On Line Book Store



## **Introduction:**

In the world of software development there lots of improvement in the area of Architectural design and principles. The philosophies and implementation details are changing as the people guiding the development of the application.

In this fantastic and yet sometimes complex world of software development there are some tried and true architecture patterns and software development guidelines employed by most architects. Also our design must have an ability to turn towards innovation instead of lending itself to common practices.

Web services are one such area where architects must lean on their creative side and hope that their solutions are still successful. In this report we will explain an exciting voyage down the road of Web services application. From requirements to use cases, to database design, to user interfaces, we will cover each and every aspect of system design required to build an application with collaborative Web services.

The reason why we selected online Bookstore web Site is everybody walking down the street has some idea about bookstores. The objective of this project is to develop an e- book store where books can be bought from the comfort of home through the Internet.

An online book store is a virtual store on the Internet where customers can browse the catalog and select books of interest. The selected books may be collected in a shopping cart. At checkout time, the items in the shopping cart will be presented as an order. At that time, more information will be needed to complete the transaction.

Usually, the customer will be asked to fill or select a billing address, a shipping address, a shipping option, and payment information such as credit card number. An e- mail notification is sent to the customer as soon as the order is placed.

Electronic Commerce (e-commerce) applications support the interaction between different parties participating in a commerce transaction via the network, as well as the management of the data involved in the process.

The increasing importance of e-commerce is apparent in the study conducted by researchers at the GVU (Graphics, Visualization, and Usability) Center at the Georgia Institute of Technology. In their summary of the findings from the eighth survey, the researchers report that "e-commerce is taking off both in terms of the number of users Shopping as well as the total amount people are spending via Internet based transactions".

Over three quarters of the 10,000 respondents report having purchased items online. The most cited reason for using the web for personal shopping was convenience (65%), followed by availability of vendor information (60%), no pressure from sales person (55%) and saving time (53%).

Although the issue of security remains the primary reasons why more people do not purchase terms online, the GVA survey also indicates that faith in the security of e-commerce is increasing. As more people gain confidence in current encryption technologies, more and more users can be expected to frequently purchase items online.

A good e-commerce site should present the following factors to the customers for better usability.

• Knowing when an item was saved or not saved in the shopping cart.

• Returning to different parts of the site after adding an item to the cart.

• Easy scanning and selecting items in a list.

• Effective categorical organization of products.

• Obvious shopping links or buttons.

• Minimal and effective security notifications or messages.

• Consistent layout of product information.

Another important factor in the design of an e-commerce site is feedback. The interactive cycle between a user and a web site is not complete until the web site responds to a command entered by the user

Finally, feedback should not distract the user. Actions and reactions made by the web site should be meaningful. Feedback should not draw the user's attention away from

The important tasks of gathering information, selecting products, and placing orders.

**Chapter Two**

**Techniques Used To collect requirements**



# **Requirement Gathering Methods**

## **Introduction**

### The purpose of this paper is to examine the different methods in gathering requirements. Requirements are one of the most vital pieces to ensuring the success of a system or project. To ensure the optimal requirements are received, the methods in which those requirements are obtained are equally important. Through this paper, we will look at what requirements are, as well as the different methods in gathering them.

The second phase of the systems development life cycle is analysis phase. The analysis phase is the most important stage in SDLC. A study by project management consulting company, PM Solutions identified the top cause of a troubled project was poor requirements. The 2004 CHAOS report from the Standish Group indicated that project success rates were only 34%, the rest were either challenged or failed. They also noted that poor requirements were very common in project failures, as they listed it third on their list common factors of project failures. Interestingly enough, they noted lack of stakeholder involvement as first on their list. Lack of stakeholder involvement, however, can also be tied into poor requirements. When the stakeholders aren’t heavily involved, or have poor dialogue between the systems analyst, this directly leads to poor requirements.

Poor requirements have a significant impact on the end results of systems or projects. Requirements are the “blueprints” that everyone involved on the project uses to work from. When there are poor requirements, this can lead to poor designs and tests, which in turn will cause delays in development and testing. The requirements must be revised; hence, all of these delays contribute to a late project. Poor product quality can result from poor requirements in situations where key components are overlooked and left out altogether. Testing is also prioritized to focus on what’s important. If the requirements are flawed or unclear, testing is not properly executed and will eventually lead to poor quality. To avoid poor requirements, it is highly vital that the analysis phase of the SDLC is thoroughly completed, without being overlooked.

The analysis phase can be broken into to two processes: requirement gathering and analyzing the requirements. Requirements, in turn, are divided into functional requirements and non-functional requirements. Functional requirements are defined as processes, information, and interactions. These are the desired functionality that the client wants built and describe the interaction between the system and its environment. Functional requirements generally describe what the system has to do. Non-functional requirements are nonfunctional characteristics that address operational and technical requirements. They may describe such factors as encryption, security, hosting, environment, disaster recovery, business continuity. Gathering requirements properly and selecting the appropriate technique can assist in ensuring that you will not have poor requirements.

