**BORDER ALERT SYSTEM FOR FISHERMEN**

**A PROJECT REPORT**

***Submitted by***

|  |  |
| --- | --- |
| **MURALI KRISHNAN A** | **(Reg. No. 201304068)** |
| **SIVA PRASANNA T** | **(Reg. No. 201304265)** |
| **ARAVINDHA SHIVA S** | **(Reg. No. 201304013)** |

***in partial fulfillment for the award of the degree***

***of***

**BACHELOR OF ENGINEERING**

**in**

COMPUTER SCIENCE AND ENGINEERING

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**MEPCO SCHLENK ENGINEERING COLLEGE, SIVAKASI**

**(An Autonomous Institution affiliated to Anna University Chennai)**

**April 2017**

**BONAFIDE CERTIFICATE**

Certified that this project report titled **PRDER ALERT SYSTEM FOR FISHERMEN** is the bonafide work of **Mr. MURALI KRISHNAN A, Mr. SIVA PRASANNA T, Mr. ARAVINDHA SHVA S** who carried out the research under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form part of any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

Mr.C.Balasubramanian, M.E., Dr. K.Muneeswaran, M.E., Ph.D.,

**Internal guide Head of the Department**

**Assistant professor (Sr. Grade), Senior Professor,**

**Department of Computer Science Department of Computer Science and Engineering, and Engineering,**

**Mepco Schlenk Engineering College, Mepco Schlenk Engineering College,**

**Sivakasi. Sivakasi.**

Submitted for viva-voce Examination held at **MEPCO SCHLENK ENGINEERING COLLEGE, SIVAKASI (AUTONOMOUS)** on **……………..............**

**Internal Examiner External Examiner**

**ACKNOWLEDGEMENT**

First and foremost, we thank the Almighty for his abundant blessings that is showered upon our past, present, future successful endeavors.

At the appellation, we would like to extend our sincere gratitude to our reverent principal, **Dr. S. Arivazhagan, M.E., Ph.D.,** for providing sufficient working environment such as system and library facilities. We also thank him very much for providing us with adequate lab facilities, which enable us to complete our project.

We are grateful to our charismatic Head of the Computer Science Engineering and Department **Dr. K. Muneeswaran, M.E., Ph.D.,** for giving us this golden opportunity to undertake a project of this nature and for his most valuable guidance given at every phase of our work.

We would also like to extend our gratitude **Mrs. S. Devi Mahalakshmi, M.E., (Ph.D)** Assistant professor (Sr. Grade), Department of Computer Science and Engineering, Mepco Schlenk Engineering College for being our project coordinator and directing us through our project.

We thank **Mr.C.Balasubramanian, M.E(Ph.D).,** Assistant professor (Sr. Grade), Department of Computer Science and Engineering, Mepco Schlenk Engineering College for his valuable guidance and encouragement.

We acknowledge with the sense of gratitude, the help given to us by all the staffs and technicians in the department.

We are very grateful to our beloved parents and friends who afforded the necessary help for us at the right time for making our project a grand success.

**ABSTRACT**

The proposed “Border alert system for fishermen” is used to save the lives of the fishermen who venture out into the sea for fishing. This system prevents them from crossing the maritime boundary giving them alarms and controlling them. They are forced to come back beyond a certain point. There is a need to track all fishermen centrally and that is done on base station. In case of emergencies alerts can be sent to them. Also any information can be transmitted.

|  |  |  |  |
| --- | --- | --- | --- |
| **TABLE OF CONTENTS** | | | |
| **CONTENTS** | |  | **PAGE NO** |
| **List of tables** | |  | vii |
| **List of figures** | |  | viii |
| **List of abbreviations** | |  | ix |
| **CHAPTER 1** | | **INTRODUCTION** |  |
|  | |  |  |
|  | 1.1 | GENERAL | 1 |
|  | 1.2 | OBJECTIVE OF THE PROJECT | 4 |
|  | 1.3 | PROBLEM STATEMENT | 4 |
|  | 1.4 | PROPOSED SYSTEM | 5 |
|  | 1.5 | APPLICABLE AREAS | 5 |
| **CHAPTER 2** |  | **LITERATURE SURVEY** |  |
|  | 2.1 | AFFINITY ANALYSIS AND ASSOCIATION RULE MINING USING APRIORI ALGORITHM IN MARKET BASKET ANALYSIS | 6 |
|  | 2.2 | DATA MINING BASED STORE LAYOUT ARCHITECTURE FOR SUPERMARKET | 17 |
|  | 2.3 | INFORMATION SYSTEM ON MARKET BASKET ANALYSIS | 20 |
|  | 2.4 | ALIGNMENT AND INTEGRATION OF COMPLEX NETWORKS BY HYPERGRAPH- BASED SPECTRAL CLUSTERING | 22 |
| **CHAPTER 3** | | **SYSTEM DESIGN** |  |
|  | 3.1 | DATA FLOW DESIGN | 37 |
|  |  | 3.3.1 Transaction REPOSITORY |  |
|  |  | 3.3.2 fILTER SINGLE ITEM ORDER |  |
|  |  | 3.3.3 FREQUENT ITEM SET MINING |  |
|  |  | 3.3.4 HYPER GRAPH CONSTRUCTION |  |
|  |  | 3.3.5 HYPER GRAPH CLUSTERING |  |
| **CHAPTER 4** | | **SYSTEM IMPLEMENTATION** |  |
|  |  |  |  |
|  | 4.1 | FILTER SINGLE ITEM ORDER IMPLEMENTATION |  |
|  | 4.2 | FREQUENT ITEM SET MINING IMPLEMENTATION |  |
|  | 4.3 | HYPER GRAPH CONSTRUCTION |  |
|  | 4.4 | HYPER GRAPH CLUSTERING IMPLEMENTATION |  |
| **CHAPTER 5** | | **TEST CASES AND TEST RESULTS** |  |
|  |  |  |  |
|  | 5.1 | TEST CASES AND RESULTS |  |
|  |  | 5.1.1 INPUT DATA |  |
|  |  | 5.1.2 FILTER SINGLE ITEM ORDER TESTING AND RESULTS |  |
|  |  | 5.1.3 FREQUENT ITEM SET MINING TESTING AND RESULTs |  |
|  |  | 5.1.4 HYPER GRAPH CONSTRUCTION TESTING AND RESULTS |  |
|  |  | 5.1.5 HYPER GRAPH CLUSTERING TESTING AND RESULTS |  |
|  |  | 5.1.6 NAMING THE PRODUCT ID |  |
|  |  | 5.1.7 representation of clusters |  |
|  |  | 5.1.8 Positive predictive value (ppv) |  |
| **CHAPTER 6** | | **conclusion and future enhancement** |  |
|  |  |  |  |
| **REFERENCES** |  |  | 55 |

**LIST OF TABLES**

**S. No. TABLE No. TABLE NAME PAGE No.**

1. **5.1 Test Case for Filter Single Item Order 12**
2. **5.2 Test Case for Frequent Item Set Mining 22**
3. **5.3 Test Case for Hyper Graph Construction 24**
4. **5.4 Test Case for Hyper Graph Clustering 25**
5. **5.5 Test Case for Name the Product ID 26**
6. **5.6 Representation of Clusters 29**

**LIST OF FIGURES**

**S. No. FIGURE No. FIGURE NAME PAGE No.**

**1. 3.1 Data Flow Design of Market Basket Analysis 6**

**2. 3.2 Construction of FP Tree from the transaction 13**

**3. 3.3 Representation of Hyper Graph 14**

**4. 5.1 Transaction of data 15**

**5. 5.2 Product ID and its corresponding name 17**

**6. 5.3 Frequent pattern tree 35**

**7. 5.4 FP Growth tree 36**

**8. 5.5 Hyper Graph edges 37**

**9. 5.6 Hyper Graph clustering 38**

**10. 5.7 Naming with product id 39**

**11. 5.8 Representation of clusters 40**

**12. 5.9 PPV vs number of items 41**

**13. 5.10 PPV vs number of transactions 42**

**ABBREVIATION**

**GPS : Global positioning system**

**BS: Base station**

**FM : Fishermen**

**IMBL : International Maritime Boundary Line**



**CHAPTER 1**

**INTRODUCTION**

* 1. **GENERAL**

In day-to- day life we hear about many Tamil fishermen being caught and put under Sri-Lankan custody and even killed. They, even today invoke the historical rights and routinely stay into the International Maritime Boundary Line (IMBL) for fishing. From Tamil Nadu about 18,000 boats of different kinds conduct fishing along the India-Sri Lanka maritime border. But by accidentally crossing the border without knowledge, they get shot by the Lankan navy. This leads to loss in the both humans as well as their economic incomes.

The sea border between the countries is not easily identifiable, which is the main reason for this cross border cruelty. Here we have designed a system using embedded system which protects the fishermen by notifying the country border to them by using Global Positioning System (GPS). We use GPS receiver to find the current location of the fishing boat or vessel. Using GPS, we can find the current latitude and longitude values and is sent to the microcontroller unit. Then the controller unit finds the current location by comparing the present latitude and longitudinal values with the predefined value. Then from the result of the comparison, this system aware the fishermen that they are about to reach the nautical border. The area is divided into 3 zones- safe zone, restricted zone, and finally the danger zone. If the boat is in safe zone, then there is no interruption or warning. Thus they can make it clear that the boat is in normal area. In case it moves further and reaches the restricted zone, an alarm is given as a warning and the speed of engine reduces as he progresses along the restricted zone. If the fisherman ignores the warning, and if the boat enters the danger zone, the engine stops automatically. Now the fishermen are allotted a buffer distance so that he can turn back and move towards safe zone again.

