



جامعة المجمعة  
Majmaah University

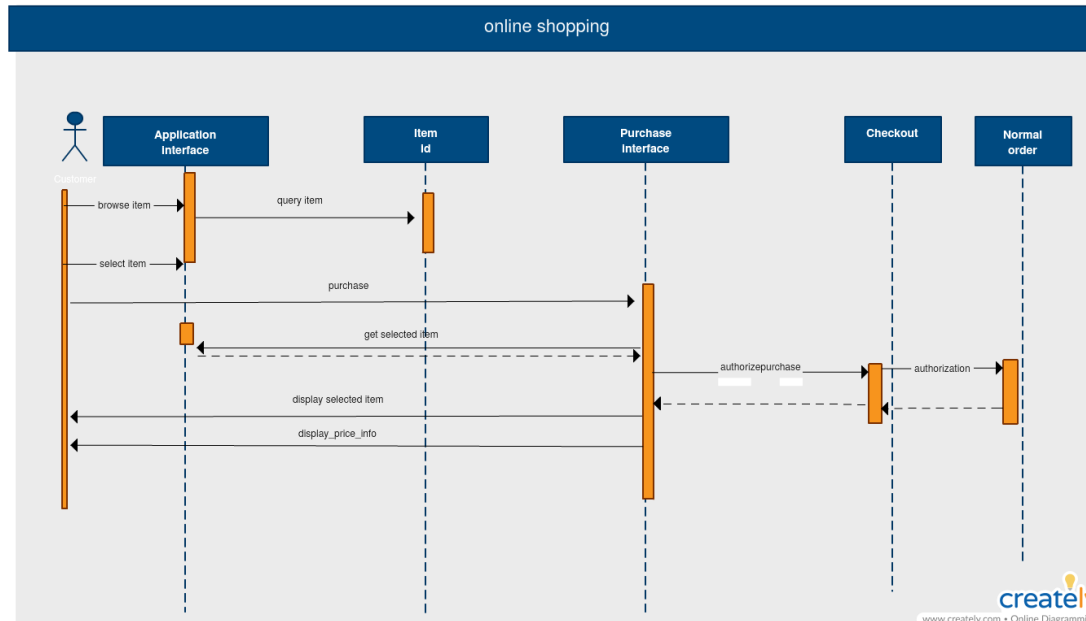
# Mashups multi stores management system

by :Mohamed Nasser Alanazi

---

--

Chapter 1.....	5
1.0    INTRODUCTION .....	5
1.1    PROBLEMS Statement.....	5
1.2    Objectives .....	6
1.3    Scope.....	6
2 Chapter 2 .....	8
2.1    Introduction .....	8
2.0    Data Flow Diagram (DFD).....	8
2.1.1    Context Diagram.....	8
2.1.2    Overview Diagram (Level o).....	9
2.1    Use case diagram .....	10
2.2    ERD diagram.....	11
2.3    Class Diagram .....	13
2.4    Object Diagram .....	15
2.5    Sequence Diagram.....	17



3	Chapter 3 : System Design .....	17
3.0	Table of contents .....	17
3.1	admin.....	18
3.2	categories.....	19
3.3	customer.....	20
3.4	integerated_system.....	21
3.5	stores .....	22
3.6	Expected Interface.....	1
4	Chapter 4.....	3
4.0	Literature Review.....	3

5	Conclusion.....	8
	References .....	8
6	Mashup .....	9

## CHAPTER 1

---

### 1.0 INTRODUCTION

Combining data and code from third-party sources has enabled a new wave of web mashups that add creativity and functionality to web applications. However, browsers are poorly designed to pass data between domains, often forcing web developers to abandon security in the name of functionality. To address this deficiency, we developed the web system, a cross-platform communication mechanism that allows efficient communication across multi stores on the web like Alipay,amazon,souq.com without sacrificing security. Our prototype requires only a small JavaScript library, and works across all major browsers. We believe that our script can serve as a new secure communication primitive for web mashups.

### 1.1 PROBLEMS STATEMENT

A mashup multi store management system introduce any easy method for e-commerce is so far of the normal method that will need to have license from the other stores and you have to going to each of them to give you permission and make so hug advertisement to publicity your website to carriage the seller for participate in your website and all this need much money and effort and multiple of people to handle all this work .

But when the mashup started as service and according to the api that come from the other resources like google, amazon,souq.com .....etc.

It become easier to create new multistore based on the others store that is famous

This method dependent at that integrate our system with the other system for stores

---

And publish their products on web so it will be like that its our products

And when the user clicks buy product system will give him about 15 days till he receive his order

So, what will happen after that?

This the important question, from the integration it has 2 option

That integration will provide the method for buy the product from its resource without the buyer knew that this product from another resources

So, he will buy and pay for us the price included the website commission so the admin will get all the requests and make the order from the resource.

With that method the website will be work like commissioner between the resource and the buyer.

## **1.2 OBJECTIVES**

Mashups multi stores management system (MMS) system provides businesses with the ability to computerize, systematize and correlate retail information. Where cash registers, including complex register systems, have limited information collection capacity, MMS systems can gather, store and return detailed reports on inventory trends and customer information. Additionally, MMS systems more easily integrate with numerous sales and ordering systems, including mail or online ordering systems used in conjunction with in-person sales.

## **1.3 SCOPE**

In recording the purchases made by customers, the following information are stored: (1) A unique transaction number assigned to every transaction,

(2) The name of the customer

(3) The name of the store

(4) The items purchased and their prices

(5) The date and time of the transaction

---

(6) Discounts applied to the transaction (if any)

(7) The total price of all the items bought or ordered after applying the discounts (if any).

For the database of items, the system will only store information about the items offered in the lounge.

Data regarding the number of stock for each item is not covered since an inventory system is no longer in our scope. The following information are stored for each item in the database:

(1) The unique product identification number

(2) The product name

(3) Item type

(4) Category

(5) The price the item was bought and The selling price.