## **One-on-One Interviews:**

One-on-one interviews are the most common technique for gathering requirements, as well as one of the primary sources of requirements. To help get the most out of an interview, they should be well thought out and prepared before sitting with the interviewee. The analyst should identify stakeholders to be interviewed. These can be users who interact with the current or new system, management, project financers or anyone else that would be involved in the system. When preparing an interview is it important to ask open-ended questions, as well as closed-ended questions. Open-ended questions generally help in obtaining valuable information, based on various individuals and the way the different way they interact with, or view, the system. These types of questions require the interviewee to explain or describe their thoughts, and cannot be simply answered with a “yes” or “no”. Asking the interviewee what they like about the current system or how they use it would be examples of open-ended questions. These types of questions can provide the consultant to further probe for more detail with follow up questions, in order to get more details. An example open-ended question would be “What are some of the problems you face on a daily basis?” Close-ended questions can also be useful, when the interviewer is looking for a specific answer. They can provide specific answers for the interviewee to choose from, in formats including true or false or multiple choice. Although close-ended questions do not provide as much detail as open-ended, they can be useful to cover more topics in a less amount of time. An example of a close-ended question would be “How many telephone orders are received per day?” Once the questions have been established, it is a good practice to provide the questions to the interviewee prior to the interview, in the event that the interviewee needs to prepare. During the interview, the interviewer should obtain permission from the interviewee that recorders may be used, to ensure that if details are missed while taking notes can easily be retrieved. At the end of the interview, the results should be provided to the interviewee, for confirmation of their responses.

## **Group Interviews:**

Group interviews are similar to one-on-one interview, except there is more than one person being interviewed. Group interviews work well when the interviewees are at the same level or position. A group interview also has an advantage when there is a time constraint. More thoughts and discussion can be generated, as someone in the group may state or suggest an idea that may have been overlooked by others, which in turn can lead to a discussion or provide more information on a particular issue. The interviewer can gauge which issues are more generally agreed upon, and which are which issues differ. A major disadvantage can be scheduling the interview. When more than one person are involved, it may be difficult, or become time consuming, in establishing date and time that works well for all parties.

## **Questionnaires/Surveys:**

Questionnaires, or surveys, allow an analyst to collect information from many people in relatively short amount of time. This is especially helpful when stakeholders are spread out geographically, or there are dozen to hundreds of respondents whose input will be needed to help establish system requirements. When using questionnaires, the questions should be focused and organized by a feature or project objective. Questionnaires should be not be too long, to ensure that users will complete them. When constructing the questionnaire, general guideline to determine the questions would be to ask “how, where, when, who, what, and why.” For how: *“How will you use this feature?” “How might we meet this business need?” “How will we know this is complete?”* For where: *“Where does the process start?” “Where would the user access this feature?” “Where would the results be visible?”* For when: *“When will this feature be used?” “When will the feature fail?” “When will we be ready to start?”* For who: *“Who will use this feature?” “Who will deliver the inputs for the feature?” “Who will deliver the outputs of the feature?”* For what: *“What do I know about this feature?” “What does this feature need to do?” “What is the end result?” “What must happen next?”*

## **Use cases**

Use cases are basically stories that describe how discrete processes work. The stories include people (actors) and describe how the solution works from a user perspective. Use cases may be easier for the users to articulate, although the use cases may need to be distilled later into the more specific detailed requirements.

## **User Observation:**

The direct approaches of interviewing and questionnaires provide valuable user feedback based on the questions asked of them; however, there are times when direct observation may be better suited in requirement gathering. To get a better understanding of a user in their in current work environment, the analyst may observe the user themselves. User observation is helpful in assisting the analyst by getting a full grasp of how the user interacts with the system, firsthand. When the objective is to improve a task, the analyst can observe the user and how their surroundings affect their interaction with the system. Sometimes stakeholders may find it difficult in explaining what exactly what their tasks consists of and what their requirements may be, observing the user in cases like these will help provide the requirements. User observation may also be useful in validating data that had been previously collected. Be it in cases where users provide misleading information, or cannot fully recollect all of their tasks in how they use the system.

User observation should be planned to ensure that all elements are constant surrounding the observation. This will assist in uncertainty, and the consultant can focus on the user and assist in knowing what to look for. The analyst will not be distracted and record, or note, irrelevant issues. The more useful information gathered, the less time it will take to the analyst to dissect and evaluate afterwards. Timing of the observation can also prove relevant when planning. For optimal results, the consultant should schedule three different periods of observation: low, normal, and peak times. This may prove helpful in because the user may interact with the system differently during different times. The consultant will take into consideration the differences in times time settings and use it to obtain to construct better requirements to assist in all three times. When observing the user, there are two approaches the consultant can take, passive or active. In passive observation, the consultant does not interact with the user while they are working. They simply observe and take notes. While in active observation, the consultant will ask the user questions during the session. Observations should always be done without bias, as in other requirement gathering techniques, the analyst must simply record what is presented to them, and avoid making comments on whether they believe what is correct or not. A checklist can provide to be useful, with key points already noted and the consultant verifying events on their list. For example, they can check if a user uses certain features, the frequency of events, triggers that cause different uses. Taking detailed notes is helpful in recording unexpected events. There may be events unknown to the analyst ahead of time, they can be captured by taking notes of the event and why it occurred. Video recorders may be used, but must always be approved with the user and their company.

While there are many advantages to use observation, there are some disadvantages associated as well. As previously mentioned, observations should take place during peak, normal, and low business times. However, it may still be difficult to capture enough information in one of these sessions. There may be the need for multiple sessions to verify that facts collected were constant, rather than isolated incidents. Analysts themselves can sometimes be biased in what they expected to see, and what they actually observed. Again, the focus is to simply collect the facts, not form any biased opinion when seeing the working environment. The users sometimes may not themselves provide an accurate depiction of their every day task and respond differently. Some users may become nervous, and not perform as they normally would. Other users may try and perform more better or worker harder when they know they are being observed. Such observations can inhibit the analyst to decipher what the true requirements actually should be.