* 1. **OBJECTIVE OF THE PROJECT**
* To save prevent them from crossing the IMBL saving their valuable lives.
* To track all the boats from the border so that they can be sent alerts in case of any emergencies.

* 1. **PROBLEM STATEMENT**

The sea border or the IMBL between countries are not easily identifiable which is the main reason for fishermen crossing the border and getting caught. This creates a need for a system to alert them before they cross the border. Also they are unaware of any predictions regarding any danger in sea.

**1.4 PROPOSED SYSTEM**

The proposed system has a GPS which tracks the current location of the fishermen. This is sent via Bluetooth to the phone where the current position is displayed on electronic map. The waters are divided into 3 zones namely safe, restricted and danger zones. The speed of the engine gradually reduces as the boat progresses in the restricted zone and stops upon entry into the danger zone. And the fishermen are given 500meter buffer to turn his boat around and come back.

* 1. **APPLICABLE AREAS**

Areas where men go for fishing:

* Back waters
* Sea
* Ocean

**1.6 ORGANIZATION OF REPORT:**

The overviews of the subjects described in the underlying chapters are given below

* **Chapter 2:** Covers discussion about the related works carried out before formulating the proposed system. It also briefs about the methods adopted by existing system and its disadvantages.
* **Chapter 3:** Deals with the overview design of the proposed system and the formulations used in the implementation of the system.
* **Chapter 4:** Outlines the design and explains the methodology of the proposed system.
* **Chapter 5:** Deals with the implementation of the proposed system, how the modules interact and how the method executes to come with the required result.
* **Chapter 6:** Explains about the experiments carried out in proving the efficiency of the system.
* **Chapter 7:** Concludes about the system and discusses about any future enhancements.



**CHAPTER 2**

**LITERATURE SURVEY**

**2.1 DESIGN OF BORDER ALERT SYSTEM FOR FISHERMEN USING GPS**

Here we have designed a system using embedded system which protects the fishermen by notifying the country border to them by using Global Positioning System (GPS) and Global system for mobile communication (GSM). We use GPS receiver to find the current location of the fishing boat or vessel. Using GPS, we can find the current latitude and longitude values and is sent to the microcontroller unit. Then the controller unit finds the current location by comparing the present latitude and longitudinal values with the predefined value. Then from the result of the comparison, this system aware the fishermen that they are about to reach the nautical border. The area is divided into four zones- normal zone, warning zone, zone near to restricted zone and finally the restricted zone. If the boat is in normal area, then the LCD displays normal zone. Thus they can make it clear that the boat is in normal area. In case it moves further and reaches the warning zone, the LCD displays warning zone. If the fisherman ignores the warning or fail to see the display and move further, and if the boat enters the zone nearer to the restricted zone the alarm will turn on and the speed of the boat engine automatically gets controlled by 50%. If the fisherman did not take any reaction about the alarm and move further, then the boat will enter into the restricted zone, the alarm continues to beep as before, and once it touches the restricted zone, the boat engine gets off by the control of fuel supply to engine.

The GPS Modem will continuously give the signal which determines the latitude and longitude and indicates the position of the fishermen to them. Then it gives the output which gets read and displayed in the LCD. The same data is sent to the mobile of the fisherman and simultaneously the same data is sent to the Sea border security. An EEPROM is used to store the data, received by GPS receiver. The hardware which interfaces with microcontroller are LCD display, GSM modem and GPS Receiver. GPS (Global Positioning System) is increasingly being used for a wide range of applications. It provides reliable positioning, navigation, and timing services to world-wide users on a continuous basis in all weather, day and night, anywhere on or near the Earth. 28 satellites inclined at 55° to the equator orbit the Earth every 11 hours and 58minutes at a height of 20,180 km on 6 different orbital Lanes and each one of these satellites has up to four atomic clocks on board. All we require is an accurate clock. By comparing the arrival time of the satellite signal with the onboard clock time, at which the signal was emitted, the latitude and longitudinal degree of the boat’s location is determined. The current design is an embedded application, which will continuously monitor a moving Boat and once the boat goes beyond the level of the defined layer the particular operation will be done. For doing so an AT89c51microcontroller is interfaced serially to a GSM MODEM AND GPS receiver. The current position received from GPS is stored as S1(latitude), S2 (longitude). The latitude S1 is compared with stored latitudes.

If latitude match, then adjacent latitude and longitudes (X1, Y1 and X2, Y2) are retrieved from stored table and substituted in the equation given below:

(Y-Y1)/(Y2-Y1) = (X-X1)/(X2-X1)

By simplification, we get ax + by = c

**LIMITATIONS**

The type of GPS used plays an important role because different GPS system has different reliability. Equation for the comparison of latitude and longitude is not accurate

**2.2 GPS-Based Vessel Position Monitoring and Display System**

A description is given of the first fully operational, GPS (Global Positioning System) Loran-based vessel monitoring system. The system's monitoring workstation, communications solutions, and variety of onboard navigation systems provide an integrated capability for the marine fleet operator and are the basis of consideration for large-scale systems which might be implemented for hazardous cargo or oil and gas product ocean transport. Issues regarding the implementation of GPS-based vessel tracking are discussed, and the design of the system is described. The vessel navigation and position reporting suite, communication of data, and vessel monitoring system applications are examined. The control station has a completely redundant installation, and a great deal of error checking is built into the upload process to preclude an erroneous upload being broadcast. In fact, the latest version of the control station will employ a complete end- around check to an actual GPS receiver to verify the upload in addition to the internal checks built in. Receivers are built to a variety of specifications. It is possible to build a receiver that tracks up to 5 satellites and automatically detect any “failed” satellite signal, for whatever reason, and a 6th satellite allows failure identification for fail- safe operation. Of course, completely dual installations can be considered as well.

So does this qualify GPS-based tracking for safety-related applications? Not necessarily, however, there is an interesting precedent in this area. In the early 1980’s, the Air Force decided to develop a GPS-based range tracking system to replace tracking radar at many of its test ranges. One of the functions of the new GPS-based system was to maintain range safety concerning aircraft separation. This official use certainly seems to justify serious consideration of GPS for such critical applications.

There is quite a bit of concern, especially in Europe, that GPS operation is available only at the whim of the U.S. Department of Defense. Indeed, occasional statements by DoD personnel seem to reinforce this phobia. However, it is a stated policy of the DoD to make the Standard Positioning Service (C/A Code) continuously available to the civil community. Furthermore, an excellent antidote for most ills conceivable with GPS is the use of differential GPS, which corrects for the errors observed at a local reference station

**LIMITATIONS**

A brute-force algorithm is not necessarily a one that requires a lot of memory but it generally requires a lot of processing time.  For many problems, brute force is only feasible for small problem dimensions. Similarly, Apriori algorithm have their own disadvantages.

**2.3 INFORMATION SYSTEM ON MARKET BASKET ANALYSIS**

In certain cases, due to such high precision of geographical position estimation, there problems connected with the accuracy of systems for the projection of the globe convex surface to the two-dimensional surface of a chart emerge. When using DGPS, one should pay closer attention if the chart datum is compliant with GPS. Although DGPS can increase the precision of the SPS several times, the accuracy improvement of the PPS is very limited. Satellite navigation systems used for non-military purposes found their application mainly in maritime and aerial navigation; however, recently they started to invade inland applications. Expensive cars are being equipped with such systems, which transmit signals from remotely controlled transmitters hidden in vehicles in case of theft. Today, one can buy an inexpensive GPS or DGPS receiver capable of displaying position and travelled distance on a map of any selected region, in selected scale and with a variety of useful information such as location of hotels, restaurants, museums, or gas stations. Many developed countries are beginning to implement GPS receivers equipped with an interactive vehicle communication system that connects them with traffic control centres.

**LIMITATION**

Candidate generation is extremely slow in apriori algorithm. Runtime of apriori algorithm increases exponentially depending on the number of different items.

**2.4 ALIGNMENT AND INTEGRATION OF COMPLEX NETWORKS BY HYPERGRAPH- BASED SPECTRAL CLUSTERING**

Tom Michoel and Bruno Nachtergaele proposed that the complex networks possess a rich, multi-scale structure reflecting the dynamical and functional organization of the systems they model. Often there is a need to analyze multiple networks simultaneously, to model a system by more than one type of interaction or to go beyond simple pairwise interactions, but currently there is a lack of theoretical and computational methods to address these problems. Here we introduce a framework for clustering and community detection in such systems using hypergraph representations. Our main result is a generalization of the Perron-Frobenius theorem from which we derive spectral clustering algorithms for directed and undirected hypergraphs. We illustrate our approach with applications for local and global alignment of protein-protein interaction networks between multiple species, for tripartite community detection in folksonomies, and for detecting clusters of overlapping regulatory pathways in directed networks.

