

I N D E X

No.	Title	Page No
1	PROJECT PROFILE <ul style="list-style-type: none"> • Project Definition • Proposed System • References 	4
2	System Requirement Specification <ul style="list-style-type: none"> • Technology Used • Front-End Technology • Front-End Tools • Back-End Technology • Testing Tools • Operating Systems • Operating Environment • External Interface Environment 	7
3	SYSTEM DESIGN <ul style="list-style-type: none"> • Use Case Diagram • Class Diagram • Sequence Diagram • Activity Diagram • Data Flow Diagram • Entity Relationship diagram 	10
4	Data Dictionary: <ul style="list-style-type: none"> • Database Name • Table Name • Admin Table • User Table 	39
5	Software Testing: <ul style="list-style-type: none"> • Selenium testing 	47
6	Input Output design	49
7	ABOUT THE TOOLS: <ul style="list-style-type: none"> • Front-End Tool • Back-End Tools 	50
8	CONCLUSION (FUTURE SCOPE, LIMITATIONS)	54

1. PROJECT PROFILE

Project Definition

- online grocery shop

Proposed System

- online food and grocery store.
- With over 18,000 products and over a 1000 brands in our catalogue
- you will find everything you are looking for. Right from fresh Fruits and Vegetables, Rice and Dals, Spices and Seasonings to Packaged products, Beverages, Personal care products, Meats – we have it all.
- Choose from a wide range of options in every category, exclusively handpicked to help
- you find the best quality available at the lowest prices.
- Select a time slot for delivery and your order will be delivered right to your doorstep

References

Text Reference:

PHP: The Complete Reference

- Book by Steven Holzner
- System Analysis & Design

Web Reference:

HTML:

- <https://www.w3schools.com/html/>
- <https://www.tutorialspoint.com/html/>
- <https://www.guru99.com/html-tutorials.html>
- <https://www.javatpoint.com/html-tutorial>
- <https://www.geeksforgeeks.org/html/>

PHP:

- <https://www.w3schools.com/php/>
- <https://www.tutorialspoint.com/php/>
- <https://www.guru99.com/php-tutorials.html>
- <https://www.javatpoint.com/php-tutorial>
- <https://www.geeksforgeeks.org/php/>

ajax:

- https://www.w3schools.com/xml/ajax_intro.asp
- <https://www.tutorialspoint.com/ajax/>
- <https://www.javatpoint.com/ajax-tutorial>

CSS:

- <https://www.w3schools.com/css/>
- <https://www.tutorialspoint.com/css/>
- <https://www.csstutorial.net/>

javascript:

- <https://www.w3schools.com/js/>
- <https://www.tutorialspoint.com/javascript/>
- <https://www.javatpoint.com/javascript-tutorial>

Bootstrap:

- <https://www.w3schools.com/bootstrap/>
- <https://www.tutorialspoint.com/bootstrap/>

SQL:

- <https://www.w3schools.com/sql/>
- <https://www.tutorialspoint.com/sql/>

System Analyst Design:

- https://www.tutorialspoint.com/system_analysis_and_design/

Software Testing And Project Management:

- https://www.tutorialspoint.com/software_engineering/

2. SYSTEM REQUIREMENT SPECIFICATION

Technology Used:

- About 80% of everything on the web is written in PHP.
- It is the language of the project and the basis for the majority of present-day
web services,
B2C and B2B applications
content management systems
eCommerce solutions
- Major PHP appeals are:
 - Reduced costs thanks to its open-source nature.
 - High performance.
 - Ability to run on any OS, be it Linux, Unix, Windows or MacOS.
 - Easy troubleshooting.
 - Powerful extensions, such as Zend, CakePHP, Laravel.
 - Support of a wide range of databases.

Front-End Technology

- HTML5
- CSS
- JQUERY
- AJAX
- BOOSTRAP

Front-End Tools:

- Sublime text
- Notepad++

Back-End Technology:

- PHP
- SQL SERVER

Testing Tools:

- Browser:
Google Chrome
Firefox
Internet Explorer

Operating Systems:

- Windows 7/8
- Windows 8.1
- Windows 10

Operating Environment:

- **Processor:**
Core to Duo
- **RAM:**
1 GB or More
- **Hard Drive:**
20 GB Disk Drive Space (Available)
- **Internet Connection:**
High speed internet connection

External Interface Environment:

- **Hardware Interfaces:-**

Client Side:

Processor: Core i3

RAM: 1 GB or More

- **Software Interfaces:-**

Client Side:

Web Browser, windows XP/7/8

Web server:

IIS Web Server

Data Base Server:

Sql Server Management Studio

- **Communications Interfaces:-**

Any sufficient browser like

Mozilla

Firefox

Google Chrome

3. SYSTEM DESIGN

Use Case Diagram:

What is a Use Case Diagram?

A use case diagram is a dynamic or behavior diagram in UML. Use case diagrams model the functionality of a system using actors and use cases. Use cases are a set of actions, services, and functions that the system needs to perform. In this context, a "system" is something being developed or operated, such as a web site. The "actors" are people or entities operating under defined roles within the system.

Why Make Use Case Diagrams?

Use case diagrams are valuable for visualizing the functional requirements of a system that will translate into design choices and development priorities.

They also help identify any internal or external factors that may influence the system and should be taken into consideration.

They provide a good high level analysis from outside the system. Use case diagrams specify how the system interacts with actors without worrying about the details of how that functionality is implemented.

Basic Use Case Diagram Symbols and Notations

System

Draw your system's boundaries using a rectangle that contains use cases. Place actors outside the system's boundaries.



Use Case

Draw use cases using ovals. Label the ovals with verbs that represent the system's functions.



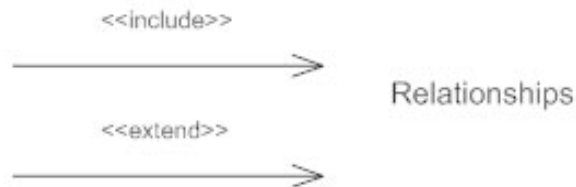
Actors

Actors are the users of a system. When one system is the actor of another system, label the actor system with the actor stereotype.



Relationships

Illustrate relationships between an actor and a use case with a simple line. For relationships among use cases, use arrows labeled either "uses" or "extends." A "uses" relationship indicates that one use case is needed by another in order to perform a task. An "extends" relationship indicates alternative options under a certain use case.



Tips for Use Case Diagrams

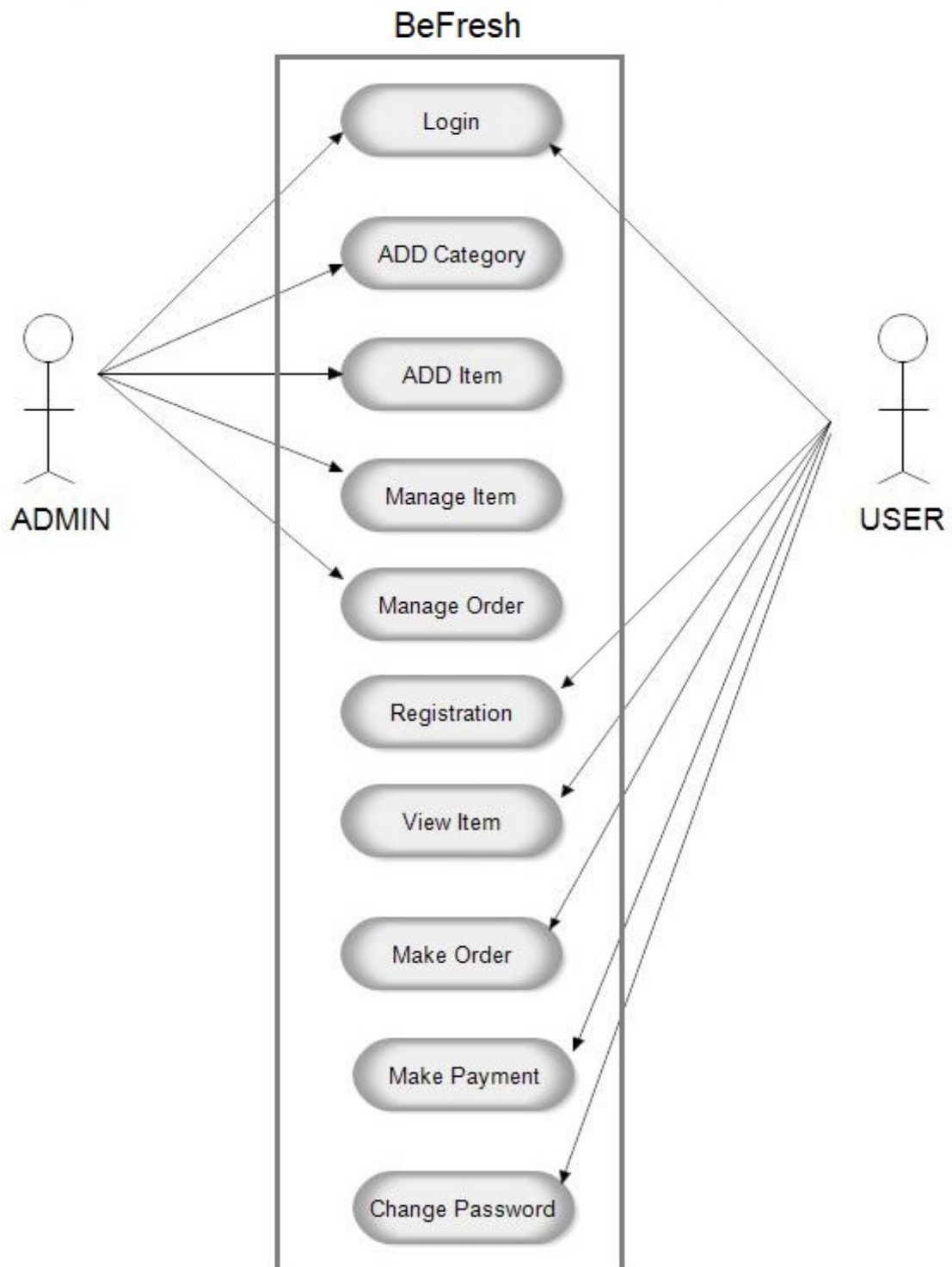
When thinking of use cases, think of the end goal of a user. They don't want to "login" or "sign up." That's not a use case. The use case is more like "make a purchase."

Actors don't have names. They're not "Bob." They represent the role of someone interacting with the system.

