



INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

ONLINE TRANSACTION FRAUD DETECTION

¹Lahari Madabhattula, ²Maridu Manikanta, ³Pradeep Kumar

¹Student, ²Student, ³Professor

¹Lovely Professional University,

²Lovely Professional University,

³Lovely Professional University

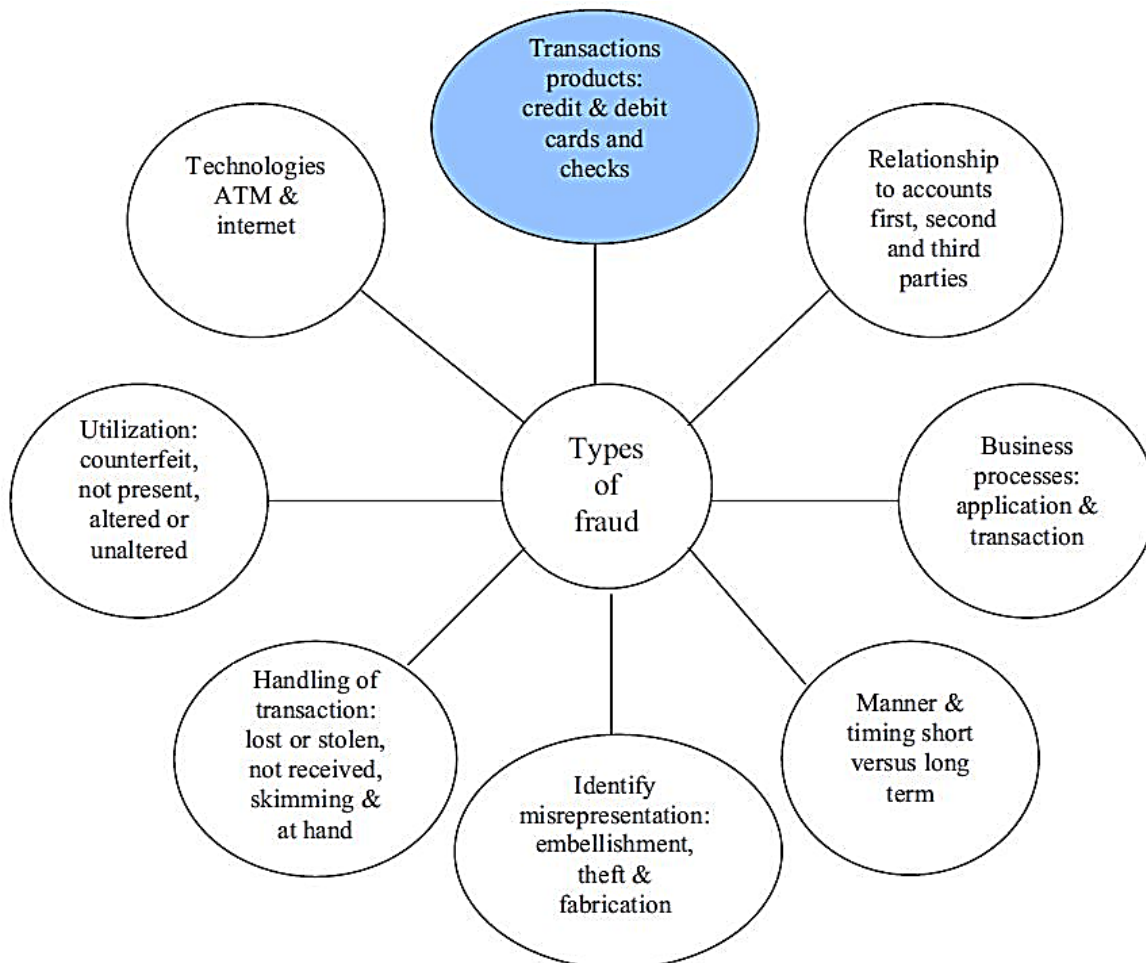
PROJECT OVERVIEW

Introduction

Today due to rapid growth in e-commerce online shopping or online transaction is grown day by day. The mode of payment is done by credit card. The credit card users are increasing day by day. It was reported that there are almost 430 million credit and debit card users across whole Europe. As the number of credit/debit card users increasing, the fraudulent users are also increasing. There are two types of credit cards.

- 1] Physical card.
- 2] Virtual card.

In the physical card, the user has to show the card while making payment. In this type, if fraudulent user wants to access his/her card then he just needs to steal that card. In virtual card, the fraudulent user needs to know the information about details information about credit card such as, CVV no, Secure code, Credit card number. Therefore the secure payment gateway is needed to identify the user and to verify that the user is legal or attacker. The most useful and appropriate technique used for fraud detection is Behavior and Location Analysis (BLA).



Scope & Objectives

The system prevents fraudulent users from misusing the details of the credit-card of the genuine users for their personal gain. The spending habits of the credit-card owner is detect the fraud. As the fake user might not be aware of the spending habits of the owner, there will be an irregularity in the spending pattern, which the system will detect. The owner is immediately alerted about the attempted fraud and the transaction is blocked. Thus, the system protects legitimate users from financial loss. The system helps in making electronic payment safer and more reliable. The principles in the proposed system can also be adopted and implemented in other electronic payment services such as online banking facility and payment gateways.

- Creating an application to detect fraud Credit Cards.
- Implementing Hidden Markov model.
- Creating database containing all relevant information of Customer.
- Providing security to the customers at the time of transaction.

Modules and their Description

This system comprises of 2 major modules with its sub-modules:

1. Admin

- a. **Login:** Admin need to login using valid login credentials in order to access the system.
- b. **Add / View Products:** Admin can add new product with its details into the system.
- c. **View Transactions:** System allows admin to view all the transactions done by the registered users.

2. User

- a. **Registration:** Here, user first need to registration themselves with details to access the system.
- b. **Login:** After a successful registration, user then need to login into the system by inputting their credentials into the system.
- c. **View Products:** User can view multiple products with its details. Interested users can purchase a product via online transaction.
- d. **Buy a Product:** User can select payment mode to perform transactions by providing the card details like card no., CVV code, Expiry Date and Holders name.
- e. **View Transaction:** List of all the transactions will be displayed to the user.

Existing System & Proposed System

❖ Problem with current scenario

In case of credit card fraud detection, the existing system is detecting the fraud after fraud has been happen. Existing system maintain the large amount of data when customer comes to know about inconsistency in transaction, he/she made complaint and then fraud detection system start it working. It first tries to detect that fraud has actually occur after that it starts to track fraud location and so on. In case of existing system there is no confirmation of recovery of fraud and customers satisfaction

PROPOSED SYSTEM

The aim of the proposed system is to develop a website which has capability to restrict and block the transaction performing by attacker from genuine user's credit card details. The system here is developed for the transactions higher than the customers current transaction limit. As we seen the existing system detects the fraud after fraud has been occurred i.e. based on customers complained. The proposed system tries to detect fraudulent transaction before transaction succeed.

In proposed system, while registration we take required information which is efficient to detect fraudulent user activity

- In proposed system, I present a Behavior and Location Analysis (BLA).
- Which does not require fraud signatures and yet is able to detect frauds by considering a cardholder's spending habit.
- Card transaction processing sequence by the stochastic process of a BLA.
- The details of items purchased in Individual transactions are usually not known to any Fraud Detection System (FDS) running at the bank that issues credit cards to the cardholders.
- Hence, I feel that BLA is an ideal choice for addressing this problem.

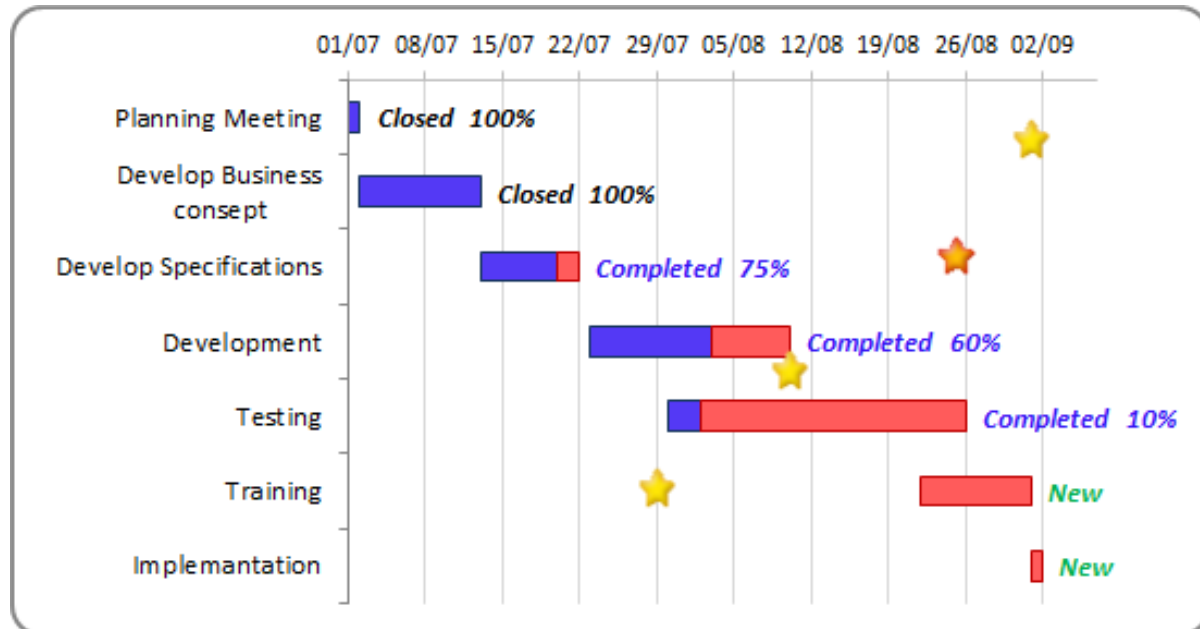
- Another important advantage of the BLA - based approach is a drastic reduction in the number of False Positives transactions identified as malicious by an FDS although they are actually genuine.
- An FDS runs at a credit card issuing bank. Each incoming transaction is submitted to the FDS for verification.
- FDS receives the card details and the value of purchase to verify, whether the transaction is genuine or not.

- The types of goods that are bought in that transaction are not known to the FDS.
- It tries to find any anomaly in the transaction based on the spending profile of the cardholder, shipping address, and billing address, etc.
- If the FDS confirms the transaction to be of fraud, it raises an alarm, and the issuing bank declines the transaction.

The credit card fraud detection features use user behavior and location scanning to check for unusual patterns. These patterns include user characteristics such as user spending patterns as well as usual user geographic locations to verify his identity. If any unusual pattern is detected, the system requires re-verification.