## **Analyzing Existing Documents:**

Analyzing existing documents can prove to be a useful technique in requirement gathering, on its own as well using it to supplement other techniques. Reviewing the current process and documentation can help the analyst understand the business, or system, and its current situation. Existing documentation will provide the analyst the titles and names of stakeholders who are involved with the system. This will help the analyst formulate questions for interviews or questionnaires to ask of stakeholders, in order to gain additional requirements. If an analyst is uncertain why certain procedures are in place, this can also help the analyst in asking these questions during interviews. When studying the requirements, the analysts may find problems that they may distinguish on their own. The analyst may find there was missing information in old documents. They may also find redundancy, in which steps are unnecessarily repeated. The consultant may look at old requirement documents and reuse of the requirements that may still be relevant, while discarding others that may be out of date. The reason why the current system is designed the way it is, which can suggest why certain features were left out. Principles and rules for the organization itself can be discovered by analyzing documents. Analyzing documents can be used as a supplement to information obtained from interviews, questionnaires, and observations. For example, if some of the interview answers are unclear, organizational documents may help in making sense of some of the interviewee’s answers. Reviewing existing documents may also assist in understanding why a user performs certain tasks while observing them. A drawback to analyzing documents is that documents may be outdated, the analyst must confirm whether the documents are current or not. Another drawback to reviewing documents is it can be very time consuming, depending on the organization and the system. A system that interacts with many different facets of the business will have large amounts of documentation to review.

## **Prototyping:**

Prototyping is another form a contemporary requirement gathering method. Prototyping is iterative process that heavily involves the users to complete. The user provides the requirements, in which the analyst can plug in directly and show the user the outcome. Prototyping is dependent on user interaction and cannot be utilized as its own method of gathering requirements. The analyst must interview or perform some other form of requirement gathering to perform before they begin prototyping. However, prototyping is very effective in specifying requirements, because of how heavily involved the user is. The user will still be sitting side by side with the analyst, providing them requirements as the analysts enters them into a working system. This will allow the user to instantly see the outcome of their requirements. At this point the user may change some of their requirements. They may see that what they provided was not what they had in mind. A form may appear cluttered with information; at this point the user can go back and adjust their information. This may also be the case in when the user forgets important information; they may not realize it until they actually see a working version of the system. The user and analyst will continue to go through different iterations, until all specifications are complete. The last prototype will be used as a model to build the actual system. Some of the disadvantages of prototyping is the user will pay too much attention to details on the screens, rather than what the prototype is meant to communicate. Executives can grow impatient as they see a complete prototype, but will not understand why the finished system takes so long to complete.

## **Conclusion:**

There are many different methods in requirement; all will have advantages and disadvantages. Cost and time proved to be the two most important factors when determining which method was to be used. Many methods were used in supplement of each other. User observation was followed by interviewing, questionnaires, or analyzing documents. As was prototyping, they cannot be used as a method on their own. Prototyping can be useful in that it heavily involves the user, but at the same the developing quick prototypes can cause the user to be misled in expectations of when the real system may be completed. JAD was very efficient in it involved everyone at one time; however, sometimes the cost of JAD may be too much. The method in gathering the requirements may vary depending on the situation and constraints, but using the various methods to supplement each other will help in achieving complete requirement

**Chapter Three**

**System Requirement**

****

## **Introduction**

As we see in previous chapter, managing and maintaining a book shop could also be controlled by efficient software. This project focuses attention on designing efficient and reliable software which controls the transactions of a bookshop. In real world, it tends to associate with automated systems as they provide many benefits than doing the same thing in manually. As above mentioned, here we have introduced a system which can be used to maintain a bookshop. When we are concerning the manual process of a bookshop, the major problem is the waste of time. A customer has to waste his/her valuable time when he needs to buy a book as all the events such as searching, purchasing are done by members of the staff .In briefly, the manual process is very slow. But automation will reduce the time taken in the whole process. In a bookshop we should deal with a large store. Then person (storekeeper) has to maintain it with documents which are recorded by him. Therefore, there may be defective reports. The purpose of this project is to provide easy shopping facility online and easy selling facility to the merchants of all categories.

Scope

The name of the project is Book Store The software provides both the customer and the merchant his Id number which isunique.The software provides the following facilities to the customers:

Facilitates easy shopping online anywhere with free shipping (conditions apply)

Provides information about the products in categories

Can avail the facility of purchasing second hand products

Can reserve or order if the particular product is not available

Customers are provided with up to date information on the products available

Provides email facility for future correspondence

Can add nearly ten products to their shopping cart at a time.

Cannot reserve the same Product twice.

Responsibility of damages

Facilitates easy bidding facility

Provides complete information about the customers

Provides complete information about their products

Can avail the facility of email correspondence

Can avail the brand catalog facility

Transport facility

Legal arguments should be settled through the courts individually

Settlement of miscalculations of the products as the database is filled with the consent of the merchant the objective of this software is to provide easy assistance to both the customer as well as the merchant with proper database and information.

### User interface Requirement: 1. Clear: Clarity is the most important element of user interface design. Indeed, the whole purpose of user interface design is to enable people to interact with your system by communicating meaning and function. If people can’t figure out how your application works or where to go on your website they’ll get confused and frustrated.

### 2. concise

Clarity in a user interface is great, however, you should be careful not to fall into the trap of over-clarifying. It is easy to add definitions and e Clarity in a user interface is great, however, you should be careful not to fall into the trap of over-clarifying. It is easy to add definitions and explanations, but every time you do that you add mass. Your interface grows. Add too many explanations and your users will have to spend too much time reading through them.

### 3. Familiar

### Many designers strive to make their interfaces ‘intuitive’. But what does intuitive really mean? It means something that can be naturally and instinctively understood and comprehended. But how can you make something intuitive? You do it by making it ‘familiar’.