**Graphs and Hyper Graph**

A graph G is defined as a pair (V, E) of vertices V and edges (pairs of vertices) E, which may be directed or not. In a weighted graph, a number is assigned to each edge which may represent, e.g., the cost, length or reliability of an edge. A hyper graph is a generalization of a graph where an edge, called hyper edge in this case, can connect any number of vertices, i.e., E is a set of arbitrarily sized subsets of V. A particular class of hyper graphs are so called k-uniform hyper graphs where each hyper edge has the same cardinality k. Algebraically, a graph can be represented by an adjacency matrix A of dimension N × N, with N the number of vertices, such that Aij = 1 if {i, j} ∈ E and 0 otherwise. For undirected graphs, A is a symmetric matrix and for weighted graphs, Aij is defined to be the weight of the edge {i, j}. For k uniform hyper graphs, the notion of adjacency matrix can be generalized to an adjacency multi-array or tensor T, with Ti1...ik = 1 if {i1, . . . , ik} ∈ E and 0 otherwise. For a general hyper graph, we define a function w on the set of subsets of V such that w(E) = 1 for E ∈ E and 0 otherwise. In general, we allow weighted hyper graphs where w can be any non-negative function.

**Spectral Clustering and Bi-Clustering in Hyper Graph**

Having a generalization of the Perron-Frobenius theorem, it is straightforward to also generalize the spectral clustering method. Define for X ⊂ V,

with x the dominant eigenvector and uX ∈ SNp now de-fined by uX,i = |X|−1/p for i ∈ X and 0 otherwise. The parameter p balances cluster size versus edge density. For p = 1, Sp is the ratio of edges to nodes in X. Taking p > 1 diminishes the influence of the denominator and progressively favors to have a high number of edges rather than a high number of edges per node in high-scoring clusters (further details in Section VII). The spectral clustering algorithm becomes:

1. Calculate the maximizer x of Rp.
2. Find the cluster Xmax which solves the restricted variational problem Xmax = argmax c>0 Sp(Xc) with Xc = {i ∈ V : xi > c}.
3. Store Xmax, remove all hyperedges between nodes in Xmax from the edge set E, and repeat the procedure until no more hyperedges remain.

**LIMITATION**

Spectral clustering cares about connectivity instead of geometrical proximity. So if the data isn't well geometrically separated, but clusters aren't connected, spectral clustering will work well. In other cases the k-means algorithm works well.

**SYSTEM STUDY**

**CHAPTER 3**

**SYSTEM STUDY**

**3 SYSTEM STUDY**

The core idea of this system is to track and control the boat of the fishermen. Initially the system comprising of Arduino and GPS is installed in the boat. Whenever the fishermen leave the shore, base station begins tracking the boat. The location of the boat is determined using GPS. Alerts are sent to fishermen from the base station to fishermen.

First of all, a detailed study has to be made on Arduino microcontrollers, GSM, GPS, DC Motor, 5V Regulator (KA7805).

**3.1 ARDUINO:**

Arduino is an open-source computer hardware and software company, project and user community that designs and manufactures microcontroller-based kits for building digital devices and interactive objects that can sense and control objects in the physical world.

The work is based on microcontroller board designs, manufactured by several vendors, using various microcontrollers. These systems provide sets of digital and analog I/O pins that can be interfaced to various expansion boards ("shields") and other circuits. The boards feature serial communications interfaces, including USB on some models, for loading programs from personal computers. For programming the microcontrollers, the Arduino project provides an integrated development environment (IDE) based on the Processing project, which includes support for the C and C++ programming languages.

The first Arduino was introduced in 2005, aiming to provide an inexpensive and easy way for novices and professionals to create devices that interact with their environment using sensors and actuators. Common examples of such devices intended for beginner hobbyists include simple robots, thermostats, and motion detectors.

An Arduino board historically consists of an Atmel 8-, 16- or 32-bit AVR microcontroller (although since 2015 other makers' microcontrollers have been used) with complementary components that facilitate programming and incorporation into other circuits. An important aspect of the Arduino is its standard connectors, which lets users connect the CPU board to a variety of interchangeable add-on modules known as shields.

Some shields communicate with the Arduino board directly over various pins, but many shields are individually addressable via an I²C serial bus—so many shields can be stacked and used in parallel. Prior to 2015 Official Arduinos had used the Atmel mega AVR series of chips, specifically the ATmega8, ATmega168, ATmega328, ATmega1280, and ATmega2560 and in 2015 units by other manufacturers were added. A handful of other processors have also been used by Arduino compatible devices.

Most boards include a 5 V linear regulator and a 16 MHz crystal oscillator (or ceramic resonator in some variants), although some designs such as the LilyPad run at 8 MHz and dispense with the onboard voltage regulator due to specific form-factor restrictions.

An Arduino's microcontroller is also pre-programmed with a boot loader that simplifies uploading of programs to the on-chip flash memory, compared with other devices that typically need an external programmer. This makes using an Arduino more straightforward by allowing the use of an ordinary computer as the programmer.

At a conceptual level, when using the Arduino integrated development environment, all boards are programmed over a serial connection. Its implementation varies with the hardware version. Some serial Arduino boards contain a level shifter circuit to convert between RS-232 logic levels and TTL-level signals.

Current Arduino boards are programmed via Universal Serial Bus (USB), implemented using USB-to-serial adapter chips such as the FTDI FT232. Some boards, such as later-model Uno boards, substitute the FTDI chip with a separate AVR chip containing USB-to-serial firmware, which is reprogrammable via its own ICSP header.

Other variants, such as the Arduino Mini and the unofficial Boarduino, use a detachable USB-to-serial adapter board or cable, Bluetooth or other methods, when used with traditional microcontroller tools instead of the Arduino IDE, standard AVR ISP programming is used.

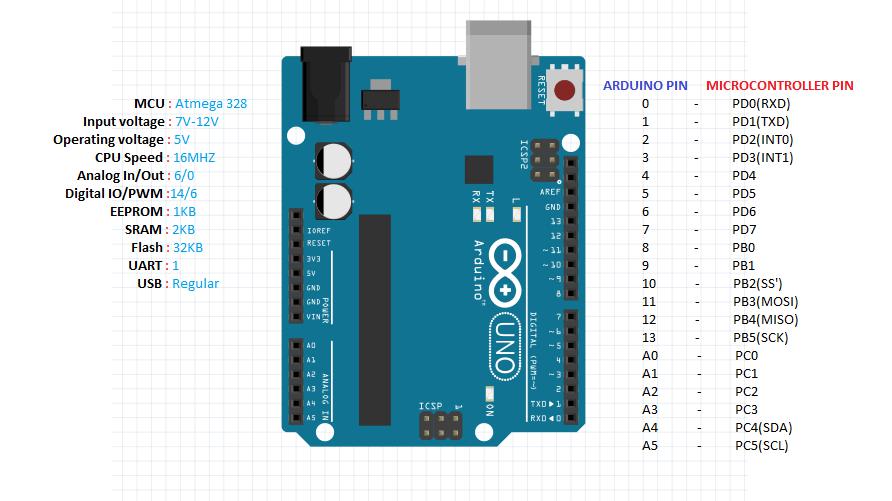
The Arduino board exposes most of the microcontroller's I/O pins for use by other circuits. The Diecimila, Duemilanove, and current Uno provide 14 digital I/O pins, six of which can produce pulse-width modulated signals, and six analog inputs, which can also be used as six digital I/O pins.

These pins are on the top of the board, via female 0.10-inch (2.5 mm) headers. Several plug-in application shields are also commercially available. The Arduino Nano, and Arduino-compatible Bare Bones Board and Boarduino boards may provide male header pins on the underside of the board that can plug into solder less breadboards.

There are many Arduino-compatible and Arduino-derived boards. Some are functionally equivalent to an Arduino and can be used interchangeably. Many enhance the basic Arduino by adding output drivers, often for use in school-level education to simplify the construction of buggies and small robots.

Others are electrically equivalent but change the form factor, sometimes retaining compatibility with shields, sometimes not. Some variants use completely different processors, with varying levels of compatibility.