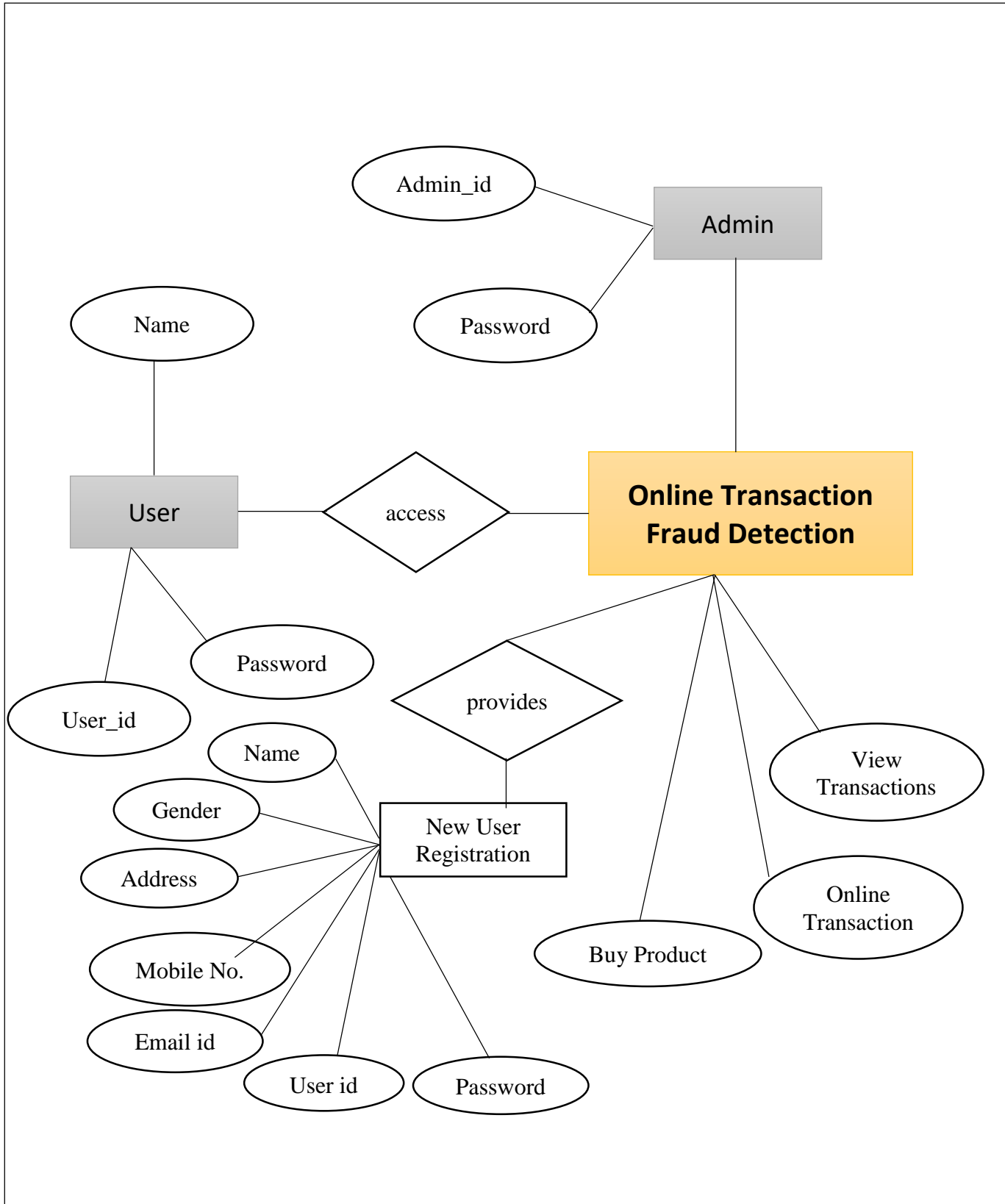
The system analyses user credit card data for various characteristics. These characteristics include user country, usual spending procedures. Based upon previous data of that user the system recognizes unusual patterns in the payment procedure. So now the system may require the user to login again or even block the user for more than 3 invalid attempts.

Gantt Chart



PROJECT DESIGN

E-R Diagram



Use Case Diagram

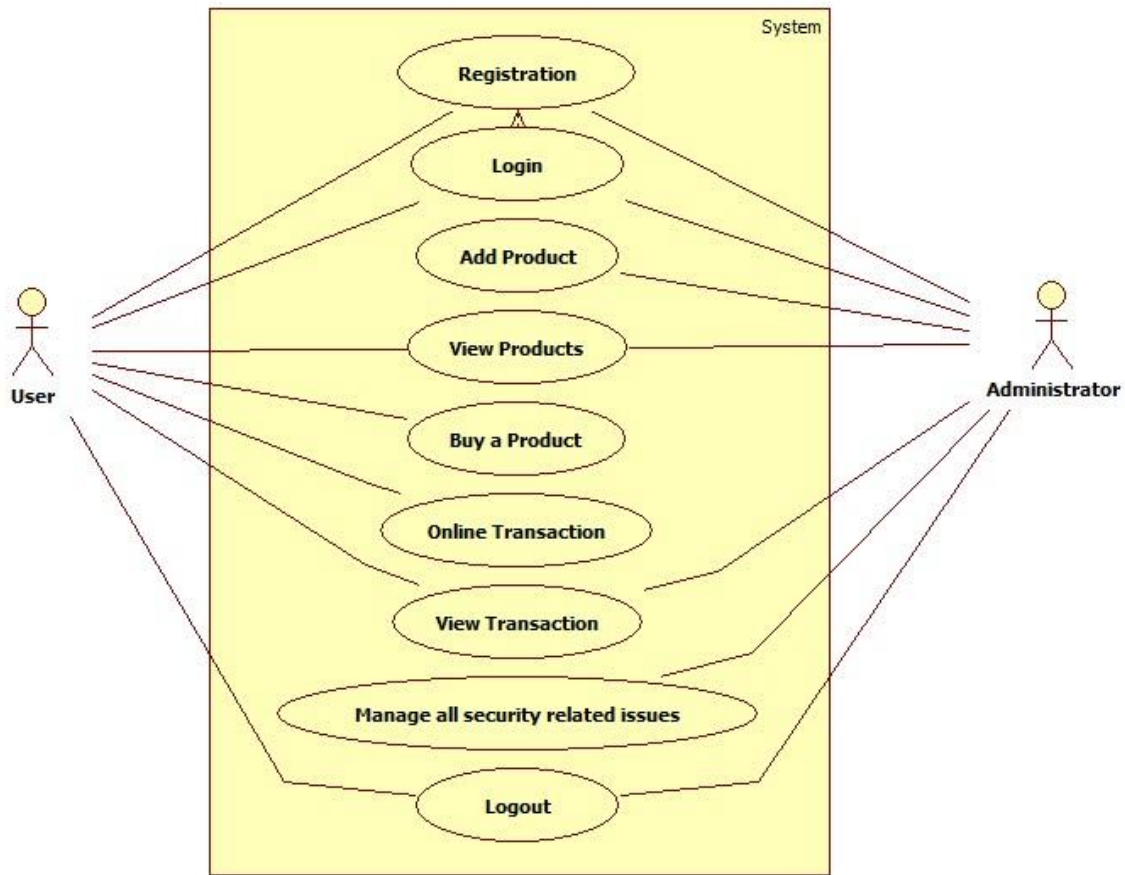
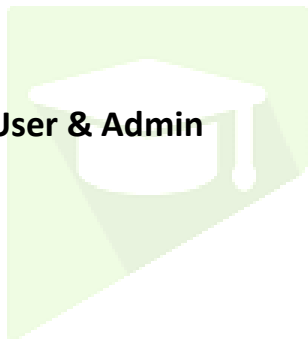
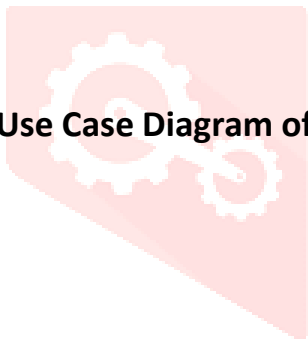
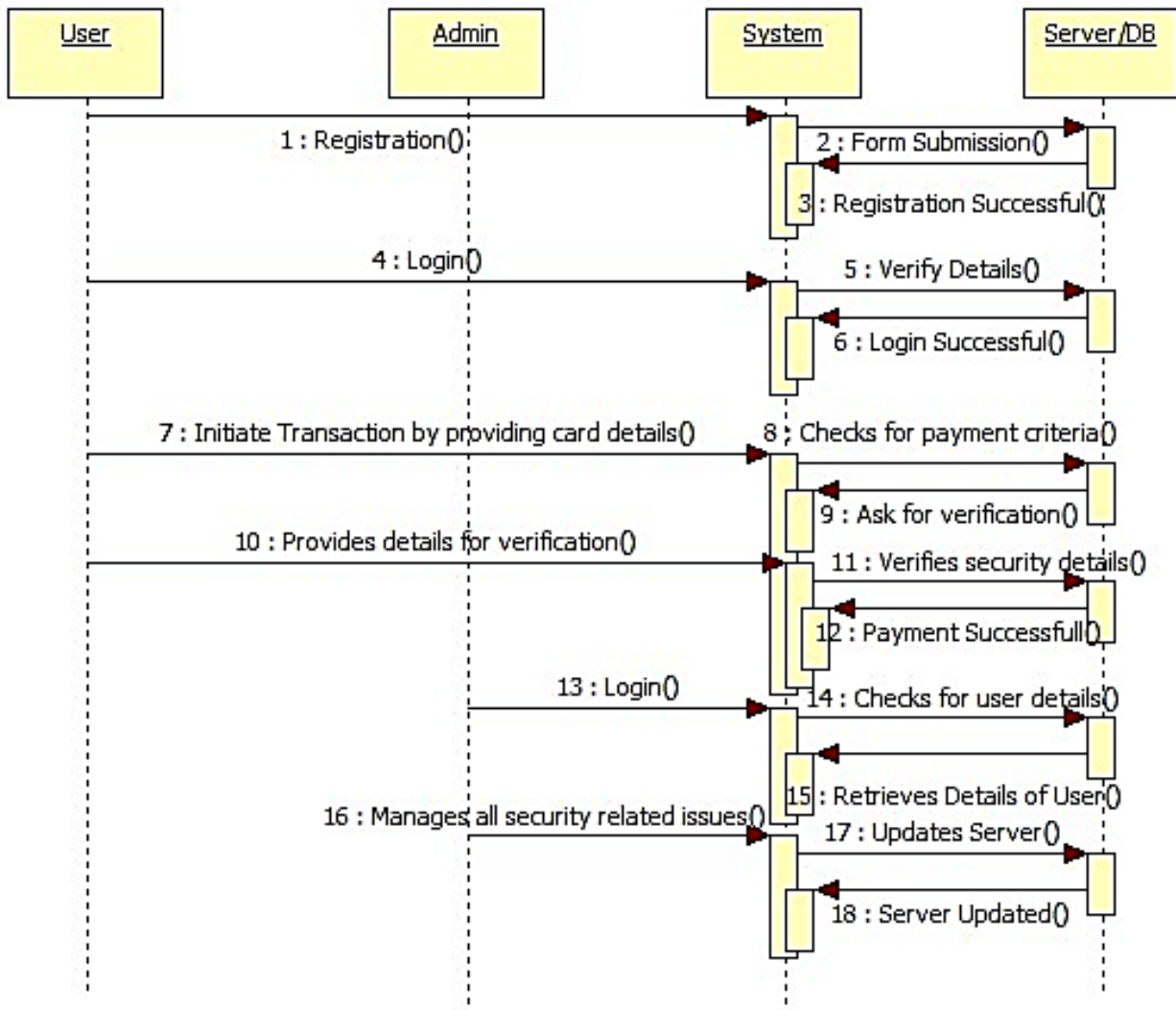


Fig. Use Case Diagram of User & Admin



Sequence Diagram



Activity Diagram

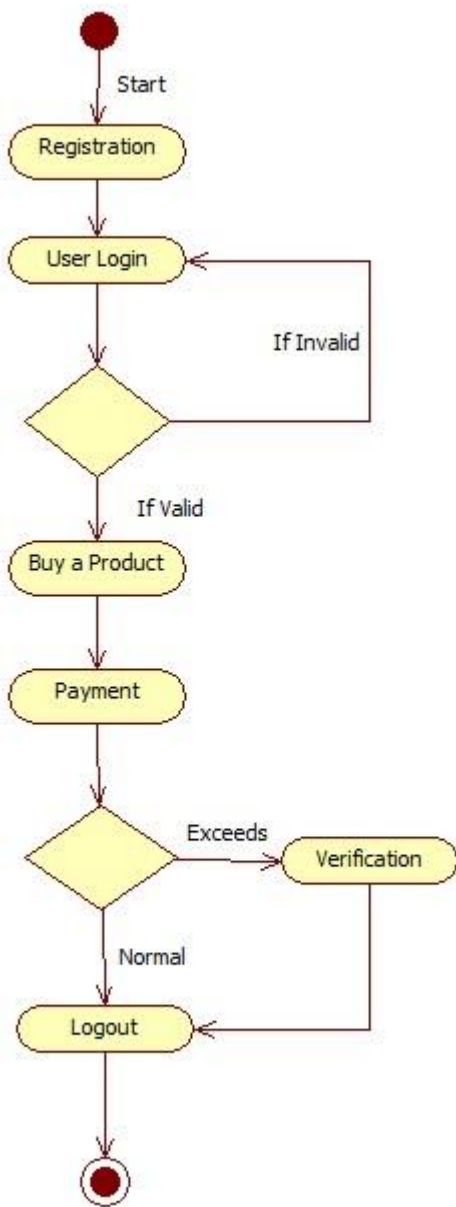


Fig. Activity Diagram of User

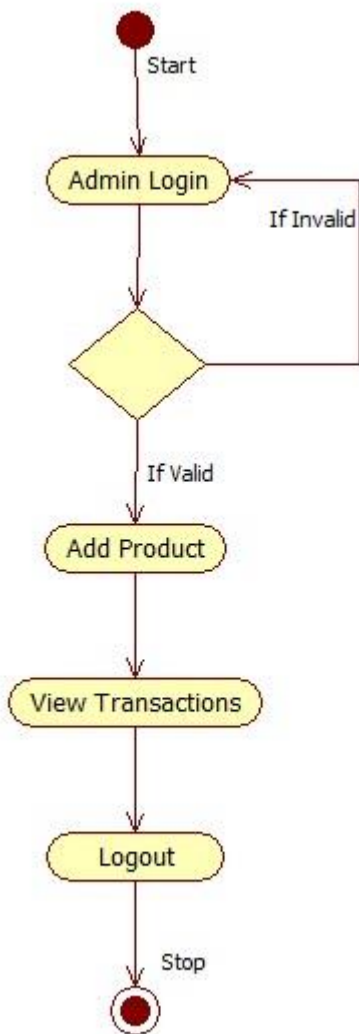
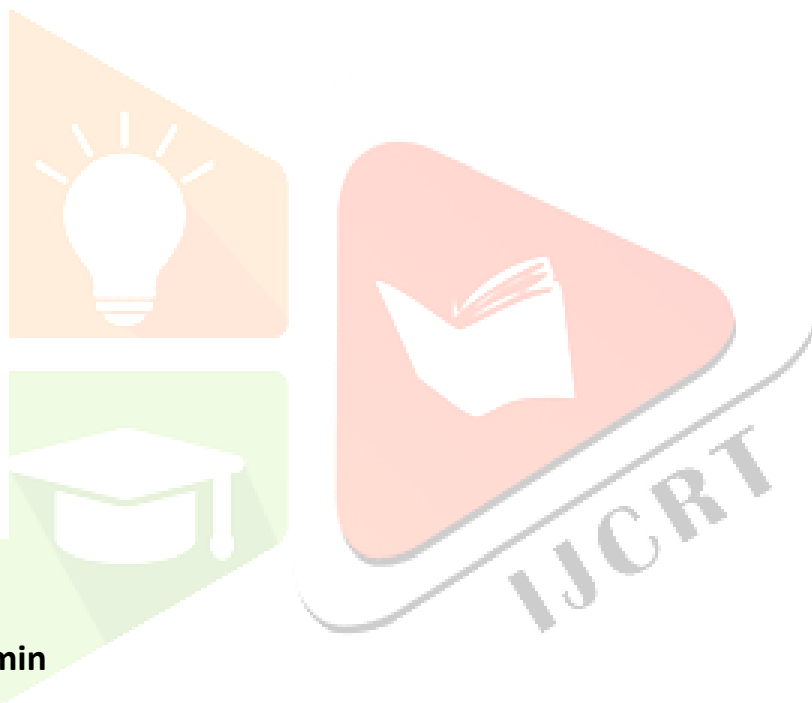
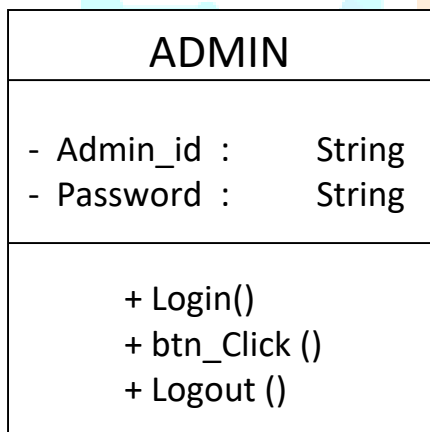
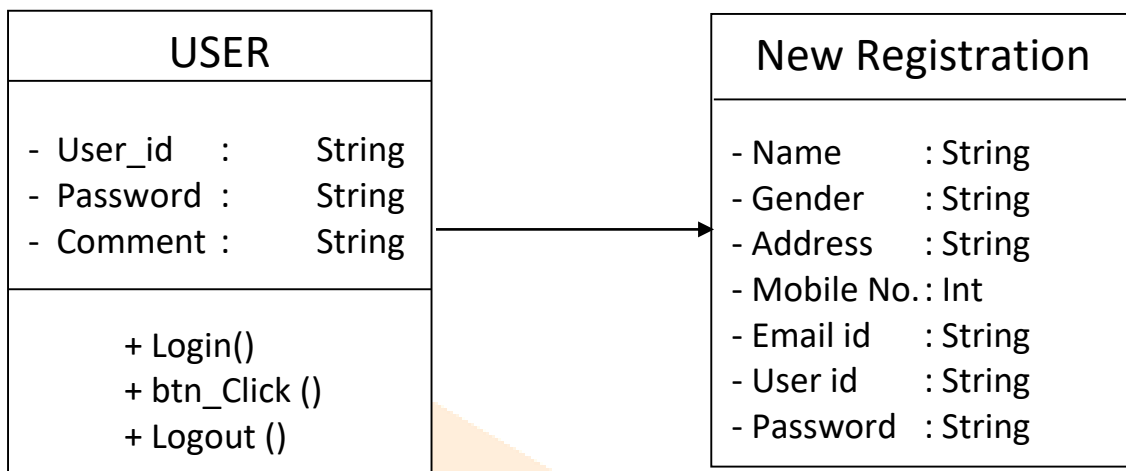


Fig. Activity Diagram of Admin



Class Diagram



Data Flow Diagram

A data flow diagram is graphical tool used to describe and analyze movement of data through a system. These are the central tool and the basis from which the other components are developed. The transformation of data from input to output, through processed, may be described logically and independently of physical components associated with the system. These are known as the logical data flow diagrams. The physical data flow diagrams show the actual implements and movement of data between people, departments and workstations. A full description of a system actually consists of a set of data flow diagrams. Using two familiar notations Yourdon, Gane and Sarson notation develops the data flow diagrams. Each component in a DFD is labeled with a descriptive name. Process is further identified with a number that will be used for identification purpose. The development of DFD's is done in several levels. Each process in lower level diagrams can be broken down into a more detailed DFD in the next level. The top-level diagram is often called context diagram. It consists a single process bit, which plays vital role in studying the current system. The process in the context level diagram is exploded into other process at the first level DFD.

The idea behind the explosion of a process into more process is that understanding at one level of detail is exploded into greater detail at the next level. This is done until further explosion is necessary and an adequate amount of detail is described for analyst to understand the process.

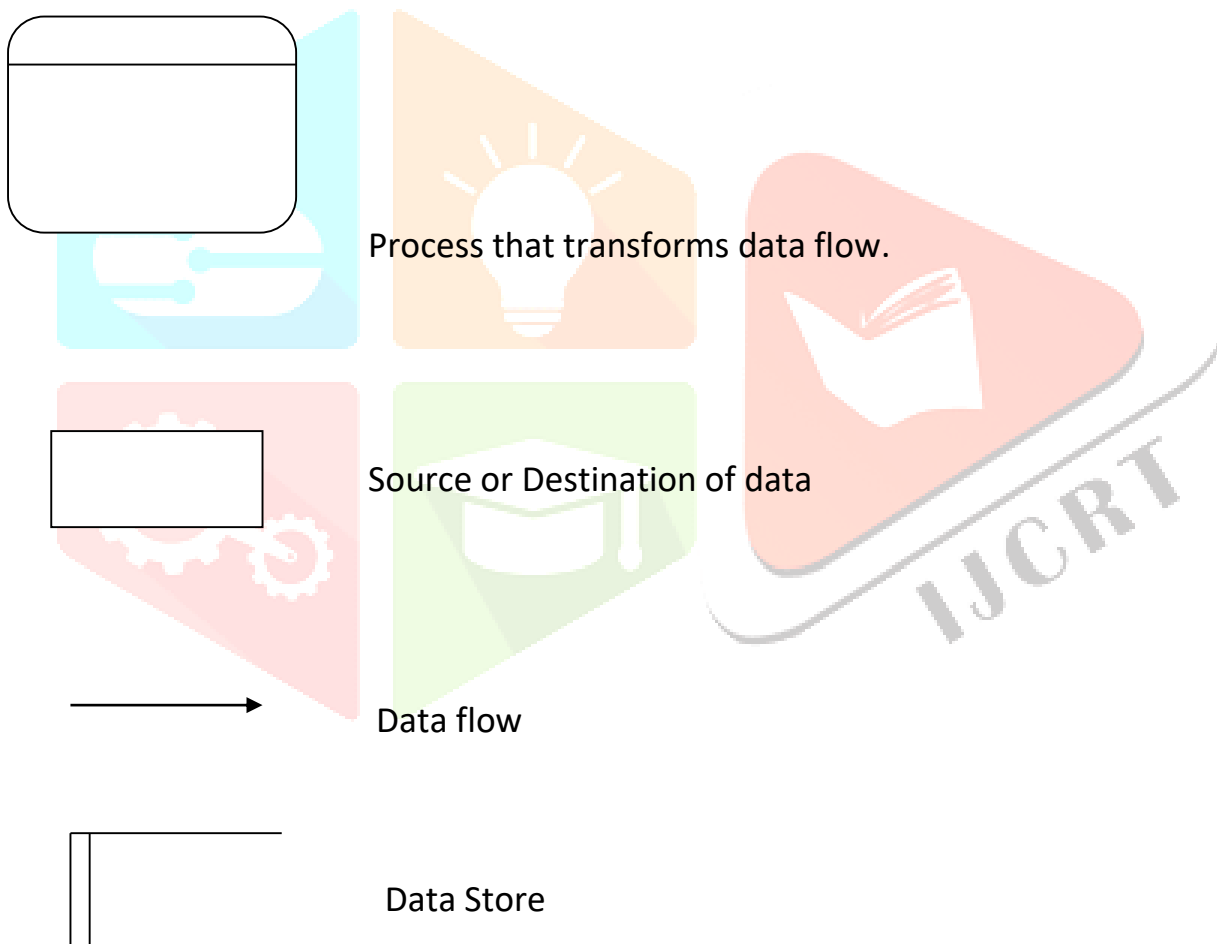
Larry Constantine first developed the DFD as a way of expressing system requirements in a graphical form, this lead to the modular design.

A DFD is also known as a "bubble Chart" has the purpose of clarifying system requirements and identifying major transformations that will become programs in system design. So it is the starting point of the design to the lowest level of detail. A DFD consists of a series of bubbles joined by data flows in the system.

DFD SYMBOLS:

In the DFD, there are four symbols

1. A square defines a source(originator) or destination of system data
2. An arrow identifies data flow. It is the pipeline through which the information flows
3. A circle or a bubble represents a process that transforms incoming data flow into outgoing data flows.
4. An open rectangle is a data store, data at rest or a temporary repository of data



CONSTRUCTING A DFD:

Several rules of thumb are used in drawing DFD's:

1. Process should be named and numbered for an easy reference. Each name should be representative of the process.
2. The direction of flow is from top to bottom and from left to right. Data traditionally flow from source to the destination although they may flow back to the source. One way to indicate this is to draw long flow line back to a source. An alternative way is to repeat the source symbol as a destination. Since it is used more than once in the DFD it is marked with a short diagonal.
3. When a process is exploded into lower level details, they are numbered.
4. The names of data stores and destinations are written in capital letters. Process and dataflow names have the first letter of each word capitalized

A DFD typically shows the minimum contents of data store. Each data store should contain all the data elements that flow in and out.

Questionnaires should contain all the data elements that flow in and out. Missing interfaces redundancies and like is then accounted for often through interviews.

SAILENT FEATURES OF DFD's

1. The DFD shows flow of data, not of control loops and decision are controlled considerations do not appear on a DFD.
2. The DFD does not indicate the time factor involved in any process whether the data flows take place daily, weekly, monthly or yearly.
3. The sequence of events is not brought out on the DFD.

TYPES OF DATA FLOW DIAGRAMS

1. Current Physical
2. Current Logical
3. New Logical
4. New Physical

CURRENT PHYSICAL:

In Current Physical DFD process label include the name of people or their positions or the names of computer systems that might provide some of the overall system-processing label includes an identification of the technology used to process the data. Similarly data flows and data stores are often labels with the names of the actual physical media on which data are stored such as file folders, computer files, business forms or computer tapes.

CURRENT LOGICAL:

The physical aspects at the system are removed as much as possible so that the current system is reduced to its essence to the data and the processors that transform them regardless of actual physical form.

NEW LOGICAL:

This is exactly like a current logical model if the user were completely happy with the user were completely happy with the functionality of the current system but had problems with how it was implemented typically through the new logical model will differ from current logical model while having additional functions, absolute function removal and inefficient flows recognized.

NEW PHYSICAL:

The new physical represents only the physical implementation of the new system.

RULES GOVERNING THE DFD'S

PROCESS

- 1) No process can have only outputs.
- 2) No process can have only inputs. If an object has only inputs than it must be a sink.
- 3) A process has a verb phrase label.

DATA STORE

- 1) Data cannot move directly from one data store to another data store, a process must move data.
- 2) Data cannot move directly from an outside source to a data store, a process, which receives, must move data from the source and place the data into data store
- 3) A data store has a noun phrase label.

SOURCE OR SINK

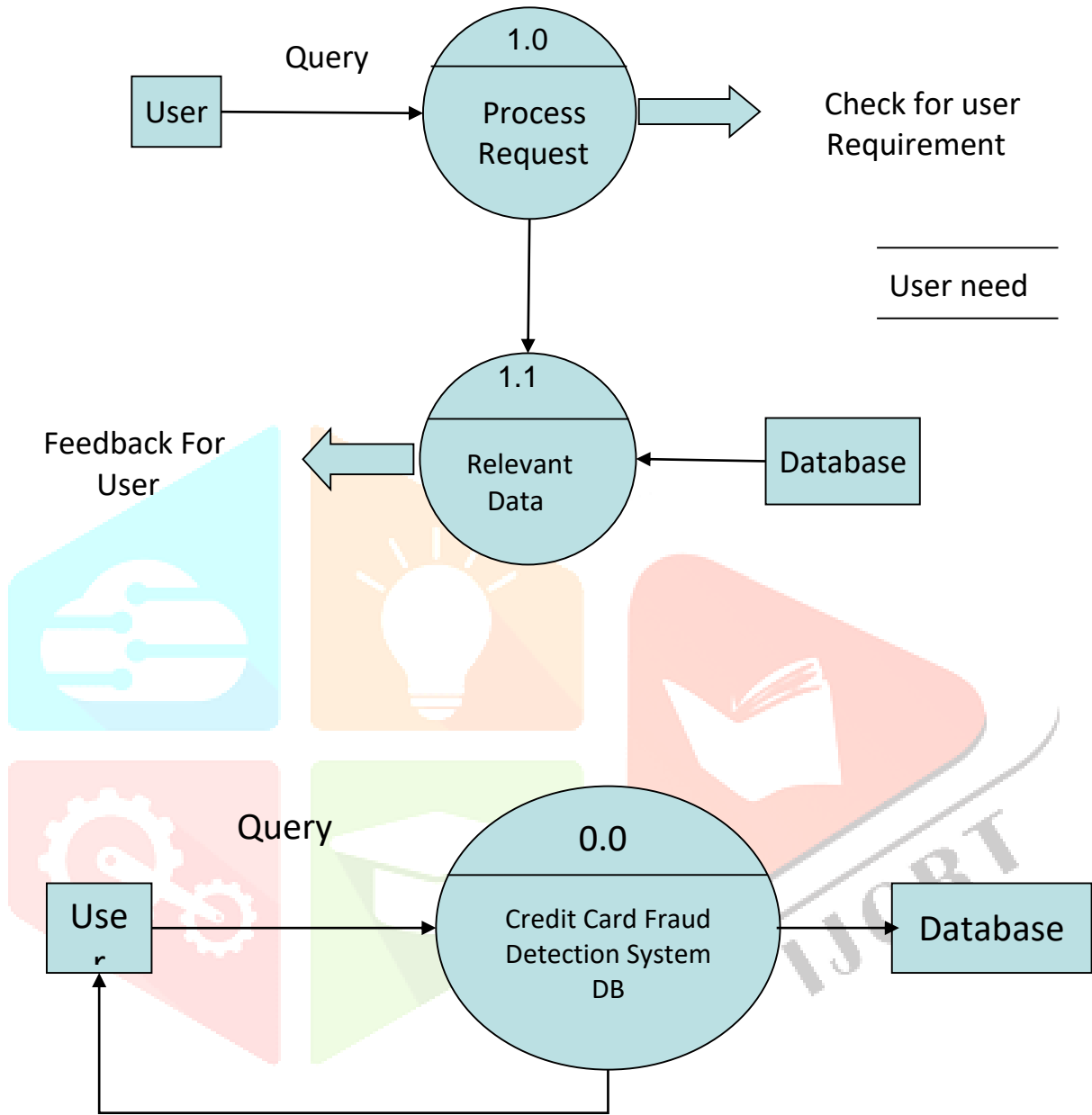
The origin and /or destination of data.

- 1) Data cannot move directly from a source to sink it must be moved by a process
- 2) A source and /or sink has a noun phrase label

DATA FLOW

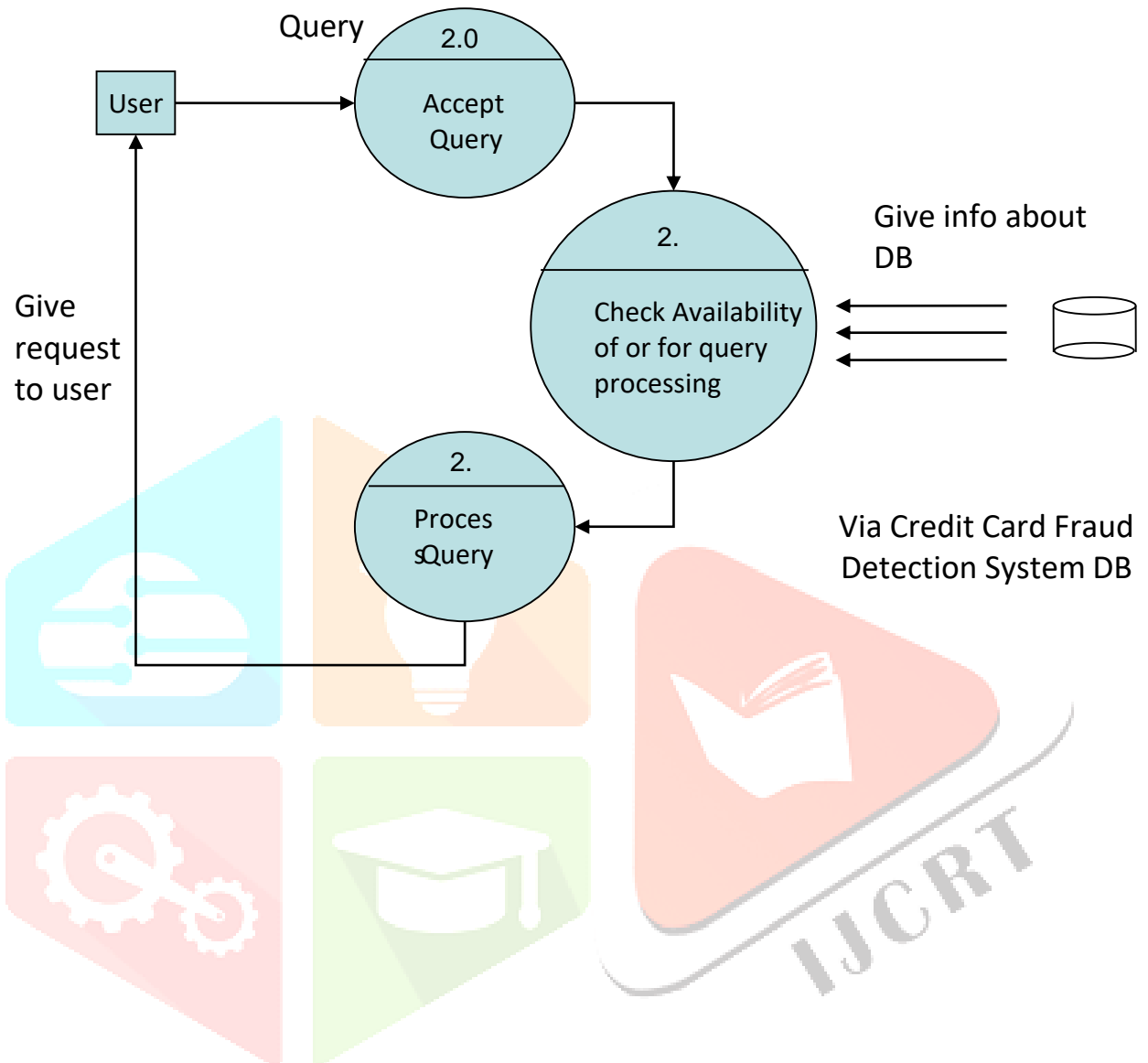
- 1) A Data Flow has only one direction of flow between symbols. It may flow in both directions between a process and a data store to show a read before an update. The latter is usually indicated however by two separate arrows since these happen at different types.
- 2) A join in DFD means that exactly the same data comes from any of two or more different processes, data store or sink to a common location.
- 3) A data flow cannot go directly back to the same process it leads. There must be at least one other process that handles the data flow, produces some other data flow, returns the original data into the beginning process.
- 4) A Data flow to a data store means update (delete or change).
- 5) A data Flow from a data store means retrieve or use.

Data Flow Diagrams



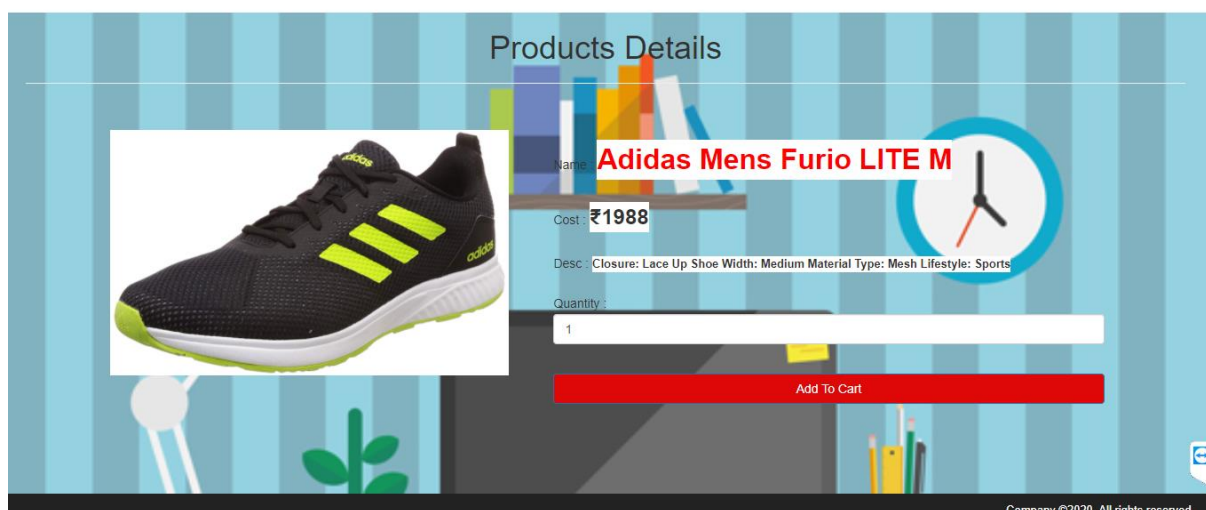
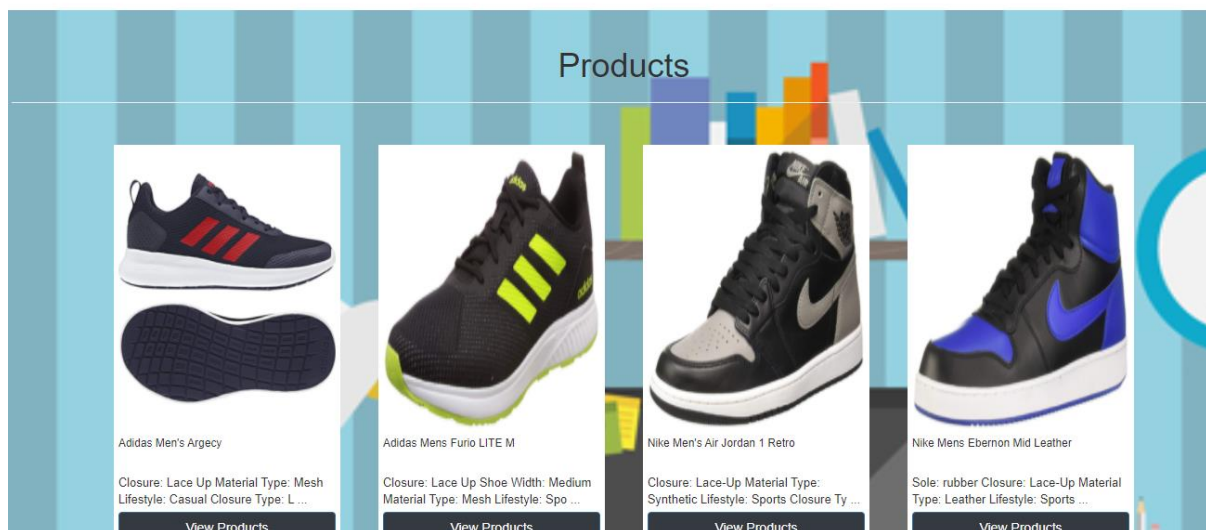
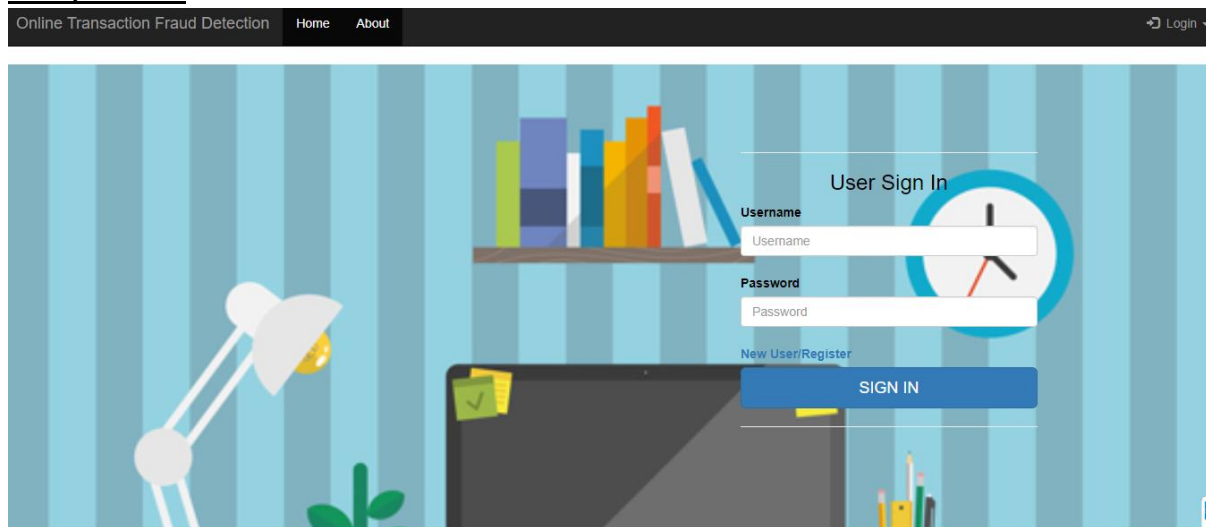
DATABASE DETAIL

LEVEL 1 DFD



LEVEL 2 DFD: PREDICTION

Snapshots



Cart

Product ID	Product Name	Cost Per/unit	Category	Quantity	Total	User id	Delete
001	Adidas Men's Argecy	2433	sneakers	1	2433	None	Delete
002	Adidas Mens Furio LITE M	1988	sneakers	1	1988	None	Delete

Total Amount
4421

Payment Continue Shopping

Product ID	Product Name	Cost Per/unit	Category	Quantity	Total	User id	Delete
001	Adidas Men's Argecy	2433	sneakers	1	2433	None	Delete
002	Adidas Mens Furio LITE M	1988	sneakers	1	1988	None	Delete

Total Amount
4421

Card Details

/

Make Payment

Back Continue Shopping

localhost:8000 says
details doesnt match

OK

PROJECT IMPLEMENTATION

Project Implementation Technology

The Project is designed and developed in Django Framework. We used Django Framework for coding of the project. Created and maintained all databases into MySQL Server, in that we create tables, write query for store data or record of project.

❖ Hardware Requirement:

- Processor –Core i3
- Hard Disk – 160 GB
- Memory – 1GB RAM
- Monitor

❖ Software Requirement:

- Windows 7 or higher
- Python
- Django framework
- MySQL database



OVERVIEW OF TECHNOLOGIES USED

INTRODUCTION

Python is a powerful multi-purpose programming language created by Guido van Rossum. It has simple easy-to-use syntax, making it the perfect language for someone trying to learn computer programming for the first time. This is a comprehensive guide on how to get started in Python, why you should learn it and how you can learn it. However, if you have some knowledge of other programming languages and want to quickly get started with Python. Python is a general-purpose language. It has wide range of applications from Web development (like: Django and Bottle), scientific and mathematical computing (Orange, SymPy, NumPy) to desktop graphical user Interfaces (Pygame, Panda3D). The syntax of the language is clean and length of the code is relatively short. It's fun to work in Python because it allows you to think about the problem rather than focusing on the syntax.



Features of Python Programming:

- A simple language which is easier to learn
Python has a very simple and elegant syntax. It's much easier to read and write Python programs compared to other languages like: C++, Java, C#. Python makes programming fun and allows you to focus on the solution rather than syntax. If you are a newbie, it's a great choice to start your journey with Python.
- Free and open-source
You can freely use and distribute Python, even for commercial use. Not only can you use and distribute software's written in it, you can even make changes to the Python's source code. Python has a large community constantly improving it in each iteration.
- Portability
You can move Python programs from one platform to another, and run it without any changes. It runs seamlessly on almost all platforms including Windows, Mac OS X and Linux.
- Extensible and Embeddable
Suppose an application requires high performance. You can easily combine pieces of C/C++ or other languages with Python code. This will give your application high performance as well as scripting capabilities which other languages may not provide out of the box.

- A high-level, interpreted language
Unlike C/C++, you don't have to worry about daunting tasks like memory management, garbage collection and so on. Likewise, when you run Python code, it automatically converts your code to the language your computer understands. You don't need to worry about any lower-level operations.
- Large standard libraries to solve common tasks
Python has a number of standard libraries which makes life of a programmer much easier since you don't have to write all the code yourself. For example: Need to connect MySQL database on a Web server? You can use MySQLdb library using `import MySQLdb`. Standard libraries in Python are well tested and used by hundreds of people. So, you can be sure that it won't break your application.
- Object-oriented
Everything in Python is an object. Object oriented programming (OOP) helps you solve a complex problem intuitively. With OOP, you are able to divide these complex problems into smaller sets by creating objects.

Features of Python Programming:

Web Applications

You can create scalable Web Apps using frameworks and CMS (Content Management System) that are built on Python. Some of the popular platforms for creating Web Apps are: Django, Flask, Pyramid, Plone, Django CMS.

Sites like Mozilla, Reddit, Instagram and PBS are written in Python.

Scientific and Numeric Computing

There are numerous libraries available in Python for scientific and numeric computing. There are libraries like: SciPy and NumPy that are used in general purpose computing. And, there are specific libraries like: EarthPy for earth science, AstroPy for Astronomy and so on.

Also, the language is heavily used in machine learning, data mining and deep learning.

Creating software Prototypes

Python is slow compared to compiled languages like C++ and Java. It might not be a good choice if resources are limited and efficiency is a must.

However, Python is a great language for creating prototypes. For example: You can use Pygame (library for creating games) to create your game's prototype first. If you like the prototype, you can use language like C++ to create the actual game.

Good Language to Teach Programming

Python is used by many companies to teach programming to kids and newbies.

It is a good language with a lot of features and capabilities. Yet, it's one of the easiest languages to learn because of its simple easy-to-use syntax.

Syntax Overview



Simple Elegant Syntax

Programming in Python is fun. It's easier to understand and write Python code. Why? The syntax feels natural. Take this source code for an example:

```
a = 2
b = 3
sum = a + b
print(sum)
```

Even if you have never programmed before, you can easily guess that this program adds two numbers and prints it.

Not overly strict

You don't need to define the type of a variable in Python. Also, it's not necessary to add semicolon at the end of the statement.

Python enforces you to follow good practices (like proper indentation). These small things can make learning much easier for beginners.

Expressiveness of the language

Python allows you to write programs having greater functionality with fewer lines of code. Here's a link to the source code of Tic-tac-toe game with a graphical interface and a smart

computer opponent in less than 500 lines of code. This is just an example. You will be amazed how much you can do with Python once you learn the basics.

Great Community and Support

Python has a large supporting community. There are numerous active forums online which can be handy if you are stuck. Some of them are:

Learn Python subreddit

Google Forum for Python

Python Questions - Stack Overflow

Django documentation

Django is a high-level Python Web framework that encourages rapid development and clean, pragmatic design. Built by experienced developers, it takes care of much of the hassle of Web development, so you can focus on writing your app without needing to reinvent the wheel. It's free and open source.

Features of Django

- Rapid Development
- Secure
- Scalable
- Fully loaded
- Versatile
- Open Source
- Vast and Supported Community

Rapid Development

Django was designed with the intention to make a framework which takes less time to build web application. The project implementation phase is a very time taken but Django creates it rapidly.

Secure

Django takes security seriously and helps developers to avoid many common security mistakes, such as SQL injection, cross-site scripting, cross-site request forgery etc. Its user authentication system provides a secure way to manage user accounts and passwords.

Scalable

Django is scalable in nature and has ability to quickly and flexibly switch from small to large scale application project.

Fully loaded

Django includes various helping task modules and libraries which can be used to handle common Web development tasks. Django takes care of user authentication, content administration, site maps, RSS feeds etc.

Versatile

Django is versatile in nature which allows it to build applications for different-different domains. Now a days, Companies are using Django to build various types of applications like: content management systems, social networks sites or scientific computing platforms etc.

Open Source

Django is an open source web application framework. It is publicly available without cost. It can be downloaded with source code from the public repository. Open source reduces the total cost of the application development.

Vast and Supported Community

Django is a one of the most popular web frameworks. It has widely supportive community and channels to share and connect.



WAMP Server

Introduction

WAMP is a Windows OS based program that installs and configures Apache web server, MySQL database server, PHP scripting language, phpMyAdmin (to manage MySQL database's), and SQLiteManager (to manage SQLite database's). WAMP is designed to

offer an easy way to install Apache, PHP and MySQL package with an easy to use installation program instead of having to install and configure everything yourself. WAMP is so easy because once it is installed it is ready to go. You don't have to do any additional configuring or tweaking of any configuration files to get it running.

There are usually two reasons why someone chooses to install WAMP. They are looking to install WAMP for development purposes or to run their own server.

WAMP Server Contains

PHP Admin

Allows you to change or add users and for making new databases phpMyAdmin is a free software tool written in PHP, intended to handle the administration of MySQL over the World Wide Web. phpMyAdmin supports a wide range of operations with MySQL. The most frequently used operations are supported by the user interface (managing databases, tables, fields, relations, indexes, users, permissions, etc.), while you still have the ability to directly execute any SQL statement.

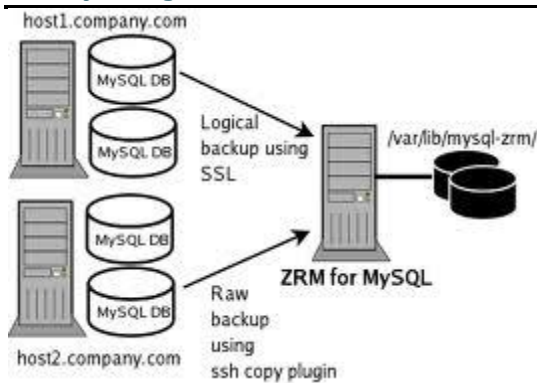
Features

- Intuitive web interface
- Support for most MySQL features:
 - Browse and drop databases, tables, views, fields and indexes.
 - Create, copy, drop, rename and alter databases, tables, fields and indexes.
 - Maintenance server, databases and tables, with proposals on server configuration.
 - Execute, edit and bookmark any SQL-statement, even batch-queries.
 - Manage MySQL users and privileges
 - Manage stored procedures and triggers.

- Import data from CSV and SQL
- Export data to various formats: CSV, SQL, XML, PDF, ISO/IEC 26300 - OpenDocument Text and Spreadsheet, Word, LATEX and others
- Administering multiple servers
- Creating PDF graphics of your database layout
- Creating complex queries using Query-by-example (QBE)
- Searching globally in a database or a subset of it
- Transforming stored data into any format using a set of predefined functions, like displaying BLOB-data as image or download-link
- And much more...



SQL Server is a relational database management system from Microsoft that's designed for the enterprise environment. SQL Server runs on T-SQL (Transact -SQL), a set of programming extensions from Sybase and Microsoft that add several features to standard SQL, including transaction control, exception and error handling, row processing, and declared variables.



Generically, any database management system (DBMS) that can respond to queries from client machines formatted in the SQL language. When capitalized, the term generally refers to either of two database management products from Sybase and Microsoft. Both companies offer client-server DBMS products called SQL Server.

Using WAMP as a Development Server

You can use WAMP to develop and test websites locally on their own computer instead of having to get a web hosting account to develop with. Most people will be using WAMP for development purposes such as learning how to create websites with HTML, PHP, and MySQL.

Using WAMP as a Production Server

WARNING: WAMP was designed to be a testing and development server, not an actual production server. WAMP does not come with any real security in place so it offers no protection from any kind of attack. Any 10-year-old with access to the internet can easily hack your WAMP server.

If your website(s) have highly sensitive data (such as credit card numbers, social security numbers, user ids, passwords, etc.), you need to take this in consideration before your put this information online. Unless you are an experienced system administrator and can configure WAMP to be more secure, you should never user WAMP for a production server.

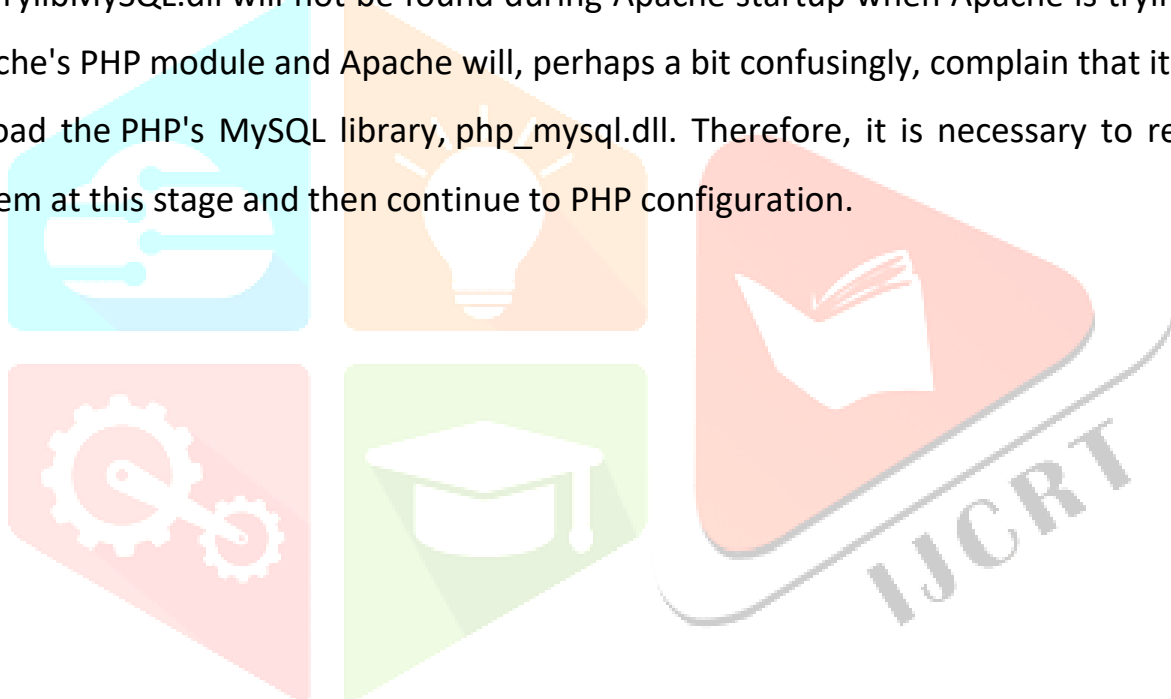
MySQL Configuration

To begin MySQL installation, first download latest version of Essentials as an MSI package.

During MySQL installation, select Typical installation and use default configuration values except for Sign-Up where you probably want to select Skip Sign-Up. When Setup Wizard is completed, make sure the option Configure the MySQL Server now is set. For MySQL Server Instance Configuration, select Standard Configuration. Next, you must set option Include Bin Directory in Windows PATH. This setting is crucial, otherwise a required library, libMySQL.dll, will not be found later during Apache startup.

Finally, enter a proper root password. There is no need to neither enable remote root access nor create an Anonymous Account.

Please inspect messages during MySQL startup and verify that MySQL has been started successfully. Then, you must reboot the system. Otherwise, the required library libMySQL.dll will not be found during Apache startup when Apache is trying to load Apache's PHP module and Apache will, perhaps a bit confusingly, complain that it is unable to load the PHP's MySQL library, php_mysql.dll. Therefore, it is necessary to reboot the system at this stage and then continue to PHP configuration.



CODING

```
FOLDERS
  OnlineTransactionFraudDetection
  AppOnlineTransactionFraudDetection
    __pycache__
    migrations
    static
    Templates
    /_init_.py
    /admin.py
    /apps.py
    /forms.py
    /models.py
    /tests.py
    /urls.py
    /views.py
  assets
  media
  OnlineTransactionFraudDetection
    __pycache__
    /_init_.py
    /settings.py
    /urls.py
    /wsgi.py
    db.sqlite3
    /manage.py

settings.py
  1 from django.db import models
  2
  3
  4 class Admin_Details(models.Model):
  5     Username = models.CharField(default=None,max_length=100)
  6     Password = models.CharField(default=None,max_length=100)
  7
  8     class Meta:
  9         db_table = 'Admin_Details'
  10
  11
  12 class User_details(models.Model):
  13     FirstName = models.CharField(default=None,max_length=100)
  14     LastName = models.CharField(default=None,max_length=100)
  15     Username = models.CharField(default=None,max_length=100)
  16     Email = models.EmailField(default=None)
  17     Password = models.CharField(default=None,max_length=100)
  18     Card_number = models.CharField(default=None,max_length=100)
  19     cvv = models.CharField(default=None,max_length=100)
  20     Expirydate = models.CharField(default=None,max_length=100)
  21     mobile = models.CharField(default=None,max_length=100)
  22     Securityquestion1 = models.CharField(default=None,max_length=500)
  23     Securityanswer1 = models.CharField(default=None,max_length=200)
  24     Securityquestion2 = models.CharField(default=None,max_length=500)
  25     Securityanswer2 = models.CharField(default=None,max_length=200)
  26     Securityquestion3 = models.CharField(default=None,max_length=500)
  27     Securityanswer3 = models.CharField(default=None,max_length=200)
  28     attempts = models.IntegerField(default=0)
  29     status = models.CharField(max_length=200,default='Unlocked')
  30
  31     class Meta:
  32         db_table = 'user_details'
  33
  34
  35 class Products(models.Model):
  36     Pid = models.CharField(max_length=200)
  37     pname = models.CharField(max_length=200)
  38     category = models.CharField(max_length=200)
  39     subcategory = models.CharField(max_length=10)
  40     cost = models.IntegerField(default=None)
  41     desc = models.CharField(max_length=200)
  42     quantity = models.CharField(max_length=200)
  43     image = models.ImageField(upload_to='img/images')
  44
  45     class Meta:
  46         db_table = 'products'
  47
  48
```

```
FOLDERS
  OnlineTransactionFraudDetection
  AppOnlineTransactionFraudDetection
    __pycache__
    migrations
    static
    Templates
    /_init_.py
    /admin.py
    /apps.py
    /forms.py
    /models.py
    /tests.py
    /urls.py
    /views.py
  assets
  media
  OnlineTransactionFraudDetection
    __pycache__
    /_init_.py
    /settings.py
    /urls.py
    /wsgi.py
    db.sqlite3
    /manage.py

models.py
  48
  49
  50 class Card_details(models.Model):
  51     Cid = models.CharField(max_length=20)
  52     uid = models.CharField(max_length=20)
  53     card_type = models.CharField(max_length=200)
  54     cardno = models.CharField(max_length=200)
  55     cvv = models.CharField(max_length=10)
  56     expiry = models.CharField(max_length=200)
  57     mobil = models.CharField(max_length=11)
  58     Email = models.EmailField()
  59     password = models.CharField(max_length=200)
  60     amount = models.IntegerField(default=None)
  61
  62     class Meta:
  63         db_table = 'card_details'
  64
  65
  66 class Order_details(models.Model):
  67     Tid = models.CharField(max_length=20)
  68     uid = models.CharField(max_length=20)
  69     TotalAmount = models.IntegerField(null=True,default=None)
  70     date = models.CharField(max_length=200)
  71     productList = models.CharField(max_length=500,default=None)
  72     quantity = models.CharField(max_length=200,default=None)
  73     status = models.CharField(max_length=200,default=None)
  74
  75     class Meta:
  76         db_table = 'order_details'
  77
  78
  79 class Cart(models.Model):
  80     Pid = models.CharField(max_length=20)
  81     Pname = models.CharField(max_length=200)
  82     qty = models.IntegerField(default=None)
  83     cost = models.IntegerField(default=None)
  84     uid = models.CharField(max_length=200)
  85     total = models.IntegerField(default=None)
  86     category = models.CharField(max_length=200)
  87
  88     class Meta:
  89         db_table = 'cart'
  90
  91
  92
```

```
88     uid = models.CharField(max_length=200)
89     total = models.IntegerField(default=None)
90     category = models.CharField(max_length=200)
91
92     class Meta:
93         db_table = 'cart'
94
95
96 class Blockrequest(models.Model):
97     user_id = models.CharField(max_length=20)
98     Card_number = models.IntegerField(default=None)
99     cvv = models.IntegerField(default=None)
100     Issue = models.CharField(max_length=1000)
101
102     class Meta:
103         db_table = 'blockrequest'
104
105
106
```

```
FOLDERS
└─ OnlineTransactionFraudDetect
  └─ AppOnlineTransactionFraud
    ├── __pycache__
    ├── migrations
    ├── static
    └─ Templates
      ├── about.html
      ├── addproduct.html
      ├── AddtoCart.html
      ├── Admin_login.html
      ├── base.html
      ├── blockedUser.html
      ├── Dashboard.html
      ├── home.html
      ├── login.html
      ├── OrderSuccess.html
      ├── ProductDetails.html
      ├── register.html
      ├── unblockusers.html
      ├── User_login.html
      ├── verify.html
      ├── vieworders.html
      └── viewproduct.html
  ├── *_init_.py
  ├── admin.py
  ├── apps.py
  ├── forms.py
  ├── models.py
  ├── tests.py
  └── urls.py
  └── views.py

settings.py
33 INSTALLED_APPS = [
34     'django.contrib.admin',
35     'django.contrib.auth',
36     'django.contrib.contenttypes',
37     'django.contrib.sessions',
38     'django.contrib.messages',
39     'django.contrib.staticfiles',
40     'AppOnlineTransactionFraudDetection',
41 ]
42
43 MIDDLEWARE = [
44     'django.middleware.security.SecurityMiddleware',
45     'django.contrib.sessions.middleware.SessionMiddleware',
46     'django.middleware.common.CommonMiddleware',
47     'django.middleware.csrf.CsrfViewMiddleware',
48     'django.contrib.auth.middleware.AuthenticationMiddleware',
49     'django.contrib.messages.middleware.MessageMiddleware',
50     'django.middleware.clickjacking.XFrameOptionsMiddleware',
51 ]
52
53 ROOT_URLCONF = 'OnlineTransactionFraudDetection.urls'
54
55 TEMPLATES = [
56     {
57         'BACKEND': 'django.template.backends.django.DjangoTemplates',
58         'DIRS': [os.path.join(BASE_DIR, 'Templates')],
59         'APP_DIRS': True,
60         'OPTIONS': {
61             'context_processors': [
62                 'django.template.context_processors.debug',
63                 'django.template.context_processors.request',
64                 'django.contrib.auth.context_processors.auth',
65                 'django.contrib.messages.context_processors.messages',
66             ],
67         },
68     },
69 ]
70
71 WSGI_APPLICATION = 'OnlineTransactionFraudDetection.wsgi.application'
72
73 # Database
74 # https://docs.djangoproject.com/en/2.1/ref/settings/#databases
75
76 DATABASES = {
77     'default': {
78         'ENGINE': 'django.db.backends.mysql',
79         'NAME': 'OnlineTransactionFraudDetection',
80     },
81 }
```

```
FOLDERS
└─ OnlineTransactionFraudDetect
  └─ AppOnlineTransactionFraud
    ├── __pycache__
    ├── migrations
    ├── static
    └─ Templates
      ├── about.html
      ├── addproduct.html
      ├── AddtoCart.html
      ├── Admin_login.html
      ├── base.html
      ├── blockedUser.html
      ├── Dashboard.html
      ├── home.html
      ├── login.html
      ├── OrderSuccess.html
      ├── ProductDetails.html
      ├── register.html
      ├── unblockusers.html
      ├── User_login.html
      ├── verify.html
      ├── vieworders.html
      └── viewproduct.html
  ├── *_init_.py
  ├── admin.py
  ├── apps.py
  ├── forms.py
  ├── models.py
  ├── tests.py
  └── urls.py
  └── views.py

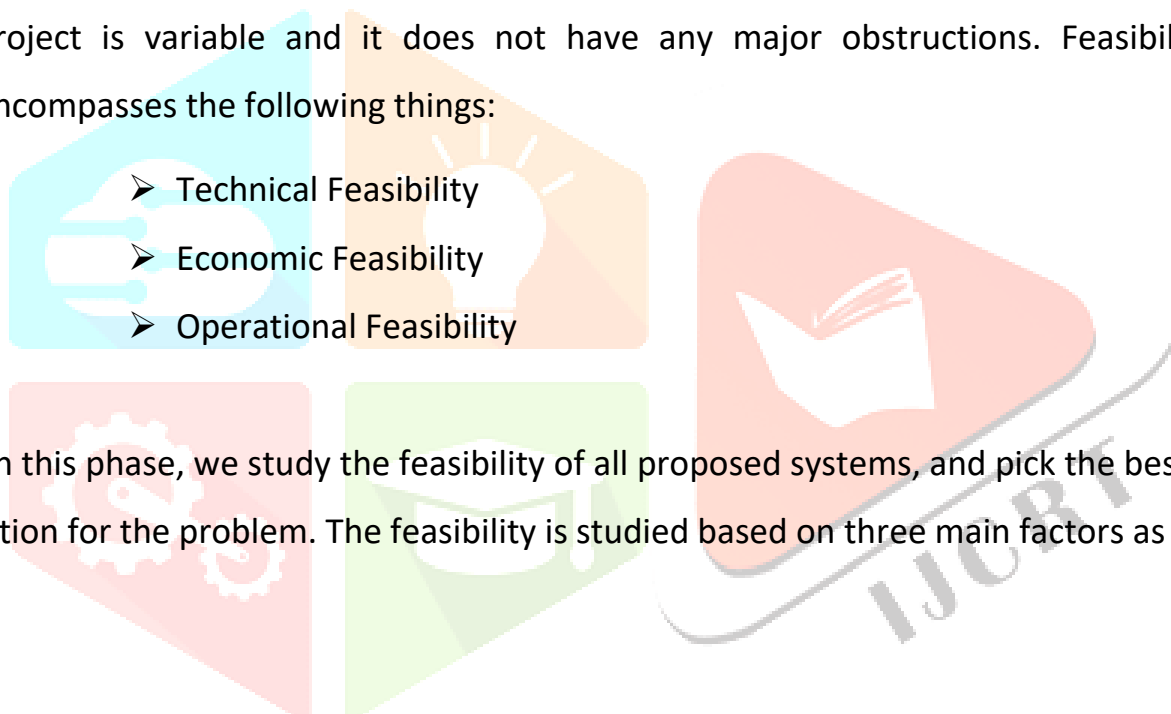
settings.py
82     'PASSWORD': '',
83     'HOST': '127.0.0.1',
84     'PORT': '3306',
85     }
86 }
87
88 DEFAULT_AUTO_FIELD = 'django.db.models.AutoField'
89
90 # Password Validation
91 # https://docs.djangoproject.com/en/2.1/ref/settings/#auth-password-validators
92
93 AUTH_PASSWORD_VALIDATORS = [
94     {
95         'NAME': 'django.contrib.auth.password_validation.UserAttributeSimilarityValidator',
96     },
97     {
98         'NAME': 'django.contrib.auth.password_validation.MinimumLengthValidator',
99     },
100    {
101        'NAME': 'django.contrib.auth.password_validation.CommonPasswordValidator',
102    },
103    {
104        'NAME': 'django.contrib.auth.password_validation.NumericPasswordValidator',
105    },
106 ]
107
108 # Internationalization
109 # https://docs.djangoproject.com/en/2.1/topics/i18n/
110
111 LANGUAGE_CODE = 'en-us'
112
113 TIME_ZONE = 'UTC'
114
115 USE_I18N = True
116
117 USE_L10N = True
118
119 USE_TZ = True
120
121 # Static files (CSS, JavaScript, Images)
122 # https://docs.djangoproject.com/en/2.1/howto/static-files/
123
124 STATIC_URL = '/static/'
125
126 STATICFILES_DIRS = [
127     os.path.join(BASE_DIR, 'AppOnlineTransactionFraudDetection/static')
128 ]
129
```

```
128 STATIC_URL = '/static/'
129 STATICFILES_DIRS = [
130     os.path.join(BASE_DIR, 'AppOnlineTransactionFraudDetection/static')
131 ]
132
133 STATIC_ROOT = (os.path.join(BASE_DIR, 'assets'))
134
135 MEDIA_URL = '/media/'
136 MEDIA_ROOT = os.path.join(BASE_DIR, 'media')
```

FEASIBILITY REPORT

Feasibility Study is a high level capsule version of the entire process intended to answer a number of questions like: What is the problem? Is there any feasible solution to the given problem? Is the problem even worth solving? Feasibility study is conducted once the problem clearly understood. Feasibility study is necessary to determine that the proposed system is Feasible by considering the technical, Operational, and Economical factors. By having a detailed feasibility study the management will have a clear-cut view of the proposed system.

The following feasibilities are considered for the project in order to ensure that the project is variable and it does not have any major obstructions. Feasibility study encompasses the following things:



In this phase, we study the feasibility of all proposed systems, and pick the best feasible solution for the problem. The feasibility is studied based on three main factors as follows.

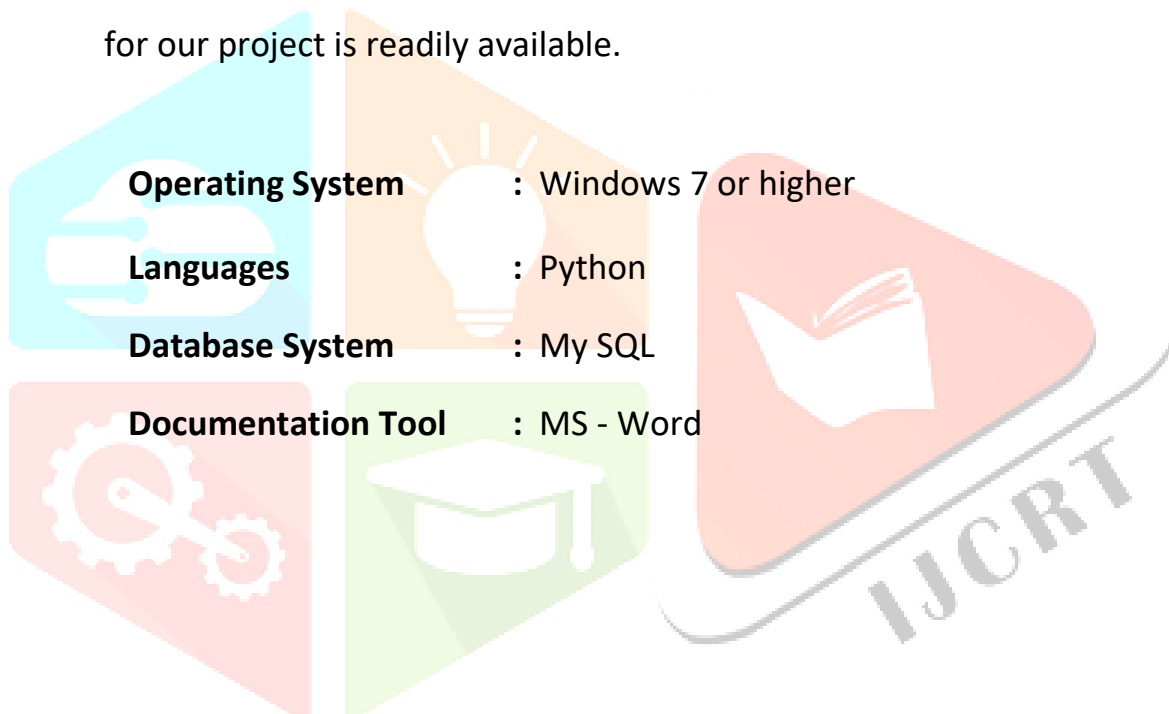
❖ Technical Feasibility

In this step, we verify whether the proposed systems are technically feasible or not. i.e., all the technologies required to develop the system are available readily or not.

Technical Feasibility determines whether the organization has the technology and skills necessary to carry out the project and how this should be obtained. The system can be feasible because of the following grounds:

- All necessary technology exists to develop the system.
- This system is too flexible and it can be expanded further.
- This system can give guarantees of accuracy, ease of use, reliability and the data security.
- This system can give instant response to inquire.

Our project is technically feasible because, all the technology needed for our project is readily available.



❖ Economic Feasibility

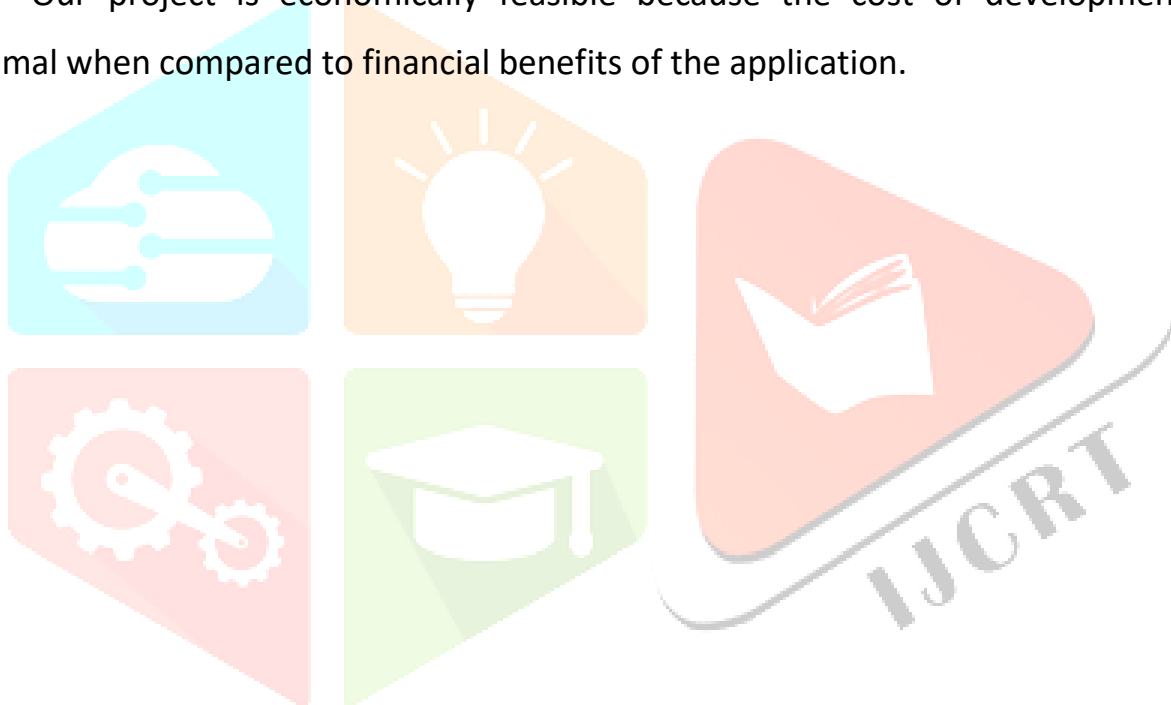
Economically, this project is completely feasible because it requires no extra financial investment and with respect to time, it's completely possible to complete this project in 6 months.

In this step, we verify which proposal is more economical. We compare the financial benefits of the new system with the investment. The new system is economically feasible only when the financial benefits are more than the

investments and expenditure. Economic Feasibility determines whether the project goal can be within the resource limits allocated to it or not. It must determine whether it is worthwhile to process with the entire project or whether the benefits obtained from the new system are not worth the costs. Financial benefits must be equal or exceed the costs. In this issue, we should consider:

- The cost to conduct a full system investigation.
- The cost of h/w and s/w for the class of application being considered.
- The development tool.
- The cost of maintenance etc...

Our project is economically feasible because the cost of development is very minimal when compared to financial benefits of the application.



❖ Operational Feasibility

In this step, we verify different operational factors of the proposed systems like man-power, time etc., whichever solution uses less operational resources, is the best operationally feasible solution. The solution should also be operationally possible to implement. Operational Feasibility determines if the proposed system satisfied user objectives could be fitted into the current system operation.

- The methods of processing and presentation are completely accepted by the clients since they can meet all user requirements.
- The clients have been involved in the planning and development of the system.
- The proposed system will not cause any problem under any circumstances.

Our project is operationally feasible because the time requirements and personnel requirements are satisfied. We are a team of four members and we worked on this project for three working months.

TESTING

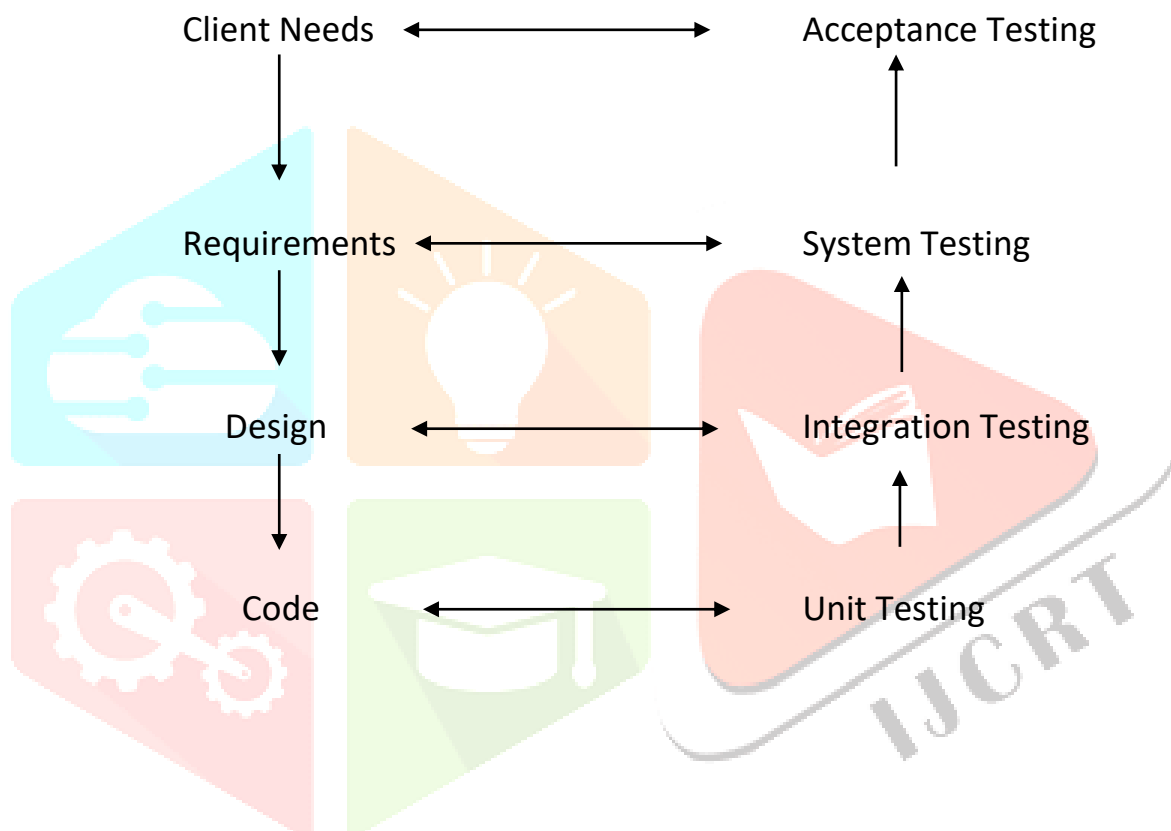
As the project is on bit large scale, we always need testing to make it successful. If each component work properly in all respect and gives desired output for all kind of inputs then project is said to be successful. So the conclusion is-to make the project successful, it needs to be tested.

The testing done here was System Testing checking whether the user requirements were satisfied. The code for the new system has been written completely using python as the coding language, Django as the interface for front-end designing. The new system has been tested well with the help of the users and all the applications have been verified from every nook and corner of the user.

Although some applications were found to be erroneous these applications have been corrected before being implemented. The flow of the forms has been found to be very much in accordance with the actual flow of data.

Levels of Testing

In order to uncover the errors present in different phases we have the concept of levels of testing. The basic levels of testing are:



A series of testing is done for the proposed system before the system is ready for the user acceptance testing.

The steps involved in Testing are:

✓ **Unit Testing**

Unit testing focuses verification efforts on the smallest unit of the software design, the module. This is also known as “Module Testing”. The modules are tested separately. This testing carried out during programming stage itself. In this testing each module is found to be working satisfactorily as regards to the expected output from the module.

✓ **Integration Testing**

Data can be grossed across an interface; one module can have adverse efforts on another. Integration testing is systematic testing for construction the program structure while at the same time conducting tests to uncover errors associated with in the interface. The objective is to take unit tested modules and build a program structure. All the modules are combined and tested as a whole. Here correction is difficult because the isolation of cause is complicate by the vast expense of the entire program. Thus in the integration testing stop, all the errors uncovered are corrected for the text testing steps.

✓ **System testing**

System testing is the stage of implementation that is aimed at ensuring that the system works accurately and efficiently for live operation commences. Testing is vital to

the success of the system. System testing makes a logical assumption that if all the parts of the system are correct, then goal will be successfully achieved.

✓ Validation Testing

At the conclusion of integration testing software is completely assembled as a package, interfacing errors have been uncovered and corrected and a final series of software tests begins, validation test begins. Validation test can be defined in many ways. But the simple definition is that validation succeeds when the software function in a manner that can reasonably expected by the customer. After validation test has been conducted one of two possible conditions exists.

One is the function or performance characteristics confirm to specifications and are accepted and the other is deviation from specification is uncovered and a deficiency list is created. Proposed system under consideration has been tested by using validation testing and found to be working satisfactorily.

✓ Output Testing

After performing validation testing, the next step is output testing of the proposed system since no system could be useful if it does not produce the required output in the specified format. Asking the users about the format required by them tests the outputs generated by the system under consideration. Here the output format is considered in two ways, one is on the screen and other is the printed format. The output format on the screen is found to be correct as the format was designed in the system designed phase according to the user needs.

For the hard copy also, the output comes as the specified requirements by the users. Hence output testing does not result any corrections in the system.

✓ User Acceptance Testing

User acceptance of a system is the key factor of the success of any system. The system under study is tested for the user acceptance by constantly keeping in touch with the prospective system users at the time of developing and making changes wherever required.

ADVANTAGES OF PROJECT

- The system stores previous transaction patterns for each user.
- Based upon the user spending ability and even country, it calculates user's characteristics.
- The system is more secured with OTP (One Time Password) implementation.
- IP address tracking at every transaction.
- Security questions for payment limit crossed.
- More than 20 -30 % deviation of user's transaction (spending history and operating country) is considered as an invalid attempt and system takes action.

System is:

1) Load Balancing:

Since the system will be available only the admin logs in the amount of load on server will be limited to time period of admin access.

2) Easy Accessibility:

Records can be easily accessed and store and other information respectively.

3) User Friendly:

The system will be giving a very user-friendly approach for all user.

4) Efficient and reliable:

Maintaining the all secured and database on the server which will be accessible according the user requirement without any maintenance cost will be a very efficient as compared to storing all the customer data on the spreadsheet or in physically in the record books.

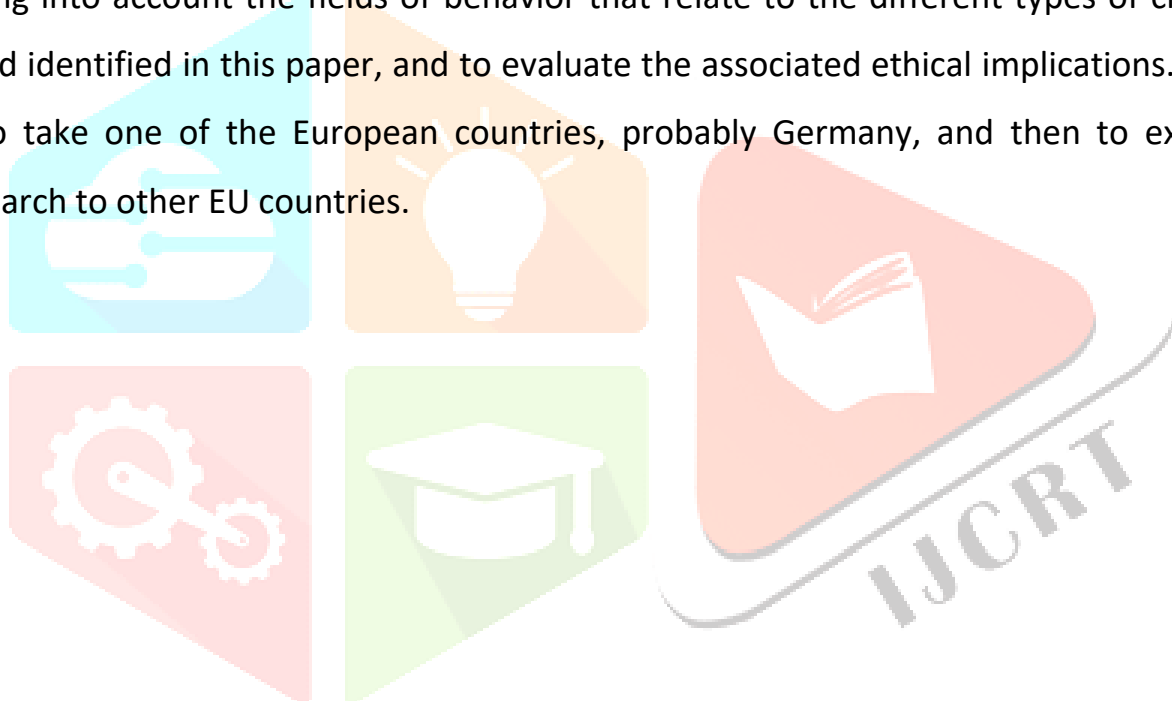
5) Easy maintenance:

Online Transaction Fraud Detection & Backlogging on E-Commerce is design as easy way. So, maintenance is also easy.

CONCLUSION

Clearly, credit card fraud is an act of criminal dishonesty. This article has reviewed recent findings in the credit card field. This paper has identified the different types of fraud, such as bankruptcy fraud, counterfeit fraud, theft fraud, application fraud and

behavioral fraud, and discussed measures to detect them. Such measures have included pair-wise matching, decision trees, clustering techniques, neural networks, and genetic algorithms. From an ethical perspective, it can be argued that banks and credit card companies should attempt to detect all fraudulent cases. Yet, the unprofessional fraudster is unlikely to operate on the scale of the professional fraudster and so the costs to the bank of their detection may be uneconomic. The bank would then be faced with an ethical dilemma. Should they try to detect such fraudulent cases or should they act in shareholder interests and avoid uneconomic costs? As the next step in this research program, the focus will be upon the implementation of a ‘suspicious’ scorecard on a real data-set and its evaluation. The main tasks will be to build scoring models to predict fraudulent behavior, taking into account the fields of behavior that relate to the different types of credit card fraud identified in this paper, and to evaluate the associated ethical implications. The plan is to take one of the European countries, probably Germany, and then to extend the research to other EU countries.



BIBLIOGRAPHY

► Websites

- ✓ en.wikipedia.org
- ✓ Microsoft Developer Network (MSDN): <http://msdn2.microsoft.com/en-us/default.aspx>: This is a valuable online resource, and is a must for any developer using Microsoft tools.
- ✓ <http://www.asp.net/>: This is the official Microsoft ASP.NET web site. It has a lot of: tutorials, training videos, and sample projects.
- ✓ <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?tp=&arnumber=5159014&queryText%3DCredit+Card+Fraud+Detection>
- ✓ <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?tp=&arnumber=323314&queryText%3DCredit+Card+Fraud+Detection>
- ✓ <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?tp=&arnumber=5762457&queryText%3DCredit+Card+Fraud+Detection>