### 4. Responsive: Responsive means a couple of things. First of all, responsive means fast. The interface, if not the software behind it, should work fast. Waiting for things to load and using leggy and slow interfaces is frustrating. Seeing things load quickly, or at the very least, an interface that loads quickly (even if the content is yet to catch up) improves the user experience.

### Responsive also means the interface provides some form of feedback. The interface should talk back to the user to inform them about what’s happening. Have you pressed that button successfully? How would you know? The button should display a ‘pressed’ state to give that feedback. Perhaps the button text could change to “Loading…” and it’s state disabled. Is the software stuck or is the content loading? Play a spinning wheel or show a progress bar to keep the user in the loop.

### 6. Attractive: This one may be a little controversial but I believe a good interface should be attractive. Attractive in a sense that it makes the use of that interface enjoyable. Yes, you can make your UI simple, easy to use, efficient and responsive, and it will do its job well – but if you can go that extra step further and make it attractive, then you will make the experience of using that interface truly satisfying. When your software is pleasant to use, your customers or staff will not simply be using it – they’ll look forward to using it.

### There are of course many different types of software and websites, all produced for different markets and audiences. What looks ‘good’ for any one particular audience will vary? This means that you should fashion the look and feel of your interface for your audience. Also, aesthetics should be used in moderation and to reinforce function. Adding a level of polish to the interface is different to loading it with superfluous eye-candy.

### 7. efficient

A user interface is the vehicle that takes you places. Those places are the different functions of the software application or website. A good interface should allow you to perform those functions faster and with less effort. Now, ‘efficient’ sounds like a fairly vague attribute – if you combine all of the other things on this list, surely the interface will end up being efficient? Almost, but not quite.

What you really really need to do to make an interface efficient is to figure out what exactly the user is trying to achieve, and then let them do exactly that without any fuss. You have to identify how your application should ‘work’ – what functions does it need to have, what are the goals you’re trying to achieve? Implement an interface that lets people easily accomplish what they want instead of simply implementing access to a list of features.

### 8. Forgiving

Nobody is perfect, and people are bound to make mistakes when using your software or website. How well you can handle those mistakes will be an important indicator of your software’s quality. Don’t punish the user – build a forgiving interface to remedy issues that come up.

A forgiving interface is one that can save your users from costly mistakes. For example, if someone deletes an important piece of information, can they easily retrieve it or undo this action.

### B) Database Requirement: What is sql server?

**Microsoft SQL Server** is a [relational database management system](https://en.wikipedia.org/wiki/Relational_database_management_system) developed by [Microsoft](https://en.wikipedia.org/wiki/Microsoft). As a [database server](https://en.wikipedia.org/wiki/Database_server), it is a [software product](https://en.wikipedia.org/wiki/Software_product) with the primary function of storing and retrieving data as requested by other [software applications](https://en.wikipedia.org/wiki/Software_application)—which may run either on the same computer or on another computer across a network (including the Internet).

Microsoft markets at least a dozen different editions of Microsoft SQL Server, aimed at different audiences and for workloads ranging from small single-machine applications to large Internet-facing

### Why we choose MSSQL server?

* Always availability
* Infinite data storage.
* Electability to add more tables and relations
* Easy to Handel.
* Commonly used with ASP.Net framework

### C) Functional Requirements:

### In this Project we try to analysis some Functions Related to every actor in our system as Follows.

New customers need to register first to get an account ID. The customer needs to provide the following information to the store: customer name, email address, and password. After registration, the customer will be assigned a unique account ID and he/she can login using the account ID (Normally E-mail is Unique} and password. One customer can only register one account and each account must belong to exact one customer.

The bookstore keeps a large amount of books. Each book is identified by its ISBN. For each book, the bookstore also needs to record its authors’ names, title, edition, publication Date, category, publisher, quantity-in-stock, and selling price.

One customer can place any number of Books. For each order, the bookstore needs to record who places this order, when, the order status, total price, shipping address, and ordered books.

Customers can also manage their shopping carts. One customer can have any only one shopping cart. However, each shopping cart has exactly one customer.

### Functions for customers:

* New user account registration
* User login
* User can update his/her address, password etc.
* Create order One or more.
* Add/delete books from shopping cart.
* Change shopping carts as orders
* Place the order
* Order trace

### Functions for analysts

* Which books are better sold in the second quarter than the first quarter?
* Which categories of books are the most profitable ones?
* What is the average time between the order placed and shipped?
* Is there any significant difference between books published by different publishers in terms of profitability?

### Function of system Admin

* Have full access to database
* Can add delete or add users
* Can extract any data for any registered user
* Can update any record
* Can delete any book or user
* Can add category
* Can add any data for book or user

### D) Non Functional Requirements:

### Usability Requirement

The system shall allow the users to access the system from the Internet using HTML or its derivative technologies like XML/CSS. The system uses a web browser as an interface. Since all users are familiar with the general usage of browsers, no special training is required. The system is user friendly and online help makes using the system easy and also

### Availability Requirement

The system is available 100% for the user and is used 24 hrs a day and 365 days year. The system shall be operational 24 hours a day and 7 days a week.

### Efficiency Requirement

Mean Time to Repair (MTTR) - Even if the system fails, the system will be recovered back up within an hour or less.

Accuracy

The system should accurately provide real time information taking into consideration various concurrency issues.

The system shall provide 100% access reliability.

### Performance Requirement

The information is refreshed at regular intervals depending upon whether some updates have occurred or not. The system shall respond to the member in not less than two seconds from the time of the request submittal. The system shall be allowed to take more time when doing large processing jobs.

