CHAPTER 2

EXISTING SYSTEM

2.1 Introduction

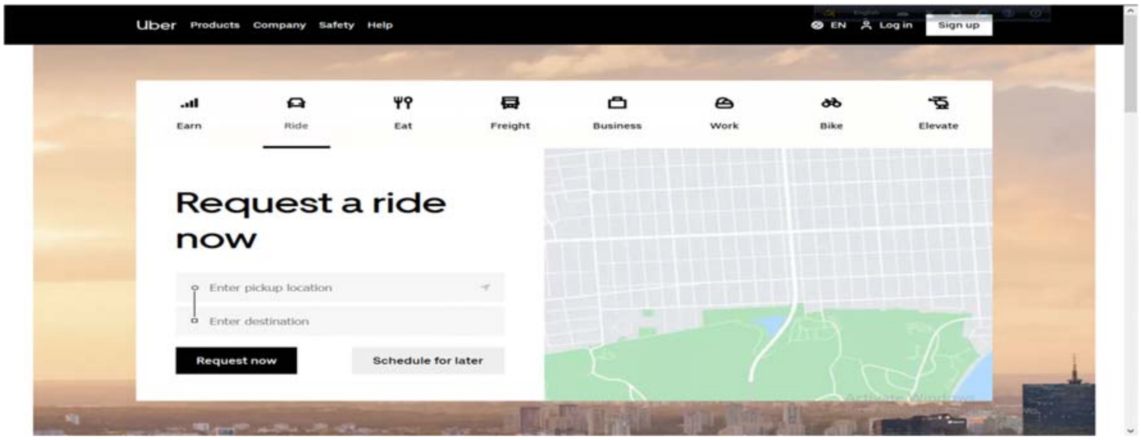
In this chapter we discuss the existing systems of 'Bachao' that able to clear our intention to develop our ambulance and medicine service system with saving time, energy. Manually, it is very energetic and time-consuming process. We aim to address this lacking by providing online base application and provide the best opportunity of the admin and the users. Bachao System also provide for the users where they have an ambulance and medicine for them. The outcome of this system also offers solution to overcome very energetic and time consuming that may occur in the manual Ambulance and Medicine Service System. The chapter also contains problem of existing system, supporting theory, analysis of existing system. The explanation of many diagram such as Entity Relationship Diagram (ERD), Data Flow Diagram (DFD) , Use Case Diagram is a part of this chapter. We also describe the manual system and their problem in this chapter. PHP, HTML, CSS, jQuery, JavaScript, MySQL are also described in this chapter.

2.2 Existing Systems

There are many ambulance and medicine service systems, who use manually their process of the clients and some are online based. Manual scheduling can be a burden for many establishments, as it requires significant time and staff resources to manage properly. After the long process, finally they decide for a deal and they reached a fare and the hand over The related works and the theory/literature are detailed below:

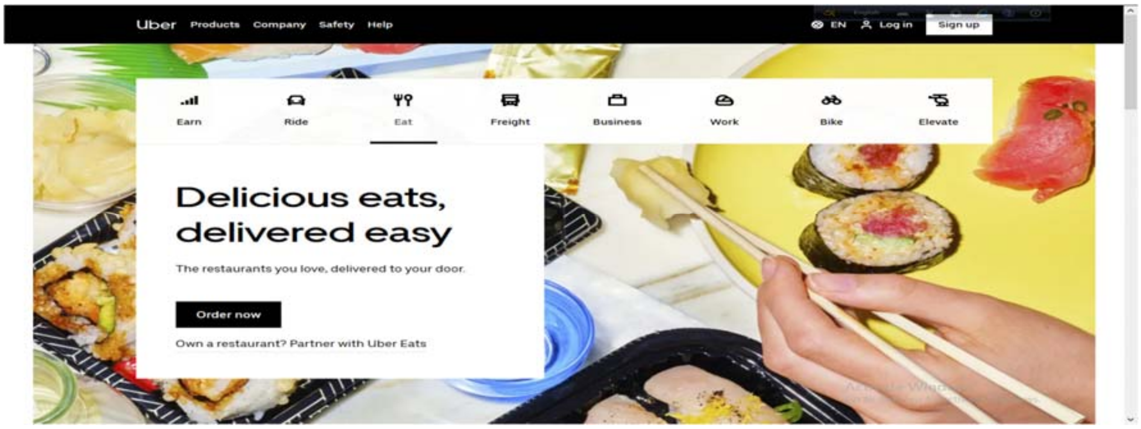
2.2.1 Uber

Uber is a platform where those who drive and deliver can connect with riders, eaters, and restaurants. In cities where Uber is available, you can use the Uber app to request a ride. When a nearby driver accepts your request, the app displays an estimated time of arrival for the driver heading to your pickup location.



Request a ride now

Figure 2.1: Uber home page



Delicious eats, delivered easy

The restaurants you love, delivered to your door.

Own a restaurant? Partner with Uber Eats

Figure 2.1: Uber home page

2.2.1.1 Features

1. Book a ride for now or later

The “how to book” is an important consideration when including this feature. You should have the following input to book the ride

i. The pick-up location
ii. The drop location
iii. The type of vehicle the user wants to select
iv. The time and date, in the case of the book later feature

2. Real-time tracking of driver

3. Allow others to follow ride in real-time

4. Multiple payment options

5. Preferred driver

6. Add multiple drop-off points

7. Driver review and rating

8. Trip history

9. In-app chat or call option

10. Save destinations

2.2.1.2 Advantages

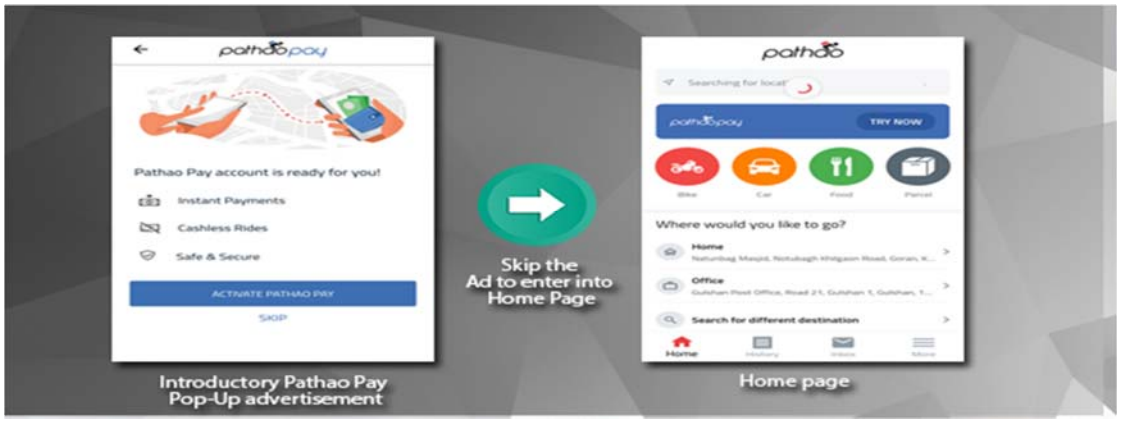
i. Convenient and Cashless
ii. Professional Service
iii. Competitive Pricing
Generally, Uber is less expensive than traditional taxis and car services.
iv. Safer and More Flexible for Drivers

2.2.1.3 Dis-advantages

i. Surge Pricing
ii. Trip Cancellations
iii. Safety Concerns
iv. Low Fares Hurt Drivers
v. Negative Impact of Price compitition

2.2.2 Pathao

Pathao follows a Super App model, providing all of its services through one app.[14] They update their app frequently to make the overall app navigation more straightforward and more natural for all users. Pathao currently provides ride-sharing, parcel, food delivery and On-demand Transport Sharing services in 3 major cities of Bangladesh, Dhaka, Chittagong and Sylhet and in Kathmandu of Nepal. Its food delivery services are currently available in Dhaka Metropolitan Regions and Chittagong Metro areas.

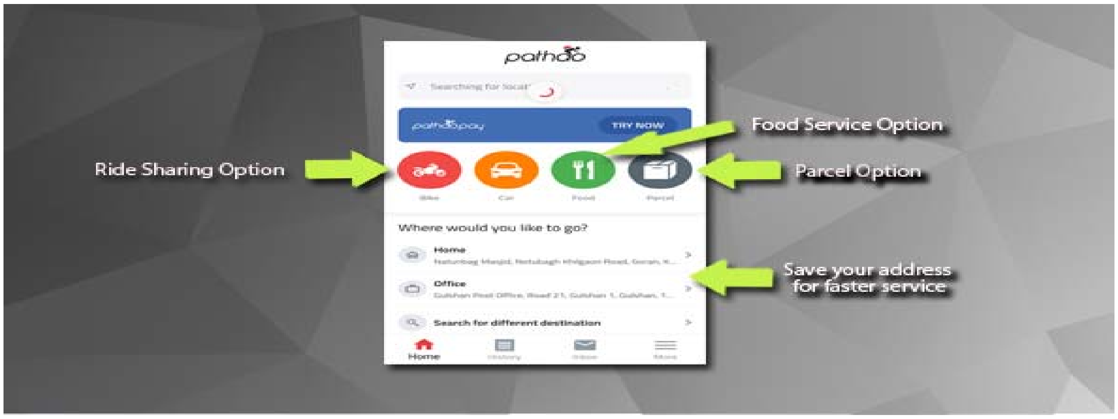


Skip the Ad to enter into Home Page

Introductory Pathao Pay Pop-Up advertisement

Home page

Figure 2.2.2: Pathao home page



Ride Sharing Option

Food Service Option

Parcel Option

Save your address for faster service

Figure 2.2.2: Pathao home page

2.2.2.1 Features

i. Insurance Coverage

ii. Rapid Response Team

iii. 2 Way Rating System

iv. Live Location Share

v. Call Centre Support

2.2.2.2 Advantages

i. Earn points with every eligible ride and food order

ii. Access to exclusive deals and discounts availability

iii. Special offers on restaurants, flights, hotels

iv. Priority Support

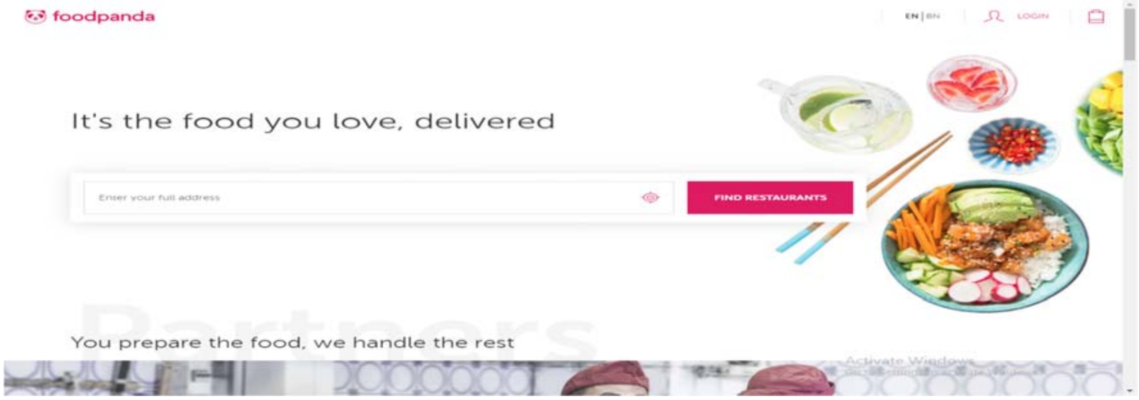
v. Special access to highly rated drivers*
vi. Free deliveries on select food orders
vii. Premium Support Hotline
viii. Special rates for your two favourite destinations*

2.2.2.3 Dis-advantages

i. Customers allege that some drivers are charging extra fares, taking detours, clocking rides before pick up and refusing to take them to their destinations.
ii. Drug addicts have begun taking work as drivers.
iii. At the end of the ride, the driver demanded a higher fare, saying the app sometimes shows incorrect readings.
iv. Sometimes driver didn’t follow the rout.
v. After the trip, they demanded a tip.

