1.1 ABOUT THE PROJECT

K-Out (Kerala Outlet) is a powerful connectivity solution that enables an organization to manage its business transaction spread all across the Kerala on the internet, providing real time access and updates to the centrally managed enterprise database while using minimal communication and IT resources.

The system has 4 main modules admin, staff, deliver and consumer. The most striking feature of K-Out (Kerala Outlet) is that it provides a subsidy rate for some items. The subsidy rates are applicable for only those customers who have a ration card. All the others can purchase items in normal rate (i.e., without ration card). The subsidy items are available once in a month.

The K-Out (Kerala Outlet) is purely concentrated for public. Now a day we saw a large queue in front of supply co and ration shop, the billing and payment sections of are also time-consuming process. So, it is essential to overcome this problem. In K-Out (Kerala Outlet), any customer can buy items via online and collect the items from supply-co with in specified date. This web application displays both subsidy rates normal rates depend on the type of ration card of the ration card holder. If the Consumer is making online purchase with ration card only subsidy rate of each item is displayed. For normal Consumers each item is priced in the normal rate. Here there is no time delay for billing section. If the allotted period is expired the booking is cancelled and amount will be refunded.

The system provides a specified code and time period for delivery. An e-wallet facility is also provided. The most striking feature of this e-wallet is that the customer can acquire a discount and hence it does not demand to reveal our banking details through internet. The payment is by e-wallet and Cash on deliver (COD).

MODULE DESCRIPTION

The main modules of these projects are:

- **Admin Module**
- **Staff Module**
- **Deliver Module**
- **Consumer Module**

1. ADMIN MODULE

Staff Management

Admin can approve or reject the registered Staff by checking the registered Staff is genuine or not, also the admin can approve and reject the delivery agent.

Product Management

Admin can allocate the product amount and quantity upon the basis of the ration card type, and admin can update the already allocated product.

Consumer Management

Admin can view the registered consumer and can remove the consumer who reported by the staff.

2. STAFF MODULE

Manage Customers

Staff can see the registered consumer and approve them to buy items from product that the Staff added. Staff also can report the wrong details of consumer as per the application given by the consumer physically.

Manage Purchase

Staff can view online purchases made by customers. Staff can also view and update the status of packing/deliver.

View Products

Staff can view product and checkout the status of a product whether it is out of stock or not, also Staff can update the status of the product details and others.

Manage Deliver Agent

Staff can appoint the deliver agent who is approved by the admin.

3. DELIVER MODULE

Order Place

Deliver agent place the order and they are ready to take new orders that specified by the Staff that is approved by the deliver agent.

View Order

Deliver agent can see their new orders that given by the staff.

4. CONSUMER MODULE

New Purchase

Consumers can make online purchases from the profile page. The products of the Supply Co. and Ration Shops, to which the consumer is registered, will be displayed. The consumer can view and make purchase the product.

View Purchases

The consumer can view his or her purchase status. That is, the consumer can view whether the product has been purchased/deliver or not.

Notifications

Packing status, payment status etc. will be notified to the consumers.

Feedback

Consumer can give feedback about the shop service or deliver agent service.

1.2 ABOUT THE ORGANIZATION

We did our project at softzane solutions located at Ayoor, in Kollam District. The company was started in 2016 with experienced IT professionals, with the mission to create innovative software solutions which aids its ability to support the business. Softzane solutions specialize in the development of web applications development and smart phone applications development. The Advanced innovation is the tool that will leverage to build available and cost-effective solutions. They have effective software solutions providing services team and an innovative management to organize to client solutions. Innovation depends not only on the technology but also we use it. They mainly focus on software development and training.

Software development

We provide comprehensive smart phone web solutions using our resources and talented 'developers'. They organized to deliver premium quality services to clients ranging from large organizations to smaller groups and individual from all over the sphere. They have the rights expertise and experience in mobile application development.

Software Training

The Softzane solutions global academic program is uniquely focused on education and ready to work with students to support teaching, learning and research goals. Their value added educational services includes: php, java, j2ee.

The main services of Softzane Solutions are:

- Web Applications - Java, PHP, Android,.Net.
- Android Applications - Development and Academic Project.
- Website Designing-Java, PHP, Android,.Net.
- Embedded System-Academic Projects.
- Softzane Education-Internship and Courses in PHP, Android, Java, .Net, Embedded, CCNA.

2.1 EXISTING SYSTEM

In earlier day the citizens have to buy goods from Supplyco and Public Distribution System(PDS) by manually. They buy products directly from the shop. This is time consuming and they have to wait too much time in the shop to get their products. Some of them are far from shop or old people who suffer from lot of disabilities or maybe they have no vehicles so they have to walk or call taxi this will make them more uncomfortable and money waste.

2.1.1 LIMITATION OF EXISTING SYSTEM

1) More man power is needed.
2) No algorithms are implemented.
3) Time consuming procedure.
4) User can’t get direct access to the system.
5) No system based cross matching.
6) No proper communications are implemented.

2.2 PROPOSED SYSTEM

Now the civil supplies are working on a manual way that a needed person wanted to come to the store and buy the needed things. In some cases some products can't available at that particular time. With this system civil supply stores can be connected and needed products can be taken.

Now the public needed to come straight to the store and buy products by standing on long queues. This can be managed by implementing delivery system. This offers so many job opportunities along with the service given. It makes the way of service different.

Many people are not satisfied with the current system. It mainly because of the long queues and the lack of availability of products, and also the senior citizens can't go to stores directly because of the health issues caused to them. These all can be resolved through this system.

In K-Out (Kerala Outlet) people can buy products online and collect the item from the specified date. This web application displays both subsidy rates normal rates depends on the type of Consumer. If the Consumer is making online purchase with ration card only subsidy product rate of each item is displayed .For normal consumers each item is priced in the normal rate. Here there is no time delay for billing session. If the allotted period is expired the booking is cancelled and amount will be refunded.

The system provides a specified code and time period for delivery. A Cash on delivery (COD) or online payment facility is also providing the most striking feature of this online payment is that the customer can acquire a discount and hence it does not demand to reveal our banking details through internet. The consumer can edit there profile for changing their address. System provides10 super coins while they buy product above 500 rupees. After completing 150 super coins then we provide free delivery.

System provides IVRS (Interactive Voice Response System)it is an automated telephone system. It contains pre-recorded messages or text to speech technology with dual tone multi frequency. IVR interface provide customers to access information without a live agent. This will help the old-ages to buy products without login into website who are not more aware about how to use the internet and smart phone, computer, etc.,. This will help them more.

System provides SMS (Short Message Service) it is a standard messaging service. At the time of buying products system provides an SMS to the customer. At the time of delivery the delivery boy can check the SMS for identifying the customer.

2.2.1 ADVANTAGES OF PROPOSED SYSTEM

1) Time saving.
2) No need to go to the shop.

3) Helpful for all citizens to get their commodities at home.
4) Door to door delivery.
5) Cash on delivery.
6) IVR (Interactive Voice Response)System.
7) WhatsApp SMS verification.

2.3 FEASIBILITY STUDY

The purpose of the feasibility study is to identify various candidate systems and evaluates they are feasible by considering technical, economical, and operational feasibility and to recommend to best candidate system. An initial investigation culminates in a proposal that determine whether an ultimate system is feasible. When a proposed system is made and approved it initiates a feasibility study.

The feasibility of such a program is listed in a simulated environment. Once all features are working property in a simulated environment, we can implement in a real platform. Depending on the results of the initial investigation the survey is now expanded to a more detailed feasibility study. “FEASIBILITY STUDY” is a test of system proposal according to its workability, impact of the organization, ability to meet needs and effective use of their sources.

Following primary areas of interest are to be considered. Investigation and generating ideas about a new system does this. During product engineering, we considered following types of feasibility.

Operational Feasibility

Proposed projects are beneficial only if they can be turned into information system that will meet the operating requirements of the organization. This test of feasibility asks if the system will work when it is developed and installed. This project satisfies all the operational conditions. The project is found to work well on an installation all types of users can operate the system without any difficulty. User interfaces are designed in such a way that even ordinary users without having much knowledge in computer technology can easily operate the system. The access time of data is considerably low and the operation is less time consuming. Traders can by their product from the organization on online.

Technical Feasibility

This assessment focuses on the technical resources available to the organization. It helps organizations determine whether the technical resources meet capacity and whether the technical team is capable of converting the ideas into working systems.

- Development risk
- Resource availability
- Technology

Economic Feasibility

A cost evaluation is weighed against the ultimate or benefit derived from the developed system or product. Economic justification is generally the “bottom-line” consideration that includes cost benefit analysis, long term cooperates income strategies, impact on other profit centers or products, cost of resources needed for development and potential market growth. When compared with the advantage obtained from implementing the system its cost is affordable. Also, the system is designed to meet the modifications required in the future. So, most of the required modifications can be done without much re- work proposed system was developed with the available resources.

Behavioral Feasibility

It understands the advantages and efficiency of the proposed system and decided to develop a new system.

Software Feasibility

In our project we planned to develop each module with the languages PHP, MySQL and XAMPP. The presentation layer includes XHTML, PHP and smarty.

Hardware Feasibility

The software can be developed with resources already existing. Here the consideration that the existing hardware resources support the technologies that is to be used by the new system. No hardware was newly brought for this project.

2.4 PROCESSING ENVIRONMENT

The given system can be developed under optimal expenses with the available hardware and software. Besides it is good economic to invest in such a kind of software from the users point of view as the benefits weigh the costs. This project is developed with the help of PHP and MySQL. So this combination results a high efficiency product. The project is legally feasible also. The main processing environment is hardware interfaces and software interfaces.

HARDWARE INTERFACES

The hardware specification is as follows:

Processor	: Intel Pentium or Higher
Memory	: 250GBRAM
Hard Disk	: 80 GB or higher
Keyboard	: Standard 102/105 keys
Mouse	: Optical Mouse

SOFTWARE INTERFACES

The software requirements for the system are selected considering the factors such as platform independency, robustness, security, etc. When we use a technology, we should use emerging technology. Because when we use any old technology, then it will affect the market and our system.

Operating Systems	: Windows10
Web server	: Apache
Database	: MySQL
Frontend	:PHP, Dream weaver
Browser	: Google Chrome, Mozilla Firefox or any others

2.5 SOFTWARE DESCRIPTION Windows7/10

An Operating System may be viewed as an organized collection of software extensions of hardware, consisting of control routines for operating a computer and for providing an environment for execution of programs. Programs rely on facilities provided by the operating system to gain access to computer-system resources such as files and input/output devices. That is the operating system acts as interface between users and the hardware of a computer system.

Windows 10 is an operating system that uses pre-emptive, multitasking and multi-threading to perform several actions at the same time. The advantage are powerful mobile computing features using the Microsoft Network, a Recycle Bin for storing the deleted files which can be cleared later and the inbox which can be used for communications like FAX, E-MAIL etc.

Windows includes a suite of programs designed to optimize our computer's efficiency, especially when used together. The key features are:

2.5.1 The Operating System automatically detects network cards and user has to just specify the name of the network and the computer can be logged off the specified network.

2.5.2 The computer can be connected to the internet if the user has access to it and the operating system needed to log on and run applications on the internet.

XAMPP Server

The XAMPP is a free and open source cross-platform web server package, consisting mainly of the Apache HTTP Server, MySQL database, and interpreters for scripts written in the PHP and Perl programming languages. XAMPP is an easy to install Apache distribution containing MySQL, PHP and Perl. XAMPP is really very easy to install and to use just download, extract and start. It is used as a development to allow website designers and programmers to test their work on their own computers without any access to the Internet.

Many important security features are disabled by default. XAMPP is sometimes used to actually serve web pages on the World Wide Web. A special tool is provided to password-protect the most important parts of the package.

XAMPP also provides support for creating and manipulating databases in MySQL and SQL among others. Once XAMPP is installed we can treat our local host like a remote host by connecting using an FTP client. Using a program like FileZilla has many advantages when installing a content management system (CMS) like Joomla. We can also connect to local host via FTP with our HTMLeditor.

Overview of PHP

Rasmus Lerdorf —Software Engineer, Apache team member, and international man of mystery is the creator and original driving force behind PHP.PHP is the Web development language written by and for Web developers. PHP stands for Hypertext Preprocessor. The product was originally named Personal Home Page Tools. But as it expanded in scope, a new and more appropriate name was selected by community vote. PHP is currently in its fifth major rewrite, called PHP5 or just plain PHP is a server-side scripting language, which can be embedded in HTML or used as a standalone binary. Proprietary products in this niche are Microsoft’s Active Server Pages, Macromedia’s ColdFusion, and Sun’s Java Server Pages. Some tech journalists used to call PHP “the open source ASP” because its functionality is similar to that of the Microsoft product—although this formulation was misleading, as PHP ASP was developed before. Over the past few years however, PHP and server-side Java have gained momentum, while ASP has lost mindshare, so this comparison no longer seems appropriate.Server-sidescriptingisacollectionofsuper-HTMLtagsorsmallprogramsthatrun inside our Web pages except on the server side, before they get sent to the browser. For example, we can use PHP to add common headers and footers to all the pages on a site or to store form-submitted data in a database.

PHP is an official module of Apache HTTP Server, the market-leading free Web server that runs about 67 percent of the World Wide Web. This means that the PHP scripting engine can be built into the Web server itself, leading to faster processing, more efficient memory allocation, and greatly simplified maintenance. Like Apache Server, PHP is fully cross-platform, meaning it runs native on several flavors of UNIX, as well as on Windows and now on Mac OS X.

All projects under the aegis of the Apache Software Foundation including PHP are open source software.

PHP (recursive acronym for PHP: Hypertext Preprocessor) is a widely used open source general-purpose scripting language that is especially suited for web development and can be embedded into HTML. Instead of lots of commands to output HTML (as seen in C or Perl), PHP pages contain HTML with embedded code that does "something" (in this case, output "Hi, I'm a PHP script!"). The PHP code is enclosed in special start and end processing instructions

What distinguishes PHP from something like client-side JavaScript is that the code is executed on the server, generating HTML which is then sent to the client. The client would receive the results of running that script, but would not know what the underlying code was. We can even configure our web server to process all our HTML files with PHP, and then there's really no way that users can tell what we have up our sleeve.

The best things in using PHP are that it is extremely simple for a newcomer, but offers many advanced features for a professional programmer. Don't be afraid reading the long list of PHP's features. We can jump in, in a short time, and start writing simple scripts in a few hours.

PHP is mainly focused on server-side scripting, so we can do anything that any other CGI program can do, such as collect form data, generate dynamic page content, or send and receive cookies. But PHP can do much more.

There are three main areas where PHP scripts are used:

Server-side scripting: This is the most traditional and main target field for PHP. We need three things to make this work. The PHP parser (CGI or server module), a web server and a web browser. We need to run the web server, with a connected PHP installation. We can access the PHP program output with a web browser, viewing the PHP page through the server. All these can run on our home machine if we are just experimenting with PHP programming.

Command line scripting: We can make a PHP script to run it without any server or browser. We only need the PHP parser to use it this way. This type of usage is ideal for scripts regularly executed using crone (on *nix or Linux) or Task Scheduler (on Windows). These scripts can also be used for simple text processing tasks.

Desktop Writing applications: PHP is probably not the very best language to create a desktop application with a graphical user interface, but if we know PHP very well, and would like to use some advanced PHP features in our client-side applications we can also use PHP-GTK to write such programs. We also have the ability to write cross-platform applications this way

PHP can be used on all major operating systems, including Linux, many UNIX variants (including HP-UX, Solaris and Open BSD), Microsoft Windows, Mac OS X, RISC OS, and probably others. PHP has also support for most of the web servers today. This includes Apache, Microsoft Internet Information Server, Personal Web Server, Netscape and planet servers, Oreille Website Pro server, Caudium, Xitami, OmniHTTPd, and many others. For the majority of the servers, PHP has a module, for the others supporting the CGI standard, PHP can work as a CGI processor. So with PHP, we have the freedom of choosing an operating system and a web server. Furthermore, we also have the choice of using procedural programming or object-oriented programming, or a mixture of them.

Although not every standard OOP feature is implemented in PHP 4, many code libraries and large applications (including the PEAR library) are written only using OOP code. PHP 5 fixes the OOP related weaknesses of PHP 4, and introduces a complete object model.

With PHP we are not limited to output HTML. PHP's ability includes outputting images, PDF files and even Flash movies (using libswf and Ming) generated on the fly. We can also output easily any text, such as XHTML and any other XML file. PHP can auto generate these files, and save them in the file system, instead of printing it out, forming a server-side cache for our dynamic content. One of the strongest and most significant features in PHP is its support for a wide range of databases. Writing a database enabled web page is incredibly simple.

PHP doesn't cost anything. We can use it for commercial and/or non-commercial use all we want. Any problem we encountered in our coding can be answered swiftly and easily with a little research. There is no interest in a particular server product or operating system. We are free to make choices that suit our needs or those of our clients.

Advantages of PHP

1) Cost is low
2) PHP is an open source software
3) PHP is easy to learn
4) PHP is embedded within HTML

The HTML- embedding of PHP has many helpful consequences:

1) PHP can quickly be added to code produced by WYSIWYG editors.
2) PHP lends itself to a division of labor between designers and scripters.
3) Every line of HTML does not need to be rewritten in a programming language.
4) PHP can reduce labor costs and increase efficiency due to its shallow learning curve and ease of use.

5) PHP has Cross-platform compatibility.
6) PHP is not tag-based.
7) PHP is stable.
8) PHP is much faster for almost every use than CGI scripts.
9) PHP makes it easy to communicate with other programs and protocols.
10) PHP is fast becoming one of the most popular choices for so called two-tier development.
11) PHP is developed and supported in a collaborative fashion by a worldwide community of users.

DATABASE

A database is a separate application that stores a collection of data. Each database has one or more distinct APIs for creating, accessing, managing, searching, and replicating the data it holds. Other kinds of data stores can be used, such as files on the file system or large hash tables in memory. A database system must provide following features:

1) A variety of user interfaces.
2) Physical data independence.
3) Logical data independence.
4) Query optimization.
5) Data integrity.
6) Concurrency control.
7) Backup and recovery.
8) Security and authorization.

MySQL

MySQL is a fast, easy-to-use RDBMS used being used for many small and big businesses. MySQL is developed, marketed, and supported by MySQL, which is a Swedish company. MySQL is an open source, SQL Relational Database Management System (RDBMS) that is free for many uses (more detail on that later). Early in its history, MySQL occasionally faced opposition due to its lack of support for some core SQL constructs such as sub selects and foreign keys. Ultimately, however, MySQL found a broad, enthusiastic user base for its liberal licensing terms, perky performance, and ease of use. Its acceptance was aided in part by the wide variety of other technologies such as PHP, Java, Perl, Python, and the like that have encouraged its use through stable, well-documented modules and extensions. MySQL has not failed to reward the loyalty of these users with the addition of both sub selects and foreign keys.

MySQL is becoming so popular because of many good reasons.

- MySQL is released under an open-source license. So, we have nothing to pay to use it.
- MySQL is a very powerful program in its own right. It handles a large subset of the functionality of the most expensive and powerful database packages.
- MySQL uses a standard form of the well-known SQL data language.
- MySQL works on many operating systems and with many languages including PHP, PERL, C, C++, JAVA, etc.
- MySQL works very quickly and works well even with large datasets.
- MySQL is very friendly to PHP, the most appreciated language for web development.
- MySQL supports large databases, up to 50 million rows or more in a table. The default file size limit for a table is 4GB, but we can increase this (if our operating system can handle it) to a theoretical limit of 8 million terabytes (TB).
- MySQL is customizable. The open-source GPL license allows programmers to modify the MySQL software to fit their own specific environments.

SMARTY

Smarty is a web template system written in PHP. Smarty is primarily promoted as a tool for separation of concerns. Smarty is intended to simplify compartmentalization, allowing the presentation of a web page to change separately from the back end. Ideally, this eases the costs and efforts associated with software maintenance.

Smarty is a template engine for PHP. More specifically, it facilitates a manageable way to separate application logic and content from its presentation. This is best described in a situation where the application programmer and the template designer play different roles, or in most cases is not the same person.

One design goal of Smarty is the separation of business logic and presentation logic. This means templates can certainly contain logic under the condition that it is for presentation only. Things such as including other templates, altering table row colors, uppercasing a variable, looping over an array of data and displaying it, etc. are all examples of presentation logic. This does not mean that Smarty forces a separation of business and presentation logic. Smarty has no knowledge of which is which, so placing business logic in the template is our own logic. Also, if we have no logic in our templates, we certainly can do so by boiling the content down to text and variables only.

Advantages

- It is extremely fast.
- It is efficient since the PHP parser does the dirty work.
- No template parsing overhead, only compiles once.
- It is smart about recompiling only the template files that have changed.
- We can make custom functions and custom variable modifiers, so the template language is extremely extensible.
- Configurable template delimiter tag syntax, so we can use {}, {{}}, <!--{} -->, etc.

- The if/else if/else/end if constructs are passed to the PHP parser, so the {if ...} expression syntax can be as simple or as complex as like.
- Unlimited nesting of sections, ifs, etc. Allowed.
- It is possible to embed PHP code right in our template files; although this may not be needed (nor recommended) since the engine is so customizable.
- Built-in caching support.
- Arbitrary template sources Custom cache handling functions.
- Plug in architecture.

CSS

- CSS stands for Cascading Style Sheets.
- Styles define how to display HTML elements.
- Styles were added to HTML 4.0 to solve a problem.
- External Style Sheets can save a lot of work.

External Style Sheets are stored in CSS files HTML was never intended to contain tags for formatting a document.HTML was intended to define the content of a document, like:

<h1>this is a heading</h1>

<p>this is a paragraph</p>

When tags like <font>, and color attributes were added to the HTML 3.2 specification, it started a nightmare for web developers. Development of large web sites, where fonts and color information were added to every single page, became a long and expensive process. To solve this problem, the World Wide Web Consortium (W3C) created CSS.

SUBLIME TEXT

Sublime text is a shareware cross platform source code editor with a python application programming interface (API).It natively support many programming languages and markup languages, and function can be added by uses with plugging, typically community-built and maintained under pre-software license.

Features:

Features of sublime text:

- "Go to anything", quick navigation to files, symbols or lines.
- "Command palette" uses adaptive matching for quick keyboard invocation of arbiter's commands.
- Project specific preference.
- Expensive customizability via JSON setting files, including project- specific and platform-specific settings.
- Cross – platform (windows, MacOS and Linux) and supportive plugins cross- platform.
- Data comparison.
- Drag and drop.
- Spell checker (requires a spell).

3. SYSTEM DESIGN

The most creative and challenging phase of the system life cycle is the system design. The term design describes a final system and the process by which it is developed. It refers to the technical specifications that will be applied in implementing the candidate system. It also includes the construction of programs and program testing.

The first step in the system design is to determine how the output is to be produced and in what format. Samples of the inputs and the output are also presented. In the second step, input data and master files are to be designed to meet requirement of the proposed output. The processing phases are handled through program construction and testing, including a list of the programs needed to meet the system's objectives and complete documentation.

Finally, details related to justification of the system and an estimate of the impact of the candidate system on the user and organization are documented and implemented and evaluated by management as a step towards implementation. The final report prior to the implementation phase includes procedure flow chart, record layouts, and a workable plan for implementing the candidate system.

System design has two phases:

- Logical design
- Physical design

In the logical design, the designer produces a specification of the major features of the system which meets the objectives. The delivered product of logical design includes current requirements of the following system components:

- Input design
- Output design
- Database design

Physical design takes this logical design blue print and produces the program software, files and a working system...

3.1 INPUT DESIGN

The input design is the process of converting the user-oriented inputs in to the computer-based format. The goal of designing input data is to make automation as easy and free from errors as possible. The input design requirements such as user friendliness, consistent format and interactive dialogue for giving the right message and help for the user at right time are also considered for the development of the project. The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps that are necessary to put transaction data into a usable form for processing data entry.

Inaccurate input data is the most common cause of error in processing data. Errors entered by the data entry operators can be controlled by the input design. The arrangement of messages as well as placement of data, headings and titles on display screens or source document is also a part of input design. The design of input also includes specifying the means by which end user and system operators direct the system what action to take.

In this project all the necessary text boxes are validated. The input forms are designed in notepad++ using GUI Controls like textboxes, list box, checkbox, frames, radio button and other controls. If any non-empty fields are not filled, it will display error message and will wait until user types the necessary and correct input. The combo boxes are used to reduce the user inputs. The user can select one of the items from combo boxes. Initially to access the services of this software, the user has to log on with a login name and password which are validated. Once logged on, he can access the various services, navigate to different profiles.

3.2 OUTPUT DESIGN

Output generally refers to the results and information that are generated by the system. The output design is specified on layout forms, sheets that describe the location characteristics and format of the column headings and pagination. In my project, the output forms are designed in PHP. Each form has a heading or caption which specifies what services is been given to the user making the software user-friendly. All requests given by the user is sent to the server which is validated and accordingly the corresponding pages are given to the client users. All information is stored in the database and when the user logs on and requests for a service, the corresponding page is fetched from the server after validation and is rendered.

3.3 DATABASE DESIGN

A database is a collection of interrelated data stored with minimum redundancy to serve users more quickly and efficiently. James Martin defines database as "**A collection of data designed to be used by different programs**". The general objective of a database is to make information access easy, quick, inexpensive, integrated, and shared by different applications and users. Database design is an important yet sometimes overlooked part of the application development lifecycle. An accurate and up-to-date data model can serve as an important reference tool for Database Administrators, developers, and other members of joint application development team. A good database design does the following:

- Provide minimum search time when locating specific records.
- Stored data in the most efficient manner possible to keep the database from growing too large
- Make data update as easy as possible.
- Is flexible enough to allow inclusion of new functions required of the programs

A database is an organized mechanism that has the capability of storing information through which a user can retrieve stored information in an effective and efficient manner. The data is the purpose of any database and must be protected. The database design is a two-level process.

In the first step user requirements are gathered together and a database is designed which will meet these requirements as cleanly as possible. This step is called Information Level Design and it is taken independent of any individual DBMS.

In the second step this information level design is transferred in to a in to a design for the specific DBMS that will be used to implement the system in question. This step is called Physical Level Design, concerned with the characteristics of the specific DBMS that will be used. A database design runs parallel with the system design. The organization of the data in the database is aimed to achieve two major objectives, the two objectives are:

- Data Integrity
- Data independence

The data base design is made up of three levels:

- Conceptual level (High level)
- Physical level (Low level)
- View level (Representation level)

Conceptual Level

Conceptual level describes the essential features of the system data. It uses symbols and is called entity-relationship analysis. An entity is a conceptual representation of an object.

Physical Level

In this level data is stored physically. That is an internal schema describes the physical storage structure of the database.

View Level

This level is used to describe how the user views the records or objects in the database. Data structuring is refined through a process called normalization.

A **foreign key** exists when the primary key of a parent entity exists in a child entity. A foreign key requires that values must be present in the parent entity before like values may be inserted in the child entity.

TABLES DESIGN

Database name : kout

No, of tables : 13

1. Table Name : login

   Primary Key : id

   Description : This table stores the details of login

Field name	Data type	Constraints	Description
id	int(10)	primary key	Login id
lkey	char(8)	not nul	Login key
email	varchar(50)	not null	E-mail of Consumer
password	varchar(32)	not null	Password of Consumer
status	enum(‘0’,’1’,’2’)	not null	Status of user
usertype	enum(‘0,’1’,’2’,’3’)	not null	Type of user
otp	varchar(10)	not null	Type of user

2. Table Name :staffreg

Primary Key : id

Description :This table stores the details of staff

Field name	**Data type**	**Constraints**	**Description**
id	int(10)	primary key	Staff id
skey	char(8)	not null	Staff key
name	varchar(50)	not null	Name of Staff
address	varchar(100)	not null	Address of Satff
phone	varchar(15)	not null	Phone no. of staff
pincode	varchar(8)	not null	Pincode of Satff
district	varchar(20)	not null	District of Satff
place	varchar(100)	not null	Taluk of Staff
dob	date	not null	Dob of Satff
gender	varchar(20)	not null	Gender
store	varchar(20)	not null	Type of Store
sname	varchar(100)	not null	Store name of Satff
storied	varchar(50)	not null	Store id of Staff
loginid	int(10)	Foreign key	Login id

3. Table Name :userreg

Primary Key :id

Description :This table stores the details of Consumer

Field name	Data type	Constraints	Description
id	int(10)	primary key	Consumer id
ukey	char(8)	not null	Consumer key
uname	varchar(50)	not null	Name of Consumer
age	date	not null	Range of age of the Consumer
ubuilding	varchar(70)	not null	Building name of Consumer
ustreet	varchar(50)	not null	Street of the Consumer
upostoffice	varchar(50)	not null	Post office of the Consumer
udistrict	varchar(50)	not null	District of Consumer
uplace	varchar(50)	not null	Taluk of Consumer
upincode	varchar(10)	not null	Pincode of Consumer
ulandmark	varchar(100)	not null	Landmark of the Consumer
uphone	varchar(15)	not null	Phone number of the Consumer
card	varchar(20)	not null	Ration Card number of Consumer
cardimage	varchar(200)	not null	Ration Card image of Consumer
cardtype	varchar(20)	not null	Ration card type of Consumer
gender	varchar(10)	not null	Gender of the Consumer
health	varchar(10)	not null	Consumer health condition disability or not
ustatus	enum('0','1','2','3')	not null	Consumer is aprroved or not
member	varchar(10)	not null	Number of members in card
qrcode	text	not null	QR code of the Consumer
loginid	int(10)	foreign key	Login id

4. Table Name :product

Primary Key :id

Description : This table stores the details of product

Field name	**Data type**	**Constraints**	**Description**
id	int(10)	primary key	Product id
prdctkey	char(8)	not null	Product key
pcat	varchar(50)	not null	Product Category
psub	varchar(50)	not null	Product Subcategory
subsidy	enum(‘0’,‘1’)	not null	Product item is subsidy or not
pname	varchar(50)	not null	Product Name
pcname	varchar(50)	not null	Product company name
pqnty	varchar(50)	not null	Product Quantity
pdetails	varchar(100)	not null	Product Details
orgamnt	varchar(10)	not null	Non-subsidy amount of Product
pamnt	varchar(10)	not null	Subsidy amount of Product
pdate	Date	not null	Product added Date
pimg	varchar(200)	not null	Image of Product
store	enum(‘0’,’1’)	not null	Product store
loginid	int(10)	foreign key	Login id

5. Table Name :cart

Primary Key :id

Description :This table stores the details of product that selected by the Consumer

Field name	**Data type**	**Constraints**	**Description**
id	int(10)	Primary Key	Cart id
cartkey	char(8)	not null	Cart key
pkey	char(8)	not null	Id of product in product table
prdctqnty	int(10)	not null	Product quantity that entered by Consumer
addqnty	varchar(10)	not null	Quantity added by the Consumer
prdctamnt	varchar(20)	not null	Amount of product
fairamnt	varchar(10)	not null	Fair amount of the product
year	year(4)	not null	Year of product added to the cart
month	int(15)	not null	Month of product added to the cart
cdate	date	not null	Current date of product that added to the cart
loginid	int(10)	foreign key	Stores the login id of Consumer
cod	enum(‘0’,‘1’)	not null	Cash on Delivery or not
pymntstatus	enum(‘0’,‘1’)	not null	Payment is done or not
deliver	enum(‘0’,‘1’)	not null	The item is delivered or not
verify	enum(‘0’,‘1’)	not null	Verify the Consumer by deliver agent when order placed
cancelstatus	enum(‘0’,‘1’)	not null	Ordered product is cancelled or not
porder	enum(‘0’,’1’)	not null	Order status of purchased product
deliverstatus	enum(‘0’,’1’)	not null	Status of product delivery
purchaseid	int(10)	not null	Id of purchased product

6. Table Name : deliverreg

Primary Key : id

Description : This table stores the details of deliver agent

Field name	**Data type**	**Constraints**	**Description**
id	int(10)	Primary Key	Id of this table
dkey	char(8)	not null	Key of this table
dname	varchar(100)	not null	Name of the Agent
daddress	varchar(150)	not null	Address of the Agent
dphone	varchar(15)	not null	Phone of the Agent
ddistrict	varchar(100)	not null	District of the Agent
dplace	varchar(100)	not null	Place of the Agent
gender	varchar(20)	not null	Gender of the Agent
appoint	enum((‘0’,‘1’)	not null	Appoint status
staffid	int(10)	not null	Appointed Staff id
loginid	int(10)	foreign key	login id of deliver agent

7. Table Name : allocate

Primary Key : id

Description : This table is used to allocate the product to Consumer

Field name	**Data type**	**Constraints**	**Description**
id	int (10)	primary key	id of the table
akey	char (8)	not null	Key of the table
cardtype	varchar (100)	not null	ration card type
nos	enum(‘0’,’1’)	not null	ration provided for each person or not
product	varchar(50)	not null	Product on the ration shop
quantity	varchar(15)	not null	Quantity of the ration product
pamount	varchar(15)	not null	Subsidy amount
orgamount	varchar(10)	not null	Non-subsidy amount
kit	enum (‘0’,’1’)	not null	Special kit is available or not
loginid	int (10)	foreign key	login id

8. Table Name : dorder

Primary Key : id

Description : This table stores the details of staff

Field name	**Data type**	**Constraints**	**Description**
id	int (10)	primary key	table id
okey	char (8)	not null	table key
pykey	char (8)	not null	key
deliverid	int (10)	not null	id of deliver agent
orderdate	date	not null	date of order placed
useraddr	varchar(15)	not null	Address of the Consumer
userid	int(10)	not nul	id of Consumer
loginid	int (10)	foreign key	loginid of staff

9. Table Name : feedback

Primary Key : id

Description : This table stores the feedback of the consumer.

Field name	Data type	Constraints	Description
id	int(10)	primary key	Table id
fbkey	char(8)	not nul	Feedback key
rate	varchar(50)	not null	Rating of store services
storeid	int(10)	not null	Store id
crntdate	date	not null	current date
loginid	int(10)	foreign key	Consumer loginid

10. Table Name : payment

Primary Key : id

Description : This table stores payment details

Field name	**Data type**	**Constraints**	**Description**
id	int (10)	primary key	id of the table
paymentkey	char (8)	not null	key of the table
cardname	varchar (50)	not null	debit/credit card holder name
cardno	varchar (25)	not null	debit/credit card number
cvv	varchar (4)	not null	cardcvv
month	varchar (10)	not null	paid month of card
year	year (4)	not null	paidyear of card
amount	varchar (50)	not null	total amount items that purchased
coin	varchar(10)	not null	super coin
prdctid	int (10)	not null	id of the product
loginid	int (10)	foreign key	login id of Consumer
crntdate	date	not null	current date of paid

11. Table Name : bank

Primary Key : id

Description : This table used to store the bank details

Field name	**Data type**	**Constraints**	**Description**
id	int (10)	primary key	id of the table
bankkey	char (8)	not null	key of the table
bankname	varchar (60)	not null	name of bank
branchname	varchar (50)	not null	bank brach name
ifsccode	varchar (50)	not null	ifsc code of account holder
accntholder	varchar (20)	\not null	account holder name
cardno	varchar (20)	not null	debit/credit card number
cvv	varchar (4)	not null	cvv of the card
month	varchar (5)	not null	expiry month of the card
year	year (4)	not null	expiry year of the card
totatamnt	varchar (20)	not null	total balance of the account holder
contactno	varchar (10)	not null	contact number of the account holder

12. Table Name : shipaddr

Primary Key : id

Description : This table is used to store auxiliary or secondary address of the consumer for shipping the order

Field Name	**Data type**	**Constraints**	**Descriptions**
id	int(10)	primary key	Id of the table
shipkey	char(8)	not null	key of the table
spdistrict	varchar(50)	not null	Secondary district of consumer
sptaluk	varchar(50)	not null	Secondary taluk of consumer
spstreet	varchar(50)	not null	Secondary street of consumer
sppostoffice	varchar(50)	not null	Secondary post office of consumer
spname	varchar(50)	not null	Secondary name of consumer
spbuilding	varchar(100)	not null	Secondary building name of consumer
shpincode	varchar(10)	not null	Secondary pincode of consumer
splandmark	varchar(100)	not null	Secondary landmark of consumer
loginid	int(10)	foreign key	loginid of consumer

13. Table Name : ration

Primary Key : id

Description : Stocks of product available in ration shop that allotted by admin

Field Name	**Data Type**	**Constraints**	**Description**
id	int(10)	Primary Key	Id of table
rtnkey	char(8)	not null	Key of the table
alctkey	char(8)	not null	Key of allocate table
totalquantity	varchar(20)	not null	Total available stock in ration shop
loginid	int(10)	foreign key	loginid of staff

3.4 DATA FLOW DIAGRAMS

Analysis model helps us to understand the relationship between different components in the system design. Analysis model shows the user clearly how a system will function. This is the first technical representation of a system. The analysis modeling must achieve three primary objectives.

- To establish a basis for creation of software design.
- To describe what the user requires.
- To define set of requirements that can be validated once the software is built.

A dataflow diagram is a graphical technique that depicts information and transforms that are applied as data move from input to output. The DFD is used to represent increasing information flow and functional details. A level-0 DFD is also called a fundamental system model represents the entire software elements as a single bible with input and output indicated by incoming and outgoing arrows respectively.

Additional process and information low parts are represented in the next level, i.e., level 1 DFD. Each of the process represented at level 1 are sub functions of overall will be further represented into sub functions in the next level, i.e., level2.

Data flow diagram is a mean of representing a system at any level of detail with a graphic network of symbols showing data flows, data process and data stores. The purpose of data flow diagram is to provide a semantic bridge between users and system developers.

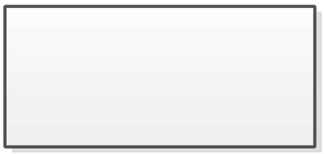
The diagram is graphical eliminating thousands of words, logical representations, modeling what system does; hierarchical, showing systems at any level of details; and jargon less, allowing user understanding and reviewing.

The goal of data flow diagram is to have a commonly understood model of system. The diagram is the basis of the structural analysis. Data flow diagrams are supported by other techniques such a decision tables, decision tree and structured English.

Components of Data Flow Diagrams

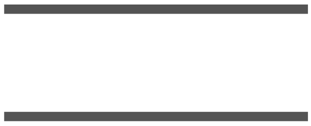
There are only four symbols that are used in the drawing of data flow diagrams. These are explained below together with the rules that apply to them.

- External Entities



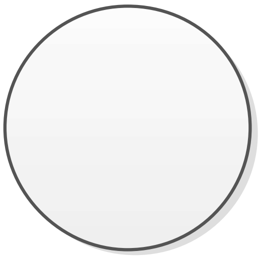
External entities represent the sources of data that enter the system or the recipients of data that leave the system. They can be duplicated, one or more times on the diagram to avoid line crossing.

- Data stores



Data stores represent stores of data within the system. Data stores may be long-term files such as sales ledgers, or may be short-term accumulations: For example batches of documents that are waiting to be processed. Each data store should be given a reference followed by an arbitrary number.

- Process

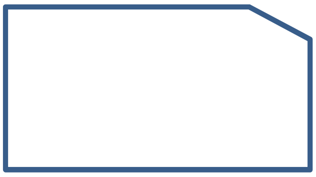


Processes represent activities in which data is manipulated by being stored or retrieved or transformed in some way. A circle represents it. The process will show the data transformation or change. Data coming into a process must be "worked-on" or transformed in some way. Thus all process may have an input and output.

- Dataflow

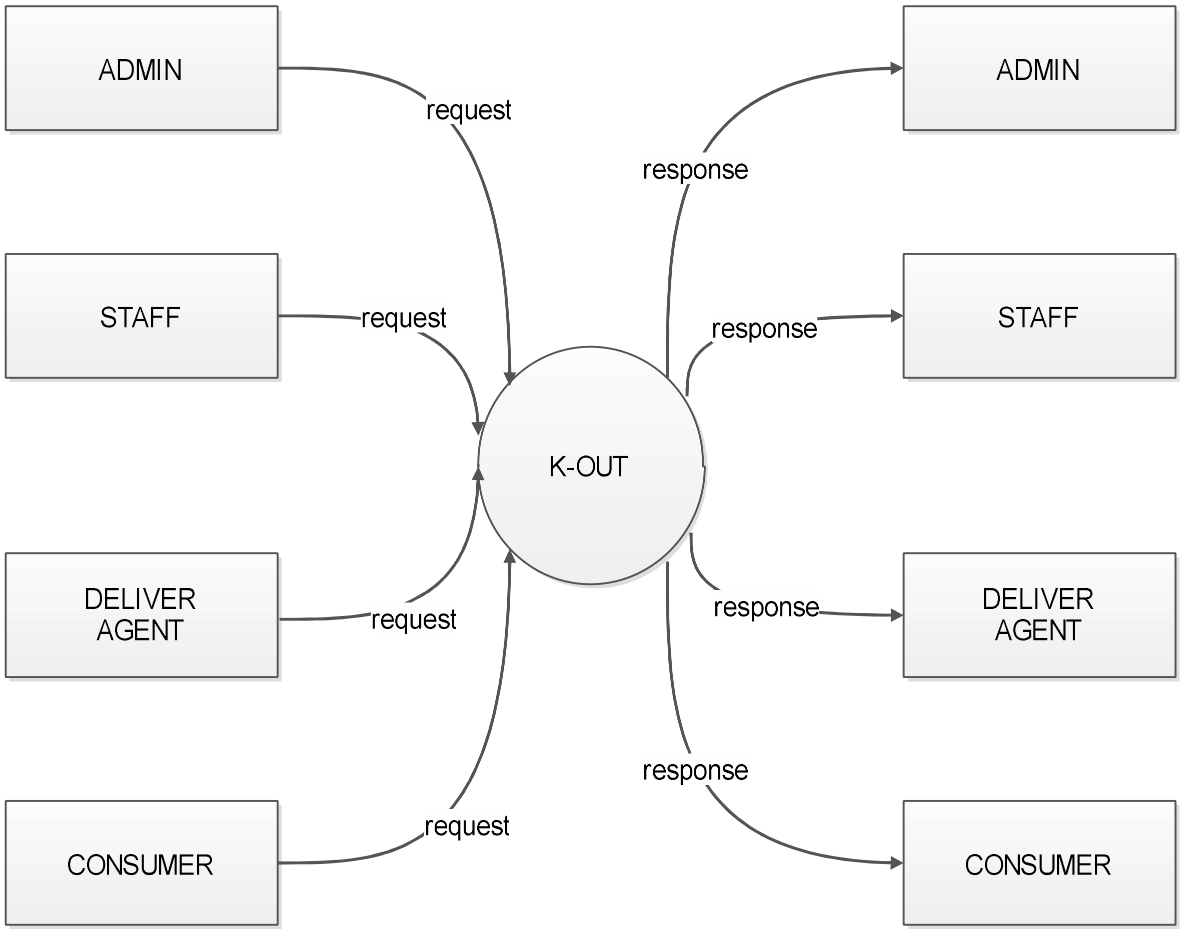
A data flow shows the flow of information from its source to its destination. A line represents a data flow, with arrowheads showing the direction of flow. Information always flows to or from a process and may be written, verbal or electronic. Each data flow may be referenced by the processes or data stores at its head and tail, or by a description of its contents.

- Output

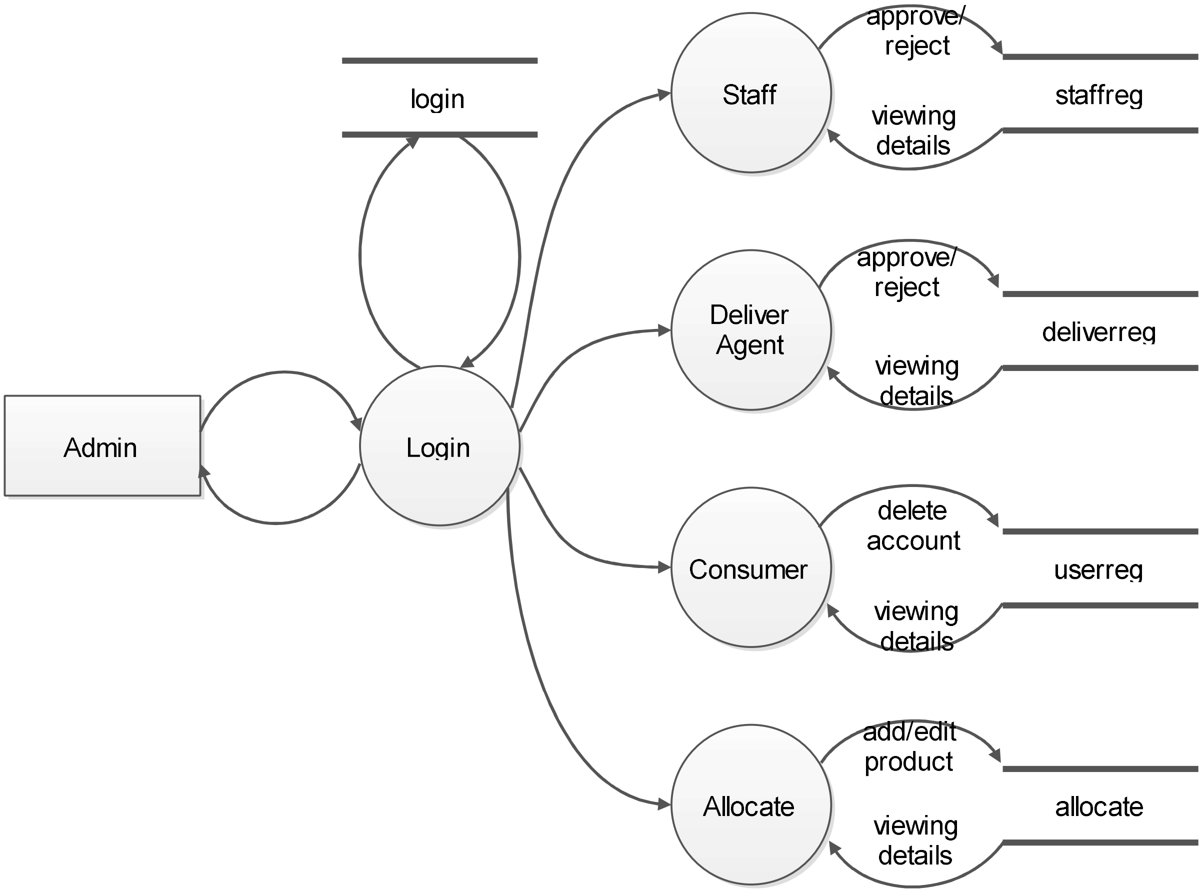


The output symbol is used when a card copy is produced and the user of the copy cannot be clearly specified.

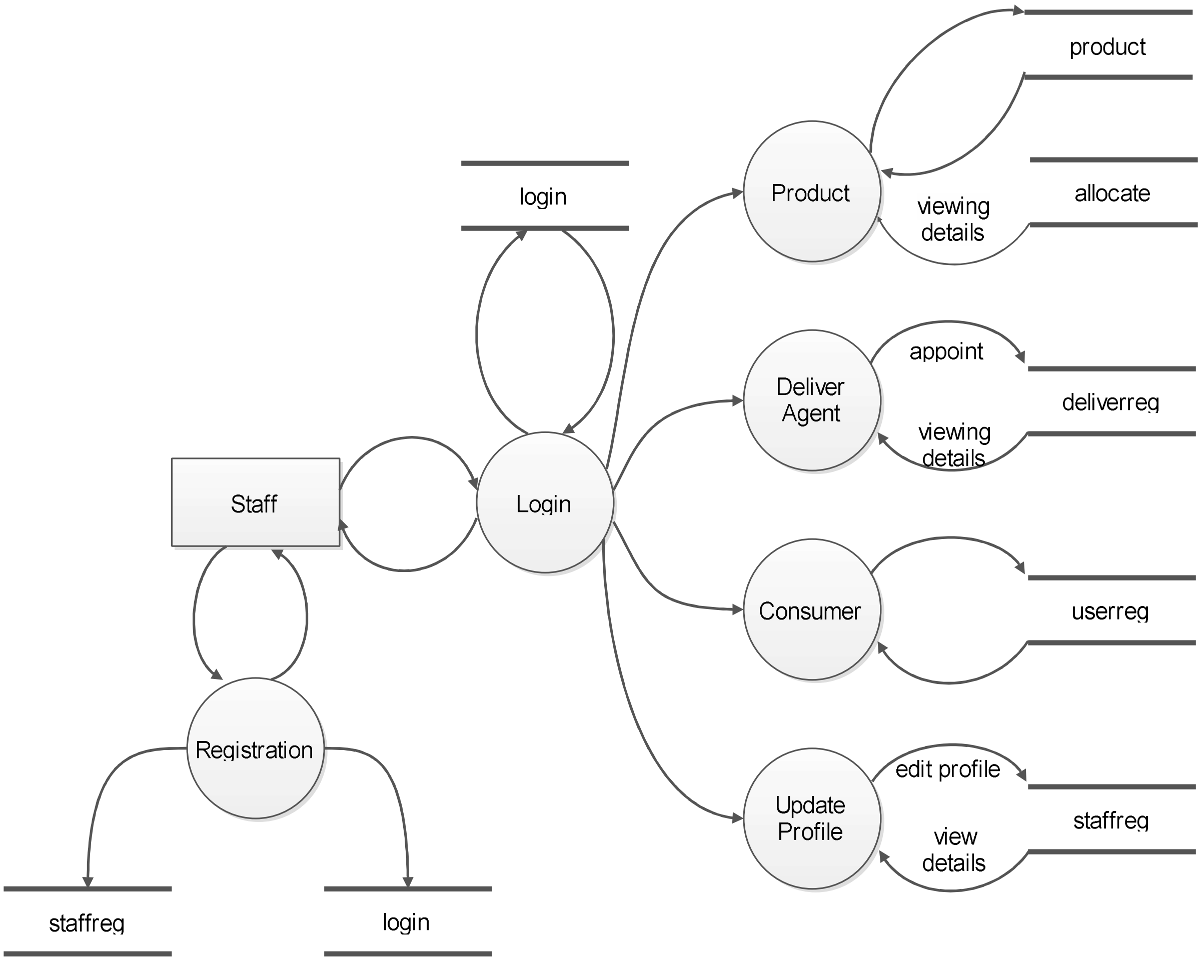
LEVEL 0: DFD 0


ADMIN
STAFF
DELIVER AGENT
CONSUMER
request
request
request
request
K-OUT
response
response
response
response
ADMIN
STAFF
DELIVER AGENT
CONSUMER

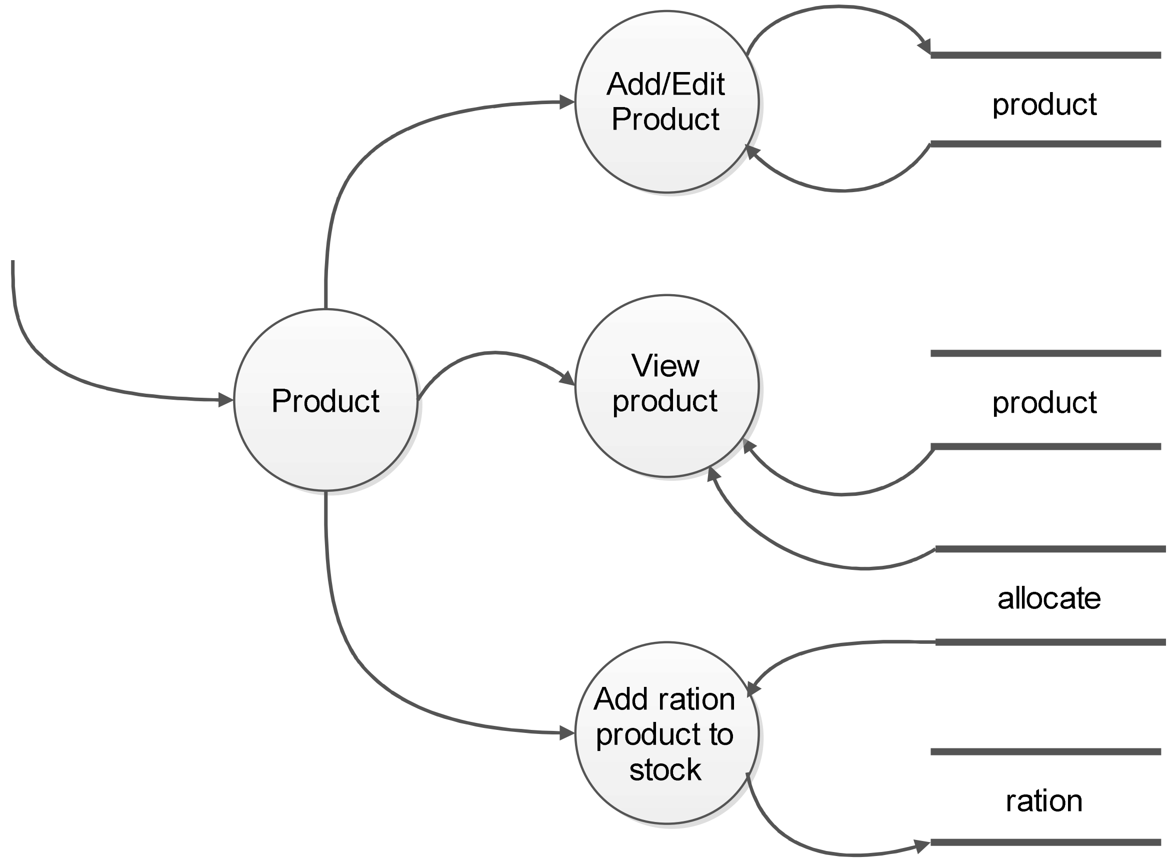
Level 1: DFD for Admin



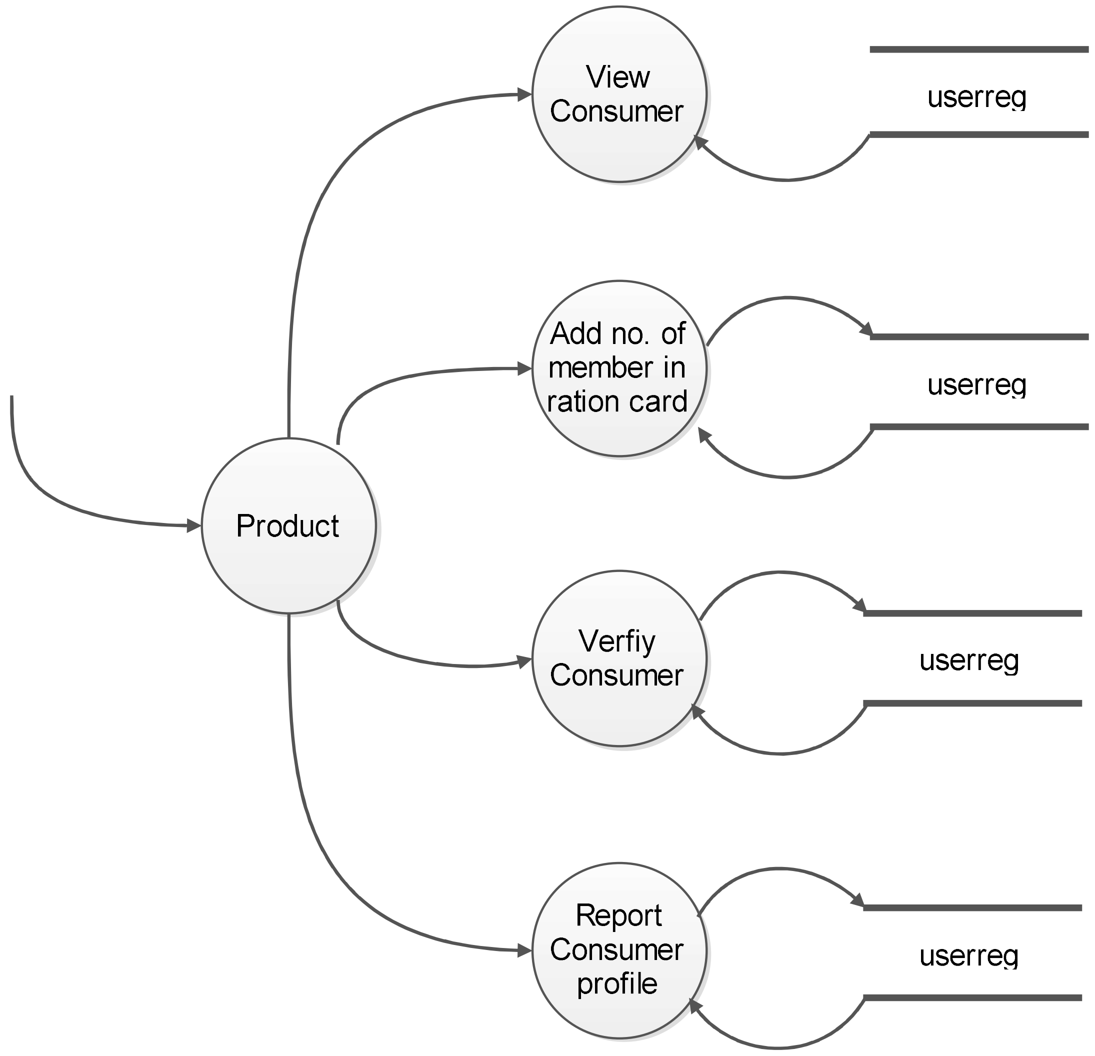
Level 2: DFD for STAFF



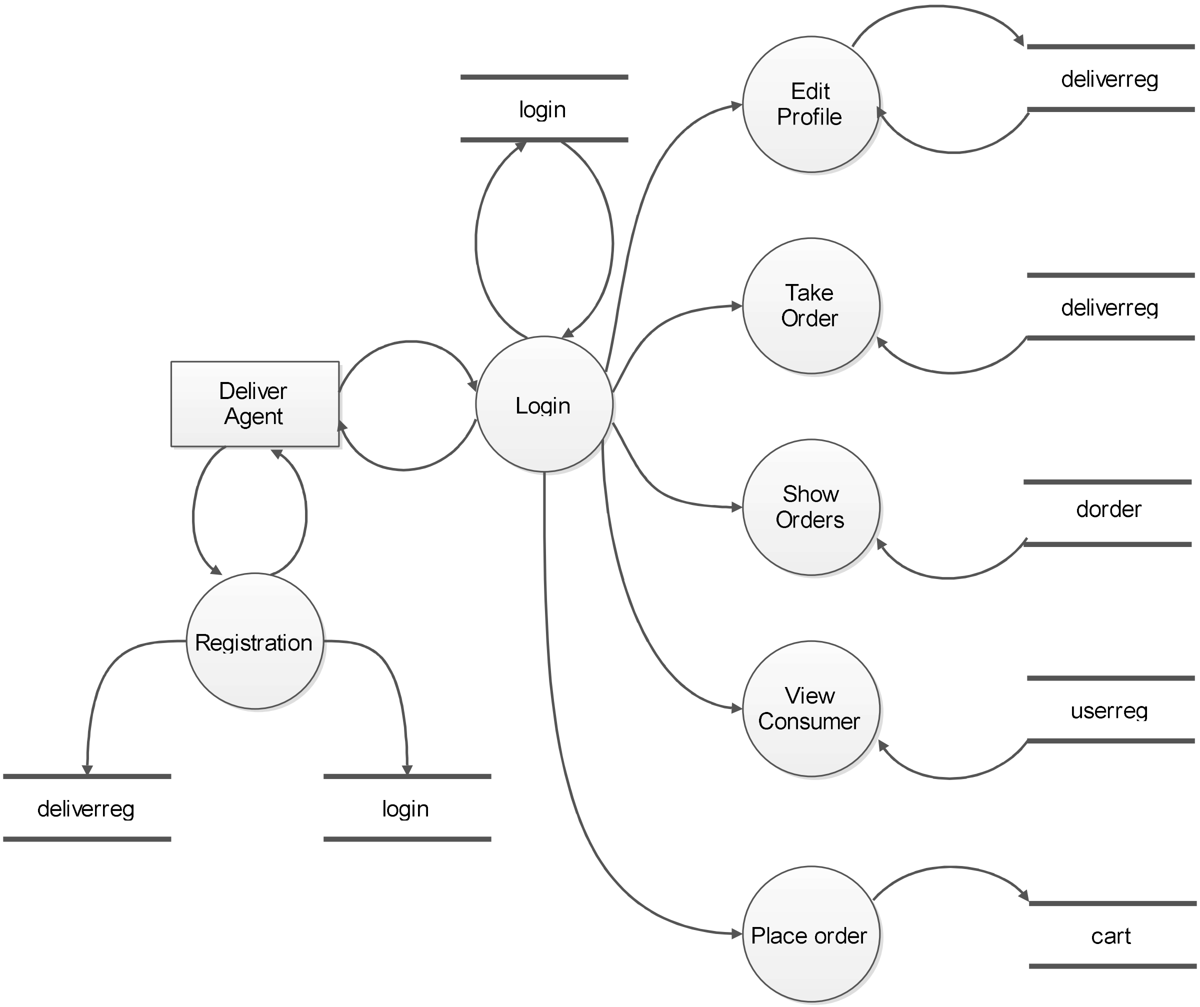
Level 2.1.1: DFD for Product



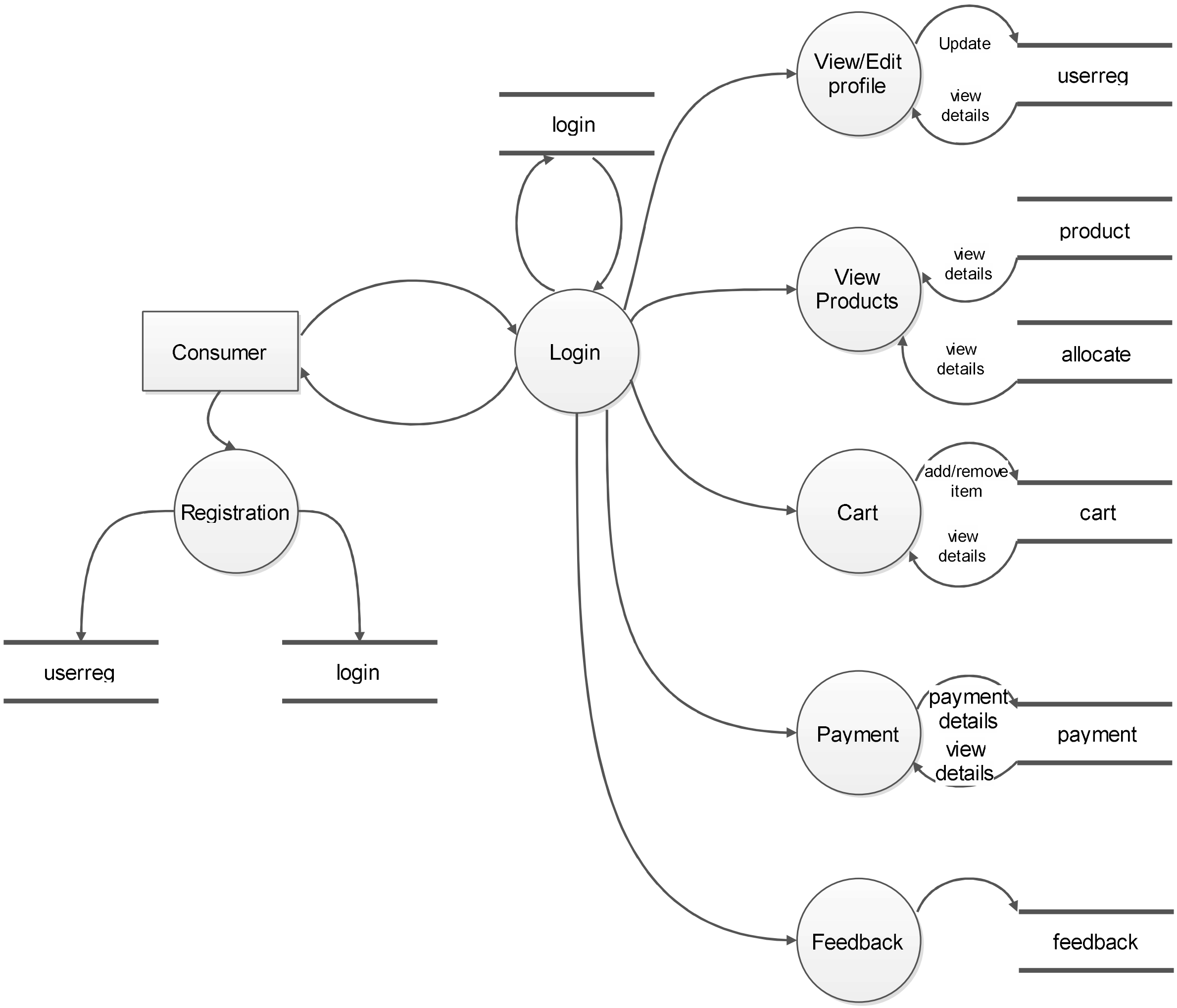
Level 2.1.3: DFD for Consumer



Level 3: DFD for DELIVER AGENT



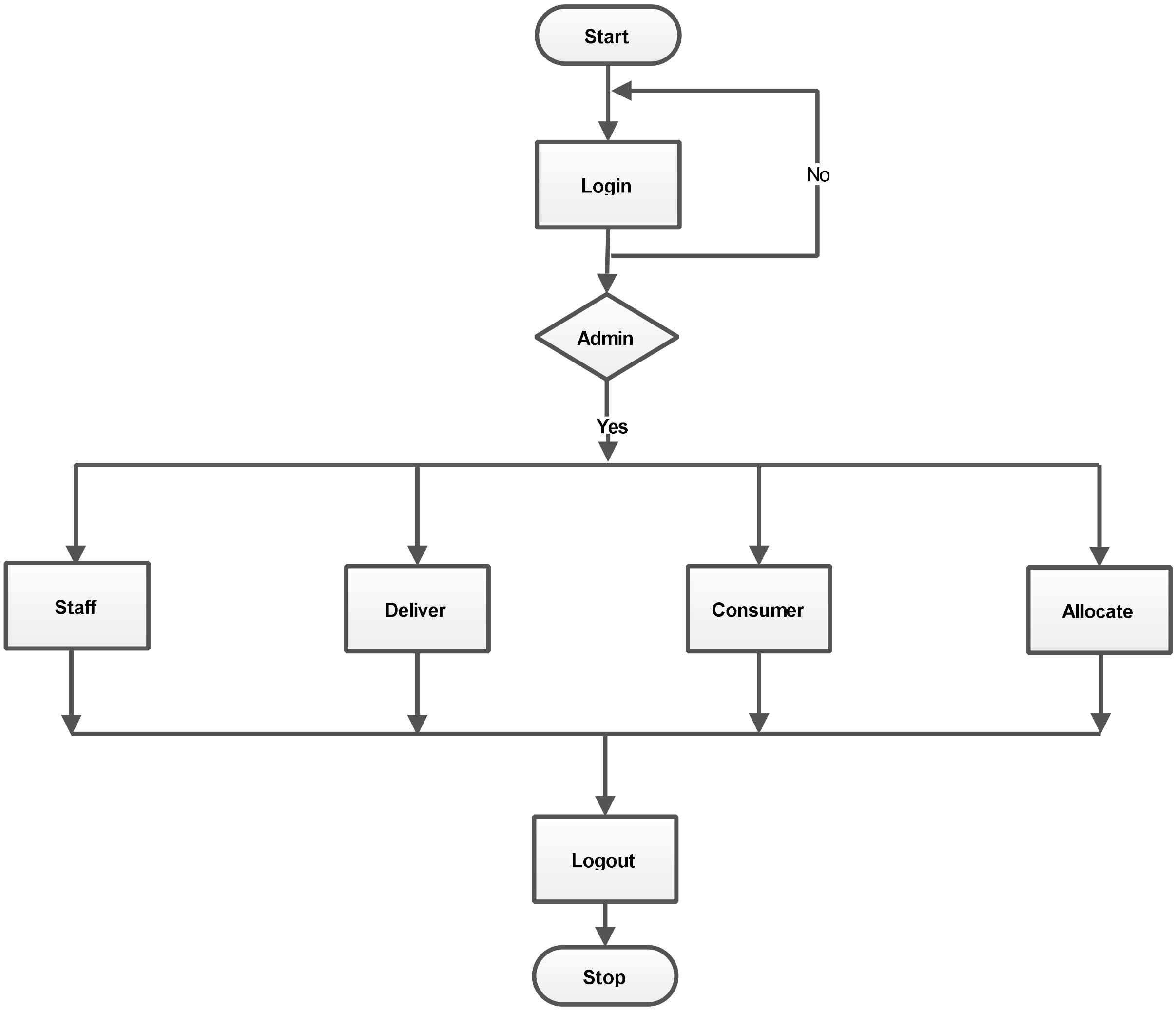
Level 4: DFD for CONSUMER



3.5 SYSTEM FLOWCHART

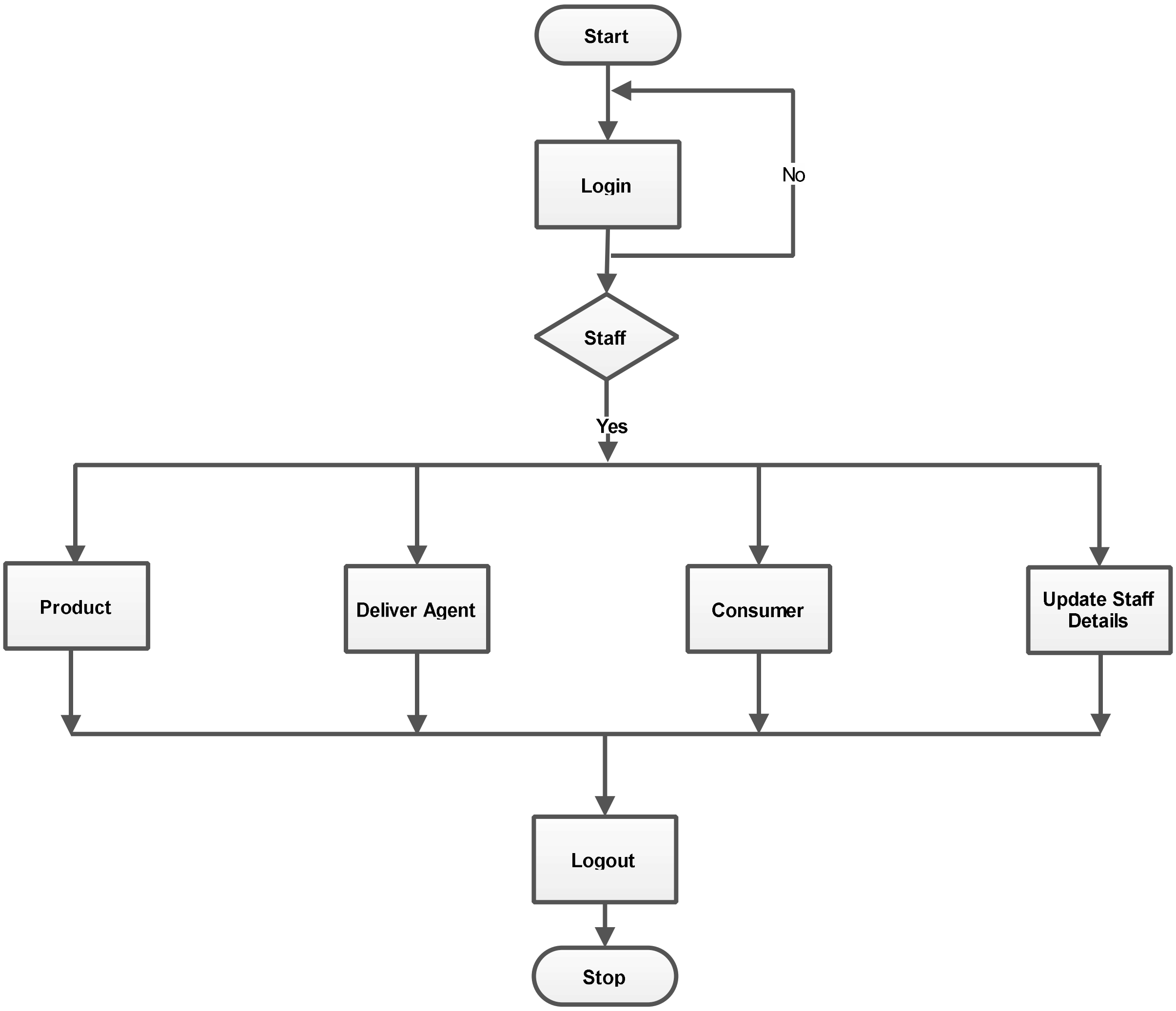
SYSTEM FLOWCHART

Admin



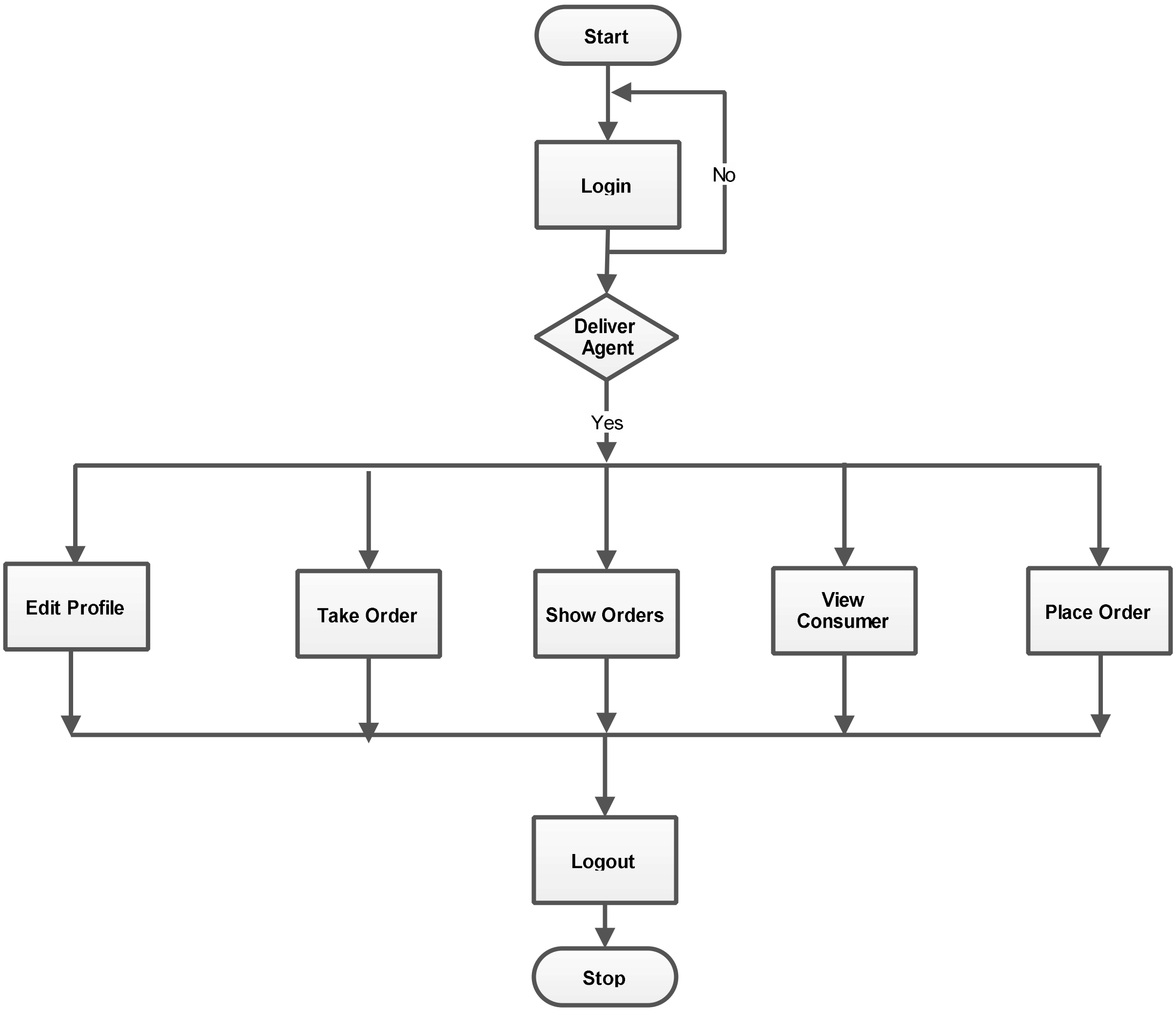
SYSTEM FLOWCHART

Staff



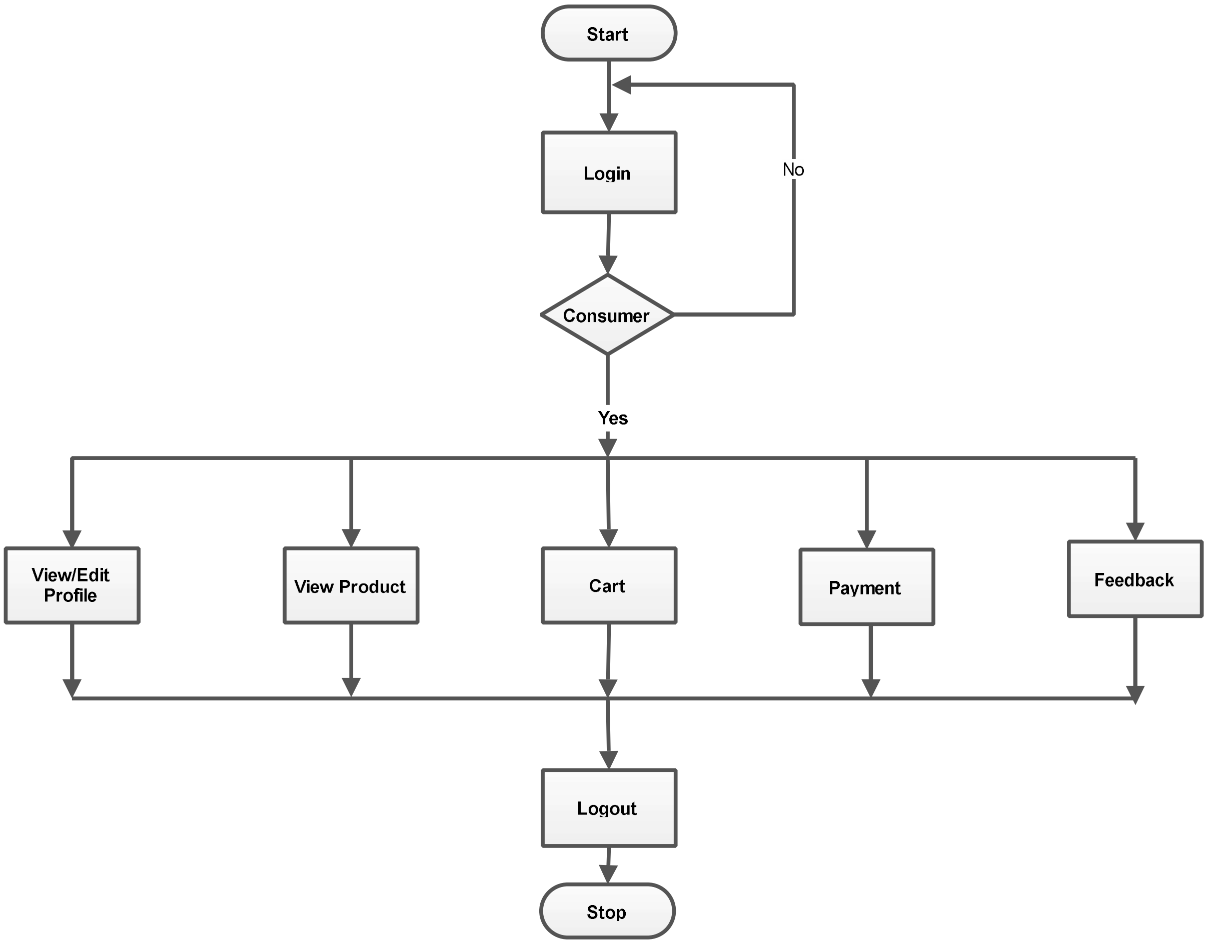
SYSTEM FLOWCHART

Deliver Agent



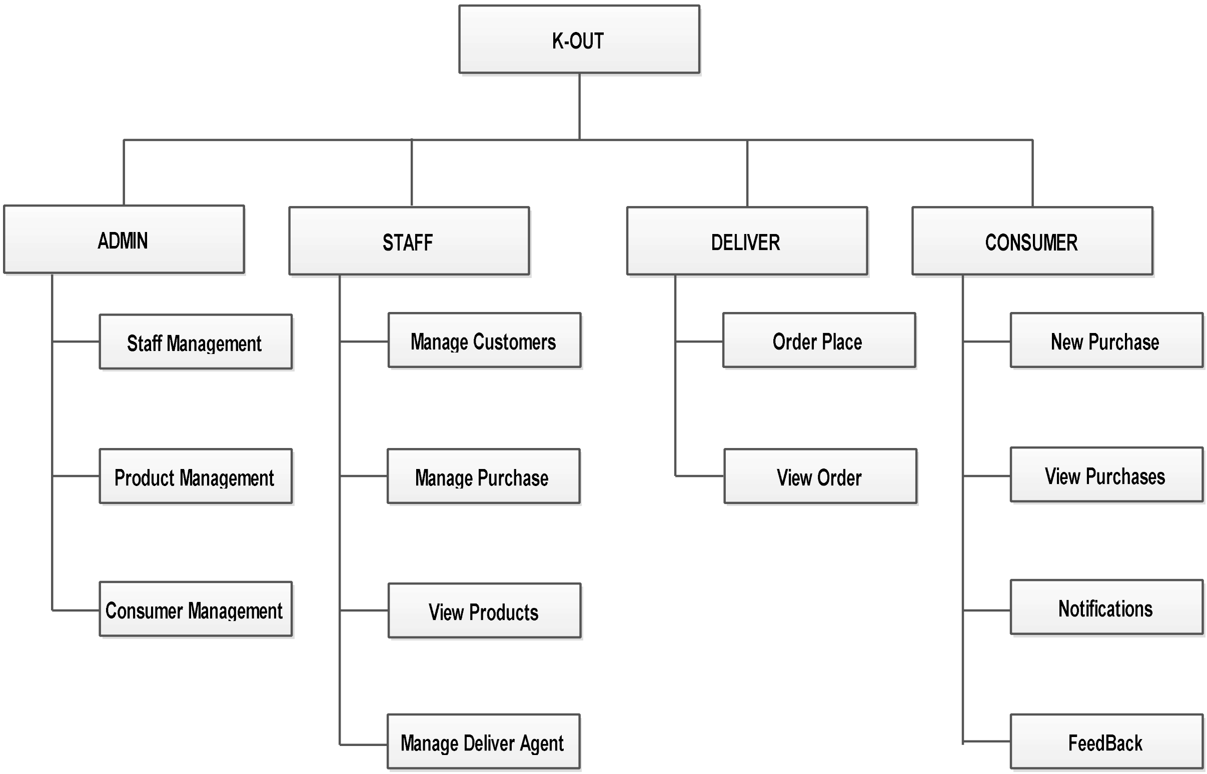
SYSTEM FLOWCHART

Consumer



3.6 STRUCTURE CHART

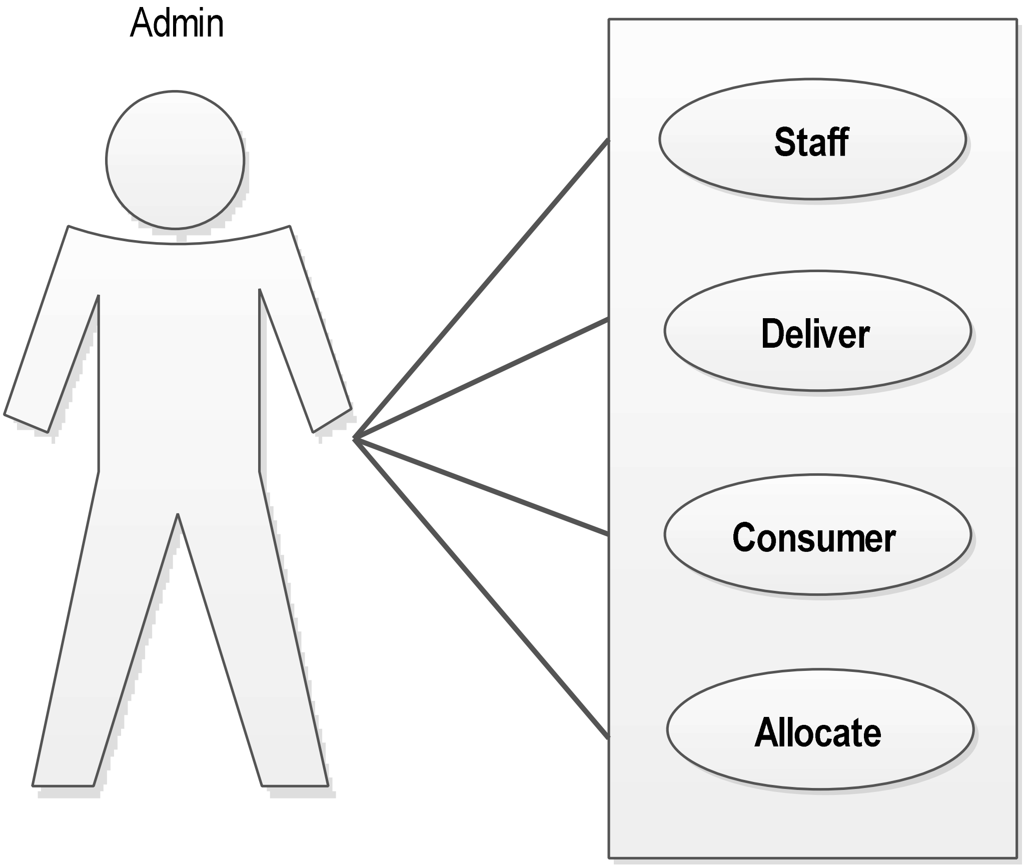
STRUCTURE CHART FOR K-OUT



3.7 USE CASE DIAGRAM

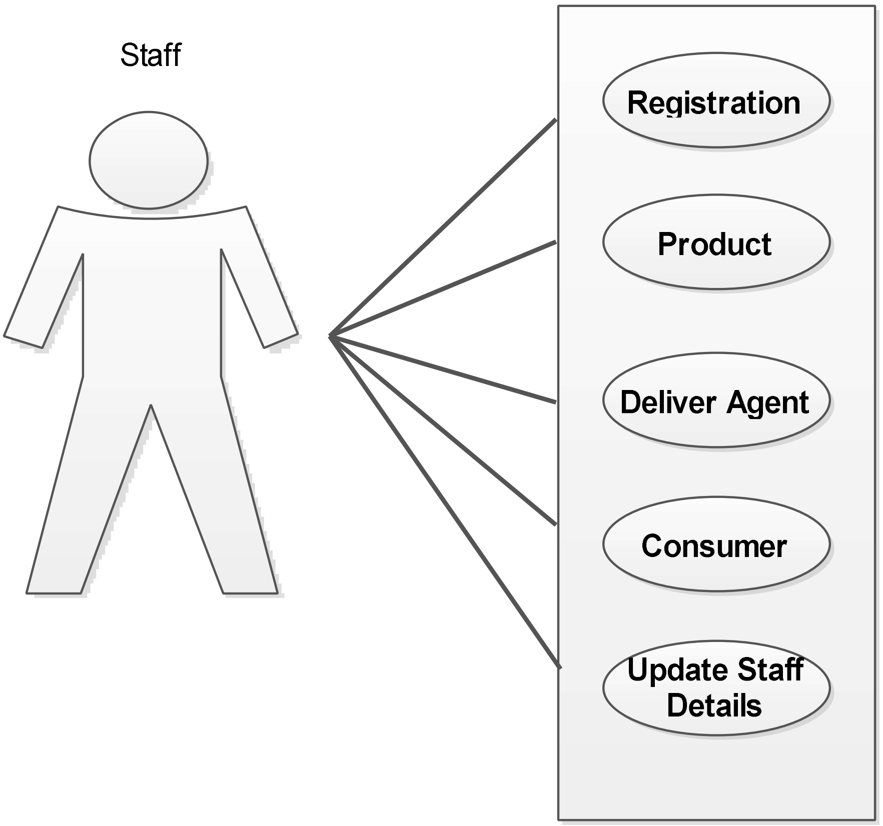
USE CASE DIAGRAM

Admin



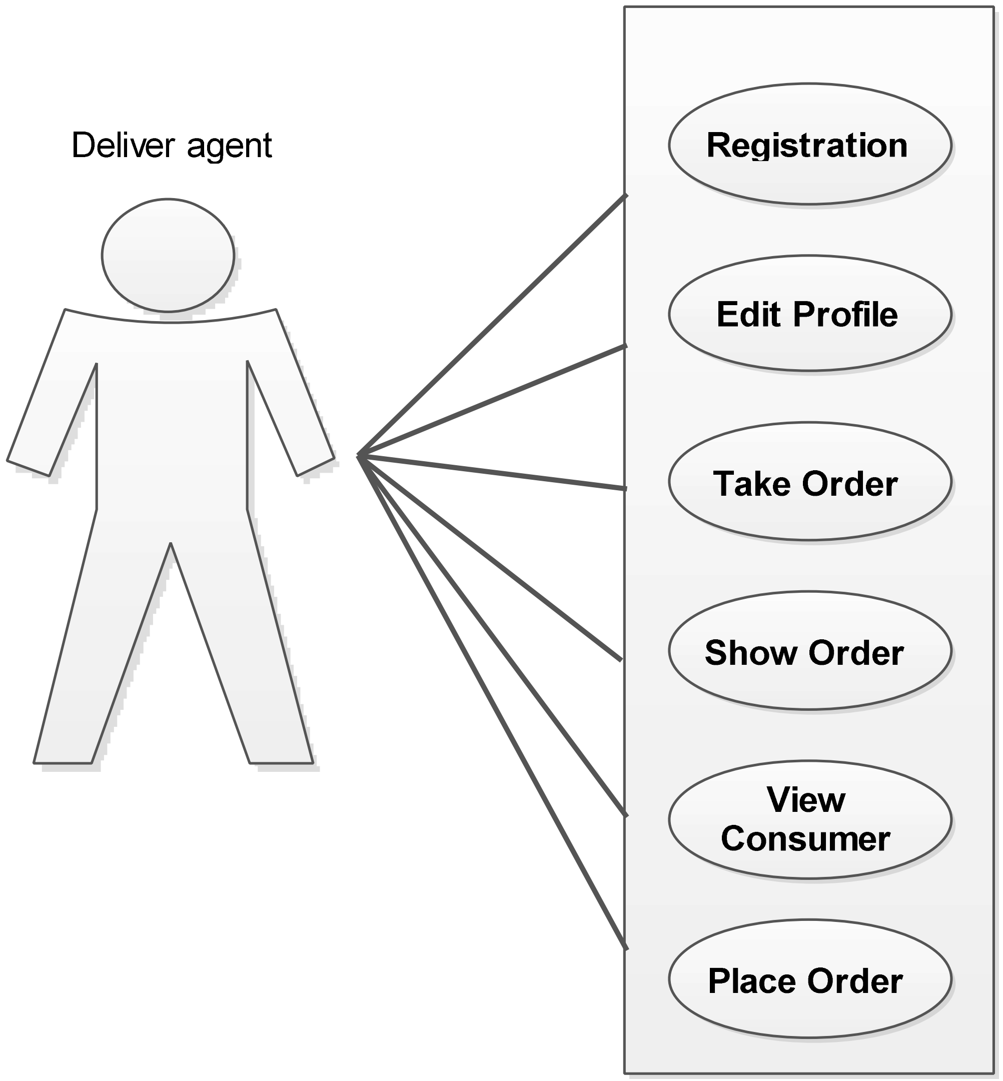
USE CASE DIAGRAM

Staff



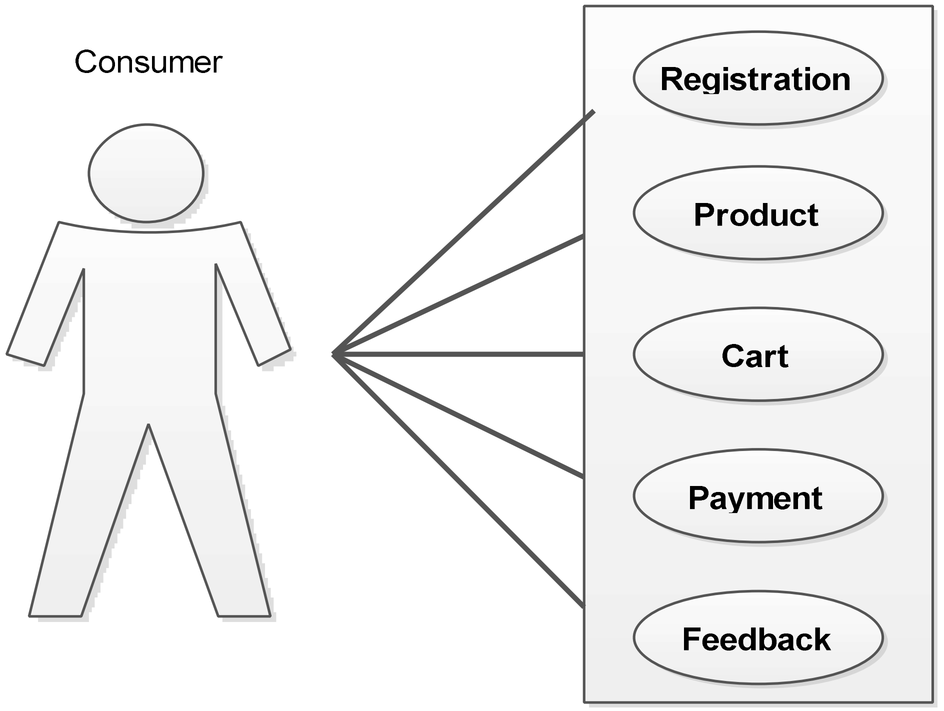
USE CASE DIAGRAM

Deliver Agent



USE CASE DIAGRAM

Consumer



4. SYSTEM CODING

The coding step is it process that transforms design into a programming language. It translates a detail design representation of software into a programming language realization. The translation process continues when a compiler accepts source code as input as produces a machine dependent object code as output. Quality of source code can be improved by the use of structures coding techniques, good coding style and readable and consistent code format. During coding, some coding standards are to be followed. This has two purposes: reducing the chance of making it easier for some time to modify the code later on. Coding phase affect on both testing and maintenance profoundly.

Database Connection Code:

```
<?php
$dbpath="localhost";
$dbusername="root";
$dbpassword="";
$dbname="kout";
?>
```

CODE

```
<?php

classuserclass

{

            function
userreg($name,$age,$address,$street,$upo,$district,$place,$pincode,$land,$card,$files=null,$car
dtype,$gender,$health,$contact,$password)

            {

      $name=strtoupper($name);

      $address=strtoupper($address);

      $street=strtoupper($street);

      $place=strtoupper($place);

      $land=strtoupper($land);

      $enc=md5($password);

      $key1=uniquekey("login","lkey");

      $check="select * from userreg where card='".$card."'";

      $ch=mysql_query($check);

      $chk=mysql_fetch_array($ch);
```

```
if($chk!=null)

	{

echo "<script> alert('Ration card number already existed');

window.location.href='userreg.php';</script>";

	}

else

	{

	$qry1="insert into login(lkey,email,password,usertype,status)values('".$key1."','".$contact."','".$enc."','2','1')";

	$exe1=mysql_query($qry1);

	$key=uniquekey("userreg","ukey");

	$id=keytoid("login","lkey",$key1);

	$qry="insert into userreg(ukey,uname,age,ubuilding,ustreet,upostoffice,udistrict,uplace,upincode,ulandmark,uphone,card,cardimage,cardtype,gender,health,loginid)values('".$key."','".$name."','".$age."','".$address."','".$street."','".$upo."','".$district."','".$place."','".$pincode."','".$land."','".$contact."','".$card."','".$files['name']."','".$cardtype."','".$gender."','".$health."','".$id."')";

	$img="cardimage/".$key;

	$exe=mysql_query($qry);

if($exe&&$exe1)
```

```
        {
mkdir($img);
move_uploaded_file($files["tmp_name"],$img."/".$files["name"]);
            echo"<script>alert('Registration successfull');
window.location.href='qrcode.php?key={$key}';</script>";
        }
else
        {
            echo"<script>alert('Registration unsuccesfull ')</script>";
        }
      }
  }
functionqrcode($key)
  {
    $uid=keytoid("userreg","ukey",$key);
    $qry="select * from userreg inner join login on login.id=userreg.loginid where 
userreg.id='".$uid."'";
    $exe=mysql_query($qry);
    $arr=array();
while($rr=mysql_fetch_array($exe))
    {
      $arr[]=$rr;
    }
return $arr;
  }
functionqrupdate($rootUrl,$key)
  {
    $uid=keytoid("userreg","ukey",$key);
    $qry="update userreg set qrcode='".$rootUrl."' where id='".$uid."'";
    // echo $qry;exit;
    $exe=mysql_query($qry);
if($exe)
    {
echo "<script> alert('Updated');
window.location.href='login.php';</script>";
    }
else
echo "<script> alert('Failed to update');
window.location.href='userreg.php';</script>";
  }
  function
staffreg($name,$address,$pincode,$district,$place,$dob,$gender,$store,$sname,$storeid,$contact
,$email,$password)
  {
      $name=strtoupper($name);
      $address=strtoupper($address);
      $place=strtoupper($place);
      $sname=strtoupper($sname);
      $enc=md5($password);
      $key1=uniquekey("login","lkey");
      $check="select * from login inner join staffreg on staffreg.loginid=login.id where
email='".$email."' or (sname='".$sname."' and store='".$store."' and storeid='".$storeid."')";
      $ch=mysql_query($check);
      $dup=mysql_fetch_array($ch);
if ($dup!=null)
      {
echo "<script> alert('Email/Store already existed');
window.location.href='staffreg.php';</script>";
      }
```

```
else
      {
       $qry1="insert                                                                    into
login(lkey,email,password,usertype,status)values('".$key1."','".$email."','".$enc."','1','1')";
       $exe1=mysql_query($qry1);
       $key=uniquekey("staffreg","skey");
       $id=keytoid("login","lkey",$key1);
       $qry="insert into  staffreg (skey, name, address, phone, pincode, district, place, dob, gender,
store,  sname,  storeid,  loginid)  values  ('".$key."',  '".$name."',  '".$address."',  '".$contact."',
'".$pincode."',  '".$district."',  '".$place."',  '".$dob."',  '".$gender."',  '".$store."',  '".$sname."',
'".$storeid."', '".$id."')";
       $exe=mysql_query($qry);
if($exe&&$exe1)
       {
echo"<script>alert('Registration successfull. Please wait for admin approval.');
window.location.href='index.php';</script>";
       }
else
       {
echo"<script>alert('Registration unsuccesfull ')</script>";
       }
      }
  }
functionadminstaffview()
  {
     $qry="select * from staffreg inner join login on login.id=staffreg.loginid order by store asc";
     $var1=mysql_query($qry);
     $var2=array();
while($var3=mysql_fetch_array($var1))
     {
            $var2[]=$var3;
      }
return $var2;
  }
functionadminuserview()
  {
      $qry="select * from userreg inner join login on login.id=userreg.loginid";
      $var1=mysql_query($qry);
      $var2=array();
while($var3=mysql_fetch_array($var1))
      {
            $var2[]=$var3;
      }
return $var2;
  }
functionstaffapprove($key)
  {
       $id=keytoid("login","lkey",$key);
       $qry="update login set status='1' where id='".$id."'";
       $exe=mysql_query($qry);
if ($exe)
       {
echo "<script> alert('Approved successfully');
window.location.href='adminstaffview.php';
</script>";
       }
else
       {
echo "<script> alert('Approve unsuccessful');
window.location.href='adminstaffview.php';</script>";
```

```
        }
  }
functionstaffreject($key)
  {
      $id=keytoid("login","lkey",$key);
       $qry="update login set status='2' where id='".$id."'";
       $exe=mysql_query($qry);
if ($exe)
        {
echo "<script> alert('Rejected successfully');
window.location.href='adminstaffview.php';</script>";
        }

else

        {

echo "<script> alert('Rejected unsuccessful');

window.location.href='adminstaffview.php';</script>";

        }

  }

function login($email,$password)

  {
      $pw=md5($password);
      $qry="select lkey,usertype,status from login where email='".$email."' and
password='".$pw."'";
      // echo $qry;exit;
      $exc=mysql_query($qry);
```

```
    $key=null;$c=0;
while ($rr=mysql_fetch_array($exc))
    {
     $key=$rr['lkey'];
     $u=$rr['usertype'];
     $s=$rr['status'];
     $c=$c+1;
    }
if ($c>0)
    {
if ($s==1)
     {
setcookie("lkey",$key);
setcookie("logined",1);
if ($u==0)
     {
header("location:admin.php");
     }
elseif($u==1)
     {
header("location:staffhome.php");
     }
elseif($u==2)
     {
header("location:userhome.php");
     }
elseif($u==3)
     {
header("location:deliverhome.php");
     }
```

```
      }
else
echo "<script>alert('waiting for admin approval')</script> ";
       }
else
echo"<script>alert('login failed')</script>";
  }
  function
sprofile($name,$address,$pincode,$district,$place,$dob,$gender,$store,$sname,$storeid,$contact
,$email,$c)
  {
      $name=strtoupper($name);
      $address=strtoupper($address);

      $place=strtoupper($place);

      $sname=strtoupper($sname);

      $sp=keytoid("login","lkey",$c);

      $qry="update staffreg set name='".$name."', address='".$address."', phone='".$contact."',
pincode='".$pincode."',district='".$district."',place='".$place."',dob='".$dob."',gender='".$gender.
"',store='".$store."',sname='".$sname."',storeid='".$storeid."' where staffreg.loginid='".$sp."'";

      $qry1="update login set email='".$email."' where id='".$sp."'";
      $exe=mysql_query($qry);
      $exe1=mysql_query($qry1);
if($exe&&$exe1)
      {
echo "<script> alert('update success')</script>";
      }
else
```

```
      {
echo "<script> alert('update unsuccess')</script>";
      }
  }
  function
rtnprofile($name,$address,$pincode,$district,$place,$dob,$gender,$store,$sname,$storeid,$conta
ct,$email,$c)
  {
      $name=strtoupper($name);
      $address=strtoupper($address);
      $place=strtoupper($place);
      $sname=strtoupper($sname);
      $sp=keytoid("login","lkey",$c);
      $qry="update staffreg set name='".$name."', address='".$address."', phone='".$contact."',
pincode='".$pincode."',district='".$district."',place='".$place."',dob='".$dob."',gender='".$gender.
"',store='".$store."',sname='".$sname."',storeid='".$storeid."' where staffreg.loginid='".$sp."'";
      $qry1="update login set email='".$email."' where id='".$sp."'";
      $exe=mysql_query($qry);
      $exe1=mysql_query($qry1);
if($exe&&$exe1)
```

```
      {
echo "<script> alert('update success')</script>";
      }
else
      {
echo "<script> alert('update unsuccess')</script>";
      }
  }
functionstaffupdate($k)
  {
      $id=keytoid("login","lkey",$k);
      $qry="select * from staffreg inner join login on login.id=staffreg.loginid where staffreg.loginid='".$id."'";
      $exe=mysql_query($qry);
      $arr=array();
while ($rr=mysql_fetch_array($exe))
      {
        $arr[]=$rr;
      }
return $arr;
  }
  function product($pcat,$psub,$pname,$subsidy,$pcname,$pqnty,$pdetails,$pamnt,$orgamnt,$files=null,$key)
  {
      $key1=uniquekey("product","prdctkey");
      $id=keytoid("login","lkey",$key);
```

```
    $date=date('Y-m-d');
    $qry="insert into product (prdctkey, pcat, psub, pname, subsidy, pcname, pqnty, pdetails,
pamnt, orgamnt, pimg, pdate, loginid) values ('".$key1."' ,'".$pcat."', '".$psub."', '".$pname."',
".$subsidy.",'".$pcname."','".$pqnty."','".$pdetails."','".$pamnt."','".$orgamnt."','".$files['name'
]."','".$date."','".$id."')";
    $img="image/".$key1;
    $exe=mysql_query($qry);
if($exe)
    {
mkdir($img);
move_uploaded_file($files["tmp_name"],$img."/".$files["name"]);
echo"<script>alert('Product added successfully');
window.location.href('staffhome.php');</script>";
    }
else
    {
echo"<script>alert('Operation unsuccesfull ');
window.location.href('product.php');</script>";
    }
  }
functionproductselect($f)
  {
   $id=keytoid("product","prdctkey",$f);
   $qry="select * from product inner join login on login.id=product.loginid where
product.id='".$id."'";
   $ene=mysql_query($qry);
   $arr=array();
while($rr=mysql_fetch_array($ene))
   {
    $arr[]=$rr;
    }
return $arr;
  }

Registration.tpl
<html ng-app="myApp">
{literal}

<style type="text/css">

.myform .required:after {

content: " *";

color: red;

font-weight: 100;

}

</style>

<script src="angularvalidation.js"></script>

<script>

functionmyFunction() {

var x = document.getElementById("pass");

if (x.type === "password") {

x.type = "text";

} else {
```

```
x.type = "password";

}

}

</script>

{/literal}

<body>

<div ng-controller="cntrl">

<form method="POST" enctype="multipart/form-data" class="myform" name="SaveForm" ng-submit="funcSave()" action="">

<table align="center">

<center>

<h2><u>Registration Form </u></h2>

<table align="center" cellspacing="6" cellpadding="6">

<tr>

<th colspan="2"><input type="hidden" name="hide" value="h"></th>

</tr>

<tr>

<th class="form-label required">Name : </th>

<td><input type="text" name="name" required class="form-control" style="text-transform:
```

```
uppercase;" ng-pattern="/^[a-zA-Z. ]*[a-zA-Z]$/" ng-model="name">

<span style="color:red" ng-show="SaveForm.name.$dirty&&SaveForm.name.$invalid" class="ng-hide">Enter  Valid Name.!  </span></td>

</tr>

<tr>

<th class="form-label required">Date of Birth:</th>

<td><input type="date" name="age" class="form-control" style="text-transform: uppercase;" ng-model="age">

<span style="color:red" ng-show="SaveForm.age.$dirty&&SaveForm.age.$invalid" class="ng-hide">Enter Valid  Date Of Birth.!  </span></td>

</tr>

<tr>

<th class="form-label required">House/Flat Name : </th>

<td><textarea name="address" required class="form-control" style="text-transform: uppercase;" ng-pattern="/^[a-zA-Z.  ]*[a-zA-Z]$/" ng-model="address"></textarea><span style="color:red" ng-show="SaveForm.address.$dirty&&SaveForm.address.$invalid" class="ng-hide">

                Enter Valid House/Flat Name.!</span></td>

</tr>

<tr>

<th class="form-label required">Street/Road Name:</th>
```

```
<td><input type="text" name="street" required class="form-control" style="text-transform:
uppercase;" ng-pattern="/^[a-zA-Z. ]*[a-zA-Z]$/" ng-model="street">

<span style="color:red" ng-show="SaveForm.street.$dirty&&SaveForm.street.$invalid"
class="ng-hide"> Enter Valid Street/Road Name.!</span></td>

</tr>

<tr>

<th class="form-label required">Post Office:</th>

<td><input type="text" name="post" required class="form-control" style="text-transform:
uppercase;" ng-pattern="/^[a-zA-Z. ]*[a-zA-Z]$/" ng-model="post"><span style="color:red"
ng-show="SaveForm.post.$dirty&&SaveForm.post.$invalid" class="ng-hide">Enter Valid Post
Office.! </span></td>

</tr>

<tr>

<th class="form-label required">Taluk (Block): </th>

<td><input type="text" required name="place" class="form-control" style="text-transform:
uppercase;" ng-pattern="/^[a-zA-Z. ]*[a-zA-Z]$/" ng-model="place"><span style="color:red"
ng-show="SaveForm.place.$dirty&&SaveForm.place.$invalid" class="ng-hide">Enter Valid
Taluk.! </span></td>

</tr>

<tr>
<th class="form-label required">District :</th>

<td><select name="district" required>
```

```
<option>---SELECT---</option>

<option>Thiruvananthapuram</option>

<option>Kollam</option>

<option>Pathanamthitta</option>

<option>Alappuzha</option>

<option>Kottayam</option>

<option>Idukki</option>

<option>Eranakulam</option>

<option>Thrissur</option>

<option>Palakkad</option>

<option>Malappuram</option>

<option>Kozhikode</option>

<option>Wayanad</option>

<option>Kannur</option>

<option>Kasargod</option>

</select></td></tr>

<tr>

<th class="form-label required">Pin Code. : </th>
```

```
<td><input type="text" required maxlength="6" ng-pattern="/^[0-9]/" name="pincode" ng-model="pincode" class="form-control">

<br/>

<span style="color:red" ng-show="SaveForm.pincode.$dirty && SaveForm.pincode.$invalid" class="ng-hide">Please Enter Valid  Pin.!  </span></td>

</tr>

<tr>

<th class="form-label required">Land Mark :</th>

<td><textarea name="landmark" class="form-control" style="text-transform: uppercase;" required ng-model="landmark"></textarea><span style="color:red" ng-show="SaveForm.landmark.$dirty&&SaveForm.landmark.$invalid" class="ng-hide"> Enter Valid Landmark.! </span></td>

</tr>

<tr>

<th class="form-label required">Ration Card No. : </th>

<td><input type="text" required name="card" class="form-control" required ng-pattern="/^[0-9]/" ng-model="card"><span style="color:red" ng-show="SaveForm.card.$dirty&&SaveForm.card.$invalid" class="ng-hide">Enter Valid Ration Card Number.! </span></td>

</tr>

<tr>
```

```
<th class="form-label required">Ration Card Image : </th>

<td>

<input type="file" required name="cardimg" class="form-control" required></td>

</tr>

<tr>

<th class="form-label required">Ration Card Type : </th>

<td>

<div><input type="radio" required name="cardtype" value="AAY">Yellow Card (AAY) </div>

<div><input type="radio" name="cardtype" value="PHH">Pink Card (PHH)

</div>

<div>

<input type="radio" name="cardtype" value="NPS">Blue Card (NPS)

</div>

<input type="radio" name="cardtype" value="NPNS">White Card (NPNS)

</div>

<input type="radio" name="cardtype" value="NPI">Brown Card (NPI)

</div>

</td>
```

```
</tr>

<tr>

<th class="form-label required">

Gender :

</th>

<td>

<div><input type="radio" required name="gender" value="Male">Male</div>

<div><input type="radio" name="gender" value="Female">Female</div>

<div><input type="radio" name="gender" value="Other">Other</div>

</td>

</tr>

<tr>

<th>Person with disability</th>

<td><div><input type="radio" name="health" value="Yes" required>YES</div>

<div><input type="radio" name="health" value="No">NO</div>

</td>

</tr>

tr>
```

```
<th class="form-label required">

Contact No :

</th>
<td><input type="text" name="contact" maxlength="10" class="form-control" required ng-pattern="/^[6-9]/" ng-model="contact">
<span style="color:red" ng-show="SaveForm.contact.$dirty&&SaveForm.contact.$invalid" class="ng-hide">
Enter Valid Contact No.!
</span>
</td>
</tr>
<tr>
<th class="form-label required">
Password :</th>
<td>
                                        <input type="password" required name="password" id="pass" class="form-control"><br>
                                        <input type="checkbox" onclick="myFunction()">Show Password
                                </td>
                        </tr>
                        <tr>

                                <td colspan="2" align="center">

                                        <input type="submit" value="Register" class="btnbtn-primary"  ng-disabled="SaveForm.$invalid"/>

                                </td>
```

```
                              </tr>

                              <tr>

                                        <td colspan="2">-> Please Verify the details you have been entered.<br>-> If the site say your ration car number already existed please contact nearest ration shop or supplyco staff</td>
                              </tr>
                              <tr>
                                        <td colspan="2" align="center">
                                                  <p>Already have an account? <ahref="login.php">Login</a></p>
                                        </td>
                              </tr>
                    </center>
          </table>
          </form>
</div>
{literal}
<script>
var app = angular.module("myApp", []);
app.controller('cntrl',function($scope){


     $scope.funcSave = function()
     {
if($scope.SaveForm.$valid) {
alert("Form is Valid..!!");
          }
else

        {
```

```
alert("Form is not Valid..!!");

          }

        }

        })

</script>
{/literal}
</body>
</html>
```

Registration php

```
<?php
include_once "settings/settings.php";
include_once "classes/userclass.php";
$obj=new userclass();
if(isset($_POST['hide'])AND($_POST['hide'])=='h')
{
if (isset($_POST['name'])AND($_POST['name'])!=null)
{
if (isset($_POST['age'])AND($_POST['age'])!=null)
{
if (isset($_POST['address'])AND($_POST['address'])!=null)
{
if (isset($_POST['street'])AND($_POST['street'])!=null)
{
```

```
if (isset($_POST['post'])AND($_POST['post'])!=null)
{
if (isset($_POST['district'])AND($_POST['district'])!=null)
{
if (isset($_POST['place'])AND($_POST['place'])!=null)
{
if (isset($_POST['pincode'])AND($_POST['pincode'])!=null)
{
if (isset($_POST['landmark'])AND($_POST['landmark'])!=null)
{
if (isset($_POST['card'])AND($_POST['card'])!=null)
{
if (isset($_POST['cardtype'])AND($_POST['cardtype'])!=null)
{
if (isset($_POST['gender'])AND($_POST['gender'])!=null)
{
if (isset($_POST['health'])AND($_POST['health'])!=null)
{
if (isset($_POST['contact'])AND($_POST['contact'])!=null)
{
if (preg_match('/^([A-Za-z0-9_\.\-])+\@(([A-Za-z0-9\-])+\.)+([a-zA-Z0-9]{2,4})|([0-9]{10})*$/',
$_POST['contact']))
{
if (isset($_POST['place'])AND($_POST['place'])!=null)
{
if (isset($_POST['password'])AND($_POST['password'])!=null)
{
$name=trim($_POST['name']);
$age=trim($_POST['age']);
$address=trim($_POST['address']);
$street=trim($_POST['street']);
$upo=trim($_POST['post']);
$district=trim($_POST['district']);
$place=trim($_POST['place']);
$pincode=trim($_POST['pincode']);
$land=trim($_POST['landmark']);
$card=trim($_POST['card']);
$cardtype=trim($_POST['cardtype']);
$gender=trim($_POST['gender']);
$health=trim($_POST['health']);
$contact=trim($_POST['contact']);
$password=trim($_POST['password']);
$obj-
>userreg($name,$age,$address,$street,$upo,$district,$place,$pincode,$land,$card,$_FILES['card
img'],$cardtype,$gender,$health,$contact,$password);
}
else
echo "<script> alert('Enter password')</script>";
}
else
echo "<script> alert('Enter place')</script>";
}
}
else
echo "<script> alert('Enter phone or e-mail')</script>";
}
else
echo "<script> alert('Choose option')</script>";
}
else
```

```
echo "<script> alert('Choose your gender')</script>";
}
else
echo "<script> alert('Choose your card type')</script>";
}
else
echo "<script> alert('Enter Card No.')</script>";
}
else
echo "<script> alert('Define a land mark')</script>";
}
else
echo "<script> alert('Enter your pincode')</script>";
}
else
echo "<script> alert('Eneter your place')</script>";
}
else
echo "<script> alert('Choose your district')</script>";
}
else
echo "<script> alert('Enter your post office')</script>";
}
else
echo "<script> alert('Enter street name')</script>";
}
else
echo "<script> alert('Enter your address')</script>";
}
else
```

```
echo "<script> alert('Choose your age range') </script>";
}
else
echo "<script> alert('Enter your name')</script>";
}
$smartyObj->display('header2.tpl');
$smartyObj->display('userreg.tpl');
$smartyObj->display('footer.tpl');
?>
```

5. SYSTEM TESTING

Testing is the last stage of the software development before we release the product to the customer. Software testing is a critical element of software quality assurance and represents the ultimate review of specification, design and coding. Software testing can be looked upon as one among the many processes. Testing cannot show the absence of defects, it can only show that software defects are present.

Software Testing Techniques

The importance of testing and its impact on software cannot be underestimated. The greater visibility of software systems and the cost associated with the software failure are motivating factors for planning through testing. It is not uncommon for a software organization to spent 40% of its effort on testing.

A number of rules that act as Testing Objectives are:

Testing is a process of executing a program with the aim of finding errors.

A good test case will have a good chance to find an undiscovered error. System testing can be broadly classified into:

- Black box testing
- White box Testing
- Unit Testing
- Integration Testing

- ❖ Validation Testing

❖ Black Box Testing

Black Box Testing is not a type of testing it instead is a testing strategy, which does not need any knowledge of internal design or code etc. As the name “black box” suggests, no knowledge of internal logic or code structure is required. The types of testing under this strategy are totally based / focused on the testing for requirements and functionality of the work product/software application.

Black box testing is sometimes also called as “Opaque testing “, “Functional/Behavioral tests” and “Closed Box testing”. The base of the Black box testing strategy lies in the selection of appropriate data as per functionality and testing it against the functional specifications in order to check for normal and abnormal behavior of the system.

❖ White Box Testing

White box testing strategy deals with the internal logic and structure of the code. White box testing is also called as glass, structural, open box or clears box testing. The tests written based on the white box testing strategy incorporate coverage of the code written, branches, paths, statements and internal logic of the code etc.

In order to implement white box testing, the tester has to deal with the code and hence is needed to possess knowledge of coding and logic i.e., internal working of the code. White box test also needs the tester to look into the code and find out which unit/ statement/ chunk of the code is malfunctioning

❖ Unit Testing

The first level testing is unit testing. Unit testing concentrates on each unit if the software as implemented in source code. Initially tests focus on each module individually, ensuring that it functions properly as a unit. The modules must then be assembling or integrated to form the complete software package.

There are tests that occur as part of unit testing. The module interfaces are tested to ensure that information properly flows into and out of a program under test. The data structures are also tested for integrity. Boundary conditions are tested to ensure that the module operates properly at boundaries established to limit or restrict processing. All independent paths through the control structures are exercised to ensure that all statements in a module have been executed at least once. Finally, all error handling are tested.

❖ Integration Testing

The next level of testing is often called as Integration testing in which many tested modules are combined into sub-system, which are then tested. Data can be lost across an interface; one module can have an adverse effort on the other sub functions, when combined may not produces the desired major functions. Integrated testing is the systematic testing for constructing the uncover errors within the interface. This testing was done with sample data. The developed system has run successful for this sample data. The need for integrated test is to find the overall system performance

❖ Validation Testing

Data Validation is the process of testing the accuracy of data; a set of rules you can apply to a control to specify the type and range of data that users can enter. It can be used to display error alerts when users enter incorrect values into a form. Rather than checking for errors after a form is completed, data validation verifies values as the form is being filled out.

A strategy for software testing integrates software test case design method in to a well-planned series of steps that result in the successful construction of the software. The strategy provides a road map that describes the step to be conducted as part of testing, when these steps are planned and then undertaken, and how much effort, time and resources will be required. Therefore, any testing strategy must incorporate test planning, test case, design, test execution and resultant data collection and evaluation. A software testing strategy should be flexible enough to promote a customized testing approach. At the same time, it must be rigid enough to promote reasonable planning and management tracking as the project progress.

The project manager, software engineers and testing specialists develop a strategy for software testing.

Need For Testing

- Testing is essential as
- Existence of program defects of inadequacies is inferred.
- Test the performance of the system.
- Verifies whether the software behaves as intended by its designer.
- Checks conformance with requirement specification/user needs.
- Assesses the operational reliability of the system.
- Reflect the frequency of actual user input.

All the above testing has been done and removes all the errors.

System tests are carried out to validate dully-developed system with a view assuring that it meets its requirements. There are essentially three kinds of system testing.

Acceptance Testing

Acceptance testing is the system testing performed by the customer to determine whether or not to accept the delivery of the system. The application is tested to ensure the requirements. Different sets of input data are entered to validate the system. In all cases the system produces the reasonable output.

6. SYSTEM IMPLEMENTATION

Implementation is the process of converting a new or revised system design into operation. It is the key stage in achieving a successful new system because, usually it reveals a lot of up heal. It must therefore be carefully planned and controlled. The system will help the various government departments to co-ordinate and control the migrant in an efficient and effective manner. Once webmaster is aware of it, the system can be tested. Implementation is the stage of the project where the theoretical design is turned into working system or it is the key stage in achieving a successful new system. Therefore, it must be carefully planned and controlled. It can also be considered to be the most crucial stage in achieving a successful new system and in giving the user confidence that the new system will work and be effective.

Implementation is the final and important phase. It is the phase where theoretical design is turned into working system, which works for the user in the most effective manner. It involves careful planning, investigation of the present system and the constraints involved, user training, system testing and successful running of developed proposed system. The implementation process begins with preparing a plan for the implementation of the system. According to this plan, the activities are to be carried out, discussions made regarding the equipment and resources and the additional equipment has to be acquired to implement the new system. The user tests the developed system and changes are made according to their needs. The testing phase involves the testing of a system using various kinds of data. This method also offers the greatest security since the old system can take over if the errors are found or inability to handle certain type of transactions while using the new system.

Implementation involves following tasks:

- Careful planning
- Investigation of system and constraints
- Design of methods to achieve the changeover
- Evolution of changeover method

FUTURE ENHANCEMENT

The new programming techniques used in the design of the system provides the scope for future expansion of the project. Future enhancement can be done with this project by developers new application with respect to user taste.

- More user-friendly.
- Ease in handling and implementation.
- More security constraints could be implemented.
- Native language can be included.
- Use location.

7. SYSTEM MAINTENANCE

It is impossible to produce systems of any size which do not need to be changed. Over the lifetime of a system, its original requirements will be modified to reflect the changing user. After implementation, maintenance is the important process. Usually once the system is implemented, the software developers and customer would sign a contract. According to the time mentioned in the contract all errors and requirements would be done free of cost. Once the maintenance period is over all the logical errors will be corrected free of cost were as all extra requirements would be charged. During the contract period we would frequently visit the site where the system is implemented and check the system performance such as response time and also how it works at peak hours. If any problem is found, it is corrected.

Software development does not freeze at the moment of delivery. Usually, software must grow and change over time. These activities are collectively referred to as software maintenance. Application updates are part of normal maintenance phase of development life cycle. A modification effort is actually a small project and must proceed through all the phases of development process.

Ease of maintenance is a part of every step-in development. If the analysis is complete, users will find the most important features in the first release of software. If the design and coding is done perfectly, then it will be very easy to maintain later. Some of the suggestions are group the changes and deliver another release rather than incremental changes, so that it will force to be more through about new researches. Give more to small requirements.

The experience in coding will be an added advantage because integrating a new code with existing code is normally difficult. The working of the system is observed in a local machine and intranet and it was found satisfactory.

The four types of maintenance activities are listed below:

Corrective Maintenance

This is concerned with fixing reported errors in the software. Coding errors are cheap to correct; design errors are more expensive as they may involve the rewriting of sever program components. Requirements errors are the most expensive to repair because of the extensive system redesign which may be necessary.

Adaptive Maintenance

This means changing the software to some new environment such as different hardware platform or for use with different operating system. The software functionality does not radically change. Any system that involves JVM can run this software.

Perfective Maintenance

This involves implementing a new functional or nonfunctional system requirement.

These are generated by software user as their organization or business changes.

Preventive Maintenance

This occurs when software is changed to improve future maintainability or reliability or to provide a better basis for future enhancements.

In the current project, all the above maintenance was implemented.

8. SYSTEM SECURITY

This software produces some security features. Before registering user's details, they have to authenticate by providing the appropriate Username and password. Securities are provided in the software so that the data remains confidential. Password facility is implemented to avoid unauthorized access. For providing more security they have to change their password with their wishes.

Security testing techniques occur for vulnerabilities or security holes in applications. These vulnerabilities leave applications open to exploitation. Ideally, security testing is implemented throughout the entire software development lifecycle (SDLC) so that vulnerabilities may be addressed in a timely and thorough manner. Unfortunately, testing is often conducted as an afterthought at the end of the development cycle.

9. UPGRADABLE POSSIBILITIES

The technology is changing day to day. The efficiency of developed system can be improved by modifications. The quality of the product can be improved by keeping wise list of reports and other documents effectively. So, it is easy to add or remove all modules.

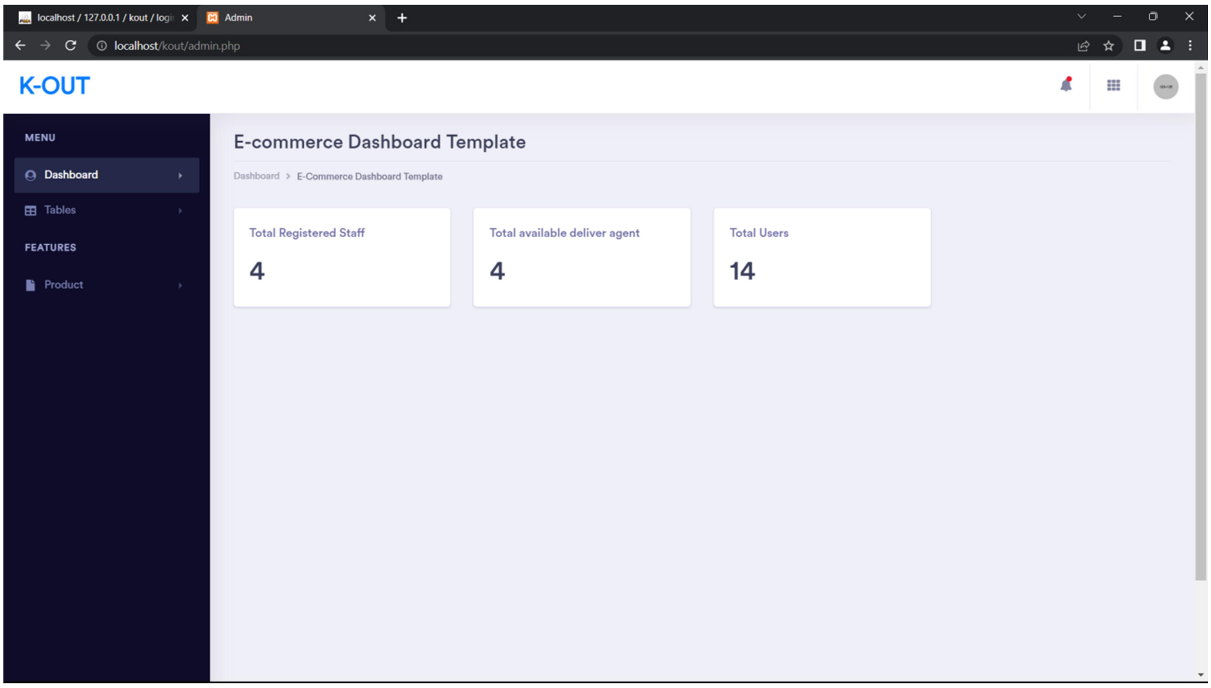
The system gives only the current situation in the organization. But in future it is possible to make changes in a system to a complete record keeping, which will give all the past and current information of the organization. Software development in PHP is very flexible and all application was tested with live data and has proved respond successful. So, it is quite and helps in smooth migration from manual to computerized system.

- Increased reliability and integrity of data.
- User friendly and flexible in all aspects.
- Data entry and updating is quite easy.
- Effective table manipulation as facilitated by the rich MySQL.
- Good validation checking.
- Facilitates registration canceling.
- Reduce complexity in data entering.
- Easy maintenance is providing.
- Now the system is only in English language, it can be changed into the regional language for the user convince.
- Now this software is pc-oriented software, in future it can be changed to mobile application for the ease of use and availability.

10. APPENDIX

APPENDIX 1: FORM LAYOUT

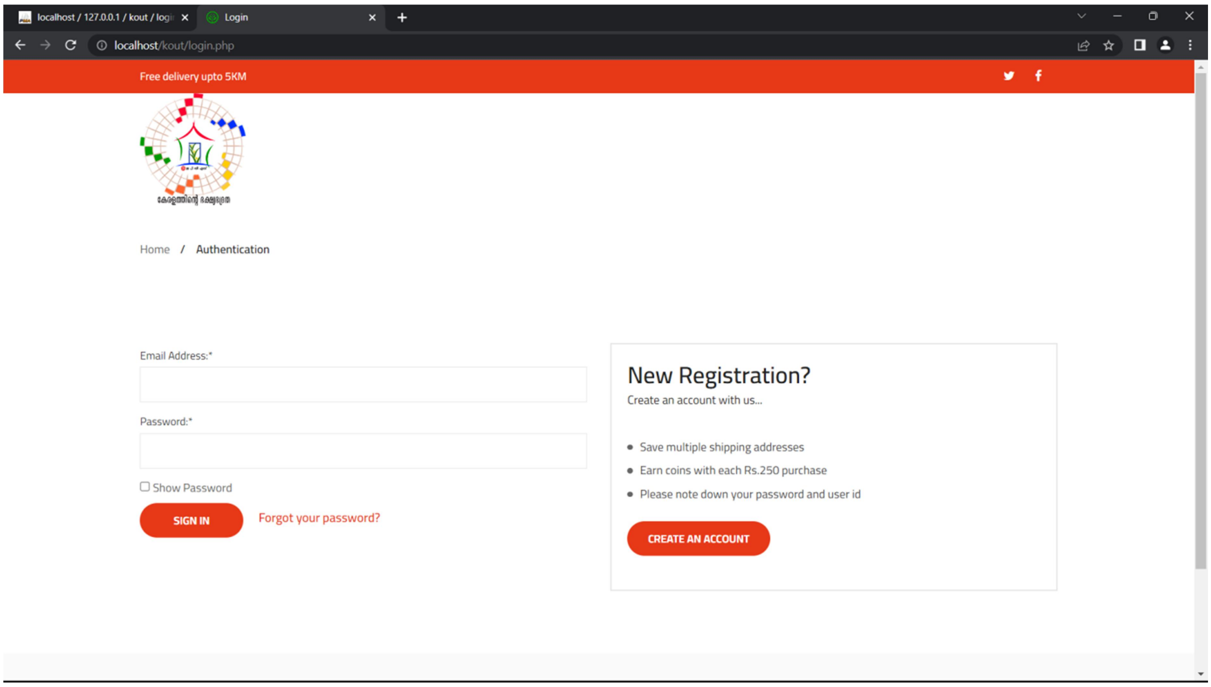
ADMIN



localhost / 127.0.0.1 / kout / logi
Admin
localhost/kout/admin.php
K-OUT
MENU
Dashboard
Tables
FEATURES
Product
E-commerce Dashboard Template
Dashboard > E-Commerce Dashboard Template

Total Registered Staff	Total available deliver agent	Total Users
4	4	14

LOGIN PAGE



Free delivery upto 5KM

Home / Authentication

Email Address:*

Password:*

□ Show Password

SIGN IN

Forgot your password?

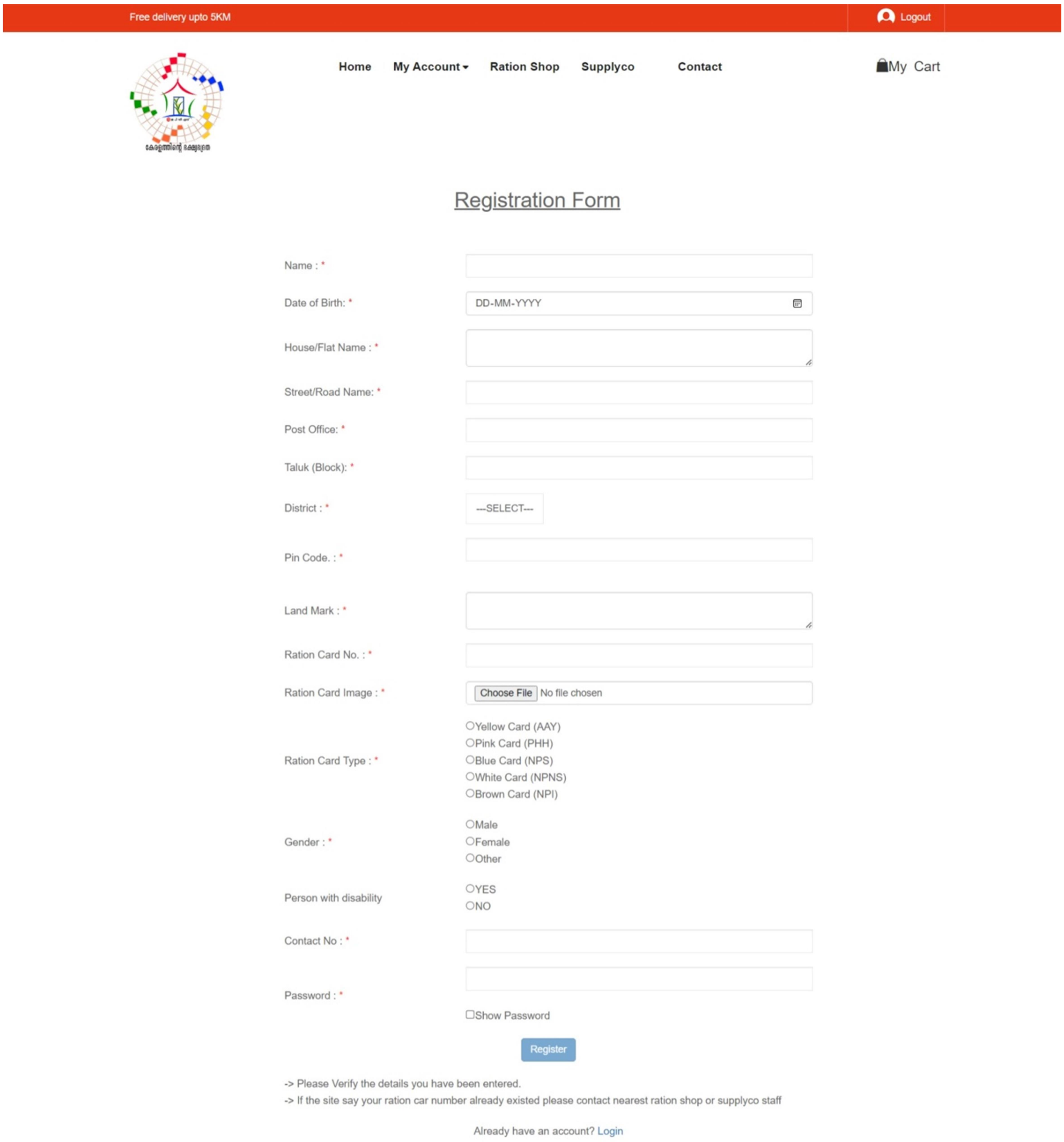
New Registration?

Create an account with us...

- Save multiple shipping addresses
- Earn coins with each Rs.250 purchase
- Please note down your password and user id

CREATE AN ACCOUNT

USER REGISTRATION



Registration Form

Name : *

Date of Birth: * DD-MM-YYYY

House/Flat Name : *

Street/Road Name: *

Post Office: *

Taluk (Block): *

District : * ---SELECT---

Pin Code. : *

Land Mark : *

Ration Card No. : *

Ration Card Image : * Choose File No file chosen

Ration Card Type : *
○Yellow Card (AAY)
○Pink Card (PHH)
○Blue Card (NPS)
○White Card (NPNS)
○Brown Card (NPI)

Gender : *
○Male
○Female
○Other

Person with disability
○YES
○NO

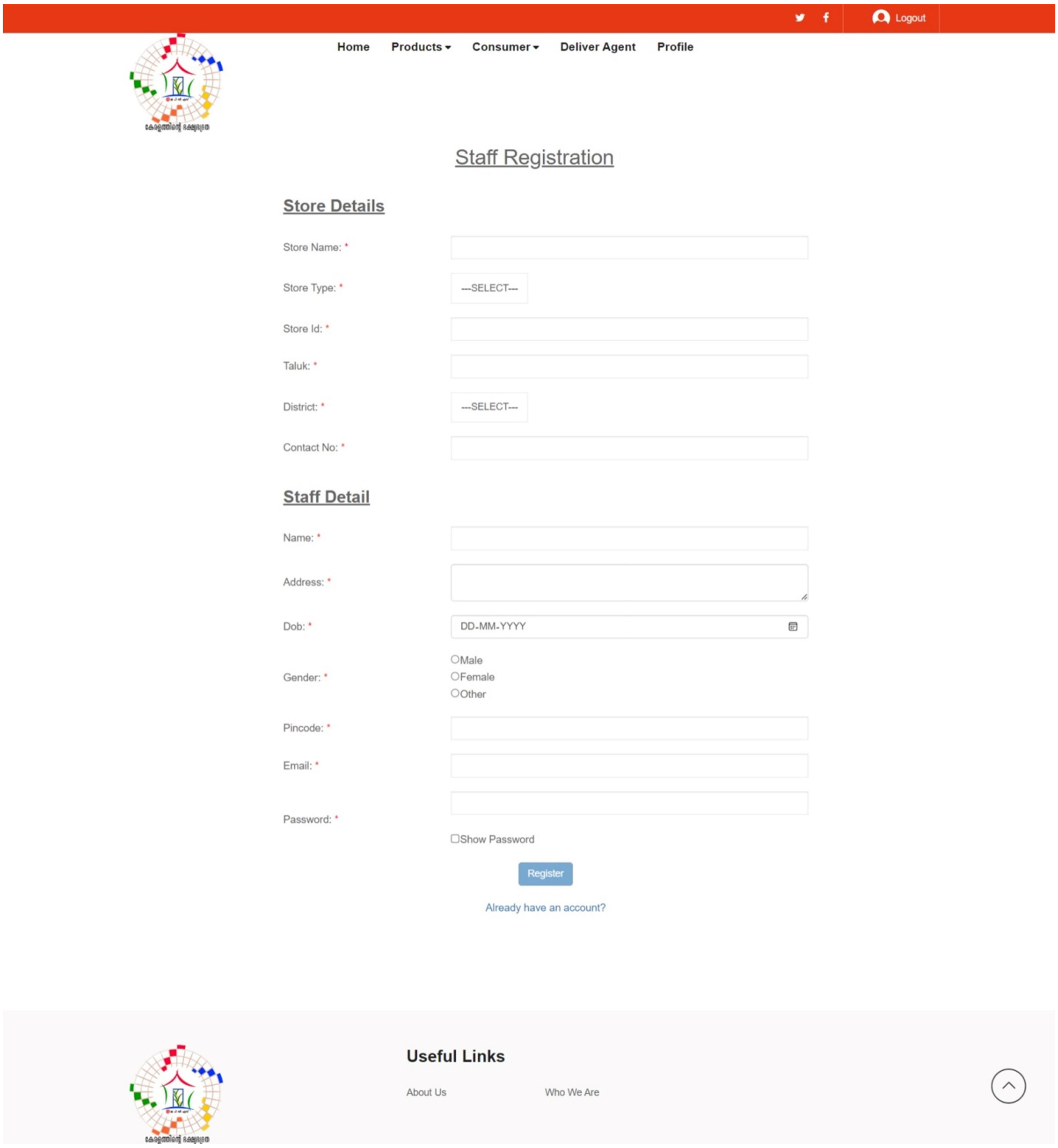
Contact No : *

Password : *
□Show Password

-> Please Verify the details you have been entered.
-> If the site say your ration car number already existed please contact nearest ration shop or supplyco staff

Already have an account? Login

STAFF REGISTRATION



Staff Registration

Store Details

Store Name: *	
Store Type: *	---SELECT---
Store Id: *	
Taluk: *	
District: *	---SELECT---
Contact No: *	

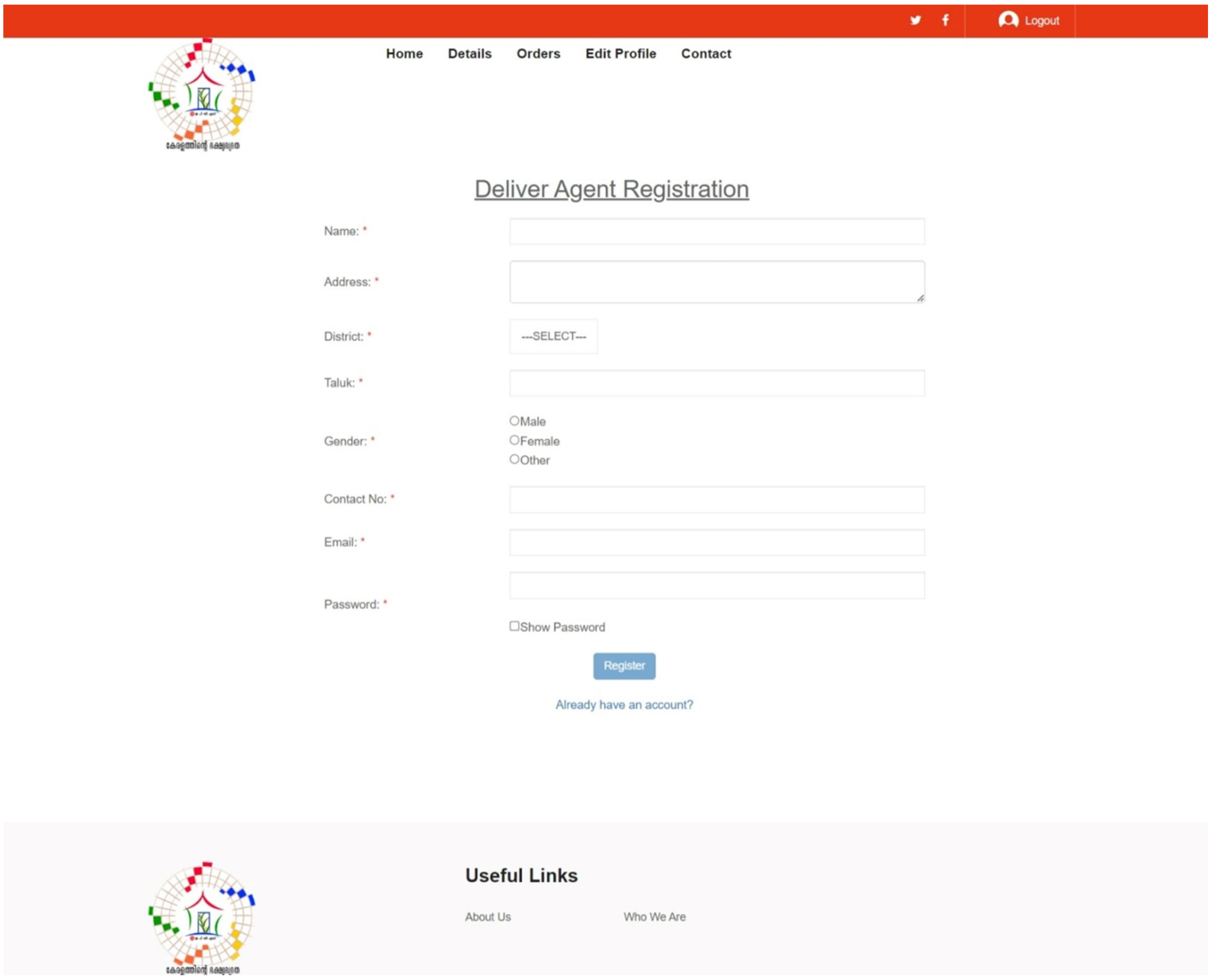
Staff Detail

Name: *	
Address: *	
Dob: *	DD-MM-YYYY
Gender: *	○Male ○Female ○Other
Pincode: *	
Email: *	
Password: *	□Show Password

Register

Already have an account?

DELIVER AGENT REGISTRATION



Deliver Agent Registration

Name: *

Address: *

District: * ---SELECT---

Taluk: *

Gender: * ○Male ○Female ○Other

Contact No: *

Email: *

Password: *

□Show Password

APPENDIX 2: MEETING MINUTES

Group Members

ABHIRAM MOHAN	**(Reg No:32020955001)**
AL AMEEN N	**(Reg No:32020955005)**
ASWIN S BIJU	**(Reg No:32020955006)**

MINUTES

Date	**:**	**22/12/2022**
Time	**:**	**10 am**
Location	**:**	**AYOOR**

We decided to do a project named "**K-OUT (Kerala Outlet)**", and we chose HTML, CSS,JS are the front end and my SQL as the back end. System study and analysis are also done in between 22/12/2022 to 31/12/2022.

MINUTES

Date	**:**	**01/1/2023**
Time	**:**	**10 am**
Location	**:**	**AYOOR**

From 01/01/2023 to 06/01/2023 we started the design process, forms and data structures.

MINUTES

Date	**:**	**07/01/2023**
Time	**:**	**10 am**
Location	**:**	**AAYOOR**

The design process is over then we started coding from 07/01/2023 to 28/01/2023. Along with coding we did the documentation also.

MINUTES

Date	**:**	**15/02/2023**
Time	**:**	**10 am**
Location	**:**	**AYOOR**

From 15/02/2023 to 25/03/2023 about 90% of coding is completed, is started testing process. It can take a long time to complete the process.

MINUTES

Date	**:**	**05/04/2023**
Time	**:**	**10 am**
Location	**:**	**AYOOR**

We successfully completed our project and the documentation on 05/04/2023.

APPENDIX 3: GANTT CHART

Task	Start(dd-mm-yyyy)	End(dd-mm-yyyy)
Initial investigation	22-12-2022	24-12-2022
Study of existing system	25-12-2022	31-12-2022
Design	01-01-2023	06-01-2023
Coding	07-01-2023	28-01-2023
Testing	15-02-2023	25-03-2023
Implementation	26-03-2023	28-03-2023
Documentation	28-03-2023	05-04-2023

Dec:22-24	Dec 25-31	Jan 01-06	Jan 07-28	Feb 15-Mar 25	Mar 26- 28	Apr 28-Apr 05
Initial investigation →						
	Study of existing system →					
		Design →				
			Coding →			
				Testing →		
					Implementation →	
						Documentation →

11. CONCLUSION

The new system has overcome most of the limitations of the existing system and work according to the design specification given. The developed systems dispense the problem and meet the needs of by providing reliable and comprehensive information. All the requirements projected by the user have been meet by the system.

The newly developed system consumes less processing time and all the details are update and processed immediately. Since the screen provides online help messages and is very user-friendly, any user will get familiarized with its usage. Modules are designed to be highly flexible so that any failure requirements can be easily added to the modules without facing man problems.

12. BIBILIOGRAPHY

Text Book Reference:

1) Fundamentals of Software Engineering (2nd, 3rd,4thedition) - RajibMall
2) An Integrated Approach to Software Engineering(3rd) - PankajJalote
3) Software Engineering (A Practioner's Approach, 6th edition) -Roger.S.Pressman
4) PHP5andMySQL - Tim Converse and Joyce Park with Clark Morgan
5) Linux apache web server administration(second edition) - Charles Auld
6) Programming PHP - Rasmus Lerdorf, Kevin Tatroe, and PeterMacintyre
7) SQL server 2005 - Sikha Saba Bagui and Richard WalshEarp
8) Software Engineering - K.L James
9) Database System Concept - Abraham Suberschrtz
10) PHP, MYSQL and Apache all in one-Julie cmelony

Web Reference:

- http://www.php.net
- http://www.Mysql.org
- http://www.apachefriends.org
- www.softzanesolutions.i