Responses to view information shall take no longer than 5 seconds to appear on the screen.

### Reliability Requirement

The system has to be 100% reliable due to the importance of data and the damages that can becaused by incorrect or incomplete data. The system will run 7 days a week, 24 hours a day.

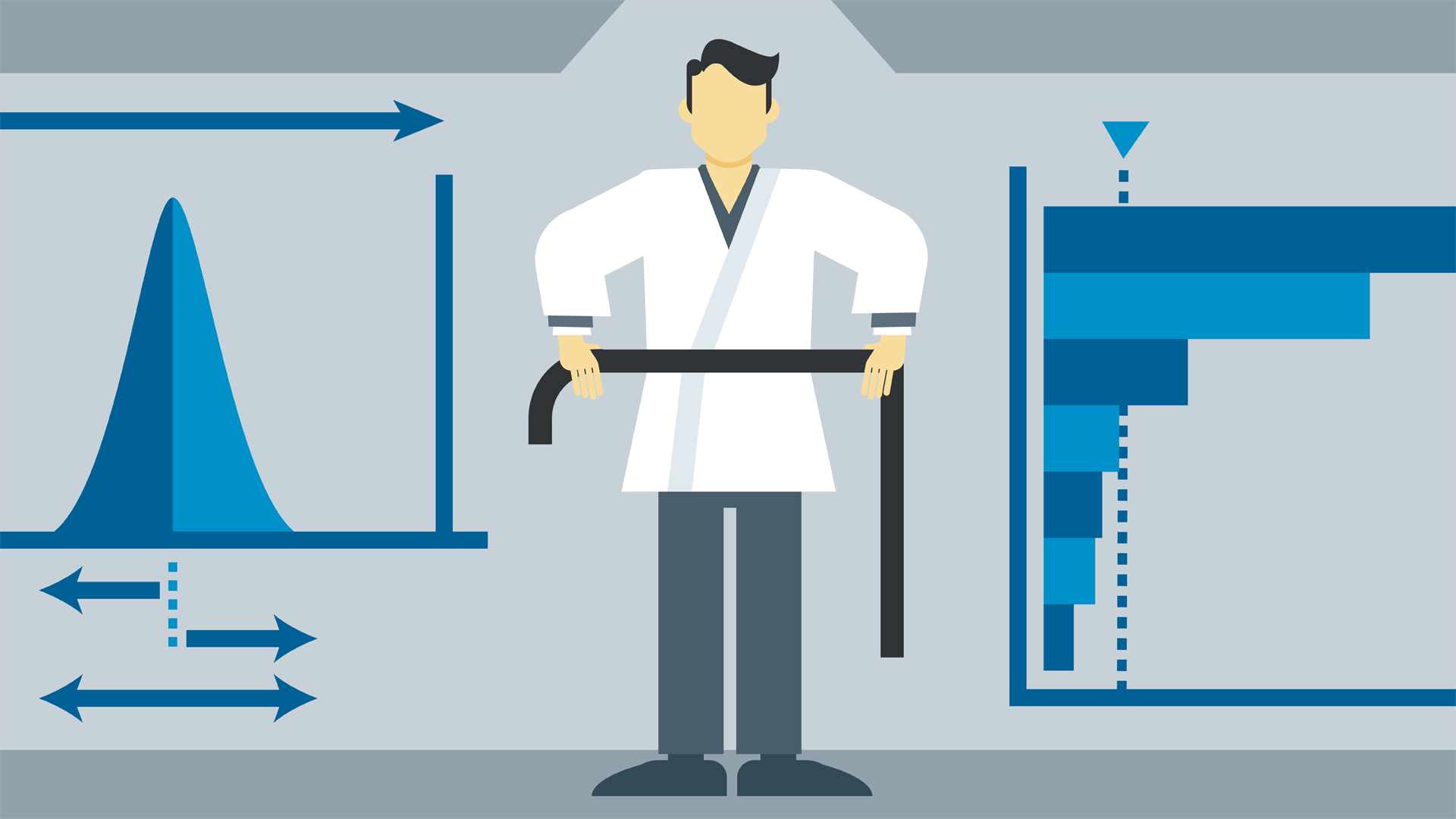
### Maintainability and Portability Requirements

Changes (new patron’s addition, password changes, database changes) must be verified once per day at least. The system should provide automatically notification to patrons by e-mail about item’s overdue, reservation results, availability of reserved item and etc.

**Conclusion:**our online Bookstore is the Best Place for who want to search, Read some description about books and also read others comment.  
It can provide also some information about book content and category, also customer can login and register for account to identify his identity and some information about him**.**