---

## **2 CHAPTER 2**

---

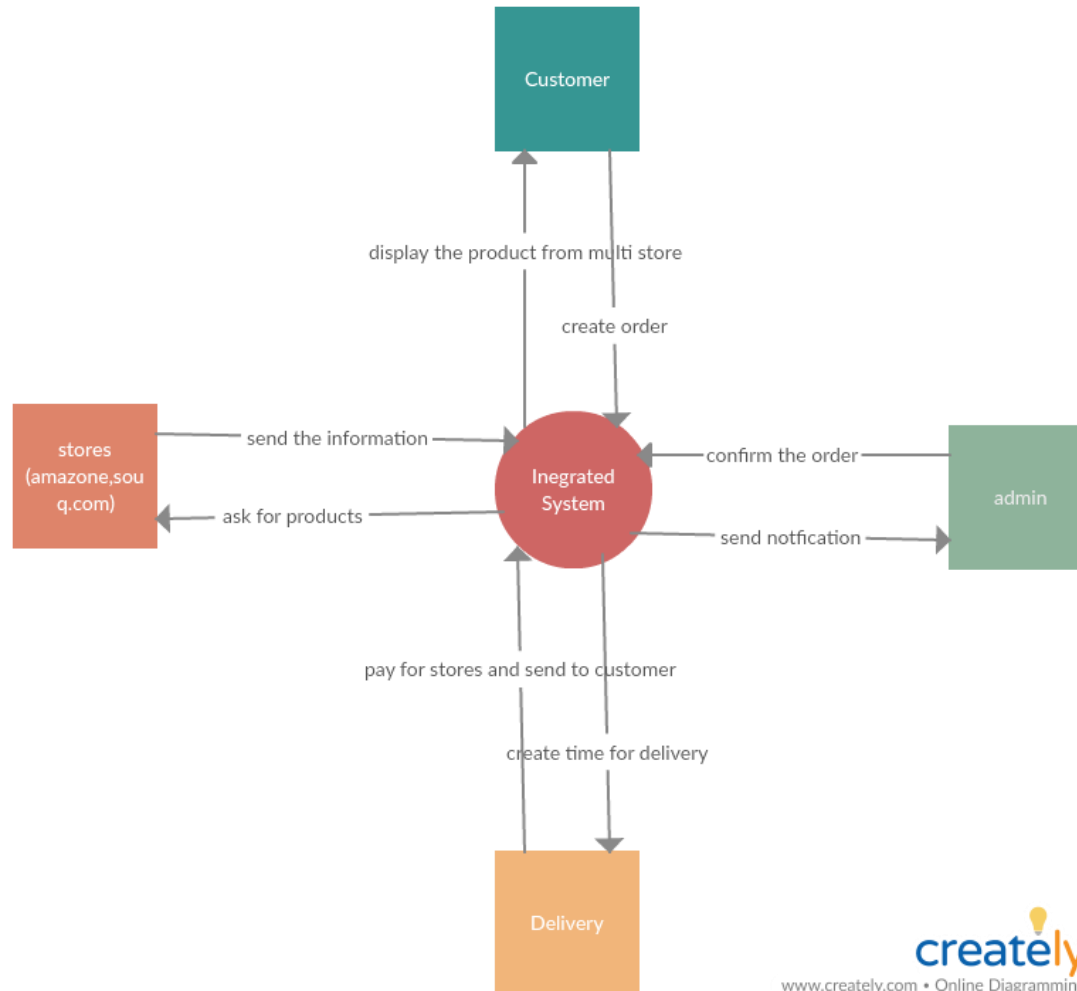
### **2.1 INTRODUCTION**

To analyze the system we draw many diagrams to explain the system, description of data flow diagram, the context diagram, overview diagram (level 0), detailed DFDs, entity relationship diagram, and class diagram. It is the most crucial phase in the development of a system. In systems design, the design functions and operations are described in detail.