Keep your names short and the size of your use cases consistent for a professional look.

For a detailed implementation of a user's goal use a sequence diagram

Use Case Diagram of BeFresh:



Class Diagram:

What is Class Diagram?

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects

Purpose of Class Diagrams

Shows static structure of classifiers in a system

Diagram provides basic notation for other structure diagrams prescribed by UML

Helpful for developers and other team members too

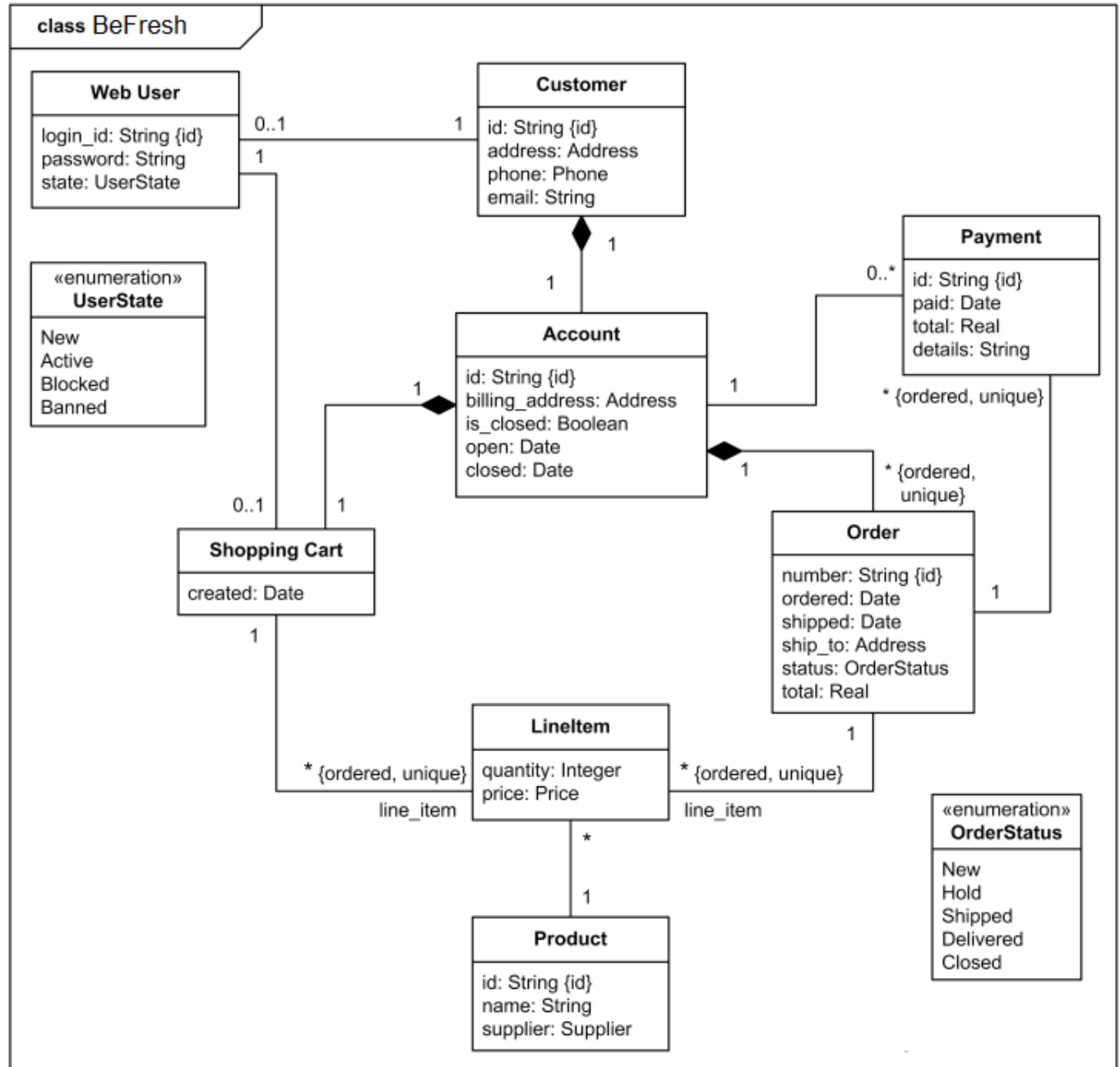
Business Analysts can use class diagrams to model systems from business perspective

A class diagram is made up of:

A set of classes and

A set of relationships between classes

Class Diagram BeFresh:



Sequence Diagram:

A sequence diagram describes an interaction among a set of objects participated in a collaboration (or scenario), arranged in a chronological order; it shows the objects participating in the interaction by their "lifelines" and the messages that they send to each other.

Sequence Diagram Notations

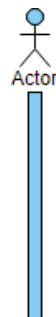
Lifeline

A lifeline represents an individual participant in the Interaction.



Actor

An Actor is a type of role played by an entity that interacts with the subject (e.g., by exchanging signals and data). An actor can also be an external to the subject (i.e., in the sense that an instance of an actor is not a part of the instance of its corresponding subject). They typically represent roles played by human users, external hardware, or other subjects.



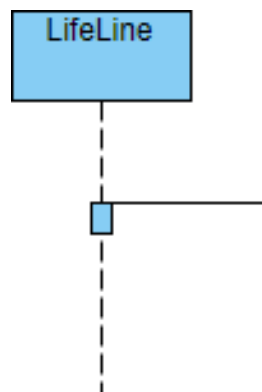
Note That:

An actor does not necessarily represent a specific physical entity but merely a particular role of some entity

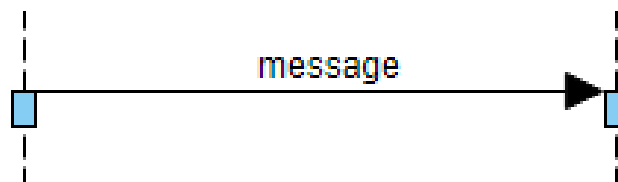
A person may play the role of several different actors and, conversely, a given actor may be played by multiple different person.

Activation

An activation is represented by a thin rectangle on a lifeline) represents the period during which an element is performing an operation. The top and the bottom of the of the rectangle are aligned with the initiation and the completion time respectively

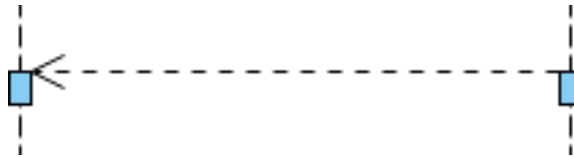
**Messages/Call Message**

A call message defines a particular communication between lifelines of an interaction, which represents an invocation of operation of target lifeline.



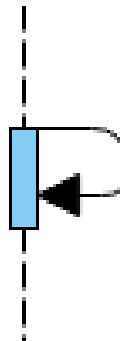
Return Message

A return message defines a particular communication between lifelines of an interaction, which represents the pass of information back to the caller of a corresponded former message.



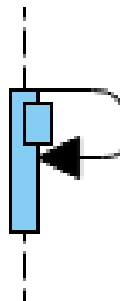
Self Message

A self message defines a particular communication between lifelines of an interaction, which represents the invocation of message of the same lifeline.



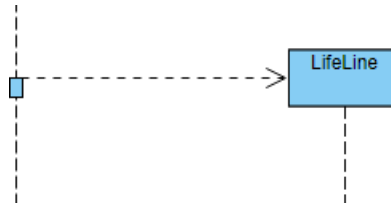
Recursive Message

A recursive message defines a particular communication between lifelines of an interaction, which represents the invocation of message of the same lifeline. Its target points to an activation on top of the activation where the message was invoked from.



Create Message

A create message defines a particular communication between lifelines of an interaction, which represents the instantiation of (target) lifeline.



Destroy Message

A destroy message defines a particular communication between lifelines of an interaction, which represents the request of destroying the lifecycle of target lifeline.



Duration Message

A duration message defines a particular communication between lifelines of an interaction, which shows the distance between two time instants for a message invocation.

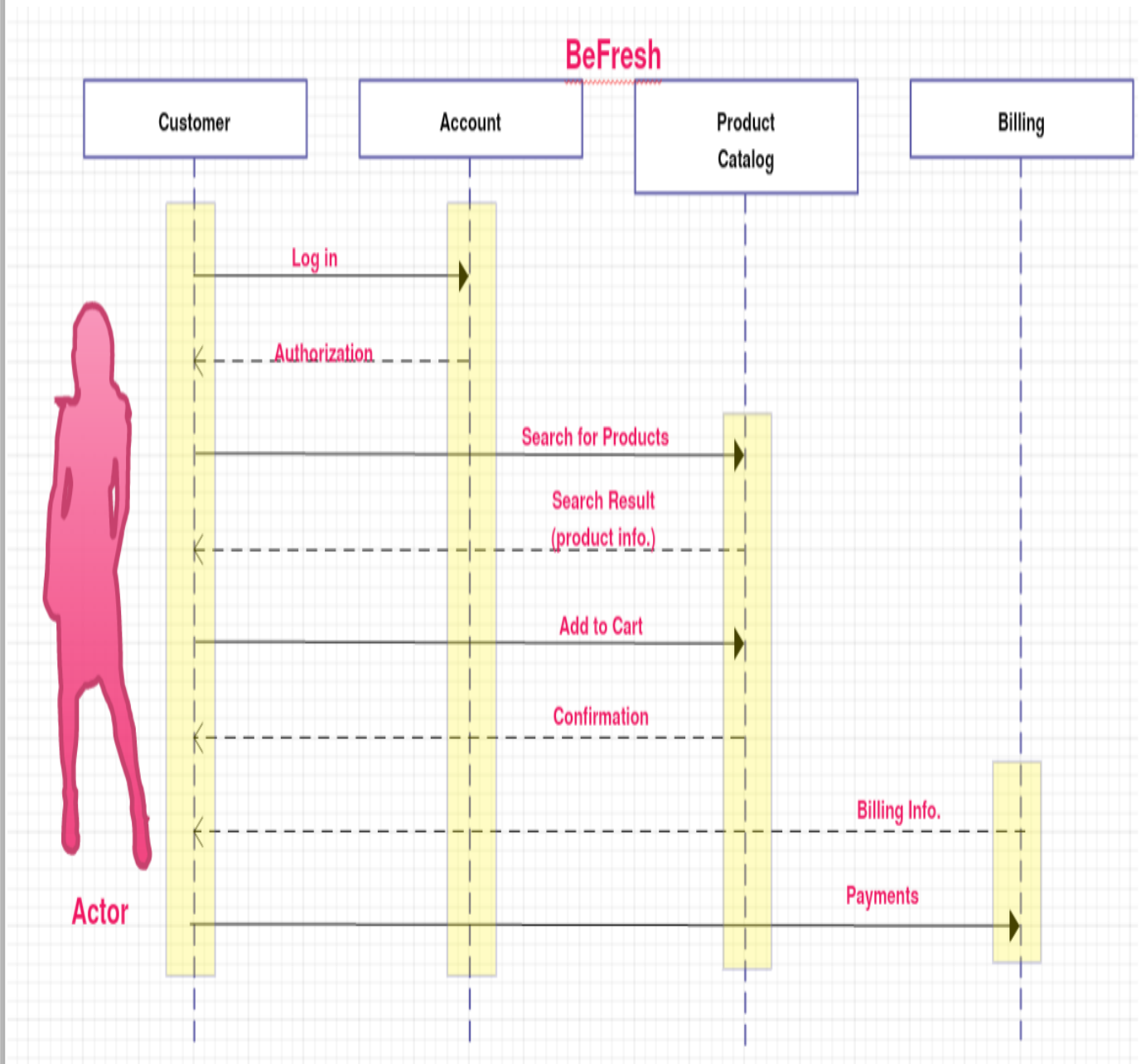


Note

A note (comment) gives the ability to attach various remarks to elements. A comment carries no semantic force, but may contain information that is useful to a modeler.



Sequence Diagram of BeFresh:



Activity Diagram:

An activity diagram visually presents a series of actions or flow of control in a system similar to a flowchart or a data flow diagram. Activity diagrams are often used in business process modeling. They can also describe the steps in a use case diagram. Activities modeled can be sequential and concurrent. In both cases an activity diagram will have a beginning (an initial state) and an end (a final state).

Basic Activity Diagram Notations and Symbols

Initial State or Start Point

A small filled circle followed by an arrow represents the initial action state or the start point for any activity diagram. For activity diagram using swimlanes, make sure the start point is placed in the top left corner of the first column.



Start Point/Initial State

Activity or Action State

An action state represents the non-interruptible action of objects. You can draw an action state in SmartDraw using a rectangle with rounded corners.



Activity

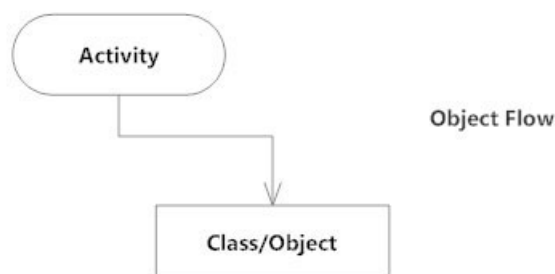
Action Flow

Action flows, also called edges and paths, illustrate the transitions from one action state to another. They are usually drawn with an arrowed line.



Object Flow

Object flow refers to the creation and modification of objects by activities. An object flow arrow from an action to an object means that the action creates or influences the object. An object flow arrow from an object to an action indicates that the action state uses the object.



Decisions and Branching

A diamond represents a decision with alternate paths. When an activity requires a decision prior to moving on to the next activity, add a diamond between the two activities. The outgoing alternates should be labeled with a condition or guard expression. You can also label one of the paths "else."



Guards

In UML, guards are a statement written next to a decision diamond that must be true before moving next to the next activity. These are not essential, but are useful when a specific answer, such as "Yes, three labels are printed," is needed before moving forward.



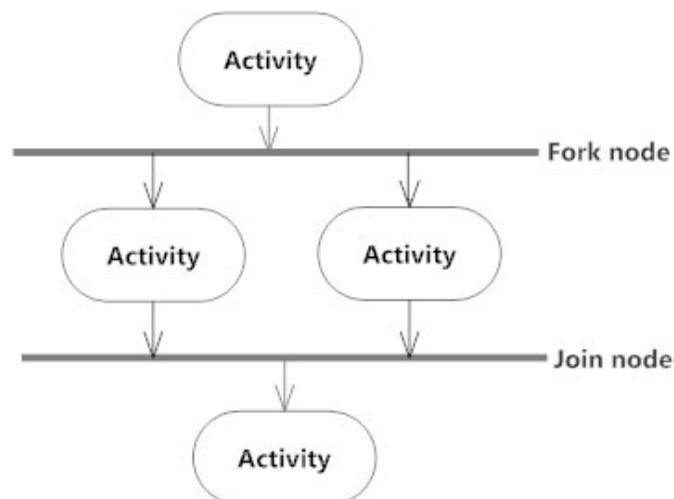
Synchronization

A fork node is used to split a single incoming flow into multiple concurrent flows. It is represented as a straight, slightly thicker line in an activity diagram.

A join node joins multiple concurrent flows back into a single outgoing flow.

A fork and join node used together are often referred to as synchronization.

Synchronization



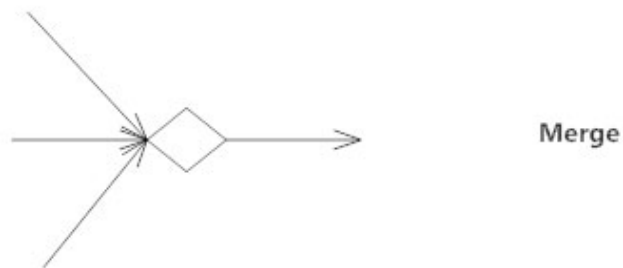
Time Event

This refers to an event that stops the flow for a time; an hourglass depicts it.



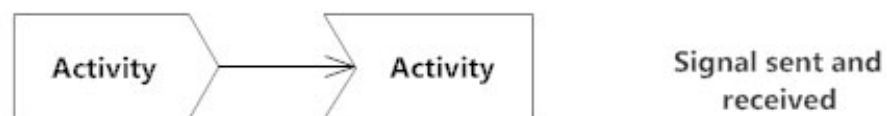
Merge Event

A merge event brings together multiple flows that are not concurrent.



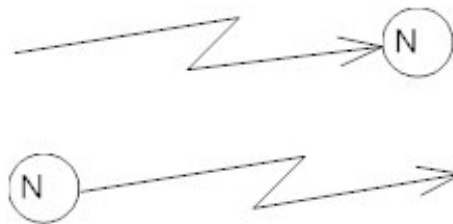
Sent and Received Signals

Signals represent how activities can be modified from outside the system. They usually appear in pairs of sent and received signals, because the state can't change until a response is received, much like synchronous messages in a sequence diagram. For example, an authorization of payment is needed before an order can be completed.



Interrupting Edge

An event, such as a cancellation, that interrupts the flow denoted with a lightning bolt.



Interrupting Edge Symbols

Final State or End Point

An arrow pointing to a filled circle nested inside another circle represents the final action state.

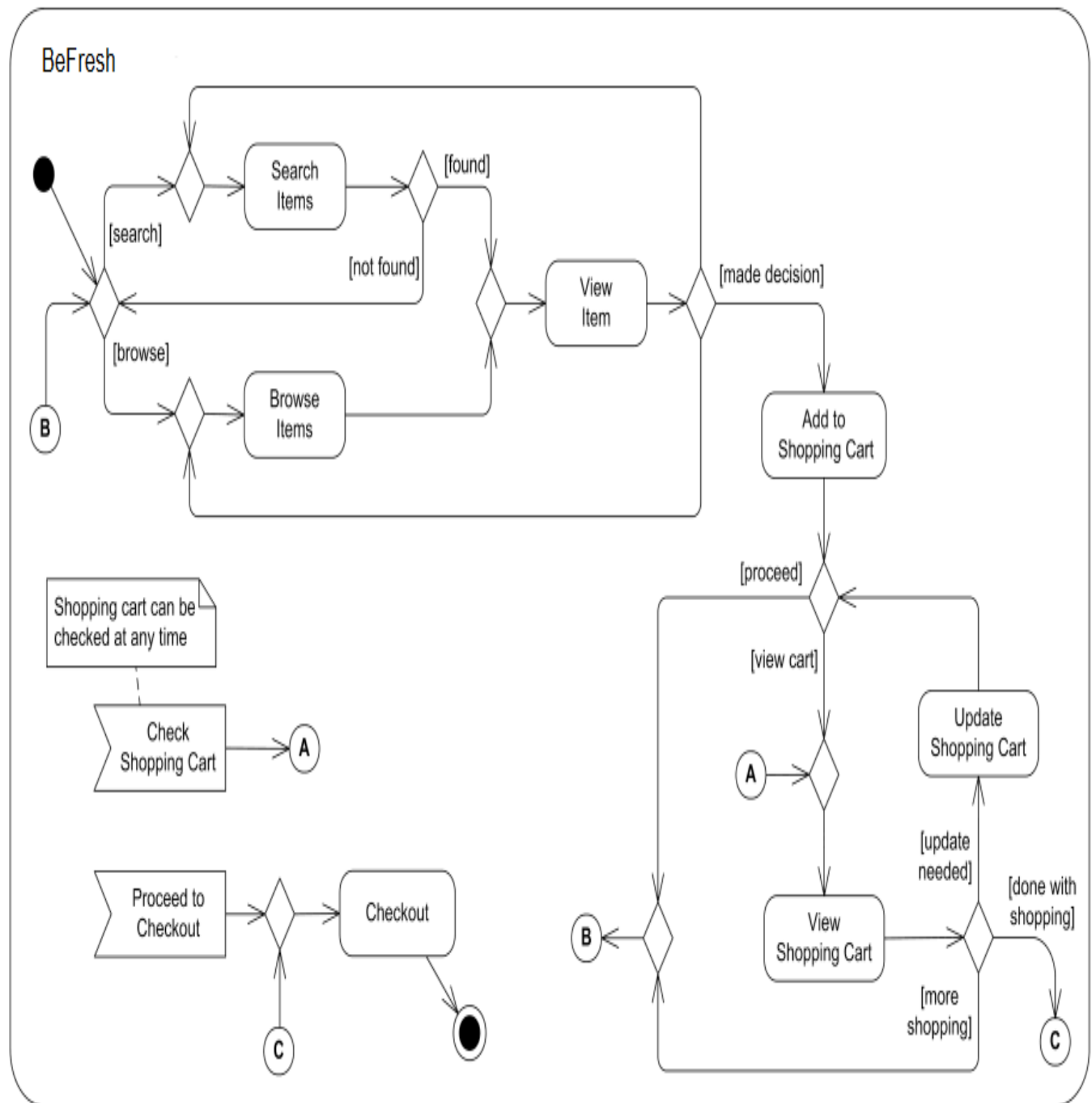


End Point Symbol

Swimlanes

Swimlanes group related activities into one column.

Activity Diagram of BeFresh:



Data Flow Diagram:

Data flow diagram notations

External Entity –

Also known as actors, sources or sinks, and terminators, external entities produce and consume data that flows between the entity and the system being diagrammed.

Process –

An activity that changes or transforms data flows. Since they transform incoming data to outgoing data, all processes must have inputs and outputs on a DFD.

Data Store –

A data store does not generate any operations but simply holds data for later access. Data stores could consist of files held long term or a batch of documents stored briefly while they wait to be processed.

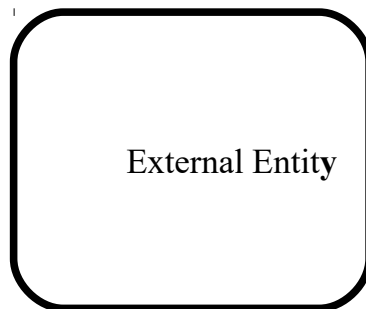
Data Flow –

Movement of data between external entities, processes and data stores is represented with an arrow symbol, which indicates the direction of flow.

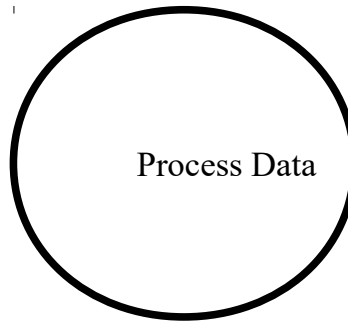
Data flow diagram symbols

A data flow diagram shows how data is processed within a system based on inputs and outputs.

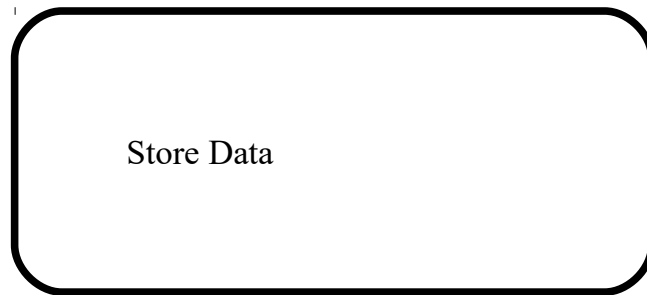
External Entity



Process Data



Store Data



Flow

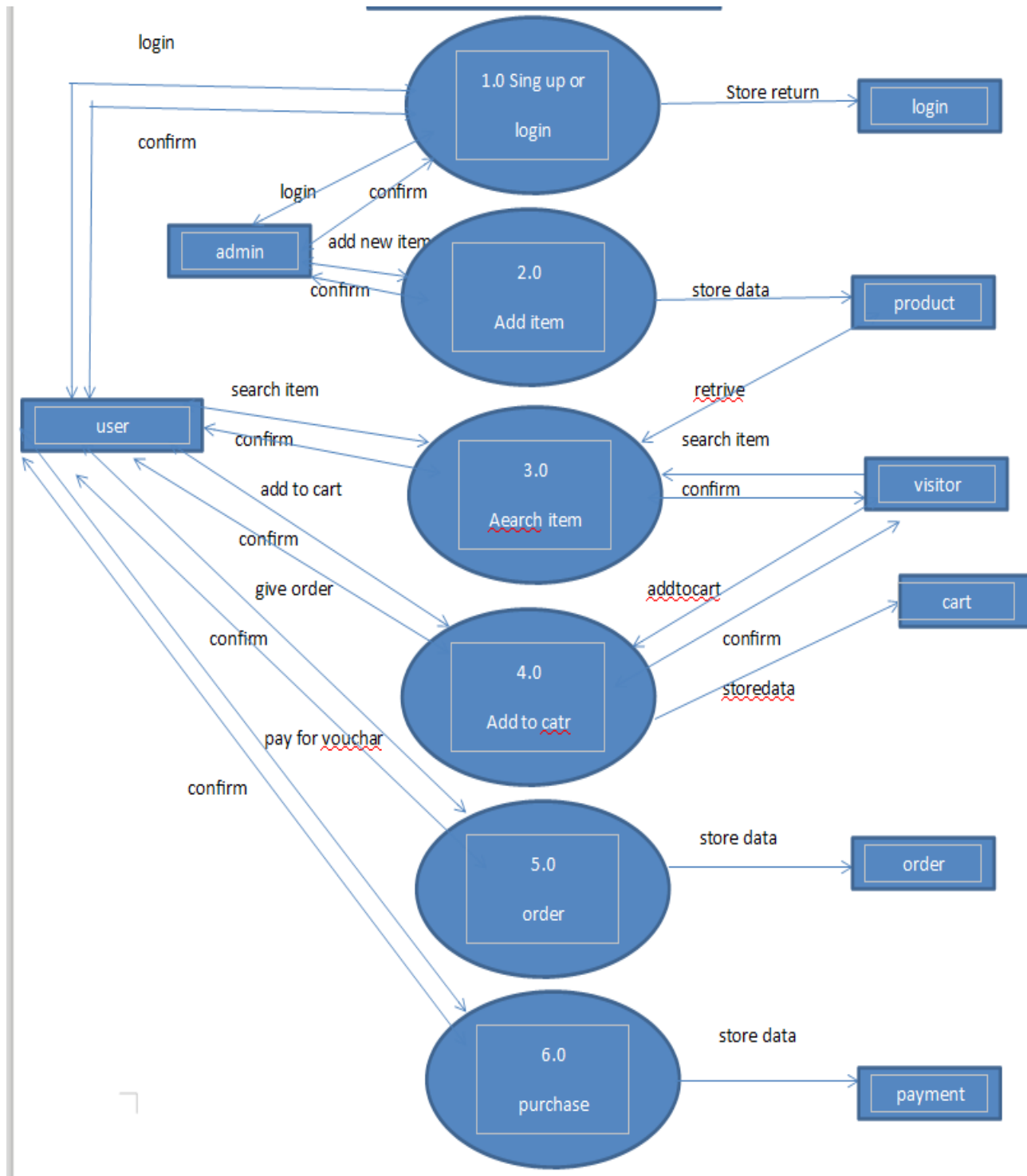


Data Flow Diagram of Befresh:

Level 0:

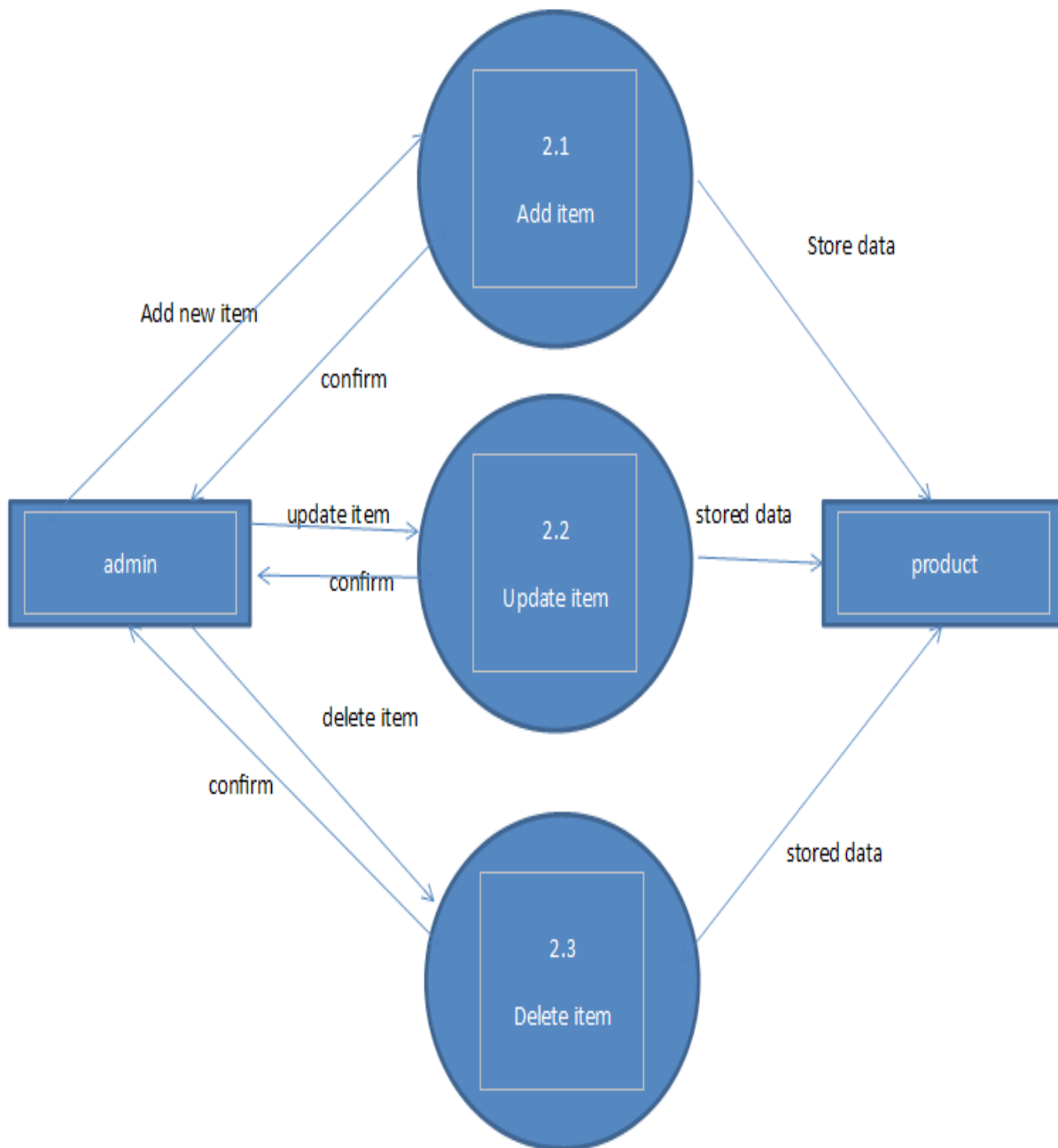


Level 1:

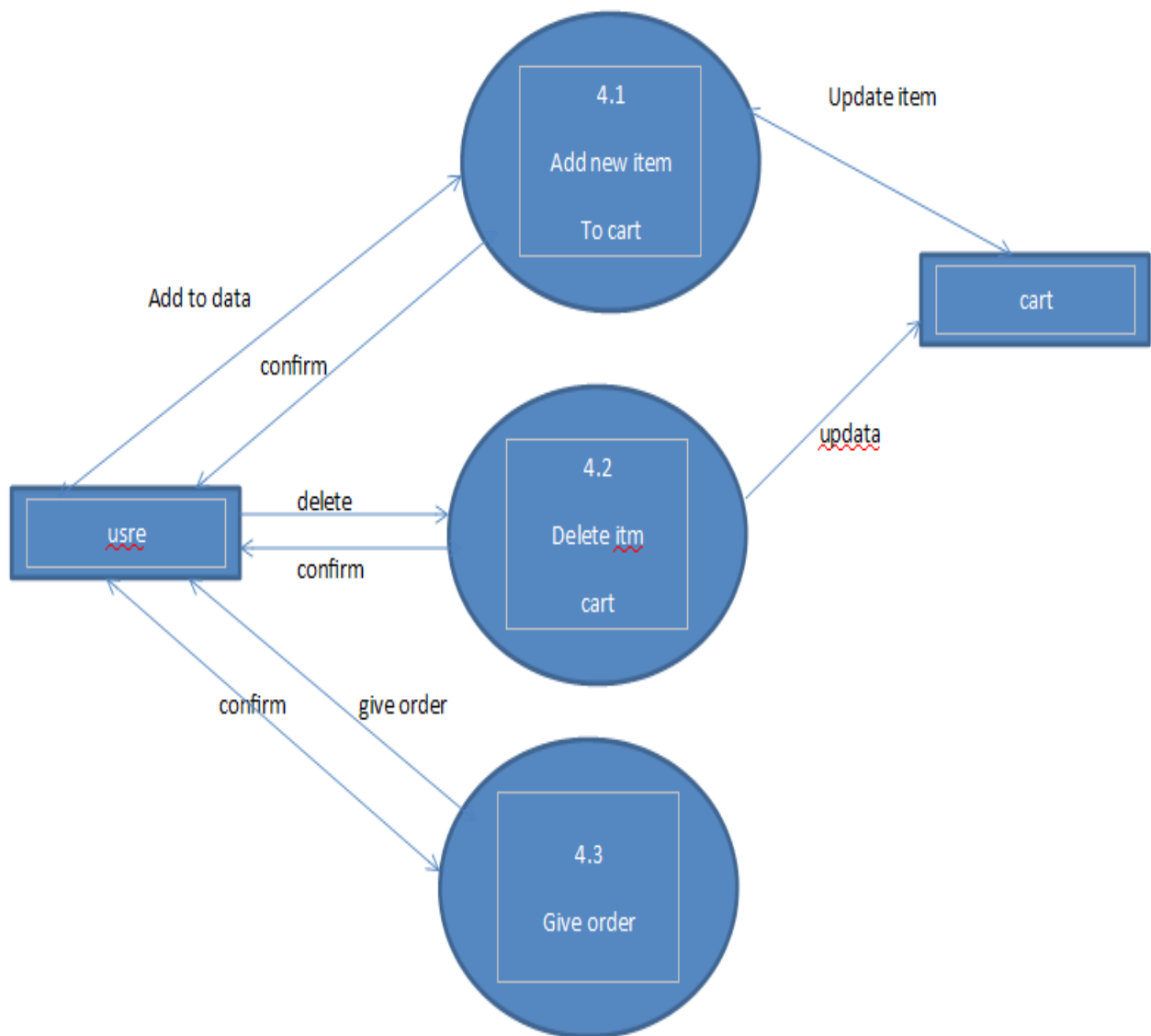


Level 2:

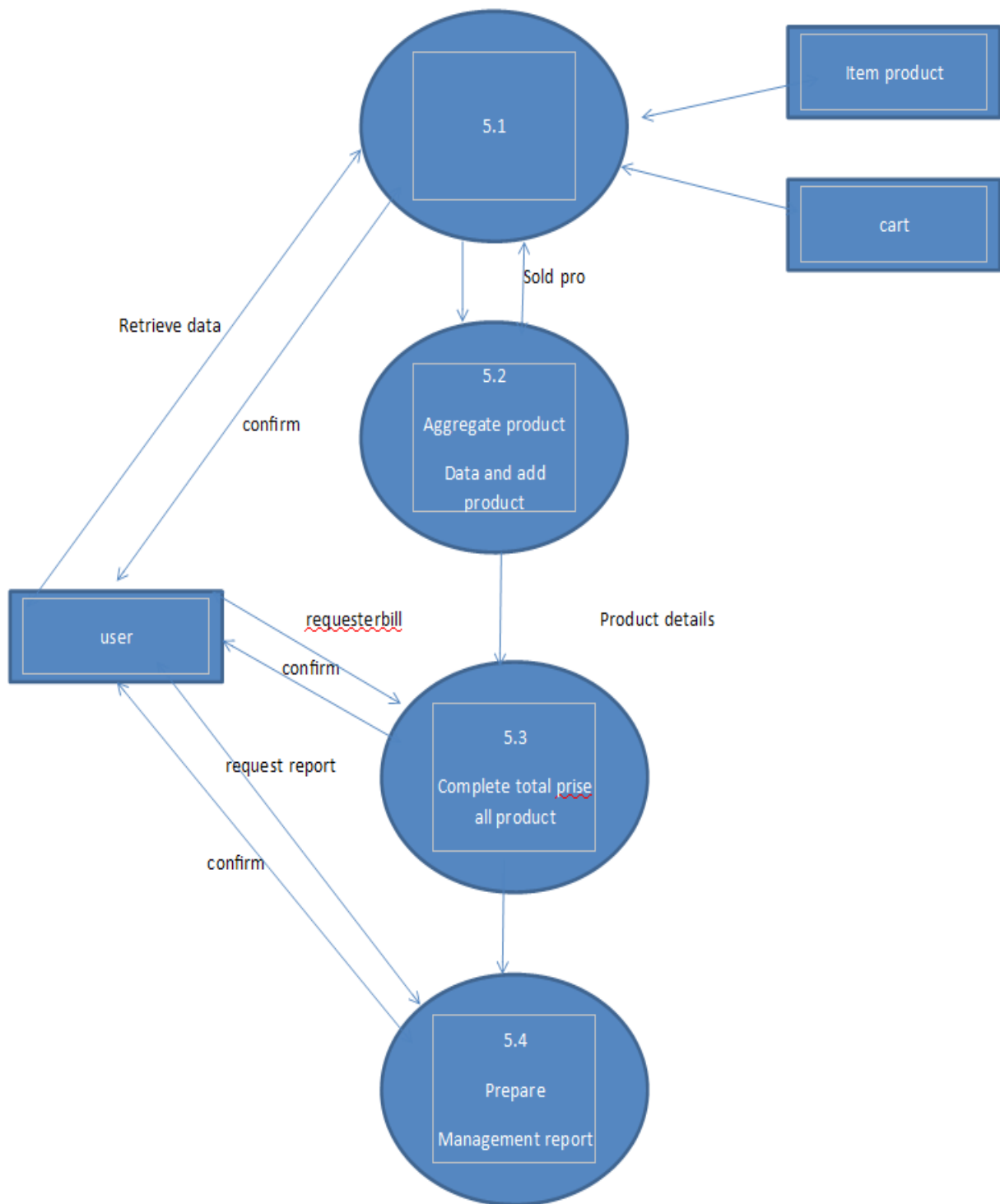
2.0:



4.0:

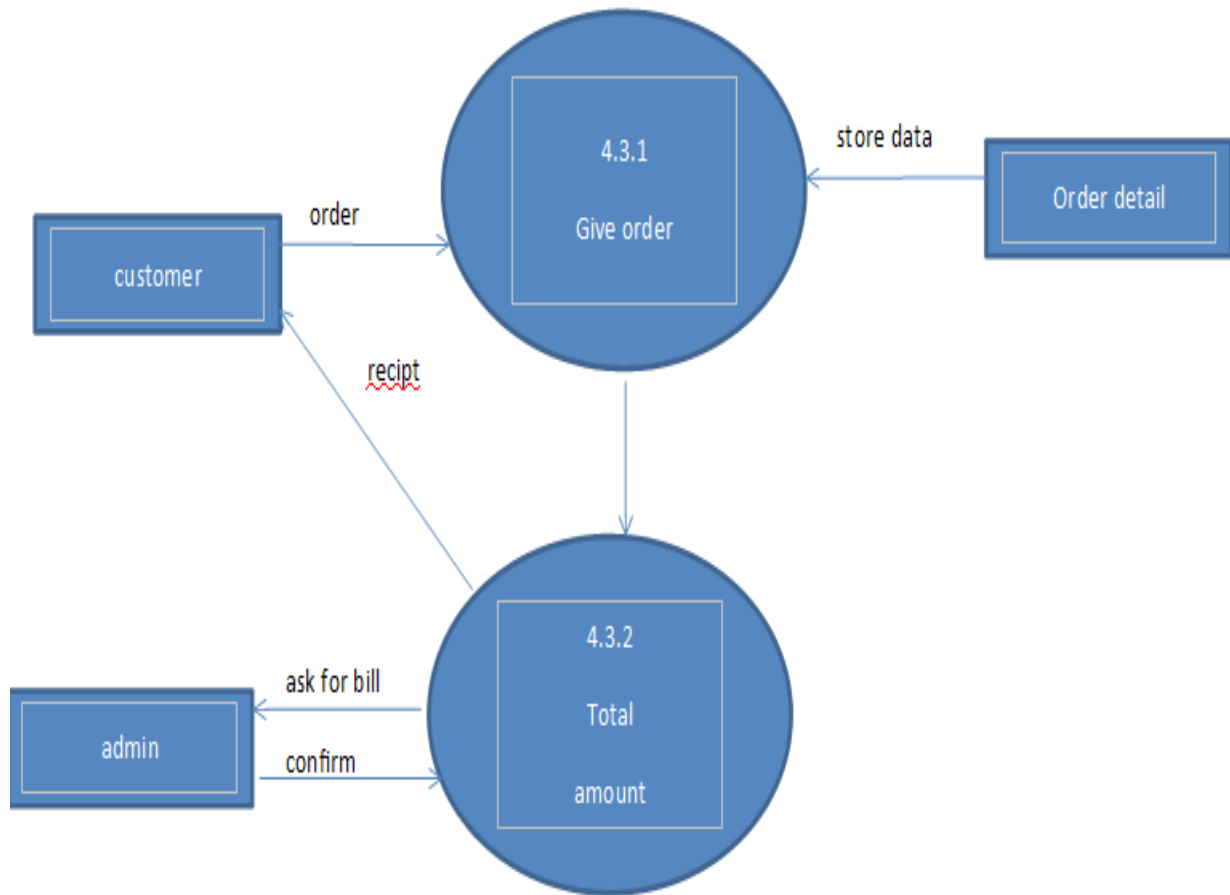


5.0:



Level 3:

4.3:



Entity Relationship diagram:

What is ER Diagrams?

Entity relationship diagram displays the relationships of entity set stored in a database. In other words, we can say that ER diagrams help you to explain the logical structure of databases. At first look, an ER diagram looks very similar to the flowchart. However, ER Diagram includes many specialized symbols, and its meanings make this model unique.

Common Entity Relationship Diagram Symbols

An ER diagram is a means of visualizing how the information a system produces is related. There are five main components of an ERD:

Entities

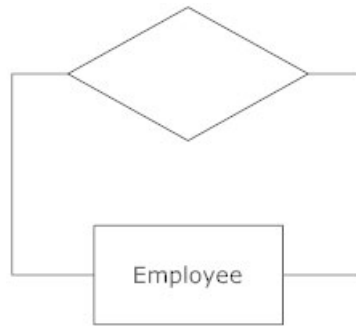
which are represented by rectangles. An entity is an object or concept about which you want to store information. weak entity is an entity that must defined by a foreign key relationship with another entity as it cannot be



Actions

which are represented by diamond shapes, show how two entities share information in the database. In some cases, entities can be self-linked. For example, employees can supervise other employees.





Attributes

which are represented by ovals. A key attribute is the unique, distinguishing characteristic of the entity. For example, an employee's social security number might be the employee's key attribute. A multivalued attribute can have more than one value. For example, an employee entity can have multiple skill

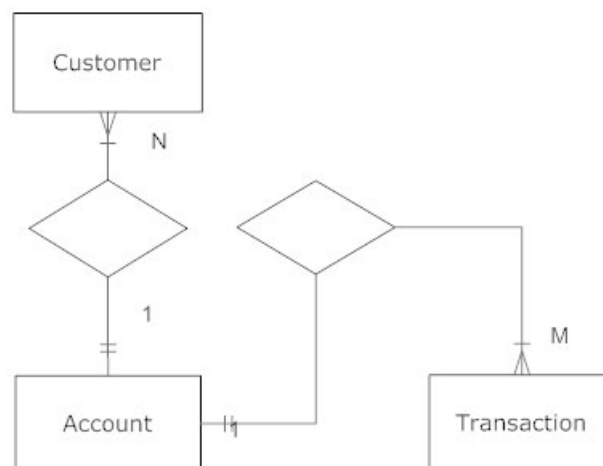


Connecting lines

solid lines that connect attributes to show the relationships of entities in the diagram.

Cardinality

specifies how many instances of an entity relate to one instance of another entity. Ordinality is also closely linked to cardinality. While cardinality specifies the occurrences of a relationship, ordinality describes the relationship as either mandatory or optional. In other words, cardinality specifies the maximum number of relationships and ordinality specifies the absolute minimum number of relationships.



Information Engineering Style



one to one



one to many (mandatory)



many to one (mandatory)



one or more (mandatory)



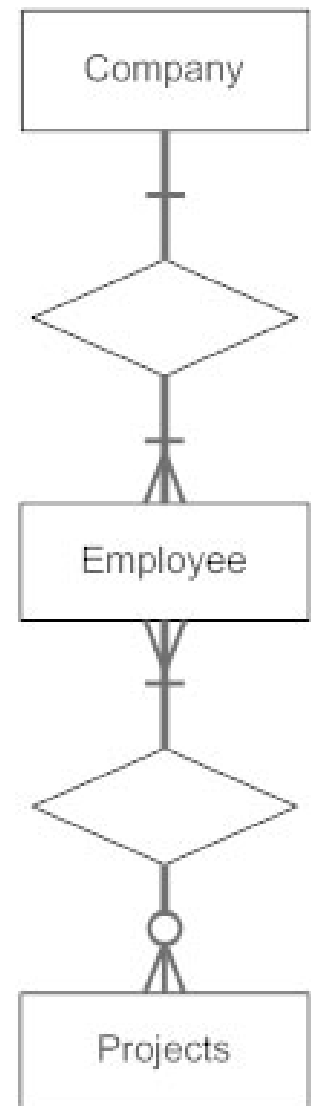
one and only one (mandatory)



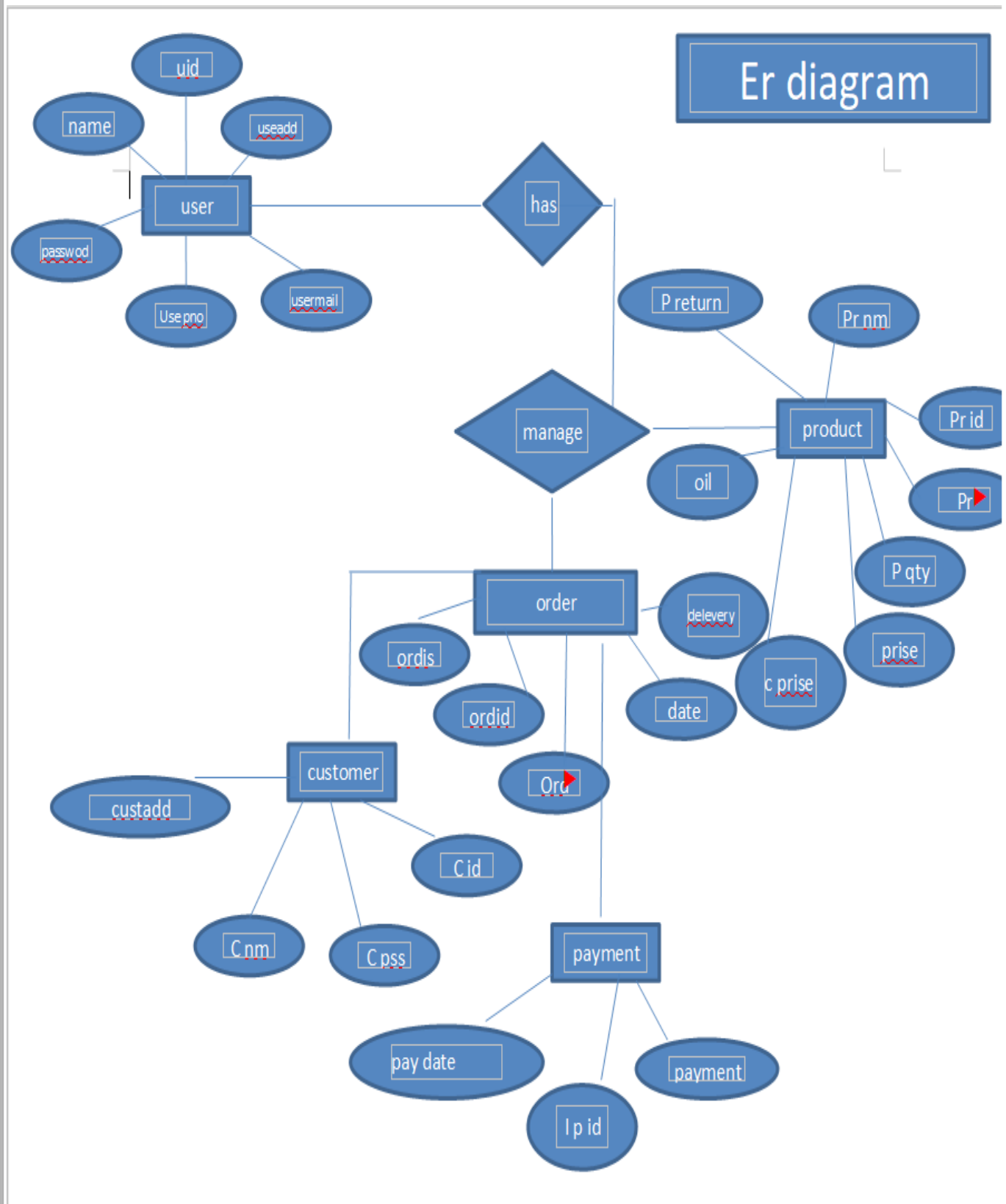
zero or one (optional)



zero or many (optional)



Entity Relationship diagram of BeFresh:



4.Data Dictionary:

Database Name:

- patel_mart

Table Name:

- addcart
- ad_login
- cat_products
- login
- producta
- pro_brand
- pro_category
- pro_contity
- pro_packing
- pro_ret
- sub_category

Admin Table:

ad_login

Structure:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1	ad_id	int(20)		No	None		AUTO_INCREMENT	Change Drop More
<input type="checkbox"/>	2	ad_name	varchar(20)	latin1_swedish_ci	No	None			Change Drop More
<input type="checkbox"/>	3	ad_password	varchar(20)	latin1_swedish_ci	No	None			Change Drop More
<input type="checkbox"/>	4	ad_order	varchar(5)	latin1_swedish_ci	No	None			Change Drop More
<input type="checkbox"/>	5	ad_product	varchar(5)	latin1_swedish_ci	No	None			Change Drop More
<input type="checkbox"/>	6	ad_customer	varchar(5)	latin1_swedish_ci	No	None			Change Drop More
<input type="checkbox"/>	7	ad_admin	varchar(5)	latin1_swedish_ci	No	None			Change Drop More

Data:

+ Options									
			ad_id	ad_name	ad_password	ad_order	ad_product	ad_customer	ad_admin
<input type="checkbox"/>	Edit	Copy	Delete	1	admin	123	1	1	1
<input type="checkbox"/>	Edit	Copy	Delete	2	ad	12	1	1	1

User Table:

addcart

Structure:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1	car_id	int(20)		No	None		AUTO_INCREMENT	Change Drop More
<input type="checkbox"/>	2	car_cust_id	int(20)		No	None			Change Drop More
<input type="checkbox"/>	3	car_pro_id	int(20)		No	None			Change Drop More
<input type="checkbox"/>	4	car_contity	int(20)		No	None			Change Drop More

Data:

+ Options										
		Edit		Copy		Delete	car_id	car_cust_id	car_pro_id	car_contity
<input type="checkbox"/>		Edit		Copy		Delete	1	10	0	0
<input type="checkbox"/>		Edit		Copy		Delete	2	71	111	7
<input type="checkbox"/>		Edit		Copy		Delete	3	71	115	1
<input type="checkbox"/>		Edit		Copy		Delete	4	71	129	2

cat_products

Structure:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1	cat_id	int(20)		No	None		AUTO_INCREMENT	Change Drop More
<input type="checkbox"/>	2	sub_id	int(20)		No	None			Change Drop More
<input type="checkbox"/>	3	cat_name	varchar(50)	latin1_swedish_ci	No	None			Change Drop More

Data:

+ Options					
<input type="checkbox"/>	Edit Copy Delete	cat_id	sub_id	cat_name	
<input type="checkbox"/>	Edit Copy Delete	1	1		
<input type="checkbox"/>	Edit Copy Delete	227	54	Potato, Onion & Tomato	
<input type="checkbox"/>	Edit Copy Delete	228	54	Cucumber & Capsicum	
<input type="checkbox"/>	Edit Copy Delete	229	54	Root Vegetables	
<input type="checkbox"/>	Edit Copy Delete	230	54	Cabbage & Cauliflower	
<input type="checkbox"/>	Edit Copy Delete	231	54	Leafy Vegetables	
<input type="checkbox"/>	Edit Copy Delete	232	54	Beans, Brinjals & Okra	
<input type="checkbox"/>	Edit Copy Delete	233	54	Gourd, Pumpkin, Drumstick	
<input type="checkbox"/>	Edit Copy Delete	234	54	Specialty	
<input type="checkbox"/>	Edit Copy Delete	235	55	Lemon, Ginger & Garlic	
<input type="checkbox"/>	Edit Copy Delete	236	55	Indian & Exotic Herbs	
<input type="checkbox"/>	Edit Copy Delete	237	56	Apples & Pomegranate	
<input type="checkbox"/>	Edit Copy Delete	238	56	Banana, Sapota & Papaya	
<input type="checkbox"/>	Edit Copy Delete	239	56	Kiwi, Melon, Citrus Fruit	
<input type="checkbox"/>	Edit Copy Delete	240	56	Seasonal Fruits	
<input type="checkbox"/>	Edit Copy Delete	241	56	Mangoes	
+					

login

Structure:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1	log_userid	int(11)		No	None		AUTO_INCREMENT	Change Drop More
<input type="checkbox"/>	2	log_username	varchar(20) latin1_swedish_ci		Yes	None			Change Drop More
<input type="checkbox"/>	3	log_phonenumber	varchar(20) latin1_swedish_ci		Yes	None			Change Drop More
<input type="checkbox"/>	4	log_email	varchar(30) latin1_swedish_ci		Yes	None			Change Drop More
<input type="checkbox"/>	5	log_address	varchar(50) latin1_swedish_ci		Yes	None			Change Drop More
<input type="checkbox"/>	6	log_pincode	int(6)		Yes	None			Change Drop More
<input type="checkbox"/>	7	log_state	varchar(20) latin1_swedish_ci		Yes	None			Change Drop More
<input type="checkbox"/>	8	log_distric	varchar(20) latin1_swedish_ci		Yes	None			Change Drop More
<input type="checkbox"/>	9	log_city	varchar(20) latin1_swedish_ci		Yes	None			Change Drop More
<input type="checkbox"/>	10	log_area	varchar(50) latin1_swedish_ci		Yes	None			Change Drop More
<input type="checkbox"/>	11	log_password	varchar(20) latin1_swedish_ci		No	None			Change Drop More
<input type="checkbox"/>	12	log_login	int(1)		No	None			Change Drop More

Data:

+ Options													
← T →													
		log_userid	log_username	log_phonenumber ▲ 1	log_email	log_address	log_pincode	log_state	log_distric	log_city	log_area	log_password	log_login
<input type="checkbox"/>	Edit Copy Delete	71	kevin patel	8140169137	kpatel9899s@gmail.com	Raviratanpark	360005	Rajkot	Rajkot	Rajkot	Near indira circle	5261	0
<input type="checkbox"/>	Edit Copy Delete	72	kevin	9426443408	kevypatel99@gmail.com	gangotripark	66666	rajkot	aa	rajkot	r	21	0

producta

Structure:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1	pro_id	int(20)		No	None		AUTO_INCREMENT	Change Drop More
<input type="checkbox"/>	2	sub_id	int(20)		No	None			Change Drop More
<input type="checkbox"/>	3	pro_simelar	int(20)		No	None			Change Drop More
<input type="checkbox"/>	4	pro_name	varchar(50)	latin1_swedish_ci	No	None			Change Drop More
<input type="checkbox"/>	5	pro_brand	varchar(50)	latin1_swedish_ci	No	None			Change Drop More
<input type="checkbox"/>	6	pro_contype	varchar(20)	latin1_swedish_ci	No	None			Change Drop More
<input type="checkbox"/>	7	pro_contity	int(100)		No	None			Change Drop More
<input type="checkbox"/>	8	pro_pack	varchar(50)	latin1_swedish_ci	No	None			Change Drop More
<input type="checkbox"/>	9	pro_price	float		No	None			Change Drop More
<input type="checkbox"/>	10	pro_sellprice	float		No	None			Change Drop More
<input type="checkbox"/>	11	pro_dis	int(20)		No	None			Change Drop More
<input type="checkbox"/>	12	pro_save	float		No	None			Change Drop More
<input type="checkbox"/>	13	pro_finelprice	float		No	None			Change Drop More
<input type="checkbox"/>	14	pro_detail	varchar(150)	latin1_swedish_ci	No	None			Change Drop More
<input type="checkbox"/>	15	pro_returntime	varchar(20)	latin1_swedish_ci	No	None			Change Drop More
<input type="checkbox"/>	16	pro_stock	int(100)		No	None			Change Drop More
<input type="checkbox"/>	17	pro_img	varchar(150)	latin1_swedish_ci	No	None			Change Drop More
<input type="checkbox"/>	18	pro_imgb	varchar(200)	latin1_swedish_ci	No	None			Change Drop More
<input type="checkbox"/>	19	pro_imgc	varchar(200)	latin1_swedish_ci	No	None			Change Drop More
<input type="checkbox"/>	20	pro_imgd	varchar(200)	latin1_swedish_ci	No	None			Change Drop More

Data:

+ Options																	
	pro_id	sub_id	pro_simelar	pro_name	pro_brand	pro_contype	pro_contity	pro_pack	pro_price	pro_sellprice	pro_dis	pro_save	pro_finelprice	pro_detail	pro_returntime	pro_stock	pro_img
<input type="checkbox"/> Edit Copy Delete	138	0	138	aasdfg	FRESHO		50		50	100	0	20	100	gh	10d	10000	productimg/3_3-fresho-1-hybrid.jpg
<input type="checkbox"/> Edit Copy Delete	137	641	137	asd	FRESHO	gm	50	Box	50	100	10	0	90	h	1d	10000	productimg/fresho-caps-green.jpg
<input type="checkbox"/> Edit Copy Delete	131	228	129	Capsicum - Green	FRESHO	gm	500	Multipack	36	52	20	0	41.6	Leaving a moderately pungent taste on the tongue. ...	1d	10000	productimg/fresho-caps-green.jpg
<input type="checkbox"/> Edit Copy Delete	130	228	129	Capsicum - Green	FRESHO	gm	250	Multipack	16	29	20	0	23.2	Leaving a moderately pungent taste on the tongue. ...	1d	10000	productimg/fresho-caps-green.jpg
<input type="checkbox"/> Edit Copy Delete	129	228	129	Capsicum - Green	FRESHO	kg	1	Multipack	70	99	20	0	79.2	Leaving a moderately pungent taste on the tongue. ...	1d	10000	productimg/fresho-caps-green.jpg
<input type="checkbox"/> Edit Copy Delete	128	227	128	New Potato	FRESHO	kg	1	Multipack	10	19	20	0	15.2	Fresho New potatoes are freshly picked from the be...	1d	10000	productimg/fresho-new-hybrid.jpg
<input type="checkbox"/> Edit Copy Delete	127	227	127	Tomato - Hybrid	Fresho Premium	pcs	8	Multipack	15	28	20	0	22.4	Tomato - Hybrid	1d	10000	productimg/fresho-prem-hybrid.jpg

pro_brand

Structure:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1	bra_id	int(20)		No	None		AUTO_INCREMENT	Change Drop More
<input type="checkbox"/>	2	bra_name	varchar(40)	latin1_swedish_ci	No	None			Change Drop More

Data:

+ Options	
	bra_id bra_name
<input type="checkbox"/>	Edit Copy Delete 15
<input type="checkbox"/>	Edit Copy Delete 16 FRESHO
<input type="checkbox"/>	Edit Copy Delete 17 Fresho Premium

pro_category

Structure:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1	cat_id	int(20)		No	None		AUTO_INCREMENT	Change Drop More
<input type="checkbox"/>	2	cat_name	varchar(20)	latin1_swedish_ci	No	None			Change Drop More

Data:

+ Options	
	cat_id cat_name
<input type="checkbox"/>	Edit Copy Delete 1
<input type="checkbox"/>	Edit Copy Delete 27 Fruits & Vegetables
<input type="checkbox"/>	Edit Copy Delete 28 Foodgrains, Oil & Ma
<input type="checkbox"/>	Edit Copy Delete 29 Bakery, Cakes & Dair
<input type="checkbox"/>	Edit Copy Delete 30 Beverages
<input type="checkbox"/>	Edit Copy Delete 31 Snacks & Branded Foo
<input type="checkbox"/>	Edit Copy Delete 32 Beauty & Hygiene
<input type="checkbox"/>	Edit Copy Delete 33 Cleaning & Household
<input type="checkbox"/>	Edit Copy Delete 34 Kitchen, Garden & Pe
<input type="checkbox"/>	Edit Copy Delete 35 Eggs, Meat & Fish
<input type="checkbox"/>	Edit Copy Delete 36 Gourmet & World Food
<input type="checkbox"/>	Edit Copy Delete 37 Baby Care
<input type="checkbox"/>	Edit Copy Delete 38 View All

pro_contity

Structure:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1	con_id	int(20)		No	None		AUTO_INCREMENT	Change Drop More
<input type="checkbox"/>	2	con_name	varchar(50)	latin1_swedish_ci	No	None			Change Drop More

Data:

+ Options

				con_id	con_name
<input type="checkbox"/>	Edit	Copy	Delete	1	
<input type="checkbox"/>	Edit	Copy	Delete	8	kg
<input type="checkbox"/>	Edit	Copy	Delete	9	gm
<input type="checkbox"/>	Edit	Copy	Delete	10	pcs

pro_packing

Structure:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1	pac_id	int(20)		No	None		AUTO_INCREMENT	Change Drop More
<input type="checkbox"/>	2	pac_name	varchar(20)	latin1_swedish_ci	No	None			Change Drop More

Data:

+ Options

				pac_id	pac_name
<input type="checkbox"/>	Edit	Copy	Delete	1	
<input type="checkbox"/>	Edit	Copy	Delete	5	Multipack
<input type="checkbox"/>	Edit	Copy	Delete	6	Box

pro_ret

Structure:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1	ret_id	int(20)		No	None		AUTO_INCREMENT	Change Drop More
<input type="checkbox"/>	2	ret_name	varchar(20)	latin1_swedish_ci	No	None			Change Drop More

Data:

+ Options		
<input type="checkbox"/>	Edit Copy Delete	ret_id ret_name
<input type="checkbox"/>	Edit Copy Delete	1
<input type="checkbox"/>	Edit Copy Delete	3 noreturn
<input type="checkbox"/>	Edit Copy Delete	7 1d
<input type="checkbox"/>	Edit Copy Delete	8 2d
<input type="checkbox"/>	Edit Copy Delete	9 3d
<input type="checkbox"/>	Edit Copy Delete	10 4d
<input type="checkbox"/>	Edit Copy Delete	11 5d
<input type="checkbox"/>	Edit Copy Delete	12 6d
<input type="checkbox"/>	Edit Copy Delete	13 7d
<input type="checkbox"/>	Edit Copy Delete	14 d8
<input type="checkbox"/>	Edit Copy Delete	15 9d
<input type="checkbox"/>	Edit Copy Delete	16 10d

sub_category

Structure:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1	sub_id	int(20)		No	None		AUTO_INCREMENT	Change Drop More
<input type="checkbox"/>	2	cat_id	int(20)		No	None			Change Drop More
<input type="checkbox"/>	3	sub_name	varchar(50)	latin1_swedish_ci	No	None			Change Drop More

Data:

+ Options					sub_id	cat_id	sub_name
<input type="checkbox"/>	Edit	Copy	Delete	1	2		
<input type="checkbox"/>	Edit	Copy	Delete	54	27	Fresh Vegetables	
<input type="checkbox"/>	Edit	Copy	Delete	55	27	Herbs & Seasonings	
<input type="checkbox"/>	Edit	Copy	Delete	56	27	Fresh Fruits	
<input type="checkbox"/>	Edit	Copy	Delete	57	27	Cuts & Sprouts	
<input type="checkbox"/>	Edit	Copy	Delete	58	27	Exotic Fruits & Veggies	
<input type="checkbox"/>	Edit	Copy	Delete	59	27	Organic Fruits & Vegetables	
<input type="checkbox"/>	Edit	Copy	Delete	60	27	Flower Bouquets, Bunche	
<input type="checkbox"/>	Edit	Copy	Delete	61	28	Atta, Flours & Sooji	
<input type="checkbox"/>	Edit	Copy	Delete	62	28	Rice & Rice Products	
<input type="checkbox"/>	Edit	Copy	Delete	63	28	Dals & Pulses	
<input type="checkbox"/>	Edit	Copy	Delete	64	28	Salt, Sugar & Jaggery	
<input type="checkbox"/>	Edit	Copy	Delete	65	28	Edible Oils & Ghee	
<input type="checkbox"/>	Edit	Copy	Delete	66	28	Organic Staplesv	
<input type="checkbox"/>	Edit	Copy	Delete	67	28	Masalas & Spices	
<input type="checkbox"/>	Edit	Copy	Delete	68	28	Dry Fruits	
<input type="checkbox"/>	Edit	Copy	Delete	69	29	Dairy	
<input type="checkbox"/>	Edit	Copy	Delete	70	29	Breads & Buns	
<input type="checkbox"/>	Edit	Copy	Delete	71	29	Cookies, Rusk & Khari	
<input type="checkbox"/>	Edit	Copy	Delete	72	29	Gourmet Breads	
<input type="checkbox"/>	Edit	Copy	Delete	73	29	Bakery Snacks	
<input type="checkbox"/>	Edit	Copy	Delete	74	29	Ice Creams & Desserts	
<input type="checkbox"/>	Edit	Copy	Delete	75	29	Cakes & Pastries	
Console							

5. INPUT – OUTPUT DESIGN (SCREEN SHOTS)

6. ABOUT THE TOOLS

Front-End Tool:

HTML:

What is HTML?

- First developed by Tim Berners-Lee in 1990,
- HTML is short for Hypertext Markup Language.
- HTML is used to create electronic documents (called pages) that are displayed on the World Wide Web.
- Each page contains a series of connections to other pages called hyperlinks.
- Every web page you see on the Internet is written using one version of HTML code or another.
- HTML code ensures the proper formatting of text and images so that your Internet browser may display them as they are intended to look.
- Without HTML, a browser would not know how to display text as elements or load images or other elements.
- HTML also provides a basic structure of the page, upon which Cascading Style Sheets are overlaid to change its appearance.
- One could think of HTML as the bones (structure) of a web page, and CSS as its skin (appearance)

Features of HTML:

- It is a very easy and simple language. It can be easily understood and modified.
- It is very easy to make effective presentation with HTML because it has a lot of formatting tags.
- It is a markup language so it provides a flexible way to design web pages along with the text.
- It facilitates programmers to add link on the web pages (by html anchor tag) , so it enhances the interest of browsing of the user.
- It is platform-independent because it can be displayed on any platform like Windows, Linux and Macintosh etc.
- It facilitates the programmer to add Graphics, Videos, and Sound to the web pages which makes it more attractive and interactive.

HTML Text Tags:

<p>, <h1>, <h2>, <h3>, <h4>, <h5>, <h6>,
, , <abbr>, <acronym>, <address>,
<bdo>, <blockquote>, <cite>, <q>, <code>, <ins>,
, <dfn>, <kbd>, <pre>, <samp>, <var> and

HTML Link Tags:

<a> and <base>

HTML Image and Object Tags:

, <area>, <map>, <param> and <object>

HTML List Tags:

, , , <dl>, <dt> and <dd>

HTML Table Tags:

table, tr, td, th, tbody, thead, tfoot, col, colgroup and caption

HTML Form Tags:

form, input, textarea, select, option, optgroup, button, label, fieldset and legend

HTML Scripting Tags:

script and noscript

CSS:

Stands for "Cascading Style Sheet." Cascading style sheets are used to format the layout of Web pages. They can be used to define text styles, table sizes, and other aspects of Web pages that previously could only be defined in a page's HTML.

Cascading Style Sheets were developed as a means for creating a consistent approach to providing style information for web documents.

- CSS is an acronym which means Cascading Style Sheets
- The main purpose of CSS is to define the presentation of HTML elements
- All styles are defined and stored in Style sheets
- After storing styles in a separate style sheet files, these are added into HTML documents.
- Defining the style outside the html document helps in saving lots of work and time
- CSS allows you to define more than one style in one style sheet; known as the cascading of styles.

2 Back-End Tools:

PHP:

PHP is an "HTML-embedded scripting language" primarily used for dynamic Web applications. The first part of this definition means that PHP code can be interspersed with HTML, making it simple to generate dynamic pieces of Web pages on the fly.

As a scripting language, PHP code requires the presence of the PHP processor. PHP code is normally run in plain-text scripts that will only run on PHP-enabled computers (conversely programming languages can create standalone binary executable files, a.k.a. programs).

PHP takes most of its syntax from C, Java, and Perl. It is an open source technology and runs on most operating systems and with most Web servers.

PHP was written in the C programming language by Rasmus Lerdorf in 1994 for use in monitoring his online resume and related personal information. For this reason, PHP originally stood for "Personal Home Page". Leadoff combined PHP with his own Form Interpreter, releasing the combination publicly as PHP/FI (generally referred to as PHP 2.0) on June 8, 1995.

Two programmers, Zeev Suraski and Andi Gutmans, rebuilt PHP's core, releasing the updated result as PHP/FI 2 in 1997. The acronym was formally changed to PHP: Hypertext Preprocessor, at this time. (This is an example of a recursive acronym: where the acronym itself is in its own definition.)

In 1998, PHP 3 was released, which was the first widely used version. PHP 4 was released in May 2000, with a new core, known as the Zend Engine 1.0. PHP 4 featured improved speed and reliability over PHP 3.

In terms of features, PHP 4 added references, the Boolean type, COM support on Windows, output buffering, many new array functions, expanded object-oriented programming, inclusion of the PCRE library, and more. Maintenance releases of PHP 4 are still available, primarily for security updates.

PHP 5 was released in July 2004, with the updated Zend Engine 2.0. Among the many new features in PHP 5 are:

- improved object-oriented programming
- embedded SQLite
- support for new MySQL features (see the image at right)
- Exception handling using a try..catch structure
- integrated SOAP support (see the image at right)
- the Filter library (in PHP 5.1)

- better XML tools
- Iterators and much more.

MySQL:

MySQL is the database construct that enables PHP and Apache to work together to access and display data in a readable format to a browser.

It is a Structured Query Language server designed for heavy loads and processing of complex queries. As a relational database system, MySQL allows many different tables to be joined together for maximum efficiency and speed.

- Multiple CPUs usable through kernel threads
- Multi-Platform operation
- Numerous column types cover virtually every type of data
- Group functions for mathematical calculations and sorting
- Commands that allow information about the database to be easily and succinctly shown to administrator
- Function names that do not affect table or column names
- A password and user verification system for added security
- MYSQL is the perfect choice for providing data via the internet because of its ability to handle heavy loads and its advanced security measures.

7. CONCLUSION (FUTURE SCOPE, LIMITATIONS)

7.1 Future scope:

- We can provide feedback.

7.2 Limitation:

- No input data (form validation) validation.

Any work may not always be perfect:

There may be some defects or errors. We have taken enough care to make the project user friendly and more interactive.

Major focus is to save time, because time is money.

Any suggestion to make this project more useful is always most welcome.

-
-
-
-

Thank
You

www.spreading-joy.org

For these amazing Little things that often go unappreciated