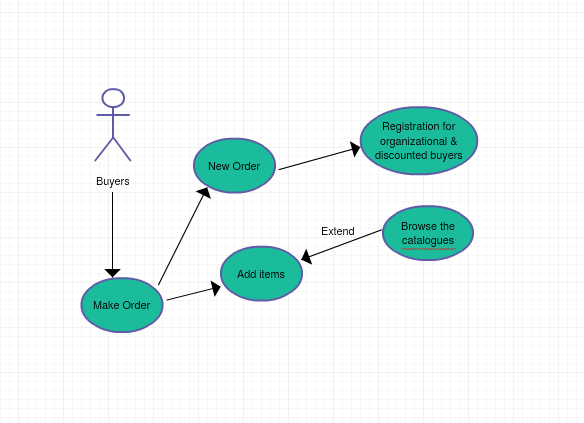
**Chapter Three**

**System Analysis**

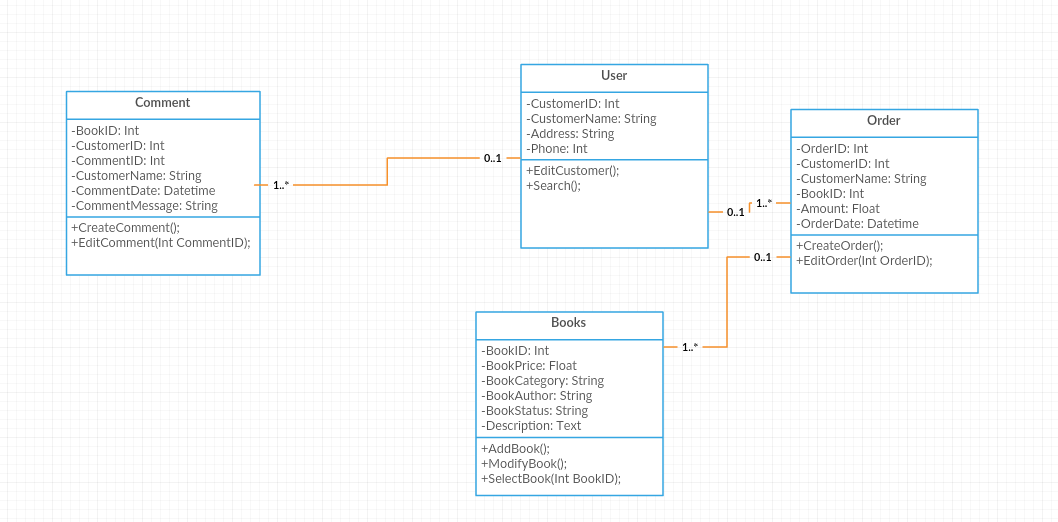


## **Abstract**

Service Science is the basis of information system and web services that ascribe to the provider/client model. The goal is to development a methodology that will add structure to a highly unstructured problem to assist in the development and success of web services.