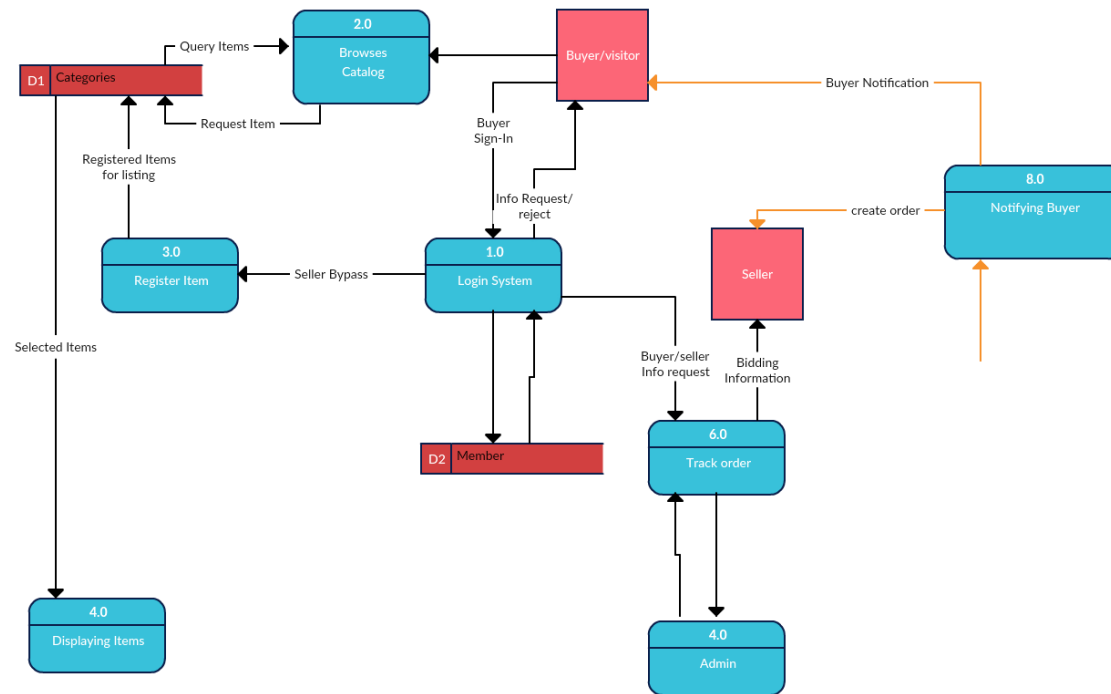
### **2.0 DATA FLOW DIAGRAM (DFD)**

#### **2.1.1 Context Diagram**

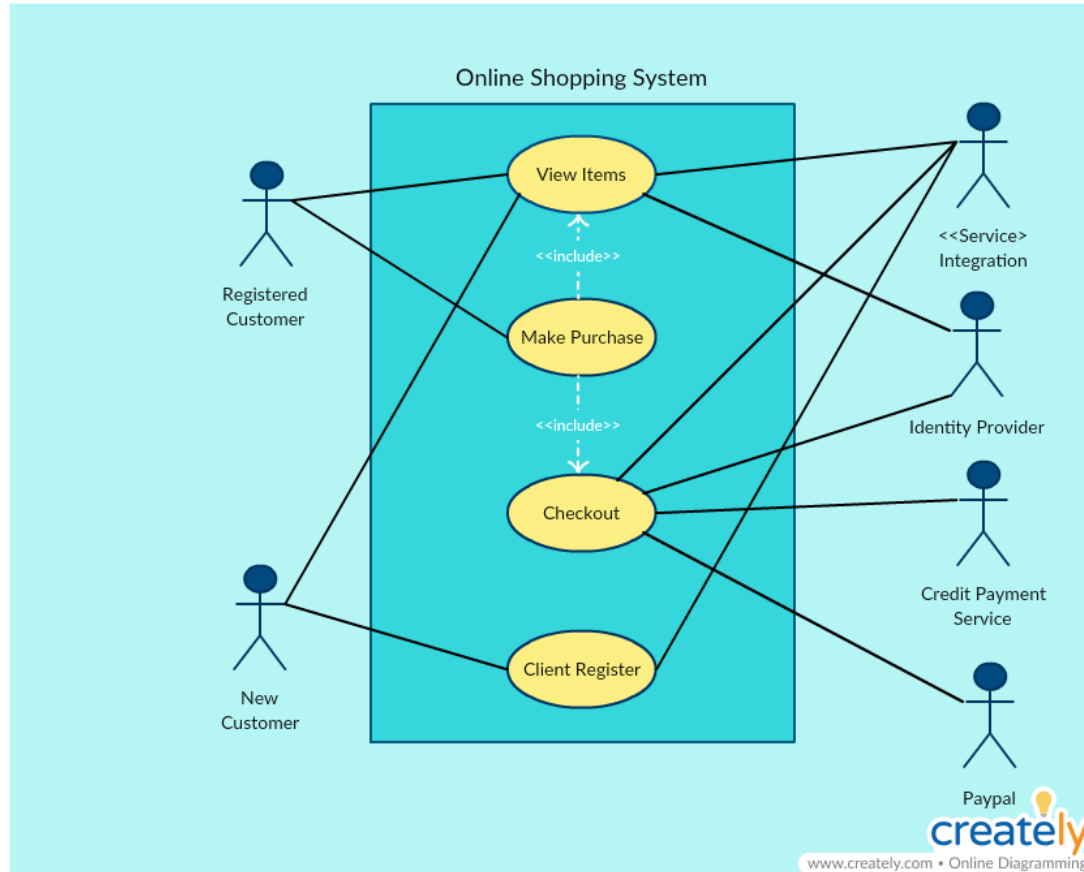




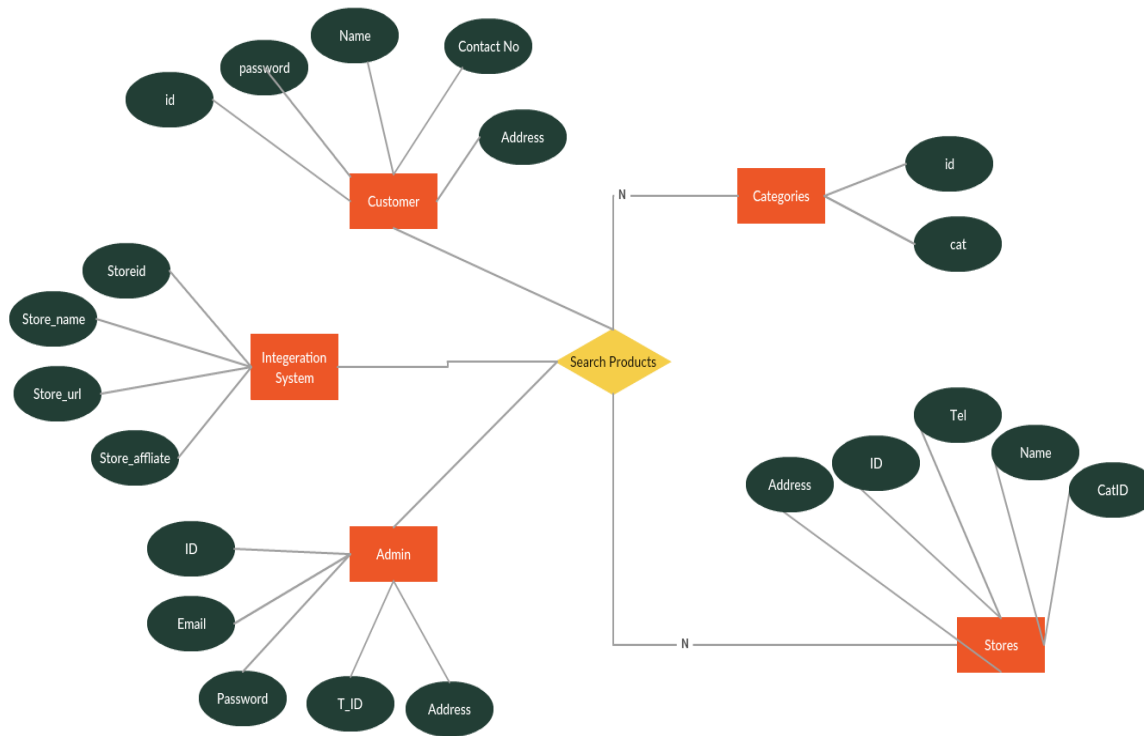
### 2.1.2 Overview Diagram (Level 0)



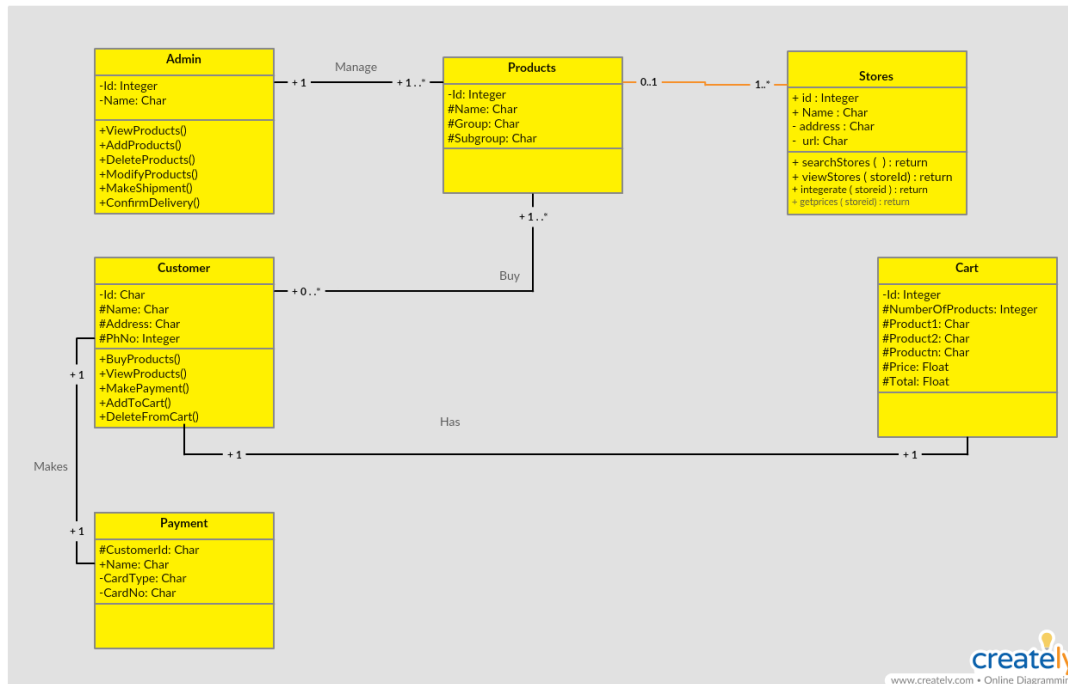
## 2.1 USE CASE DIAGRAM



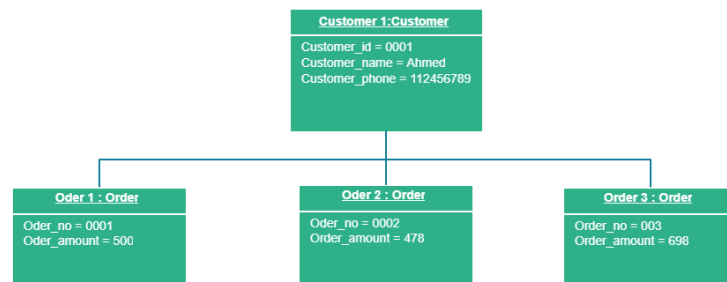
## 2.2 ERD DIAGRAM



**2.3 CLASS DIAGRAM**



## 2.4 OBJECT DIAGRAM

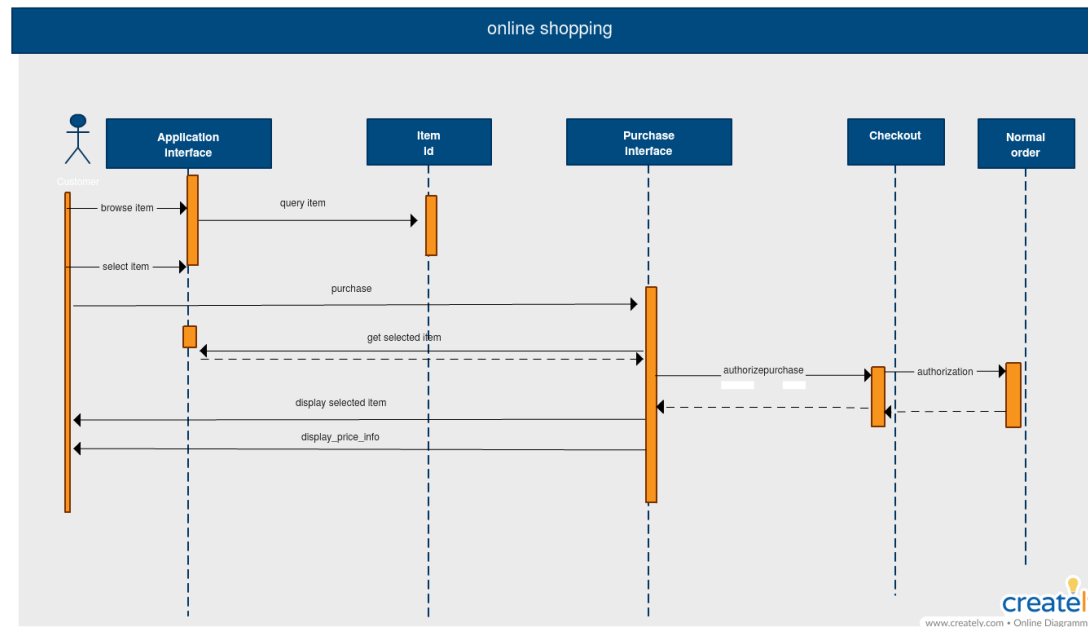


---

--



## 2.5 SEQUENCE DIAGRAM



## 3 CHAPTER 3 : SYSTEM DESIGN

### 3.0 TABLE OF CONTENTS

1 admin Page number: 2

2 categories Page number: 3

- 3 customer Page number: 4
- 4 integerated\_system Page number: 5
- 5 stores Page number: 6
- 6 Relational schema Page number: 7

3.1 ADMIN

Creation: Apr 05, 2019 at 12:35 AM

Column	Type	Attributes	Null	Default	Extra	Links to	Comments	MIME
id	int(11)		No		auto_increment			
email	varchar(300)		No					
password	varchar(50)		No					
T_id	int(11)		No					
address	text		No					

3.2 CATEGORIES

Creation: Apr 05, 2019 at 12:36 AM

Column	Type	Attributes	Null	Default	Extra	Links to	Comments	MIME
id	int(11)		No		auto_increment			
cat	varchar(100)		No					