2.2.3 Foodpanda

foodpanda (in Asia and Europe) and hellofood (in Middle East), is a global online food delivery marketplace headquartered in Berlin, Germany. The service allows users to select from local restaurants and place orders via the website or mobile application. The foodpanda group currently employs more than 3,000 people worldwide, and continues to grow at an impressive speed. In June 2014, foodpanda, together with its affiliated brand hellofood, won the European E-Commerce Award as the best E-Commerce start-up in Europe.It is part of the DELIVERY HERO GROUP, which is head quartered in Berlin, Germany. The DELIVERY HERO GROUP is currently processing over 20 million orders per month across more than 50 countries and experiencing phenomenal growth.foodpanda Bangladesh is the leading online food delivery marketplace in Bangladesh. It is partnered with hundreds of restaurants and offers customers the most convenient food ordering and delivery experience in Bangladesh. It is operational in Dhaka, Chittagong and Sylhet.



It's the food you love, delivered

You prepare the food, we handle the rest

Figure 2.2.3 : Foodpanda home page

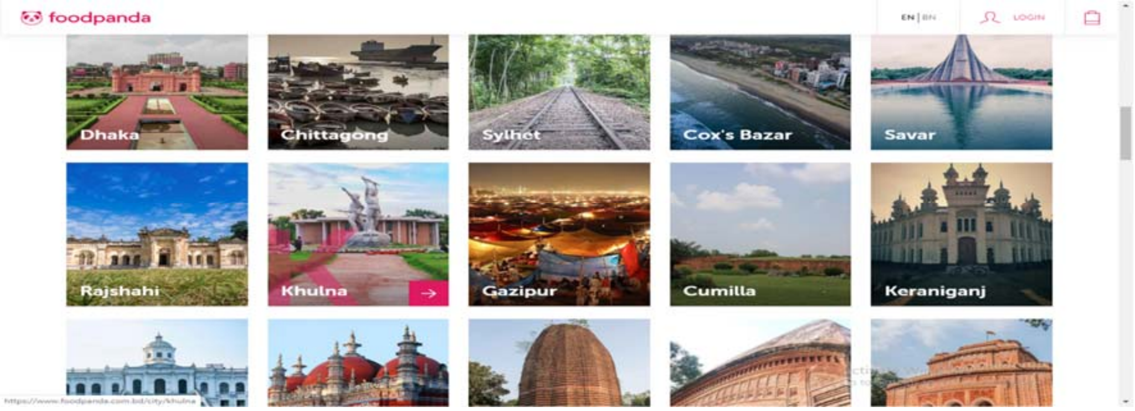


Figure 2.2.3 : Foodpanda home page

2.2.3.1 Features

i. Permits pre-orders several hours before requested delivery
ii. Displays the live location of the delivery driver
iii. In India Only) Works in collaboration with the railway's e-catering services
iv. Contact the rider directly

This new feature allows consumers to start a conversation with the riders directly.

Now instead of calling restaurants and foodpanda directly, customers can chat with the rider directly if they'd like to drop more instructions on their location or to even inquire about late deliveries.

v. Track food in real time

From pick up to delivery points, all food orders can now be tracked in real time through the app.

Once the order is confirmed, the 'contact your rider' tab allows to start a conversation with your rider.

From the restaurant to your front door, now it's easier to know where your food is at all times.

2.2.3.2 Advantages

i. Works within a restaurant's current delivery/pickup structure
ii. Allows user reviews
iii. Curates the list of available restaurants

2.2.3.3 Dis-advantages

i. Does not offer many delivery options (depending on the location)
ii. Works in only 22 countries, most of which are in Eastern Europe.
iii. Has difficulty locating a user based on a typed-in address

2.3 Supporting Theory

Our whole system is web based. We have implemented our system by the following supporting various web technology and tools for both frontend and backend.

2.3.1 HTML

HTML is a computer language devised to allow website creation. These websites can then be viewed by anyone else connected to the Internet. It is relatively easy to learn, with the basics being accessible to most people in one sitting; and quite powerful in what it allows you to create. It is constantly undergoing revision and evolution to meet the demands and requirements of the growing Internet audience under the direction of the » W3C, the organization charged with designing and maintaining the language. HTML consists of a series of short codes typed into a text-file by the site author — these are the tags. The text is then saved as an html file, and viewed through a browser, like Internet Explorer or Netscape Navigator. This browser reads the file and translates the text into a visible form, hopefully rendering the page as the author had intended. Writing your own HTML entails using tags correctly to create your vision. You can use anything from a rudimentary text-editor to a powerful graphical editor to create HTML pages. The tags are what separate normal text from HTML code. You might know them as the words between the <angle-brackets>. They allow all the cool stuff like images and tables and stuff, just by telling your browser what to render on the page. Different tags will perform different functions. The tags themselves don’t appear when you view your page through a browser, but their effects do. The simplest tags do nothing more than apply formatting to some text, like this: <b>these words will be bold</b>, and these will not. In the example above, the <b> tags were wrapped around some text, and their effect will be that the contained text will be bolded when viewed through an ordinary

web browser. If you want to see a list of a load of tags to see what’s ahead of you, look at this tag reference. Learning the tags themselves is dealt with in the next section of this website.HTML stands for Hyper Text Markup Language. It can be thought of as a programming language that is used to place text, images and other contents on a webpage. It is the foundation of almost any page you visit on your browser. Although not a true programming language (it doesn't process or manipulate data, it is only a language that defines layouts) , it's a great place to start if you want to get involved with computers, primarily because it's exciting and you can see the results of your learning almost instantly. It is where I started.

2.3.1 Basic Form of HTML Looks Like

The example of HTML is given below.

```
<DOCTYPE!>

<html>

<head>

<title> Page title </title>

</head>

<body>

<h1>This is a Heading</h1>

<p>This is a paragraph</p>

</body>

</html>
```

2.3.2 CSS

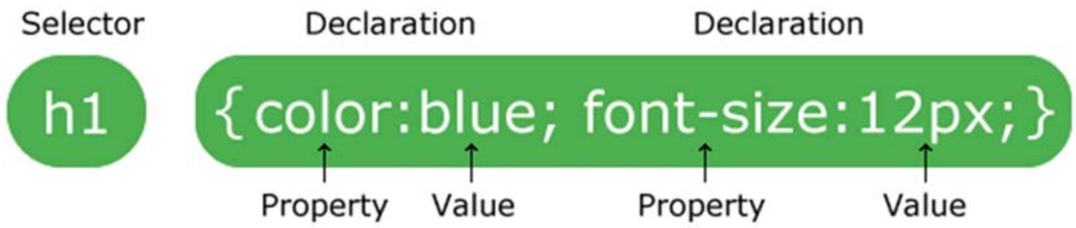
CSS is the language for describing the presentation of Web pages, including colors, layout, and fonts. It allows one to adapt the presentation to different types of devices, such as large screens, small screens, or printers. CSS is independent of HTML and can be used with any XML-based markup language.CSS (Cascading Style Sheets) allows you to create great looking web pages, but how does it work under the hood? This article explains what CSS is, how the browser turns HTML into a Document Object Model (DOM), how CSS is applied to parts of the DOM, some very basic syntax examples, and what code is used to actually include our CSS in our web page. As we have mentioned before, CSS is a language for specifying how documents are presented to users — how they are styled, laid out etc. A document is usually a text file structured using a markup language — HTML is the most common markup language, but you will also come across other markup languages such as SVG or XML. Presenting a document to a user means converting it into a usable form for your audience. Browser, like Firefox, Chrome or Internet Explorer, is designed to present documents visually, for example, on a computer screen, projector or printer. CSS can style almost any HTML tag that creates a visible element on the page, including all the HTML tags used to create headings, paragraphs, links, images, lists, and tables list. Specifically, CSS allows you to style.

i. Text size, color, style, typeface, and alignment.
ii. Link color and style
iii. Image size and alignment
iv. List bullet styles and indentation
v. Table size, shading, borders, and alignment

2.3.2.1 CSS Syntax

A CSS rule set consists of a selector and a declaration

block



The selector points to the HTML element you want to style.

The declaration block contains one or more declarations separated by semicolons.

Each declaration includes a CSS property name and a value, separated by a colon.

A CSS declaration always ends with a semicolon, and declaration blocks are surrounded by curly braces.

2.3.2.2 CSS Example

In this example all <p> elements will be center-aligned, with a red text

color:

```
 p{ color: red;
 text-align:
center; }
```

2.3.3 PHP

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. PHP is mainly focused on server-side scripting, so you can do anything any other CGI program can do, such as collect form data, generate dynamic page content, or send and receive cookies. You can access the PHP program output with a web browser, viewing the PHP page through the server. PHP is a language that creates web pages, but exactly how does it do this? If you know some HTML, you might be wondering what PHP have to do with it and how it integrates into a web page. You can think of PHP as a general-purpose computing language if you want to, but it was designed with one task in mind and it is almost exclusively used for that task - generating web pages. So, while it might be more flattering to PHP to introduce it in the widest possible context this would be misleading and it would make the job of learning how to use it harder than it needs to be. So, let's say the obvious to make it 100% clear. PHP is a language that creates web pages. What this means in practice is that a PHP program's objective in life is to generate HTML or JavaScript or anything else that you might find in a web page. In most cases and certainly when you are first learning PHP the web technology that is used is HTML. The output of a typical PHP program is HTML this means that to make any sense of PHP you also have to know about the web technology that the program is generating, and in particular HTML. In practice, this should not be a huge problem because HTML is not difficult and mostly the way that PHP makes use of it is simple. However, it is important to know that it is possible that you could have a problem with understanding a PHP program simply because you cannot understand the HTML it is generating.

2.3.3.1 PHP Work Flow

PHP framework to build Workflow Management system. There are many frameworks through which you can generate a graph of your existing flow. So, it is basically interested in defining a workflow rather than generating the graph for existing framework. And it should be an open source framework (basically PHP).

The following figure is given below,

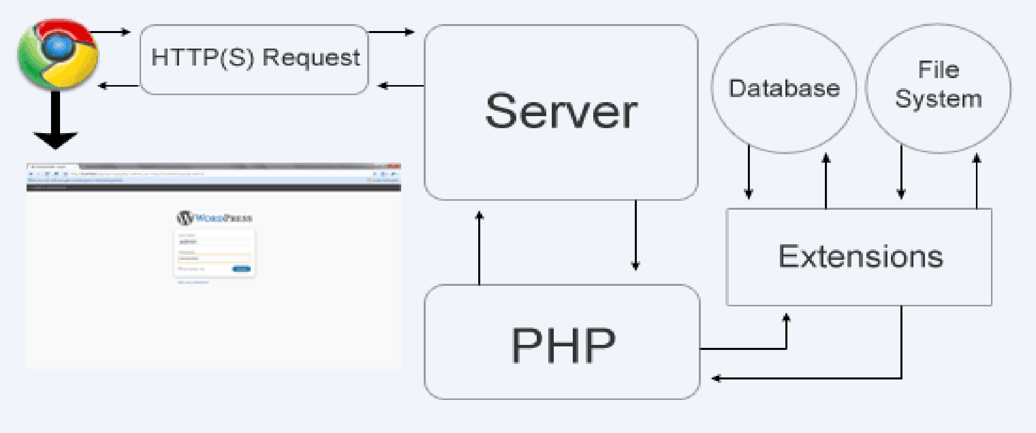


Figure:2.4 Php workflow

2.3.4 MySQL

MySQL is an open source relational database management system (RDBMS) based on Structured Query Language (SQL). MySQL runs on virtually all platforms, including Linux, UNIX, and Windows. Although it can be used in a wide range of applications, MySQL is most often associated with web-based applications and online publishing and is an important component of an open source enterprise stack called LAMP. LAMP is a Web development platform that uses Linux as the operating system, Apache as the Web server and MySQL as the relational database management system and PHP as the object-oriented scripting language. (Sometimes Perl or Python is used instead of PHP.) MySQL, the most popular Open Source SQL database management system, is developed, distributed, and supported by Oracle Corporation. MySQL is a freely available open source Relational Database Management System (RDBMS) that uses Structured Query Language (SQL). SQL is the most popular language for adding, accessing and managing content in a database. It is most noted for its quick processing, proven reliability, ease and flexibility of use. MySQL is an essential part of almost every open source PHP application. Good examples for PHP & MySQL-based scripts are WordPress, Joomla, Magneto and Drupal. One of the most important things about using MySQL is to have a MySQL specialized host. Here are some of the things Site Ground can offer: We have long experience in providing technical support for MySQL-based web sites. Thanks to it our servers are perfectly optimized to offer the best overall performance for most MySQL applications. We offer a lot of free MySQL tools including CMS systems, forums, galleries, blogs, shopping carts and more. We support MySQL 5 and we provide unlimited MySQL databases on all our hosting plans.

2.3.4.1 Reasons of use MySQL

The following reasons of using MYSQL are given below

i. Scalability
ii. High Performance
iii. High Availability
iv. Robust Transactional Support
v. Web and data Warehouse Strengths
vi. Strong data protection
vii. Management Ease
viii. Very fast, reliable, and easy to use
ix. ix. Ideal for both small and large application.

2.3.5 XAMPP

Xampp stands for Cross-platform (x), Apache (a), Maria BD (M), PHP (P) and Perl (P). It is a simple lightweight Apache distribution that makes it extremely easy for developers to create a local web server for testing purpose. Everything needs to set up a web server –server application (Apache), database (MySQL), and scripting language (PHP)-is included in a simple extractable file. XAMPP is also cross-platform, which means it works equally well on Linux, Mac and Windows.

The following are the XAMPP included tools:

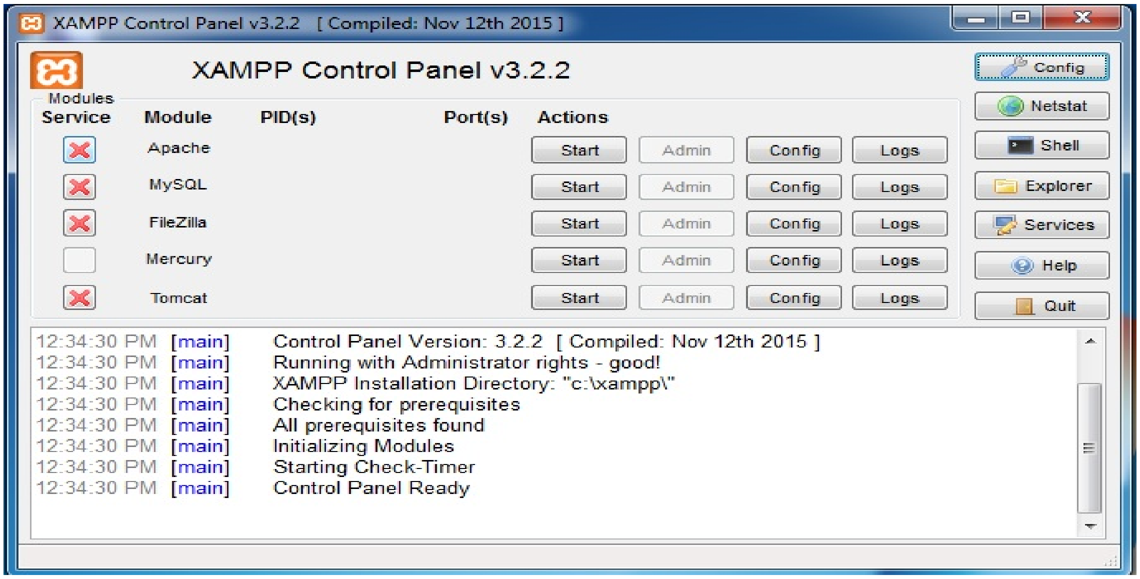
i. Apache
ii. MySQL
iii. PHP+PEAR
iv. Perl v. OpenSSL
v. phpMyAdmin
vi. Xampp Control Panel
vii. Webalizer
viii. SQLite
ix. ADODB
x. Zend Optimizer
xi. Xampp Security
xii. Tomcat

xiii. FileZilla FTP Server

2.3.5.1 Xampp Control Panel

XAMPP is a software distribution which provides the Apache web server, MySQL database (actually MariaDB), Php and Perl (as command-line executables and Apache modules) all in one package. It is available for Windows, MAC and Linux systems. No configuration is necessary to integrate Php with MySQL.

Here is the screenshot of xampp control panel -



```
12:34:30 PM  [main]   Control Panel Version: 3.2.2  [ Compiled: Nov 12th 2015 ]
12:34:30 PM  [main]   Running with Administrator rights - good!
12:34:30 PM  [main]   XAMPP Installation Directory: "c:\xampp\"
12:34:30 PM  [main]   Checking for prerequisites
12:34:30 PM  [main]   All prerequisites found
12:34:30 PM  [main]   Initializing Modules
12:34:30 PM  [main]   Starting Check-Timer
12:34:30 PM  [main]   Control Panel Ready
```

Figure :2.5 Xampp control panel

2.5.3.2 Entity-Relationship Diagram (ERD)

An Entity Relationship (ER) Diagram is a type of flowchart that illustrates how “entities” such as people, objects or concepts relate to each other within a system. ER Diagrams are most often used to design or debug relational databases in the fields of software engineering, business information systems, education and research. Also known as ERDs or ER Models, they use a defined set of symbols such as rectangles, diamonds, ovals and connecting lines to depict the interconnectedness

of entities, relationships and their attributes. They mirror grammatical structure, with entities as nouns and relationships as verbs.

2.3.5.3 Uses of Entity Relationship Diagrams

i. **Database design:** ER diagrams are used to model and design relational databases, in terms of logic and business rules (in a logical data model) and in terms of the specific technology to be implemented (in a physical data model.) In software engineering, an ER diagram is often an initial step in determining requirements for an information systems project. It's also later used to model a particular database or databases. A relational database has an equivalent relational table and can potentially be expressed that way as needed.

ii. **Database troubleshooting:** ER diagrams are used to analyze existing databases to find and resolve problems in logic or deployment. Drawing the diagram should reveal where it's going wrong.

iii. **Business information systems:** The diagrams are used to design or analyze relational databases used in business processes. Any business process that uses fielded data involving entities, actions and interplay can potentially benefit from a relational database. It can streamline processes, uncover information more easily and improve results.

iv. **Business process re-engineering (BPR):** ER diagrams help in analyzing databases used in business process re-engineering and in modeling a new database setup.

v. **Education:** Databases are today's method of storing relational information for educational purposes and later retrieval, so ER Diagrams can be valuable in planning those data structures.

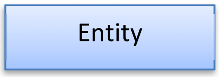
vi. **Research:** Since so much research focuses on structured data, ER diagrams can play a key role in setting up useful databases to analyze the data.

2.3.5.4 The Components and Features of an ER Diagram

ER Diagrams are composed of entities, relationships and attributes. They also depict cardinality, which defines relationships in terms of numbers. Here’s a glossary:

i. Entity

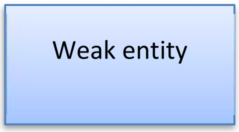
A definable thing such as a person, object, concept or event that can have data stored about it. Think of entities as nouns. Examples: a user, client, car or product. Typically shown as a rectangle.

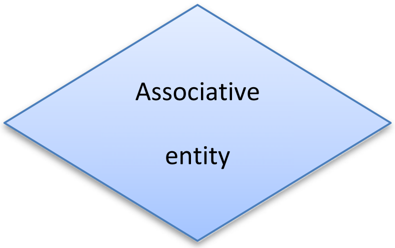


a. **Entity type:** A group of definable things, such as users, whereas the entity would be the specific user. Other examples: customers, cars or products.

b. **Entity set:** Same as an entity type, but defined at a particular point in time, such as users visited it in the first day. Other examples: Customers who purchased last month, cars currently registered in Florida. A related term is instance, in which the specific person or car would be an instance of the entity set.

c. **Entity categories:** Entities are categorized as strong, weak or associative. A strong entity can be defined solely by its own attributes, while a weak entity cannot. An associative entity associates entities (or elements) within an entity set.





d. **Entity keys:** Refers to an attribute that uniquely defines an entity in an entity set. Entity keys can be super, candidate or primary.

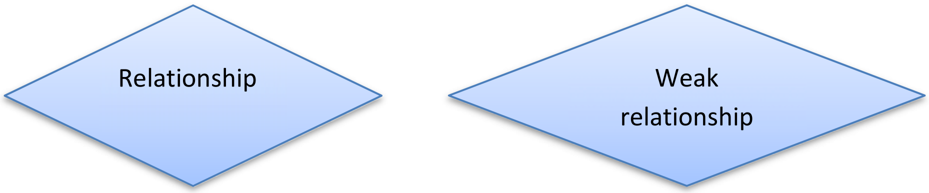
e. **Super key:** A set of attributes (one or more) that together define an entity in an entity set.

f. **Candidate key:** A minimal super key, meaning it has the least possible number of attributes to still be a super key. An entity set may have more than one candidate key.

g. **Primary key:** A candidate key chosen by the database designer to uniquely identify the entity set. Foreign key identifies the relationship between entities.

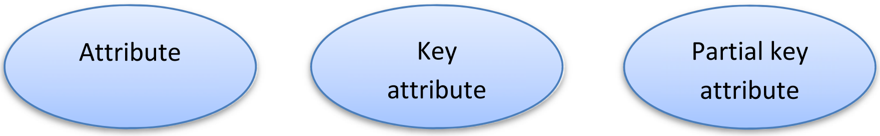
ii. Relationship

How entities act upon each other or are associated with each other. Think of relationships as verbs. For example, the named student might register for a course. The two entities would be the users and the properties, and the relationship depicted is the act of enrolling, connecting the two entities in that way. Relationships are typically shown as diamonds or labels directly on the connecting lines.



a. **Recursive relationship:** The same entity participates more than once in the relationship.

b. **Attribute:** A property or characteristic of an entity. It is often shown as an oval or circle.



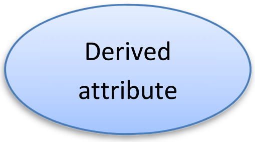
c. **Descriptive attribute:** A property or characteristic of a relationship (versus of an entity.)

d. **Attribute categories:** Attributes are categorized as simple, composite, derived, as well as single-value or multi-value.

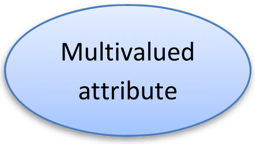
e. **Simple:** Means the attribute value is atomic and can't be further divided, such as a phone number.

f. **Composite:** Sub-attributes spring from an attribute.

g. **Derived:** Attributed is calculated or otherwise derived from another attribute, such as age from a birth date.



h. **Multi-value:** More than one attribute value is denoted, such as multiple phone numbers for a person.



Single-value: Just one attribute value. The types can be combined, such as: simple single-value attributes or composite multi-value attributes.

iii. Cardinality

Defines the numerical attributes of the relationship between two entities or entity sets. The three main cardinal relationships are one-to-one, one-to-many, and many-many.

a. **Cardinality views:** Cardinality can be shown as look-across or same-side, depending on where the symbols are shown.

b. **Cardinality constraints:** The minimum or maximum numbers that apply to a relationship.

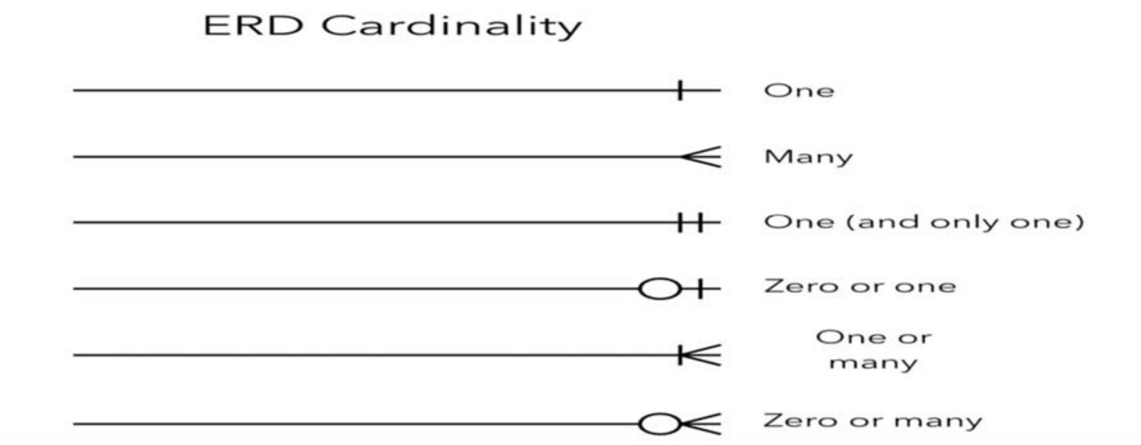


Figure:2.6 Relational symbol

iv. Mapping Natural Language

ER components can be equated to parts of speech, as Peter Chen did. This shows how an ER Diagram compares to a grammar diagram:

i. Common noun: Entity type. Example: Users.
ii. Proper noun: Entity. Example: Sally Smith.
iii. Verb: Relationship type. Example: Enrolls. (Such as in a Flat_id, which would be another entity type.)
iv. Adjective: Attribute for entity. Example: sophomore.
v. Adverb: Attribute for relationship. Example: digitally.

The database query language ERROL actually mimics natural language constructs. ERROL is based on reshaped relational algebra (RRA) and works with ER models, capturing their linguistic aspects.

v. ER Diagram

ER diagram represent the who is the entity and who is the attribute and what is the relation between one table to another table. Here we show the admin has the login table and manage the flat, sale and rent table. Admin also manage the user's comments and check the all record. Users provide a comment and search the rent, flat and sale table. The following example of ER diagram is given below,

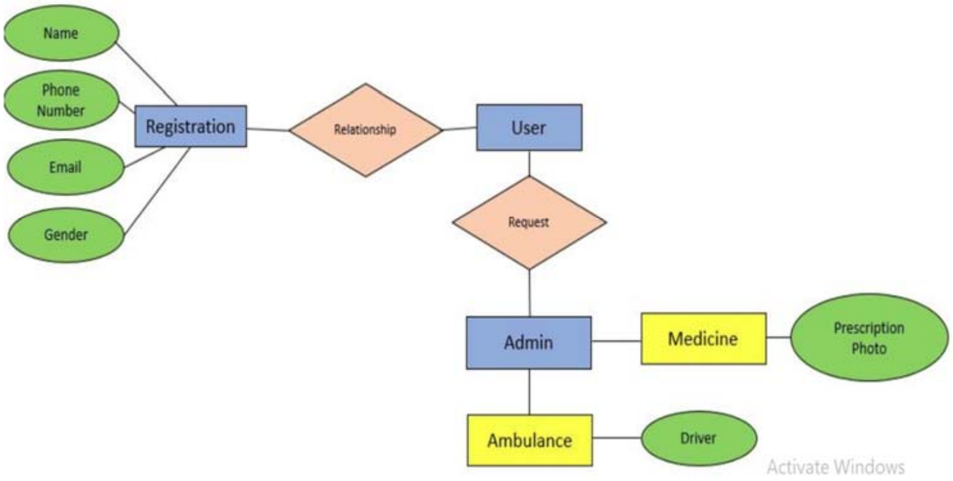


Figure 2.7: ER diagram

2.3.7 Data Flow Diagram (DFD)

A data flow diagram (DFD) maps out the flow of information for any process or system. It uses defined symbols like rectangles, circles and arrows, plus short text labels, to show data inputs, outputs, storage points and the routes between each destination. Data flowcharts can range from simple, even hand-drawn process overviews, to in-depth, multi-level DFDs that dig progressively deeper into how the data is handled. They can be used to analyze an existing system or model a new one. Like all the best diagrams and charts, a DFD can often visually "say" things that would be hard to explain in words, and they work for both technical and nontechnical audiences, from developer to CEO. That's why DFDs remain so popular after all these years. While they work well for data flow software and systems, they are less applicable nowadays to visualizing interactive, real-time or database-oriented software or systems.

2.3.7.1 Symbols and Notations Used in DFDs

One main difference in their symbols is that Yourdon-Coad and Yourdon-DeMarco use circles for processes, while Gane and Sarson use rectangles with rounded corners, sometimes called lozenges. There are other symbol variations in use as well, so the important thing to keep in mind is to be clear and consistent in the shapes and notations you use to communicate and collaborate with others. Using any convention's DFD rules or guidelines, the symbols depict the four components of data flow diagrams.

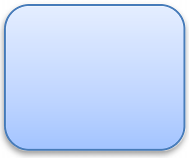
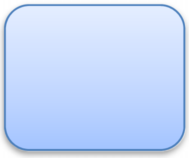
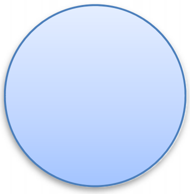
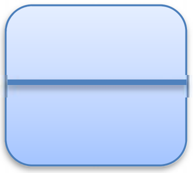
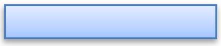
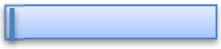
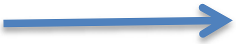
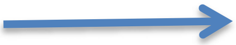
i. **External entity:** an outside system that sends or receives data, communicating with the system being diagrammed. They are the sources and destinations of information entering or leaving the system. They might be an outside organization or person, a computer system or a business system. They are also known as terminators, sources and sinks or actors. They are typically drawn on the edges of the diagram.

ii. **Process:** any process that changes the data, producing an output. It might perform computations, or sort data based on logic, or direct the data flow based on business rules. A short label is used to describe the process, such as “Submit payment”.

iii. **Data store:** files or repositories that hold information for later use, such as a database table or a membership form. Each data store receives a simple label, such as “Orders.”

iv. **Data flow:** the route that data takes between the external entities, processes and data stores. It portrays the interface between the other components and is shown with arrows, typically labeled with a short data name, like “Billing details”.

Here is a comprehensive look at diagram symbols and notations and how they’re used.

Notation	Yourdon and Coad	Gane and Sarson
External entity		
Process		
Data store		
Data flow		

2.3.7.2 DFD Rules and Tips

i. Each process should have at least one input and an output.
ii. Each data store should have at least one data flow in and one data flow out.
iii. Data stored in a system must go through a process.
iv. All processes in a DFD go to another process or a data store.

2.3.7.3 DFD Levels and Layers: From Context Diagrams to Pseudo Code

A data flow diagram can dive into progressively more detail by using levels and layers, zeroing in on a particular piece. DFD levels are numbered 0, 1 or 2, and occasionally go to even Level 3 or beyond. The necessary level of detail depends on the scope of what you are trying to accomplish.

i. DFD Level 0 is also called a Context Diagram. It's a basic overview of the whole system or process being analyzed or modeled. It's designed to be an at-a-glance view, showing the system as a single high-level process, with its relationship to external entities. A wide audience, including stakeholders, business analysts, data analysts and developers, should easily understand it.
ii. DFD Level 1 provides a more detailed breakout of pieces of the Context Level Diagram. You will highlight the main functions carried out by the system, as you break down the high-level process of the Context Diagram into its sub processes.
iii. DFD Level 2 then goes one-step deeper into parts of Level 1. It may require more text to reach the necessary level of detail about the system's functioning.
iv. Progression to Levels 3, 4 and beyond is possible, but going beyond Level 3 is uncommon. Doing so can create complexity that makes it difficult to communicate, compare or model effectively.

Using DFD layers, the cascading levels can be nested directly in the diagram, providing a cleaner look with easy access to the deeper dive. By becoming sufficiently detailed in the DFD, developers and designers can use it to write pseudo code, which is a combination of English and the coding language. Pseudo code facilitates the development of the actual code.

2.3.7.4 Examples of How DFDs can be Used

Data flow diagrams are well suited for analysis or modeling of various types of systems in different fields.

i. **DFD in software engineering:** This is where data flow diagrams got their main start in the 1970s. DFDs can provide a focused approached to technical development, in which more research is done up front to get to coding.

ii. **DFD in business analysis:** Business analysts use DFDs to analyze existing systems and find inefficiencies. Diagramming the process can uncover steps that might otherwise be missed or not fully understood.

iii. **DFD in business process re-engineering:** DFDs can be used to model a better, more efficient flow of data through a business process. BPR was pioneered in the 1990s to help organizations cut operational costs, improve customer service and better compete in the market.

iv. **DFD in agile development:** DFDs can be used to visualize and understand business and technical requirements and plan the next steps. They can be a simple yet powerful tool for communication and collaboration to focus rapid development.

v. **DFD in system structures:** Any system or process can be analyzed in progressive detail to improve it, on both a technical and non-technical basis.

The data flow diagram, the circle is representing to the process and the rectangle represents the data storage. Here the figure left rectangle Admin where he/she publish a new project, read write the project information, show the content details and edit the information. The users can show the all post and search a property and message.

The example of following DFD diagram:

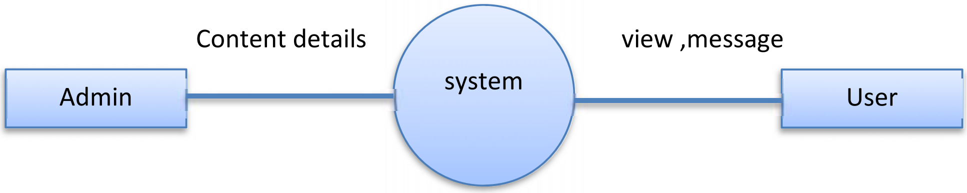


Figure : DFD diagram

2.3.8 Conclusions

Bachao is very necessary thing for people after analyzing the existing system and the problem faced by people that old system, we try to make such a system that we help people to find an easy way for the users. For saving time and easily access to this application. “Bachao” is also user friendly that everyone can use the system with their immature knowledge.