**Simple Use Case Diagram for Online Bookstore**   


**Class Diagram for Online Bookstore**

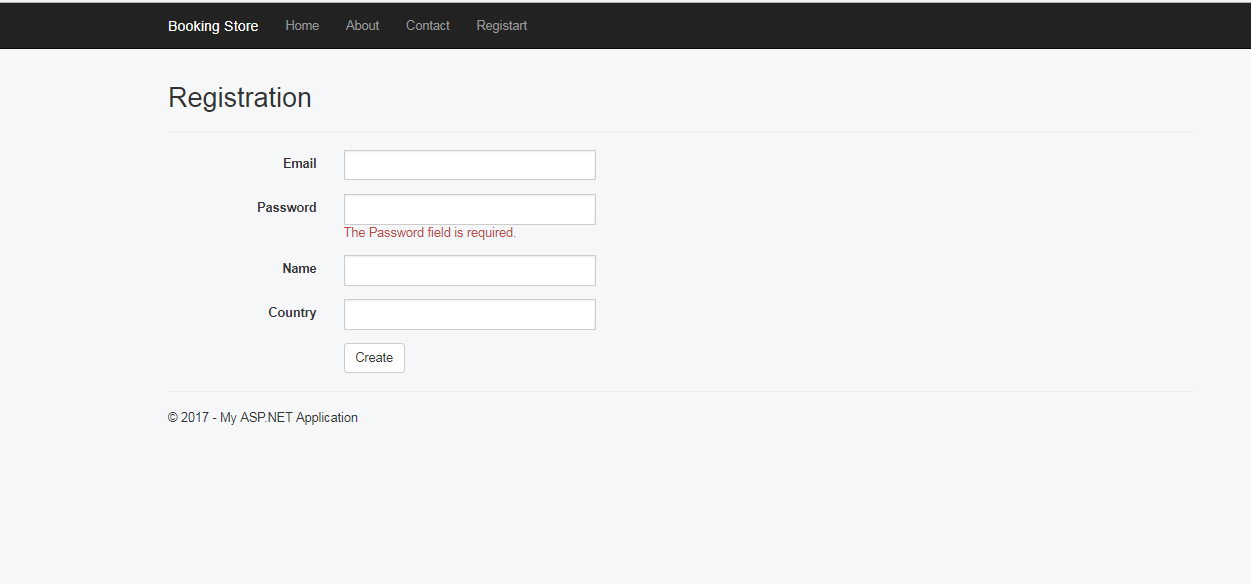


**Chapter Four**

**System Implementation**

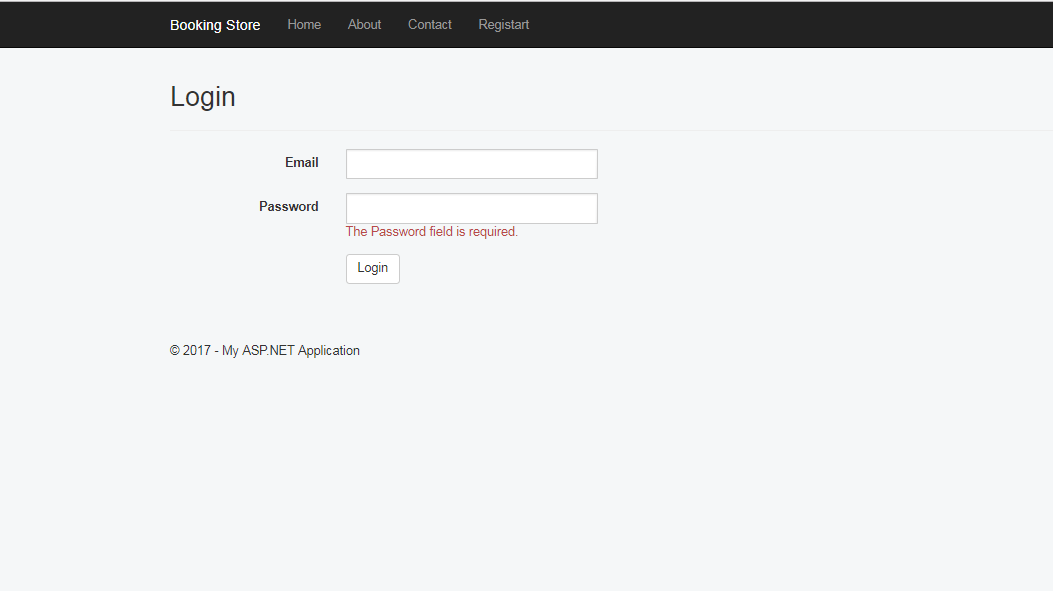
****

**User interface Implementation:   
Registration**User cannot Access any link without Register and Login   
here is the registration Page.  
Email and Password are Mandatory and Name will be displayed in Profile Page .



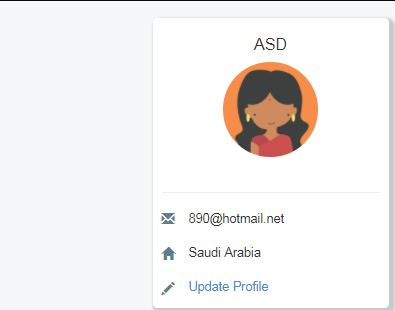
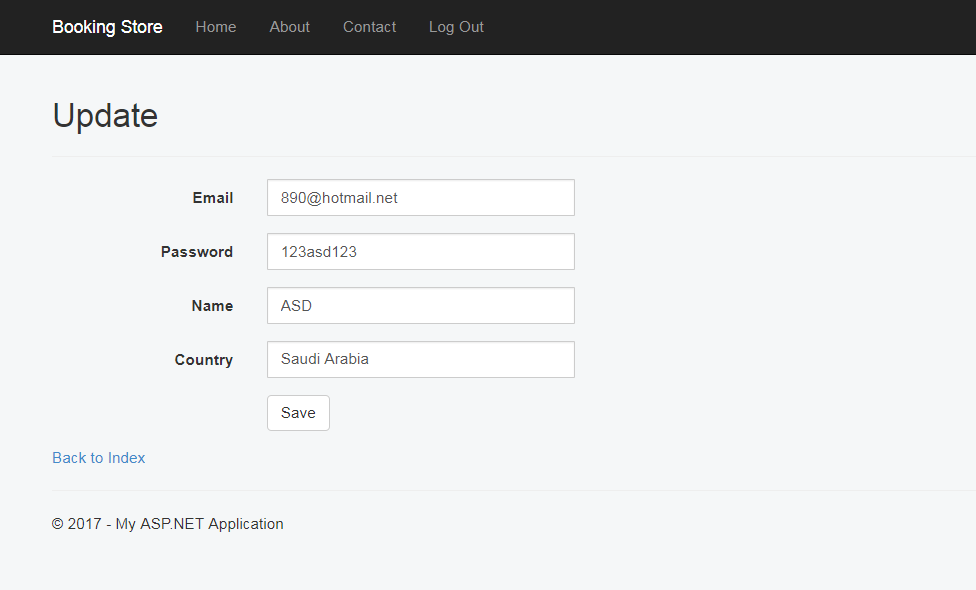
**Login**

if user already registered can Enter his Username and Password and Login Directly   
username and Password are Mandatory and Unique .

