Web Application for “Kho Phyu”

May Barani Ko Ko

AGENDA

Task 1

(1.1) The Functions and Advantages of Web Application

What is Web Application?

A web application is also known as "web app". It is a client-server software application. It is an application that is accessed via web over a network such as the Internet or an intranet. That is stored on a remote server and delivered over the internet through a browser interface. Modern website have two important components that are flexible web browsers and web applications; both available to all and various at no cost.

Web browsers are software applications that allow users to retrieve data and interact with content located on web pages within a website.

Web pages may also run client-side scripts that "change" the internet browser into an interface for such applications as web mail (e.g. Gmail, Yahoo Mail etc.)

How does Web Application Work?

Web applications work quite different from regular windows applications. Web page

URL has 3 parts. They are – protocol, server name, file name.

For Example - **http://www.gustohei.com/HND_Diploma.aspx**

The protocol - http:

The server name - www.gustohei.com

The file name – HND_Diploma.aspx

Function of Web Application

Web applications are popular due to the ability to update and maintain Web applications without distributing and installing software on potentially thousands of client computers is a key reason for their popularity. Web applications are used to implement Webmail, online retail sales, online auctions, wikis, discussion boards, Weblogs, MMORPGs and many other functions.

In earlier types of client-server computing, each application had its own client program which served as its user interface and had to be separately installed on each user's personal computer. An upgrade to the server part of the application would typically require an upgrade to the clients installed on each user workstation, adding to the support cost and decreasing productivity.

In compare, Web applications dynamically generate a series of Web documents in a standard format supported by common browsers such as HTML/XHTML. Client-side scripting in a standard language such as JavaScript is commonly included to add dynamic elements to the user interface. Generally, each individual Web page is delivered to the client as a static document, but the sequence of pages can provide an interactive experience, as user input is returned through Web form elements embedded in the page markup. During the session, the Web browser interprets and displays the pages, and acts as the universal client for any Web application.

Advantages of Web Application

Web apps have many advantages than desktop applications. There are some advantages of web application.

(i) Developers do not need to develop web apps for multiple platforms (e.g. a single application that runs in Chrome will work on both Windows and OS).

(ii) Developers do not need to distribute software updates to users when the web app is by updating the application on the server, all users have access to the updated version.

(iii) The data you enter into a web application is processed and saved remotely. This allows you to access the same data from multiple devices, rather than transferring files between computer systems.

(iv) Cross platform compatibility. Most web based applications are more compatible across platforms than windows application. These web browsers are available for a multitude of operating systems and so whether you use Windows, Linux or Mac OS you can still run the web application.

(v) Web apps are more manageable. Web based systems need only be installed on the server placing minimal requirements on the end user workstation. This can make maintaining and updating the system much simpler as usually it can all be done on the server. Any client updates can be deployed via the web server with relative ease.

(vi) Widen access to our systems, simplify processes and improve relationships by providing more of our customers, suppliers and third parties with access to our systems.

(vii) Secure live data. Normally in larger more complex systems data is stored and moved around separate systems and data sources. In web based systems and processes can be combined reducing the need to transfer data around.

(viii) Web based applications also provide an added layer of security by removing the need for the user to have access to the data and back end servers.

(ix) Reduced costs. Web based applications can affectedly lower costs due to reduced support and maintenance, lower requirements on the end user system and simplified architecture. By further streamlining our business operations as a result of our web based application additional savings can often be found.

(x) Huge community. There are lots of amazing packages available to help our development and most of them are MIT licensed.

(xi) Fast development cycles. JavaScript, HTML, and CSS make it very easy to build powerful applications very quickly. It can be distributed offline as well. Web development is standards based and we are not required to use software that locks us to the platform.

(xii) Accessible at anywhere, web systems are accessible anytime, anywhere, via a PC with an Internet connection, putting the user in charge of where and when they access the application.

(xiii) Easily customizable. The user interface of web-based applications is easier to customize than it is in desktop applications. This makes it easier to update the look and feel of the application, or to customize the presentation of information to different user groups.

(xiv) Accessible for a range of devices. It can customizing satisfied for user groups, and can also be customized for presentation on any device connected to the internet, including mobile phones, etc.

(xv) Web app is easier installation and maintenance**.** Installation and maintenance becomes less complicated.

Conclusion

Web-based applications are:

- Easier to develop more useful for our users.
- Easier to install and maintain and keep secure.
- Easier to grow as you grow.
- Reduce business costs - less time spent talking to customers over the phone; eliminate printed materials; allow users to update their own details.
- Centralized data is secure and easy to backup.
- Quick and easy updates.
- Each anybody, anywhere in the world.
- Available 24 hours a day, 7 days a week.
- Low specification PCs or smart phones can be used.
- Online training can be completed at user's own time and pace.

- Direct access to latest information - for Employees where they are located.

- Always up-to-date.

(1.2) Comparison of Different Client-Side and Server-Side

At Client Side

Definition

Client-Side programming is writing code that will run on the client, and is done in languages that can be executed by the browser.

Languages

- JavaScript (primarily)
- HTML
- CSS

JavaScript

JavaScript is the most popular programming language in the world. JavaScript was invented by Brendan Eich in 1995, and became an ECMA standard in 1997. ECMA-262 is the official name. ECMAScript 6 (released in June 2015) is the latest official version of JavaScript.

JavaScript can change HTML content, HTML attributes, HTML styles (css) and can validate data.

HTML

HTML stands for Hyper Text Markup Language. HTML is a markup language for describing web documents (web pages).

A markup language is a set of markup tags. HTML documents are described by HTML tags.

CSS

CSS stands for Cascading Style Sheet. CSS is a style language that defines layout of HTML documents.

HTML can be used to add layout to websites. But CSS offers more options and is more accurate and sophisticated. CSS is supported by all browsers today.

Knowledge have in each side

- Javascript
- CSS

- HTML
- Basic Graphic Design
- Ajax
- Some 3rd party JavaScript libraries like "JQuery, etc."

Functions/Works (How Works)

Client side programming sends all of the code to the client, and the result is then computed on the client machine.

Usages

It makes interactive webpages, make stuff happen dynamically on the web page, interact with temporary storage, and local storage (Cookies, local Storage).

User

User opens his/her web browser (the client), browses to web page. Client sends a request to the server for their home page. Client receives the page's source, and renders it into a human viewable website.

Used

Client side scripting is used when the user's browser already has all the code and the page is altered on the basis of the users input.

Database

Client side scripting cannot be used to connect to the databases on the web server.

Programming (Used)

Much like the server- side, Client-side programming is the name for all of the programs which are run on the Client.

- Make interactive webpages.
- Make stuff happen dynamically on the web page.
- Interact with temporary storage, and local storage (Cookies, local Storage).
- Send requests to the server, and retrieve data from it.

File System

Client side scripting can't access the file system that resides at the web Server.

Setting

The files and settings that are local at the user's computer can be accessed using Client side scripting.

Blocked

Client side scripting is possible to be blocked by the use.

Response

Response from a client-side script is faster as compared to a server-side script because the scripts are processed on the local computer

At Server Side

Definition

Server-side programming is writing code that runs on the server, using languages supported by the server.

Languages

- PHP
- JSP
- ASP.Net
- Perl
- Python
- Ruby

PHP

PHP is an amazing and popular language. PHP stands for "PHP: Hypertext Preprocessor". PHP is a widely-used, open source scripting language. PHP scripts are executed on the server. PHP is free to download and use.

It is powerful enough to be at the core of the biggest blogging system on the web. It is deep enough to run the largest social network. It is also easy enough to be a beginner's first server side language.

JSP

JSP stands for JavaServer Pages. JSP is a server-side programming technology that enables the creation of dynamic, platform-independent method for building Web-based applications.

JSP have access to the entire family of Java APIs, including the JDBC API to access enterprise databases.

ASP.Net

ASP.Net is an open-source server-side web application framework designed for web development to produce dynamic web pages. It was developed by Microsoft to allow programmers to build dynamic web sites, web applications and web services. ASP.Net is in the process of being re-implemented as a modern and modular web framework, together with other frameworks like Entity Framework.

Knowledge have in each side

- HTML
- SQL
- Linux/Unix
- Shell scripting
- OOP, etc.

Functions/Works (How Works)

Server side code and applications run on a server machine, which sends the computed result to the client who requested it.

Usages

It is Process user input, display pages, structure web applications, and interact with permanent storage (SQL, files).

User

Server acknowledges the request, and replies the client with some Meta data, followed by the page's source. Server processes that data, and replies with a page matching the search results.

Used

Server side scripting is used to create dynamic pages based a number of conditions when the users browser makes a request to the server.

Database

Server side scripting is used to connect to the databases that reside on the web server.

Programming (Used)

Server-side programming, is the general name for the kinds of programs which are run on the Server.

- Process user input.
- Display pages.
- Structure web applications.
- Much like the server- side, Client-side programming is the name for all of the programs which are run on the Client.
- Make interactive webpages.
- Make stuff happen dynamically on the web page.
- Interact with permanent storage (SQL, files).

File System

Server side scripting can access the file system residing at the web server.

Setting

The settings that belong to Web server can be accessed using Server side scripting.

Blocked

Server side scripting can’t be blocked by the user.

Response

Response from a server-side script is slower as compared to a client-side script because the scripts are processed on the remote computer.

.Net Framework Architecture

The Microsoft .NET architecture is the programming model for the .NET platform. The .Net Framework provides a managed execution environment, simplified development and deployment and integration with a wide variety of programming languages.

The .NET Framework class library is a comprehensive, object-oriented collection of reusable types that we can use to develop applications. The .NET Framework class library includes ADO.NET, ASP.NET, and Window Forms.

Why should we choose ASP.NET?

This is a slight comparison of ASP.NET upon other (web) programming languages in the market. I have been searching to find out the difference between ASP.NET and other programming languages, to why should one use ASP.NET to develop the websites, when one can easily develop the website using PHP, Node.js, etc. I will try to explain why one should use ASP.NET over other programming languages.

Most people hate ASP.NET because ASP.NET is developed by Microsoft, and Microsoft like all other giants has some business rivals, who're never done exposing Microsoft to be evil. But they're evil just in the same way as Microsoft is, its business.

The very first reason to hate ASP.NET and to not use it is that was developed by Microsoft and is not an open-source project. People don't try ASP.NET by just reading about it being a product of Microsoft and then skip it. At least they must once try it out, I started from PHP, but I shifted to ASP.NET as soon as I wrote the first web page.

The syntax of ASP.NET as compared to other languages is insanely simple and easy to write and interpret. Not only this, creating other stuff that is a part of programming is really easy and fast in ASP.NET relative to other programming languages. You can see, a professional PHP developer would be able to get to the core concepts of the first code, whereas anyone can understand the second code, it takes two integer values and returns their sum. So, other developers who didn't develop this project can edit or update.

And then ASP.NET is shipped with .NET Framework, so they can develop their applications over the .NET Framework, which is a part of the Windows OS.

(1.3) Web Security Concerns and Recommendation for Security Improvements

Web application security is a branch of Information Security that deals specifically with security of websites, web applications and web services. At a high level, Web application security draws on the principles of application security but applies them specifically to Internet and Web systems. Typically web applications are developed using programming languages such as PHP, Java EE, Java, Python, Ruby, ASP.NET, C #, and VB.NET or Classic ASP. In Kho Phyu's web application, we developed with ASP.NET.

Since the internet operates over structured networks which are programmed, security problems are unavoidable. Loop holes, hacking and viruses are common areas where vulnerabilities will be taken advantage of with disruptive and disastrous results. Website security, is imperative for all online business or website owners and requires constant attention and updates. There are always new ways for "internet criminals" or hackers to "beat the system" and cause disruptions, especially where a website offers its internet users interactive convenience facilities.

Web applications security is fundamentally about protecting assets. Assets may be tangible items, such as a Web page or Kho Phyu's customer database or they may be less tangible, such as Kho Phyu's reputation. Security is a path, not a destination.

As we analyze our infrastructure and applications, we identify potential threats and understand that each threat presents a degree of risk. Security is about risk management and implementing effective countermeasures.

Common Website Security Problems

Website security problems can be divided into two categories:

- System Security – this ensures that a general internet user cannot change a website, altering content on web pages.
- Information Security – this ensures that the personal or private details of an internet user are secure and safe from prying eyes.

Human Error

Human beings are not by nature, perfect. Mistakes are, one could say, inherently a part of our "general make up". Consequently, most security problems on the internet come down to human error. Human beings programmer and run websites. Where mistakes are made, vulnerabilities are created. Website developers need to properly plan and proof test scripts that are coded into website programmers and applications often as hackers and other internet criminals will find ways to extract confidential information and do with it as they please. Particular errors will be exploited where the opportunity presents itself.

Privacy Neglect

More often than not, the general internet user will become too comfortable with the notion that internet surfing is "safe" and "anonymous" and openly part with personal details all too easily. Parting with this type of information could seemingly be as innocent as giving away a personal email address on a public forum and others of a more confidential nature such as credit card details. Hackers and internet criminals make use of "crawler bots" (small programmes coded to collect email addresses) who's function it is to locate addresses and add them to mass emailing lists, for the sole purpose of distributing SPAM to internet users. This isn't necessarily a serious security breach for websites, but when used in the same way to accumulate user names and passwords on sites, damage can be done.

Hacking Hacker's generally have little information or none at all at their disposal about their specific targets and establish a breakthrough almost entirely based on his or her own knowledge. The general internet user is usually not the main target. Internet or website servers of large corporations and organizations generally suffer with regular security breaches and should constantly be updated with newer security software versions.

Password Problems

Passwords can be intercepted in the following ways by internet criminals and hackers:

- Guessing – simple passwords such as a mother's maiden name, a pet's name that can be easily guessed.
- Brute force search which allows as many guesses as desired to be entered.
- Social engineering – tricking people into revealing password information.
- Obtaining stored passwords – passwords can be retrieved whereby people have stored them on computer systems etc.
- Obtaining shared passwords – the same passwords may be used for more than one system.
- Installing Trojans – "Trojan horse" software programs may install invisibly on a computer and monitor key strokes made by a user.
- Interception – passwords are sent across an unencrypted connection, which can then be intercepted and transmitted.

Security Risks with Web Applications

Webmaster is mostly affected by common issues and problems that internet criminals target. From the very minute that a web server is installed, a "window" (of opportunity) into a local network is opened. Anyone, anywhere with online access has the ability to "peer through" this window. Most internet users are content with what they're presented with and aren't likely to "nose around" and peek at things that were never really intended for public consumption, many other individuals are "free" to figure out ways to snoop. This sort of behavior can be likened to not being able to "look without touching". These individuals will attempt to force their way inside this opened window and cause programming or structural damage by, for instance inserting a "bug". Surfing the web may, to the general, innocent internet user, be viewed as a safe and anonymous environment. The simple truth is that the internet isn't quite all that safe and anonymous at all. In a sense the internet "has eyes" everywhere. Web browsers can be easily exposed to viruses and malicious software, causing a user's personal system to experience malfunctions and problems.

Web application security is one of the aspects most at risk from the adoption of web-based technologies for conducting business online. While web applications have enabled organizations to connect seamlessly with suppliers, customers and other stakeholders; web application vulnerabilities have also exposed a multitude of previously unknown security risks.

If web application security is not taken care of, meaning that web application vulnerability is allowed to happen, then not only your entire database of sensitive information is at serious risk, but your website can become the launch site of criminal activities such as hosting phishing sites or used to transfer illegal content.

Some hackers take advantage of this lack of web application security from web application vulnerabilities such as SQL Injection or Cross-Site Scripting and may maliciously inject code within vulnerable web applications to trick users and redirect them towards phishing sites. Recent research shows that 75% of cyber-attacks are done at web application level. Hence ensuring web application security is crucial.

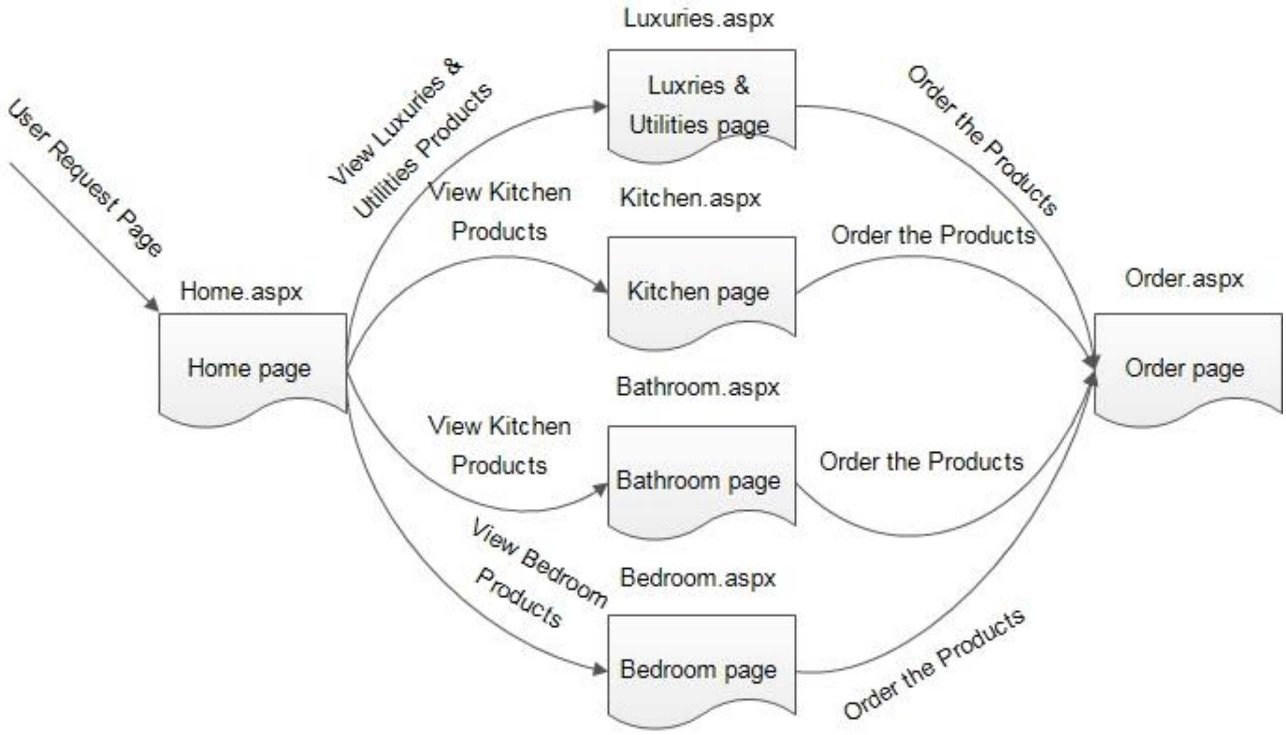
Types of security risks

Types of security risks whereby network eavesdropping can occur include:

- Bugs or mis-configuration problems in a web server – this allows confidential documents to be "stolen", commands on the server host machine to be modified and web server host machine vulnerabilities able to be "broken into" etc.
- Browser – side risks – this allows active for content to crash the browser, damage an internet user's system and breach a user's privacy.
- Interception of network data (sent and received) – this allows hackers the ability to operate from any point on the pathway between a web browser and server causing disruptions.

Task 2

(2.1) Screen-Flow Diagram for the "Kho Phyu"



(2.2) Client-side and Server-side Functionality in a Web Application

Client-side Environment

The client-side environment used to run scripts is usually a browser. The processing takes place on the end users computer. The source code is transferred from the web server to the user’s computer over the internet and run directly in the browser.

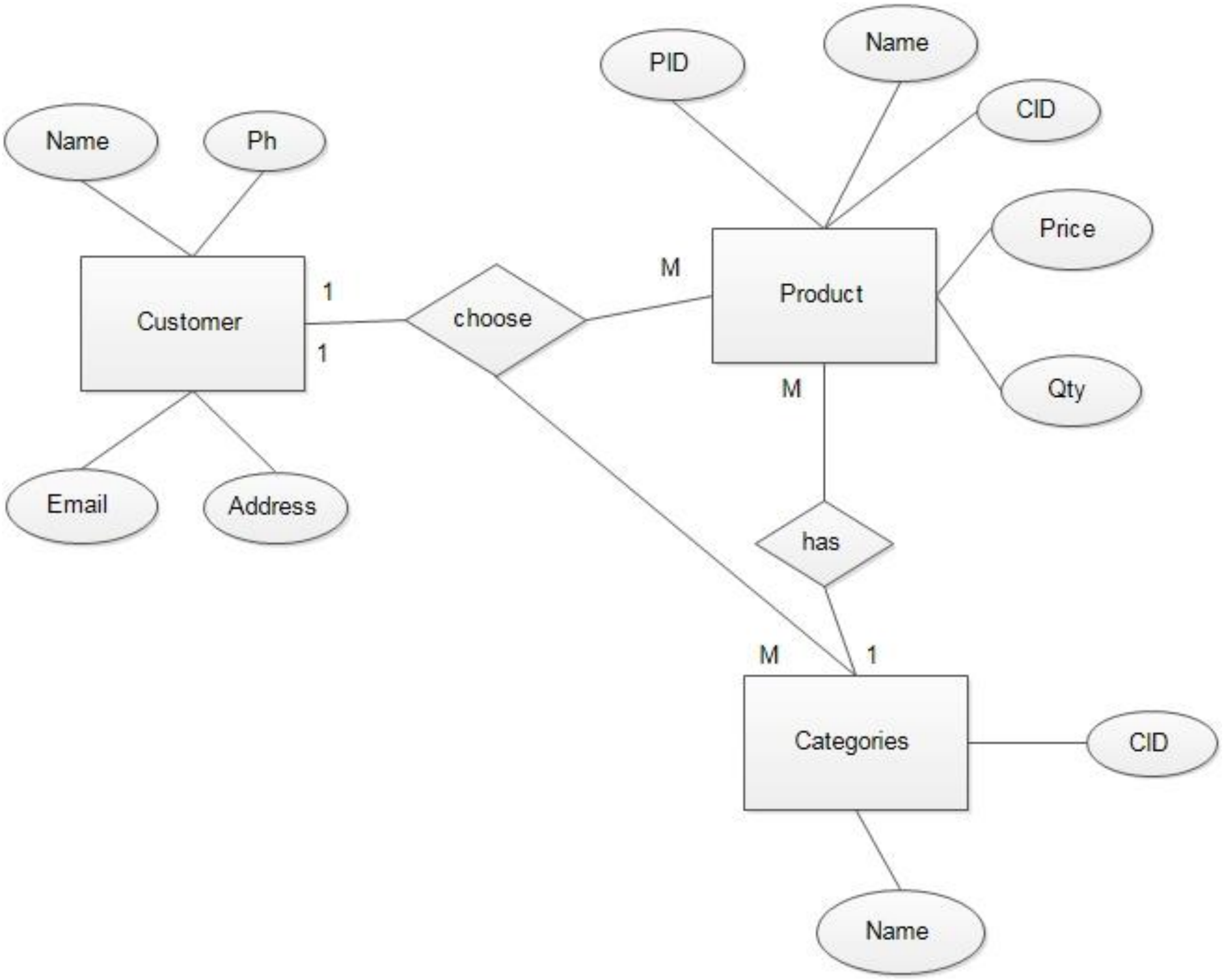
The scripting language needs to be enabled on the client computer. Sometimes if a user is conscious of security risks they may switch the scripting facility off. When this is the case a message usually pops up to alert the user when script is attempting to run.

Server-side Environment

The server-side environment that runs a scripting language is a web server. A user's request is fulfilled by running a script directly on the web server to generate dynamic HTML pages. This HTML is then sent to the client browser. It is usually used to provide interactive web sites that interface to databases or other data stores on the server.

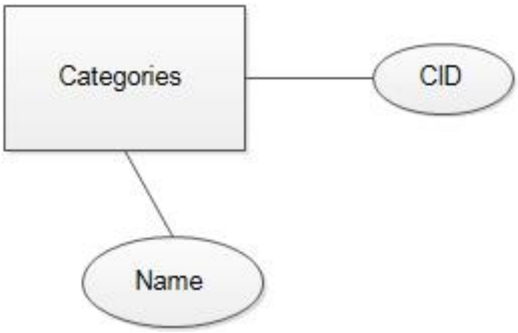
This is different from client-side scripting where scripts are run by the viewing web browser, usually in JavaScript. The primary advantage to server-side scripting is the ability to highly customize the response based on the user's requirements, access rights, or queries into data stores.

(2.3) Entity Relationship Diagram (ERD)



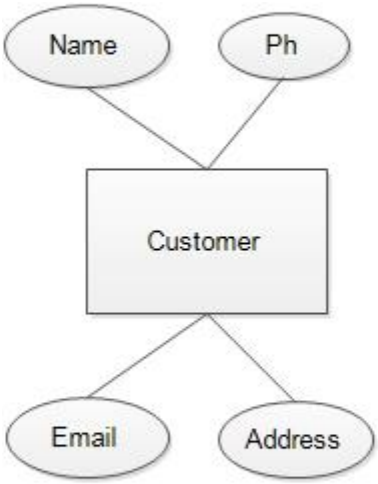
Data Dictionary

Categories Table



Field Name	Data Type	Field Length	Description	Primary Key/ Foreign Key
CID	int	-	Does not allow null value, ID of Categories	Primary Key
Name	varchar	50	Does not allow null value, Name of Categories	-

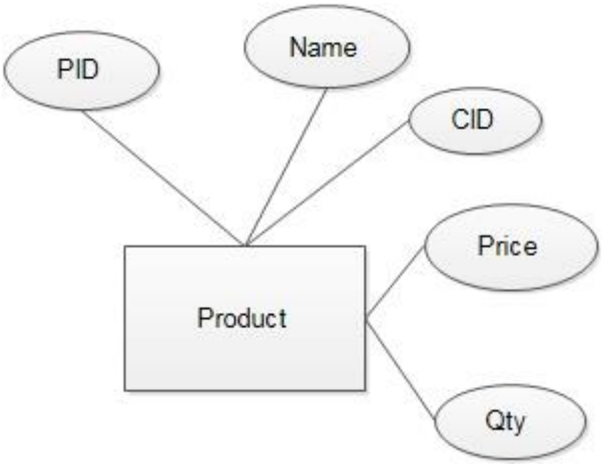
Customer Table



Field Name	Data Type	Field Length	Description	Primary Key/ Foreign Key
Name	varchar	50	Does not allow null value, Name of Customer	-

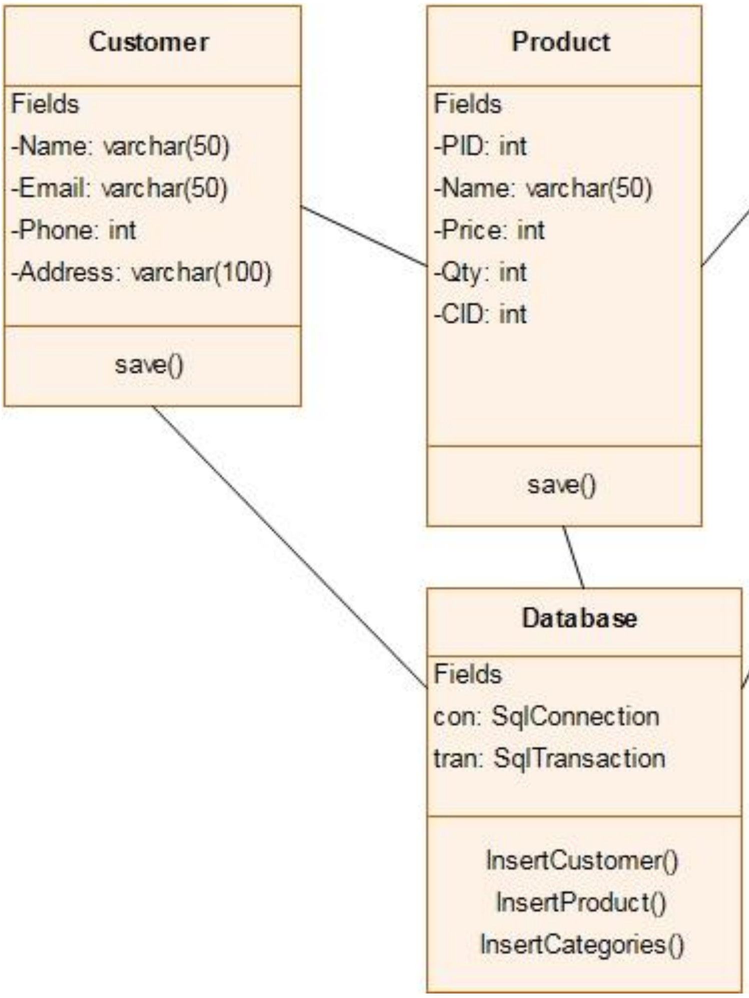
Phone	int	-	Does not allow null value, Ph of Customer	-
Email	varchar	50	Does not allow null value, Email of Customer	-
Address	varchar	100	Does not allow null value, Address of Customer	-

Product Table



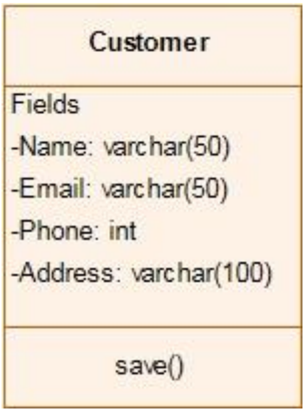
Field Name	Data Type	Field Length	Description	Primary Key/ Foreign Key
PID	int	-	does not allow null value, ID of Product	Primary Key
Name	varchar	50	Does not allow null value, Name of Product	-
CID	int	-	Does not allow null value, ID of Categories	Foreign Key
Price	int	-	Does not allow null value, Price of Product	-
Qty	int	-	Does not allow null value, Quantity of Product	-

(2.4) Class Diagram



Class Behavior and Data Description Table

Customer Class



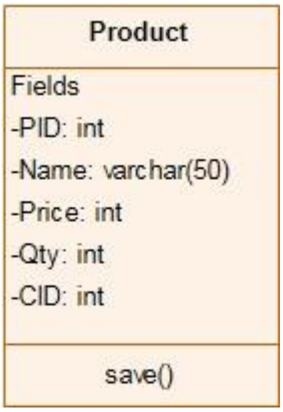
Data Description for Customer Class

Access Modifier	Data Type	Variable Name	Description
private	varchar(50)	Name	To know the customer’s name
private	varchar(50)	Email	To know the customer’s email
private	int	Phone	To know the customer’s phone
private	varchar(100)	Address	To know the customer’s address

Methods for Customer Class

Access Modifier	Return Type	Method Name	Parameters	Description
private	-	-	-save(Name:varchar(50), Email:varchar(50),Phone:int,Address:varchar(100))	-

Product Class



Data Description for Product Class

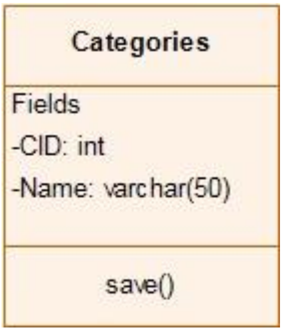
Access Modifier	Data Type	Variable Name	Description
private	int	PID	To know the product’s id
private	varchar(50)	Name	To know the product’s name

private	int	Price	To know the product’s price
private	int	Qty	To know the product’s qty
Private	int	CID	To know the product’s categories id

Methods for Product Class

Access Modifier	Return Type	Method Name	Parameters	Description
private	-	-	-save(PID:int, Name:varchar(50), Price:int, Qty:int, CID:int)	-

Categories Class



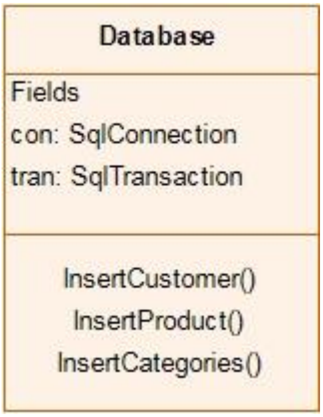
Data Description for Categories Class

Access Modifier	Data Type	Variable Name	Description
Private	int	CID	To know the categories’ id
private	varchar(50)	Name	To know the categories’ name

Methods for Categories Class

Access Modifier	Return Type	Method Name	Parameters	Description
private	-	-	-save(CID:int, Name:varchar(50))	-

Database Class



Database

Fields
con: SqlConnection
tran: SqlTransaction

InsertCustomer()
InsertProduct()
InsertCategories()

Methods for Database Class

Access Modifier	Return Type	Method Name	Parameters	Description
private	void	InsertCustomer()	-	Insert when the customer ordered the product
private	void	InsertProduct()	-	Insert when the new product is arrived
private	void	InsertCategories()	-	Insert when the new category is added