3.3 CUSTOMER

Creation: Apr 05, 2019 at 12:32 AM

Column	Type	Attributes	Null	Default	Extra	Links to	Comments	MIME
id	int(11)		No		auto_increment			
username	varchar(100)		No					
password	varchar(50)		No					
name	int(100)		No					
contact_no	varchar(100)		No					
address	text		No					

3.4 INTEGRATED\_SYSTEM

Creation: Apr 05, 2019 at 12:34 AM

Column	Type	Attributes	Null	Default	Extra	Links to	Comments	MIME
id	int(11)		No		auto_increment			
store_id	int(11)		No					
store_name	varchar(100)		No					
store_url	varchar(300)		No					
store_affiliate	text		No					

3.5 STORES

Creation: Apr 05, 2019 at 12:38 AM

Column	Type	Attributes	Null	Default	Extra	Links to	Comments	MIME
id	int(11)		No		auto_increment			
name	varchar(100)		No					
cat_id	int(11)		No					
tel	varchar(100)		No					
address	text		No					



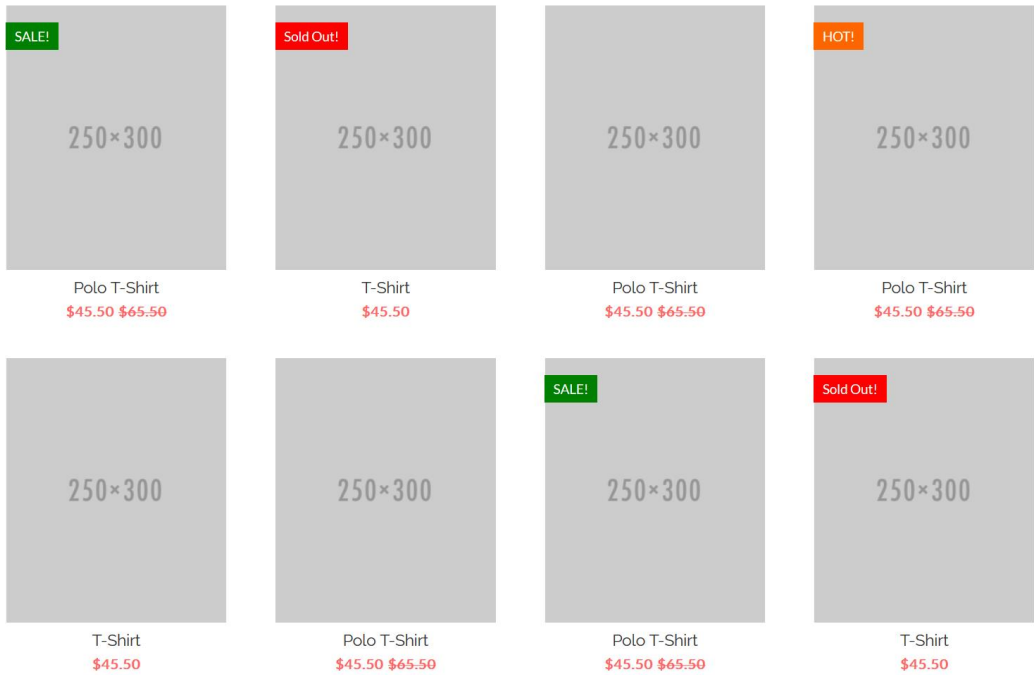
---

## 3.6 EXPECTED INTERFACE

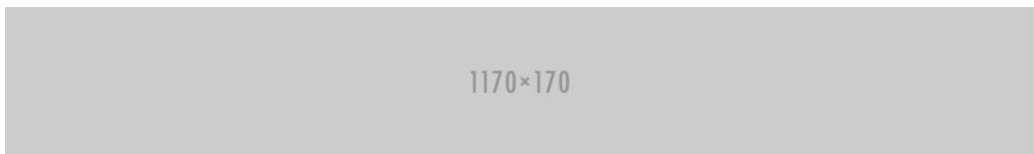




MEN WOMEN SPORTS ELECTRONICS



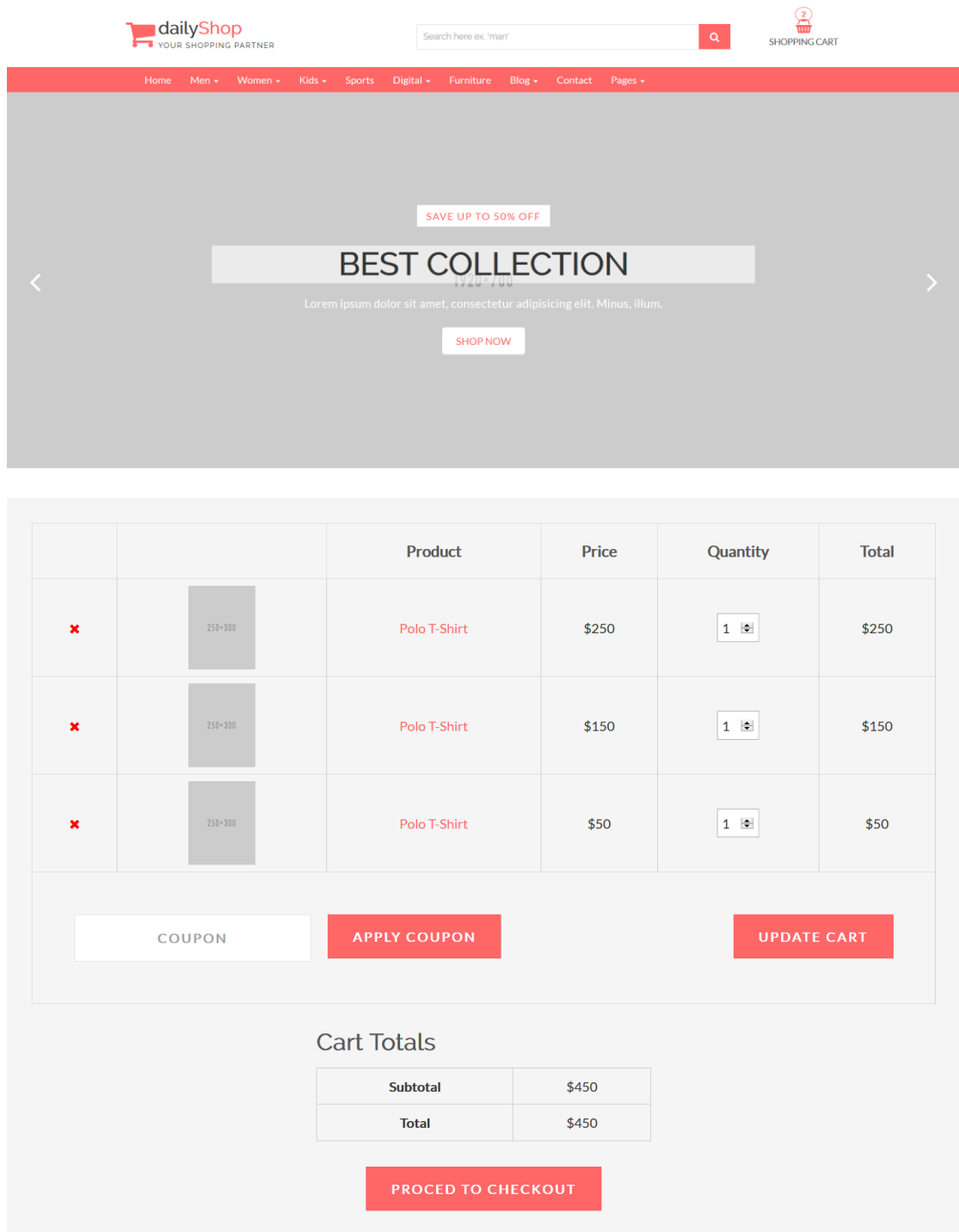
Browse all Product →



POPULAR FEATURED LATEST



Browse all Product →



## 4 CHAPTER 4

### 4.0 LITERATURE REVIEW

Currently there's a lot of buzz surrounding web mashups in industry and academic communities.

The term mashup was actually derived from the music industry, where disc jockeys' remix original content from various artists to create new material [9].

---

Therefore, the idea behind a web mashup is to synthesize new information by reusing and combining existing content from disparate information sources. Mashups allow end-users to combine information and knowledge from a plethora of sources, and integrate them into customized, goal-oriented applications.

Web 2.0 mashups have emerged as a trend in website design that focuses on synthesizing new content by combining data and information from multiple sources in a unique way.

As such, mashups have proliferated because of how quickly they can be created, given that they're composed of preexisting data

For example, Huynh et al.

found that all users were able to learn their mashup interface and complete a mashup within 45 minutes. The expansion of mashups can be viewed from three different perspectives. The first area of expansion is simply in the number of new mashups that are continually being created. Since their building blocks are existing data sources, building new mashups is a relatively quick process.

Secondly, the number of sources being used to build mashups has also greatly expanded. Initially, many of the resources used in mashups were publicly available data (e.g. web services and RSS feeds), but now mashups are also being composed of databases, data warehouses, and even legacy systems

The number of mashup research domains is another area of expansion found in mashup literature, for example, Hoyer and Fischer distinguish between consumer and enterprise mashups. The remainder of this section presents the results of the literature review and categorization process described in the previous section.

the technical challenges involved in mashing legacy systems is much different than mashing multiple RSS feeds. As is the case of enterprise mashups, where the back-end resources being utilized by the mashup are legacy systems or databases, access control becomes a legitimate concern for several reasons [68]. First, it can be difficult to include backend systems in the mashup because of the access control systems that they have in place. Often times the workaround is for the user to give the mashup their credentials and for the mashup to go on .

A fundamental feature of mashup applications is that they allow users to combine data from a variety of sources. However, when the underlying data resources are available from different providers, security issues can arise. Cross communication between multiple back end systems poses a problem in mashup development because the current generation of web browsers cannot adequately support it. Currently, browsers are designed around a "same origin" policy, where data (or code) from one source can only interact with content from its same origin. Consequently, mashup developers must choose between security and functionality, which often results in giving uncontrolled cross domain execution through the use of <script> tags and extending the browser with plug-ins for cross domain interactions [17]. This sort of cross-site scripting introduces a computer security vulnerability which might enable malicious attackers to inject client-side script into web pages viewed by other users. It can also allow one back end system complete control over another [18]. Figure 1 illustrates a

warning message from a Google personal page, where a gadget requires “inlining” which provides the gadget more control over the full Google page because it is not wrapped in an “inline” frame. This allow the gadget to change aspects of the Google page and may also allow the gadget to access the user's Google account .

### 3.1 Mashup Frameworks

This section discusses mashups from an aggregated scope, in review of frameworks that address the different attributes of mashups collectively. Frameworks are extremely important in the initial phases of a technology's lifecycle as they provide an initial foundation that can be applied by early adopters in industry and that can be evaluated (or extended) by researchers in academia. Since mashups are designed by end users for specific contexts with specific purposes, they tend to be very ad hoc in nature and span a wide variety of domains. As such, many frameworks have been developed to help designers develop mashups for the different domains. Mostarda and Palmisano present a mashup framework that features a hybrid scripting language that supports unification as it is based on the type morphing paradigm. Much like inheritance in object oriented environments, type morphing is the ability of the language to cast any primitive type to another one where needed. The transformation of data across heterogeneous types is done by following a set of predefined rules. Through an interactive interface, users can iteratively overload (or mash) model elements until a sufficient solution is developed. Vancea et al. also use an object-oriented data model for the exchange of data between the disparate data models. Furthermore they argue that current web mashup frameworks lack sufficient data models to handle data interchange and propose “a database-driven approach to web mashups that supports integration at the database level and enables mashup developers to work with a uniform abstract model and have direct access to powerful features of database systems” . Abiteboul et al. present a mashup model that quantifies the different roles of mashups which are: query data sources, import other mashups, use external Web services, and specify complex interaction patterns between its components. Their model is derived from semantic web, and various object oriented concepts like inheritance. One concept that is commonly discussed in mashup frameworks is interactive (or iterative) processes. This is especially true in the ‘End User Programming’ literature and will be discussed in the following section, and can be explained by the fact that multiple mashup variations are frequently considered before an acceptable solution is discovered. Cetin et al. present a framework for migrating legacy systems toward service-oriented mashups. In their framework, they place emphasis on modeling the business requirement up front and then analyzing the existing legacy systems to see if such functionality currently exists and can be implemented into the mashup. "The iterative mapping process is as follows: (a) if a business requirement can be satisfied by one of the existing legacy components, then simply wrap it by considering the QoS attributes; (b) if there is a gap with the existing legacy component and requirement, and the gap can be filled during service wrapping, then accustom the legacy component into a new service; (c) if the gap cannot be fulfilled, then develop a new service for the requirement" .

### 3.2 End User Programming

Another challenge being addressed by mashup researchers is how to seamlessly package mashup technologies in such a way that non-technical users can easily and

effectively create mashup application in a myriad of different domains. As represented by the attention given to this topic in literature, developing mashup environments that are conducive to non-technical end user development is one of the more difficult challenges. Of the 60 publications that were reviewed in Appendix A, 25% (15) can be classified as 'End User Programming'. Two different approaches are commonly taken in addressing (enabling) end user mashup development. The first approach is passive by nature and focuses on designing plugins that work with the user's current browser, observe what the user is viewing, to suggest related sources for potential. There are several benefits to extending the user's current browser for the mashup process. First, the mashup process is very close to the user's web browsing experience, if the user encounters data that they would like to manipulate, the user would not have to launch a separate program to begin mashing . Secondly, as mentioned previously in the client based mashup agent discussion, it makes it easier to access local data on the user's machine that the browser normally has access to. Lastly, installing a browser plug-in is easier than installing a separate application, and thus may foster use .

A second approach to end user mashup development is proactive by nature and is necessary when the mashup process becomes more complicated (e.g. process modeling or advanced interface integration) . Tuchinda et al. present a tool that enables users to develop mashups in complicated integration domains by first providing examples of what the end mashup should look like; the tool then aims to mimic the format of the end result, allowing for mashups to be developed by end users who don't have programming experience. Tatemura et al. , take a similar 'by example'

approach in developing a tool that allows users to mashup disparate data sources by creating abstracted target schemas that are populated based on examples provided by the user. Mashroom is another end user mashing application that is based on the nested relational model, and allows users to iteratively construct mashups through continuous refinement . As mentioned previously, end user programming literature seems to generally fall into one of two categories, passive or proactive. Passive approaches piggyback onto the user's current web browsing experience and function as browser plugins, and proactive approaches are executed from a separate application and are designed for more complicated mashup domains. Figure 2 illustrates a third approach that is beginning to appear in literature that is a middle ground to the passive and proactive approaches . In this scenario, the user's browsing experience begins as normal with the mashup process being passive. The browser has a plugin installed that observes the user's browsing patterns. Next, the plugin activates a server based inference engine (e.g. Google Suggest) that suggests potential mashup sources in a side bar. The user can continue browsing or at any point begin creating a mashup.

### 3.3 Enterprise Mashups

Business researchers have identified an emerging opportunity for organizations to use technology to exploit information from desktops, the web, and other non-traditional enterprise sources, in order to react to situational business needs [65]. This tract of research has been labeled as 'Enterprise Mashups', and has been distinguished from consumer mashups because there are a plethora of legal and accountability related issues (e.g. security, availability, quality) that are specific to the organizational domain

[16]. Similar to end user programming, enterprise mashups have received a significant amount of attention in mashup literature, of the 60 publications reviewed for this study, 20% (12) were classified as enterprise mashups. Two areas of existing enterprise related research that support enterprise mashup domains are Service-oriented Architecture and Enterprise Information Integration . Service-oriented Architectures (SOA) are based on semantic web services and have widely been considered a key technology for achieving business-to-business integration within corporate intranets . However, as Web 2.0 concepts are being applied to enterprise domains (e.g. enterprises mashups) the need to offer a user-centered focus to improve business productivity and innovation has been revealed . This user centered approach is beyond the functionality of traditional B2B SOA's . Therefore, a big focus in enterprise mashup related research is on developing tools and frameworks to extend SOA's into mashup-able applications.

3.7 other example for Mashups integration :

[Trendsmap](#) is one example of a service that was a popular mashup, which combined Twitter's trending topics with data from another Twitter trend site and displayed it on a Google Maps-powered map. Today, however, trends on Trendsmap are calculated through the service's own algorithm.

Many mobile apps also function as mashups. For example, a restaurant app might combine menu information with location data to tell you about what's good to eat at nearby places.

A prime example of this is Google Maps, which is a very popular interface to use in mashups. Google allows developers to access their maps through API's. The developer can then combine these maps with another stream of data to create something new and unique.

[Housingmaps.com](#) is a mashup that applies real estate information, such as apartments for rent or homes for sale, from Craigslist to Google Maps. The end result is a system that lets you sort available apartments and homes by price and location. You can also see each listing marked on an interactive map, allowing you to quickly and efficiently browse your many options.

[centrepoinstores.com:](#)

is mashup that applies Store shopping such as clothes, shows ,machines, electronics ....etc.it collect the information from multiple stores like splash ,shoemart and lifestyle The end result is a system that lets you sort available products and by price and brand. You can also see each listing marked on an category, allowing you to quickly and efficiently browse your many options.

---

## 5 CONCLUSION

---

In this reports we discuss the idea of website for Mashup, we explain the problems of the project and determine the solution, and then we draw diagrams to illustrate the stages of work. Our feature work is implementation our website using suitable tools

---

## REFERENCES

---

- ATTWELL, G., BIMROSE, J. & BROWN, A. (2008) Maturing learning: Mash up Personal Learning Environments (MUPPLE08). *First International Workshop on Mashup Personal Learning Environments* CEUR.
- BENSLIMANE, D., DUSTDAR, S. & SHETH, A. (2008) Services Mashups: The New Generation of Web Applications. *IEEE Internet Computing*, 12.
- BOYD, D. (2008) *Why Youth Love Social Network Sites: The Role of Networked Publics in Teenage Social Life.*, Cambridge, MA: The MIT Press.
- CONNER, M. (2004) *Learn more now : 10 simple steps to learning better, smarter, and faster*, Wiley.
- DEMOS REPORT (2007) Young people are spending their time in a space which adults find difficult to supervise or understand ..
- DOWNES, S. (2006) Learning Networks and Connective Knowledge. *Instructional Technology Forum*.
- ERAUT, M. (2000) Non-formal learning, implicit learning and tacit knowledge. *British Journal of Educational Psychology*, Vol. 70 Pages 113-136.
- FELDER, R. M. (2006) An Introduction to Learning Styles: How Students Learn, How Teachers Teach, and What Usually Goes Wrong with the Process.
- FELDER, R. M. & BRENT, R. (2001) EFFECTIVE STRATEGIES FOR COOPERATIVE LEARNING. *J. Cooperation & Collaboration in College Teaching*, 10, 69–75.
- GREWE, L. (2009) *OpenSocial Network Programming*, WROX.
- HI5 Developer Platform [online]. <http://www.hi5networks.com/developer/>.
- JENKINS, H. (2006) Confronting the Challenges of Participatory Culture: Media Education for the 21st Century. *Building the new field of digital media and learning*. Chicago, Illinois, MacArthur Foundation.
- L-TREE [online]. [www.l-tree.org](http://www.l-tree.org).
- LIN L., VAN'T HOOFT M. & SWAN K. (2007) Learning with teenagers: Differences between teenage students and teachers using and learning to use digital technologies. [online]. <http://www.iste.org/>
- MOODLE 2.0 (2009) Progress tracking feature [online]. [http://docs.moodle.org/en/Development:Progress\\_tracking#Standards\\_support](http://docs.moodle.org/en/Development:Progress_tracking#Standards_support)
- MOODLE 2.0 (2009a) Student information API. [online]. [http://docs.moodle.org/en/Student\\_Information\\_API](http://docs.moodle.org/en/Student_Information_API).
- MOODLE 2.0 (2009b) Web Services module. [online]. [http://docs.moodle.org/en/Development:Web\\_services](http://docs.moodle.org/en/Development:Web_services).
- MOODLE [online]. <http://moodle.org/>.

- NIKOS MANOUSELIS & CONSTANTINA COSTOPOULOU (2007) Analysis and Classification of MultiCriteria Recommender Systems. *World Wide Web* 10, 415-441.
- NSBA REPORT (2007) CREATING & CONNECTING: Research and Guidelines on Online Social — and Educational — Networking.
- OFCOM (2008) Social Networking A quantitative and qualitative research report into attitude, behaviours and use.
- PETTENATI, M. C. & CIGOGNINI, M. E. (2007) Social Networking Theories and Tools to Support Connectivist Learning Activities. *International Journal of Web-based Learning and Teaching Technologies*, Vol. 2.
- PRENSKY, M. (2001) Digital Natives Digital Immigrants. *NCB University Press*, 9.
- SEGARAN, T. (2007) *Programming Collective Intelligence Building Smart Web 2.0 Applications*, O'Reilly.
- SIEMENS, G. (2005) Connectivism: A Learning Theory for the Digital Age. *eLearn Magazine*. ACMed.
- SILVA, J. F. & RESTIVO, F. J. (2008a) E-assessment through Social Networks in a Portuguese High School Context - A Conceptual Framework. *ICL*. Villach-Austria.
- SILVA, J. F. & RESTIVO, F. J. (2008b) Social Networks Usage in a Portuguese High School Learning Context. *Web 2.0 Meeting*. Braga, Portugal.
- STUTZMAN, F. A. K.-D., J., (2008) Modelling Cultural Acquisition in Online Social Networks. *Proceedings ASIST 2008*.
- TRINDER, K., GUILLER, J., MARGARYAN, A., LITTLEJOHN, A. & NICOL, D. (2008) Learning from digital natives: bridging formal and informal learning. [www.academy.gcal.ac.uk](http://www.academy.gcal.ac.uk).
- VARANDAS, L. (2009) Uma aplicação social utilizando estilos de aprendizagem (A social application using learning styles). *Informatics Engineering*. Porto, Porto.
- WILD, F., MÖDRITSCH, F. & SIGURDARSON, S. E. (2008) Designing for Change: Mash-Up Personal Learning Environments. *eLearning Papers*.
- ZAKRZEWSKA, D. (2008) Using Clustering Technique for Student's Grouping in Intelligent E-learning Systems. IN HOLZINGER, A. (Ed.) *4th Symposium of the Workgroup Human Computer Interaction and Usability Engineering of the Austrian Computer Society, USAB2008*. Austria, Springer.



Survey

## 6 MASHUP






---



1. ماهى نسبة الربح المتوقعة التى سوف تتدورها من فكرة الماش اب والتى تتمثل فى جلب منتجات من السمتاجر المنتشرة على الانترنت وعرضها على موقعك

			Response Percent	Response Total
1	50%		88.89%	8
2	30%		11.11%	1
3	60%		0.00%	0
4	10%		0.00%	0
5	0%		0.00%	0
6	لا اعرف		0.00%	0
<b>Analysis</b>		Mean: 1.11 Std. Deviation: 0.31 Satisfaction Rate: 2.22	answered	9
		Variance: 0.1 Std. Error: 0.1	skipped	0




2. ماهى اهم المتاجر التى تطمح بعرضها فى موقعك

			Response Percent	Response Total
1	امازون		88.89%	8
2	بوكيج		77.78%	7
3	سوق كوم		88.89%	8
4	اجودا		77.78%	7
5	ويجو		77.78%	7
<b>Analysis</b>		Mean: 12.11 Std. Deviation: 18.8 Satisfaction Rate: 200	answered	9
		Variance: 353.54 Std. Error: 6.27	skipped	0

3. ماهى المنتجات المستهدفه للمشروع

			Response Percent	Response Total
1	ملابس		77.78%	7

### 3. ماهى المنتجات المستهدفة للمشروع

							Response Percent	Response Total
2	اطعمه وماكولات						77.78%	7
3	اجهزه الكترونيه						100.00%	9
4	جوالات						77.78%	7
Analysis	Mean:	8.44	Std. Deviation:	10.97	Satisfaction Rate:	170.37	answered	9
	Variance :	120.41	Std. Error:	3.66			skipped	0