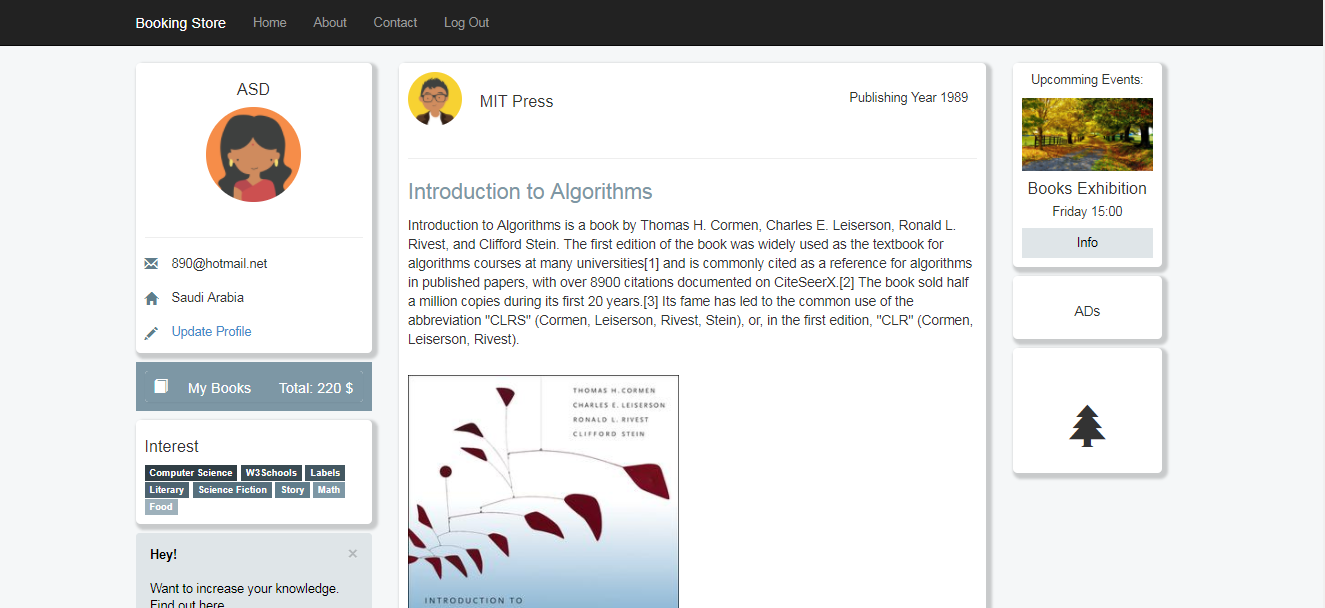
****

**User Profile**

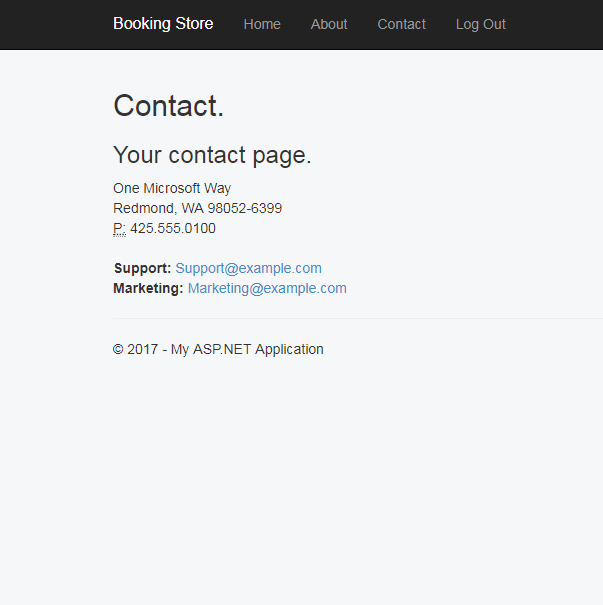
After registration user can find all his data in profile section

**Update Profile**  
user can Update his Profile by Press Update Profile Button this Button will redirect him to Update Profile Page.  
By default all fields are fill with his previous data and User can update any Field or Chang its Content.

**Home Page**

User can find all needed information about Book in every post,   
design is so friendly and simple for everyone.   
User can add any book to cart, price will be added and user can go to cart section to see all selected Books.   
User can add book or remove Book easy.  
User also can see some news or events in event section.

**Contact and Support Page**

User can find contacts or support material in this page.

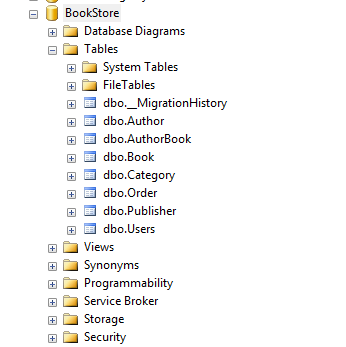
**About Page**

User can find some information about the web site author in this Page.

## C:\Users\m.lewes\Desktop\Capture.PNG**Database Implementation:**

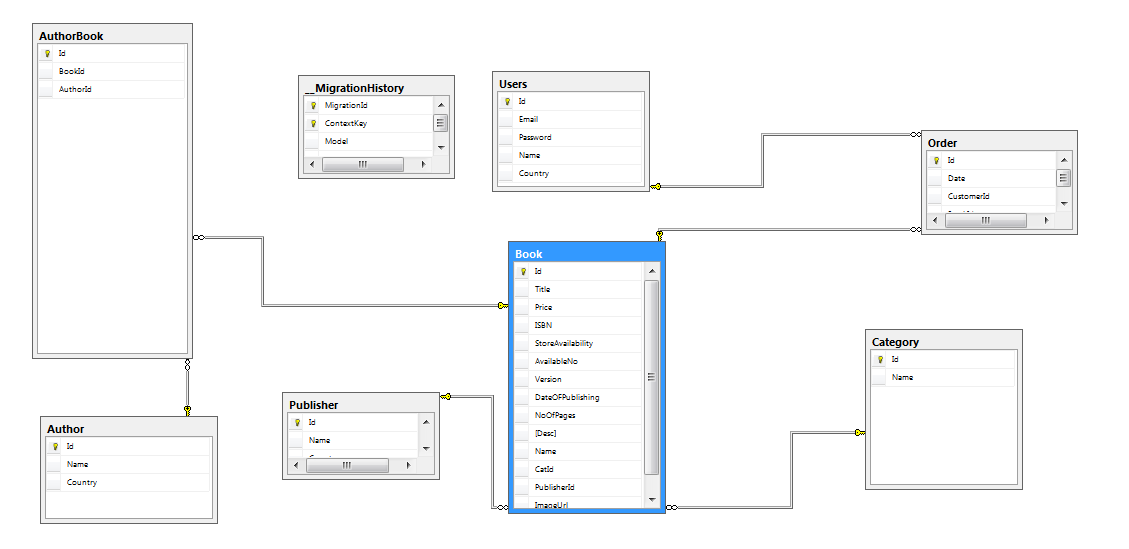
### Drawing ERD for Database Entities: C:\Users\m.lewes\Desktop\1122.jpg

(Live photo of Our Database ERD Entities)

** Mapping design to Real tables:**- Creating Tables and Relations ships between them after analyzing using ERD:

\

(Real photo to Database after Creating Tables)

After mapping all Relations between Tables the final Diagram Will be Like This:

**We have the Following Entities:**

* Author
* Book
* User
* Order
* Publisher
* Category
* And Some Lookups Tables

### Add Some Fack Data To Database:

