .DrawString("1", Font, New SolidBrush(TextColor), New Rectangle(1,
1, Width, Height), CenterSF)
    ElseIf FindForm.WindowState = FormWindowState.Normal Then
        .DrawString("2", Font, New SolidBrush(TextColor), New Rectangle(1,
1, Width, Height), CenterSF)
    EndIf

    '-- Hover/down
    SelectCase State
    Case MouseState.Over
        .FillRectangle(New SolidBrush(Color.FromArgb(30,
Color.White)), Base)
    Case MouseState.Down
        .FillRectangle(New SolidBrush(Color.FromArgb(30,
Color.Black)), Base)
    EndSelect
    EndWith

 MyBase.OnPaint(e)
    G.Dispose()
    e.Graphics.InterpolationMode = 7
    e.Graphics.DrawImageUnscaled(B, 0, 0)
    B.Dispose()
EndSub
EndClass

ClassFlatMini : InheritsControl

#Region" Variables"

Private State AsMouseState = MouseState.None
Private x AsInteger

#EndRegion

#Region" Properties"

#Region" Mouse States"

ProtectedOverridesSub OnMouseEnter(ByVal e AsEventArgs)
MyBase.OnMouseEnter(e)
    State = MouseState.Over : Invalidate()
EndSub
ProtectedOverridesSub OnMouseDown(ByVal e AsMouseEventArgs)
MyBase.OnMouseDown(e)

```

```

        State = MouseState.Down : Invalidate()
EndSub
ProtectedOverridesSub OnMouseLeave(ByVal e AsEventArgs)
 MyBase.OnMouseLeave(e)
 State = MouseState.None : Invalidate()
EndSub
ProtectedOverridesSub OnMouseUp(ByVal e AsMouseEventArgs)
 MyBase.OnMouseUp(e)
 State = MouseState.Over : Invalidate()
EndSub
ProtectedOverridesSub OnMouseMove(ByVal e AsMouseEventArgs)
 MyBase.OnMouseMove(e)
 x = e.X : Invalidate()
EndSub

ProtectedOverridesSub OnClick(ByVal e AsEventArgs)
 MyBase.OnClick(e)
 SelectCase FindForm.WindowState
 Case FormWindowState.Normal
 FindForm.WindowState = FormWindowState.Minimized
 Case FormWindowState.Maximized
 FindForm.WindowState = FormWindowState.Minimized
 EndSelect
EndSub

#EndRegion

#Region" Colors"

<Category("Colors")> _
PublicProperty BaseColor AsColor
Get
Return _BaseColor
EndGet
Set(ByVal value As Color)
_BaseColor = value
EndSet
EndProperty

<Category("Colors")> _
PublicProperty TextColor AsColor
Get
Return _TextColor
EndGet
Set(ByVal value As Color)
_TextColor = value
EndSet
EndProperty

#EndRegion

ProtectedOverridesSub OnResize(ByVal e AsEventArgs)
 MyBase.OnResize(e)
 Size = New Size(18, 18)
EndSub

#EndRegion

#Region" Colors"

Private _BaseColor AsColor = Color.FromArgb(45, 47, 49)
Private _TextColor AsColor = Color.FromArgb(243, 243, 243)

```

```

#EndRegion

SubNew()
    SetStyle(ControlStyles.AllPaintingInWmPaint Or ControlStyles.UserPaint Or
    -
        ControlStyles.ResizeRedraw Or
    ControlStyles.OptimizedDoubleBuffer, True)
        DoubleBuffered = True
        BackColor = Color.White
        Size = New Size(18, 18)
        Anchor = AnchorStyles.Top Or AnchorStyles.Right
        Font = New Font("Marlett", 12)
EndSub

ProtectedOverridesSub OnPaint(ByVal e AsPaintEventArgs)
Dim B AsNew Bitmap(Width, Height)
Dim G As Graphics = Graphics.FromImage(B)

Dim Base AsNew Rectangle(0, 0, Width, Height)

With G
    .SmoothingMode = 2
    .PixelOffsetMode = 2
    .TextRenderingHint = 5
    .Clear(BackColor)

'-- Base
    .FillRectangle(New SolidBrush(_BaseColor), Base)

'-- Minimize
    .DrawString("0", Font, New SolidBrush(TextColor), New Rectangle(2, 1,
    Width, Height), CenterSF)

'-- Hover/down
SelectCase State
Case MouseState.Over
    .FillRectangle(New SolidBrush(Color.FromArgb(30,
    Color.White)), Base)
Case MouseState.Down
    .FillRectangle(New SolidBrush(Color.FromArgb(30,
    Color.Black)), Base)
EndSelect
EndWith

 MyBase.OnPaint(e)
    G.Dispose()
    e.Graphics.InterpolationMode = 7
    e.Graphics.DrawImageUnscaled(B, 0, 0)
    B.Dispose()
EndSub
EndClass

ClassFlatColorPalette : InheritsControl

#Region" Variables"

Private W, H AsInteger

#EndRegion

#Region" Properties"

```

```

ProtectedOverridesSub OnResize(ByVal e AsEventArgs)
 MyBase.OnResize(e)
    Width = 180
    Height = 80
EndSub

#Region" Colors"

<Category("Colors")> _
PublicProperty Red AsColor
Get
Return _Red
EndGet
Set(ByVal value As Color)
    _Red = value
EndSet
EndProperty

<Category("Colors")> _
PublicProperty Cyan AsColor
Get
Return _Cyan
EndGet
Set(ByVal value As Color)
    _Cyan = value
EndSet
EndProperty

<Category("Colors")> _
PublicProperty Blue AsColor
Get
Return _Blue
EndGet
Set(ByVal value As Color)
    _Blue = value
EndSet
EndProperty

<Category("Colors")> _
PublicProperty LimeGreen AsColor
Get
Return _LimeGreen
EndGet
Set(ByVal value As Color)
    _LimeGreen = value
EndSet
EndProperty

<Category("Colors")> _
PublicProperty Orange AsColor
Get
Return _Orange
EndGet
Set(ByVal value As Color)
    _Orange = value
EndSet
EndProperty

<Category("Colors")> _
PublicProperty Purple AsColor
Get

```

```

Return _Purple
EndGet
Set(ByVal value As Color)
    _Purple = value
EndSet
EndProperty

<Category("Colors")> _
PublicProperty Black AsColor
Get
Return _Black
EndGet
Set(ByVal value As Color)
    _Black = value
EndSet
EndProperty

<Category("Colors")> _
PublicProperty Gray AsColor
Get
Return _Gray
EndGet
Set(ByVal value As Color)
    _Gray = value
EndSet
EndProperty

<Category("Colors")> _
PublicProperty White AsColor
Get
Return _White
EndGet
Set(ByVal value As Color)
    _White = value
EndSet
EndProperty

#EndRegion

#EndRegion

#Region" Colors"

Private _Red AsColor = Color.FromArgb(220, 85, 96)
Private _Cyan AsColor = Color.FromArgb(10, 154, 157)
Private _Blue AsColor = Color.FromArgb(0, 128, 255)
Private _LimeGreen AsColor = Color.FromArgb(35, 168, 109)
Private _Orange AsColor = Color.FromArgb(253, 181, 63)
Private _Purple AsColor = Color.FromArgb(155, 88, 181)
Private _Black AsColor = Color.FromArgb(45, 47, 49)
Private _Gray AsColor = Color.FromArgb(63, 70, 73)
Private _White AsColor = Color.FromArgb(243, 243, 243)

#EndRegion

SubNew()
    SetStyle(ControlStyles.AllPaintingInWmPaint Or ControlStyles.UserPaint Or
    ControlStyles.OptimizedDoubleBuffer, True)
    DoubleBuffered = True
    BackColor = Color.FromArgb(60, 70, 73)

```

```

        Size = New Size(160, 80)
        Font = New Font("Segoe UI", 12)
EndSub

ProtectedOverridesSub OnPaint(ByVal e AsPaintEventArgs)
    B = New Bitmap(Width, Height) : G = Graphics.FromImage(B)
    W = Width - 1 : H = Height - 1

    With G
        .SmoothingMode = 2
        .PixelOffsetMode = 2
        .TextRenderingHint = 5
        .Clear(BackColor)

        '-- Colors
        .FillRectangle(New SolidBrush(_Red), New Rectangle(0, 0, 20, 40))
        .FillRectangle(New SolidBrush(_Cyan), New Rectangle(20, 0, 20, 40))
        .FillRectangle(New SolidBrush(_Blue), New Rectangle(40, 0, 20, 40))
        .FillRectangle(New SolidBrush(_LimeGreen), New Rectangle(60, 0, 20,
40))
        .FillRectangle(New SolidBrush(_Orange), New Rectangle(80, 0, 20, 40))
        .FillRectangle(New SolidBrush(_Purple), New Rectangle(100, 0, 20, 40))
        .FillRectangle(New SolidBrush(_Black), New Rectangle(120, 0, 20, 40))
        .FillRectangle(New SolidBrush(_Gray), New Rectangle(140, 0, 20, 40))
        .FillRectangle(New SolidBrush(_White), New Rectangle(160, 0, 20, 40))

        '-- Text
        .DrawString("Color Palette", Font, New SolidBrush(_White), New
Rectangle(0, 22, W, H), CenterSF)
    EndWith

 MyBase.OnPaint(e)
    G.Dispose()
    e.Graphics.InterpolationMode = 7
    e.Graphics.DrawImageUnscaled(B, 0, 0)
    B.Dispose()
EndSub
EndClass

ClassFlatGroupBox : InheritsContainerControl

#Region" Variables"

Private W, H AsInteger
Private _ShowText AsBoolean = True

#EndRegion

#Region" Properties"

<Category("Colors")> _
PublicProperty BaseColor AsColor
Get
Return _BaseColor
EndGet
Set(ByVal value As Color)
    _BaseColor = value
EndSet
EndProperty

PublicProperty ShowText AsBoolean
Get

```

```

Return _ShowText
EndGet
Set(ByName value AsBoolean)
    _ShowText = value
EndSet
EndProperty

#EndRegion

#Region" Colors"

Private _BaseColor AsColor = Color.FromArgb(60, 70, 73)

#EndRegion

SubNew()
    SetStyle(ControlStyles.AllPaintingInWmPaint Or ControlStyles.UserPaint Or
    - ControlStyles.ResizeRedraw Or ControlStyles.OptimizedDoubleBuffer
Or _ ControlStyles.SupportsTransparentBackColor, True)
    DoubleBuffered = True
    BackColor = Color.Transparent
    Size = New Size(240, 180)
    Font = New Font("Segoe ui", 10)
EndSub

ProtectedOverridesSub OnPaint(ByVal e AsPaintEventArgs)
    B = New Bitmap(Width, Height) : G = Graphics.FromImage(B)
    W = Width - 1 : H = Height - 1

    Dim GP, GP2, GP3 AsNew GraphicsPath
    Dim Base AsNew Rectangle(8, 8, W - 16, H - 16)

    With G
        .SmoothingMode = 2
        .PixelOffsetMode = 2
        .TextRenderingHint = 5
        .Clear(BackColor)

        '-- Base
        GP = Helpers.RoundRec(Base, 8)
        .FillPath(New SolidBrush(_BaseColor), GP)

        '-- Arrows
        GP2 = Helpers.DrawArrow(28, 2, False)
        .FillPath(New SolidBrush(_BaseColor), GP2)
        GP3 = Helpers.DrawArrow(28, 8, True)
        .FillPath(New SolidBrush(Color.FromArgb(60, 70, 73)), GP3)

        '-- if ShowText
        If ShowText Then
            .DrawString(Text, Font, New SolidBrush(_FlatColor), New
            Rectangle(16, 16, W, H), NearSF)
        EndIf
    EndWith

    MyBase.OnPaint(e)
    G.Dispose()
    e.Graphics.InterpolationMode = 7
    e.Graphics.DrawImageUnscaled(B, 0, 0)
    B.Dispose()

```

```

EndSub
EndClass

ClassFlatButton : InheritsControl

#Region" Variables"

Private W, H AsInteger
Private _Rounded AsBoolean = False
Private State AsMouseState = MouseState.None

#EndRegion

#Region" Properties"

#Region" Colors"

<Category("Colors")>
PublicProperty BaseColor AsColor
Get
Return _BaseColor
EndGet
Set(ByVal value As Color)
    _BaseColor = value
EndSet
EndProperty

<Category("Colors")>
PublicProperty TextColor AsColor
Get
Return _TextColor
EndGet
Set(ByVal value As Color)
    _TextColor = value
EndSet
EndProperty

<Category("Options")>
PublicProperty Rounded AsBoolean
Get
Return _Rounded
EndGet
Set(ByVal value AsBoolean)
    _Rounded = value
EndSet
EndProperty

#EndRegion

#Region" Mouse States"

ProtectedOverridesSub OnMouseDown(ByVal e As MouseEventArgs)
 MyBase.OnMouseDown(e)
     State = MouseState.Down : Invalidate()
EndSub
ProtectedOverridesSub OnMouseUp(ByVal e As MouseEventArgs)
 MyBase.OnMouseUp(e)
     State = MouseState.Over : Invalidate()
EndSub
ProtectedOverridesSub OnMouseEnter(ByVal e As EventArgs)
 MyBase.OnMouseEnter(e)
     State = MouseState.Over : Invalidate()

```

```

EndSub
ProtectedOverridesSub OnMouseLeave(ByVal e AsEventArgs)
 MyBase.OnMouseLeave(e)
 State = MouseState.None : Invalidate()
EndSub

#EndRegion

#EndRegion

#Region" Colors"

Private _BaseColor AsColor = _FlatColor
Private _TextColor AsColor = Color.FromArgb(243, 243, 243)

#EndRegion

SubNew()
 SetStyle(ControlStyles.AllPaintingInWmPaint Or ControlStyles.UserPaint Or
 -
 ControlStyles.ResizeRedraw Or ControlStyles.OptimizedDoubleBuffer
 Or -
 ControlStyles.SupportsTransparentBackColor, True)
 DoubleBuffered = True
 Size = New Size(106, 32)
 BackColor = Color.Transparent
 Font = New Font("Segoe UI", 12)
 Cursor = Cursors.Hand
EndSub

ProtectedOverridesSub OnPaint(ByVal e AsPaintEventArgs)
 B = New Bitmap(Width, Height) : G = Graphics.FromImage(B)
 W = Width - 1 : H = Height - 1

Dim GP AsNew GraphicsPath
Dim Base AsNew Rectangle(0, 0, W, H)

With G
 .SmoothingMode = 2
 .PixelOffsetMode = 2
 .TextRenderingHint = 5
 .Clear(BackColor)

SelectCase State
Case MouseState.None
If Rounded Then
'-- Base
 GP = Helpers.RoundRec(Base, 6)
 .FillPath(New SolidBrush(_BaseColor), GP)

'-- Text
 .DrawString(Text, Font, New SolidBrush(_TextColor), Base,
CenterSF)
Else
'-- Base
 .FillRectangle(New SolidBrush(_BaseColor), Base)

'-- Text
 .DrawString(Text, Font, New SolidBrush(_TextColor), Base,
CenterSF)
EndIf
Case MouseState.Over

```

```

If Rounded Then
'-- Base
    GP = Helpers.RoundRec(Base, 6)
    .FillPath(New SolidBrush(_BaseColor), GP)
    .FillPath(New SolidBrush(Color.FromArgb(20, Color.White)), 
    GP)

'-- Text
    .DrawString(Text, Font, New SolidBrush(_TextColor), Base,
    CenterSF)
Else
'-- Base
    .FillRectangle(New SolidBrush(_BaseColor), Base)
    .FillRectangle(New SolidBrush(Color.FromArgb(20,
    Color.White)), Base)

'-- Text
    .DrawString(Text, Font, New SolidBrush(_TextColor), Base,
    CenterSF)
EndIf
Case MouseState.Down
If Rounded Then
'-- Base
    GP = Helpers.RoundRec(Base, 6)
    .FillPath(New SolidBrush(_BaseColor), GP)
    .FillPath(New SolidBrush(Color.FromArgb(20, Color.Black)), 
    GP)

'-- Text
    .DrawString(Text, Font, New SolidBrush(_TextColor), Base,
    CenterSF)
Else
'-- Base
    .FillRectangle(New SolidBrush(_BaseColor), Base)
    .FillRectangle(New SolidBrush(Color.FromArgb(20,
    Color.Black)), Base)

'-- Text
    .DrawString(Text, Font, New SolidBrush(_TextColor), Base,
    CenterSF)
EndIf
EndSelect
EndWith

 MyBase.OnPaint(e)
    G.Dispose()
    e.Graphics.InterpolationMode = 7
    e.Graphics.DrawImageUnscaled(B, 0, 0)
    B.Dispose()
EndSub
EndClass

<DefaultEvent("CheckedChanged")>ClassFlatToggle : InheritsControl

#Region" Variables"

Private W, H AsInteger
Private O As_Options
Private _Checked AsBoolean = False
Private State AsMouseState = MouseState.None

#EndRegion

```

```

#Region" Properties"
PublicEvent CheckedChanged(ByVal sender AsObject)

<Flags()> _
Enum _Options
    Style1
    Style2
    Style3
    Style4 '-- TODO: New Style
    Style5 '-- TODO: New Style
EndEnum

#Region" Options"
<Category("Options")> _
PublicProperty Options As _Options
Get
Return 0
EndGet
Set(ByVal value As _Options)
    0 = value
EndSet
EndProperty

<Category("Options")> _
PublicProperty Checked AsBoolean
Get
Return _Checked
EndGet
Set(ByVal value AsBoolean)
    _Checked = value
EndSet
EndProperty

#EndRegion

ProtectedOverridesSub OnTextChanged(ByVal e AsEventArgs)
 MyBase.OnTextChanged(e) : Invalidate()
EndSub

ProtectedOverridesSub OnResize(ByVal e AsEventArgs)
 MyBase.OnResize(e)
    Width = 76
    Height = 33
EndSub

#Region" Mouse States"

ProtectedOverridesSub OnMouseEnter(ByVal e As System.EventArgs)
 MyBase.OnMouseEnter(e)
    State = MouseState.Over : Invalidate()
EndSub
ProtectedOverridesSub OnMouseDown(ByVal e As System.Windows.Forms.MouseEventArgs)
 MyBase.OnMouseDown(e)
    State = MouseState.Down : Invalidate()
EndSub
ProtectedOverridesSub OnMouseLeave(ByVal e As System.EventArgs)
 MyBase.OnMouseLeave(e)
    State = MouseState.None : Invalidate()
EndSub
ProtectedOverridesSub OnMouseUp(ByVal e As System.Windows.Forms.MouseEventArgs)

```

```

 MyBase.OnMouseUp(e)
     State = MouseState.Over : Invalidate()
 EndSub
 ProtectedOverridesSub OnClick(ByVal e AsEventArgs)
 MyBase.OnClick(e)
     _Checked = Not _Checked
 RaiseEvent CheckedChanged(Me)
 EndSub

#EndRegion

#EndRegion

#Region" Colors"

Private BaseColor AsColor = _FlatColor
Private BaseColorRed AsColor = Color.FromArgb(220, 85, 96)
Private BGColor AsColor = Color.FromArgb(84, 85, 86)
Private ToggleColor AsColor = Color.FromArgb(45, 47, 49)
Private TextColor AsColor = Color.FromArgb(243, 243, 243)

#EndRegion

SubNew()
    SetStyle(ControlStyles.AllPaintingInWmPaint Or ControlStyles.UserPaint Or
    -           ControlStyles.ResizeRedraw Or ControlStyles.OptimizedDoubleBuffer
Or -           ControlStyles.SupportsTransparentBackColor, True)
    DoubleBuffered = True
    BackColor = Color.Transparent
    Size = New Size(44, Height + 1)
    Cursor = Cursors.Hand
    Font = New Font("Segoe UI", 10)
    Size = New Size(76, 33)
EndSub

ProtectedOverridesSub OnPaint(ByVal e AsPaintEventArgs)
    B = New Bitmap(Width, Height) : G = Graphics.FromImage(B)
    W = Width - 1 : H = Height - 1

    Dim GP, GP2 AsNew GraphicsPath
    Dim Base AsNew Rectangle(0, 0, W, H), Toggle AsNew Rectangle(CInt(W \ 2), 0, 38,
H)

    With G
        .SmoothingMode = 2
        .PixelOffsetMode = 2
        .TextRenderingHint = 5
        .Clear(BackColor)
    EndWith

    SelectCase 0
    Case _Options.Style1    '-- Style 1
    '-- Base
        GP = Helpers.RoundRec(Base, 6)
        GP2 = Helpers.RoundRec(Toggle, 6)
        .FillPath(New SolidBrush(BGColor), GP)
        .FillPath(New SolidBrush(ToggleColor), GP2)

    '-- Text
        .DrawString("OFF", Font, New SolidBrush(BGColor), New
Rectangle(19, 1, W, H), CenterSF)
    EndSelect
EndSub

```

```

If Checked Then
'-- Base
    GP = Helpers.RoundRec(Base, 6)
    GP2 = Helpers.RoundRec(New Rectangle(CInt(W \ 2), 0, 38,
H), 6)
        .FillPath(New SolidBrush(ToggleColor), GP)
        .FillPath(New SolidBrush(BaseColor), GP2)

'-- Text
    .DrawString("ON", Font, New SolidBrush(BaseColor), New
Rectangle(8, 7, W, H), NearSF)
EndIf
Case _Options.Style2  '-- Style 2
'-- Base
    GP = Helpers.RoundRec(Base, 6)
    Toggle = New Rectangle(4, 4, 36, H - 8)
    GP2 = Helpers.RoundRec(Toggle, 4)
        .FillPath(New SolidBrush(BaseColorRed), GP)
        .FillPath(New SolidBrush(ToggleColor), GP2)

'-- Lines
    .DrawLine(New Pen(BGColor), 18, 20, 18, 12)
    .DrawLine(New Pen(BGColor), 22, 20, 22, 12)
    .DrawLine(New Pen(BGColor), 26, 20, 26, 12)

'-- Text
    .DrawString("r", New Font("Marlett", 8), New
SolidBrush(TextColor), New Rectangle(19, 2, Width, Height), CenterSF)

If Checked Then
    GP = Helpers.RoundRec(Base, 6)
    Toggle = New Rectangle(CInt(W \ 2) - 2, 4, 36, H - 8)
    GP2 = Helpers.RoundRec(Toggle, 4)
        .FillPath(New SolidBrush(BaseColor), GP)
        .FillPath(New SolidBrush(ToggleColor), GP2)

'-- Lines
    .DrawLine(New Pen(BGColor), CInt(W \ 2) + 12, 20, CInt(W \
2) + 12, 12)
    .DrawLine(New Pen(BGColor), CInt(W \ 2) + 16, 20, CInt(W \
2) + 16, 12)
    .DrawLine(New Pen(BGColor), CInt(W \ 2) + 20, 20, CInt(W \
2) + 20, 12)

'-- Text
    .DrawString("ü", New Font("Wingdings", 14), New
SolidBrush(TextColor), New Rectangle(8, 7, Width, Height), NearSF)
EndIf
Case _Options.Style3  '-- Style 3
'-- Base
    GP = Helpers.RoundRec(Base, 16)
    Toggle = New Rectangle(W - 28, 4, 22, H - 8)
    GP2.AddEllipse(Toggle)
        .FillPath(New SolidBrush(ToggleColor), GP)
        .FillPath(New SolidBrush(BaseColorRed), GP2)

'-- Text
    .DrawString("OFF", Font, New SolidBrush(BaseColorRed), New
Rectangle(-12, 2, W, H), CenterSF)

If Checked Then

```

```

'-- Base
    GP = Helpers.RoundRec(Base, 16)
    Toggle = New Rectangle(6, 4, 22, H - 8)
    GP2.Reset()
    GP2.AddEllipse(Toggle)
    .FillPath(New SolidBrush(ToggleColor), GP)
    .FillPath(New SolidBrush(BaseColor), GP2)

'-- Text
    .DrawString("ON", Font, New SolidBrush(BaseColor), New
    Rectangle(12, 2, W, H), CenterSF)
EndIf
Case _Options.Style4
'-- TODO: New Styles
If Checked Then
'--
EndIf
Case _Options.Style5
'-- TODO: New Styles
If Checked Then
'--
EndIf
EndSelect

EndWith

 MyBase.OnPaint(e)
    G.Dispose()
    e.Graphics.InterpolationMode = 7
    e.Graphics.DrawImageUnscaled(B, 0, 0)
    B.Dispose()
EndSub
EndClass

<DefaultEvent("CheckedChanged")>ClassRadioButton : InheritsControl

#Region" Variables"

Private State AsMouseState = MouseState.None
Private W, H AsInteger
Private O As_Options
Private _Checked AsBoolean

#EndRegion

#Region" Properties"
Property Checked() AsBoolean
Get
Return _Checked
EndGet
Set(ByVal value AsBoolean)
    _Checked = value
    InvalidateControls()
RaiseEvent CheckedChanged(Me)
    Invalidate()
EndSet
EndProperty
Event CheckedChanged(ByVal sender AsObject)
ProtectedOverridesSub OnClick(ByVal e AsEventArgs)
IfNot _Checked Then Checked = True
 MyBase.OnClick(e)
EndSub

```

```

PrivateSub InvalidateControls()
IfNot IsHandleCreated OrElseNot _Checked ThenReturn
ForEach C As Control In Parent.Controls
If C IsNotMeAndAlsoTypeOf C Is RadioButton Then
DirectCast(C, RadioButton).Checked = False
Invalidate()
EndIf
Next
EndSub
ProtectedOverridesSub OnCreateControl()
 MyBase.OnCreateControl()
 InvalidateControls()
EndSub

<Flags()> _
Enum _Options
    Style1
    Style2
EndEnum

<Category("Options")> _
PublicProperty Options As _Options
Get
Return 0
EndGet
Set(ByVal value As _Options)
    0 = value
EndSet
EndProperty

ProtectedOverridesSub OnResize(ByVal e As EventArgs)
 MyBase.OnResize(e)
 Height = 22
EndSub

#Region" Mouse States"

ProtectedOverridesSub OnMouseDown(ByVal e As MouseEventArgs)
 MyBase.OnMouseDown(e)
 State = MouseState.Down : Invalidate()
EndSub
ProtectedOverridesSub OnMouseUp(ByVal e As MouseEventArgs)
 MyBase.OnMouseUp(e)
 State = MouseState.Over : Invalidate()
EndSub
ProtectedOverridesSub OnMouseEnter(ByVal e As EventArgs)
 MyBase.OnMouseEnter(e)
 State = MouseState.Over : Invalidate()
EndSub
ProtectedOverridesSub OnMouseLeave(ByVal e As EventArgs)
 MyBase.OnMouseLeave(e)
 State = MouseState.None : Invalidate()
EndSub

#EndRegion
#EndRegion

#Region" Colors"

Private _BaseColor AsColor = Color.FromArgb(45, 47, 49)
Private _BorderColor AsColor = _FlatColor
Private _TextColor AsColor = Color.FromArgb(243, 243, 243)

```

```

#EndRegion

SubNew()
    SetStyle(ControlStyles.AllPaintingInWmPaint Or ControlStyles.UserPaint Or
    -
        ControlStyles.ResizeRedraw Or
    ControlStyles.OptimizedDoubleBuffer, True)
    DoubleBuffered = True
    Cursor = Cursors.Hand
    Size = New Size(100, 22)
    BackColor = Color.FromArgb(60, 70, 73)
    Font = New Font("Segoe UI", 10)
EndSub
ProtectedOverridesSub OnPaint(ByVal e AsPaintEventArgs)
    B = New Bitmap(Width, Height) : G = Graphics.FromImage(B)
    W = Width - 1 : H = Height - 1

    Dim Base AsNew Rectangle(0, 2, Height - 5, Height - 5), Dot AsNew Rectangle(4, 6,
    H - 12, H - 12)

    With G
        .SmoothingMode = 2
        .TextRenderingHint = 5
        .Clear(BackColor)

        SelectCase 0
        Case _Options.Style1
        '-- Base
            .FillEllipse(New SolidBrush(_BaseColor), Base)

        SelectCase State
        Case MouseState.Over
            .DrawEllipse(New Pen(_BorderColor), Base)
        Case MouseState.Down
            .DrawEllipse(New Pen(_BorderColor), Base)
        EndSelect

        '-- If Checked
        If Checked Then
            .FillEllipse(New SolidBrush(_BorderColor), Dot)
        EndIf
        Case _Options.Style2
        '-- Base
            .FillEllipse(New SolidBrush(_BaseColor), Base)

        SelectCase State
        Case MouseState.Over
        '-- Base
            .DrawEllipse(New Pen(_BorderColor), Base)
            .FillEllipse(New SolidBrush(Color.FromArgb(118, 213,
        170)), Base)
        Case MouseState.Down
        '-- Base
            .DrawEllipse(New Pen(_BorderColor), Base)
            .FillEllipse(New SolidBrush(Color.FromArgb(118, 213,
        170)), Base)
        EndSelect

        '-- If Checked
        If Checked Then
        '-- Base

```

```

        .FillEllipse(New SolidBrush(_BorderColor), Dot)
EndIf
EndSelect

        .DrawString(Text, Font, New SolidBrush(_TextColor), New Rectangle(20,
2, W, H), NearSF)
EndWith

 MyBase.OnPaint(e)
    G.Dispose()
    e.Graphics.InterpolationMode = 7
    e.Graphics.DrawImageUnscaled(B, 0, 0)
    B.Dispose()
EndSub
EndClass

<DefaultEvent("CheckedChanged")>ClassFlatCheckBox : InheritsControl

#Region" Variables"

Private W, H AsInteger
Private State AsMouseState = MouseState.None
Private O As_Options
Private _Checked AsBoolean

#EndRegion

#Region" Properties"
ProtectedOverridesSub OnTextChanged(ByVal e As System.EventArgs)
MyBase.OnTextChanged(e)
    Invalidate()
EndSub

Property Checked() AsBoolean
Get
Return _Checked
EndGet
Set(ByVal value AsBoolean)
    _Checked = value
    Invalidate()
EndSet
EndProperty

Event CheckedChanged(ByVal sender AsObject)
ProtectedOverridesSub OnClick(ByVal e As System.EventArgs)
    _Checked = Not _Checked
    RaiseEvent CheckedChanged(Me)
MyBase.OnClick(e)
EndSub

<Flags()> _
Enum_Options
    Style1
    Style2
EndEnum

<Category("Options")> _
PublicProperty Options As_Options
Get
Return O
EndGet
Set(ByVal value As _Options)

```

```

    0 = value
EndSet
EndProperty

ProtectedOverridesSub OnResize(ByVal e AsEventArgs)
MyBase.OnResize(e)
    Height = 22
EndSub

#Region" Colors"

<Category("Colors")> _
PublicProperty BaseColor AsColor
Get
Return _BaseColor
EndGet
Set(ByVal value As Color)
    _BaseColor = value
EndSet
EndProperty

<Category("Colors")> _
PublicProperty BorderColor AsColor
Get
Return _BorderColor
EndGet
Set(ByVal value As Color)
    _BorderColor = value
EndSet
EndProperty

#EndRegion

#Region" Mouse States"

ProtectedOverridesSub OnMouseDown(ByVal e AsMouseEventArgs)
 MyBase.OnMouseDown(e)
    State = MouseState.Down : Invalidate()
EndSub
ProtectedOverridesSub OnMouseUp(ByVal e AsMouseEventArgs)
 MyBase.OnMouseUp(e)
    State = MouseState.Over : Invalidate()
EndSub
ProtectedOverridesSub OnMouseEnter(ByVal e AsEventArgs)
 MyBase.OnMouseEnter(e)
    State = MouseState.Over : Invalidate()
EndSub
ProtectedOverridesSub OnMouseLeave(ByVal e AsEventArgs)
 MyBase.OnMouseLeave(e)
    State = MouseState.None : Invalidate()
EndSub

#EndRegion

#EndRegion

#Region" Colors"

Private _BaseColor AsColor = Color.FromArgb(45, 47, 49)
Private _BorderColor AsColor = _FlatColor
Private _TextColor AsColor = Color.FromArgb(243, 243, 243)

```

```

#EndRegion

SubNew()
    SetStyle(ControlStyles.AllPaintingInWmPaint Or ControlStyles.UserPaint Or
    -
        ControlStyles.ResizeRedraw Or
    ControlStyles.OptimizedDoubleBuffer, True)
    DoubleBuffered = True
    BackColor = Color.FromArgb(60, 70, 73)
    Cursor = Cursors.Hand
    Font = New Font("Segoe UI", 10)
    Size = New Size(112, 22)
EndSub

ProtectedOverridesSub OnPaint(ByVal e As PaintEventArgs)
    B = New Bitmap(Width, Height) : G = Graphics.FromImage(B)
    W = Width - 1 : H = Height - 1

    Dim Base AsNew Rectangle(0, 2, Height - 5, Height - 5)

    With G
        .SmoothingMode = 2
        .TextRenderingHint = 5
        .Clear(BackColor)
    SelectCase 0
    Case _Options.Style1 '-- Style 1
    '-- Base
        .FillRectangle(New SolidBrush(_BaseColor), Base)

    SelectCase State
    Case MouseState.Over
    '-- Base
        .DrawRectangle(New Pen(_BorderColor), Base)
    Case MouseState.Down
    '-- Base
        .DrawRectangle(New Pen(_BorderColor), Base)
    EndSelect

    '-- If Checked
    If Checked Then
        .DrawString("ü", New Font("Wingdings", 18), New
SolidBrush(_BorderColor), New Rectangle(5, 7, H - 9, H - 9), CenterSF)
    EndIf

    '-- If Enabled
    IfMe.Enabled = FalseThen
        .FillRectangle(New SolidBrush(Color.FromArgb(54, 58, 61)), Base)
        .DrawString(Text, Font, New SolidBrush(Color.FromArgb(140,
142, 143)), New Rectangle(20, 2, W, H), NearSF)
    EndIf

    '-- Text
        .DrawString(Text, Font, New SolidBrush(_TextColor), New
Rectangle(20, 2, W, H), NearSF)
    Case _Options.Style2 '-- Style 2
    '-- Base
        .FillRectangle(New SolidBrush(_BaseColor), Base)

    SelectCase State
    Case MouseState.Over
    '-- Base

```

```

        .DrawRectangle(New Pen(_BorderColor), Base)
        .FillRectangle(New SolidBrush(Color.FromArgb(118, 213,
170)), Base)
Case MouseState.Down
'-- Base
        .DrawRectangle(New Pen(_BorderColor), Base)
        .FillRectangle(New SolidBrush(Color.FromArgb(118, 213,
170)), Base)
EndSelect

'-- If Checked
If Checked Then
        .DrawString("ü", New Font("Wingdings", 18), New
SolidBrush(_BorderColor), New Rectangle(5, 7, H - 9, H - 9), CenterSF)
EndIf

'-- If Enabled
IfMe.Enabled = FalseThen
        .FillRectangle(New SolidBrush(Color.FromArgb(54, 58, 61)),
Base)
        .DrawString(Text, Font, New SolidBrush(Color.FromArgb(48,
119, 91)), New Rectangle(20, 2, W, H), NearSF)
EndIf

'-- Text
        .DrawString(Text, Font, New SolidBrush(_TextColor), New
Rectangle(20, 2, W, H), NearSF)
EndSelect
EndWith

 MyBase.OnPaint(e)
    G.Dispose()
    e.Graphics.InterpolationMode = 7
    e.Graphics.DrawImageUnscaled(B, 0, 0)
    B.Dispose()
EndSub
EndClass

<DefaultEvent("TextChanged")>ClassFlatTextBox : InheritsControl

#Region" Variables"

Private W, H AsInteger
Private State AsMouseState = MouseState.None
Private WithEvents TB As Windows.Forms.TextBox

#EndRegion

#Region" Properties"

#Region" TextBox Properties"

Private _TextAlign AsHorizontalAlignment = HorizontalAlignment.Left
<Category("Options")>
Property TextAlign() AsHorizontalAlignment
Get
Return _TextAlign
EndGet
Set(ByVal value As HorizontalAlignment)
    _TextAlign = value
If TB IsNot NothingThen
    TB.TextAlign = value
EndIf
EndRegion
EndClass

```

```

EndIf
EndSet
EndProperty
Private _MaxLength AsInteger = 32767
<Category("Options")> -
Property MaxLength() AsInteger
Get
Return _MaxLength
EndGet
Set(ByName value AsInteger)
    _MaxLength = value
If TB IsNotNothingThen
    TB.MaxLength = value
EndIf
EndSet
EndProperty
Private _ReadOnly AsBoolean
<Category("Options")> -
Property [ReadOnly]() AsBoolean
Get
Return _ReadOnly
EndGet
Set(ByName value AsBoolean)
    _ReadOnly = value
If TB IsNotNothingThen
    TB.ReadOnly = value
EndIf
EndSet
EndProperty
Private _UseSystemPasswordChar AsBoolean
<Category("Options")> -
Property UseSystemPasswordChar() AsBoolean
Get
Return _UseSystemPasswordChar
EndGet
Set(ByName value AsBoolean)
    _UseSystemPasswordChar = value
If TB IsNotNothingThen
    TB.UseSystemPasswordChar = value
EndIf
EndSet
EndProperty
Private _Multiline AsBoolean
<Category("Options")> -
Property Multiline() AsBoolean
Get
Return _Multiline
EndGet
Set(ByName value AsBoolean)
    _Multiline = value
If TB IsNotNothingThen
    TB.Multiline = value

If value Then
    TB.Height = Height - 11
Else
    Height = TB.Height + 11
EndIf

EndIf
EndSet
EndProperty

```

```

<Category("Options")> _
OverridesProperty Text AsString
Get
Return MyBase.Text
EndGet
Set( ByVal value AsString)
MyBase.Text = value
If TB IsNot Nothing Then
    TB.Text = value
EndIf
EndSet
EndProperty
<Category("Options")> _
OverridesProperty Font AsFont
Get
Return MyBase.Font
EndGet
Set( ByVal value As Font)
MyBase.Font = value
If TB IsNot Nothing Then
    TB.Font = value
    TB.Location = New Point(3, 5)
    TB.Width = Width - 6

IfNot _Multiline Then
    Height = TB.Height + 11
EndIf
EndIf
EndSet
EndProperty

Protected Overrides Sub OnCreateControl()
 MyBase.OnCreateControl()
 IfNot Controls.Contains(TB) Then
     Controls.Add(TB)
 EndIf
EndSub
Private Sub OnBaseTextChanged(ByVal s As Object, ByVal e As EventArgs)
    Text = TB.Text
EndSub
Private Sub OnBaseKeyDown(ByVal s As Object, ByVal e As KeyEventArgs)
    If e.Control AndAlso e.KeyCode = Keys.A Then
        TB.SelectAll()
        e.SuppressKeyPress = True
    EndIf
    If e.Control AndAlso e.KeyCode = Keys.C Then
        TB.Copy()
        e.SuppressKeyPress = True
    EndIf
EndSub
Protected Overrides Sub OnResize(ByVal e As EventArgs)
    TB.Location = New Point(5, 5)
    TB.Width = Width - 10

    If _Multiline Then
        TB.Height = Height - 11
    Else
        Height = TB.Height + 11
    EndIf
EndSub
MyBase.OnResize(e)
EndSub

```

```

#EndRegion

#Region" Colors"

<Category("Colors")>
PublicProperty TextColor AsColor
Get
Return _TextColor
EndGet
Set(ByVal value As Color)
    _TextColor = value
EndSet
EndProperty

PublicOverridesProperty ForeColor() AsColor
Get
Return _TextColor
EndGet
Set(ByVal value As Color)
    _TextColor = value
EndSet
EndProperty

#EndRegion

#Region" Mouse States"

ProtectedOverridesSub OnMouseDown(ByVal e As MouseEventArgs)
 MyBase.OnMouseDown(e)
    State = MouseState.Down : Invalidate()
EndSub
ProtectedOverridesSub OnMouseUp(ByVal e As MouseEventArgs)
 MyBase.OnMouseUp(e)
    State = MouseState.Over : TB.Focus() : Invalidate()
EndSub
ProtectedOverridesSub OnMouseEnter(ByVal e As EventArgs)
 MyBase.OnMouseEnter(e)
    State = MouseState.Over : TB.Focus() : Invalidate()
EndSub
ProtectedOverridesSub OnMouseLeave(ByVal e As EventArgs)
 MyBase.OnMouseLeave(e)
    State = MouseState.None : Invalidate()
EndSub

#EndRegion

#EndRegion

#Region" Colors"

Private _BaseColor AsColor = Color.FromArgb(45, 47, 49)
Private _TextColor AsColor = Color.FromArgb(192, 192, 192)
Private _BorderColor AsColor = _FlatColor

#EndRegion

SubNew()
    SetStyle(ControlStyles.AllPaintingInWmPaint Or ControlStyles.UserPaint Or
    - ControlStyles.ResizeRedraw Or ControlStyles.OptimizedDoubleBuffer
    Or _

```

```

        ControlStyles.SupportsTransparentBackColor, True)
DoubleBuffered = True

BackColor = Color.Transparent

TB = New Windows.Forms.TextBox
TB.Font = New Font("Segoe UI", 10)
TB.Text = Text
TB.BackColor = _BaseColor
TB.ForeColor = _TextColor
TB.MaxLength = _MaxLength
TB.Multiline = _Multiline
TB.ReadOnly = _ReadOnly
TB.UseSystemPasswordChar = _UseSystemPasswordChar
TB.BorderStyle = BorderStyle.None
TB.Location = New Point(5, 5)
TB.Width = Width - 10

TB.Cursor = Cursors.IBeam

If _Multiline Then
    TB.Height = Height - 11
Else
    Height = TB.Height + 11
EndIf

AddHandler TB.TextChanged, AddressOf OnBaseTextChanged
AddHandler TB.KeyDown, AddressOf OnBaseKeyDown
EndSub

ProtectedOverridesSub OnPaint(ByVal e AsPaintEventArgs)
    B = New Bitmap(Width, Height) : G = Graphics.FromImage(B)
    W = Width - 1 : H = Height - 1

Dim Base AsNew Rectangle(0, 0, W, H)

With G
    .SmoothingMode = 2
    .PixelOffsetMode = 2
    .TextRenderingHint = 5
    .Clear(BackColor)

'-- Colors
    TB.BackColor = _BaseColor
    TB.ForeColor = _TextColor

'-- Base
    .FillRectangle(New SolidBrush(_BaseColor), Base)
EndWith

 MyBase.OnPaint(e)
    G.Dispose()
    e.Graphics.InterpolationMode = 7
    e.Graphics.DrawImageUnscaled(B, 0, 0)
    B.Dispose()
EndSub

EndClass

ClassFlatTabControl : InheritsTabControl

#Region" Variables"

```

```

Private W, H AsInteger
#EndRegion

#Region" Properties"
ProtectedOverridesSub CreateHandle()
 MyBase.CreateHandle()
 Alignment = TabAlignment.Top
EndSub

#Region" Colors"

<Category("Colors")> _
PublicProperty BaseColor AsColor
Get
Return _BaseColor
EndGet
Set(ByVal value As Color)
    _BaseColor = value
EndSet
EndProperty

<Category("Colors")> _
PublicProperty ActiveColor AsColor
Get
Return _ActiveColor
EndGet
Set(ByVal value As Color)
    _ActiveColor = value
EndSet
EndProperty

#EndRegion

#EndRegion

#Region" Colors"

Private BGColor AsColor = Color.FromArgb(60, 70, 73)
Private _BaseColor AsColor = Color.FromArgb(45, 47, 49)
Private _ActiveColor AsColor = _FlatColor

#EndRegion

SubNew()
    SetStyle(ControlStyles.AllPaintingInWmPaint Or ControlStyles.UserPaint Or
    - ControlStyles.ResizeRedraw Or
ControlStyles.OptimizedDoubleBuffer, True)
    DoubleBuffered = True
    BackColor = Color.FromArgb(60, 70, 73)

    Font = New Font("Segoe UI", 10)
   SizeMode = TabSizeMode.Fixed
    ItemSize = New Size(120, 40)
EndSub

ProtectedOverridesSub OnPaint(ByVal e AsPaintEventArgs)
    B = New Bitmap(Width, Height) : G = Graphics.FromImage(B)
    W = Width - 1 : H = Height - 1

```

```

With G
    .SmoothingMode = 2
    .PixelOffsetMode = 2
    .TextRenderingHint = 5
    .Clear(_BaseColor)

Try : SelectedTab.BackColor = BGColor : Catch : EndTry

For i = 0 To TabCount - 1
Dim Base AsNew Rectangle(New Point(GetTabRect(i).Location.X + 2,
GetTabRect(i).Location.Y), New Size(GetTabRect(i).Width, GetTabRect(i).Height))
Dim BaseSize AsNew Rectangle(Base.Location, New Size(Base.Width, Base.Height))

If i = SelectedIndex Then
'-- Base
    .FillRectangle(New SolidBrush(_BaseColor), BaseSize)

'-- Gradiant
'.fill
    .FillRectangle(New SolidBrush(_ActiveColor), BaseSize)

'-- ImageList
If ImageList IsNotNothingThen
Try
If ImageList.Images(TabPages(i).ImageIndex) IsNotNothingThen
'-- Image

    .DrawImage(ImageList.Images(TabPages(i).ImageIndex), New Point(BaseSize.Location.X
+ 8, BaseSize.Location.Y + 6))
'-- Text
        .DrawString("      "& TabPages(i).Text, Font,
Brushes.White, BaseSize, CenterSF)
Else
'-- Text
        .DrawString(TabPages(i).Text, Font, Brushes.White,
BaseSize, CenterSF)
EndIf
Catch ex As Exception
ThrowNew Exception(ex.Message)
EndTry
Else
'-- Text
        .DrawString(TabPages(i).Text, Font, Brushes.White,
BaseSize, CenterSF)
EndIf
Else
'-- Base
    .FillRectangle(New SolidBrush(_BaseColor), BaseSize)

'-- ImageList
If ImageList IsNotNothingThen
Try
If ImageList.Images(TabPages(i).ImageIndex) IsNotNothingThen
'-- Image

    .DrawImage(ImageList.Images(TabPages(i).ImageIndex), New Point(BaseSize.Location.X
+ 8, BaseSize.Location.Y + 6))
'-- Text
        .DrawString("      "& TabPages(i).Text, Font, New
SolidBrush(Color.White), BaseSize, New StringFormat With {.LineAlignment =
StringAlignment.Center, .Alignment = StringAlignment.Center})

```

```

Else
'-- Text
    .DrawString(TabPages(i).Text, Font, New
SolidBrush(Color.White), BaseSize, New StringFormat With {.LineAlignment =
StringAlignment.Center, .Alignment = StringAlignment.Center})
EndIf
Catch ex As Exception
ThrowNew Exception(ex.Message)
EndTry
Else
'-- Text
    .DrawString(TabPages(i).Text, Font, New
SolidBrush(Color.White), BaseSize, New StringFormat With {.LineAlignment =
StringAlignment.Center, .Alignment = StringAlignment.Center})
EndIf
EndIf
Next
EndWith

 MyBase.OnPaint(e)
    G.Dispose()
    e.Graphics.InterpolationMode = 7
    e.Graphics.DrawImageUnscaled(B, 0, 0)
    B.Dispose()
EndSub
EndClass

ClassFlatAlertBox : InheritsControl

''' <summary>
''' How to use: FlatAlertBox.ShowControl(Kind, String, Interval)
''' </summary>
''' <remarks></remarks>

#Region" Variables"

Private W, H AsInteger
Private K As_Kind
Private _TextAsString
Private State AsMouseState = MouseState.None
Private X AsInteger
Private WithEvents T AsTimer

#EndRegion

#Region" Properties"

<Flags()> _
Enum_Kind
    [Success]
    [Error]
    [Info]
EndEnum

#Region" Options"

<Category("Options")> _
PublicProperty kind As_Kind
Get
Return K
EndGet
Set(ByVal value As _Kind)

```

```

        K = value
EndSet
EndProperty

<Category("Options")> _
OverridesProperty Text AsString
Get
Return MyBase.Text
EndGet
Set(ByVal value AsString)
MyBase.Text = value
If _Text IsNot NothingThen
    _Text = value
EndIf
EndSet
EndProperty

<Category("Options")> _
ShadowsProperty Visible AsBoolean
Get
Return MyBase.Visible = False
EndGet
Set(ByVal value AsBoolean)
MyBase.Visible = value
EndSet
EndProperty

#EndRegion

ProtectedOverridesSub OnTextChanged(ByVal e AsEventArgs)
MyBase.OnTextChanged(e) : Invalidate()
EndSub

ProtectedOverridesSub OnResize(ByVal e AsEventArgs)
MyBase.OnResize(e)
    Height = 42
EndSub

PublicSub ShowControl(ByVal Kind As_Kind, ByVal Str AsString, ByVal Interval
AsInteger)
    K = Kind
    Text = Str
Me.Visible = True
    T = New Timer
    T.Interval = Interval
    T.Enabled = True
EndSub

PrivateSub T_Tick(ByVal sender AsObject, ByVal e AsEventArgs) Handles T.Tick
Me.Visible = False
    T.Enabled = False
    T.Dispose()
EndSub

#Region" Mouse States"

ProtectedOverridesSub OnMouseDown(ByVal e AsMouseEventArgs)
MyBase.OnMouseDown(e)
    State = MouseState.Down : Invalidate()
EndSub
ProtectedOverridesSub OnMouseUp(ByVal e AsMouseEventArgs)
MyBase.OnMouseUp(e)

```

```

        State = MouseState.Over : Invalidate()
EndSub
ProtectedOverridesSub OnMouseEnter(ByVal e AsEventArgs)
 MyBase.OnMouseEnter(e)
        State = MouseState.Over : Invalidate()
EndSub
ProtectedOverridesSub OnMouseLeave(ByVal e AsEventArgs)
 MyBase.OnMouseLeave(e)
        State = MouseState.None : Invalidate()
EndSub

ProtectedOverridesSub OnMouseMove(ByVal e AsMouseEventArgs)
 MyBase.OnMouseMove(e)
        X = e.X : Invalidate()
EndSub

ProtectedOverridesSub OnClick(ByVal e AsEventArgs)
 MyBase.OnClick(e)
 Me.Visible = False
EndSub

#EndRegion

#EndRegion

#Region " Colors"
    Private SuccessColor AsColor = Color.FromArgb(60, 85, 79)
    Private SuccessText AsColor = Color.FromArgb(35, 169, 110)
    Private ErrorColor AsColor = Color.FromArgb(87, 71, 71)
    Private ErrorText AsColor = Color.FromArgb(254, 142, 122)
    Private InfoColor AsColor = Color.FromArgb(70, 91, 94)
    Private InfoText AsColor = Color.FromArgb(97, 185, 186)

#EndRegion

SubNew()
    SetStyle(ControlStyles.AllPaintingInWmPaint Or ControlStyles.UserPaint Or
    -
        ControlStyles.ResizeRedraw Or
    ControlStyles.OptimizedDoubleBuffer, True)
        DoubleBuffered = True
        BackColor = Color.FromArgb(60, 70, 73)
        Size = New Size(576, 42)
        Location = New Point(10, 61)
        Font = New Font("Segoe UI", 10)
        Cursor = Cursors.Hand
EndSub

ProtectedOverridesSub OnPaint(ByVal e AsPaintEventArgs)
    B = New Bitmap(Width, Height) : G = Graphics.FromImage(B)
    W = Width - 1 : H = Height - 1

    Dim Base AsNew Rectangle(0, 0, W, H)

    With G
        .SmoothingMode = 2
        .TextRenderingHint = 5
        .Clear(BackColor)

    SelectCase K
    Case _Kind.Success

```

```

'-- Base
    .FillRectangle(New SolidBrush(SuccessColor), Base)

'-- Ellipse
    .FillEllipse(New SolidBrush(SuccessText), New Rectangle(8, 9,
24, 24))
    .FillEllipse(New SolidBrush(SuccessColor), New Rectangle(10,
11, 20, 20))

'-- Checked Sign
    .DrawString("ü", New Font("Wingdings", 22), New
SolidBrush(SuccessText), New Rectangle(7, 7, W, H), NearSF)
    .DrawString(Text, Font, New SolidBrush(SuccessText), New
Rectangle(48, 12, W, H), NearSF)

'-- X button
    .FillEllipse(New SolidBrush(Color.FromArgb(35, Color.Black)), New
Rectangle(W - 30, H - 29, 17, 17))
    .DrawString("r", New Font("Marlett", 8), New
SolidBrush(SuccessColor), New Rectangle(W - 28, 16, W, H), NearSF)

SelectCase State '-- Mouse Over
Case MouseState.Over
    .DrawString("r", New Font("Marlett", 8), New
SolidBrush(Color.FromArgb(25, Color.White)), New Rectangle(W - 28, 16, W, H),
NearSF)
EndSelect

Case _Kind.Error
'-- Base
    .FillRectangle(New SolidBrush(ErrorColor), Base)

'-- Ellipse
    .FillEllipse(New SolidBrush(ErrorText), New Rectangle(8, 9,
24, 24))
    .FillEllipse(New SolidBrush(ErrorColor), New Rectangle(10, 11,
20, 20))

'-- X Sign
    .DrawString("r", New Font("Marlett", 16), New
SolidBrush(ErrorText), New Rectangle(6, 11, W, H), NearSF)
    .DrawString(Text, Font, New SolidBrush(ErrorText), New
Rectangle(48, 12, W, H), NearSF)

'-- X button
    .FillEllipse(New SolidBrush(Color.FromArgb(35, Color.Black)), New
Rectangle(W - 32, H - 29, 17, 17))
    .DrawString("r", New Font("Marlett", 8), New
SolidBrush(ErrorColor), New Rectangle(W - 30, 17, W, H), NearSF)

SelectCase State
Case MouseState.Over '-- Mouse Over
    .DrawString("r", New Font("Marlett", 8), New
SolidBrush(Color.FromArgb(25, Color.White)), New Rectangle(W - 30, 15, W, H),
NearSF)
EndSelect

Case _Kind.Info
'-- Base
    .FillRectangle(New SolidBrush(InfoColor), Base)

'-- Ellipse

```

```

        .FillEllipse(New SolidBrush(InfoText), New Rectangle(8, 9, 24,
24))
                .FillEllipse(New SolidBrush(InfoColor), New Rectangle(10, 11,
20, 20))

'-- Info Sign
        .DrawString("i", New Font("Segoe UI", 20, FontStyle.Bold), New
SolidBrush(InfoText), New Rectangle(12, -4, W, H), NearSF)
        .DrawString(Text, Font, New SolidBrush(InfoText), New
Rectangle(48, 12, W, H), NearSF)

'-- X button
        .FillEllipse(New SolidBrush(Color.FromArgb(35, Color.Black)), New
Rectangle(W - 32, H - 29, 17, 17))
        .DrawString("r", New Font("Marlett", 8), New
SolidBrush(InfoColor), New Rectangle(W - 30, 17, W, H), NearSF)

SelectCase State
Case MouseState.Over ' -- Mouse Over
        .DrawString("r", New Font("Marlett", 8), New
SolidBrush(Color.FromArgb(25, Color.White)), New Rectangle(W - 30, 17, W, H),
NearSF)
EndSelect
EndSelect

EndWith

 MyBase.OnPaint(e)
        G.Dispose()
        e.Graphics.InterpolationMode = 7
        e.Graphics.DrawImageUnscaled(B, 0, 0)
        B.Dispose()
EndSub

EndClass

ClassFlatProgressBar : InheritsControl

#Region" Variables"

Private W, H AsInteger
Private _Value AsInteger = 0
Private _Maximum AsInteger = 100

#EndRegion

#Region" Properties"

#Region" Control"

<Category("Control")>
PublicProperty Maximum() AsInteger
Get
Return _Maximum
EndGet
Set(ByVal V AsInteger)
SelectCase V
CaseIs < _Value
        _Value = V
EndSelect
        _Maximum = V
        Invalidate()
EndProperty
EndRegion
EndClass

```

```

EndSet
EndProperty

<Category("Control")>
PublicProperty Value() AsInteger
Get
SelectCase _Value
Case 0
Return 0
Invalidate()
Case Else
Return _Value
Invalidate()
EndSelect
EndGet
Set(ByVal V AsInteger)
SelectCase V
CaseIs > _Maximum
V = _Maximum
Invalidate()
EndSelect
_Value = V
Invalidate()
EndSet
EndProperty

#EndRegion

#Region" Colors"

<Category("Colors")>
PublicProperty ProgressColor AsColor
Get
Return _ProgressColor
EndGet
Set(ByVal value As Color)
_ProgressColor = value
EndSet
EndProperty

<Category("Colors")>
PublicProperty DarkerProgress AsColor
Get
Return _DarkerProgress
EndGet
Set(ByVal value As Color)
_DarkerProgress = value
EndSet
EndProperty

#EndRegion

ProtectedOverridesSub OnResize(ByVal e AsEventArgs)
 MyBase.OnResize(e)
 Height = 42
EndSub

ProtectedOverridesSub CreateHandle()
 MyBase.CreateHandle()
 Height = 42
EndSub

```

```

PublicSub Increment(ByVal Amount AsInteger)
    Value += Amount
EndSub

#EndRegion

#Region" Colors"

Private _BaseColor AsColor = Color.FromArgb(45, 47, 49)
Private _ProgressColor AsColor = _FlatColor
Private _DarkerProgress AsColor = Color.FromArgb(23, 148, 92)

#EndRegion

SubNew()
    SetStyle(ControlStyles.AllPaintingInWmPaint Or ControlStyles.UserPaint Or
    -
        ControlStyles.ResizeRedraw Or
    ControlStyles.OptimizedDoubleBuffer, True)
    DoubleBuffered = True
    BackColor = Color.FromArgb(60, 70, 73)
    Height = 42
EndSub

ProtectedOverridesSub OnPaint(ByVal e AsPaintEventArgs)
    B = New Bitmap(Width, Height) : G = Graphics.FromImage(B)
    W = Width - 1 : H = Height - 1

    Dim Base AsNew Rectangle(0, 24, W, H)
    Dim GP, GP2, GP3 AsNew GraphicsPath

    With G
        .SmoothingMode = 2
        .PixelOffsetMode = 2
        .TextRenderingHint = 5
        .Clear(BackColor)

        '-- Progress Value
        Dim iValue AsInteger = CInt(_Value / _Maximum * Width)

        SelectCase Value
        Case 0
            '-- Base
            .FillRectangle(New SolidBrush(_BaseColor), Base)
            '--Progress
            .FillRectangle(New SolidBrush(_ProgressColor), New
            Rectangle(0, 24, iValue - 1, H - 1))
        Case 100
            '-- Base
            .FillRectangle(New SolidBrush(_BaseColor), Base)
            '--Progress
            .FillRectangle(New SolidBrush(_ProgressColor), New
            Rectangle(0, 24, iValue - 1, H - 1))
        Case Else
            '-- Base
            .FillRectangle(New SolidBrush(_BaseColor), Base)
            '--Progress
            GP.AddRectangle(New Rectangle(0, 24, iValue - 1, H - 1))
            FillPath(New SolidBrush(_ProgressColor), GP)
        EndSelect
    EndWith
EndSub

```

```

Dim HB AsNew HatchBrush(HatchStyle.Plaid, _DarkerProgress, _ProgressColor)
    .FillRectangle(HB, New Rectangle(0, 24, iValue - 1, H - 1))

'-- Balloon
Dim Balloon AsNew Rectangle(iValue - 18, 0, 34, 16)
    GP2 = Helpers.RoundRec(Balloon, 4)
    .FillPath(New SolidBrush(_BaseColor), GP2)

'-- Arrow
    GP3 = Helpers.DrawArrow(iValue - 9, 16, True)
    .FillPath(New SolidBrush(_BaseColor), GP3)

'-- Value > You can add "%" > value & "%"
    .DrawString(Value, New Font("Segoe UI", 10), New
SolidBrush(_ProgressColor), New Rectangle(iValue - 11, -2, W, H), NearSF)
EndSelect
EndWith

 MyBase.OnPaint(e)
    G.Dispose()
    e.Graphics.InterpolationMode = 7
    e.Graphics.DrawImageUnscaled(B, 0, 0)
    B.Dispose()
EndSub

EndClass

ClassFlatComboBox : Inherits Windows.Forms.ComboBox

#Region" Variables"

Private W, H AsInteger
Private _startIndex AsInteger = 0
Private x, y AsInteger

#EndRegion

#Region" Properties"

#Region" Mouse States"

Private State AsMouseState = MouseState.None
ProtectedOverridesSub OnMouseDown(ByVal e AsMouseEventArgs)
    MyBase.OnMouseDown(e)
        State = MouseState.Down : Invalidate()
EndSub
ProtectedOverridesSub OnMouseUp(ByVal e AsMouseEventArgs)
    MyBase.OnMouseUp(e)
        State = MouseState.Over : Invalidate()
EndSub
ProtectedOverridesSub OnMouseEnter(ByVal e AsEventArgs)
    MyBase.OnMouseEnter(e)
        State = MouseState.Over : Invalidate()
EndSub
ProtectedOverridesSub OnMouseLeave(ByVal e AsEventArgs)
    MyBase.OnMouseLeave(e)
        State = MouseState.None : Invalidate()
EndSub

ProtectedOverridesSub OnMouseMove(ByVal e AsMouseEventArgs)
    MyBase.OnMouseMove(e)
        x = e.Location.X

```

```

        y = e.Location.Y
        Invalidate()
If e.X < Width - 41 Then Cursor = Cursors.IBeam Else Cursor = Cursors.Hand
EndSub

ProtectedOverridesSub OnDrawItem(ByVal e As DrawItemEventArgs)
 MyBase.OnDrawItem(e) : Invalidate()
If (e.State And DrawItemState.Selected) = DrawItemState.Selected Then
    Invalidate()
EndIf
EndSub

ProtectedOverridesSub OnClick(ByVal e As EventArgs)
 MyBase.OnClick(e) : Invalidate()
EndSub

#EndRegion

#Region " Colors"

<Category("Colors")> _
PublicProperty HoverColor AsColor
Get
Return _HoverColor
EndGet
Set(ByVal value As Color)
    _HoverColor = value
EndSet
EndProperty

#EndRegion

PrivateProperty StartIndex AsInteger
Get
Return _StartIndex
EndGet
Set(ByVal value AsInteger)
    _StartIndex = value
Try
MyBase.SelectedIndex = value
Catch
EndTry
    Invalidate()
EndSet
EndProperty

Sub DrawItem_(ByVal sender As System.Object, ByVal e As
System.Windows.Forms.DrawItemEventArgs) HandlesMe.DrawItem
If e.Index < 0 ThenExit Sub
    e.DrawBackground()
    e.DrawFocusRectangle()

    e.Graphics.SmoothingMode = 2
    e.Graphics.PixelOffsetMode = 2
    e.Graphics.TextRenderingHint = 5
    e.Graphics.InterpolationMode = 7

If (e.State And DrawItemState.Selected) = DrawItemState.Selected Then
    '-- Selected item
        e.Graphics.FillRectangle(New SolidBrush(_HoverColor), e.Bounds)
Else
    '-- Not Selected

```

```

        e.Graphics.FillRectangle(New SolidBrush(_BaseColor), e.Bounds)
EndIf

'-- Text
        e.Graphics.DrawString(MyBase.GetItemText(MyBase.Items(e.Index)), New
Font("Segoe UI", 8), _
Brushes.White, New Rectangle(e.Bounds.X + 2, e.Bounds.Y + 2,
e.Bounds.Width, e.Bounds.Height))

        e.Graphics.Dispose()
EndSub

ProtectedOverridesSub OnResize(ByVal e AsEventArgs)
MyBase.OnResize(e)
    Height = 18
EndSub

#EndRegion

#Region" Colors"

Private _BaseColor AsColor = Color.FromArgb(25, 27, 29)
Private _BColor AsColor = Color.FromArgb(45, 47, 49)
Private _HoverColor AsColor = Color.FromArgb(35, 168, 109)

#EndRegion

SubNew()
    SetStyle(ControlStyles.AllPaintingInWmPaint Or ControlStyles.UserPaint Or
-
ControlStyles.ResizeRedraw Or
ControlStyles.OptimizedDoubleBuffer, True)
    DoubleBuffered = True

    DrawMode = DrawMode.OwnerDrawFixed
    BackColor = Color.FromArgb(45, 45, 48)
    ForeColor = Color.White
    DropDownStyle = ComboBoxStyle.DropDownList
    Cursor = Cursors.Hand
    StartIndex = 0
    ItemHeight = 18
    Font = New Font("Segoe UI", 8, FontStyle.Regular)
EndSub

ProtectedOverridesSub OnPaint(ByVal e AsPaintEventArgs)
    B = New Bitmap(Width, Height) : G = Graphics.FromImage(B)
    W = Width : H = Height

    Dim Base AsNew Rectangle(0, 0, W, H)
    Dim Button AsNew Rectangle(CInt(W - 40), 0, W, H)
    Dim GP, GP2 AsNew GraphicsPath

    With G
        .Clear(Color.FromArgb(45, 45, 48))
        .SmoothingMode = 2
        .PixelOffsetMode = 2
        .TextRenderingHint = 5

'-- Base
        .FillRectangle(New SolidBrush(_BColor), Base)
    EndWith
EndSub

```

```

'-- Button
    GP.Reset()
    GP.AddRectangle(Button)
    .SetClip(GP)
    .FillRectangle(New SolidBrush(_BaseColor), Button)
    .ResetClip()

'-- Lines
    .DrawLine(Pens.White, W - 10, 6, W - 30, 6)
    .DrawLine(Pens.White, W - 10, 12, W - 30, 12)
    .DrawLine(Pens.White, W - 10, 18, W - 30, 18)

'-- Text
    .DrawString(Text, Font, Brushes.White, New Point(4, 6), NearSF)
EndWith

    G.Dispose()
    e.Graphics.InterpolationMode = 7
    e.Graphics.DrawImageUnscaled(B, 0, 0)
    B.Dispose()
EndSub
EndClass

ClassFlatStickyButton : InheritsControl

#Region" Variables"

Private W, H AsInteger
Private State AsMouseState = MouseState.None
Private _Rounded AsBoolean = False

#EndRegion

#Region" Properties"

#Region" MouseStates"

ProtectedOverridesSub OnMouseDown(ByVal e As MouseEventArgs)
 MyBase.OnMouseDown(e)
     State = MouseState.Down : Invalidate()
EndSub
ProtectedOverridesSub OnMouseUp(ByVal e As MouseEventArgs)
 MyBase.OnMouseUp(e)
     State = MouseState.Over : Invalidate()
EndSub
ProtectedOverridesSub OnMouseEnter(ByVal e As EventArgs)
 MyBase.OnMouseEnter(e)
     State = MouseState.Over : Invalidate()
EndSub
ProtectedOverridesSub OnMouseLeave(ByVal e As EventArgs)
 MyBase.OnMouseLeave(e)
     State = MouseState.None : Invalidate()
EndSub

#EndRegion

#Region" Function"

PrivateFunction GetConnectedSides() AsBoolean()
Dim Bool = NewBoolean(3) {False, False, False, False}

ForEach B As Control In Parent.Controls

```

```

IfTypeOf B Is FlatStickyButton Then
If B IsMeOrNot Rect.IntersectsWith(Rect) ThenContinue For
Dim A = Math.Atan2(Left() - B.Left, Top - B.Top) * 2 / Math.PI
If A \ 1 = A Then Bool(A + 1) = True
EndIf
Next

Return Bool
EndFunction

PrivateReadOnlyProperty Rect() AsRectangle
Get
ReturnNew Rectangle(Left, Top, Width, Height)
EndGet
EndProperty

#EndRegion

#Region" Colors"

<Category("Colors")> _
PublicProperty BaseColor AsColor
Get
Return _BaseColor
EndGet
Set(ByVal value As Color)
    _BaseColor = value
EndSet
EndProperty

<Category("Colors")> _
PublicProperty TextColor AsColor
Get
Return _TextColor
EndGet
Set(ByVal value As Color)
    _TextColor = value
EndSet
EndProperty

<Category("Options")> _
PublicProperty Rounded AsBoolean
Get
Return _Rounded
EndGet
Set(ByVal value AsBoolean)
    _Rounded = value
EndSet
EndProperty

#EndRegion

ProtectedOverridesSub OnResize(ByVal e AsEventArgs)
 MyBase.OnResize(e)
 'Height = 32
EndSub

ProtectedOverridesSub OnCreateControl()
 MyBase.OnCreateControl()
 'Size = New Size(112, 32)
EndSub

```

```

#EndRegion

#Region" Colors"

Private _BaseColor AsColor = _FlatColor
Private _TextColor AsColor = Color.FromArgb(243, 243, 243)

#EndRegion

SubNew()
    SetStyle(ControlStyles.AllPaintingInWmPaint Or ControlStyles.UserPaint Or
    ControlStyles.ResizeRedraw Or ControlStyles.OptimizedDoubleBuffer Or
    ControlStyles.SupportsTransparentBackColor, True)
    DoubleBuffered = True
    Size = New Size(106, 32)
    BackColor = Color.Transparent
    Font = New Font("Segoe UI", 12)
    Cursor = Cursors.Hand
EndSub

ProtectedOverridesSub OnPaint(ByVal e AsPaintEventArgs)
    B = New Bitmap(Width, Height) : G = Graphics.FromImage(B)
    W = Width : H = Height

    Dim GP AsNew GraphicsPath

    Dim GCS = GetConnectedSides()
    Dim RoundedBase = Helpers.RoundRect(0, 0, W, H, , Not (GCS(2) Or GCS(1)), Not
    (GCS(1) Or GCS(0)), Not (GCS(3) Or GCS(0)), Not (GCS(3) Or GCS(2)))
    Dim Base AsNew Rectangle(0, 0, W, H)

    With G
        .SmoothingMode = 2
        .PixelOffsetMode = 2
        .TextRenderingHint = 5
        .Clear(BackColor)
    EndWith

    SelectCase State
    Case MouseState.None
    If Rounded Then
        '-- Base
            GP = RoundedBase
            .FillPath(New SolidBrush(_BaseColor), GP)

        '-- Text
            .DrawString(Text, Font, New SolidBrush(_TextColor), Base,
            CenterSF)
        Else
            '-- Base
                .FillRectangle(New SolidBrush(_BaseColor), Base)

            '-- Text
                .DrawString(Text, Font, New SolidBrush(_TextColor), Base,
                CenterSF)
        EndIf
    Case MouseState.Over
    If Rounded Then
        '-- Base
            GP = RoundedBase
            .FillPath(New SolidBrush(_BaseColor), GP)
    EndIf
    EndSelect
EndSub

```

```

        .FillPath(New SolidBrush(Color.FromArgb(20, Color.White)),
GP)

'-- Text
        .DrawString(Text, Font, New SolidBrush(_TextColor), Base,
CenterSF)
Else
'-- Base
        .FillRectangle(New SolidBrush(_BaseColor), Base)
        .FillRectangle(New SolidBrush(Color.FromArgb(20,
Color.White)), Base)

'-- Text
        .DrawString(Text, Font, New SolidBrush(_TextColor), Base,
CenterSF)
EndIf
Case MouseState.Down
If Rounded Then
'-- Base
        GP = RoundedBase
        .FillPath(New SolidBrush(_BaseColor), GP)
        .FillPath(New SolidBrush(Color.FromArgb(20, Color.Black)),
GP)

'-- Text
        .DrawString(Text, Font, New SolidBrush(_TextColor), Base,
CenterSF)
Else
'-- Base
        .FillRectangle(New SolidBrush(_BaseColor), Base)
        .FillRectangle(New SolidBrush(Color.FromArgb(20,
Color.Black)), Base)

'-- Text
        .DrawString(Text, Font, New SolidBrush(_TextColor), Base,
CenterSF)
EndIf
EndSelect

EndWith

 MyBase.OnPaint(e)
 G.Dispose()
 e.Graphics.InterpolationMode = 7
 e.Graphics.DrawImageUnscaled(B, 0, 0)
 B.Dispose()
EndSub

EndClass

ClassFlatNumeric : InheritsControl

#Region" Variables"

Private W, H AsInteger
Private State As MouseState = MouseState.None
Private x, y AsInteger
Private _Value, _Min, _Max AsLong
Private Bool AsBoolean

#EndRegion

```

```

#Region" Properties"

PublicProperty Value AsLong
Get
Return _Value
EndGet
Set(ByVal value AsLong)
If value <= _Max And value >= _Min Then _Value = value
    Invalidate()
EndSet
EndProperty

PublicProperty Maximum AsLong
Get
Return _Max
EndGet
Set(ByVal value AsLong)
If value > _Min Then _Max = value
If _Value > _Max Then _Value = _Max
    Invalidate()
EndSet
EndProperty

PublicProperty Minimum AsLong
Get
Return _Min
EndGet
Set(ByVal value AsLong)
If value < _Max Then _Min = value
If _Value < _Min Then _Value = Minimum
    Invalidate()
EndSet
EndProperty

ProtectedOverridesSub OnMouseMove(ByVal e As MouseEventArgs)
 MyBase.OnMouseMove(e)
    x = e.Location.X
    y = e.Location.Y
    Invalidate()
If e.X < Width - 23 Then Cursor = Cursors.IBeam Else Cursor = Cursors.Hand
EndSub

ProtectedOverridesSub OnMouseDown(ByVal e As MouseEventArgs)
 MyBase.OnMouseDown(e)
 If x > Width - 21 AndAlso x < Width - 3 Then
 If y < 15 Then
 If (Value + 1) <= _Max Then _Value += 1
 Else
 If (Value - 1) >= _Min Then _Value -= 1
 EndIf
 Else
        Bool = Not Bool
        Focus()
    EndIf
    Invalidate()
EndSub

ProtectedOverridesSub OnKeyPress(ByVal e As KeyPressEventArgs)
 MyBase.OnKeyPress(e)
 Try
 If Bool Then _Value = CStr(CStr(_Value) & e.KeyChar.ToString())
 If _Value > _Max Then _Value = _Max

```

```

        Invalidate()
Catch : EndTry
EndSub

ProtectedOverridesSub OnKeyDown(ByVal e As KeyEventArgs)
 MyBase.OnKeyDown(e)
If e.KeyCode = Keys.Back Then
    Value = 0
EndIf
EndSub

ProtectedOverridesSub OnResize(ByVal e As EventArgs)
 MyBase.OnResize(e)
 Height = 30
EndSub

#Region" Colors"

<Category("Colors")>
PublicProperty BaseColor AsColor
Get
Return _BaseColor
EndGet
Set(ByVal value As Color)
    _BaseColor = value
EndSet
EndProperty

<Category("Colors")>
PublicProperty ButtonColor AsColor
Get
Return _ButtonColor
EndGet
Set(ByVal value As Color)
    _ButtonColor = value
EndSet
EndProperty

#EndRegion

#EndRegion

#Region" Colors"

Private _BaseColor As Color = Color.FromArgb(45, 47, 49)
Private _ButtonColor As Color = _FlatColor

#EndRegion

SubNew()
    SetStyle(ControlStyles.AllPaintingInWmPaint Or ControlStyles.UserPaint Or
    - ControlStyles.ResizeRedraw Or ControlStyles.OptimizedDoubleBuffer Or _
    ControlStyles.SupportsTransparentBackColor, True)
    DoubleBuffered = True
    Font = New Font("Segoe UI", 10)
    BackColor = Color.FromArgb(60, 70, 73)
    ForeColor = Color.White
    _Min = 0
    _Max = 999999
EndSub

```

```

ProtectedOverridesSub OnPaint(ByVal e As PaintEventArgs)
    B = New Bitmap(Width, Height) : G = Graphics.FromImage(B)
    W = Width : H = Height

Dim Base AsNew Rectangle(0, 0, W, H)

With G
    .SmoothingMode = 2
    .PixelOffsetMode = 2
    .TextRenderingHint = 5
    .Clear(BackColor)

    '-- Base
    .FillRectangle(New SolidBrush(_BaseColor), Base)
    .FillRectangle(New SolidBrush(_ButtonColor), New Rectangle(Width - 24,
0, 24, H))

    '-- Add
    .DrawString("+", New Font("Segoe UI", 12), Brushes.White, New
Point(Width - 12, 8), CenterSF)
    '-- Subtract
    .DrawString("-", New Font("Segoe UI", 10, FontStyle.Bold),
Brushes.White, New Point(Width - 12, 22), CenterSF)

    '-- Text
    .DrawString(Value, Font, Brushes.White, New Rectangle(5, 1, W, H), New
StringFormat() With {.LineAlignment = StringAlignment.Center})
EndWith

 MyBase.OnPaint(e)
    G.Dispose()
    e.Graphics.InterpolationMode = 7
    e.Graphics.DrawImageUnscaled(B, 0, 0)
    B.Dispose()
EndSub

EndClass

ClassFlatListBox : InheritsControl

#Region" Variables"

PrivateWithEvents ListBx AsNew ListBox
Private _items AsString() = {"}

#EndRegion

#Region" Properties"

<Category("Options")> _
PublicProperty items AsString()
Get
Return _items
EndGet
Set(ByVal value AsString())
    _items = value
    ListBx.Items.Clear()
    ListBx.Items.AddRange(value)
    Invalidate()
EndSet
EndProperty

```

```

<Category("Colors")> -
PublicProperty SelectedColor AsColor
Get
Return _SelectedColor
EndGet
Set(ByName value As Color)
    _SelectedColor = value
EndSet
EndProperty

PublicReadOnlyProperty SelectedItem() AsString
Get
Return ListBx.SelectedItem
EndGet
EndProperty

PublicReadOnlyProperty SelectedIndex() AsInteger
Get
Return ListBx.SelectedIndex
If ListBx.SelectedIndex < 0 ThenExit Property
EndGet
EndProperty

PublicSub Clear()
    ListBx.Items.Clear()
EndSub

PublicSub ClearSelected()
For i AsInteger = (ListBx.SelectedItems.Count - 1) To 0 Step -1
    ListBx.Items.Remove(ListBx.SelectedItems(i))
Next
EndSub

Sub Drawitem( ByVal sender AsObject, ByVal e As DrawItemEventArgs) Handles
ListBx.DrawItem
If e.Index < 0 ThenExit Sub
    e.DrawBackground()
    e.DrawFocusRectangle()

    e.Graphics.SmoothingMode = 2
    e.Graphics.PixelOffsetMode = 2
    e.Graphics.InterpolationMode = 7
    e.Graphics.TextRenderingHint = 5

If InStr(e.State.ToString, "Selected,") > 0 Then'-- if selected
'-- Base
    e.Graphics.FillRectangle(New SolidBrush(_SelectedColor), New
    Rectangle(e.Bounds.X, e.Bounds.Y, e.Bounds.Width, e.Bounds.Height))

'-- Text
    e.Graphics.DrawString(" "& ListBx.Items(e.Index).ToString(), New
    Font("Segoe UI", 8), Brushes.White, e.Bounds.X, e.Bounds.Y + 2)
Else
'-- Base
    e.Graphics.FillRectangle(New SolidBrush(Color.FromArgb(51, 53, 55)), New
    Rectangle(e.Bounds.X, e.Bounds.Y, e.Bounds.Width, e.Bounds.Height))

'-- Text
    e.Graphics.DrawString(" "& ListBx.Items(e.Index).ToString(), New
    Font("Segoe UI", 8), Brushes.White, e.Bounds.X, e.Bounds.Y + 2)
EndIf

```

```

        e.Graphics.Dispose()
EndSub

ProtectedOverridesSub OnCreateControl()
 MyBase.OnCreateControl()
 IfNot Controls.Contains(ListBx) Then
     Controls.Add(ListBx)
 EndIf
EndSub

Sub AddRange(ByVal items AsObject())
    ListBx.Items.Remove("")
    ListBx.Items.AddRange(items)
EndSub

Sub AddItem(ByVal item AsObject)
    ListBx.Items.Remove("")
    ListBx.Items.Add(item)
EndSub

#EndRegion

#Region " Colors"

Private BaseColor As Color = Color.FromArgb(45, 47, 49)
Private _SelectedColor As Color = _FlatColor

#EndRegion

SubNew()
    SetStyle(ControlStyles.AllPaintingInWmPaint Or ControlStyles.UserPaint Or
    - ControlStyles.ResizeRedraw Or ControlStyles.OptimizedDoubleBuffer,
    True)
    DoubleBuffered = True

    ListBx.DrawMode = Windows.Forms.DrawMode.OwnerDrawFixed
    ListBx.ScrollAlwaysVisible = False
    ListBx.HorizontalScrollbar = False
    ListBx.BorderStyle = BorderStyle.None
    ListBx.BackColor = BaseColor
    ListBx.ForeColor = Color.White
    ListBx.Location = New Point(3, 3)
    ListBx.Font = New Font("Segoe UI", 8)
    ListBx.ItemHeight = 20
    ListBx.Items.Clear()
    ListBx.IntegralHeight = False

    Size = New Size(131, 101)
    BackColor = BaseColor
EndSub

ProtectedOverridesSub OnPaint(ByVal e As PaintEventArgs)
    B = New Bitmap(Width, Height) : G = Graphics.FromImage(B)

    Dim Base AsNew Rectangle(0, 0, Width, Height)

    With G
        .SmoothingMode = 2
        .PixelOffsetMode = 2
        .TextRenderingHint = 5
    EndWith
EndSub

```

```

        .Clear(BackColor)

'-- Size
    ListBx.Size = New Size(Width - 6, Height - 2)

'-- Base
    .FillRectangle(New SolidBrush(BaseColor), Base)
EndWith

 MyBase.OnPaint(e)
    G.Dispose()
    e.Graphics.InterpolationMode = 7
    e.Graphics.DrawImageUnscaled(B, 0, 0)
    B.Dispose()
EndSub

EndClass

ClassFlatContextMenuStrip : InheritsContextMenuStrip

ProtectedOverridesSub OnTextChanged(ByVal e As EventArgs)
MyBase.OnTextChanged(e) : Invalidate()
EndSub

SubNew()
MyBase.New()
    Renderer = New ToolStripProfessionalRenderer(New TColorTable())
    ShowImageMargin = False
    ForeColor = Color.White
    Font = New Font("Segoe UI", 8)
EndSub

ProtectedOverridesSub OnPaint(ByVal e As PaintEventArgs)
MyBase.OnPaint(e)
    e.Graphics.TextRenderingHint = 5
EndSub

ClassTColorTable : Inherits ProfessionalColorTable

#Region" Properties"

#Region" Colors"

<Category("Colors")> _
PublicProperty _BackColor As Color
Get
Return BackColor
EndGet
Set(ByVal value As Color)
    BackColor = value
EndSet
EndProperty

<Category("Colors")> _
PublicProperty _CheckedColor As Color
Get
Return CheckedColor
EndGet
Set(ByVal value As Color)
    CheckedColor = value
EndSet
EndProperty

```

```
<Category("Colors")> _
PublicProperty _BorderColor As Color
Get
Return BorderColor
EndGet
Set(ByVal value As Color)
    BorderColor = value
EndSet
EndProperty

#EndRegion

#Region" Colors"

Private BackColor As Color = Color.FromArgb(45, 47, 49)
Private CheckedColor As Color = _FlatColor
Private BorderColor As Color = Color.FromArgb(53, 58, 60)

#EndRegion

#Region" Overrides"

PublicOverridesReadOnlyProperty ButtonSelectedBorder As Color
Get
Return BackColor
EndGet
EndProperty
PublicOverridesReadOnlyProperty CheckBackground() As Color
Get
Return CheckedColor
EndGet
EndProperty
PublicOverridesReadOnlyProperty CheckPressedBackground() As Color
Get
Return CheckedColor
EndGet
EndProperty
PublicOverridesReadOnlyProperty CheckSelectedBackground() As Color
Get
Return CheckedColor
EndGet
EndProperty
PublicOverridesReadOnlyProperty ImageMarginGradientBegin() As Color
Get
Return CheckedColor
EndGet
EndProperty
PublicOverridesReadOnlyProperty ImageMarginGradientEnd() As Color
Get
Return CheckedColor
EndGet
EndProperty
PublicOverridesReadOnlyProperty ImageMarginGradientMiddle() As Color
Get
Return CheckedColor
EndGet
EndProperty
PublicOverridesReadOnlyProperty MenuBorder() As Color
Get
```

```

Return BorderColor
EndGet
EndProperty
PublicOverridesReadOnlyProperty MenuItemBorder() As Color
Get
Return BorderColor
EndGet
EndProperty
PublicOverridesReadOnlyProperty MenuItemSelected() As Color
Get
Return CheckedColor
EndGet
EndProperty
PublicOverridesReadOnlyProperty SeparatorDark() As Color
Get
Return BorderColor
EndGet
EndProperty
PublicOverridesReadOnlyProperty ToolStripDropDownBackground() As Color
Get
Return BackColor
EndGet
EndProperty

#EndRegion

EndClass

EndClass

<DefaultEvent("Scroll")>ClassFlatTrackBar : InheritsControl

#Region" Variables"

Private W, H AsInteger
Private Val AsInteger
Private Bool AsBoolean
Private Track As Rectangle
Private Knob As Rectangle
Private Style_ As _Style

#EndRegion

#Region" Properties"

#EndRegion" Mouse States"

ProtectedOverridesSub OnMouseDown(ByVal e As MouseEventArgs)
 MyBase.OnMouseDown(e)
 If e.Button = Windows.Forms.MouseButtons.Left Then
     Val = CInt((_Value - _Minimum) / (_Maximum - _Minimum) * (Width - 11))
     Track = New Rectangle(Val, 0, 10, 20)

     Bool = Track.Contains(e.Location)
 EndIf
 EndSub

ProtectedOverridesSub OnMouseMove(ByVal e As MouseEventArgs)
 MyBase.OnMouseMove(e)
 If Bool AndAlso e.X > -1 AndAlso e.X < (Width + 1) Then
     Value = _Minimum + CInt((_Maximum - _Minimum) * (e.X / Width))
 EndIf

```

```

EndSub

ProtectedOverridesSub OnMouseUp(ByVal e As MouseEventArgs)
MyBase.OnMouseUp(e) : Bool = False
EndSub

#EndRegion

#Region" Styles"

<Flags()> _
Enum_Style
    Slider
    Knob
EndEnum

PublicProperty Style As_Style
Get
Return Style_
EndGet
Set(ByVal value As _Style)
    Style_ = value
EndSet
EndProperty

#EndRegion

#Region" Colors"

<Category("Colors")> _
PublicProperty TrackColor AsColor
Get
Return _TrackColor
EndGet
Set(ByVal value As Color)
    _TrackColor = value
EndSet
EndProperty

<Category("Colors")> _
PublicProperty HatchColor AsColor
Get
Return _HatchColor
EndGet
Set(ByVal value As Color)
    _HatchColor = value
EndSet
EndProperty

#EndRegion

Event Scroll(ByVal sender AsObject)
Private _Minimum AsInteger
PublicProperty Minimum AsInteger
Get
Return Minimum
EndGet
Set(ByVal value AsInteger)
If value < 0 Then
EndIf

    _Minimum = value

```

```

If value > _Value Then _Value = value
If value > _Maximum Then _Maximum = value
    Invalidate()
EndSet
EndProperty
Private _Maximum AsInteger = 10
PublicProperty Maximum AsInteger
Get
Return _Maximum
EndGet
Set(ByVal value AsInteger)
If value < 0 Then
EndIf

    _Maximum = value
If value < _Value Then _Value = value
If value < _Minimum Then _Minimum = value
    Invalidate()
EndSet
EndProperty
Private _Value AsInteger
PublicProperty Value AsInteger
Get
Return _Value
EndGet
Set(ByVal value AsInteger)
If value = _Value ThenReturn

If value > _Maximum OrElse value < _Minimum Then
EndIf

    _Value = value
    Invalidate()
RaiseEvent Scroll(Me)
EndSet
EndProperty
Private _ShowValue AsBoolean = False
PublicProperty ShowValue AsBoolean
Get
Return _ShowValue
EndGet
Set(ByVal value AsBoolean)
    _ShowValue = value
EndSet
EndProperty

ProtectedOverridesSub OnKeyDown(ByVal e As KeyEventArgs)
 MyBase.OnKeyDown(e)
If e.KeyCode = Keys.Subtract Then
If Value = 0 ThenExit Sub
    Value -= 1
ElseIf e.KeyCode = Keys.Add Then
If Value = _Maximum ThenExit Sub
    Value += 1
EndIf
EndSub

ProtectedOverridesSub OnTextChanged(ByVal e As EventArgs)
 MyBase.OnTextChanged(e) : Invalidate()
EndSub

```

```

ProtectedOverridesSub OnResize(ByVal e As EventArgs)
 MyBase.OnResize(e)
 Height = 23
EndSub

#EndRegion

#Region" Colors"

Private BaseColor As Color = Color.FromArgb(45, 47, 49)
Private _TrackColor As Color = _FlatColor
Private SliderColor As Color = Color.FromArgb(25, 27, 29)
Private _HatchColor As Color = Color.FromArgb(23, 148, 92)

#EndRegion

SubNew()
 SetStyle(ControlStyles.AllPaintingInWmPaint Or ControlStyles.UserPaint Or
 -
 ControlStyles.ResizeRedraw Or
 ControlStyles.OptimizedDoubleBuffer, True)
 DoubleBuffered = True
 Height = 18

 BackColor = Color.FromArgb(60, 70, 73)
EndSub

ProtectedOverridesSub OnPaint(ByVal e As PaintEventArgs)
 B = New Bitmap(Width, Height) : G = Graphics.FromImage(B)
 W = Width - 1 : H = Height - 1

Dim Base AsNew Rectangle(1, 6, W - 2, 8)
Dim GP, GP2 AsNew GraphicsPath

With G
 .SmoothingMode = 2
 .PixelOffsetMode = 2
 .TextRenderingHint = 5
 .Clear(BackColor)

'-- Value
 Val = CInt((_Value - _Minimum) / (_Maximum - _Minimum) * (W - 10))
 Track = New Rectangle(Val, 0, 10, 20)
 Knob = New Rectangle(Val, 4, 11, 14)

'-- Base
 GP.AddRectangle(Base)
 .SetClip(GP)
 .FillRectangle(New SolidBrush(BaseColor), New Rectangle(0, 7, W, 8))
 .FillRectangle(New SolidBrush(_TrackColor), New Rectangle(0, 7,
 Track.X + Track.Width, 8))
 .ResetClip()

'-- Hatch Brush
 Dim HB AsNew HatchBrush(HatchStyle.Plaid, HatchColor, _TrackColor)
 .FillRectangle(HB, New Rectangle(-10, 7, Track.X + Track.Width, 8))

'-- Slider/Knob
 SelectCase Style
 Case _Style.Slider
 GP2.AddRectangle(Track)
 .FillPath(New SolidBrush(SliderColor), GP2)

```

```

Case _Style.Knob
    GP2.AddEllipse(Knob)
        .FillPath(New SolidBrush(SliderColor), GP2)
EndSelect

'-- Show the value
If ShowValue Then
    .DrawString(Value, New Font("Segoe UI", 8), Brushes.White, New
    Rectangle(1, 6, W, H), New StringFormat()
    With {.Alignment = StringAlignment.Far, .LineAlignment = StringAlignment.Far})
EndIf
EndWith

 MyBase.OnPaint(e)
    G.Dispose()
    e.Graphics.InterpolationMode = 7
    e.Graphics.DrawImageUnscaled(B, 0, 0)
    B.Dispose()
EndSub
EndClass

ClassFlatStatusBar : InheritsControl

#Region" Variables"

Private W, H AsInteger
Private _ShowTimeDate AsBoolean = False

#EndRegion

#Region" Properties"

ProtectedOverridesSub CreateHandle()
MyBase.CreateHandle()
    Dock = DockStyle.Bottom
EndSub

ProtectedOverridesSub OnTextChanged(ByVal e As EventArgs)
MyBase.OnTextChanged(e) : Invalidate()
EndSub

#Region" Colors"

<Category("Colors")>
PublicProperty BaseColor AsColor
Get
Return _BaseColor
EndGet
Set(ByVal value As Color)
    _BaseColor = value
EndSet
EndProperty

<Category("Colors")>
PublicProperty TextColor AsColor
Get
Return _TextColor
EndGet
Set(ByVal value As Color)
    _TextColor = value
EndSet
EndProperty

```

```

<Category("Colors")>
PublicProperty RectColor AsColor
Get
Return _RectColor
EndGet
Set(ByVal value As Color)
    _RectColor = value
EndSet
EndProperty

#EndRegion

PublicProperty ShowTimeDate AsBoolean
Get
Return _ShowTimeDate
EndGet
Set(ByVal value AsBoolean)
    _ShowTimeDate = value
EndSet
EndProperty

Function GetTimeDate() AsString
Return DateTime.Now.Date & " " & DateTime.Now.Hour & ":" & DateTime.Now.Minute
EndFunction

#EndRegion

#Region" Colors"

Private _BaseColor As Color = Color.FromArgb(45, 47, 49)
Private _TextColor As Color = Color.White
Private _RectColor As Color = _FlatColor

#EndRegion

SubNew()
    SetStyle(ControlStyles.AllPaintingInWmPaint Or ControlStyles.UserPaint Or
    - ControlStyles.ResizeRedraw Or
ControlStyles.OptimizedDoubleBuffer, True)
    DoubleBuffered = True
    Font = New Font("Segoe UI", 8)
    ForeColor = Color.White
    Size = New Size(Width, 20)
EndSub

ProtectedOverridesSub OnPaint(ByVal e As PaintEventArgs)
    B = New Bitmap(Width, Height) : G = Graphics.FromImage(B)
    W = Width : H = Height

    Dim Base AsNew Rectangle(0, 0, W, H)

    With G
        .SmoothingMode = 2
        .PixelOffsetMode = 2
        .TextRenderingHint = 5
        .Clear(BaseColor)

        '-- Base
        .FillRectangle(New SolidBrush(BaseColor), Base)
    EndWith
EndSub

```

```

'-- Text
    .DrawString(Text, Font, Brushes.White, New Rectangle(10, 4, W, H),
NearSF)

'-- Rectangle
    .FillRectangle(New SolidBrush(_RectColor), New Rectangle(4, 4, 4, 14))

'-- TimeDate
If ShowTimeDate Then
    .DrawString(GetTimeDate, Font, New SolidBrush(_TextColor), New
Rectangle(-4, 2, W, H), New StringFormat() _
With {.Alignment = StringAlignment.Far, .LineAlignment = StringAlignment.Center})
EndIf
EndWith

 MyBase.OnPaint(e)
    G.Dispose()
    e.Graphics.InterpolationMode = 7
    e.Graphics.DrawImageUnscaled(B, 0, 0)
    B.Dispose()
EndSub
EndClass

ClassFlatButton : InheritsLabel

ProtectedOverridesSub OnTextChanged(ByVal e As EventArgs)
MyBase.OnTextChanged(e) : Invalidate()
EndSub

SubNew()
    SetStyle(ControlStyles.SupportsTransparentBackColor, True)
    Font = New Font("Segoe UI", 8)
    ForeColor = Color.White
    BackColor = Color.Transparent
    Text = Text
EndSub

EndClass

ClassFlatTreeView : InheritsTreeView

#Region" Variables"

Private State As TreeNodeStates

#EndRegion

#Region" Properties"

ProtectedOverridesSub OnDrawNode(ByVal e As DrawTreeNodeEventArgs)
Try
Dim Bounds AsNew Rectangle(e.Bounds.Location.X, e.Bounds.Location.Y,
e.Bounds.Width, e.Bounds.Height)
'e.Node.Nodes.Item.
SelectCase State
Case TreeNodeStates.Default
    e.Graphics.FillRectangle(Brushes.Red, Bounds)
    e.Graphics.DrawString(e.Node.Text, New Font("Segoe UI", 8),
Brushes.LimeGreen, New Rectangle(Bounds.X + 2, Bounds.Y + 2, Bounds.Width,
Bounds.Height), NearSF)
    Invalidate()
Case TreeNodeStates.Checked

```

```

e.Graphics.FillRectangle(Brushes.Green, Bounds)
e.Graphics.DrawString(e.Node.Text, New Font("Segoe UI", 8),
Brushes.Black, New Rectangle(Bounds.X + 2, Bounds.Y + 2, Bounds.Width,
Bounds.Height), NearSF)
Invalidate()
Case TreeNodeStates.Selected
    e.Graphics.FillRectangle(Brushes.Green, Bounds)
    e.Graphics.DrawString(e.Node.Text, New Font("Segoe UI", 8),
Brushes.Black, New Rectangle(Bounds.X + 2, Bounds.Y + 2, Bounds.Width,
Bounds.Height), NearSF)
    Invalidate()
EndSelect

Catch ex As Exception
    MsgBox(ex.Message)
EndTry
 MyBase.OnDrawNode(e)
EndSub

#EndRegion
#Region" Colors"

Private _BaseColor As Color = Color.FromArgb(45, 47, 49)
Private _LineColor As Color = Color.FromArgb(25, 27, 29)

#EndRegion
SubNew()

    SetStyle(ControlStyles.AllPaintingInWmPaint Or ControlStyles.UserPaint Or
-
        ControlStyles.ResizeRedraw Or
ControlStyles.OptimizedDoubleBuffer, True)
    DoubleBuffered = True

    BackColor = _BaseColor
    ForeColor = Color.White
    LineColor = _LineColor
    DrawMode = TreeViewDrawMode.OwnerDrawAll
EndSub

ProtectedOverridesSub OnPaint(ByVal e As PaintEventArgs)
    B = New Bitmap(Width, Height) : G = Graphics.FromImage(B)

    Dim Base AsNew Rectangle(0, 0, Width, Height)

    With G
        .SmoothingMode = 2
        .PixelOffsetMode = 2
        .TextRenderingHint = 5
        .Clear(BackColor)

        .FillRectangle(New SolidBrush(_BaseColor), Base)
        .DrawString(Text, New Font("Segoe UI", 8), Brushes.Black, New
Rectangle(Bounds.X + 2, Bounds.Y + 2, Bounds.Width, Bounds.Height), NearSF)
    EndWith
    MyBase.OnPaint(e)
    G.Dispose()
    e.Graphics.InterpolationMode = 7
    e.Graphics.DrawImageUnscaled(B, 0, 0)
    B.Dispose()
EndSub
EndClass

```

## **APPENDIX D**

### **Letter to the Panelist**



Republic of the Philippines  
**JOSE RIZAL MEMORIAL STATE UNIVERSITY**  
The Premier University in Zamboanga Del Norte  
Main Campus, Dapitan city



May 17, 2017

**ENGR. ANGELITO M. PUNZAL**  
Computer Scinece Department  
Jose Rizal Memorial State University

Sir,

The undersigned are the 4<sup>th</sup> year BS Computer Science who presently conducting the study entitled "**Intelligent IP Address Locator via Google Map**". This is in partial fulfillment of the subject Project Thesis.

We would like to request your present as one of the panelists on May 17, 2017 at Research Extension Room, Dapitan City.

Hoping for your kind response on this request.

Respectfully yours,

Jay Marie A. Egoogan  
Lea E. Monopollo  
Khimberly R. Ebe

Approved:

**ENGR. ANGELITO M. PUNZAL, MSIT**  
Adviser/Instructor



Republic of the Philippines  
**JOSE RIZAL MEMORIAL STATE UNIVERSITY**  
The Premier University in Zamboanga Del Norte  
Main Campus, Dapitan city



May 17, 2017

**ENGR. JOSEPH AURELIUS P. JACINTO**  
Computer Scinece Department  
Jose Rizal Memorial State University

Sir,

The undersigned are the 4<sup>th</sup> year BS Computer Science who presently conducting the study entitled "**Intelligent IP Address Locator via Google Map**". This is in partial fulfillment of the subject Project Thesis.

We would like to request your present as one of the panelists on May 17, 2017 at Research Extension Room, Dapitan City.

Hoping for your kind response on this request.

Respectfully yours,

Jay Marie A. Egoogan  
Lea E. Monopollo  
Khimberly R. Ebe

Approved:

**ENGR. ANGELITO M. PUNZAL, MSIT**  
Adviser/Instructor



Republic of the Philippines  
**JOSE RIZAL MEMORIAL STATE UNIVERSITY**  
The Premier University in Zamboanga Del Norte  
Main Campus, Dapitan city



May 17, 2017

**PROF. JHON D. SAGAPSAPAN**

Computer Scinece Department  
Jose Rizal Memorial State University

Sir,

The undersigned are the 4<sup>th</sup> year BS Computer Science who presently conducting the study entitled "**Intelligent IP Address Locator via Google Map**". This is in partial fulfillment of the subject Project Thesis.

We would like to request your present as one of the panelists on May 17, 2017 at Research Extension Room, Dapitan City.

Hoping for your kind response on this request.

Respectfully yours,

Jay Marie A. Egoogan  
Lea E. Monopollo  
Khimberly R. Ebe

Approved:

**ENGR. ANGELITO M. PUNZAL, MSIT**  
Adviser/Instructor



Republic of the Philippines  
**JOSE RIZAL MEMORIAL STATE UNIVERSITY**  
The Premier University in Zamboanga Del Norte  
Main Campus, Dapitan city



May 17, 2017

**ARMANDO T. SAGUIN JR.**  
Computer Scinece Department  
Jose Rizal Memorial State University

Sir,

The undersigned are the 4<sup>th</sup> year BS Computer Science who presently conducting the study entitled "**Intelligent IP Address Locator via Google Map**". This is in partial fulfillment of the subject Project Thesis.

We would like to request your present as one of the panelists on May 17, 2017 at Research Extension Room, Dapitan City.

Hoping for your kind response on this request.

Respectfully yours,

Jay Marie A. Egoogan  
Lea E. Monopollo  
Khimberly R. Ebe

Approved:

**ENGR. ANGELITO M. PUNZAL, MSIT**  
Adviser/Instructor



Republic of the Philippines  
**JOSE RIZAL MEMORIAL STATE UNIVERSITY**  
The Premier University in Zamboanga Del Norte  
Main Campus, Dapitan city



May 17, 2017

**PROF. ED NEIL O. MARATAS**  
Computer Scinece Department  
Jose Rizal Memorial State University

Sir,

The undersigned are the 4<sup>th</sup> year BS Computer Science who presently conducting the study entitled "**Intelligent IP Address Locator via Google Map**". This is in partial fulfillment of the subject Project Thesis.

We would like to request your present as one of the panelists on May 17, 2017 at Research Extension Room, Dapitan City.

Hoping for your kind response on this request.

Respectfully yours,

Jay Marie A. Egoogan  
Lea E. Monopollo  
Khimberly R. Ebe

Approved:

**ENGR. ANGELITO M. PUNZAL, MSIT**  
Adviser/Instructor

## **APPENDIX E**

### **Evaluation Letter**



Republic of the Philippines  
**JOSE RIZAL MEMORIAL STATE UNIVERSITY**  
The Premier University in Zamboanga Del Norte  
Main Campus, Dapitan city



## **EVALUATION LETTER**

**Dear Respondents,**

The undersigned is gathering data for thesis entitled, "**Intelligent IP Address Locator via Google Map**". Please evaluate the software with all your sincerity and honesty.

Your cooperation will contribute to the data in this study.

Thank you very much.

Very truly yours,

Jay Marie A. Egoogan

Lea E. Monopollo

Khimberly R. Ebe

## INTELLIGENT IP ADDRESS LOCATOR VIA GOOGLE MAP

### Evaluation Sheet

Name of Evaluator/Rater: \_\_\_\_\_

Position /Job Description: \_\_\_\_\_

**Direction:** Please evaluate the developed Intelligent IP Address Locator via Google Map system software along the levels of functionality, reliability, usability, efficiency, maintainability, and portability, Description of each level is indicated below.

**Functionality** pertains to the sum or any aspect of what Intelligent IP Address Locator via Google Map can do for the user. Indicate the functionality level of the system software by putting a check in the box that fits your response. The numbers are coded as follows:

- 5- Very Much Functional
- 4- Much Functional
- 3- Functional
- 2- Fairly Functional
- 1 -Not Functional

FUNCTIONALITY	5	4	3	2	1
1. Intended use of the software					
2. Data manipulation					
3. Compliance of end-user needs					
4. Security of system data					
5. Compatibility of other system					
6. Speed in data processing					
7. Fitness of its intended use					
8. Minimization of its run-time error					
9. Detection of error					
10. Data storage					

**Reliability** refers to the ability of the Intelligent IP Address Locator via Google Map to consistently perform its intended or required function or mission on demand and without degradation or failure. Indicate the reliability level of the system software by putting a check box that fits your numbers are coded as follows:

- 5- Very Much Reliable
- 4- Much Reliable
- 3 - Reliable
- 2- Fairly Reliable
- 1 -Not Reliable

RELIABILITY	5	4	3	2	1
1. Error tolerance					
2. Ease in data recovery					
3. Program security					
4. Data security					
5. Creation of back-up system					
6. Accuracy of data capture					
7. Accuracy of Results					
8. Data storage volume					
9. Understand ability of output					
10. Completeness of the system					

**Usability** pertains to the user's total satisfaction received from using the proposed Intelligent IP Address Locator via Google Map. To determine also the usability level of the system software, put a check in the box that fits your response. The numbers are coded as follows:

- 5- Very Much Usable
- 4- Much Usable
- 3- Usable
- 2- Fairly Usable
- 1 - Not Usable

USABILITY	5	4	3	2	1
1. User friendly program					
2. Quick driven program					
3. Simple manipulation features					
4. Wrong key input errors detection					
5. Wrong time input errors detection					
6. Data storage					
7. Data retrieval					
8. Data edit/correction					
9. Tolerable difficulty level					
10. Production of data output					

**Efficiency** refers to the Intelligent IP Address Locator via Google Map of what is actually produced or performed with what can be achieved in contrast with the traditional employee's attendance system. To find also the efficiency level of the system software, put a check in the box that fits your response. The numbers are coded as follows:

- 5- Very Much Efficient
- 4- Much Efficient
- 3 – Efficient
- 2- Fairly Efficient
- 1 -Not Efficient

EFFICIENCY	5	4	3	2	1
1. Support on minimum facilities					
2. Support on minimum requirements					
3. Provision of configurable automation					
4. Support on business workflow process					
5. Support on number of user's					
6. Speed of navigation and production of outputs					
7. Speed of data capture and retrieval					
8. Hardware utilization					
9. Support on interfacing with other devices					
10. Compatibility with interfaced devices					

**Maintainability** pertains to be characteristics of the Intelligent IP Address Locator via Google Map and its installation which determines the probability that its failure can be restored to its normal operable state within a given timeframe, using the prescribed practices and procedures. That involves two main components, namely: serviceability which includes the ease of conducting scheduled inspections and servicing, and reparability which describes the ease of restoring service after a failure. To evaluate the maintainability level of the system software, put a check in the box that fits your response. The numbers are coded as follows:

- 5- Very Much Maintainable
- 4- Much Maintainable
- 3- Maintainable
- 2- Fairly Maintainable
- 1 -Not Maintainable

MAINTAINABILITY	5	4	3	2	1
1. Modification of the system software					
2. Change of software capabilities					
3. Increase program capabilities					
4. Improving performance					
5. Correction of program defects					
6. Accessibility for maintenance					
7. Configuration of system operation					
8. Flexibility for system modification					
9. Compliance of concurrent system requirements					
10. Advance feature for recent technology					

**Portability** refers to the ability of the Intelligent IP Address Locator via Google Map to match the actual expected performance of the system being measured. Please indicate also the level of portability of the system software by putting a check in the box that fits your response. The numbers are coded as follows:

- 5- Very Much Portable
- 4- Much Portable
- 3- Portable
- 2- Fairly Portable
- 1 -Not Portable

PORATABILITY	5	4	3	2	1
1. Modification of the system					
2. Adaptability to other environment					
3. Adaptability to other applications					
4. Flexibility to other settings					
5. Support in any form of network communication					
6. Adaptability to new version of system requirements					
7. System supports on maximum hardware requirements					
8. User capability/capacity					
9. Its intended application					
10. Its intended design					

**Remarks:** \_\_\_\_\_

**Signature of Evaluator/Rater:** \_\_\_\_\_

**Date:** \_\_\_\_\_

### **Table /Tally of Result**

<b>Numerical Rating</b>	<b>Corresponding Rating</b>
4.21 – 5.00	Very Much functional
3.41 – 4.20	Much Functional
2.61 – 3.40	Moderately Functional
1.81 – 2.60	Less Functional
1.00 – 1.80	Not Reliable Functional

**Functionality.** Pertains to the sum or any aspect of what Intelligent IP Address Locator via Google Map can do for the user. Indicate the functionality level of the system software by putting a check in the box that fits your response. The numbers are coded as follows:

**Table. Tally Result Functionality**

<b>FUNCTIONALITY</b>	<b>5</b>	<b>4</b>	<b>3</b>	<b>2</b>	<b>1</b>	<b>Mean</b>	<b>Description</b>
1. Intended use of the software						<b>4.53</b>	Very Much Functional
2. Data manipulation						<b>4.3</b>	Very Much Functional
3. Compliance of end-user needs						<b>4.3</b>	Very Much Functional
4. Security of system data						<b>4.3</b>	Very Much Functional
5. Compatibility of other system						<b>4.4</b>	Very Much Functional
6. Speed in data processing						<b>4.5</b>	Very Much Functional
7. Fitness of its intended use						<b>4.4</b>	Very Much Functional
8. Minimization of its run-time error						<b>4.5</b>	Very Much Functional
9. Detection of error						<b>4.5</b>	Very Much Functional
10. Data storage						<b>4.6</b>	Very Much Functional
<b>Average Mean</b>						<b>4.43</b>	<b>Very Much Functional</b>

**Reliability.** The ability of a system to perform its required functions under stated conditions whenever required functions under stated conditions whenever required – having a long mean time between failures.

**Table. Tally Result Reliability**

RELIABILITY	5	4	3	2	1	Mean	Description
1. Error tolerance						<b>4.4</b>	Very Much Reliable
2. Ease in data recovery						<b>4.4</b>	Very Much Reliable
3. Program security						<b>4.5</b>	Very Much Reliable
4. Data security						<b>4.5</b>	Very Much Reliable
5. Creation of back-up system						<b>4.5</b>	Very Much Reliable
6. Accuracy of data capture						<b>4.1</b>	Much Reliable
7. Accuracy of Results						<b>4.3</b>	Very Much Reliable
8. Data storage volume						<b>4.3</b>	Very Much Reliable
9. Understand ability of output						<b>4.3</b>	Very Much Reliable
10. Completeness of the system						<b>4.5</b>	Very Much Reliable
<b>Average</b>						<b>4.38</b>	<b>Very Much Reliable</b>

**Usability.** The software application can be used by specified consumers to achieve quantified objectives with effectiveness, efficiency, and satisfaction in a quantified context of use.

**Table Tally Result Usability**

USABILITY	5	4	3	2	1	Mean	Description
1. User friendly program						<b>4.6</b>	Very Much Usable
2. Quick driven program						<b>4.3</b>	Very Much Usable
3. Simple manipulation features						<b>4.2</b>	Very Much Usable
4. Wrong key input errors detection						<b>4.2</b>	Very Much Usable
5. Wrong time input errors detection						<b>4.3</b>	Very Much Usable
6. Data storage						<b>4.2</b>	Very Much Usable
7. Data retrieval						<b>4.4</b>	Very Much Usable
8. Data edit/correction						<b>4.4</b>	Very Much Usable
9. Tolerable difficulty level						<b>4.4</b>	Very Much Usable
10. Production of data output						<b>4.3</b>	Very Much Usable
<b>Average</b>						<b>4.33</b>	<b>Very Much Usable</b>

**Efficiency.** Software that being developed to provide a reliable, secure and efficient method of recording the IP Addresses.

**Table Tally Result Efficiency**

EFFICIENCY	5	4	3	2	1	Mean	Description
1. Support on minimum facilities						<b>4.5</b>	Very Much Efficient
2. Support on minimum requirements						<b>4.4</b>	Very Much Efficient
3. Provision of configurable automation						<b>4.4</b>	Very Much Efficient
4. Support on business workflow process						<b>4.4</b>	Very Much Efficient
5. Support on number of user's						<b>4.4</b>	Very Much Efficient
6. Speed of navigation and production of outputs						<b>4.3</b>	Very Much Efficient
7. Speed of data capture and retrieval						<b>4.5</b>	Very Much Efficient
8. Hardware utilization						<b>4.4</b>	Very Much Efficient
9. Support on interfacing with other devices						<b>4.5</b>	Very Much Efficient
10. Compatibility with interfaced devices						<b>4.6</b>	Very Much Efficient
<b>Average</b>						<b>4.44</b>	<b>Very Much Efficient</b>

**Portability.** A software that is capable of being transferred from one employer to another.

PORABILITY	5	4	3	2	1	Mean	Description
1. Modification of the system						<b>4.5</b>	Very Much Portable
2. Adaptability to other environment						<b>4.4</b>	Very Much Portable
3. Adaptability to other applications						<b>4.4</b>	Very Much Portable
4. Flexibility to other settings						<b>4.4</b>	Very Much Portable
5. Support in any form of network communication						<b>4.6</b>	Very Much Portable
6. Adaptability to new version of system requirements						<b>4.3</b>	Very Much Portable
7. System supports on maximum hardware requirements						<b>4.4</b>	Very Much Portable
8. User capability/capacity						<b>4.5</b>	Very Much Portable
9. Its intended application						<b>4.5</b>	Very Much Portable
10. Its intended design						<b>4.6</b>	Very Much Portable
<b>Average</b>						<b>4.46</b>	<b>Very Much Portable</b>

**Maintainability.** Intelligent IP Address Locator via Google Maps is measures the ease and speed with which a system can be restored to operational status after a failure occurs.

MAINTAINABILITY	5	4	3	2	1	Mean	Description
1. Modification of the system software						<b>4.6</b>	Very Much Maintainable
2. Change of software capabilities						<b>4.5</b>	Very Much Maintainable
3. Increase program capabilities						<b>4.3</b>	Very Much Maintainable
4. Improving performance						<b>4.3</b>	Very Much Maintainable
5. Correction of program defects						<b>4.4</b>	Very Much Maintainable
6. Accessibility for maintenance						<b>4.4</b>	Very Much Maintainable
7. Configuration of system operation						<b>4.4</b>	Very Much Maintainable
8. Flexibility for system modification						<b>4.5</b>	Very Much Maintainable
9. Compliance of concurrent system requirements						<b>4.3</b>	Very Much Maintainable
10. Advance feature for recent technology						<b>4.6</b>	Very Much Maintainable
<b>Average</b>						<b>4.43</b>	<b>Very Much Maintainable</b>

**APPENDIX F****Minutes of Corrections**



Republic of the Philippines  
**JOSE RIZAL MEMORIAL STATE UNIVERSITY**  
The Premier University in Zamboanga Del Norte  
Main Campus, Dapitan city



### **MINUTES OF CORRECTIONS**

On May 17, 2017 we start our final defend at 8:00 am in the morning. We present our research study to the panelist entitled “Intelligent IP Address Locator via Google Map”, and after our presentation the Chairman of the panelist Engr. Joseph Aurelius P. Jacinto and his co-panelist Sir Armando T. Saguin JR., Sir John D. Sagapsapan and Engr. Angelito M. Punzal, Prof. Ed Neil O. Maratas have some brief questions and corrections that follows:

**PROF. ARMANDO T. SAGUIN JR.**

- ❖ Chapter 4 (page 54) – Comparison

**ORAL:**

- Add Comments
- Like

**PROF. JOHN D. SAGAPSAPAN**

- ❖ (page vii-x) – Table of Contents (indentation and spacing)
- ❖ (page x-xi) – Appendecies (indentation and spacing)
- ❖ (page 2-12) – chapter 1 (margin, indentation, spacing)
- ❖ (page 15-28) – chapter 2 (spacing, margin,indention and boxin all figures)
- ❖ (page 29-39) – chapter 3 (spacing, margin,indention and boxin all figures)

**ORAL:**

- Add Comments
- Like

**ENGR. JOSEPH AURELIUS P. JACINTO**

- ❖ (page i) – Title Page (single space)
- ❖ (page v) – Dedication (margin)
- ❖ (page vi) – Acknowledgement (margin)
- ❖ (page vii) – Abstract (background goal)
- ❖ (page 1-12) – Chapter 1 (change font size, indentation and margin)
- ❖ (page 13) – Chapter 2 (change font size)

ORAL:

- Add Comments
- Like

PROF. ED NEIL O MARATAS

ORAL:

- Add Comments
- Like

ENGR. ANGELITO M. PUNZAL

- ❖ (page i) – Title Page (single space)
- ❖ (pagevii) – Table of Contents (change of order)
- ❖ (page x) – Appendecies (specing)
- ❖ (page xi) – List of Tables (list all tables)

ORAL:

- Add Comments
- Like

## **APPENDIX G**

### **Curriculum Vitae**

# APPENDICES

## APPENDIX A

### Bibliography

#### Internet Source

#### Literature

Vigneshwaran, Sumithra & Janani (2015). An Intelligent Tracking System Based on GSM and GPS Using Smartphones. College of Engineering and Technology, Tiruchirappalli, Tamilnadu, India1&3 Associate Professor, Department of ECE, Pavendar Bharathidasan College of Engineering and

Technology, Tiruchirappalli, Tamilnadu, IndiaVol. 4, Issue 5, May 2015.[https://www.ijareeie.com/upload/2015/may/16\\_AN-1.pdf](https://www.ijareeie.com/upload/2015/may/16_AN-1.pdf)

Bhatia, J. & Verma, P. (2013).Design and Development of GPS-GSM Based Tracking System with Google Map Based Monitoring. Centre for Development of Advanced Computing, Mohali, Punjab, IndiaVol.3, No.3, June 2013.[https://www.academia.edu/22721340/GSM\\_BASED\\_TRACKING\\_SYSTEM\\_WITH GOOGLE\\_MAP\\_BASED\\_MONITORINGDESIGN\\_AND\\_DEVELOPMENT\\_OF\\_GPS-](https://www.academia.edu/22721340/GSM_BASED_TRACKING_SYSTEM_WITH GOOGLE_MAP_BASED_MONITORINGDESIGN_AND_DEVELOPMENT_OF_GPS-)

MihirGarude, NirmalHaldikar (2014).Department of Electronics Engineering, Datta Meghe College of Engineering, Airoli.VOLUME 4, ISSUE 9, SEPTEMBER 2014 <http://www.ijsrp.org/research-paper-0914/ijsrp-p3362.pdf>

B. P. S. Sahoo and Satyajit Rath (2012).Integrating GPS, GSM and Cellular Phone for Location Tracking and MonitoringCSIR-Institute of Minerals & Materials Technology, Bhubaneswar, IndiaProceedings of the International Conference on Geospatial Technologies and Applications, Geomatrix'12 February 26 – 29, 2012, IIT Bombay, Mumbai, India<https://arxiv.org/ftp/arxiv/papers/1307/1307.3147.pdf>

B.Suchitha Samuel, J.Mrudula (2013).Design of Intelligent solar Tracker Robot for surveillancePost Graduate Scholar, Department of Electronics & Communication Engineering, Geethanjali College of Engineering, Keesara, Andhra Pradesh, India1 Associate Professor, Department of Electronics& Communication Engineering, Geethanjali College of Engineering, Keesara, Andhra Pradesh, IndiaVol. 2, Issue 10, October 2013<https://www.ijareeie.com/upload/2013/october/8UDesign.pdf>

## **Local Studies**

P.S.Kiran Kumar, Dr. Shankaraiah (2014). INTELLIGENT STUDENTS TRACKING SYSTEM IN CAMPUS BASED ON RFID AND ZIGBEE (Assistant Professor in E & E Engineering, VVIET, Mysore, Karnataka, India)(Professor, Dept. of E & C Engineering, SJCE, Mysore, Karnataka, India)Volume 5, Issue 8, August (2014), pp. 117-126  
<http://www.iaeme.com/MasterAdmin/uploadfolder/INTELLIGENT%20STUDENTS%20TRACKING%20SYSTEM%20IN%20CAMPUS%20BASED%20ON%20RFID%20AND%20ZIGBEE/INTELLIGENT%20STUDENTS%20TRACKING%20SYSTEM%20IN%20CAMPUS%20BASED%20ON%20RFID%20AND%20ZIGBEE.pdf>

Laurie Thomas Lee, Ph.D. (2015). Location- based Communication Systems a Look at Intelligent Networking and Privacy Concerns College of Journalism and Mass Communications University of Nebraska-

Lincoln <http://www.globalmediajournal.com/open-access/locationbased-communication-systems-a-look-at-intelligent-networking-and-privacy-concerns.php?aid=35306>

Shermin Sultana, Asma Enayet and Ishrat Jahan Mouri (2015). A Smart Location Based Time and Attendance Tracking System Using Android ApplicationDepartment of Computer Science and Engineering, Stamford University Bangladesh, Dhaka, BangladeshVol. 5, No.1, February 2015<http://airccse.org/journal/ijcsein/papers/5115ijcsein01.pdf>

Komal Satish Agarwal, Kranti Dive (2014). GPS and RFID Based Intelligent Bus Management and Monitoring SystemInternational Journal of Engineering Research & TechnologyVol. 3 - Issue 7 (July - 2014)<http://www.ijert.org/view-pdf/10385/rfid-based-intelligent-bus-management-and-monitoring-system>

## **APPENDIX B**

### **Users Manual**

#### **USER'S MANUAL**

*USER'S MANUAL*

- ❖ Using the Online system “Intelligent IP Address Locator via Google Map”.

**STEPS:**

- Welcome to the Setup Wizard for MySQL Server 5.1. Click Next to continue.



- Select a setup type - Typical, Complete, and custom. Select Typical and click next.



- Ready to install MySQL.



- Installation in progress.



- More information about MySQL Enterprise Subscription.



- Setup Wizard Completed.

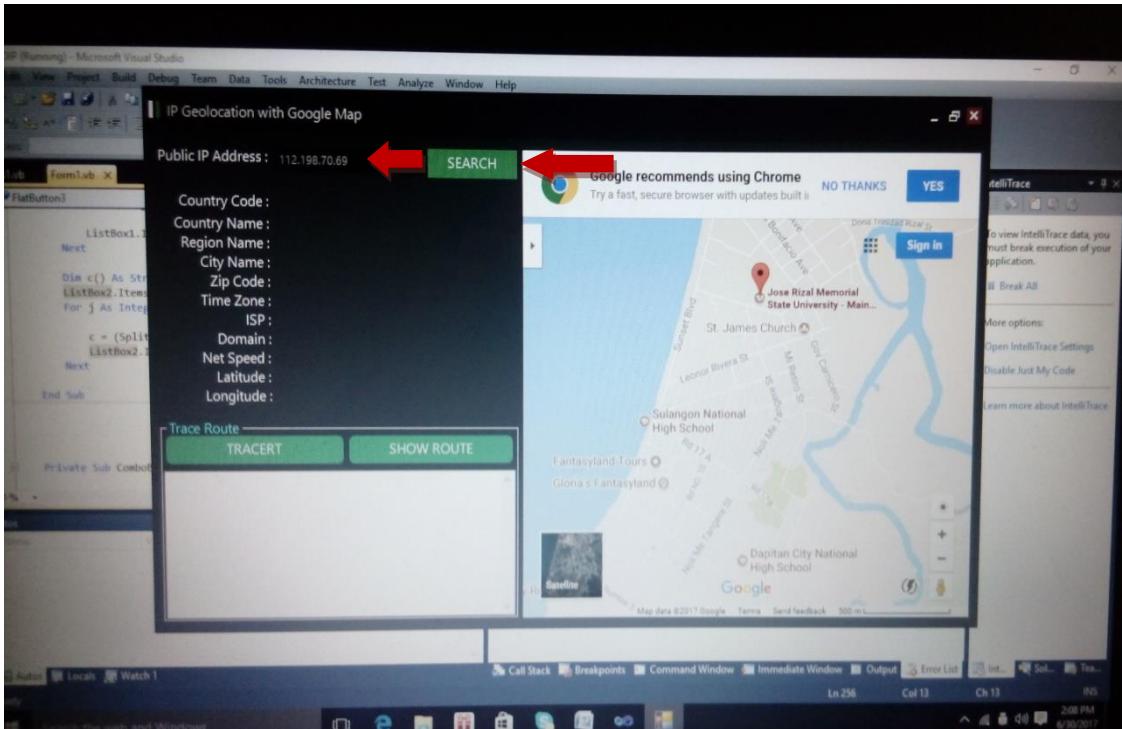


(Intelligent IP Address Locator via Google Maps)

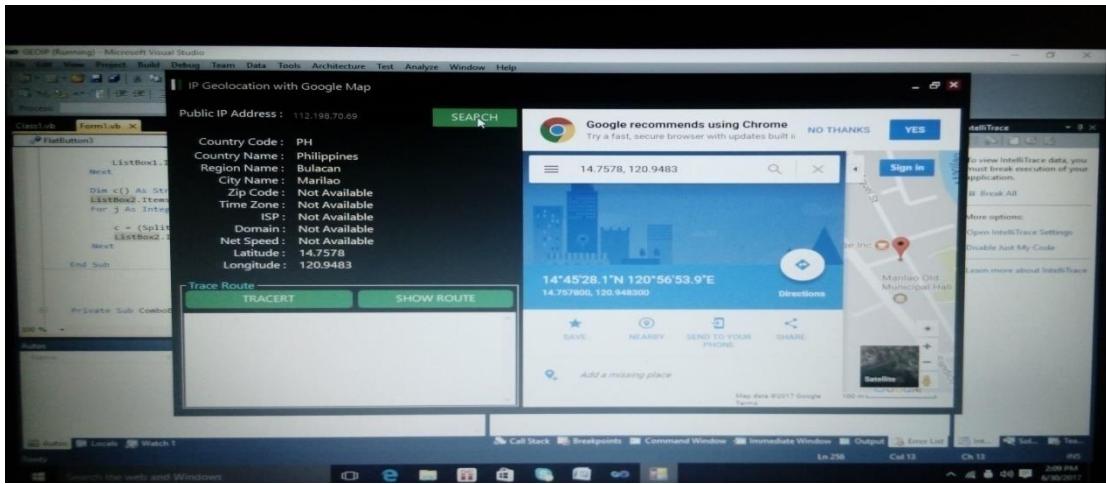
### Main menu of the system.

For Admin. The admin can view the menu bars like home, maps, IP address, and Location. Admin can search a location a certain person through IP Address. At first set-up the internet and once it done open the system.

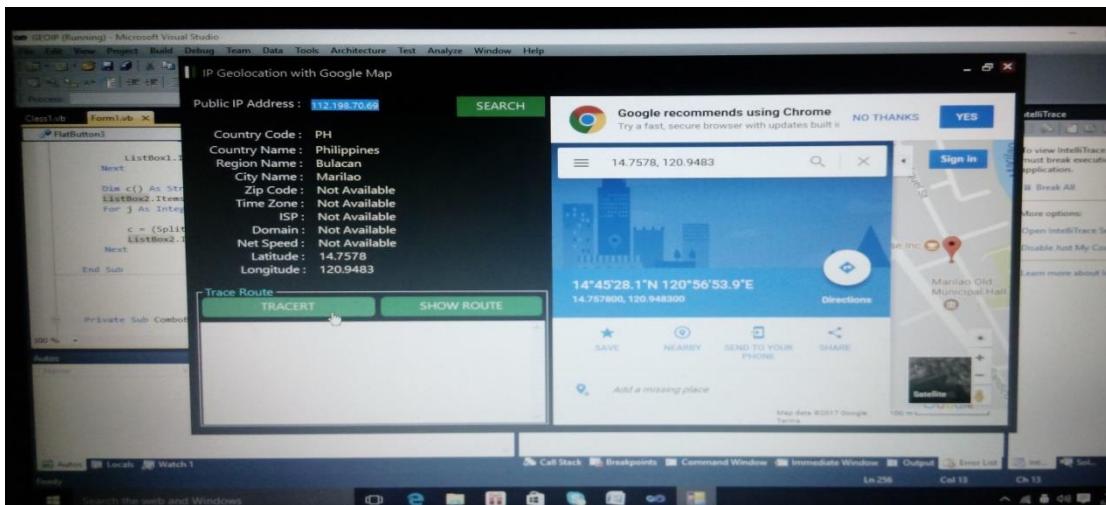
- Input IP Address and click Search.



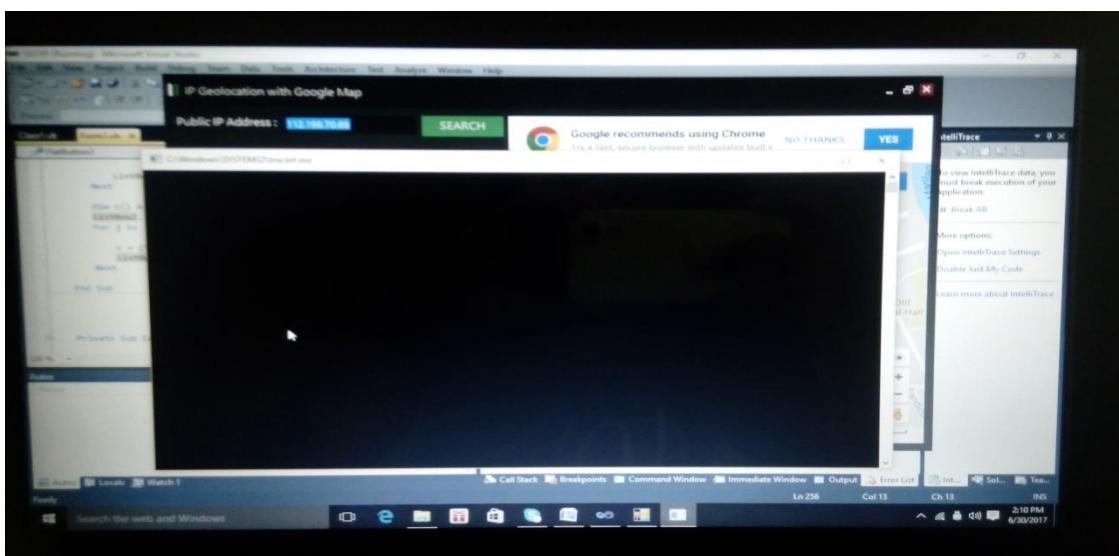
- Then you see the Location of a certain IP Address.



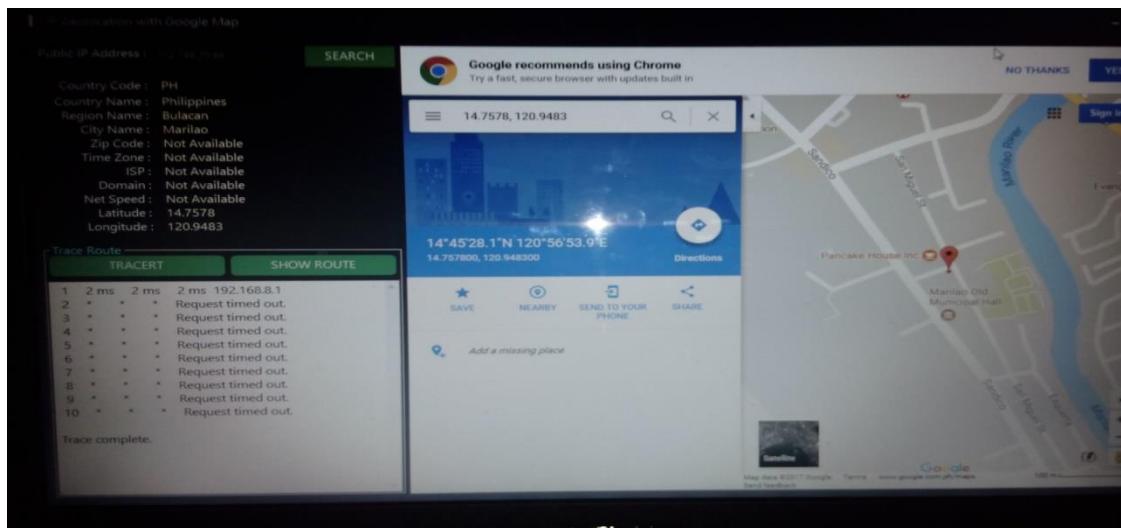
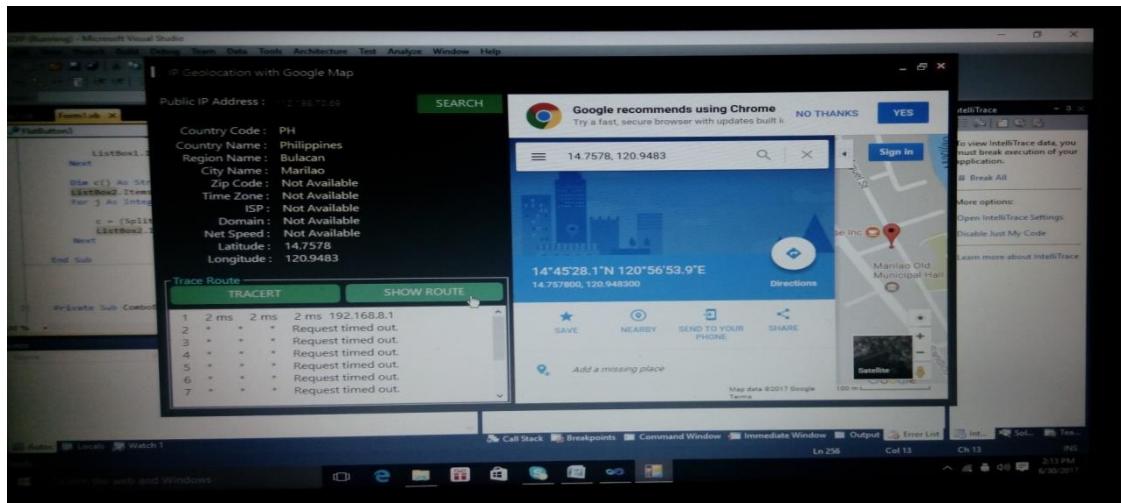
➤ Click the Tracert button.



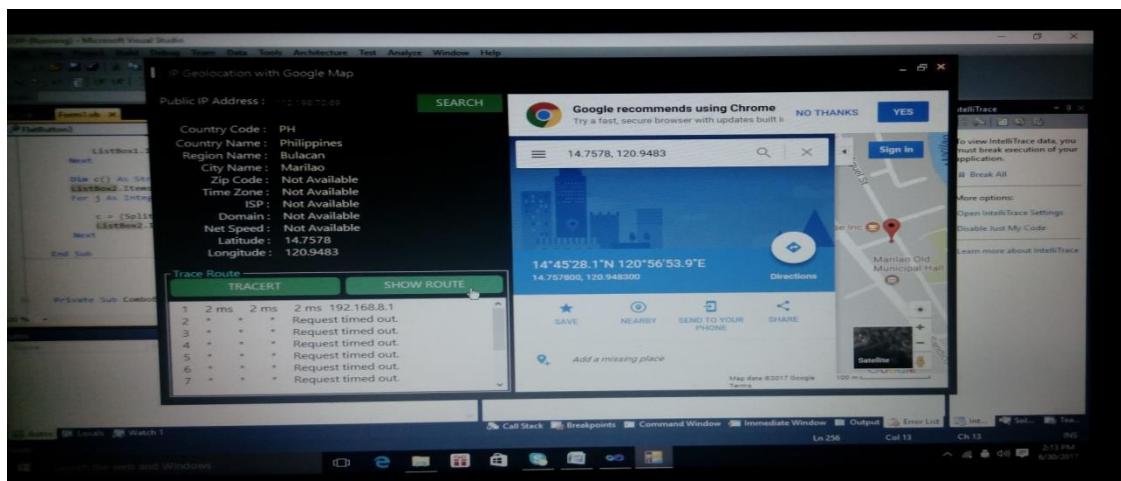
➤ The cmd prompt will pop up.



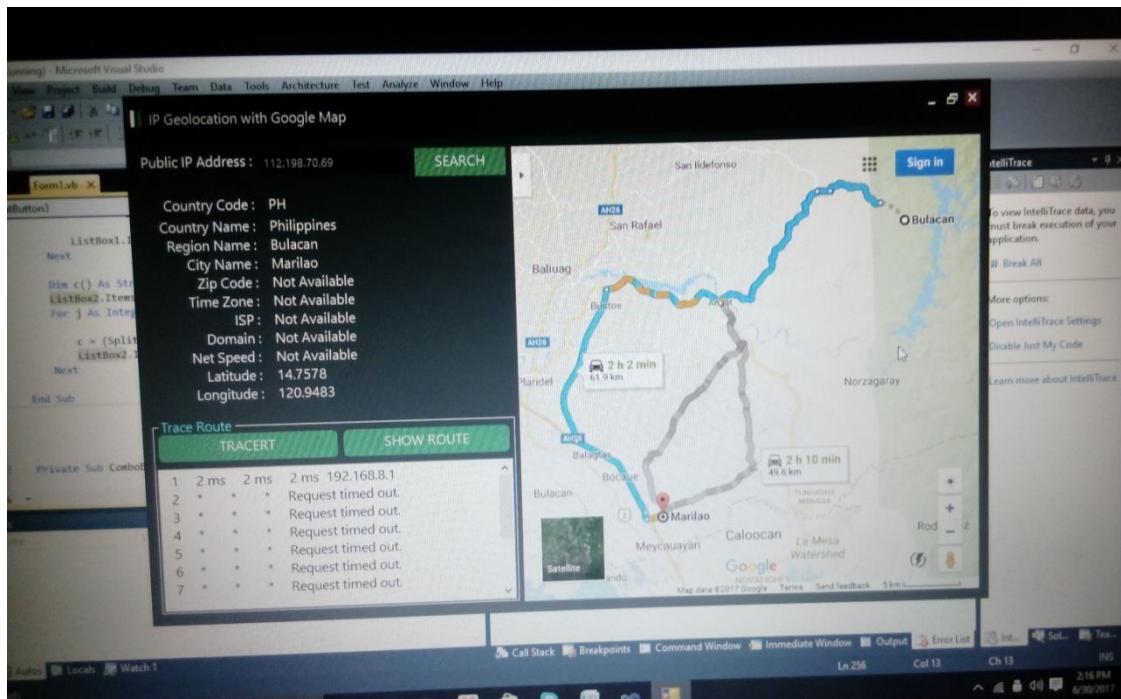
- The IP address was connected to this different connection that was converted IPs.



- After that click the button show route.



- And all show the route of the certain public IP address.



## **APPENDIX C**

### **Source Code**

```

Imports System.Xml
Imports System.Text
Imports System.Net
Imports System.Net.NetworkInformation
Imports System.Globalization

PublicClassForm1
Dim WithEvents pingSender AsNewPing()
Dim PINGERROR AsBoolean = False

PrivateConst InternetExplorerRootKeyAsString = "Software\Microsoft\Internet
Explorer"

PrivateConst BrowserEmulationKeyAsString = InternetExplorerRootKey +
"\Main\FeatureControl\FEATURE_BROWSER_EMULATION"

PublicSub SUBPING()
Try
    pingSender.SendAsync(TextBoxTRACEIP.Text, Nothing)
Catch ex AsException
    MsgBox(ex.Message)
EndTry

EndSub

PublicSub TRAZADO()
Dim IP AsIPAddress
Try
    IP = Dns.GetHostEntry(TextBoxTRACEIP.Text).AddressList(0)
Catch ex AsException
    MsgBox(ex.Message)
EndTry
Dim pingOptions AsNewPingOptions()
    pingOptions.Ttl = 1
Dim maxHops AsInteger = 30
Dim stopWatch AsNewStopwatch()

'FlatButton2.Enabled = False

For i AsInteger = 1 To maxHops

    stopWatch.Reset()
    stopWatch.Start()

Dim pingReply AsPingReply = pingSender.Send(IP, 5000, NewByte(31) {}, pingOptions)
    stopWatch.[Stop]()
Try

If PingReply.Address IsNotNothingThen

    TextBoxTRACE.Text = TextBoxTRACE.Text & i &"..." &
stopWatch.ElapsedMilliseconds.ToString &" ms. "& pingReply.Address.ToString &
vbCrLf
        ListBox1.Items.Add(pingReply.Address.ToString)
Else

    TextBoxTRACE.Text = TextBoxTRACE.Text &"Success"& vbCrLf
Exit For

```

```

EndIf

Catch ex AsException
    MsgBox(ex.Message)
EndTry

If pingReply.Status = IPStatus.Success Then
    TextBoxTRACE.Text = TextBoxTRACE.Text &"Trace Complete"& vbCrLf

Exit For
EndIf
    pingOptions.Ttl += 1
Next
    FlatButton2.Enabled = True
EndSub

PrivateSub geo()
Dim xmldoc AsNewXmlDocument
Dim xmlnode AsXmlNodeList
Dim i AsInteger

'xmldoc.Load("http://api.ip2location.com/?ip=" + FlatTextBox1.Text +
"&key=demo&package=WS24&format=xml")
xmldoc.Load("http://www.geoplugin.net/xml.gp?ip=" + FlatTextBox1.Text)
xmlnode = xmldoc.GetElementsByTagName("geoPlugin")

For i = 0 To xmlnode.Count - 1
    xmlnode(i).ChildNodes.Item(0).InnerText.Trim()

    lbl11.Text = xmlnode(i).ChildNodes.Item(7).InnerText.Trim()
    lbl12.Text = xmlnode(i).ChildNodes.Item(8).InnerText.Trim()
    lbl13.Text = xmlnode(i).ChildNodes.Item(4).InnerText.Trim()
    lbl14.Text = xmlnode(i).ChildNodes.Item(3).InnerText.Trim()
    lbl15.Text = "Not
Available'">xmlnode(i).ChildNodes.Item(6).InnerText.Trim()
    lbl16.Text = "Not Available"
    xmlnode(i).ChildNodes.Item(7).InnerText.Trim()
    lbl17.Text = "Not
Available'">xmlnode(i).ChildNodes.Item(8).InnerText.Trim()
    lbl18.Text = "Not
Available'">xmlnode(i).ChildNodes.Item(9).InnerText.Trim()
    lbl19.Text = "Not
Available'">xmlnode(i).ChildNodes.Item(10).InnerText.Trim()
    lbl110.Text = xmlnode(i).ChildNodes.Item(10).InnerText.Trim()
    lbl111.Text = xmlnode(i).ChildNodes.Item(11).InnerText.Trim()

Next
EndSub

PrivateSub FlatButton1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs)
EndSub

PrivateSub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles MyBase.Load

```

```

Try
Dim queryAddress As New StringBuilder

queryAddress.Append("https://www.google.com.ph/maps/place/Jose+Rizal+Memorial+Stat
e+University+-+
+Main+Campus/@8.6516083,123.4220903,15.25z/data=!4m!3m4!1s0x0:0xfc69ea2ff39b5bfd!
8m2!3d8.6559168!4d123.4226257?hl=en")
' queryAddress.Append("https://www.google.com/maps/dir/Current+Location/43.12345,-
76.12345")
    WebBrowser1.Navigate(queryAddress.ToString())
Catch ex As Exception
MessageBox.Show("Unable to Retrive Data")
EndTry
EndSub

Private Sub FlatButton1_Click_1(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles FlatButton1.Click
    geo()

Dim lat As String = lbl10.Text
Dim lon As String = lbl11.Text

Try
Dim queryAddress As New StringBuilder

    queryAddress.Append("http://maps.google.com/maps?q=")
If lbl10.Text <> String.Empty Then
        queryAddress.Append(lat + "," & "+")
EndIf
If lbl11.Text <> String.Empty Then
        queryAddress.Append(lon)
EndIf

    WebBrowser1.Navigate(queryAddress.ToString())
Catch ex As Exception
MessageBox.Show("Unable to Retrive Data")
EndTry
EndSub

Private Sub FlatButton2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles FlatButton2.Click
    FlatButton2.Enabled = False

'ListBox1.Items.Clear()
'PINGERROR = False
'TextBoxTRACE.Clear()
'TextBoxTRACEIP.Text = FlatTextBox1.Text

'TextBoxTRACEIP.SelectAll()

'If TextBoxTRACEIP.Text.Length <> 0 Then
'    SUBPING()
'    pingSender.SendAsyncCancel()
'    If PINGERROR = False Then
'        Try
'            TRAZADO()
'        Catch ex As Exception
'            MsgBox(ex.Message)
'        EndTry
'    EndIf
'EndIf
EndSub

```

```

        End Try

    Else
        TextBoxTRACE.Text = "Cant Perform Right Now"
    End If

'Else
'    MessageBox.Show("Pls. Fill up the fields")
'End If

If ListBox1.Items.Count <> 0 Then
    ListBox3.Items.Clear()

For i AsInteger = 1 To ListBox2.Items.Count - 1

If ListBox2.Items.Item(i) <>"timed"Then

    FlatTextBox1.Text = ""
    FlatTextBox1.Text = ListBox2.Items.Item(i)
    geo()
    ListBox3.Items.Add(lbl10.Text &, "& lbl11.Text)

EndIf

Next

' WebBrowser1.Navigate("https://www.google.com/maps/dir/ 8.655907,
123.423357/8.58944, 123.34139/14.08776, 120.97098/" & lbl10.Text & ", " &
lbl11.Text)

Dim query AsNewStringBuilder
'http://maps.google.com/?saddr=34.052222,-118.243611&daddr=37.322778,-122.031944

        query.Append("https://www.google.com.ph/maps/dir/")
'query.Append("http://maps.google.com/?saddr=")

For j AsInteger = 0 To ListBox3.Items.Count - 1

        query.Append(ListBox3.Items.Item(j) &"/")
' query.Append(j & "&daddr=")

Next
'query.Append("@12.4589865,120.2933587,7z/")
        WebBrowser1.Navigate(query.ToString())
EndIf

FlatButton2.Enabled = True

EndSub

PrivateSub FlatButton3_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles FlatButton3.Click
Dim oProcess AsNewProcess()
'Dim oStartInfo As New ProcessStartInfo("tracert.exe", "-d -h 10 117.92.174.86")
Dim oStartInfo AsNewProcessStartInfo("tracert.exe", "-d -h 10 "&
FlatTextBox1.Text)
        oStartInfo.UseShellExecute = False
        oStartInfo.RedirectStandardOutput = True
        oProcess.StartInfo = oStartInfo

```

```

oProcess.Start()

Dim sOutput AsString
Using oStreamReader As System.IO.StreamReader = oProcess.StandardOutput
    sOutput = oStreamReader.ReadToEnd()
EndUsing
    TextBoxTRACE.Text = sOutput


Dim newlist AsList(OfString) = TextBoxTRACE.Lines.ToList()

    newlist.RemoveAt(newlist.Count - 1)
    newlist.RemoveAt(0)
    newlist.RemoveAt(0)
    newlist.RemoveAt(0)
    TextBoxTRACE.Lines = newlist.ToArray()

Dim b AsString() = Split(TextBoxTRACE.Text, vbNewLine)

For i AsInteger = 0 To b.Length - 1
    ListBox1.Items.Add(b(i))
Next

Dim c() AsString
    ListBox2.Items.Clear()
For j AsInteger = 0 To ListBox1.Items.Count - 3

    c = (Split(ListBox1.Items.Item(j), " "))
    ListBox2.Items.Add(c(c.Length - 2))
Next

EndSub

PrivateSub ComboBox1_SelectedIndexChanged(ByVal sender As System.Object, ByVal e As System.EventArgs)

EndSub

PrivateSub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
Dim urls AsString
    urls = WebBrowser1.Url.ToString()
    ComboBox1.Text = urls
EndSub
EndClass

Imports System.Drawing.Drawing2D, System.ComponentModel, System.Windows.Forms

''' <summary>
''' Flat UI Theme
''' Creator: iSynthesis (HF)
''' Version: 1.0.4
''' Date Created: 17/06/2013
''' Date Changed: 26/06/2013
''' UID: 374648
''' For any bugs / errors, PM me.

```

```

''' </summary>
''' <remarks></remarks>

ModuleHelpers

#Region" Variables"
Friend G AsGraphics, B AsBitmap
Friend _FlatColor AsColor = Color.FromArgb(35, 168, 109)
Friend NearSF AsNewStringFormat() With {.Alignment = StringAlignment.Near,
.LineAlignment = StringAlignment.Near}
Friend CenterSF AsNewStringFormat() With {.Alignment = StringAlignment.Center,
.LineAlignment = StringAlignment.Center}
#EndRegion

#Region" Functions"

PublicFunction RoundRec(ByVal Rectangle AsRectangle, ByVal Curve AsInteger)
AsGraphicsPath
Dim P As GraphicsPath = New GraphicsPath()
Dim ArcRectangleWidth AsInteger = Curve * 2
    P.AddArc(New Rectangle(Rectangle.X, Rectangle.Y, ArcRectangleWidth,
ArcRectangleWidth), -180, 90)
    P.AddArc(New Rectangle(Rectangle.Width - ArcRectangleWidth + Rectangle.X,
Rectangle.Y, ArcRectangleWidth, ArcRectangleWidth), -90, 90)
    P.AddArc(New Rectangle(Rectangle.Width - ArcRectangleWidth + Rectangle.X,
Rectangle.Height - ArcRectangleWidth + Rectangle.Y, ArcRectangleWidth,
ArcRectangleWidth), 0, 90)
    P.AddArc(New Rectangle(Rectangle.X, Rectangle.Height - ArcRectangleWidth +
Rectangle.Y, ArcRectangleWidth, ArcRectangleWidth), 90, 90)
    P.AddLine(New Point(Rectangle.X, Rectangle.Height - ArcRectangleWidth +
Rectangle.Y), New Point(Rectangle.X, Curve + Rectangle.Y))
Return P
EndFunction

PublicFunction RoundRect(ByVal x!, ByVal y!, ByVal w!, ByVal h!, OptionalByVal r!
= 0.3, OptionalByVal TL AsBoolean = True, OptionalByVal TR AsBoolean = True,
OptionalByVal BR AsBoolean = True, OptionalByVal BL AsBoolean = True)
AsGraphicsPath
Dim d! = Math.Min(w, h) * r, xw = x + w, yh = y + h
    RoundRect = New GraphicsPath

    With RoundRect
        If TL Then .AddArc(x, y, d, d, 180, 90) Else .AddLine(x, y, x, y)
        If TR Then .AddArc(xw - d, y, d, d, 270, 90) Else .AddLine(xw, y, xw, y)
        If BR Then .AddArc(xw - d, yh - d, d, d, 0, 90) Else .AddLine(xw, yh, xw, yh)
        If BL Then .AddArc(x, yh - d, d, d, 90, 90) Else .AddLine(x, yh, x, yh)

        .CloseFigure()
    EndWith
EndFunction

'-- Credit: AeonHack
PublicFunction DrawArrow(ByVal x AsInteger, ByVal y AsInteger, ByVal flip
AsBoolean) AsGraphicsPath
Dim GP AsNew GraphicsPath()

Dim W AsInteger = 12
Dim H AsInteger = 6

If flip Then
    GP.AddLine(x + 1, y, x + W + 1, y)
    GP.AddLine(x + W, y, x + H, y + H - 1)

```

```

Else
    GP.AddLine(x, y + H, x + W, y + H)
    GP.AddLine(x + W, y + H, x + H, y)
EndIf

    GP.CloseFigure()
Return GP
EndFunction

#EndRegion

EndModule

#Region" Mouse States"

EnumMouseStateAsByte
    None = 0
    Over = 1
    Down = 2
    Block = 3
EndEnum

#EndRegion

ClassFormSkin : InheritsContainerControl

#Region" Variables"

Private W, H AsInteger
Private Cap AsBoolean = False
Private _HeaderMaximize AsBoolean = False
Private MousePoint AsNewPoint(0, 0)
Private MoveHeight = 50

#EndRegion

#Region" Properties"

#Region" Colors"

<Category("Colors")>
PublicProperty HeaderColor() AsColor
Get
Return _HeaderColor
EndGet
Set(ByVal value As Color)
    _HeaderColor = value
EndSet
EndProperty
<Category("Colors")>
PublicProperty BaseColor() AsColor
Get
Return _BaseColor
EndGet
Set(ByVal value As Color)
    _BaseColor = value
EndSet
EndProperty
<Category("Colors")>
PublicProperty BorderColor() AsColor
Get
Return _BorderColor
EndGet
Set(ByVal value As Color)
    _BorderColor = value
EndSet
EndProperty

```

```

EndGet
Set(ByVal value As Color)
    _BorderColor = value
EndSet
EndProperty
<Category("Colors")>
PublicProperty FlatColor() AsColor
Get
Return _FlatColor
EndGet
Set(ByVal value As Color)
    _FlatColor = value
EndSet
EndProperty

#EndRegion

#Region" Options"

<Category("Options")> _
PublicProperty HeaderMaximize AsBoolean
Get
Return _HeaderMaximize
EndGet
Set(ByVal value AsBoolean)
    _HeaderMaximize = value
EndSet
EndProperty

#EndRegion

ProtectedOverridesSub OnMouseDown(ByVal e As MouseEventArgs)
 MyBase.OnMouseDown(e)
 If e.Button = Windows.Forms.MouseButtons.Left AndNew Rectangle(0, 0, Width,
 MoveHeight).Contains(e.Location) Then
     Cap = True
     MousePoint = e.Location
 EndIf
EndSub

PrivateSub FormSkin_MouseDoubleClick(ByVal sender AsObject, ByVal e
 AsEventArgs) HandlesMe.MouseDoubleClick
 If HeaderMaximize Then
     If e.Button = Windows.Forms.MouseButtons.Left AndNew Rectangle(0, 0, Width,
 MoveHeight).Contains(e.Location) Then
         If FindForm.WindowState = FormWindowState.Normal Then
             FindForm.WindowState = FormWindowState.Maximized :
             FindForm.Refresh()
         ElseIf FindForm.WindowState = FormWindowState.Maximized Then
             FindForm.WindowState = FormWindowState.Normal :
             FindForm.Refresh()
         EndIf
     EndIf
     EndIf
 EndSub

ProtectedOverridesSub OnMouseUp(ByVal e As MouseEventArgs)
 MyBase.OnMouseUp(e) : Cap = False
EndSub

ProtectedOverridesSub OnMouseMove(ByVal e As MouseEventArgs)
 MyBase.OnMouseMove(e)

```

```

If Cap Then
    Parent.Location = MousePosition - MousePoint
EndIf
EndSub

ProtectedOverridesSub OnCreateControl()
MyBase.OnCreateControl()
    ParentForm.FormBorderStyle = FormBorderStyle.None
    ParentForm.AllowTransparency = False
    ParentForm.TransparencyKey = Color.Fuchsia
    ParentForm.FindForm.StartPosition = FormStartPosition.CenterScreen
    Dock = DockStyle.Fill
    Invalidate()
EndSub

#EndRegion

#Region" Colors"
#Region" Dark Colors"

Private _HeaderColor AsColor = Color.FromArgb(45, 47, 49)
Private _BaseColor AsColor = Color.FromArgb(60, 70, 73)
Private _BorderColor AsColor = Color.FromArgb(53, 58, 60)
Private TextColor AsColor = Color.FromArgb(234, 234, 234)

#EndRegion

#Region" Light Colors"

Private _HeaderLight AsColor = Color.FromArgb(171, 171, 172)
Private _BaseLight AsColor = Color.FromArgb(196, 199, 200)
Public TextLight AsColor = Color.FromArgb(45, 47, 49)

#EndRegion

#EndRegion

SubNew()
    SetStyle(ControlStyles.AllPaintingInWmPaint Or ControlStyles.UserPaint Or
    -
        ControlStyles.ResizeRedraw Or
    ControlStyles.OptimizedDoubleBuffer, True)
    DoubleBuffered = True
    BackColor = Color.White
    Font = New Font("Segoe UI", 12)
EndSub

ProtectedOverridesSub OnPaint(ByVal e AsPaintEventArgs)
    B = New Bitmap(Width, Height) : G = Graphics.FromImage(B)
    W = Width : H = Height

Dim Base AsNew Rectangle(0, 0, W, H), Header AsNew Rectangle(0, 0, W, 50)

With G
    .SmoothingMode = 2
    .PixelOffsetMode = 2
    .TextRenderingHint = 5
    .Clear(BackColor)

'-- Base
    .FillRectangle(New SolidBrush(_BaseColor), Base)

```

```

'-- Header
    .FillRectangle(New SolidBrush(_HeaderColor), Header)

'-- Logo
    .FillRectangle(New SolidBrush(Color.FromArgb(243, 243, 243)), New
    Rectangle(8, 16, 4, 18))
    .FillRectangle(New SolidBrush(_FlatColor), 16, 16, 4, 18)
    .DrawString(Text, Font, New SolidBrush(TextColor), New Rectangle(26,
    15, W, H), NearSF)

'-- Border
    .DrawRectangle(New Pen(_BorderColor), Base)
EndWith

 MyBase.OnPaint(e)
    G.Dispose()
    e.Graphics.InterpolationMode = 7
    e.Graphics.DrawImageUnscaled(B, 0, 0)
    B.Dispose()
EndSub
EndClass

ClassFlatClose : InheritsControl

#Region" Variables"

Private State AsMouseState = MouseState.None
Private x AsInteger

#EndRegion

#Region" Properties"

#Region" Mouse States"

ProtectedOverridesSub OnMouseEnter(ByVal e AsEventArgs)
MyBase.OnMouseEnter(e)
    State = MouseState.Over : Invalidate()
EndSub
ProtectedOverridesSub OnMouseDown(ByVal e AsMouseEventArgs)
MyBase.OnMouseDown(e)
    State = MouseState.Down : Invalidate()
EndSub
ProtectedOverridesSub OnMouseLeave(ByVal e AsEventArgs)
MyBase.OnMouseLeave(e)
    State = MouseState.None : Invalidate()
EndSub
ProtectedOverridesSub OnMouseUp(ByVal e AsMouseEventArgs)
MyBase.OnMouseUp(e)
    State = MouseState.Over : Invalidate()
EndSub
ProtectedOverridesSub OnMouseMove(ByVal e AsMouseEventArgs)
MyBase.OnMouseMove(e)
    x = e.X : Invalidate()
EndSub

ProtectedOverridesSub OnClick(ByVal e AsEventArgs)
MyBase.OnClick(e)
Environment.Exit(0)
EndSub

```

```

#EndRegion

ProtectedOverridesSub OnResize(ByVal e AsEventArgs)
 MyBase.OnResize(e)
    Size = New Size(18, 18)
EndSub

#Region" Colors"

<Category("Colors")> _
PublicProperty BaseColor AsColor
Get
Return _BaseColor
EndGet
Set(ByVal value As Color)
    _BaseColor = value
EndSet
EndProperty

<Category("Colors")> _
PublicProperty TextColor AsColor
Get
Return _TextColor
EndGet
Set(ByVal value As Color)
    _TextColor = value
EndSet
EndProperty

#EndRegion

#EndRegion

#Region" Colors"

Private _BaseColor AsColor = Color.FromArgb(168, 35, 35)
Private _TextColor AsColor = Color.FromArgb(243, 243, 243)

#EndRegion

SubNew()
    SetStyle(ControlStyles.AllPaintingInWmPaint Or ControlStyles.UserPaint Or
    -
        ControlStyles.ResizeRedraw Or
    ControlStyles.OptimizedDoubleBuffer, True)
        DoubleBuffered = True
        BackColor = Color.White
        Size = New Size(18, 18)
        Anchor = AnchorStyles.Top Or AnchorStyles.Right
        Font = New Font("Marlett", 10)
EndSub

ProtectedOverridesSub OnPaint(ByVal e AsPaintEventArgs)
Dim B AsNew Bitmap(Width, Height)
Dim G As Graphics = Graphics.FromImage(B)

Dim Base AsNew Rectangle(0, 0, Width, Height)

With G
    .SmoothingMode = 2
    .PixelOffsetMode = 2
    .TextRenderingHint = 5

```

```

        .Clear(BackColor)

'-- Base
        .FillRectangle(New SolidBrush(_BaseColor), Base)

'-- X
        .DrawString("r", Font, New SolidBrush(TextColor), New Rectangle(0, 0,
Width, Height), CenterSF)

'-- Hover/down
SelectCase State
Case MouseState.Over
        .FillRectangle(New SolidBrush(Color.FromArgb(30,
Color.White)), Base)
Case MouseState.Down
        .FillRectangle(New SolidBrush(Color.FromArgb(30,
Color.Black)), Base)
EndSelect
EndWith

 MyBase.OnPaint(e)
    G.Dispose()
    e.Graphics.InterpolationMode = 7
    e.Graphics.DrawImageUnscaled(B, 0, 0)
    B.Dispose()
EndSub
EndClass

ClassFlatMax : InheritsControl

#Region" Variables"

Private State AsMouseState = MouseState.None
Private x AsInteger

#EndRegion

#Region" Properties"

#Region" Mouse States"

ProtectedOverridesSub OnMouseEnter(ByVal e AsEventArgs)
MyBase.OnMouseEnter(e)
    State = MouseState.Over : Invalidate()
EndSub
ProtectedOverridesSub OnMouseDown(ByVal e AsMouseEventArgs)
MyBase.OnMouseDown(e)
    State = MouseState.Down : Invalidate()
EndSub
ProtectedOverridesSub OnMouseLeave(ByVal e AsEventArgs)
MyBase.OnMouseLeave(e)
    State = MouseState.None : Invalidate()
EndSub
ProtectedOverridesSub OnMouseUp(ByVal e AsMouseEventArgs)
MyBase.OnMouseUp(e)
    State = MouseState.Over : Invalidate()
EndSub
ProtectedOverridesSub OnMouseMove(ByVal e AsMouseEventArgs)
MyBase.OnMouseMove(e)
    x = e.X : Invalidate()
EndSub

```

```

ProtectedOverridesSub OnClick(ByVal e AsEventArgs)
 MyBase.OnClick(e)
 SelectCase FindForm.WindowState
 Case FormWindowState.Maximized
     FindForm.WindowState = FormWindowState.Normal
 Case FormWindowState.Normal
     FindForm.WindowState = FormWindowState.Maximized
 EndSelect
 EndSub

#EndRegion

#Region" Colors"

<Category("Colors")> _
PublicProperty BaseColor AsColor
Get
Return _BaseColor
EndGet
Set(ByVal value As Color)
    _BaseColor = value
EndSet
EndProperty

<Category("Colors")> _
PublicProperty TextColor AsColor
Get
Return _TextColor
EndGet
Set(ByVal value As Color)
    _TextColor = value
EndSet
EndProperty

#EndRegion

ProtectedOverridesSub OnResize(ByVal e AsEventArgs)
 MyBase.OnResize(e)
     Size = New Size(18, 18)
EndSub

#EndRegion

#Region" Colors"

Private _BaseColor AsColor = Color.FromArgb(45, 47, 49)
Private _TextColor AsColor = Color.FromArgb(243, 243, 243)

#EndRegion

SubNew()
    SetStyle(ControlStyles.AllPaintingInWmPaint Or ControlStyles.UserPaint Or
    -
        ControlStyles.ResizeRedraw Or
ControlStyles.OptimizedDoubleBuffer, True)
    DoubleBuffered = True
    BackColor = Color.White
    Size = New Size(18, 18)
    Anchor = AnchorStyles.Top Or AnchorStyles.Right
    Font = New Font("Marlett", 12)
EndSub

```

```

ProtectedOverridesSub OnPaint(ByVal e AsPaintEventArgs)
Dim B AsNew Bitmap(Width, Height)
Dim G As Graphics = Graphics.FromImage(B)

Dim Base AsNew Rectangle(0, 0, Width, Height)

With G
    .SmoothingMode = 2
    .PixelOffsetMode = 2
    .TextRenderingHint = 5
    .Clear(BackColor)

    '-- Base
    .FillRectangle(New SolidBrush(_BaseColor), Base)

    '-- Maximize
    If FindForm.WindowState = FormWindowState.Maximized Then
        .DrawString("1", Font, New SolidBrush(TextColor), New Rectangle(1,
1, Width, Height), CenterSF)
    ElseIf FindForm.WindowState = FormWindowState.Normal Then
        .DrawString("2", Font, New SolidBrush(TextColor), New Rectangle(1,
1, Width, Height), CenterSF)
    EndIf

    '-- Hover/down
    SelectCase State
    Case MouseState.Over
        .FillRectangle(New SolidBrush(Color.FromArgb(30,
Color.White)), Base)
    Case MouseState.Down
        .FillRectangle(New SolidBrush(Color.FromArgb(30,
Color.Black)), Base)
    EndSelect
    EndWith

 MyBase.OnPaint(e)
    G.Dispose()
    e.Graphics.InterpolationMode = 7
    e.Graphics.DrawImageUnscaled(B, 0, 0)
    B.Dispose()
EndSub
EndClass

ClassFlatMini : InheritsControl

#Region" Variables"

Private State AsMouseState = MouseState.None
Private x AsInteger

#EndRegion

#Region" Properties"

#Region" Mouse States"

ProtectedOverridesSub OnMouseEnter(ByVal e AsEventArgs)
MyBase.OnMouseEnter(e)
    State = MouseState.Over : Invalidate()
EndSub
ProtectedOverridesSub OnMouseDown(ByVal e AsMouseEventArgs)
MyBase.OnMouseDown(e)

```

```

        State = MouseState.Down : Invalidate()
EndSub
ProtectedOverridesSub OnMouseLeave(ByVal e AsEventArgs)
 MyBase.OnMouseLeave(e)
 State = MouseState.None : Invalidate()
EndSub
ProtectedOverridesSub OnMouseUp(ByVal e AsMouseEventArgs)
 MyBase.OnMouseUp(e)
 State = MouseState.Over : Invalidate()
EndSub
ProtectedOverridesSub OnMouseMove(ByVal e AsMouseEventArgs)
 MyBase.OnMouseMove(e)
 x = e.X : Invalidate()
EndSub

ProtectedOverridesSub OnClick(ByVal e AsEventArgs)
 MyBase.OnClick(e)
SelectCase FindForm.WindowState
Case FormWindowState.Normal
    FindForm.WindowState = FormWindowState.Minimized
Case FormWindowState.Maximized
    FindForm.WindowState = FormWindowState.Minimized
EndSelect
EndSub

#EndRegion

#Region" Colors"

<Category("Colors")> _
PublicProperty BaseColor AsColor
Get
Return _BaseColor
EndGet
Set(ByVal value As Color)
    _BaseColor = value
EndSet
EndProperty

<Category("Colors")> _
PublicProperty TextColor AsColor
Get
Return _TextColor
EndGet
Set(ByVal value As Color)
    _TextColor = value
EndSet
EndProperty

#EndRegion

ProtectedOverridesSub OnResize(ByVal e AsEventArgs)
 MyBase.OnResize(e)
 Size = New Size(18, 18)
EndSub

#EndRegion

#Region" Colors"

Private _BaseColor AsColor = Color.FromArgb(45, 47, 49)
Private _TextColor AsColor = Color.FromArgb(243, 243, 243)

```

```

#EndRegion

SubNew()
    SetStyle(ControlStyles.AllPaintingInWmPaint Or ControlStyles.UserPaint Or
    -
        ControlStyles.ResizeRedraw Or
    ControlStyles.OptimizedDoubleBuffer, True)
        DoubleBuffered = True
        BackColor = Color.White
        Size = New Size(18, 18)
        Anchor = AnchorStyles.Top Or AnchorStyles.Right
        Font = New Font("Marlett", 12)
EndSub

ProtectedOverridesSub OnPaint(ByVal e AsPaintEventArgs)
Dim B AsNew Bitmap(Width, Height)
Dim G As Graphics = Graphics.FromImage(B)

Dim Base AsNew Rectangle(0, 0, Width, Height)

With G
    .SmoothingMode = 2
    .PixelOffsetMode = 2
    .TextRenderingHint = 5
    .Clear(BackColor)

'-- Base
    .FillRectangle(New SolidBrush(_BaseColor), Base)

'-- Minimize
    .DrawString("0", Font, New SolidBrush(TextColor), New Rectangle(2, 1,
Width, Height), CenterSF)

'-- Hover/down
SelectCase State
Case MouseState.Over
    .FillRectangle(New SolidBrush(Color.FromArgb(30,
Color.White)), Base)
Case MouseState.Down
    .FillRectangle(New SolidBrush(Color.FromArgb(30,
Color.Black)), Base)
EndSelect
EndWith

 MyBase.OnPaint(e)
    G.Dispose()
    e.Graphics.InterpolationMode = 7
    e.Graphics.DrawImageUnscaled(B, 0, 0)
    B.Dispose()
EndSub
EndClass

ClassFlatColorPalette : InheritsControl

#Region" Variables"

Private W, H AsInteger

#EndRegion

#Region" Properties"

```

```

ProtectedOverridesSub OnResize(ByVal e AsEventArgs)
 MyBase.OnResize(e)
    Width = 180
    Height = 80
EndSub

#Region" Colors"

<Category("Colors")> _
PublicProperty Red AsColor
Get
Return _Red
EndGet
Set(ByVal value As Color)
    _Red = value
EndSet
EndProperty

<Category("Colors")> _
PublicProperty Cyan AsColor
Get
Return _Cyan
EndGet
Set(ByVal value As Color)
    _Cyan = value
EndSet
EndProperty

<Category("Colors")> _
PublicProperty Blue AsColor
Get
Return _Blue
EndGet
Set(ByVal value As Color)
    _Blue = value
EndSet
EndProperty

<Category("Colors")> _
PublicProperty LimeGreen AsColor
Get
Return _LimeGreen
EndGet
Set(ByVal value As Color)
    _LimeGreen = value
EndSet
EndProperty

<Category("Colors")> _
PublicProperty Orange AsColor
Get
Return _Orange
EndGet
Set(ByVal value As Color)
    _Orange = value
EndSet
EndProperty

<Category("Colors")> _
PublicProperty Purple AsColor
Get

```

```

    Return _Purple
EndGet
Set(ByVal value As Color)
    _Purple = value
EndSet
EndProperty

<Category("Colors")> _
PublicProperty Black AsColor
Get
Return _Black
EndGet
Set(ByVal value As Color)
    _Black = value
EndSet
EndProperty

<Category("Colors")> _
PublicProperty Gray AsColor
Get
Return _Gray
EndGet
Set(ByVal value As Color)
    _Gray = value
EndSet
EndProperty

<Category("Colors")> _
PublicProperty White AsColor
Get
Return _White
EndGet
Set(ByVal value As Color)
    _White = value
EndSet
EndProperty

#EndRegion

#EndRegion

#Region" Colors"

Private _Red AsColor = Color.FromArgb(220, 85, 96)
Private _Cyan AsColor = Color.FromArgb(10, 154, 157)
Private _Blue AsColor = Color.FromArgb(0, 128, 255)
Private _LimeGreen AsColor = Color.FromArgb(35, 168, 109)
Private _Orange AsColor = Color.FromArgb(253, 181, 63)
Private _Purple AsColor = Color.FromArgb(155, 88, 181)
Private _Black AsColor = Color.FromArgb(45, 47, 49)
Private _Gray AsColor = Color.FromArgb(63, 70, 73)
Private _White AsColor = Color.FromArgb(243, 243, 243)

#EndRegion

SubNew()
    SetStyle(ControlStyles.AllPaintingInWmPaint Or ControlStyles.UserPaint Or
    ControlStyles.OptimizedDoubleBuffer, True)
    DoubleBuffered = True
    BackColor = Color.FromArgb(60, 70, 73)

```

```

        Size = New Size(160, 80)
        Font = New Font("Segoe UI", 12)
EndSub

ProtectedOverridesSub OnPaint(ByVal e AsPaintEventArgs)
    B = New Bitmap(Width, Height) : G = Graphics.FromImage(B)
    W = Width - 1 : H = Height - 1

    With G
        .SmoothingMode = 2
        .PixelOffsetMode = 2
        .TextRenderingHint = 5
        .Clear(BackColor)

        '-- Colors
        .FillRectangle(New SolidBrush(_Red), New Rectangle(0, 0, 20, 40))
        .FillRectangle(New SolidBrush(_Cyan), New Rectangle(20, 0, 20, 40))
        .FillRectangle(New SolidBrush(_Blue), New Rectangle(40, 0, 20, 40))
        .FillRectangle(New SolidBrush(_LimeGreen), New Rectangle(60, 0, 20,
40))
        .FillRectangle(New SolidBrush(_Orange), New Rectangle(80, 0, 20, 40))
        .FillRectangle(New SolidBrush(_Purple), New Rectangle(100, 0, 20, 40))
        .FillRectangle(New SolidBrush(_Black), New Rectangle(120, 0, 20, 40))
        .FillRectangle(New SolidBrush(_Gray), New Rectangle(140, 0, 20, 40))
        .FillRectangle(New SolidBrush(_White), New Rectangle(160, 0, 20, 40))

        '-- Text
        .DrawString("Color Palette", Font, New SolidBrush(_White), New
Rectangle(0, 22, W, H), CenterSF)
    EndWith

 MyBase.OnPaint(e)
    G.Dispose()
    e.Graphics.InterpolationMode = 7
    e.Graphics.DrawImageUnscaled(B, 0, 0)
    B.Dispose()
EndSub
EndClass

ClassFlatGroupBox : InheritsContainerControl

#Region" Variables"

Private W, H AsInteger
Private _ShowText AsBoolean = True

#EndRegion

#Region" Properties"

<Category("Colors")> _
PublicProperty BaseColor AsColor
Get
Return _BaseColor
EndGet
Set(ByVal value As Color)
    _BaseColor = value
EndSet
EndProperty

PublicProperty ShowText AsBoolean
Get

```

```

Return _ShowText
EndGet
Set(ByName value AsBoolean)
    _ShowText = value
EndSet
EndProperty

#EndRegion

#Region" Colors"

Private _BaseColor AsColor = Color.FromArgb(60, 70, 73)

#EndRegion

SubNew()
    SetStyle(ControlStyles.AllPaintingInWmPaint Or ControlStyles.UserPaint Or
    - ControlStyles.ResizeRedraw Or ControlStyles.OptimizedDoubleBuffer
Or _ ControlStyles.SupportsTransparentBackColor, True)
    DoubleBuffered = True
    BackColor = Color.Transparent
    Size = New Size(240, 180)
    Font = New Font("Segoe ui", 10)
EndSub

ProtectedOverridesSub OnPaint(ByVal e AsPaintEventArgs)
    B = New Bitmap(Width, Height) : G = Graphics.FromImage(B)
    W = Width - 1 : H = Height - 1

    Dim GP, GP2, GP3 AsNew GraphicsPath
    Dim Base AsNew Rectangle(8, 8, W - 16, H - 16)

    With G
        .SmoothingMode = 2
        .PixelOffsetMode = 2
        .TextRenderingHint = 5
        .Clear(BackColor)

        '-- Base
        GP = Helpers.RoundRec(Base, 8)
        .FillPath(New SolidBrush(_BaseColor), GP)

        '-- Arrows
        GP2 = Helpers.DrawArrow(28, 2, False)
        .FillPath(New SolidBrush(_BaseColor), GP2)
        GP3 = Helpers.DrawArrow(28, 8, True)
        .FillPath(New SolidBrush(Color.FromArgb(60, 70, 73)), GP3)

        '-- if ShowText
        If ShowText Then
            .DrawString(Text, Font, New SolidBrush(_FlatColor), New
            Rectangle(16, 16, W, H), NearSF)
        EndIf
    EndWith

    MyBase.OnPaint(e)
    G.Dispose()
    e.Graphics.InterpolationMode = 7
    e.Graphics.DrawImageUnscaled(B, 0, 0)
    B.Dispose()

```

```

EndSub
EndClass

ClassFlatButton : InheritsControl

#Region" Variables"

Private W, H AsInteger
Private _Rounded AsBoolean = False
Private State AsMouseState = MouseState.None

#EndRegion

#Region" Properties"

#Region" Colors"

<Category("Colors")>
PublicProperty BaseColor AsColor
Get
Return _BaseColor
EndGet
Set(ByVal value As Color)
    _BaseColor = value
EndSet
EndProperty

<Category("Colors")>
PublicProperty TextColor AsColor
Get
Return _TextColor
EndGet
Set(ByVal value As Color)
    _TextColor = value
EndSet
EndProperty

<Category("Options")>
PublicProperty Rounded AsBoolean
Get
Return _Rounded
EndGet
Set(ByVal value AsBoolean)
    _Rounded = value
EndSet
EndProperty

#EndRegion

#Region" Mouse States"

ProtectedOverridesSub OnMouseDown(ByVal e As MouseEventArgs)
 MyBase.OnMouseDown(e)
     State = MouseState.Down : Invalidate()
EndSub
ProtectedOverridesSub OnMouseUp(ByVal e As MouseEventArgs)
 MyBase.OnMouseUp(e)
     State = MouseState.Over : Invalidate()
EndSub
ProtectedOverridesSub OnMouseEnter(ByVal e As EventArgs)
 MyBase.OnMouseEnter(e)
     State = MouseState.Over : Invalidate()

```

```

EndSub
ProtectedOverridesSub OnMouseLeave(ByVal e AsEventArgs)
 MyBase.OnMouseLeave(e)
 State = MouseState.None : Invalidate()
EndSub

#EndRegion

#EndRegion

#Region" Colors"

Private _BaseColor AsColor = _FlatColor
Private _TextColor AsColor = Color.FromArgb(243, 243, 243)

#EndRegion

SubNew()
 SetStyle(ControlStyles.AllPaintingInWmPaint Or ControlStyles.UserPaint Or
 -
 ControlStyles.ResizeRedraw Or ControlStyles.OptimizedDoubleBuffer
 Or -
 ControlStyles.SupportsTransparentBackColor, True)
 DoubleBuffered = True
 Size = New Size(106, 32)
 BackColor = Color.Transparent
 Font = New Font("Segoe UI", 12)
 Cursor = Cursors.Hand
EndSub

ProtectedOverridesSub OnPaint(ByVal e AsPaintEventArgs)
 B = New Bitmap(Width, Height) : G = Graphics.FromImage(B)
 W = Width - 1 : H = Height - 1

Dim GP AsNew GraphicsPath
Dim Base AsNew Rectangle(0, 0, W, H)

With G
 .SmoothingMode = 2
 .PixelOffsetMode = 2
 .TextRenderingHint = 5
 .Clear(BackColor)

SelectCase State
Case MouseState.None
If Rounded Then
'-- Base
 GP = Helpers.RoundRec(Base, 6)
 .FillPath(New SolidBrush(_BaseColor), GP)

'-- Text
.CenterSF)
.DrawString(Text, Font, New SolidBrush(_TextColor), Base,
Else
'-- Base
.FillRectangle(New SolidBrush(_BaseColor), Base)

'-- Text
.CenterSF)
.DrawString(Text, Font, New SolidBrush(_TextColor), Base,
EndIf
Case MouseState.Over

```

```

If Rounded Then
'-- Base
    GP = Helpers.RoundRec(Base, 6)
    .FillPath(New SolidBrush(_BaseColor), GP)
    .FillPath(New SolidBrush(Color.FromArgb(20, Color.White)), 
    GP)

'-- Text
    .DrawString(Text, Font, New SolidBrush(_TextColor), Base,
    CenterSF)
Else
'-- Base
    .FillRectangle(New SolidBrush(_BaseColor), Base)
    .FillRectangle(New SolidBrush(Color.FromArgb(20,
    Color.White)), Base)

'-- Text
    .DrawString(Text, Font, New SolidBrush(_TextColor), Base,
    CenterSF)
EndIf
Case MouseState.Down
If Rounded Then
'-- Base
    GP = Helpers.RoundRec(Base, 6)
    .FillPath(New SolidBrush(_BaseColor), GP)
    .FillPath(New SolidBrush(Color.FromArgb(20, Color.Black)), 
    GP)

'-- Text
    .DrawString(Text, Font, New SolidBrush(_TextColor), Base,
    CenterSF)
Else
'-- Base
    .FillRectangle(New SolidBrush(_BaseColor), Base)
    .FillRectangle(New SolidBrush(Color.FromArgb(20,
    Color.Black)), Base)

'-- Text
    .DrawString(Text, Font, New SolidBrush(_TextColor), Base,
    CenterSF)
EndIf
EndSelect
EndWith

 MyBase.OnPaint(e)
    G.Dispose()
    e.Graphics.InterpolationMode = 7
    e.Graphics.DrawImageUnscaled(B, 0, 0)
    B.Dispose()
EndSub
EndClass

<DefaultEvent("CheckedChanged")>ClassFlatToggle : InheritsControl

#Region" Variables"

Private W, H AsInteger
Private O As_Options
Private _Checked AsBoolean = False
Private State AsMouseState = MouseState.None

#EndRegion

```

```

#Region" Properties"
PublicEvent CheckedChanged(ByVal sender AsObject)

<Flags()> _
Enum _Options
    Style1
    Style2
    Style3
    Style4 '-- TODO: New Style
    Style5 '-- TODO: New Style
EndEnum

#Region" Options"
<Category("Options")> _
PublicProperty Options As _Options
Get
Return 0
EndGet
Set(ByVal value As _Options)
    0 = value
EndSet
EndProperty

<Category("Options")> _
PublicProperty Checked AsBoolean
Get
Return _Checked
EndGet
Set(ByVal value AsBoolean)
    _Checked = value
EndSet
EndProperty

#EndRegion

ProtectedOverridesSub OnTextChanged(ByVal e AsEventArgs)
 MyBase.OnTextChanged(e) : Invalidate()
EndSub

ProtectedOverridesSub OnResize(ByVal e AsEventArgs)
 MyBase.OnResize(e)
    Width = 76
    Height = 33
EndSub

#Region" Mouse States"

ProtectedOverridesSub OnMouseEnter(ByVal e As System.EventArgs)
 MyBase.OnMouseEnter(e)
    State = MouseState.Over : Invalidate()
EndSub
ProtectedOverridesSub OnMouseDown(ByVal e As System.Windows.Forms.MouseEventArgs)
 MyBase.OnMouseDown(e)
    State = MouseState.Down : Invalidate()
EndSub
ProtectedOverridesSub OnMouseLeave(ByVal e As System.EventArgs)
 MyBase.OnMouseLeave(e)
    State = MouseState.None : Invalidate()
EndSub
ProtectedOverridesSub OnMouseUp(ByVal e As System.Windows.Forms.MouseEventArgs)

```

```

 MyBase.OnMouseUp(e)
     State = MouseState.Over : Invalidate()
 EndSub
 ProtectedOverridesSub OnClick(ByVal e AsEventArgs)
 MyBase.OnClick(e)
     _Checked = Not _Checked
 RaiseEvent CheckedChanged(Me)
 EndSub

#EndRegion

#EndRegion

#Region" Colors"

Private BaseColor AsColor = _FlatColor
Private BaseColorRed AsColor = Color.FromArgb(220, 85, 96)
Private BGColor AsColor = Color.FromArgb(84, 85, 86)
Private ToggleColor AsColor = Color.FromArgb(45, 47, 49)
Private TextColor AsColor = Color.FromArgb(243, 243, 243)

#EndRegion

SubNew()
    SetStyle(ControlStyles.AllPaintingInWmPaint Or ControlStyles.UserPaint Or
    - ControlStyles.ResizeRedraw Or ControlStyles.OptimizedDoubleBuffer
Or - ControlStyles.SupportsTransparentBackColor, True)
    DoubleBuffered = True
    BackColor = Color.Transparent
    Size = New Size(44, Height + 1)
    Cursor = Cursors.Hand
    Font = New Font("Segoe UI", 10)
    Size = New Size(76, 33)
EndSub

ProtectedOverridesSub OnPaint(ByVal e AsPaintEventArgs)
    B = New Bitmap(Width, Height) : G = Graphics.FromImage(B)
    W = Width - 1 : H = Height - 1

    Dim GP, GP2 AsNew GraphicsPath
    Dim Base AsNew Rectangle(0, 0, W, H), Toggle AsNew Rectangle(CInt(W \ 2), 0, 38,
H)

    With G
        .SmoothingMode = 2
        .PixelOffsetMode = 2
        .TextRenderingHint = 5
        .Clear(BackColor)
    EndWith

    SelectCase 0
    Case _Options.Style1    '-- Style 1
    '-- Base
        GP = Helpers.RoundRec(Base, 6)
        GP2 = Helpers.RoundRec(Toggle, 6)
        .FillPath(New SolidBrush(BGColor), GP)
        .FillPath(New SolidBrush(ToggleColor), GP2)

    '-- Text
        .DrawString("OFF", Font, New SolidBrush(BGColor), New
Rectangle(19, 1, W, H), CenterSF)
    EndSelect
EndSub

```

```

If Checked Then
'-- Base
    GP = Helpers.RoundRec(Base, 6)
    GP2 = Helpers.RoundRec(New Rectangle(CInt(W \ 2), 0, 38,
H), 6)
        .FillPath(New SolidBrush(ToggleColor), GP)
        .FillPath(New SolidBrush(BaseColor), GP2)

'-- Text
    .DrawString("ON", Font, New SolidBrush(BaseColor), New
Rectangle(8, 7, W, H), NearSF)
EndIf
Case _Options.Style2  '-- Style 2
'-- Base
    GP = Helpers.RoundRec(Base, 6)
    Toggle = New Rectangle(4, 4, 36, H - 8)
    GP2 = Helpers.RoundRec(Toggle, 4)
        .FillPath(New SolidBrush(BaseColorRed), GP)
        .FillPath(New SolidBrush(ToggleColor), GP2)

'-- Lines
    .DrawLine(New Pen(BGColor), 18, 20, 18, 12)
    .DrawLine(New Pen(BGColor), 22, 20, 22, 12)
    .DrawLine(New Pen(BGColor), 26, 20, 26, 12)

'-- Text
    .DrawString("r", New Font("Marlett", 8), New
SolidBrush(TextColor), New Rectangle(19, 2, Width, Height), CenterSF)

If Checked Then
    GP = Helpers.RoundRec(Base, 6)
    Toggle = New Rectangle(CInt(W \ 2) - 2, 4, 36, H - 8)
    GP2 = Helpers.RoundRec(Toggle, 4)
        .FillPath(New SolidBrush(BaseColor), GP)
        .FillPath(New SolidBrush(ToggleColor), GP2)

'-- Lines
    .DrawLine(New Pen(BGColor), CInt(W \ 2) + 12, 20, CInt(W \
2) + 12, 12)
    .DrawLine(New Pen(BGColor), CInt(W \ 2) + 16, 20, CInt(W \
2) + 16, 12)
    .DrawLine(New Pen(BGColor), CInt(W \ 2) + 20, 20, CInt(W \
2) + 20, 12)

'-- Text
    .DrawString("ü", New Font("Wingdings", 14), New
SolidBrush(TextColor), New Rectangle(8, 7, Width, Height), NearSF)
EndIf
Case _Options.Style3  '-- Style 3
'-- Base
    GP = Helpers.RoundRec(Base, 16)
    Toggle = New Rectangle(W - 28, 4, 22, H - 8)
    GP2.AddEllipse(Toggle)
        .FillPath(New SolidBrush(ToggleColor), GP)
        .FillPath(New SolidBrush(BaseColorRed), GP2)

'-- Text
    .DrawString("OFF", Font, New SolidBrush(BaseColorRed), New
Rectangle(-12, 2, W, H), CenterSF)

If Checked Then

```

```

'-- Base
    GP = Helpers.RoundRec(Base, 16)
    Toggle = New Rectangle(6, 4, 22, H - 8)
    GP2.Reset()
    GP2.AddEllipse(Toggle)
    .FillPath(New SolidBrush(ToggleColor), GP)
    .FillPath(New SolidBrush(BaseColor), GP2)

'-- Text
    .DrawString("ON", Font, New SolidBrush(BaseColor), New
    Rectangle(12, 2, W, H), CenterSF)
EndIf
Case _Options.Style4
'-- TODO: New Styles
If Checked Then
'--
EndIf
Case _Options.Style5
'-- TODO: New Styles
If Checked Then
'--
EndIf
EndSelect

EndWith

 MyBase.OnPaint(e)
    G.Dispose()
    e.Graphics.InterpolationMode = 7
    e.Graphics.DrawImageUnscaled(B, 0, 0)
    B.Dispose()
EndSub
EndClass

<DefaultEvent("CheckedChanged")>ClassRadioButton : InheritsControl

#Region" Variables"

Private State AsMouseState = MouseState.None
Private W, H AsInteger
Private O As_Options
Private _Checked AsBoolean

#EndRegion

#Region" Properties"
Property Checked() AsBoolean
Get
Return _Checked
EndGet
Set(ByVal value AsBoolean)
    _Checked = value
    InvalidateControls()
RaiseEvent CheckedChanged(Me)
    Invalidate()
EndSet
EndProperty
Event CheckedChanged(ByVal sender AsObject)
ProtectedOverridesSub OnClick(ByVal e AsEventArgs)
IfNot _Checked Then Checked = True
 MyBase.OnClick(e)
EndSub

```

```

PrivateSub InvalidateControls()
IfNot IsHandleCreated OrElseNot _Checked ThenReturn
ForEach C As Control In Parent.Controls
If C IsNotMeAndAlsoTypeOf C Is RadioButton Then
DirectCast(C, RadioButton).Checked = False
Invalidate()
EndIf
Next
EndSub
ProtectedOverridesSub OnCreateControl()
 MyBase.OnCreateControl()
 InvalidateControls()
EndSub

<Flags()> _
Enum _Options
    Style1
    Style2
EndEnum

<Category("Options")> -
PublicProperty Options As _Options
Get
Return 0
EndGet
Set(ByVal value As _Options)
    0 = value
EndSet
EndProperty

ProtectedOverridesSub OnResize(ByVal e As EventArgs)
 MyBase.OnResize(e)
 Height = 22
EndSub

#Region" Mouse States"

ProtectedOverridesSub OnMouseDown(ByVal e As MouseEventArgs)
 MyBase.OnMouseDown(e)
 State = MouseState.Down : Invalidate()
EndSub
ProtectedOverridesSub OnMouseUp(ByVal e As MouseEventArgs)
 MyBase.OnMouseUp(e)
 State = MouseState.Over : Invalidate()
EndSub
ProtectedOverridesSub OnMouseEnter(ByVal e As EventArgs)
 MyBase.OnMouseEnter(e)
 State = MouseState.Over : Invalidate()
EndSub
ProtectedOverridesSub OnMouseLeave(ByVal e As EventArgs)
 MyBase.OnMouseLeave(e)
 State = MouseState.None : Invalidate()
EndSub

#EndRegion
#EndRegion

#Region" Colors"

Private _BaseColor AsColor = Color.FromArgb(45, 47, 49)
Private _BorderColor AsColor = _FlatColor
Private _TextColor AsColor = Color.FromArgb(243, 243, 243)

```

```

#EndRegion

SubNew()
    SetStyle(ControlStyles.AllPaintingInWmPaint Or ControlStyles.UserPaint Or
    -
        ControlStyles.ResizeRedraw Or
    ControlStyles.OptimizedDoubleBuffer, True)
    DoubleBuffered = True
    Cursor = Cursors.Hand
    Size = New Size(100, 22)
    BackColor = Color.FromArgb(60, 70, 73)
    Font = New Font("Segoe UI", 10)
EndSub
ProtectedOverridesSub OnPaint(ByVal e AsPaintEventArgs)
    B = New Bitmap(Width, Height) : G = Graphics.FromImage(B)
    W = Width - 1 : H = Height - 1

    Dim Base AsNew Rectangle(0, 2, Height - 5, Height - 5), Dot AsNew Rectangle(4, 6,
    H - 12, H - 12)

    With G
        .SmoothingMode = 2
        .TextRenderingHint = 5
        .Clear(BackColor)

        SelectCase 0
        Case _Options.Style1
        '-- Base
            .FillEllipse(New SolidBrush(_BaseColor), Base)

        SelectCase State
        Case MouseState.Over
            .DrawEllipse(New Pen(_BorderColor), Base)
        Case MouseState.Down
            .DrawEllipse(New Pen(_BorderColor), Base)
        EndSelect

        '-- If Checked
        If Checked Then
            .FillEllipse(New SolidBrush(_BorderColor), Dot)
        EndIf
        Case _Options.Style2
        '-- Base
            .FillEllipse(New SolidBrush(_BaseColor), Base)

        SelectCase State
        Case MouseState.Over
        '-- Base
            .DrawEllipse(New Pen(_BorderColor), Base)
            .FillEllipse(New SolidBrush(Color.FromArgb(118, 213,
        170)), Base)
        Case MouseState.Down
        '-- Base
            .DrawEllipse(New Pen(_BorderColor), Base)
            .FillEllipse(New SolidBrush(Color.FromArgb(118, 213,
        170)), Base)
        EndSelect

        '-- If Checked
        If Checked Then
        '-- Base

```

```

        .FillEllipse(New SolidBrush(_BorderColor), Dot)
EndIf
EndSelect

        .DrawString(Text, Font, New SolidBrush(_TextColor), New Rectangle(20,
2, W, H), NearSF)
EndWith

 MyBase.OnPaint(e)
    G.Dispose()
    e.Graphics.InterpolationMode = 7
    e.Graphics.DrawImageUnscaled(B, 0, 0)
    B.Dispose()
EndSub
EndClass

<DefaultEvent("CheckedChanged")>ClassFlatCheckBox : InheritsControl

#Region" Variables"

Private W, H AsInteger
Private State AsMouseState = MouseState.None
Private O As_Options
Private _Checked AsBoolean

#EndRegion

#Region" Properties"
ProtectedOverridesSub OnTextChanged(ByVal e As System.EventArgs)
MyBase.OnTextChanged(e)
    Invalidate()
EndSub

Property Checked() AsBoolean
Get
Return _Checked
EndGet
Set(ByVal value AsBoolean)
    _Checked = value
    Invalidate()
EndSet
EndProperty

Event CheckedChanged(ByVal sender AsObject)
ProtectedOverridesSub OnClick(ByVal e As System.EventArgs)
    _Checked = Not _Checked
    RaiseEvent CheckedChanged(Me)
MyBase.OnClick(e)
EndSub

<Flags()> _
Enum_Options
    Style1
    Style2
EndEnum

<Category("Options")> _
PublicProperty Options As_Options
Get
Return O
EndGet
Set(ByVal value As _Options)

```

```

        0 = value
EndSet
EndProperty

ProtectedOverridesSub OnResize(ByVal e AsEventArgs)
 MyBase.OnResize(e)
    Height = 22
EndSub

#Region" Colors"

<Category("Colors")> _
PublicProperty BaseColor AsColor
Get
Return _BaseColor
EndGet
Set(ByVal value As Color)
    _BaseColor = value
EndSet
EndProperty

<Category("Colors")> _
PublicProperty BorderColor AsColor
Get
Return _BorderColor
EndGet
Set(ByVal value As Color)
    _BorderColor = value
EndSet
EndProperty

#EndRegion

#Region" Mouse States"

ProtectedOverridesSub OnMouseDown(ByVal e AsMouseEventArgs)
 MyBase.OnMouseDown(e)
    State = MouseState.Down : Invalidate()
EndSub
ProtectedOverridesSub OnMouseUp(ByVal e AsMouseEventArgs)
 MyBase.OnMouseUp(e)
    State = MouseState.Over : Invalidate()
EndSub
ProtectedOverridesSub OnMouseEnter(ByVal e AsEventArgs)
 MyBase.OnMouseEnter(e)
    State = MouseState.Over : Invalidate()
EndSub
ProtectedOverridesSub OnMouseLeave(ByVal e AsEventArgs)
 MyBase.OnMouseLeave(e)
    State = MouseState.None : Invalidate()
EndSub

#EndRegion

#EndRegion

#Region" Colors"

Private _BaseColor AsColor = Color.FromArgb(45, 47, 49)
Private _BorderColor AsColor = _FlatColor
Private _TextColor AsColor = Color.FromArgb(243, 243, 243)

```

```

#EndRegion

SubNew()
    SetStyle(ControlStyles.AllPaintingInWmPaint Or ControlStyles.UserPaint Or
    -
        ControlStyles.ResizeRedraw Or
    ControlStyles.OptimizedDoubleBuffer, True)
    DoubleBuffered = True
    BackColor = Color.FromArgb(60, 70, 73)
    Cursor = Cursors.Hand
    Font = New Font("Segoe UI", 10)
    Size = New Size(112, 22)
EndSub

ProtectedOverridesSub OnPaint(ByVal e As PaintEventArgs)
    B = New Bitmap(Width, Height) : G = Graphics.FromImage(B)
    W = Width - 1 : H = Height - 1

    Dim Base AsNew Rectangle(0, 2, Height - 5, Height - 5)

    With G
        .SmoothingMode = 2
        .TextRenderingHint = 5
        .Clear(BackColor)
    SelectCase 0
    Case _Options.Style1 '-- Style 1
    '-- Base
        .FillRectangle(New SolidBrush(_BaseColor), Base)

    SelectCase State
    Case MouseState.Over
    '-- Base
        .DrawRectangle(New Pen(_BorderColor), Base)
    Case MouseState.Down
    '-- Base
        .DrawRectangle(New Pen(_BorderColor), Base)
    EndSelect

    '-- If Checked
    If Checked Then
        .DrawString("ü", New Font("Wingdings", 18), New
SolidBrush(_BorderColor), New Rectangle(5, 7, H - 9, H - 9), CenterSF)
    EndIf

    '-- If Enabled
    IfMe.Enabled = FalseThen
        .FillRectangle(New SolidBrush(Color.FromArgb(54, 58, 61)), Base)
        .DrawString(Text, Font, New SolidBrush(Color.FromArgb(140,
142, 143)), New Rectangle(20, 2, W, H), NearSF)
    EndIf

    '-- Text
        .DrawString(Text, Font, New SolidBrush(_TextColor), New
Rectangle(20, 2, W, H), NearSF)
    Case _Options.Style2 '-- Style 2
    '-- Base
        .FillRectangle(New SolidBrush(_BaseColor), Base)

    SelectCase State
    Case MouseState.Over
    '-- Base

```

```

        .DrawRectangle(New Pen(_BorderColor), Base)
        .FillRectangle(New SolidBrush(Color.FromArgb(118, 213,
170)), Base)
Case MouseState.Down
'-- Base
        .DrawRectangle(New Pen(_BorderColor), Base)
        .FillRectangle(New SolidBrush(Color.FromArgb(118, 213,
170)), Base)
EndSelect

'-- If Checked
If Checked Then
        .DrawString("ü", New Font("Wingdings", 18), New
SolidBrush(_BorderColor), New Rectangle(5, 7, H - 9, H - 9), CenterSF)
EndIf

'-- If Enabled
IfMe.Enabled = FalseThen
        .FillRectangle(New SolidBrush(Color.FromArgb(54, 58, 61)),
Base)
        .DrawString(Text, Font, New SolidBrush(Color.FromArgb(48,
119, 91)), New Rectangle(20, 2, W, H), NearSF)
EndIf

'-- Text
        .DrawString(Text, Font, New SolidBrush(_TextColor), New
Rectangle(20, 2, W, H), NearSF)
EndSelect
EndWith

 MyBase.OnPaint(e)
    G.Dispose()
    e.Graphics.InterpolationMode = 7
    e.Graphics.DrawImageUnscaled(B, 0, 0)
    B.Dispose()
EndSub
EndClass

<DefaultEvent("TextChanged")>ClassFlatTextBox : InheritsControl

#Region" Variables"

Private W, H AsInteger
Private State AsMouseState = MouseState.None
Private WithEvents TB As Windows.Forms.TextBox

#EndRegion

#Region" Properties"

#Region" TextBox Properties"

Private _TextAlign AsHorizontalAlignment = HorizontalAlignment.Left
<Category("Options")>
Property TextAlign() AsHorizontalAlignment
Get
Return _TextAlign
EndGet
Set(ByVal value As HorizontalAlignment)
    _TextAlign = value
If TB IsNot NothingThen
    TB.TextAlign = value
EndIf
EndRegion
EndClass

```

```

EndIf
EndSet
EndProperty
Private _MaxLength AsInteger = 32767
<Category("Options")> -
Property MaxLength() AsInteger
Get
Return _MaxLength
EndGet
Set(ByName value AsInteger)
    _MaxLength = value
If TB IsNotNothingThen
    TB.MaxLength = value
EndIf
EndSet
EndProperty
Private _ReadOnly AsBoolean
<Category("Options")> -
Property [ReadOnly]() AsBoolean
Get
Return _ReadOnly
EndGet
Set(ByName value AsBoolean)
    _ReadOnly = value
If TB IsNotNothingThen
    TB.ReadOnly = value
EndIf
EndSet
EndProperty
Private _UseSystemPasswordChar AsBoolean
<Category("Options")> -
Property UseSystemPasswordChar() AsBoolean
Get
Return _UseSystemPasswordChar
EndGet
Set(ByName value AsBoolean)
    _UseSystemPasswordChar = value
If TB IsNotNothingThen
    TB.UseSystemPasswordChar = value
EndIf
EndSet
EndProperty
Private _Multiline AsBoolean
<Category("Options")> -
Property Multiline() AsBoolean
Get
Return _Multiline
EndGet
Set(ByName value AsBoolean)
    _Multiline = value
If TB IsNotNothingThen
    TB.Multiline = value

If value Then
    TB.Height = Height - 11
Else
    Height = TB.Height + 11
EndIf

EndIf
EndSet
EndProperty

```

```

<Category("Options")> _
OverridesProperty Text AsString
Get
Return MyBase.Text
EndGet
Set(ByVal value AsString)
MyBase.Text = value
If TB IsNot NothingThen
    TB.Text = value
EndIf
EndSet
EndProperty
<Category("Options")> _
OverridesProperty Font AsFont
Get
Return MyBase.Font
EndGet
Set(ByVal value As Font)
MyBase.Font = value
If TB IsNot NothingThen
    TB.Font = value
    TB.Location = New Point(3, 5)
    TB.Width = Width - 6

IfNot _Multiline Then
    Height = TB.Height + 11
EndIf
EndIf
EndSet
EndProperty

ProtectedOverridesSub OnCreateControl()
 MyBase.OnCreateControl()
 IfNot Controls.Contains(TB) Then
     Controls.Add(TB)
 EndIf
EndSub
PrivateSub OnBaseTextChanged(ByVal s AsObject, ByVal e AsEventArgs)
    Text = TB.Text
EndSub
PrivateSub OnBaseKeyDown(ByVal s AsObject, ByVal e AsKeyEventArgs)
    If e.Control AndAlso e.KeyCode = Keys.A Then
        TB.SelectAll()
        e.SuppressKeyPress = True
    EndIf
    If e.Control AndAlso e.KeyCode = Keys.C Then
        TB.Copy()
        e.SuppressKeyPress = True
    EndIf
EndSub
ProtectedOverridesSub OnResize(ByVal e AsEventArgs)
    TB.Location = New Point(5, 5)
    TB.Width = Width - 10

    If _Multiline Then
        TB.Height = Height - 11
    Else
        Height = TB.Height + 11
    EndIf

    MyBase.OnResize(e)
EndSub

```

```

#EndRegion

#Region" Colors"

<Category("Colors")>
PublicProperty TextColor AsColor
Get
Return _TextColor
EndGet
Set(ByVal value As Color)
    _TextColor = value
EndSet
EndProperty

PublicOverridesProperty ForeColor() AsColor
Get
Return _TextColor
EndGet
Set(ByVal value As Color)
    _TextColor = value
EndSet
EndProperty

#EndRegion

#Region" Mouse States"

ProtectedOverridesSub OnMouseDown(ByVal e As MouseEventArgs)
 MyBase.OnMouseDown(e)
    State = MouseState.Down : Invalidate()
EndSub
ProtectedOverridesSub OnMouseUp(ByVal e As MouseEventArgs)
 MyBase.OnMouseUp(e)
    State = MouseState.Over : TB.Focus() : Invalidate()
EndSub
ProtectedOverridesSub OnMouseEnter(ByVal e As EventArgs)
 MyBase.OnMouseEnter(e)
    State = MouseState.Over : TB.Focus() : Invalidate()
EndSub
ProtectedOverridesSub OnMouseLeave(ByVal e As EventArgs)
 MyBase.OnMouseLeave(e)
    State = MouseState.None : Invalidate()
EndSub

#EndRegion

#EndRegion

#Region" Colors"

Private _BaseColor AsColor = Color.FromArgb(45, 47, 49)
Private _TextColor AsColor = Color.FromArgb(192, 192, 192)
Private _BorderColor AsColor = _FlatColor

#EndRegion

SubNew()
    SetStyle(ControlStyles.AllPaintingInWmPaint Or ControlStyles.UserPaint Or
    - ControlStyles.ResizeRedraw Or ControlStyles.OptimizedDoubleBuffer
    Or _

```

```

        ControlStyles.SupportsTransparentBackColor, True)
DoubleBuffered = True

BackColor = Color.Transparent

TB = New Windows.Forms.TextBox
TB.Font = New Font("Segoe UI", 10)
TB.Text = Text
TB.BackColor = _BaseColor
TB.ForeColor = _TextColor
TB.MaxLength = _MaxLength
TB.Multiline = _Multiline
TB.ReadOnly = _ReadOnly
TB.UseSystemPasswordChar = _UseSystemPasswordChar
TB.BorderStyle = BorderStyle.None
TB.Location = New Point(5, 5)
TB.Width = Width - 10

TB.Cursor = Cursors.IBeam

If _Multiline Then
    TB.Height = Height - 11
Else
    Height = TB.Height + 11
EndIf

AddHandler TB.TextChanged, AddressOf OnBaseTextChanged
AddHandler TB.KeyDown, AddressOf OnBaseKeyDown
EndSub

ProtectedOverridesSub OnPaint(ByVal e AsPaintEventArgs)
    B = New Bitmap(Width, Height) : G = Graphics.FromImage(B)
    W = Width - 1 : H = Height - 1

Dim Base AsNew Rectangle(0, 0, W, H)

With G
    .SmoothingMode = 2
    .PixelOffsetMode = 2
    .TextRenderingHint = 5
    .Clear(BackColor)

'-- Colors
    TB.BackColor = _BaseColor
    TB.ForeColor = _TextColor

'-- Base
    .FillRectangle(New SolidBrush(_BaseColor), Base)
EndWith

 MyBase.OnPaint(e)
    G.Dispose()
    e.Graphics.InterpolationMode = 7
    e.Graphics.DrawImageUnscaled(B, 0, 0)
    B.Dispose()
EndSub

EndClass

ClassFlatTabControl : InheritsTabControl

#Region" Variables"

```

```

Private W, H AsInteger
#EndRegion

#Region" Properties"
ProtectedOverridesSub CreateHandle()
 MyBase.CreateHandle()
 Alignment = TabAlignment.Top
EndSub

#Region" Colors"

<Category("Colors")> _
PublicProperty BaseColor AsColor
Get
Return _BaseColor
EndGet
Set(ByVal value As Color)
    _BaseColor = value
EndSet
EndProperty

<Category("Colors")> _
PublicProperty ActiveColor AsColor
Get
Return _ActiveColor
EndGet
Set(ByVal value As Color)
    _ActiveColor = value
EndSet
EndProperty

#EndRegion

#EndRegion

#Region" Colors"

Private BGColor AsColor = Color.FromArgb(60, 70, 73)
Private _BaseColor AsColor = Color.FromArgb(45, 47, 49)
Private _ActiveColor AsColor = _FlatColor

#EndRegion

SubNew()
    SetStyle(ControlStyles.AllPaintingInWmPaint Or ControlStyles.UserPaint Or
    - ControlStyles.ResizeRedraw Or
ControlStyles.OptimizedDoubleBuffer, True)
    DoubleBuffered = True
    BackColor = Color.FromArgb(60, 70, 73)

    Font = New Font("Segoe UI", 10)
   SizeMode = TabSizeMode.Fixed
    ItemSize = New Size(120, 40)
EndSub

ProtectedOverridesSub OnPaint(ByVal e AsPaintEventArgs)
    B = New Bitmap(Width, Height) : G = Graphics.FromImage(B)
    W = Width - 1 : H = Height - 1

```

```

With G
    .SmoothingMode = 2
    .PixelOffsetMode = 2
    .TextRenderingHint = 5
    .Clear(_BaseColor)

Try : SelectedTab.BackColor = BGColor : Catch : EndTry

For i = 0 To TabCount - 1
Dim Base AsNew Rectangle(New Point(GetTabRect(i).Location.X + 2,
GetTabRect(i).Location.Y), New Size(GetTabRect(i).Width, GetTabRect(i).Height))
Dim BaseSize AsNew Rectangle(Base.Location, New Size(Base.Width, Base.Height))

If i = SelectedIndex Then
'-- Base
    .FillRectangle(New SolidBrush(_BaseColor), BaseSize)

'-- Gradiant
'.fill
    .FillRectangle(New SolidBrush(_ActiveColor), BaseSize)

'-- ImageList
If ImageList IsNotNothingThen
Try
If ImageList.Images(TabPages(i).ImageIndex) IsNotNothingThen
'-- Image

    .DrawImage(ImageList.Images(TabPages(i).ImageIndex), New Point(BaseSize.Location.X
+ 8, BaseSize.Location.Y + 6))
'-- Text
        .DrawString("      "& TabPages(i).Text, Font,
Brushes.White, BaseSize, CenterSF)
Else
'-- Text
        .DrawString(TabPages(i).Text, Font, Brushes.White,
BaseSize, CenterSF)
EndIf
Catch ex As Exception
ThrowNew Exception(ex.Message)
EndTry
Else
'-- Text
        .DrawString(TabPages(i).Text, Font, Brushes.White,
BaseSize, CenterSF)
EndIf
Else
'-- Base
    .FillRectangle(New SolidBrush(_BaseColor), BaseSize)

'-- ImageList
If ImageList IsNotNothingThen
Try
If ImageList.Images(TabPages(i).ImageIndex) IsNotNothingThen
'-- Image

    .DrawImage(ImageList.Images(TabPages(i).ImageIndex), New Point(BaseSize.Location.X
+ 8, BaseSize.Location.Y + 6))
'-- Text
        .DrawString("      "& TabPages(i).Text, Font, New
SolidBrush(Color.White), BaseSize, New StringFormat With {.LineAlignment =
StringAlignment.Center, .Alignment = StringAlignment.Center})

```

```

Else
'-- Text
    .DrawString(TabPages(i).Text, Font, New
SolidBrush(Color.White), BaseSize, New StringFormat With {.LineAlignment =
StringAlignment.Center, .Alignment = StringAlignment.Center})
EndIf
Catch ex As Exception
ThrowNew Exception(ex.Message)
EndTry
Else
'-- Text
    .DrawString(TabPages(i).Text, Font, New
SolidBrush(Color.White), BaseSize, New StringFormat With {.LineAlignment =
StringAlignment.Center, .Alignment = StringAlignment.Center})
EndIf
EndIf
Next
EndWith

 MyBase.OnPaint(e)
    G.Dispose()
    e.Graphics.InterpolationMode = 7
    e.Graphics.DrawImageUnscaled(B, 0, 0)
    B.Dispose()
EndSub
EndClass

ClassFlatAlertBox : InheritsControl

''' <summary>
''' How to use: FlatAlertBox.ShowControl(Kind, String, Interval)
''' </summary>
''' <remarks></remarks>

#Region" Variables"

Private W, H AsInteger
Private K As_Kind
Private _TextAsString
Private State AsMouseState = MouseState.None
Private X AsInteger
Private WithEvents T AsTimer

#EndRegion

#Region" Properties"

<Flags()> _
Enum_Kind
    [Success]
    [Error]
    [Info]
EndEnum

#Region" Options"

<Category("Options")> _
PublicProperty kind As_Kind
Get
Return K
EndGet
Set(ByVal value As _Kind)

```

```

        K = value
EndSet
EndProperty

<Category("Options")> _
OverridesProperty Text AsString
Get
Return MyBase.Text
EndGet
Set(ByVal value AsString)
MyBase.Text = value
If _Text IsNot Nothing Then
    _Text = value
EndIf
EndSet
EndProperty

<Category("Options")> _
ShadowsProperty Visible AsBoolean
Get
Return MyBase.Visible = False
EndGet
Set(ByVal value AsBoolean)
MyBase.Visible = value
EndSet
EndProperty

#EndRegion

ProtectedOverridesSub OnTextChanged(ByVal e AsEventArgs)
MyBase.OnTextChanged(e) : Invalidate()
EndSub

ProtectedOverridesSub OnResize(ByVal e AsEventArgs)
MyBase.OnResize(e)
    Height = 42
EndSub

PublicSub ShowControl(ByVal Kind AsKind, ByVal StrAsString, ByVal Interval
AsInteger)
    K = Kind
    Text = Str
Me.Visible = True
    T = New Timer
    T.Interval = Interval
    T.Enabled = True
EndSub

PrivateSub T_Tick(ByVal sender AsObject, ByVal e AsEventArgs) Handles T.Tick
Me.Visible = False
    T.Enabled = False
    T.Dispose()
EndSub

#Region" Mouse States"

ProtectedOverridesSub OnMouseDown(ByVal e AsMouseEventArgs)
MyBase.OnMouseDown(e)
    State = MouseState.Down : Invalidate()
EndSub
ProtectedOverridesSub OnMouseUp(ByVal e AsMouseEventArgs)
MyBase.OnMouseUp(e)

```

```

        State = MouseState.Over : Invalidate()
EndSub
ProtectedOverridesSub OnMouseEnter(ByVal e AsEventArgs)
 MyBase.OnMouseEnter(e)
        State = MouseState.Over : Invalidate()
EndSub
ProtectedOverridesSub OnMouseLeave(ByVal e AsEventArgs)
 MyBase.OnMouseLeave(e)
        State = MouseState.None : Invalidate()
EndSub

ProtectedOverridesSub OnMouseMove(ByVal e AsMouseEventArgs)
 MyBase.OnMouseMove(e)
        X = e.X : Invalidate()
EndSub

ProtectedOverridesSub OnClick(ByVal e AsEventArgs)
 MyBase.OnClick(e)
 Me.Visible = False
EndSub

#EndRegion

#EndRegion

#Region " Colors"
    Private SuccessColor AsColor = Color.FromArgb(60, 85, 79)
    Private SuccessText AsColor = Color.FromArgb(35, 169, 110)
    Private ErrorColor AsColor = Color.FromArgb(87, 71, 71)
    Private ErrorText AsColor = Color.FromArgb(254, 142, 122)
    Private InfoColor AsColor = Color.FromArgb(70, 91, 94)
    Private InfoText AsColor = Color.FromArgb(97, 185, 186)

#EndRegion

SubNew()
    SetStyle(ControlStyles.AllPaintingInWmPaint Or ControlStyles.UserPaint Or
    -
        ControlStyles.ResizeRedraw Or
    ControlStyles.OptimizedDoubleBuffer, True)
        DoubleBuffered = True
        BackColor = Color.FromArgb(60, 70, 73)
        Size = New Size(576, 42)
        Location = New Point(10, 61)
        Font = New Font("Segoe UI", 10)
        Cursor = Cursors.Hand
EndSub

ProtectedOverridesSub OnPaint(ByVal e AsPaintEventArgs)
    B = New Bitmap(Width, Height) : G = Graphics.FromImage(B)
    W = Width - 1 : H = Height - 1

    Dim Base AsNew Rectangle(0, 0, W, H)

    With G
        .SmoothingMode = 2
        .TextRenderingHint = 5
        .Clear(BackColor)

    SelectCase K
    Case _Kind.Success

```

```

'-- Base
    .FillRectangle(New SolidBrush(SuccessColor), Base)

'-- Ellipse
    .FillEllipse(New SolidBrush(SuccessText), New Rectangle(8, 9,
24, 24))
    .FillEllipse(New SolidBrush(SuccessColor), New Rectangle(10,
11, 20, 20))

'-- Checked Sign
    .DrawString("ü", New Font("Wingdings", 22), New
SolidBrush(SuccessText), New Rectangle(7, 7, W, H), NearSF)
    .DrawString(Text, Font, New SolidBrush(SuccessText), New
Rectangle(48, 12, W, H), NearSF)

'-- X button
    .FillEllipse(New SolidBrush(Color.FromArgb(35, Color.Black)), New
Rectangle(W - 30, H - 29, 17, 17))
    .DrawString("r", New Font("Marlett", 8), New
SolidBrush(SuccessColor), New Rectangle(W - 28, 16, W, H), NearSF)

SelectCase State '-- Mouse Over
Case MouseState.Over
    .DrawString("r", New Font("Marlett", 8), New
SolidBrush(Color.FromArgb(25, Color.White)), New Rectangle(W - 28, 16, W, H),
NearSF)
EndSelect

Case _Kind.Error
'-- Base
    .FillRectangle(New SolidBrush(ErrorColor), Base)

'-- Ellipse
    .FillEllipse(New SolidBrush(ErrorText), New Rectangle(8, 9,
24, 24))
    .FillEllipse(New SolidBrush(ErrorColor), New Rectangle(10, 11,
20, 20))

'-- X Sign
    .DrawString("r", New Font("Marlett", 16), New
SolidBrush(ErrorText), New Rectangle(6, 11, W, H), NearSF)
    .DrawString(Text, Font, New SolidBrush(ErrorText), New
Rectangle(48, 12, W, H), NearSF)

'-- X button
    .FillEllipse(New SolidBrush(Color.FromArgb(35, Color.Black)), New
Rectangle(W - 32, H - 29, 17, 17))
    .DrawString("r", New Font("Marlett", 8), New
SolidBrush(ErrorColor), New Rectangle(W - 30, 17, W, H), NearSF)

SelectCase State
Case MouseState.Over '-- Mouse Over
    .DrawString("r", New Font("Marlett", 8), New
SolidBrush(Color.FromArgb(25, Color.White)), New Rectangle(W - 30, 15, W, H),
NearSF)
EndSelect

Case _Kind.Info
'-- Base
    .FillRectangle(New SolidBrush(InfoColor), Base)

'-- Ellipse

```

```

        .FillEllipse(New SolidBrush(InfoText), New Rectangle(8, 9, 24,
24))
                .FillEllipse(New SolidBrush(InfoColor), New Rectangle(10, 11,
20, 20))

'-- Info Sign
        .DrawString("i", New Font("Segoe UI", 20, FontStyle.Bold), New
SolidBrush(InfoText), New Rectangle(12, -4, W, H), NearSF)
        .DrawString(Text, Font, New SolidBrush(InfoText), New
Rectangle(48, 12, W, H), NearSF)

'-- X button
        .FillEllipse(New SolidBrush(Color.FromArgb(35, Color.Black)), New
Rectangle(W - 32, H - 29, 17, 17))
        .DrawString("r", New Font("Marlett", 8), New
SolidBrush(InfoColor), New Rectangle(W - 30, 17, W, H), NearSF)

SelectCase State
Case MouseState.Over ' -- Mouse Over
        .DrawString("r", New Font("Marlett", 8), New
SolidBrush(Color.FromArgb(25, Color.White)), New Rectangle(W - 30, 17, W, H),
NearSF)
EndSelect
EndSelect

EndWith

 MyBase.OnPaint(e)
        G.Dispose()
        e.Graphics.InterpolationMode = 7
        e.Graphics.DrawImageUnscaled(B, 0, 0)
        B.Dispose()
EndSub

EndClass

ClassFlatProgressBar : InheritsControl

#Region" Variables"

Private W, H AsInteger
Private _Value AsInteger = 0
Private _Maximum AsInteger = 100

#EndRegion

#Region" Properties"

#Region" Control"

<Category("Control")>
PublicProperty Maximum() AsInteger
Get
Return _Maximum
EndGet
Set(ByVal V AsInteger)
SelectCase V
CaseIs < _Value
        _Value = V
EndSelect
        _Maximum = V
        Invalidate()
EndProperty
EndRegion
EndClass

```

```

EndSet
EndProperty

<Category("Control")>
PublicProperty Value() AsInteger
Get
SelectCase _Value
Case 0
Return 0
Invalidate()
Case Else
Return _Value
Invalidate()
EndSelect
EndGet
Set(ByVal V AsInteger)
SelectCase V
CaseIs > _Maximum
V = _Maximum
Invalidate()
EndSelect
_Value = V
Invalidate()
EndSet
EndProperty

#EndRegion

#Region" Colors"

<Category("Colors")>
PublicProperty ProgressColor AsColor
Get
Return _ProgressColor
EndGet
Set(ByVal value As Color)
_ProgressColor = value
EndSet
EndProperty

<Category("Colors")>
PublicProperty DarkerProgress AsColor
Get
Return _DarkerProgress
EndGet
Set(ByVal value As Color)
_DarkerProgress = value
EndSet
EndProperty

#EndRegion

ProtectedOverridesSub OnResize(ByVal e AsEventArgs)
 MyBase.OnResize(e)
 Height = 42
EndSub

ProtectedOverridesSub CreateHandle()
 MyBase.CreateHandle()
 Height = 42
EndSub

```

```

PublicSub Increment(ByVal Amount AsInteger)
    Value += Amount
EndSub

#EndRegion

#Region" Colors"

Private _BaseColor AsColor = Color.FromArgb(45, 47, 49)
Private _ProgressColor AsColor = _FlatColor
Private _DarkerProgress AsColor = Color.FromArgb(23, 148, 92)

#EndRegion

SubNew()
    SetStyle(ControlStyles.AllPaintingInWmPaint Or ControlStyles.UserPaint Or
    -
        ControlStyles.ResizeRedraw Or
    ControlStyles.OptimizedDoubleBuffer, True)
    DoubleBuffered = True
    BackColor = Color.FromArgb(60, 70, 73)
    Height = 42
EndSub

ProtectedOverridesSub OnPaint(ByVal e AsPaintEventArgs)
    B = New Bitmap(Width, Height) : G = Graphics.FromImage(B)
    W = Width - 1 : H = Height - 1

    Dim Base AsNew Rectangle(0, 24, W, H)
    Dim GP, GP2, GP3 AsNew GraphicsPath

    With G
        .SmoothingMode = 2
        .PixelOffsetMode = 2
        .TextRenderingHint = 5
        .Clear(BackColor)

        '-- Progress Value
        Dim iValue AsInteger = CInt(_Value / _Maximum * Width)

        SelectCase Value
        Case 0
            '-- Base
            .FillRectangle(New SolidBrush(_BaseColor), Base)
        Case Else
            '-- Progress
            .FillRectangle(New SolidBrush(_ProgressColor), New
            Rectangle(0, 24, iValue - 1, H - 1))
        Case 100
            '-- Base
            .FillRectangle(New SolidBrush(_BaseColor), Base)
        Case Else
            '-- Progress
            .FillRectangle(New SolidBrush(_ProgressColor), New
            Rectangle(0, 24, iValue - 1, H - 1))
        EndSelect
    EndWith

    GP.AddRectangle(New Rectangle(0, 24, iValue - 1, H - 1))
    FillPath(New SolidBrush(_ProgressColor), GP)

    '-- Hatch Brush

```

```

Dim HB AsNew HatchBrush(HatchStyle.Plaid, _DarkerProgress, _ProgressColor)
    .FillRectangle(HB, New Rectangle(0, 24, iValue - 1, H - 1))

'-- Balloon
Dim Balloon AsNew Rectangle(iValue - 18, 0, 34, 16)
    GP2 = Helpers.RoundRec(Balloon, 4)
    .FillPath(New SolidBrush(_BaseColor), GP2)

'-- Arrow
    GP3 = Helpers.DrawArrow(iValue - 9, 16, True)
    .FillPath(New SolidBrush(_BaseColor), GP3)

'-- Value > You can add "%" > value & "%"
    .DrawString(Value, New Font("Segoe UI", 10), New
SolidBrush(_ProgressColor), New Rectangle(iValue - 11, -2, W, H), NearSF)
EndSelect
EndWith

 MyBase.OnPaint(e)
    G.Dispose()
    e.Graphics.InterpolationMode = 7
    e.Graphics.DrawImageUnscaled(B, 0, 0)
    B.Dispose()
EndSub

EndClass

ClassFlatComboBox : Inherits Windows.Forms.ComboBox

#Region" Variables"

Private W, H AsInteger
Private _startIndex AsInteger = 0
Private x, y AsInteger

#EndRegion

#Region" Properties"

#Region" Mouse States"

Private State AsMouseState = MouseState.None
ProtectedOverridesSub OnMouseDown(ByVal e AsMouseEventArgs)
    MyBase.OnMouseDown(e)
        State = MouseState.Down : Invalidate()
EndSub
ProtectedOverridesSub OnMouseUp(ByVal e AsMouseEventArgs)
    MyBase.OnMouseUp(e)
        State = MouseState.Over : Invalidate()
EndSub
ProtectedOverridesSub OnMouseEnter(ByVal e AsEventArgs)
    MyBase.OnMouseEnter(e)
        State = MouseState.Over : Invalidate()
EndSub
ProtectedOverridesSub OnMouseLeave(ByVal e AsEventArgs)
    MyBase.OnMouseLeave(e)
        State = MouseState.None : Invalidate()
EndSub

ProtectedOverridesSub OnMouseMove(ByVal e AsMouseEventArgs)
    MyBase.OnMouseMove(e)
        x = e.Location.X

```

```

        y = e.Location.Y
        Invalidate()
If e.X < Width - 41 Then Cursor = Cursors.IBeam Else Cursor = Cursors.Hand
EndSub

ProtectedOverridesSub OnDrawItem(ByVal e As DrawItemEventArgs)
 MyBase.OnDrawItem(e) : Invalidate()
If (e.State And DrawItemState.Selected) = DrawItemState.Selected Then
    Invalidate()
EndIf
EndSub

ProtectedOverridesSub OnClick(ByVal e As EventArgs)
 MyBase.OnClick(e) : Invalidate()
EndSub

#EndRegion

#Region " Colors"

<Category("Colors")> _
PublicProperty HoverColor AsColor
Get
Return _HoverColor
EndGet
Set(ByVal value As Color)
    _HoverColor = value
EndSet
EndProperty

#EndRegion

PrivateProperty StartIndex AsInteger
Get
Return _StartIndex
EndGet
Set(ByVal value AsInteger)
    _StartIndex = value
Try
MyBase.SelectedIndex = value
Catch
EndTry
    Invalidate()
EndSet
EndProperty

Sub DrawItem_(ByVal sender As System.Object, ByVal e As
System.Windows.Forms.DrawItemEventArgs) HandlesMe.DrawItem
If e.Index < 0 ThenExit Sub
    e.DrawBackground()
    e.DrawFocusRectangle()

    e.Graphics.SmoothingMode = 2
    e.Graphics.PixelOffsetMode = 2
    e.Graphics.TextRenderingHint = 5
    e.Graphics.InterpolationMode = 7

If (e.State And DrawItemState.Selected) = DrawItemState.Selected Then
    '-- Selected item
        e.Graphics.FillRectangle(New SolidBrush(_HoverColor), e.Bounds)
Else
    '-- Not Selected

```

```

        e.Graphics.FillRectangle(New SolidBrush(_BaseColor), e.Bounds)
EndIf

'-- Text
        e.Graphics.DrawString(MyBase.GetItemText(MyBase.Items(e.Index)), New
Font("Segoe UI", 8), _
Brushes.White, New Rectangle(e.Bounds.X + 2, e.Bounds.Y + 2,
e.Bounds.Width, e.Bounds.Height))

        e.Graphics.Dispose()
EndSub

ProtectedOverridesSub OnResize(ByVal e AsEventArgs)
MyBase.OnResize(e)
    Height = 18
EndSub

#EndRegion

#Region" Colors"

Private _BaseColor AsColor = Color.FromArgb(25, 27, 29)
Private _BColor AsColor = Color.FromArgb(45, 47, 49)
Private _HoverColor AsColor = Color.FromArgb(35, 168, 109)

#EndRegion

SubNew()
    SetStyle(ControlStyles.AllPaintingInWmPaint Or ControlStyles.UserPaint Or
-
ControlStyles.ResizeRedraw Or
ControlStyles.OptimizedDoubleBuffer, True)
    DoubleBuffered = True

    DrawMode = DrawMode.OwnerDrawFixed
    BackColor = Color.FromArgb(45, 45, 48)
    ForeColor = Color.White
    DropDownStyle = ComboBoxStyle.DropDownList
    Cursor = Cursors.Hand
    StartIndex = 0
    ItemHeight = 18
    Font = New Font("Segoe UI", 8, FontStyle.Regular)
EndSub

ProtectedOverridesSub OnPaint(ByVal e AsPaintEventArgs)
    B = New Bitmap(Width, Height) : G = Graphics.FromImage(B)
    W = Width : H = Height

    Dim Base AsNew Rectangle(0, 0, W, H)
    Dim Button AsNew Rectangle(CInt(W - 40), 0, W, H)
    Dim GP, GP2 AsNew GraphicsPath

    With G
        .Clear(Color.FromArgb(45, 45, 48))
        .SmoothingMode = 2
        .PixelOffsetMode = 2
        .TextRenderingHint = 5

'-- Base
        .FillRectangle(New SolidBrush(_BColor), Base)
    EndWith
EndSub

```

```

'-- Button
    GP.Reset()
    GP.AddRectangle(Button)
    .SetClip(GP)
    .FillRectangle(New SolidBrush(_BaseColor), Button)
    .ResetClip()

'-- Lines
    .DrawLine(Pens.White, W - 10, 6, W - 30, 6)
    .DrawLine(Pens.White, W - 10, 12, W - 30, 12)
    .DrawLine(Pens.White, W - 10, 18, W - 30, 18)

'-- Text
    .DrawString(Text, Font, Brushes.White, New Point(4, 6), NearSF)
EndWith

    G.Dispose()
    e.Graphics.InterpolationMode = 7
    e.Graphics.DrawImageUnscaled(B, 0, 0)
    B.Dispose()
EndSub
EndClass

ClassFlatStickyButton : InheritsControl

#Region" Variables"
Private W, H AsInteger
Private State AsMouseState = MouseState.None
Private _Rounded AsBoolean = False
#EndRegion

#Region" Properties"
#EndRegion

#Region" MouseStates"
ProtectedOverridesSub OnMouseDown(ByVal e As MouseEventArgs)
 MyBase.OnMouseDown(e)
     State = MouseState.Down : Invalidate()
EndSub
ProtectedOverridesSub OnMouseUp(ByVal e As MouseEventArgs)
 MyBase.OnMouseUp(e)
     State = MouseState.Over : Invalidate()
EndSub
ProtectedOverridesSub OnMouseEnter(ByVal e As EventArgs)
 MyBase.OnMouseEnter(e)
     State = MouseState.Over : Invalidate()
EndSub
ProtectedOverridesSub OnMouseLeave(ByVal e As EventArgs)
 MyBase.OnMouseLeave(e)
     State = MouseState.None : Invalidate()
EndSub
#EndRegion

#Region" Function"
PrivateFunction GetConnectedSides() AsBoolean()
Dim Bool = NewBoolean(3) {False, False, False, False}

ForEach B As Control In Parent.Controls

```

```

IfTypeOf B Is FlatStickyButton Then
If B IsMeOrNot Rect.IntersectsWith(Rect) ThenContinue For
Dim A = Math.Atan2(Left() - B.Left, Top - B.Top) * 2 / Math.PI
If A \ 1 = A Then Bool(A + 1) = True
EndIf
Next

Return Bool
EndFunction

PrivateReadOnlyProperty Rect() AsRectangle
Get
ReturnNew Rectangle(Left, Top, Width, Height)
EndGet
EndProperty

#EndRegion

#Region" Colors"

<Category("Colors")> _
PublicProperty BaseColor AsColor
Get
Return _BaseColor
EndGet
Set(ByVal value As Color)
    _BaseColor = value
EndSet
EndProperty

<Category("Colors")> _
PublicProperty TextColor AsColor
Get
Return _TextColor
EndGet
Set(ByVal value As Color)
    _TextColor = value
EndSet
EndProperty

<Category("Options")> _
PublicProperty Rounded AsBoolean
Get
Return _Rounded
EndGet
Set(ByVal value AsBoolean)
    _Rounded = value
EndSet
EndProperty

#EndRegion

ProtectedOverridesSub OnResize(ByVal e AsEventArgs)
 MyBase.OnResize(e)
 'Height = 32
EndSub

ProtectedOverridesSub OnCreateControl()
 MyBase.OnCreateControl()
 'Size = New Size(112, 32)
EndSub

```

```

#EndRegion

#Region" Colors"

Private _BaseColor AsColor = _FlatColor
Private _TextColor AsColor = Color.FromArgb(243, 243, 243)

#EndRegion

SubNew()
    SetStyle(ControlStyles.AllPaintingInWmPaint Or ControlStyles.UserPaint Or
    ControlStyles.ResizeRedraw Or ControlStyles.OptimizedDoubleBuffer Or
    ControlStyles.SupportsTransparentBackColor, True)
    DoubleBuffered = True
    Size = New Size(106, 32)
    BackColor = Color.Transparent
    Font = New Font("Segoe UI", 12)
    Cursor = Cursors.Hand
EndSub

ProtectedOverridesSub OnPaint(ByVal e AsPaintEventArgs)
    B = New Bitmap(Width, Height) : G = Graphics.FromImage(B)
    W = Width : H = Height

    Dim GP AsNew GraphicsPath

    Dim GCS = GetConnectedSides()
    Dim RoundedBase = Helpers.RoundRect(0, 0, W, H, , Not (GCS(2) Or GCS(1)), Not
    (GCS(1) Or GCS(0)), Not (GCS(3) Or GCS(0)), Not (GCS(3) Or GCS(2)))
    Dim Base AsNew Rectangle(0, 0, W, H)

    With G
        .SmoothingMode = 2
        .PixelOffsetMode = 2
        .TextRenderingHint = 5
        .Clear(BackColor)
    EndWith

    SelectCase State
    Case MouseState.None
    If Rounded Then
        '-- Base
            GP = RoundedBase
            .FillPath(New SolidBrush(_BaseColor), GP)

        '-- Text
            .DrawString(Text, Font, New SolidBrush(_TextColor), Base,
            CenterSF)
        Else
            '-- Base
                .FillRectangle(New SolidBrush(_BaseColor), Base)

            '-- Text
                .DrawString(Text, Font, New SolidBrush(_TextColor), Base,
                CenterSF)
        EndIf
    Case MouseState.Over
    If Rounded Then
        '-- Base
            GP = RoundedBase
            .FillPath(New SolidBrush(_BaseColor), GP)
    EndIf
    EndSelect
EndSub

```

```

        .FillPath(New SolidBrush(Color.FromArgb(20, Color.White)),
GP)

'-- Text
        .DrawString(Text, Font, New SolidBrush(_TextColor), Base,
CenterSF)
Else
'-- Base
        .FillRectangle(New SolidBrush(_BaseColor), Base)
        .FillRectangle(New SolidBrush(Color.FromArgb(20,
Color.White)), Base)

'-- Text
        .DrawString(Text, Font, New SolidBrush(_TextColor), Base,
CenterSF)
EndIf
Case MouseState.Down
If Rounded Then
'-- Base
        GP = RoundedBase
        .FillPath(New SolidBrush(_BaseColor), GP)
        .FillPath(New SolidBrush(Color.FromArgb(20, Color.Black)),
GP)

'-- Text
        .DrawString(Text, Font, New SolidBrush(_TextColor), Base,
CenterSF)
Else
'-- Base
        .FillRectangle(New SolidBrush(_BaseColor), Base)
        .FillRectangle(New SolidBrush(Color.FromArgb(20,
Color.Black)), Base)

'-- Text
        .DrawString(Text, Font, New SolidBrush(_TextColor), Base,
CenterSF)
EndIf
EndSelect

EndWith

 MyBase.OnPaint(e)
 G.Dispose()
 e.Graphics.InterpolationMode = 7
 e.Graphics.DrawImageUnscaled(B, 0, 0)
 B.Dispose()
EndSub

EndClass

ClassFlatNumeric : InheritsControl

#Region" Variables"

Private W, H AsInteger
Private State As MouseState = MouseState.None
Private x, y AsInteger
Private _Value, _Min, _Max AsLong
Private Bool AsBoolean

#EndRegion

```

```

#Region" Properties"

PublicProperty Value AsLong
Get
Return _Value
EndGet
Set(ByVal value AsLong)
If value <= _Max And value >= _Min Then _Value = value
    Invalidate()
EndSet
EndProperty

PublicProperty Maximum AsLong
Get
Return _Max
EndGet
Set(ByVal value AsLong)
If value > _Min Then _Max = value
If _Value > _Max Then _Value = _Max
    Invalidate()
EndSet
EndProperty

PublicProperty Minimum AsLong
Get
Return _Min
EndGet
Set(ByVal value AsLong)
If value < _Max Then _Min = value
If _Value < _Min Then _Value = Minimum
    Invalidate()
EndSet
EndProperty

ProtectedOverridesSub OnMouseMove(ByVal e As MouseEventArgs)
 MyBase.OnMouseMove(e)
    x = e.Location.X
    y = e.Location.Y
    Invalidate()
If e.X < Width - 23 Then Cursor = Cursors.IBeam Else Cursor = Cursors.Hand
EndSub

ProtectedOverridesSub OnMouseDown(ByVal e As MouseEventArgs)
 MyBase.OnMouseDown(e)
 If x > Width - 21 AndAlso x < Width - 3 Then
 If y < 15 Then
 If (Value + 1) <= _Max Then _Value += 1
 Else
 If (Value - 1) >= _Min Then _Value -= 1
 EndIf
 Else
        Bool = Not Bool
        Focus()
    EndIf
    Invalidate()
EndSub

ProtectedOverridesSub OnKeyPress(ByVal e As KeyPressEventArgs)
 MyBase.OnKeyPress(e)
 Try
 If Bool Then _Value = CStr(CStr(_Value) & e.KeyChar.ToString())
 If _Value > _Max Then _Value = _Max

```

```

        Invalidate()
Catch : EndTry
EndSub

ProtectedOverridesSub OnKeyDown(ByVal e As KeyEventArgs)
 MyBase.OnKeyDown(e)
If e.KeyCode = Keys.Back Then
    Value = 0
EndIf
EndSub

ProtectedOverridesSub OnResize(ByVal e As EventArgs)
 MyBase.OnResize(e)
 Height = 30
EndSub

#Region" Colors"

<Category("Colors")>
PublicProperty BaseColor AsColor
Get
Return _BaseColor
EndGet
Set(ByVal value As Color)
    _BaseColor = value
EndSet
EndProperty

<Category("Colors")>
PublicProperty ButtonColor AsColor
Get
Return _ButtonColor
EndGet
Set(ByVal value As Color)
    _ButtonColor = value
EndSet
EndProperty

#EndRegion

#EndRegion

#Region" Colors"

Private _BaseColor As Color = Color.FromArgb(45, 47, 49)
Private _ButtonColor As Color = _FlatColor

#EndRegion

SubNew()
    SetStyle(ControlStyles.AllPaintingInWmPaint Or ControlStyles.UserPaint Or
    - ControlStyles.ResizeRedraw Or ControlStyles.OptimizedDoubleBuffer Or _
    ControlStyles.SupportsTransparentBackColor, True)
    DoubleBuffered = True
    Font = New Font("Segoe UI", 10)
    BackColor = Color.FromArgb(60, 70, 73)
    ForeColor = Color.White
    _Min = 0
    _Max = 999999
EndSub

```

```

ProtectedOverridesSub OnPaint(ByVal e As PaintEventArgs)
    B = New Bitmap(Width, Height) : G = Graphics.FromImage(B)
    W = Width : H = Height

Dim Base AsNew Rectangle(0, 0, W, H)

With G
    .SmoothingMode = 2
    .PixelOffsetMode = 2
    .TextRenderingHint = 5
    .Clear(BackColor)

    '-- Base
    .FillRectangle(New SolidBrush(_BaseColor), Base)
    .FillRectangle(New SolidBrush(_ButtonColor), New Rectangle(Width - 24,
0, 24, H))

    '-- Add
    .DrawString("+", New Font("Segoe UI", 12), Brushes.White, New
Point(Width - 12, 8), CenterSF)
    '-- Subtract
    .DrawString("-", New Font("Segoe UI", 10, FontStyle.Bold),
Brushes.White, New Point(Width - 12, 22), CenterSF)

    '-- Text
    .DrawString(Value, Font, Brushes.White, New Rectangle(5, 1, W, H), New
StringFormat() With {.LineAlignment = StringAlignment.Center})
EndWith

 MyBase.OnPaint(e)
    G.Dispose()
    e.Graphics.InterpolationMode = 7
    e.Graphics.DrawImageUnscaled(B, 0, 0)
    B.Dispose()
EndSub

EndClass

ClassFlatListBox : InheritsControl

#Region" Variables"

PrivateWithEvents ListBx AsNew ListBox
Private _items AsString() = {"}

#EndRegion

#Region" Properties"

<Category("Options")> _
PublicProperty items AsString()
Get
Return _items
EndGet
Set(ByVal value AsString())
    _items = value
    ListBx.Items.Clear()
    ListBx.Items.AddRange(value)
    Invalidate()
EndSet
EndProperty

```

```

<Category("Colors")> -
PublicProperty SelectedColor AsColor
Get
Return _SelectedColor
EndGet
Set(ByName value As Color)
    _SelectedColor = value
EndSet
EndProperty

PublicReadOnlyProperty SelectedItem() AsString
Get
Return ListBx.SelectedItem
EndGet
EndProperty

PublicReadOnlyProperty SelectedIndex() AsInteger
Get
Return ListBx.SelectedIndex
If ListBx.SelectedIndex < 0 ThenExit Property
EndGet
EndProperty

PublicSub Clear()
    ListBx.Items.Clear()
EndSub

PublicSub ClearSelected()
For i AsInteger = (ListBx.SelectedItems.Count - 1) To 0 Step -1
    ListBx.Items.Remove(ListBx.SelectedItems(i))
Next
EndSub

Sub Drawitem( ByVal sender AsObject, ByVal e As DrawItemEventArgs) Handles
ListBx.DrawItem
If e.Index < 0 ThenExit Sub
    e.DrawBackground()
    e.DrawFocusRectangle()

    e.Graphics.SmoothingMode = 2
    e.Graphics.PixelOffsetMode = 2
    e.Graphics.InterpolationMode = 7
    e.Graphics.TextRenderingHint = 5

If InStr(e.State.ToString, "Selected,") > 0 Then'-- if selected
'-- Base
    e.Graphics.FillRectangle(New SolidBrush(_SelectedColor), New
    Rectangle(e.Bounds.X, e.Bounds.Y, e.Bounds.Width, e.Bounds.Height))

'-- Text
    e.Graphics.DrawString(" "& ListBx.Items(e.Index).ToString(), New
    Font("Segoe UI", 8), Brushes.White, e.Bounds.X, e.Bounds.Y + 2)
Else
'-- Base
    e.Graphics.FillRectangle(New SolidBrush(Color.FromArgb(51, 53, 55)), New
    Rectangle(e.Bounds.X, e.Bounds.Y, e.Bounds.Width, e.Bounds.Height))

'-- Text
    e.Graphics.DrawString(" "& ListBx.Items(e.Index).ToString(), New
    Font("Segoe UI", 8), Brushes.White, e.Bounds.X, e.Bounds.Y + 2)
EndIf

```

```

        e.Graphics.Dispose()
EndSub

ProtectedOverridesSub OnCreateControl()
 MyBase.OnCreateControl()
 IfNot Controls.Contains(ListBx) Then
     Controls.Add(ListBx)
 EndIf
EndSub

Sub AddRange(ByVal items AsObject())
    ListBx.Items.Remove("")
    ListBx.Items.AddRange(items)
EndSub

Sub AddItem(ByVal item AsObject)
    ListBx.Items.Remove("")
    ListBx.Items.Add(item)
EndSub

#EndRegion

#Region " Colors"

Private BaseColor As Color = Color.FromArgb(45, 47, 49)
Private _SelectedColor As Color = _FlatColor

#EndRegion

SubNew()
    SetStyle(ControlStyles.AllPaintingInWmPaint Or ControlStyles.UserPaint Or
    - ControlStyles.ResizeRedraw Or ControlStyles.OptimizedDoubleBuffer,
    True)
    DoubleBuffered = True

    ListBx.DrawMode = Windows.Forms.DrawMode.OwnerDrawFixed
    ListBx.ScrollAlwaysVisible = False
    ListBx.HorizontalScrollbar = False
    ListBx.BorderStyle = BorderStyle.None
    ListBx.BackColor = BaseColor
    ListBx.ForeColor = Color.White
    ListBx.Location = New Point(3, 3)
    ListBx.Font = New Font("Segoe UI", 8)
    ListBx.ItemHeight = 20
    ListBx.Items.Clear()
    ListBx.IntegralHeight = False

    Size = New Size(131, 101)
    BackColor = BaseColor
EndSub

ProtectedOverridesSub OnPaint(ByVal e As PaintEventArgs)
    B = New Bitmap(Width, Height) : G = Graphics.FromImage(B)

    Dim Base AsNew Rectangle(0, 0, Width, Height)

    With G
        .SmoothingMode = 2
        .PixelOffsetMode = 2
        .TextRenderingHint = 5
    EndWith
EndSub

```

```

        .Clear(BackColor)

'-- Size
    ListBx.Size = New Size(Width - 6, Height - 2)

'-- Base
    .FillRectangle(New SolidBrush(BaseColor), Base)
EndWith

 MyBase.OnPaint(e)
    G.Dispose()
    e.Graphics.InterpolationMode = 7
    e.Graphics.DrawImageUnscaled(B, 0, 0)
    B.Dispose()
EndSub

EndClass

ClassFlatContextMenuStrip : InheritsContextMenuStrip

ProtectedOverridesSub OnTextChanged(ByVal e As EventArgs)
MyBase.OnTextChanged(e) : Invalidate()
EndSub

SubNew()
MyBase.New()
    Renderer = New ToolStripProfessionalRenderer(New TColorTable())
    ShowImageMargin = False
    ForeColor = Color.White
    Font = New Font("Segoe UI", 8)
EndSub

ProtectedOverridesSub OnPaint(ByVal e As PaintEventArgs)
MyBase.OnPaint(e)
    e.Graphics.TextRenderingHint = 5
EndSub

ClassTColorTable : Inherits ProfessionalColorTable

#Region" Properties"

#Region" Colors"

<Category("Colors")> _
PublicProperty _BackColor As Color
Get
Return BackColor
EndGet
Set(ByVal value As Color)
    BackColor = value
EndSet
EndProperty

<Category("Colors")> _
PublicProperty _CheckedColor As Color
Get
Return CheckedColor
EndGet
Set(ByVal value As Color)
    CheckedColor = value
EndSet
EndProperty

```

```
<Category("Colors")> _
PublicProperty _BorderColor As Color
Get
Return BorderColor
EndGet
Set(ByVal value As Color)
    BorderColor = value
EndSet
EndProperty

#EndRegion

#Region" Colors"

Private BackColor As Color = Color.FromArgb(45, 47, 49)
Private CheckedColor As Color = _FlatColor
Private BorderColor As Color = Color.FromArgb(53, 58, 60)

#EndRegion

#Region" Overrides"

PublicOverridesReadOnlyProperty ButtonSelectedBorder As Color
Get
Return BackColor
EndGet
EndProperty
PublicOverridesReadOnlyProperty CheckBackground() As Color
Get
Return CheckedColor
EndGet
EndProperty
PublicOverridesReadOnlyProperty CheckPressedBackground() As Color
Get
Return CheckedColor
EndGet
EndProperty
PublicOverridesReadOnlyProperty CheckSelectedBackground() As Color
Get
Return CheckedColor
EndGet
EndProperty
PublicOverridesReadOnlyProperty ImageMarginGradientBegin() As Color
Get
Return CheckedColor
EndGet
EndProperty
PublicOverridesReadOnlyProperty ImageMarginGradientEnd() As Color
Get
Return CheckedColor
EndGet
EndProperty
PublicOverridesReadOnlyProperty ImageMarginGradientMiddle() As Color
Get
Return CheckedColor
EndGet
EndProperty
PublicOverridesReadOnlyProperty MenuBorder() As Color
Get
```

```

Return BorderColor
EndGet
EndProperty
PublicOverridesReadOnlyProperty MenuItemBorder() As Color
Get
Return BorderColor
EndGet
EndProperty
PublicOverridesReadOnlyProperty MenuItemSelected() As Color
Get
Return CheckedColor
EndGet
EndProperty
PublicOverridesReadOnlyProperty SeparatorDark() As Color
Get
Return BorderColor
EndGet
EndProperty
PublicOverridesReadOnlyProperty ToolStripDropDownBackground() As Color
Get
Return BackColor
EndGet
EndProperty

#EndRegion

EndClass
EndClass

<DefaultEvent("Scroll")>ClassFlatTrackBar : InheritsControl

#Region" Variables"

Private W, H AsInteger
Private Val AsInteger
Private Bool AsBoolean
Private Track As Rectangle
Private Knob As Rectangle
Private Style_ As _Style

#EndRegion

#Region" Properties"

#EndRegion" Mouse States"

ProtectedOverridesSub OnMouseDown(ByVal e As MouseEventArgs)
 MyBase.OnMouseDown(e)
 If e.Button = Windows.Forms.MouseButtons.Left Then
     Val = CInt((_Value - _Minimum) / (_Maximum - _Minimum) * (Width - 11))
     Track = New Rectangle(Val, 0, 10, 20)

     Bool = Track.Contains(e.Location)
 EndIf
 EndSub

ProtectedOverridesSub OnMouseMove(ByVal e As MouseEventArgs)
 MyBase.OnMouseMove(e)
 If Bool AndAlso e.X > -1 AndAlso e.X < (Width + 1) Then
     Value = _Minimum + CInt((_Maximum - _Minimum) * (e.X / Width))
 EndIf

```

```

EndSub

ProtectedOverridesSub OnMouseUp(ByVal e As MouseEventArgs)
MyBase.OnMouseUp(e) : Bool = False
EndSub

#EndRegion

#Region" Styles"

<Flags()> _
Enum_Style
    Slider
    Knob
EndEnum

PublicProperty Style As_Style
Get
Return Style_
EndGet
Set(ByVal value As _Style)
    Style_ = value
EndSet
EndProperty

#EndRegion

#Region" Colors"

<Category("Colors")> _
PublicProperty TrackColor AsColor
Get
Return _TrackColor
EndGet
Set(ByVal value As Color)
    _TrackColor = value
EndSet
EndProperty

<Category("Colors")> _
PublicProperty HatchColor AsColor
Get
Return _HatchColor
EndGet
Set(ByVal value As Color)
    _HatchColor = value
EndSet
EndProperty

#EndRegion

Event Scroll(ByVal sender AsObject)
Private _Minimum AsInteger
PublicProperty Minimum AsInteger
Get
Return Minimum
EndGet
Set(ByVal value AsInteger)
If value < 0 Then
EndIf

    _Minimum = value

```

```

If value > _Value Then _Value = value
If value > _Maximum Then _Maximum = value
    Invalidate()
EndSet
EndProperty
Private _Maximum AsInteger = 10
PublicProperty Maximum AsInteger
Get
Return _Maximum
EndGet
Set(ByVal value AsInteger)
If value < 0 Then
EndIf

    _Maximum = value
If value < _Value Then _Value = value
If value < _Minimum Then _Minimum = value
    Invalidate()
EndSet
EndProperty
Private _Value AsInteger
PublicProperty Value AsInteger
Get
Return _Value
EndGet
Set(ByVal value AsInteger)
If value = _Value ThenReturn

If value > _Maximum OrElse value < _Minimum Then
EndIf

    _Value = value
    Invalidate()
RaiseEvent Scroll(Me)
EndSet
EndProperty
Private _ShowValue AsBoolean = False
PublicProperty ShowValue AsBoolean
Get
Return _ShowValue
EndGet
Set(ByVal value AsBoolean)
    _ShowValue = value
EndSet
EndProperty

ProtectedOverridesSub OnKeyDown(ByVal e As KeyEventArgs)
 MyBase.OnKeyDown(e)
If e.KeyCode = Keys.Subtract Then
If Value = 0 ThenExit Sub
    Value -= 1
ElseIf e.KeyCode = Keys.Add Then
If Value = _Maximum ThenExit Sub
    Value += 1
EndIf
EndSub

ProtectedOverridesSub OnTextChanged(ByVal e As EventArgs)
 MyBase.OnTextChanged(e) : Invalidate()
EndSub

```

```

ProtectedOverridesSub OnResize(ByVal e As EventArgs)
 MyBase.OnResize(e)
 Height = 23
EndSub

#EndRegion

#Region" Colors"

Private BaseColor As Color = Color.FromArgb(45, 47, 49)
Private _TrackColor As Color = _FlatColor
Private SliderColor As Color = Color.FromArgb(25, 27, 29)
Private _HatchColor As Color = Color.FromArgb(23, 148, 92)

#EndRegion

SubNew()
 SetStyle(ControlStyles.AllPaintingInWmPaint Or ControlStyles.UserPaint Or
 -
 ControlStyles.ResizeRedraw Or
 ControlStyles.OptimizedDoubleBuffer, True)
 DoubleBuffered = True
 Height = 18

 BackColor = Color.FromArgb(60, 70, 73)
EndSub

ProtectedOverridesSub OnPaint(ByVal e As PaintEventArgs)
 B = New Bitmap(Width, Height) : G = Graphics.FromImage(B)
 W = Width - 1 : H = Height - 1

Dim Base AsNew Rectangle(1, 6, W - 2, 8)
Dim GP, GP2 AsNew GraphicsPath

With G
 .SmoothingMode = 2
 .PixelOffsetMode = 2
 .TextRenderingHint = 5
 .Clear(BackColor)

'-- Value
 Val = CInt((_Value - _Minimum) / (_Maximum - _Minimum) * (W - 10))
 Track = New Rectangle(Val, 0, 10, 20)
 Knob = New Rectangle(Val, 4, 11, 14)

'-- Base
 GP.AddRectangle(Base)
 .SetClip(GP)
 .FillRectangle(New SolidBrush(BaseColor), New Rectangle(0, 7, W, 8))
 .FillRectangle(New SolidBrush(_TrackColor), New Rectangle(0, 7,
 Track.X + Track.Width, 8))
 .ResetClip()

'-- Hatch Brush
 Dim HB AsNew HatchBrush(HatchStyle.Plaid, HatchColor, _TrackColor)
 .FillRectangle(HB, New Rectangle(-10, 7, Track.X + Track.Width, 8))

'-- Slider/Knob
 SelectCase Style
 Case _Style.Slider
 GP2.AddRectangle(Track)
 .FillPath(New SolidBrush(SliderColor), GP2)

```

```

Case _Style.Knob
    GP2.AddEllipse(Knob)
        .FillPath(New SolidBrush(SliderColor), GP2)
EndSelect

'-- Show the value
If ShowValue Then
    .DrawString(Value, New Font("Segoe UI", 8), Brushes.White, New
    Rectangle(1, 6, W, H), New StringFormat()
    With {.Alignment = StringAlignment.Far, .LineAlignment = StringAlignment.Far})
EndIf
EndWith

 MyBase.OnPaint(e)
    G.Dispose()
    e.Graphics.InterpolationMode = 7
    e.Graphics.DrawImageUnscaled(B, 0, 0)
    B.Dispose()
EndSub
EndClass

ClassFlatStatusBar : InheritsControl

#Region" Variables"

Private W, H AsInteger
Private _ShowTimeDate AsBoolean = False

#EndRegion

#Region" Properties"

ProtectedOverridesSub CreateHandle()
MyBase.CreateHandle()
    Dock = DockStyle.Bottom
EndSub

ProtectedOverridesSub OnTextChanged(ByVal e As EventArgs)
MyBase.OnTextChanged(e) : Invalidate()
EndSub

#Region" Colors"

<Category("Colors")>
PublicProperty BaseColor AsColor
Get
Return _BaseColor
EndGet
Set(ByVal value As Color)
    _BaseColor = value
EndSet
EndProperty

<Category("Colors")>
PublicProperty TextColor AsColor
Get
Return _TextColor
EndGet
Set(ByVal value As Color)
    _TextColor = value
EndSet
EndProperty

```

```

<Category("Colors")>
PublicProperty RectColor AsColor
Get
Return _RectColor
EndGet
Set(ByVal value As Color)
    _RectColor = value
EndSet
EndProperty

#EndRegion

PublicProperty ShowTimeDate AsBoolean
Get
Return _ShowTimeDate
EndGet
Set(ByVal value AsBoolean)
    _ShowTimeDate = value
EndSet
EndProperty

Function GetTimeDate() AsString
Return DateTime.Now.Date & " " & DateTime.Now.Hour & ":" & DateTime.Now.Minute
EndFunction

#EndRegion

#Region" Colors"

Private _BaseColor As Color = Color.FromArgb(45, 47, 49)
Private _TextColor As Color = Color.White
Private _RectColor As Color = _FlatColor

#EndRegion

SubNew()
    SetStyle(ControlStyles.AllPaintingInWmPaint Or ControlStyles.UserPaint Or
    - ControlStyles.ResizeRedraw Or
ControlStyles.OptimizedDoubleBuffer, True)
    DoubleBuffered = True
    Font = New Font("Segoe UI", 8)
    ForeColor = Color.White
    Size = New Size(Width, 20)
EndSub

ProtectedOverridesSub OnPaint(ByVal e As PaintEventArgs)
    B = New Bitmap(Width, Height) : G = Graphics.FromImage(B)
    W = Width : H = Height

    Dim Base AsNew Rectangle(0, 0, W, H)

    With G
        .SmoothingMode = 2
        .PixelOffsetMode = 2
        .TextRenderingHint = 5
        .Clear(BaseColor)

        '-- Base
        .FillRectangle(New SolidBrush(BaseColor), Base)
    EndWith
EndSub

```

```

'-- Text
    .DrawString(Text, Font, Brushes.White, New Rectangle(10, 4, W, H),
NearSF)

'-- Rectangle
    .FillRectangle(New SolidBrush(_RectColor), New Rectangle(4, 4, 4, 14))

'-- TimeDate
If ShowTimeDate Then
    .DrawString(GetTimeDate, Font, New SolidBrush(_TextColor), New
Rectangle(-4, 2, W, H), New StringFormat() _
With {.Alignment = StringAlignment.Far, .LineAlignment = StringAlignment.Center})
EndIf
EndWith

 MyBase.OnPaint(e)
    G.Dispose()
    e.Graphics.InterpolationMode = 7
    e.Graphics.DrawImageUnscaled(B, 0, 0)
    B.Dispose()
EndSub
EndClass

ClassFlatButton : InheritsLabel

ProtectedOverridesSub OnTextChanged(ByVal e As EventArgs)
MyBase.OnTextChanged(e) : Invalidate()
EndSub

SubNew()
    SetStyle(ControlStyles.SupportsTransparentBackColor, True)
    Font = New Font("Segoe UI", 8)
    ForeColor = Color.White
    BackColor = Color.Transparent
    Text = Text
EndSub

EndClass

ClassFlatTreeView : InheritsTreeView

#Region" Variables"

Private State As TreeNodeStates

#EndRegion

#Region" Properties"

ProtectedOverridesSub OnDrawNode(ByVal e As DrawTreeNodeEventArgs)
Try
Dim Bounds AsNew Rectangle(e.Bounds.Location.X, e.Bounds.Location.Y,
e.Bounds.Width, e.Bounds.Height)
'e.Node.Nodes.Item.
SelectCase State
Case TreeNodeStates.Default
    e.Graphics.FillRectangle(Brushes.Red, Bounds)
    e.Graphics.DrawString(e.Node.Text, New Font("Segoe UI", 8),
Brushes.LimeGreen, New Rectangle(Bounds.X + 2, Bounds.Y + 2, Bounds.Width,
Bounds.Height), NearSF)
    Invalidate()
Case TreeNodeStates.Checked

```

```

        e.Graphics.FillRectangle(Brushes.Green, Bounds)
        e.Graphics.DrawString(e.Node.Text, New Font("Segoe UI", 8),
Brushes.Black, New Rectangle(Bounds.X + 2, Bounds.Y + 2, Bounds.Width,
Bounds.Height), NearSF)
        Invalidate()
Case TreeNodeStates.Selected
        e.Graphics.FillRectangle(Brushes.Green, Bounds)
        e.Graphics.DrawString(e.Node.Text, New Font("Segoe UI", 8),
Brushes.Black, New Rectangle(Bounds.X + 2, Bounds.Y + 2, Bounds.Width,
Bounds.Height), NearSF)
        Invalidate()
EndSelect

Catch ex As Exception
        MsgBox(ex.Message)
EndTry
 MyBase.OnDrawNode(e)
EndSub

#EndRegion
#Region" Colors"

Private _BaseColor As Color = Color.FromArgb(45, 47, 49)
Private _LineColor As Color = Color.FromArgb(25, 27, 29)

#EndRegion
SubNew()

        SetStyle(ControlStyles.AllPaintingInWmPaint Or ControlStyles.UserPaint Or
-
ControlStyles.ResizeRedraw Or
ControlStyles.OptimizedDoubleBuffer, True)
        DoubleBuffered = True

        BackColor = _BaseColor
        ForeColor = Color.White
        LineColor = _LineColor
        DrawMode = TreeViewDrawMode.OwnerDrawAll
EndSub

ProtectedOverridesSub OnPaint(ByVal e As PaintEventArgs)
        B = New Bitmap(Width, Height) : G = Graphics.FromImage(B)

        Dim Base AsNew Rectangle(0, 0, Width, Height)

        With G
            .SmoothingMode = 2
            .PixelOffsetMode = 2
            .TextRenderingHint = 5
            .Clear(BackColor)

            .FillRectangle(New SolidBrush(_BaseColor), Base)
            .DrawString(Text, New Font("Segoe UI", 8), Brushes.Black, New
Rectangle(Bounds.X + 2, Bounds.Y + 2, Bounds.Width, Bounds.Height), NearSF)
        EndWith
        MyBase.OnPaint(e)
        G.Dispose()
        e.Graphics.InterpolationMode = 7
        e.Graphics.DrawImageUnscaled(B, 0, 0)
        B.Dispose()
EndSub
EndClass

```

## **APPENDIX D**

### **Letter to the Panelist**



Republic of the Philippines  
**JOSE RIZAL MEMORIAL STATE UNIVERSITY**  
 The Premier University in Zamboanga Del Norte  
 Main Campus, Dapitan city



May 17, 2017

**ENGR. ANGELITO M. PUNZAL**  
 Computer Scinece Department  
 Jose Rizal Memorial State University

Sir,

The undersigned are the 4<sup>th</sup> year BS Computer Science who presently conducting the study entitled "**Intelligent IP Address Locator via Google Map**". This is in partial fulfillment of the subject Project Thesis.

We would like to request your present as one of the panelists on May 17, 2017 at Research Extension Room, Dapitan City.

Hoping for your kind response on this request.

Respectfully yours,

Jay Marie A. Egoogan  
 Lea E. Monopollo  
 Khimberly R. Ebe

Approved:

**ENGR. ANGELITO M. PUNZAL, MSIT**  
 Adviser/Instructor



Republic of the Philippines  
**JOSE RIZAL MEMORIAL STATE UNIVERSITY**  
The Premier University in Zamboanga Del Norte  
Main Campus, Dapitan city



May 17, 2017

**ENGR. JOSEPH AURELIUS P. JACINTO**  
Computer Scinece Department  
Jose Rizal Memorial State University

Sir,

The undersigned are the 4<sup>th</sup> year BS Computer Science who presently conducting the study entitled "**Intelligent IP Address Locator via Google Map**". This is in partial fulfillment of the subject Project Thesis.

We would like to request your present as one of the panelists on May 17, 2017 at Research Extension Room, Dapitan City.

Hoping for your kind response on this request.

Respectfully yours,

Jay Marie A. Egoogan  
Lea E. Monopollo  
Khimberly R. Ebe

Approved:

**ENGR. ANGELITO M. PUNZAL, MSIT**  
Adviser/Instructor



Republic of the Philippines  
**JOSE RIZAL MEMORIAL STATE UNIVERSITY**  
The Premier University in Zamboanga Del Norte  
Main Campus, Dapitan city



May 17, 2017

**PROF. JHON D. SAGAPSAPAN**

Computer Scinece Department  
Jose Rizal Memorial State University

Sir,

The undersigned are the 4<sup>th</sup> year BS Computer Science who presently conducting the study entitled "**Intelligent IP Address Locator via Google Map**". This is in partial fulfillment of the subject Project Thesis.

We would like to request your present as one of the panelists on May 17, 2017 at Research Extension Room, Dapitan City.

Hoping for your kind response on this request.

Respectfully yours,

Jay Marie A. Egoogan  
Lea E. Monopollo  
Khimberly R. Ebe

Approved:

**ENGR. ANGELITO M. PUNZAL, MSIT**  
Adviser/Instructor



Republic of the Philippines  
**JOSE RIZAL MEMORIAL STATE UNIVERSITY**  
The Premier University in Zamboanga Del Norte  
Main Campus, Dapitan city



May 17, 2017

**ARMANDO T. SAGUIN JR.**  
Computer Scinece Department  
Jose Rizal Memorial State University

Sir,

The undersigned are the 4<sup>th</sup> year BS Computer Science who presently conducting the study entitled "**Intelligent IP Address Locator via Google Map**". This is in partial fulfillment of the subject Project Thesis.

We would like to request your present as one of the panelists on May 17, 2017 at Research Extension Room, Dapitan City.

Hoping for your kind response on this request.

Respectfully yours,

Jay Marie A. Egoogan  
Lea E. Monopollo  
Khimberly R. Ebe

Approved:

**ENGR. ANGELITO M. PUNZAL, MSIT**  
Adviser/Instructor



Republic of the Philippines  
**JOSE RIZAL MEMORIAL STATE UNIVERSITY**  
The Premier University in Zamboanga Del Norte  
Main Campus, Dapitan city



May 17, 2017

**PROF. ED NEIL O. MARATAS**  
Computer Scinece Department  
Jose Rizal Memorial State University

Sir,

The undersigned are the 4<sup>th</sup> year BS Computer Science who presently conducting the study entitled "**Intelligent IP Address Locator via Google Map**". This is in partial fulfillment of the subject Project Thesis.

We would like to request your present as one of the panelists on May 17, 2017 at Research Extension Room, Dapitan City.

Hoping for your kind response on this request.

Respectfully yours,

Jay Marie A. Egoogan  
Lea E. Monopollo  
Khimberly R. Ebe

Approved:

**ENGR. ANGELITO M. PUNZAL, MSIT**  
Adviser/Instructor

## **APPENDIX E**

### **Evaluation Letter**



Republic of the Philippines  
**JOSE RIZAL MEMORIAL STATE UNIVERSITY**  
The Premier University in Zamboanga Del Norte  
Main Campus, Dapitan city



## **EVALUATION LETTER**

**Dear Respondents,**

The undersigned is gathering data for thesis entitled, "**Intelligent IP Address Locator via Google Map**". Please evaluate the software with all your sincerity and honesty.

Your cooperation will contribute to the data in this study.

Thank you very much.

Very truly yours,

Jay Marie A. Egoogan

Lea E. Monopollo

Khimberly R. Ebe

## INTELLIGENT IP ADDRESS LOCATOR VIA GOOGLE MAP

### Evaluation Sheet

Name of Evaluator/Rater: \_\_\_\_\_

Position /Job Description: \_\_\_\_\_

**Direction:** Please evaluate the developed Intelligent IP Address Locator via Google Map system software along the levels of functionality, reliability, usability, efficiency, maintainability, and portability, Description of each level is indicated below.

**Functionality** pertains to the sum or any aspect of what Intelligent IP Address Locator via Google Map can do for the user. Indicate the functionality level of the system software by putting a check in the box that fits your response. The numbers are coded as follows:

- 5- Very Much Functional
- 4- Much Functional
- 3- Functional
- 2- Fairly Functional
- 1 -Not Functional

FUNCTIONALITY	5	4	3	2	1
1. Intended use of the software					
2. Data manipulation					
3. Compliance of end-user needs					
4. Security of system data					
5. Compatibility of other system					
6. Speed in data processing					
7. Fitness of its intended use					
8. Minimization of its run-time error					
9. Detection of error					
10. Data storage					

**Reliability** refers to the ability of the Intelligent IP Address Locator via Google Map to consistently perform its intended or required function or mission on demand and without degradation or failure. Indicate the reliability level of the system software by putting a check box that fits your numbers are coded as follows:

- 5- Very Much Reliable
- 4- Much Reliable
- 3 - Reliable
- 2- Fairly Reliable
- 1 -Not Reliable

RELIABILITY	5	4	3	2	1
1. Error tolerance					
2. Ease in data recovery					
3. Program security					
4. Data security					
5. Creation of back-up system					
6. Accuracy of data capture					
7. Accuracy of Results					
8. Data storage volume					
9. Understand ability of output					
10. Completeness of the system					

**Usability** pertains to the user's total satisfaction received from using the proposed Intelligent IP Address Locator via Google Map. To determine also the usability level of the system software, put a check in the box that fits your response. The numbers are coded as follows:

- 5- Very Much Usable
- 4- Much Usable
- 3- Usable
- 2- Fairly Usable
- 1 - Not Usable

USABILITY	5	4	3	2	1
1. User friendly program					
2. Quick driven program					
3. Simple manipulation features					
4. Wrong key input errors detection					
5. Wrong time input errors detection					
6. Data storage					
7. Data retrieval					
8. Data edit/correction					
9. Tolerable difficulty level					
10. Production of data output					

**Efficiency** refers to the Intelligent IP Address Locator via Google Map of what is actually produced or performed with what can be achieved in contrast with the traditional employee's attendance system. To find also the efficiency level of the system software, put a check in the box that fits your response. The numbers are coded as follows:

- 5- Very Much Efficient
- 4- Much Efficient
- 3 – Efficient
- 2- Fairly Efficient
- 1 -Not Efficient

EFFICIENCY	5	4	3	2	1
1. Support on minimum facilities					
2. Support on minimum requirements					
3. Provision of configurable automation					
4. Support on business workflow process					
5. Support on number of user's					
6. Speed of navigation and production of outputs					
7. Speed of data capture and retrieval					
8. Hardware utilization					
9. Support on interfacing with other devices					
10. Compatibility with interfaced devices					

**Maintainability** pertains to be characteristics of the Intelligent IP Address Locator via Google Map and its installation which determines the probability that its failure can be restored to its normal operable state within a given timeframe, using the prescribed practices and procedures. That involves two main components, namely: serviceability which includes the ease of conducting scheduled inspections and servicing, and reparability which describes the ease of restoring service after a failure. To evaluate the maintainability level of the system software, put a check in the box that fits your response. The numbers are coded as follows:

- 5- Very Much Maintainable
- 4- Much Maintainable
- 3- Maintainable
- 2- Fairly Maintainable
- 1 -Not Maintainable

MAINTAINABILITY	5	4	3	2	1
1. Modification of the system software					
2. Change of software capabilities					
3. Increase program capabilities					
4. Improving performance					
5. Correction of program defects					
6. Accessibility for maintenance					
7. Configuration of system operation					
8. Flexibility for system modification					
9. Compliance of concurrent system requirements					
10. Advance feature for recent technology					

**Portability** refers to the ability of the Intelligent IP Address Locator via Google Map to match the actual expected performance of the system being measured. Please indicate also the level of portability of the system software by putting a check in the box that fits your response. The numbers are coded as follows:

- 5- Very Much Portable
- 4- Much Portable
- 3- Portable
- 2- Fairly Portable
- 1 -Not Portable

PORATABILITY	5	4	3	2	1
1. Modification of the system					
2. Adaptability to other environment					
3. Adaptability to other applications					
4. Flexibility to other settings					
5. Support in any form of network communication					
6. Adaptability to new version of system requirements					
7. System supports on maximum hardware requirements					
8. User capability/capacity					
9. Its intended application					
10. Its intended design					

**Remarks:** \_\_\_\_\_

**Signature of Evaluator/Rater:** \_\_\_\_\_

**Date:** \_\_\_\_\_

### Table /Tally of Result

Numerical Rating	Corresponding Rating
4.21 – 5.00	Very Much functional
3.41 – 4.20	Much Functional
2.61 – 3.40	Moderately Functional
1.81 – 2.60	Less Functional
1.00 – 1.80	Not Reliable Functional

**Functionality.** Pertains to the sum or any aspect of what Intelligent IP Address Locator via Google Map can do for the user. Indicate the functionality level of the system software by putting a check in the box that fits your response. The numbers are coded as follows:

**Table. Tally Result Functionality**

FUNCTIONALITY	5	4	3	2	1	Mean	Description
1. Intended use of the software						<b>4.53</b>	Very Much Functional
2. Data manipulation						<b>4.3</b>	Very Much Functional
3. Compliance of end-user needs						<b>4.3</b>	Very Much Functional
4. Security of system data						<b>4.3</b>	Very Much Functional
5. Compatibility of other system						<b>4.4</b>	Very Much Functional
6. Speed in data processing						<b>4.5</b>	Very Much Functional
7. Fitness of its intended use						<b>4.4</b>	Very Much Functional
8. Minimization of its run-time error						<b>4.5</b>	Very Much Functional
9. Detection of error						<b>4.5</b>	Very Much Functional
10. Data storage						<b>4.6</b>	Very Much Functional
<b>Average Mean</b>						<b>4.43</b>	<b>Very Much Functional</b>

**Reliability.** The ability of a system to perform its required functions under stated conditions whenever required functions under stated conditions whenever required – having a long mean time between failures.

**Table. Tally Result Reliability**

RELIABILITY	5	4	3	2	1	Mean	Description
1. Error tolerance						<b>4.4</b>	Very Much Reliable
2. Ease in data recovery						<b>4.4</b>	Very Much Reliable
3. Program security						<b>4.5</b>	Very Much Reliable
4. Data security						<b>4.5</b>	Very Much Reliable
5. Creation of back-up system						<b>4.5</b>	Very Much Reliable
6. Accuracy of data capture						<b>4.1</b>	Much Reliable
7. Accuracy of Results						<b>4.3</b>	Very Much Reliable
8. Data storage volume						<b>4.3</b>	Very Much Reliable
9. Understand ability of output						<b>4.3</b>	Very Much Reliable
10. Completeness of the system						<b>4.5</b>	Very Much Reliable
<b>Average</b>						<b>4.38</b>	<b>Very Much Reliable</b>

**Usability.** The software application can be used by specified consumers to achieve quantified objectives with effectiveness, efficiency, and satisfaction in a quantified context of use.

**Table Tally Result Usability**

USABILITY	5	4	3	2	1	Mean	Description
1. User friendly program						<b>4.6</b>	Very Much Usable
2. Quick driven program						<b>4.3</b>	Very Much Usable
3. Simple manipulation features						<b>4.2</b>	Very Much Usable
4. Wrong key input errors detection						<b>4.2</b>	Very Much Usable
5. Wrong time input errors detection						<b>4.3</b>	Very Much Usable
6. Data storage						<b>4.2</b>	Very Much Usable
7. Data retrieval						<b>4.4</b>	Very Much Usable
8. Data edit/correction						<b>4.4</b>	Very Much Usable
9. Tolerable difficulty level						<b>4.4</b>	Very Much Usable
10. Production of data output						<b>4.3</b>	Very Much Usable
<b>Average</b>						<b>4.33</b>	<b>Very Much Usable</b>

**Efficiency.** Software that being developed to provide a reliable, secure and efficient method of recording the IP Addresses.

**Table Tally Result Efficiency**

EFFICIENCY	5	4	3	2	1	Mean	Description
1. Support on minimum facilities						<b>4.5</b>	Very Much Efficient
2. Support on minimum requirements						<b>4.4</b>	Very Much Efficient
3. Provision of configurable automation						<b>4.4</b>	Very Much Efficient
4. Support on business workflow process						<b>4.4</b>	Very Much Efficient
5. Support on number of user's						<b>4.4</b>	Very Much Efficient
6. Speed of navigation and production of outputs						<b>4.3</b>	Very Much Efficient
7. Speed of data capture and retrieval						<b>4.5</b>	Very Much Efficient
8. Hardware utilization						<b>4.4</b>	Very Much Efficient
9. Support on interfacing with other devices						<b>4.5</b>	Very Much Efficient
10. Compatibility with interfaced devices						<b>4.6</b>	Very Much Efficient
<b>Average</b>						<b>4.44</b>	<b>Very Much Efficient</b>

**Portability.** A software that is capable of being transferred from one employer to another.

PORABILITY	5	4	3	2	1	Mean	Description
1. Modification of the system						<b>4.5</b>	Very Much Portable
2. Adaptability to other environment						<b>4.4</b>	Very Much Portable
3. Adaptability to other applications						<b>4.4</b>	Very Much Portable
4. Flexibility to other settings						<b>4.4</b>	Very Much Portable
5. Support in any form of network communication						<b>4.6</b>	Very Much Portable
6. Adaptability to new version of system requirements						<b>4.3</b>	Very Much Portable
7. System supports on maximum hardware requirements						<b>4.4</b>	Very Much Portable
8. User capability/capacity						<b>4.5</b>	Very Much Portable
9. Its intended application						<b>4.5</b>	Very Much Portable
10. Its intended design						<b>4.6</b>	Very Much Portable
<b>Average</b>						<b>4.46</b>	<b>Very Much Portable</b>

**Maintainability.** Intelligent IP Address Locator via Google Maps is measures the ease and speed with which a system can be restored to operational status after a failure occurs.

MAINTAINABILITY	5	4	3	2	1	Mean	Description
1. Modification of the system software						<b>4.6</b>	Very Much Maintainable
2. Change of software capabilities						<b>4.5</b>	Very Much Maintainable
3. Increase program capabilities						<b>4.3</b>	Very Much Maintainable
4. Improving performance						<b>4.3</b>	Very Much Maintainable
5. Correction of program defects						<b>4.4</b>	Very Much Maintainable
6. Accessibility for maintenance						<b>4.4</b>	Very Much Maintainable
7. Configuration of system operation						<b>4.4</b>	Very Much Maintainable
8. Flexibility for system modification						<b>4.5</b>	Very Much Maintainable
9. Compliance of concurrent system requirements						<b>4.3</b>	Very Much Maintainable
10. Advance feature for recent technology						<b>4.6</b>	Very Much Maintainable
<b>Average</b>						<b>4.43</b>	<b>Very Much Maintainable</b>

**APPENDIX F****Minutes of Corrections**



### **MINUTES OF CORRECTIONS**

On May 17, 2017 we start our final defend at 8:00 am in the morning. We present our research study to the panelist entitled “Intelligent IP Address Locator via Google Map”, and after our presentation the Chairman of the panelist Engr. Joseph Aurelius P. Jacinto and his co-panelist Sir Armando T. Saguin JR., Sir John D. Sagapsapan and Engr. Angelito M. Punzal, Prof. Ed Neil O. Maratas have some brief questions and corrections that follows:

**PROF. ARMANDO T. SAGUIN JR.**

- ❖ Chapter 4 (page 54) – Comparison

**ORAL:**

- Add Comments
- Like

**PROF. JOHN D. SAGAPSAPAN**

- ❖ (page vii-x) – Table of Contents (indentation and spacing)
- ❖ (page x-xi) – Appendecies (indentation and spacing)
- ❖ (page 2-12) – chapter 1 (margin, indentation, spacing)
- ❖ (page 15-28) – chapter 2 (spacing, margin,indention and boxin all figures)
- ❖ (page 29-39) – chapter 3 (spacing, margin,indention and boxin all figures)

**ORAL:**

- Add Comments
- Like

**ENGR. JOSEPH AURELIUS P. JACINTO**

- ❖ (page i) – Title Page (single space)
- ❖ (page v) – Dedication (margin)
- ❖ (page vi) – Acknowledgement (margin)
- ❖ (page vii) – Abstract (background goal)
- ❖ (page 1-12) – Chapter 1 (change font size, indentation and margin)
- ❖ (page 13) – Chapter 2 (change font size)

ORAL:

- Add Comments
- Like

PROF. ED NEIL O MARATAS

ORAL:

- Add Comments
- Like

ENGR. ANGELITO M. PUNZAL

- ❖ (page i) – Title Page (single space)
- ❖ (pagevii) – Table of Contents (change of order)
- ❖ (page x) – Appendecies (specing)
- ❖ (page xi) – List of Tables (list all tables)

ORAL:

- Add Comments
- Like

## **APPENDIX G**

### **Curriculum Vitae**