**3.1.1 ARDUINO PIN DIAGRAMS:**

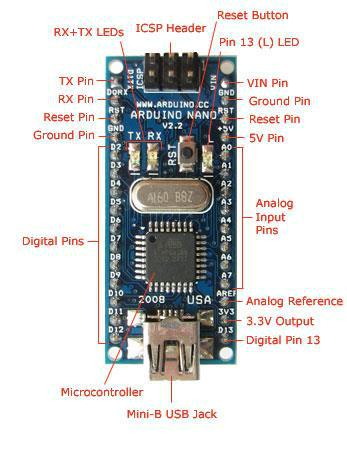
** ARDUINO UNO R3**

**Figure 3.1: Arduino UNO R3 pin diagram**

**Table 3.1 Pin configuration of Arduino Uno**

|  |  |
| --- | --- |
| Pin 0 to Pin 13 | DIGITAL MODE PINS |
| Pin A0 to Pin A5 | ANALOG MODE PINS |
| Pin 0 | Tx Pin |
| Pin 1 | Rx Pin |

**ARDUINO NANO**

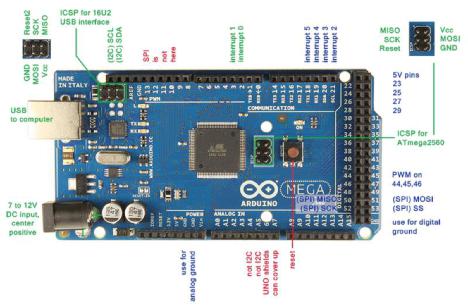
****

**Figure 3.2 Arduino Nano Pin Diagram**

**Table 3.2 Pin configuration of Arduino Nano**

|  |  |
| --- | --- |
| Pin D2 to pin D12 | DIGITAL MODE PINS |
| Pin A0 to Pin A7 | ANALOG MODE PINS |
| GND , RST Pin | GROUND AND RESET PIN |
| AREF | ANALOG REFERENCE |

**ARDUINO MEGA 2560**

****

**Figure 3.3 Arduino Mega 2560 Pin diagram**

**Table 3.3 Pin configuration of Arduino Mega 2560**

|  |  |
| --- | --- |
| Pin 0 to Pin 53 | DIGITAL MODE PINS |
| Pin A0 to Pin A15 | ANALOG MODE PINS |
| Pin 18 to Pin 21 | INTERRUPT PINS |
| Pins 23,25,27,29 | 5V PINS |

**3.2 BLUETOOTH**



**Figure 3.4 BLUETOOTH**

Bluetooth is a [wireless](https://en.wikipedia.org/wiki/Wireless) technology standard for exchanging data over short distances (using short-wavelength [UHF](https://en.wikipedia.org/wiki/UHF) [radio waves](https://en.wikipedia.org/wiki/Radio_waves) in the [ISM band](https://en.wikipedia.org/wiki/ISM_band) from 2.4 to 2.485 GHz) from fixed and mobile devices, and building [personal area networks](https://en.wikipedia.org/wiki/Personal_area_network) (PANs). Invented by telecom vendor [Ericsson](https://en.wikipedia.org/wiki/Ericsson) in 1994, it was originally conceived as a wireless alternative to [RS-232](https://en.wikipedia.org/wiki/RS-232) data cables. It can connect up to seven devices, overcoming problems that older technologies had when attempting to connect to each other.

Bluetooth is managed by the [Bluetooth Special Interest Group](https://en.wikipedia.org/wiki/Bluetooth_Special_Interest_Group) (SIG), which has more than 30,000 member companies in the areas of telecommunication, computing, networking, and consumer electronics. The [IEEE](https://en.wikipedia.org/wiki/Institute_of_Electrical_and_Electronics_Engineers) standardized Bluetooth as IEEE 802.15.1, but no longer maintains the standard. The Bluetooth SIG oversees development of the specification, manages the qualification program, and protects the trademarks. A manufacturer must meet [Bluetooth SIG standards](https://en.wikipedia.org/wiki/Bluetooth_Special_Interest_Group#Qualification) to market it as a Bluetooth device. A network of [patents](https://en.wikipedia.org/wiki/Patent) apply to the technology, which are licensed to individual qualifying devices.

Bluetooth operates at frequencies between 2402 and 2480 MHz, or 2400 and 2483.5 MHz including [guard bands](https://en.wikipedia.org/wiki/Guard_band) 2 MHz wide at the bottom end and 3.5 MHz wide at the top. This is in the globally unlicensed (but not unregulated) Industrial, Scientific and Medical ([ISM](https://en.wikipedia.org/wiki/ISM_band)) 2.4 GHz short-range radio frequency band. Bluetooth uses a radio technology called [frequency-hopping spread spectrum](https://en.wikipedia.org/wiki/Frequency-hopping_spread_spectrum). Bluetooth divides transmitted data into packets, and transmits each packet on one of 79 designated Bluetooth channels. Each channel has a bandwidth of 1 MHz. It usually performs 800 hops per second, with [Adaptive Frequency-Hopping](https://en.wikipedia.org/wiki/Adaptive_frequency-hopping_spread_spectrum) (AFH) enabled. [Bluetooth low energy](https://en.wikipedia.org/wiki/Bluetooth_low_energy) uses 2 MHz spacing, which accommodates 40 channels.

Originally, [Gaussian frequency-shift keying](https://en.wikipedia.org/wiki/Gaussian_frequency-shift_keying) (GFSK) modulation was the only modulation scheme available. Since the introduction of Bluetooth 2.0+EDR, π/4-[DQPSK](https://en.wikipedia.org/wiki/DQPSK) (Differential Quadrature Phase Shift Keying) and 8DPSK modulation may also be used between compatible devices. Devices functioning with GFSK are said to be operating in basic rate (BR) mode where an instantaneous [data rate](https://en.wikipedia.org/wiki/Bit_rate) of 1 [Mbit/s](https://en.wikipedia.org/wiki/Data_rate_units) is possible. The term Enhanced Data Rate (EDR) is used to describe π/4-DPSK and 8DPSK schemes, each giving 2 and 3 Mbit/s respectively. The combination of these (BR and EDR) modes in Bluetooth radio technology is classified as a "BR/EDR radio".

**3.3 GPS (GLOBAL POSITIONING SYSTEM):**

****

**Figure 3.5 GPS Module**

The **Global Positioning System** (**GPS**) shown in Fig 3.5 is a space-based radio navigation system owned by the United States government and operated by the United States Air Force. It is a [global navigation satellite system](https://en.wikipedia.org/wiki/Satellite_navigation) that provides [geolocation](https://en.wikipedia.org/wiki/Geolocation) and time information to a [GPS receiver](https://en.wikipedia.org/wiki/GPS_receiver) anywhere on or near the Earth where there is an unobstructed line of sight to four or more GPS satellites.

A GPS device can retrieve from the GPS system **location and time information** in all weather conditions, anywhere on or near the Earth. A GPS reception requires an unobstructed line of sight to four or more GPS satellites, and is subject to poor satellite signal conditions. In exceptionally poor signal conditions, for example in urban areas, satellite signals may exhibit multipath propagation where signals skip off structures, or are weakened by meteorological conditions. Obstructed lines of sight may arise from a tree canopy or inside a structure, such as in a building, garage or tunnel.

The GPS module continuously produces a set of data regarding the position of the earth surface where it is situated which includes the current position with respect to the equator of the earth in terms of Latitude and Longitude. This data can be decoded and printed into the readable format with the help of a microcontroller only. In this project the data regarding the geographical coordinate is extracted from the **GPS**output with the help of the **Arduino**. The Arduino can be used as a stand-alone board of which the output or inputs can be taken from the boards or given to the board. They can communicate using standard communication ports like USART, TWI, SPI etc. which enables them to be connected with various kinds of devices. The Arduino board is designed for easy prototyping and the IDE used for coding is very simple and provides so many libraries for interfacing with common external devices.

Using the function **Serial.read()** the Arduino continuously reads the data from the GPS module, looking for Latitude-Longitude details. The GPS send the data in standard NMEA format which consist of the real time data regarding the current position. The format includes so many sentences and among them one particular sentence referred to as “Global Positioning System Fix Data” is extracted to read the Latitude Longitude.

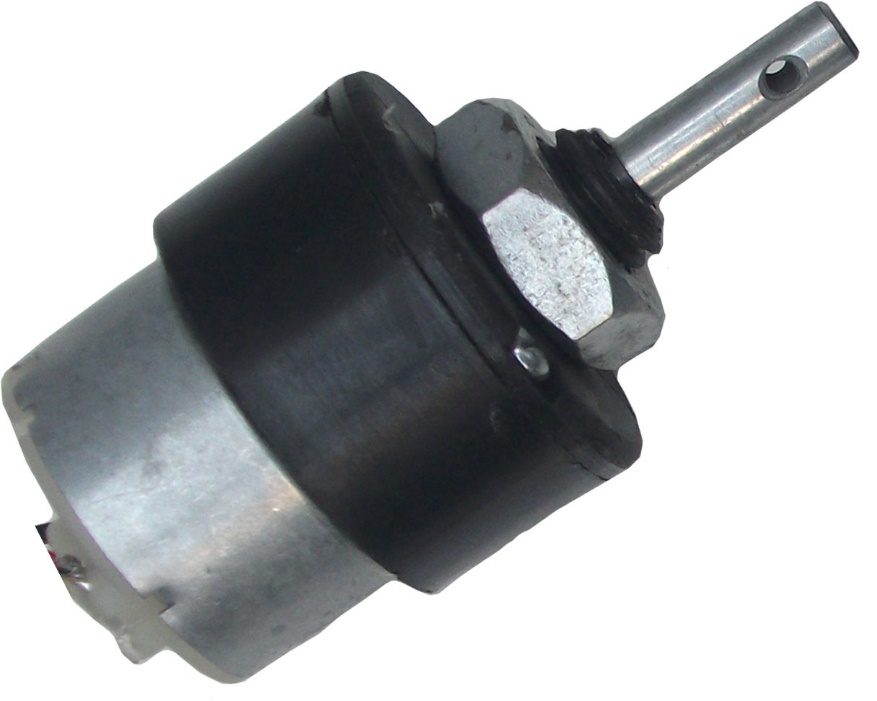
**Example:**

$GPRMC,092204.999,A,4250.5589,S,14718.5084,E,0.00,89.68,211200,,\*25**`**

**Table 3.4 Sample GPS Data and Sentence Format**

|  |  |  |
| --- | --- | --- |
| **Field** | **Example** | **Comments** |
| Sentence ID | $GPRMC |  |
| UTC Time | 092204.999 | hhmmss.sss |
| Status | A | A = Valid, V = Invalid |
| Latitude | 4250.5589 | ddmm.mmmm |
| N/S Indicator | S | N = North, S = South |
| Longitude | 14718.5084 | dddmm.mmmm |
| E/W Indicator | E | E = East, W = West |
| Speed over ground | 0.00 | Knots |
| Course over ground | 0.00 | Degrees |
| UTC Date | 211200 | DDMMYY |
| Magnetic variation |  | Degrees |
| Magnetic variation |  | E = East, W = West |
| Checksum | \*25 |  |

**3.4 12V DC MOTOR:**

****

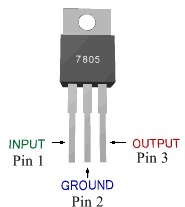
**Fig 3.6 12V DC Motor**

A **DC motor** (Fig 3.6) is any of a class of rotary electrical machines that converts direct current electrical energy into mechanical energy. The most common types rely on the forces produced by magnetic fields. Nearly all types of DC motors have some internal mechanism, either electromechanical or electronic, to periodically change the direction of current flow in part of the motor.

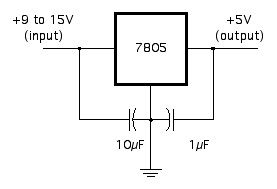
Here, the **DC motor** is used to emulate the motorcycle engine. The **Motorcycle engine** is similar to the DC motor. A motorcycle engine is an engine that powers a motorcycle. Motorcycle engines are typically two-stroke or four-stroke internal combustion engines, but other engine types (such as Wankels and electric motors) have been used in small numbers.

It is a variable speed motor. The speed is low at high Page on torque at light or no load, the motor speed attains dangerously high speed. The motor has a high starting torque.

**3.5 5V VOLTAGE REGULATOR (KA7805):**

** **

**Fig 3.7 Regulator module Fig 3.8 Pin Diagram of 7805**

****

**Fig 3.9 Circuit Connection of 7805**

**7805** shown in Fig 3.7 is a voltage regulator integrated circuit. It is a member of 78xx series of fixed linear voltage regulator ICs. The voltage source in a circuit may have fluctuations and would not give the fixed voltage output. The voltage regulator IC maintains the output voltage at a constant value. The xx in 78xx indicates the fixed output voltage it is designed to provide. 7805 provides +5V regulated power supply. Capacitors of suitable values can be connected at input and output pins depending upon the respective voltage levels.

**Table 3.5 Pin Description of 7805**

|  |  |  |
| --- | --- | --- |
| **Pin No** | **Function** | **Name** |
| 1 | Input voltage (5V-18V) | Input |
| 2 | Ground (0V) | Ground |
| 3 | Regulated output; 5V (4.8V-5.2V) | Output |

A regulated power supply is very much essential for several electronic devices due to the semiconductor material employed in them have a fixed rate of current as well as voltage. The device may get damaged if there is any deviation from the fixed rate. The AC power supply gets converted into constant DC by this circuit. By the help of a voltage regulator DC, unregulated output will be fixed to a constant voltage. The circuit is made up of linear voltage regulator 7805 along with capacitors and resistors with bridge rectifier made up from diodes. From giving an unchanging voltage supply to building confident that output reaches uninterrupted to the appliance, the diodes along with capacitors handle elevated efficient signal conveyal.



**CHAPTER 4**

**4 SYSTEM DESIGN**

This chapter will provide the system flow diagram and also explain the design of the various modules of the parking system. This system consists of some major phases which are described below.

**4.1 INTRODUCTION**

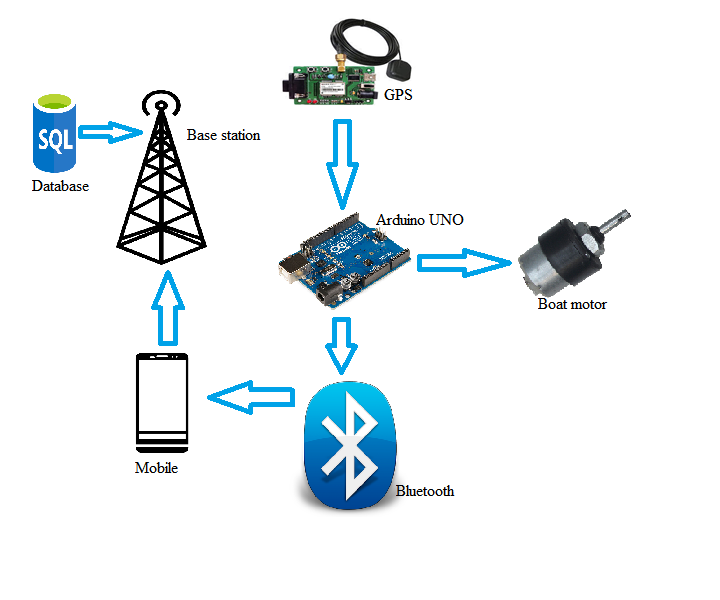
A sequence of modules is used in the system design. Each object embedded with Arduino Uno R3 board, GPS module, and 5V Regulator (KA7805).

Programs can be loaded on to it from the easy-to-use Arduino computer program. The Arduino has an extensive support community, which makes it a very easy way to get started working with embedded electronics.

The Arduino can "talk", (transmit or receive data data) via a serial channel, so any other device with serial capabilities can communicate with an Arduino. It doesn't matter what program/programming language is driving the other device.

Serial is used for communication between the Arduino board and a computer or other devices. All Arduino boards have at least one serial port (also known as a UART or USART): **Serial**. It communicates on digital pins 0 (RX) and 1 (TX) as well as with the computer via USB. Thus, if you use these functions, you cannot also use pins 0 and 1 for digital input or output.

There is also an Arduino environment's built-in serial monitor to communicate with an Arduino board. Click the serial monitor button in the toolbar and select the same baud rate used in the call to begin ().



**Fig 4.1 Graphical view of fishermen border alert system**

**4.2 FLOW CHART DIAGRAM**

**4.3 USE CASE DIAGRAM**

**4.4 DATABASE DESIGN**

**4.5 ARCHITECTURAL DIAGRAM**

**4.6 MODULES DESIGN**

* **LOCATION TRACKING AND ALERTING THE FISHERMEN.**
* **DISPLAYING E-MAP.**
* **MONITORING AND ALERT IN BASE STATION.**

**4.6.1 LOCATION TRACKING AND ALERTING THE FISHERMEN:**

**4.6.2 DISPLAYING E-MAP:**

**4.6.3 MONITORING AND ALERT IN BASE STATION:**

**4.7 ARDUINO IDE**

Arduino programs may be written in any programming language with a compiler that produces binary machine code. Atmel provides a development environment for their microcontrollers, AVR Studio and the newer Atmel Studio.

Arduino is an open-source hardware. The hardware reference designs are distributed under a Creative Commons Attribution Share-Alike 2.5 license and are available on the Arduino website. Layout and production files for some versions of the hardware are also available. The source code for the IDE is released under the GNU General Public License, version 2.

The Arduino project provides the Arduino integrated development environment (IDE), which is a cross-platform application written in the programming language Java. It originated from the IDE for the languages Processing and Wiring. It is designed to introduce programming to artists and other newcomers unfamiliar with software development.

It includes a code editor with features such as syntax highlighting, brace matching, and automatic indentation, and provides simple one-click mechanism to compile and load programs to an Arduino board. A program written with the IDE for Arduino is called a "sketch".

The Arduino IDE supports the languages C and C++ using special rules to organize code. The Arduino IDE supplies a software library called Wiring from the Wiring project, which provides many common input and output procedures. A typical Arduino C/C++ sketch consist of two functions that are compiled and linked with a program stub main () into an executable cyclic executive program:

**Setup ()**: a function that runs once at the start of a program and that can initialize settings.

**Loop ()**: a function called repeatedly until the board powers off.

A typical program for a beginning Arduino programmer blinks a light-emitting diode (LED) on and off. This program is usually loaded in the Arduino board by the manufacturer. In the Arduino environment, a user might write such a program as shown below.

#define LED\_PIN 13

void setup() {

pinMode(LED\_PIN, OUTPUT); // Enable pin 13 for digital output

}

void loop() {

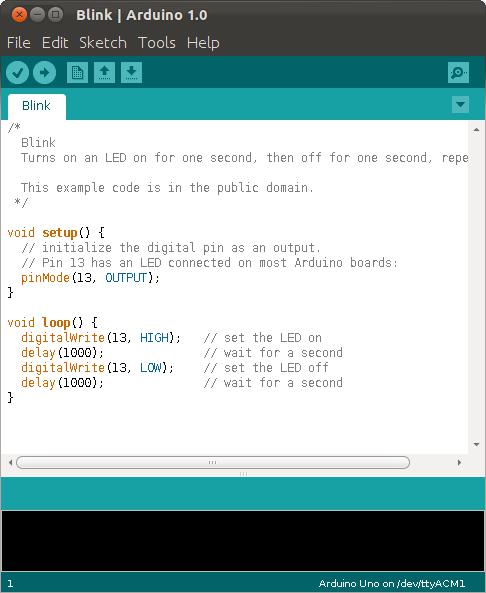
digitalWrite(LED\_PIN, HIGH); // Turn on the LED

delay(1000); // Wait one second (1000 milliseconds)

digitalWrite(LED\_PIN, LOW); // Turn off the LED

delay(1000); // Wait one second

}



**Fig 4.4 A sample program running in Arduino IDE**

After compiling and linking with the GNU toolchain, also included with the IDE distribution, the Arduino IDE employs the program avrdude to convert the executable code into a text file in hexadecimal coding that is loaded into the Arduino board by a loader program in the board's firmware.



**CHAPTER 5**

**SYSTEM IMPLEMENTATION**

**5.1 Filter Single Item Order Implementation**

In filter single item order, a Frequent Pattern (FP) Tree is constructed with the item sets which are taken from the transaction repository.

Algorithm : FP Tree

Input : A transaction database DB and a minimum support threshold.

Output : FP-tree, the frequent-pattern tree of DB.

Method : The FP-tree is constructed as follows:

1. Scan the transaction database DB once. Collect F, the set of frequent items, and the support of each frequent item. Sort F in support-descending order as FList, the list of frequent items.
2. Create the root of an FP-tree, T, and label it as “null”. For each transaction Trans in DB do the following:

* Select the frequent items in Trans and sort them according to the order of FList. Let the sorted frequent-item list in Trans be [ p | P], where p is the first element and P is the remaining list. Call insert tree([ p | P], T ).
* The function insert tree([ p | P], T ) is performed as follows: If T has a child N such that N.item-name = p.item-name then, increment N’s count by 1; else create a new node N, with its count initialized to 1, its parent link linked to T, and its node-link linked to the nodes with the same item-name via the node-link structure. If P is nonempty, call insert tree (P, N ) recursively.

**5.2 Frequent Item Set Mining Implementation**

In frequent item set mining, the Frequent Pattern Growth algorithm is used for finding the frequent item sets from the FP tree.

Algorithm : FP-Growth

Input : A database DB, represented by FP-tree constructed and a minimum support threshold.

Output : The complete set of frequent patterns.

Method : call FP-growth(FP-tree, null).

Procedure FP-growth(Tree, a) {

if Tree contains a single prefix path then // Mining single prefix-path FP-tree {

let P be the single prefix-path part of Tree;

let Q be the multipath part with the top branching node replaced by a null root;

for each combination (denoted as ß) of the nodes in the path P do

generate pattern ß ∪ a with support = minimum support of nodes in ß;

Let freq pattern set(P) be the set of patterns so generated;

}

else let Q be Tree;

for each item ai in Q do { // Mining multipath FP-tree

generate pattern ß = ai ∪ a with support = ai .support;

construct ß’s conditional pattern-base and then ß’s conditional FP-tree Tree ß;

if Tree ß ≠ Ø then

call FP-growth(Tree ß , ß);

Let freq pattern set(Q) be the set of patterns so generated; }

return(freq pattern set(P) ∪ freq pattern set(Q) ∪ (freq pattern set(P) × freq pattern set(Q)))

}

**5.3 Hyper Graph Construction**

The relationships identified by Frequent Item Set Mining are mapped to a weighted hyper graph. The properties of the constructed hyper graph is as follows.

1. Every item represent a node in the hyper graph.

2. For every frequent item set, an edge is created that connects all nodes representing the items present in the frequent item set and the edge weight is (number of transactions containing all items in the FIS / number of transactions containing at least one item in the FIS )

**5.4 Hyper Graph Clustering Implementation**

The hyper graph clustering is implemented using the Perron-Frobenius clustering algorithm in three steps namely

* Preprocessing
* Spectral Representation
* Clustering

1. **Preprocessing :**

Construct the graph and the similarity matrix representing the dataset.

1. **Spectral Representation :**

* The laplacian matrix L is calculated by the formula

L = D – A

where D is the degree matrix and A is the adjacency or vertex matrix of the graph.

* Compute eigenvalues and eigenvectors of the Laplacian matrix.
* Map each point to a lower-dimensional representation based on one or more eigenvectors.

1. **Clustering :**

* Recursively apply bi-partitioning algorithm in a hierarchical divisive manner.
* The basic idea is to recursively apply bi-partitioning algorithm in a hierarchical way: after partitioning the graph into two, reapply the same procedure to the sub-graphs.
* The number of groups is supposed to be given or directly controlled by the threshold allowed to the objective function.

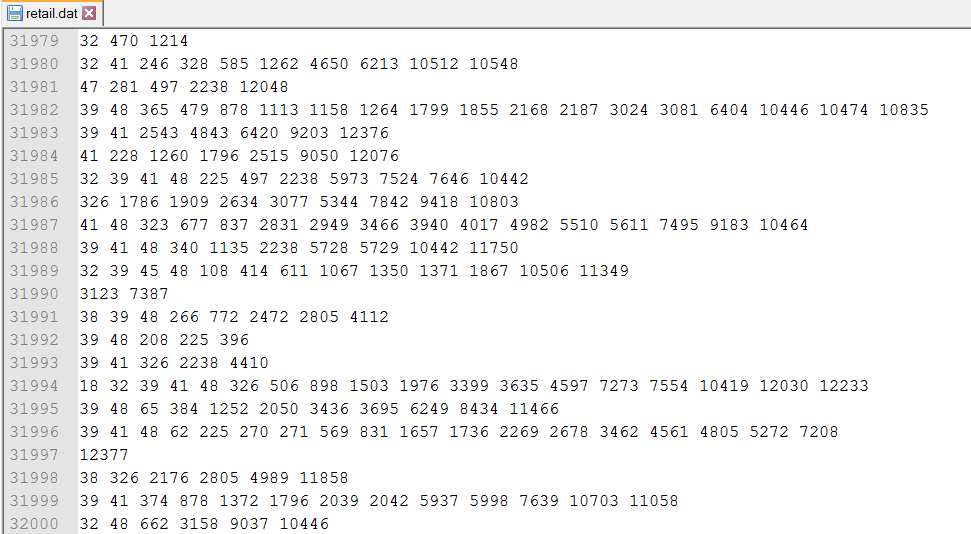


**CHAPTER 6**

**TEST CASES AND TEST RESULTS**

**6.1 TEST CASES AND RESULT**

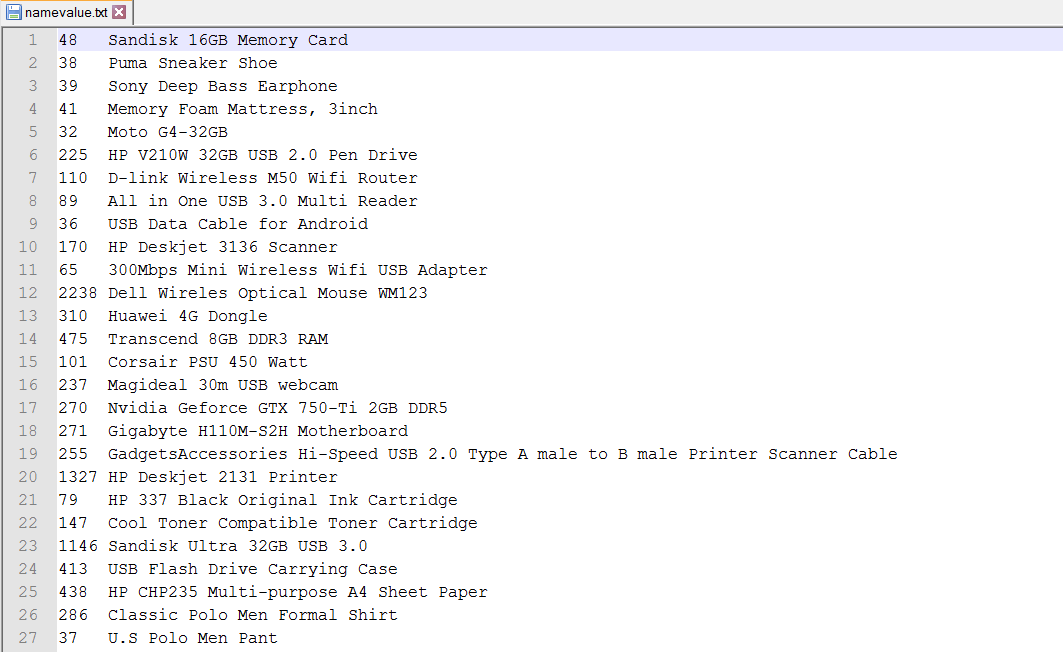
**6.1.1 Input Data**

****

**Figure 6.1:** Transaction data

Figure 6.1 shows the input data (i.e.) the online transaction details of the retailer from the Data Repository. The each line represents a single transaction and numbers in each transaction is the product id. This is the initial input data used in finding the frequent item sets.

The transaction data consists of nearly 80,000 transactions and more than 600 items. These transaction data is used for finding the cluster of frequent item sets.



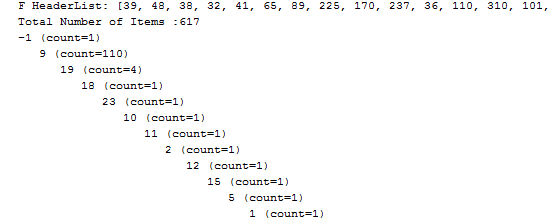
**Figure 6.2:** Product ID and its corresponding Name

Figure 6.2 shows the product id of the product and the corresponding name of the product used in the transaction data. The NamingItem class uses this file as input to name the product id with the corresponding value.

**6.1.2 Filter Single Item Order Testing and Results**

**Table 6.1:** Test case for Filter Single Item Order

|  |  |
| --- | --- |
| ***Test Case ID*** | ***T1*** |
| ***Test Case Description*** | *Construction of FP tree* |
| ***Test Data*** | *A transaction database DB and a minimum support threshold.* |
| ***Expected output*** | *FP-tree, the frequent-pattern tree of DB.* |
| ***Test Result*** | *PASS* |



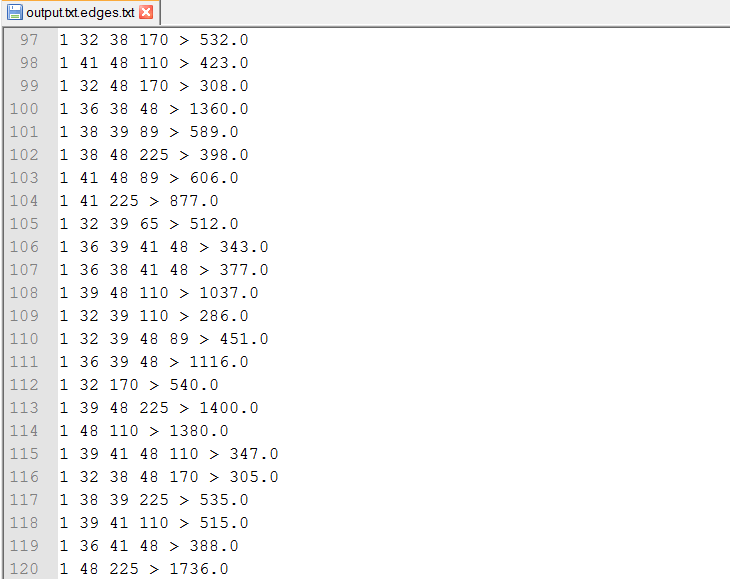
**Figure 6.3:** Frequent Pattern Tree

Figure 6.3 shows the Filtering Single Item Order by construction of the FP Tree which contains all the transaction in the transaction repository. The Frequent Pattern tree structure which contains the all the transaction data above the support threshold is constructed for the item sets.

**6.1.3 Frequent Item Set Mining Testing and Results**

**Table 6.2:** Test case for Frequent Item SetMining

|  |  |
| --- | --- |
| ***Test Case ID*** | ***T2*** |
| ***Test Case Description*** | *Construction of FP growth tree* |
| ***Test Data*** | *A database DB, represented by FP-tree constructed, and a minimum support threshold* |
| ***Expected output*** | *The complete set of frequent patterns* |
| ***Test Result*** | *PASS* |

****

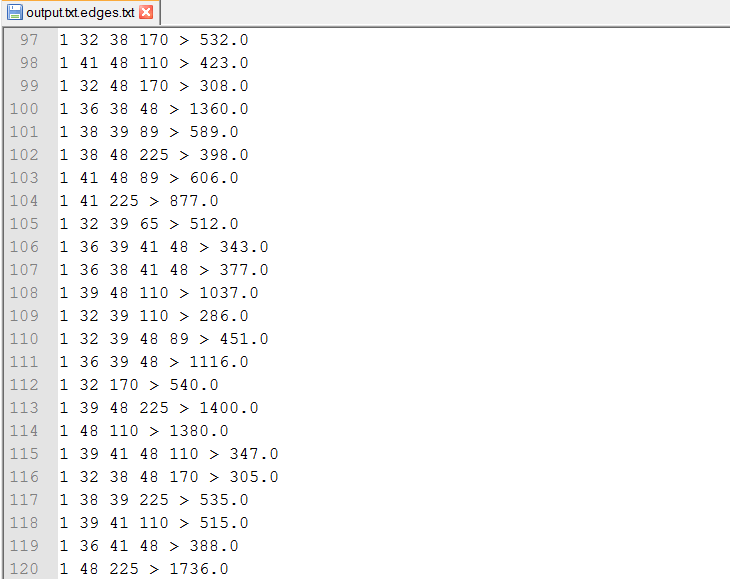
**Figure 6.4 :** FP Growth Tree

Figure 6.4 shows the frequent item set mining by the construction of the FP growth tree. The FP tree with a minimum threshold is given as input and a complete set of frequent patterns are generated with the support count at the end of each transactions which are represented after ‘>’ symbol.

**6.1.4 Hyper Graph Construction Testing and Results**

**Table 6.3:** Test case for Hyper Graph Construction

|  |  |
| --- | --- |
| ***Test Case ID*** | ***T3*** |
| ***Test Case Description*** | *Construction of Hyper Graph* |
| ***Test Data*** | *FP growth tree with weighted edges* |
| ***Expected output*** | *Hyper Graph edges* |
| ***Test Result*** | *PASS* |

****

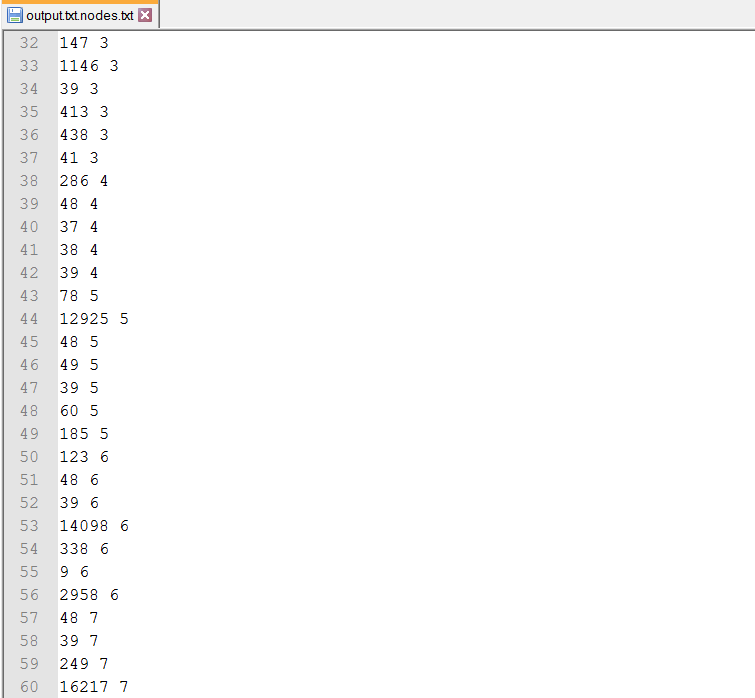
**Figure 6.5:** Hyper Graph edges

Figure 6.5 shows the hyper graph edges constructed from the FP growth tree. The single item transactions are eliminated while constructing hyper graph from the weighted graph.

**6.1.5 Hyper Graph Clustering Testing and Results**

**Table 6.4:** Test case for Hyper Graph Clustering

|  |  |
| --- | --- |
| ***Test Case ID*** | ***T4*** |
| ***Test Case Description*** | *Clustering of Frequent Items.* |
| ***Test Data*** | *Hyper graph edges* |
| ***Expected output*** | *Product Clusters* |
| ***Test Result*** | *PASS* |

****

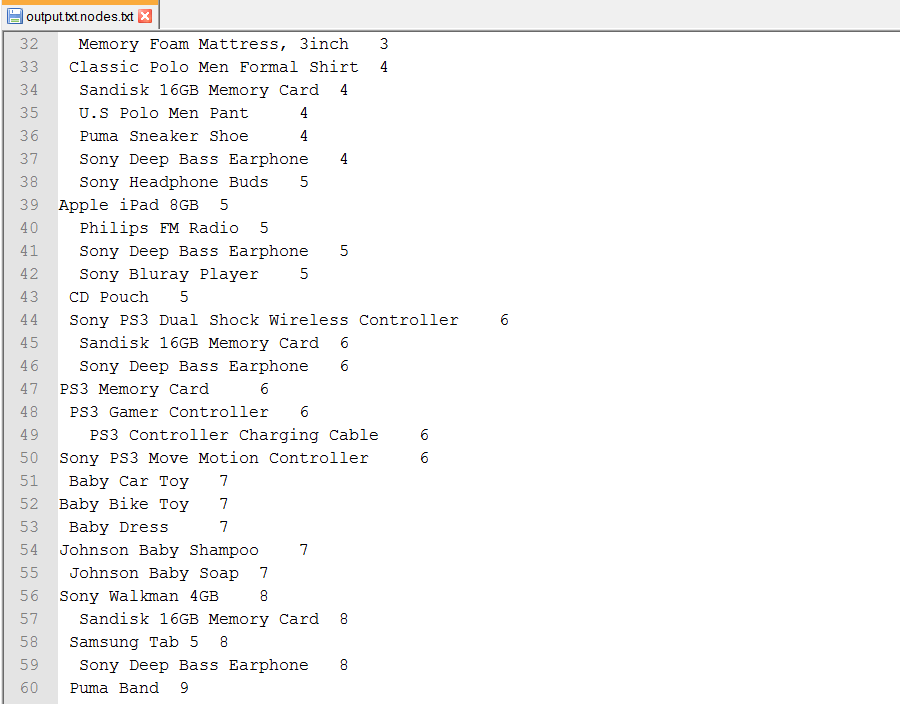
**Figure 6.6:** Hyper Graph clustering

Figure 6.6 shows the cluster of frequent item sets obtained by the clustering of hyper graph. The hyper graph is clustered using bi-partitioning algorithm to form the product clusters.

**6.1.6 Naming the Product IDs**

**Table 6.5:** Test Case for Naming the Product ID

|  |  |
| --- | --- |
| ***Test Case ID*** | ***T5*** |
| ***Test Case Description*** | *Naming the product ID with their corresponding names* |
| ***Test Data*** | *namevalue.txt* |
| ***Expected output*** | *output file with name and its clusters* |
| ***Test Result*** | *PASS* |

****

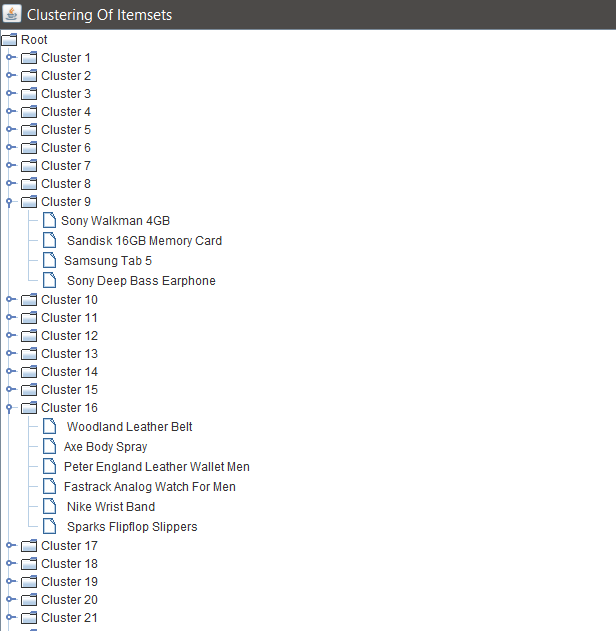
**Figure 6.7:** Naming with product id.

Figure 6.7 shows the list of product clusters whose product id is assigned with corresponding names using Hashmap.

**6.1.7 Representation of Clusters**

**Table 6.6:** Test Case for Representation of Clusters

|  |  |
| --- | --- |
| ***Test Case ID*** | ***T6*** |
| ***Test Case Description*** | *Representation of the clusters in a Jtree structure* |
| ***Test Data*** | *Output file consists of name and its clusters* |
| ***Expected output*** | *Jtree structure with clusters and its product* |
| ***Test Result*** | *PASS* |

****

**Figure 6.8:** Representation of clusters.

Figure 6.8 shows the list of clusters which are obtained after the hyper graph clustering are represented using the Jtree.

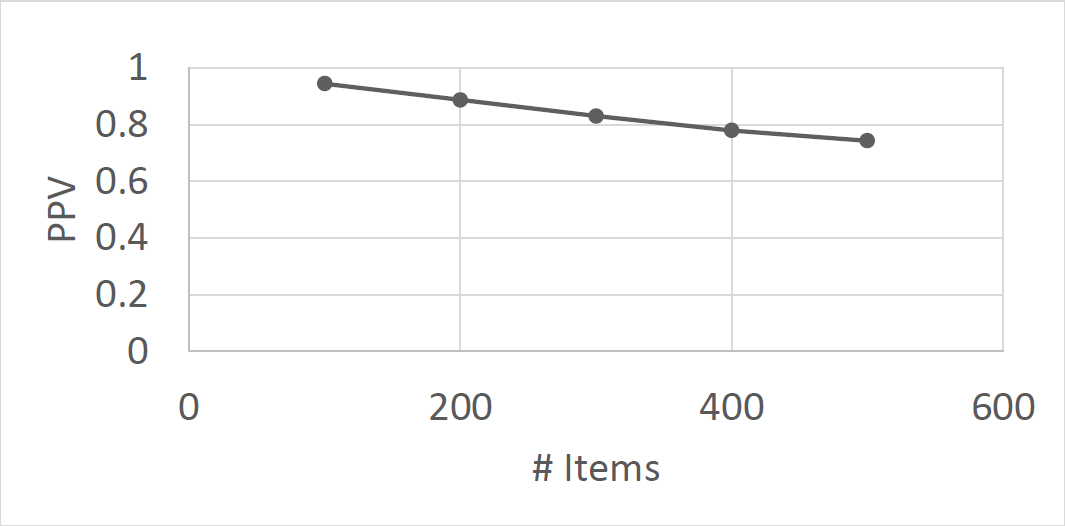
**6.1.8 Positive Predictive Value (PPV)**

Positive Predictive Value can be defined as follows

𝑃𝑃𝑉 = 𝑛𝑢𝑚𝑏𝑒𝑟 𝑜𝑓 𝑡𝑟𝑢𝑒 𝑝𝑜𝑠𝑖𝑡𝑖𝑣𝑒𝑠

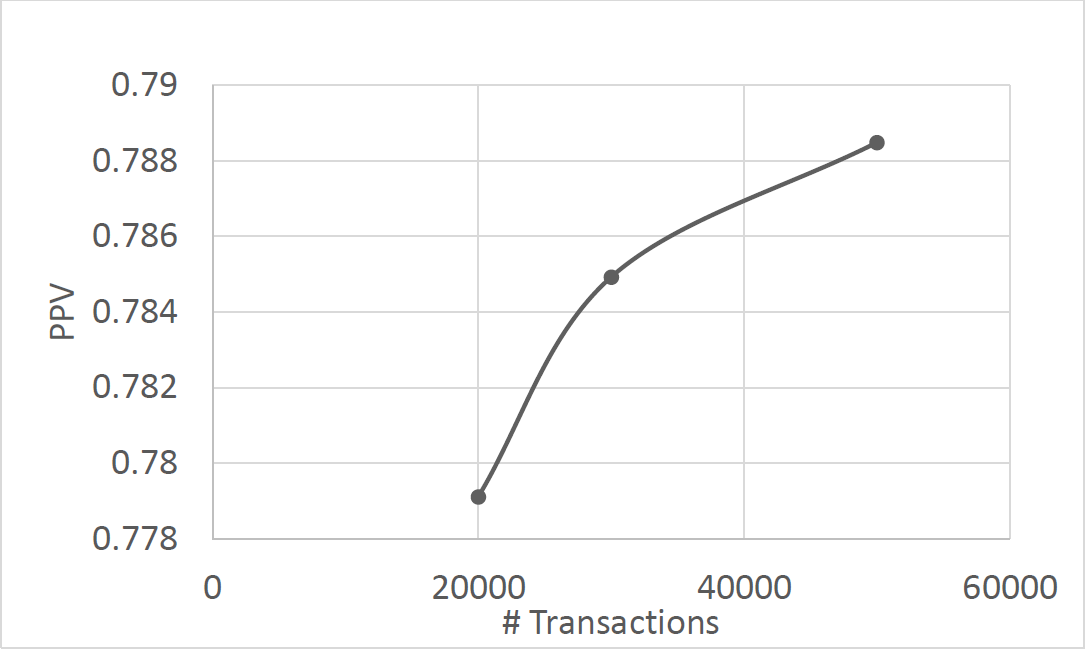
𝑛𝑢𝑚𝑏𝑒𝑟 𝑜𝑓 𝑡𝑟𝑢𝑒 𝑝𝑜𝑠𝑖𝑡𝑖𝑣𝑒𝑠 + 𝑛𝑢𝑚𝑏𝑒𝑟 𝑜𝑓 𝑓𝑎𝑙𝑠𝑒 𝑝𝑜𝑠𝑖𝑡𝑖𝑣𝑒𝑠

The value of PPV when the number of items is increased with fixed order count was studied. The result is shown in fig 6.9. The number of orders were fixed at 80,000. It is clear from the chart that as number of items increases the PPV decreases.



**Figure 6.9:** PPV vs number of items

Keeping the number of distinct items as 500 the effect of number of transaction is studied (Fig 6.10). When the number of transactions increase PPV increases. This is because more data is available for learning leading to better prediction. Using this data the number of transaction required to produce a particular PPV can be found given the number of distinct items.

****

**Figure 6.10:** PPV vs number of transactions

**CHAPTER 7**

**CONCLUSION**

Thus a system that tracks and alerts the fishermen preventing them from crossing the border is implemented. The speed of the engine is controlled automatically. Their current position displayed on the electronic map in phone keeps them aware of their position.

A notification system by which alert messages on account of emergencies can be sent to fishermen was implemented. Central tracking of all fishermen is done in base station

**FUTURE ENHANCEMENT**

Future optimizations to the algorithm can help the online retailers who are at different locations to sell their product by packing them in the same package which are in the same cluster.

In current methodology the relationship between orders made by same user has not been used. Making use of this may produce better clusters. Also items belonging to same product category are more likely to be bought together. Such features can also be brought into the algorithm.

Number of orders made through online portals are increasing. A framework for clustering has been proposed in this work. Scalability of the algorithm for large datasets has to be studied in detail.

**Appendix A**

**SOURCE CODE**