Synopsis Of

**“ONLINEBOOKSTORE.COM”**

Submitted By

|  |  |
| --- | --- |
| Shakti Singh | * 1621631046 |
| Shubham Kumar | * 1621631050 |
| Shubham Goel | * 1621610099 |
| Md. Imteyaz | * 1621610059 |
| Ammar Arfeen | * 1621610020 |

Under the guidance of

**Ms. Deepa Basantani**

In partial fulfillment of

**Bachelor of Computer Applications**

**Center of Computer Education**

IIMT COLLEGE OF ENGINEERING

GREATER NOIDA

(AKTU)

JULY - 2019

**INDEX**

1 Introduction

2 Motivation

3 Objective(s) and Scope

4 Target Users

5 Technical Platform

1. **INTRODUCTION**

Many web portals claim to provide a good BOOKS and STUDY MATERIALS for the students and teachers who are looking for a one stop solution for their problems. The different web portals functioning now a days promise their user’s (the registered ones) to provide them with books for the subjects for which they are looking for, and lead them to their desired one.

Also, their exist no way that users as well may give views about how they think about the books(well commenting about it is not a valid thing to do) however, these reviews hold a high opinion for other new users who want to join the portal.

The Project “Online Book Store” is a user interactive and delivery based website where the order can be booked and payments can be done shortly along with getting all the required details regarding each and every mobile. Business done by e-commerce is far more efficient, faster and reliable than traditional commerce.

This method of adoption of doing business via e-commerce is a user-friendly system which will help the user to access the product easily

1. **MOTIVATION**

- Customers can get their book delivered instead of actually going and buying

the book. They can make payment online itself.

- Managing of inventory in the shop for shopkeeper becomes easier as

customers are not visiting and ordering online.

- This system saves both time and travelling cost of customers.

- User can get to know different kinds of books that they were unaware of by

just searching in the system using keywords.

1. **OBJECTIVE(S) AND SCOPE**

**Objectives of the system are:-**

1. The new user must register with the portal before availing any services from it.
2. For each registration, maintain the record in the database for the new user.
3. For each new registration a profile is to be created if it is for the user then it will contain information about the user such as his/her email, password, name and mobile number.
4. When searching for the books by the user, their need for the book of a particular subject, and mode of study is the key area of concern.
5. When the book is found and selected by the user, payment for the cost of book is to be done if user wants to purchase it.
6. The payment could be done in two ways by offline payment (cash on delivery) and through online payment system(Card payment).
7. The user may comment on the quality and content of the book whether the book or service is appreciable or needs improvement. They can also upload their course materials on the portal which may be used later.
8. Admins may review these comments to know about the progress.

**Modules in the System:-**

The Onlinebookstore.com can be divided into two modules, namely:

**Module 1: Site Administrator**

This module is responsible for maintaining the overall system based on defined business rules and it perform the following functions:

1. It can delete user.
2. It can modify the user details.
3. It can update website details.
4. It can upload online materials.

6. Respond to e-mail request.

7. It can receive payments.

8. It can moderate the comments.

**Module 2: User**

The web portal as stated earlier assists them in finding them a suitable books and suggesting them their types of book, of which category it is as per they wish. They can search their desired books by typing the keywords in search bar or by seeing all the books present in the buying section according to their categories.

The demands and area of concern are asked from the users at the time of searching for books and based on this information the criterions are matched with the profile for the books which holds their interest, area of specialization and other related information. Once the suitable book is found a buying cost is charged from the user with online/offline payment options as directed by the Administrator.

1. **TECHNICAL PLATFORM:**

|  |  |
| --- | --- |
| **Operating System:** | Windows 10 |
| **Language Requirements**: | Java, J2EE |
| **Web Browser:** | Google Chrome, Mozilla Firefox |
| **Database Technology:** | MySQL |
| **Tools & Development:** | Eclipse IDE 9, HTML, JSP< bootstrap,Apache tomcat9 |

|  |
| --- |
| Information Gathering |
|  |

Project Name: ONLINEBOOKSTORE.COM

Submitter Name: Shakti singh,

Shubham kumar,

Shubham goel,

Md. Imteyaz,

Ammar arfeen

Date Submitted: 30/07/2019

|  |
| --- |
| **About Onlinebookstore.com System** |

In the contemporary world where the competition is everywhere from politics to science, research to education, it’s a really important task for the parents and students as well to locate a good and reliable study materials and books for themselves, reliable course materials to be studied and practiced. Onlinebookstore.com assists in finding best books for the students by matching the need of the students and also provides a good study source to registered users, where anyone from anywhere could learn, study and teach.

|  |
| --- |
| **Explain different functionalities of Onlinebookstore.com System** |

The system Onlinebookstore.com has following major functionalities:-

1. Helping students and teachers to find best books for them.
2. Providing latest online materials and overall solutions to their subjects.
3. Registering new users to the portal.
4. Providing access to the study materials from anywhere.
5. Users can give comments on the comment section for any queries or any problem facing which may be reviewed by the admin.

|  |
| --- |
| **Analysis of** **Existing Business Process** |

The existing business process provides facilities such as searching for books,novels,study materials for users, registering of users in their portals, receives payments for purchasing of books in different ways and provides online support to the users if admins are available.

|  |
| --- |
| **Challenges in existing Business Process** |

The existing business system although provides online support but fails to provide every online material to the students which they are looking for. Moreover, the materials often provided doesn’t contain total solutions to the problems of the students.

Earlier you could sell anything online and make money, the competition was minimal,technology was not as affordable as it is now, and the product were limited.

But the scenerio is totally different today, everyone can afford the technology nowdays and competition is very tough. You can say that there are major competitors for every e-commerce portal.

Here we are going to list out the common challenges faced -

-Finding the right product to sell

-Attracting the perfect customer

-Get more customer who are ready to purchase

-Achieving profitable long term growth

**SOLUTION ARCHITECTURE DOCUMENT**

**Onlinebookstore.com**

**Document History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Author** | **Reason for Change** |
| 1.0 | 30-july-2019 | Shakti singh,Shubham kumar,Shubham Goel,Md imteyaz,Ammar arfeen |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Table of Contents**

[1 General Information 4](#__RefHeading___Toc219544091)

[1.1 Introduction 4](#__RefHeading___Toc219544092)

[2 J2EE – Solution Architecture 4](#__RefHeading___Toc219544093)

[2.1 High Level Architecture 5](#__RefHeading___Toc219544094)

[2.2 Description 6](#__RefHeading___Toc219544095)

# General Information

The J2EE platform is a multi-tiered system. A tier is a logical or functional partitioning of a system. When the developers are not disciplined, the display logic, business logic and database logic are muddled up and/or duplicated in a

2- tier client server system.

The advantages of the multi-tier architecture are:

* Forced separation of user interface logic and business logic.
* Business logic sits on small number of centralized machines (may be just one).
* Easy to maintain, to manage, to scale, loosely coupled etc.

Each tier is assigned a unique responsibility in a 3-tier system. Each tier is logically separated and loosely coupled from each other, and may be distributed.

The advantages of a 3-tiered or n-tiered application: 3-tier or multi-tier architectures force separation among presentation logic, business logic and database logic. Let us look at some of the key benefits:

* **Manageability**: Each tier can be monitored, tuned and upgraded independently and different people can have clearly defined responsibilities.
* **Scalability**: More hardware can be added and allows clustering (i.e. horizontal scaling).
* **Maintainability**: Changes and upgrades can be performed without affecting other components.
* **Availability**: Clustering and load balancing can provide availability.
* **Extensibility**: Additional features can be easily added.

## Introduction

The **Goal** of the proposed system is to create a web–based application with the following capabilities**:**

* **Performance and scalability:** System should be efficient and scalable.
* **Code and Design Reuse:** Code reuse decreases the cost of development and increases the stability of the code. Also, following the best design practices and architectural and design patterns will enable us to reduce the risk of inconsistent design.
* **Logical Functional Decomposition:** Every class in the design will have a clearly defined responsibility to play in the application. This will result in an application, which is easier to understand, maintain and extend.
* **Minimize Network Traffic**. Avoid transmitting unnecessary and redundant data. To achieve this goal, we would be using value objects to pass the data to and from the web layer to the business layer and the DB layer.
* **Maintainability:** This architecture results in a system that would be easy to maintain. The configuration would be controlled primarily by configuration file, which would control the application flow and logic.
* **Reliable:** The developed system must be reliable enough to avoid unexpected behavior of the system and each of the methods, procedures and functions doing Insert, Update, Delete, Create Table or Select must include error management.

# J2EE – Solution Architecture

J2EE Solution Architecture in enterprise architecture is a kind of architecture domain, that aims to address specific problems and requirements, usually through the design of specific information systems or applications. The solution architecture is required to implement solutions to meet business requirements and ensures alignment with the Enterprise Architecture.

## High Level Architecture

**Online Bookstore.COM Web Portal Architecture**

HttpServletRequest

HttpServletResponse

HTML, DHTML,

JavaScript,

CSS

Client Tier

Oracle 10g Database

Service Tier

JSP,

Servlets

Web Container

Web Server

Presentation Tier

## Description

The target application has the following layers:

* HTML pages are built using HTML editor and Dreamweaver acting as the presentation layer and will be developed by HTML tags, JavaScript and style sheets.
* **HTML**-Hyper Text Markup Language (HTML) is the main [markup language](http://en.wikipedia.org/wiki/Markup_language) for [web pages](http://en.wikipedia.org/wiki/Web_page). HTML elements are the basic building-blocks of web pages.
* **JavaScript**-A scripting language developed by Netscape and used to create interactive Web sites.
* **Style Sheets**-It is used for describing the [presentation semantics](http://en.wikipedia.org/wiki/Presentation_semantics) (the look and formatting) of a document written in a [markup language](http://en.wikipedia.org/wiki/Markup_language).
* Business Logic Layer will be developed by using Servlets and Java Server Page (JSP)s.

1. **JSP**-Java Server Page (JSP) is a technology for controlling the content or appearance of Web pages through the use of [servlet](http://searchsoa.techtarget.com/definition/servlet)s, small programs that are specified in the Web page and run on the Web server to modify the Web page before it is sent to the user who requested it.
2. **Servlet-** A Servlet is a small Java program that runs within a Web server. Servlets receive and respond to requests from Web clients, usually across HTTP( the HyperText Transfer Protocol).

* JavaBeans are reusable software components for java.
* **JavaBeans**-They are classes written in the [Java programming language](http://en.wikipedia.org/wiki/Java_(programming_language)) conforming to a particular convention. They are used to encapsulate many objects into a single object (the bean), so that they can be passed around as a single bean object instead of as multiple individual objects. A JavaBean is a Java Object that is [serializable](http://en.wikipedia.org/wiki/Serialization), and allows access to properties using [getter and setter methods](http://en.wikipedia.org/wiki/Mutator_method).
* Reusable components are used in Logging, Registering, Searching, Making Payments, Commenting, Modifying Details, Downloading/Uploading materials, exception Handling and Data Access and will be written in java servlets and beans.
* Oracle10g Database are used as a Database Layer
* **Oracle 10g**- It is [Oracle](http://searchoracle.techtarget.com/definition/Oracle)'s [grid computing](http://searchdatacenter.techtarget.com/definition/grid-computing) product group including (among other things) a database management system ([DBMS](http://searchsqlserver.techtarget.com/definition/database-management-system)) and an [application server](http://searchsqlserver.techtarget.com/definition/application-server).

**Client Tier**

* Client tier represents Web browser.
* The client tier makes requests to the Web server.
* The Web server, who will be serving the request by either returning static content if it is present in the Web server or forwards the request to either Servlet or JSP in the application server for either static or dynamic content.

**Presentation Tier**

* Presentation tier encapsulates the presentation logic required to serve clients.
* JSP and Servlets which are the web container elements forms the presentation layer.
* A Servlet or JSP in the presentation tier intercepts client requests, manages logons, sessions, accesses the business services, and finally constructs a response, which gets delivered to client.

**Service Tier**

* This tier is the external resource such as a database.
* It is responsible for storing the data.
* This tier is also known as Data Tier or EIS (Enterprise Information System) Tier.
* It uses Java Beans classes to access Oracle 10g database.
* The data access will be done through Java Database Connectivity classes which enable the Oracle database to be connected to the code and operations may be performed on it.

|  |  |
| --- | --- |
| **Item(s) to Review** | **ADMINISTRATOR LOGIN** |
| **Description of Item(s), Background and Context** | Since customers can't feel and examine items before making a purchase decision, they rely on colorful descriptions to inform them. This lends itself well to books, which are often sold based on the "description" on the inside flap. The threat of duplicate content penalties makes it imperative to write descriptions that summarize the unique elements of a book without blatantly copying the manufacturer's synopsis. |
| **Objectives of Review** | The sole objective of the review is to correct the incorrect functionality and if any of functions are left to be completed must be completed. However code reviews have the following objectives:   * Quality, defect-free software * Systems that are appropriate and complete meeting requirements * Software that complies with enterprise coding standards |
| **Owner(s) of Item(s)** | Administrator (Shakti Singh, Shubham Goel, Shubham Kumar, Ammar Arfeen, Md Imtiyaz) |
| **Who is Impacted? Who Depends on these Items?** | Tutor, Student and Parent modules are directly impacted by the Administrator. The significant increase in popularity of online bookstores over the last decade has had a monumental impact on offline bookstores, covering all facets of their business. Online bookstores offer a number of advantages over physical stores. |
| **Reviewed by** | Shakti Singh, Shubham Goel, Shubham Kumar, Ammar Arfeen, Md Imtiyaz. |
| **Next Steps** | The next step to be taken after review is to enter into maintenance phase. |

# Logical Design and Layering Approach

Many web portals claim to provide a good selection of books for the students and parents who are looking for a one stop solution for their problems. The different web portals functioning now a days’ promise their user’s (the registered ones) to provide them with suitable books and materials for the subjects for which they are looking for, and lead them to these portals more often. These Web portals has its own unique style of suggesting books and imparting knowledge to the students, and providing them with their own course materials, there is no way to generate reviews about how these online portals are performing in the market and what the students and other people thinks about such service.

Also, their exist no way that parents or students as well may give views about how they think about the portal. However, these reviews hold a high opinion for other new users who want to join the portal.

**Online Bookstore** is a user friendly web-portal which provides solution to all these above stated problems from providing one-stop solution to provision of online Book purchasing through the portal depending upon the availability of the books, to the provision of online course material which is authentic and up to the standards.

# Maintainability, Adaptability

Assess factors that will impact the ease with which the software may be maintained or altered over time.

|  |  |  |
| --- | --- | --- |
| **Area of Review** | **Observations** | **Suggestions** |
| Duplicated Code | The code was found duplicated in Admin class where it is reused for implementing various functions such as deleteUser(), login(), uploadMaterials(), receivePayments() and updateDetails().  The code is reused as it allows us to save effort and to redesign the solution again and again. | Whenever you see code that looks like it was copied and pasted, even with a few minor changes, explore opportunities for reuse. You might consider implementing classes that encapsulate the logic, or creating helper classes. |
| Use of Short Methods | Short methods are used in Admin class for implementing functions such as update(), updateLatestbooks(), updateJobboard(), updateNews(), etc. This allows us to easily maintain these codes. | Try to keep all methods short. Long methods become very hard to maintain over time |
| Variable Scoping | Since using Advanced Java as the platform for development, thus no global variables had been used. All the variables declared are local to the function where they are declared. | Try to keep variables scoped to the lowest level possible. Global variables should be avoided. |
| [Cohesion](http://en.wikipedia.org/wiki/Cohesion_(computer_science)) of Logic in Classes | The functions used in the Admin class have their functionality purely independent of the other and each uses unique parameters to implement the functionality, which are to be supplied by the human via the interface, and not by any other function. | Do everything you can to encourage high cohesion in all classes. |
| Coupling: Long Parameter Lists | Short parameter lists had been used to implement the designed functionality. This makes the code easy to maintain in future. | Generally, methods with long parameter lists create higher degrees of coupling and therefore decrease maintainability |
| Coupling: Control Coupling | Control coupling has been implemented in the form of flags used which return the assigned value when the function is called. Based upon these values the decision is taken whether the function has executed or not. It has been implemented in Admin class in functions such as deleteUser(), login(), modifyHome(), updateUser(), updateLatestbooks(), updateJobboard(), updateNews(), uploadMaterials(). | If you use “control flags” to drive the internal behavior of a method, explore opportunities for specialized classes or overloaded methods |
| Coupling: Global Data Coupling | The session variable is used as a type of global variable which stores the username of the Administrator. The username is displayed on the home page of the Administrator. | This typically occurs when global variables are used to drive the behavior of a group of classes |
| Coupling: Solution Sprawl Across Classes | The Administrator can delete the user from the system thus this will affect the modules of Books. Also no functionality has been sprawled across other classes and is limited to a single class where the function is implemented. | When you need to make changes across a large number of classes in order to implement a change in the application’s behavior, you’ve got solution sprawl. |
| Coupling: Inter-Layer Dependencies | The Admin class has a large number of Inter-Layer Dependencies as initially the values are received from the interface to the AdminPage class which is a .jsp page and then these values are used to call the functions in the Admin class. | The more method calls you have from one class in a given assembly or “logical layer” to other classes in a different assembly or “logical layer”, the tighter the coupling, and the harder it will be to maintain over time. |
| Conditional Complexity, Level of Nesting, Use of Flags, use of switch statements | No deeply nested “If” statements have been used in implementing the logic. The Logic is directly implemented by defining each in a separate function for independent functionality. | Consider applying the “[Extract Method](http://refactoring.com/catalog/extractMethod.html)” refactoring to move code from within an “If Block” to a method that describes what that code does.  Whenever you see switch statements or “If” statements in a class, you might have an opportunity to use class specializations instead.  If a method has deeply nested “If” statements, uses flags (e.g. Booleans, etc.) to drive logic, it can become very difficult to read and maintain |
| [Encapsulation](http://en.wikipedia.org/wiki/Information_hiding), Information Hiding, Inappropriate Intimacy between Classes | Since there is only a single class Admin which is used to drive the logic of the Administrator functionality and a .jsp page comprising of a AdminPage class which is used to call the functions of the Admin class, however it doesn’t knows much about the Admin class. | When classes know too much about the internals of each other, they become very tightly coupled and hard to maintain. |
| Magic Numbers and Literals | Not Applicable. | Try to replace magic numbers and literals with constants that have meaningful names. Magic Numbers and literals are numeric or alpha-numeric values in the code whose meaning may not be self-explanatory. |
| Speculative Generality | Every functionality designed has been used in the solution. | A.K.A. [You Aren’t Going to Need It](http://www.extremeprogramming.org/stories/simple2.html) |
| Versioning Approach | The logic of the module has been programmed by using Servlet and JSP page, further versioning can be implemented by using Struts, as a logic of implementation leading to advanced version. | Has the developer produced an approach that may be easily versioned over time? |
| Use of Interfaces | The interfaces are reused leading to high reusability of the code. The interfaces are simple, user-friendly and full of instructions of how the operation is done. | Are interfaces used appropriately? |
| Simplicity of Solution | The designed solution is simple and user-friendly. |  |
| General readability and intuitive naming of fields, properties, variables, methods, etc. | The name of variables, methods used in the module have an appropriate name applying the same logic as to be used in the application of module. |  |
| [Appropriate Use of Comments](http://www.designpatternsfor.net/default.aspx?pid=24) | The inline comments have been used in the module. |  |
| General adherence to Microsoft Coding Standards |  |  |
| Unit-Tests were created to support [regression testing](http://en.wikipedia.org/wiki/Regression_testing) | The various unit tests for the Administrator module are implemented in the test case document and tested and is found working correct against these tests. |  |

# Robustness

Assess the primary factors that affect how well the software handles incorrect data or unforeseen scenarios.

|  |  |  |
| --- | --- | --- |
| **Area of Review** | **Observations** | **Suggestions** |
| Defensive Programming | The values received through the interface of the module are checked that they may be entered into the table for insertion, and for modification that the value for the modification already exist in the module. | Performs early parameter checking (e.g. boundary checks, type checks, assertions, etc.) before executing main body of logic  Checks return values received from service or method calls  Checks for nulls when appropriate  *Avoids “Apocalypse Ready” designs; These are designs that handle exceptions that will probably never happen or “fringe case” issues* |
| Proper use of Exception Handling | The try-catch statement has been used to implement the Exception Handling. In order to avoid errors t be handled by the programmer the exceptions are thrown to Java which handles these exceptions and displays the appropriate error message.  The use of try-catch enables the system to keep working even when one of its methods do not function correctly. | All exceptions are caught and handled at the “top of call-stack”  Lower in the call stack, exceptions are only caught to log or gather information, add information to the exception, perform cleanup, or attempt to recover  Prefer the use of standard framework-defined exceptions when possible  Exceptions are thrown only for clearly abnormal cases; Exceptions aren’t used to control application flow. |
| Exceptions are Logged to Facilitate Debugging | The thrown exceptions may be used by the developer to repair and maintain the code accordingly. |  |
| Parameters are strongly typed | The parameters used in the Student module had been declared beforehand and are all defined data type of Java and we know Java is strongly typed. |  |

# Performance

Assess the areas that will typically have the greatest impact on application performance.

|  |  |  |
| --- | --- | --- |
| **Area of Review** | **Observations** | **Suggestions** |
| Style of Communication | The query or object retrieves only that much information from the database which is asked in the logic or requested by the user; the view records function of the interface displays only the needed records of the user to the Administrator hiding the confidential information such as username and password of the user.  The code also has minimal cross-process class and finishes a logic in one function only. | Code has minimal cross-machine calls  Code has minimal cross-process calls  Code favors chunky vs. chatty communications to services  Avoidance of “Data Buffet” anti-pattern; This occurs when a query or object retrieves more data than it’s consumer will probably use |
| Evaluate use of [Boxing / Unboxing](http://msdn.microsoft.com/msdnmag/issues/1200/dotnet/) | Conversion from value type to object type and then again back to value type has been not implemented in the module. | Avoid conversion to/from value and reference types where possible |
| Loop considerations | The loops terminate as soon as the logic has been met, the expression are evaluated after the loop has terminated to store the final value only.  Logic getting the same results are not used within the body of the loop. | Loops are exited as soon as conditions met  Expressions are not re-evaluated from within the loop controller statements  Logic that always gets same results does not occur within the body of loops  Use “For” instead of “For Each” when appropriate |
| String Handling | Not string concatenation and StringBuilder approaches are used as no appending i.e to join or add on to the end of something is done. | Assess string concatenation approaches  Assess use of StringBuilder |
| Resource Cleanup | The imports used in the module are essential for implementing the logic and are acquired late and released early.  The finalizers are not used in the module to avoid function overlapping i.e., whether the logic has executed or not the statements in the final block are implemented. | Resources are acquired late and released early  Assess potential for [Generation 2 garbage collection](http://msdn2.microsoft.com/en-us/library/ms973837.aspx)  Evaluate appropriate use of “using” statement  The [Dispose pattern](http://msdn2.microsoft.com/en-us/library/b1yfkh5e(VS.71).aspx) is implemented for managed resources  Finalizers are avoided |
| Appropriate Use of Caching | Since in Admin module each value is a different one thus nothing is cached and is not used frequently.  Every value used in the function is unique and is implemented only when the values are supplied via the interface. | Items that change frequently are not cached  Know your cache-hit ratio; don’t bother caching items that aren’t retrieved frequently |
| Appropriate use of ViewState and Postback checking | We have not stored the ViewState on the server side in a session and then passing the viewstate id to the client side via a hidden field. However when a user signs in,his session is maintained and when he logs out the session expires and he cannot roll back. |  |
| Consider opportunities for Asynchronous or Queued Operations | No need was found to perform the Queued operations. |  |
| Solution makes minimal or no use of Reflection | No use of reflection is done. |  |
| Use of Code Instrumentation | No use of Code Instrumentation. |  |
| [Premature Optimization](http://en.wikipedia.org/wiki/Optimization_(computer_science))? | Premature optimization means if the value of core product is weak, doubling the percentage of users will not help much and it will hurt as every unit of effort put into optimization is one less unit that can be put into improving core product. We will not begin optimizing until users say that they are very disappointed with the product. Hence there is no premature optimization. |  |

# Supporting Documentation

Assess the design solution’s need for supporting documentation

|  |  |  |
| --- | --- | --- |
| **Area of Review** | **Observations** | **Suggestions** |
| Object Models | The Object Models are not implemented in this module. |  |
| Sequence Diagrams | The sequence diagram has not been implemented as the developed solution is a web-application and to define the timeline for a particular user is very difficult. |  |
| Entity-Relationship Diagrams | The ER diagram used to depict the Administrator module proved very useful as most of the functioning has been clearly stated in it and during coding proved very helpful in deciding the logic of the code. |  |
| Other Diagram | The class diagram also proved to be very useful in implementing the logic as all the name of functions, their return type, their variable names had been decided earlier; however the variables used to hold the values supplied through the interface has not been declared and defined in this diagram |  |
| Use of [nDoc](http://sourceforge.net/projects/ndoc/) or [XML Comments](http://msdn.microsoft.com/msdnmag/issues/02/06/XMLC/) | No use of XML comments. But comments are provided to make a clear understanding in Student module. |  |
| Where will Documentation be Stored? How will it be Maintained? | The various templates holds the documentation and can be maintained easily. |  |

# Migration Considerations

Assess how current or future users (i.e. developers) dependent on this solution will migrate to the proposed approach. How will the risk of breaking changes be minimized?

The sole aim of Online Bookstore is to find a collection of good and reliable books for the students and applied to parents for their wards as to find a good and reliable information provider in contemporary world is a cumbersome task. The solution can be used by the Coaching Institutes, users from remote locations etc. The solutions originally developed under Advanced Java framework provides a base and can be migrated to other platforms by breaking this project module-wise.

* The individual source code files which were written in pure Java have to be rewritten in the proposed environment language.
* The ojdbc Jar files has to be suitably replaced by the libraries of the proposed new environment.
* The HTML, CSS and JavaScript files hardly will need any changes because they are independent of the platform in which they are used and are dependent on the browsers interpretation of them.
* If the migration is being done to a environment which supports Object Oriented Programming paradigms then classes which are defined in this project need not be redefined. Only certain names and appropriate statements need to be written.
* The pre requirement for portability is the generalized abstraction between the application logic and system interfaces.

|  |  |
| --- | --- |
| **Item(s) to Review** | **USER LOGIN** |
| **Description of Item(s), Background and Context** | The User module of the project Online Bookstore implements the basic functions of Register to the portal, Search Books, Comment for particular Book, accessing online material, check for payments and Download online materials. |
| **Objectives of Review** | The sole objective of the review is to correct the incorrect functionality , if any of functions are left to be completed must be completed and code fixing. |
| **Owner(s) of Item(s)** | Shakti Singh, Shubham Goel, Shubham Kumar, Ammar Arfeen, Md Imteyaz |
| **Who is Impacted? Who Depends on these Items?** | No other module is impacted. |
| **Reviewed by** | Shakti Singh, Shubham Goel, Shubham Kumar, Ammar Arfeen, Md Imteyaz |
| **Next Steps** | The next step to be taken after review is to enter into maintenance phase. |

# Logical Design and Layering Approach

Many web portals claim to provide a good selection of books for the students and parents who are looking for a one stop solution for their problems. The different web portals functioning now a days’ promise their user’s (the registered ones) to provide them with suitable books and materials for the subjects for which they are looking for, and lead them to these portals more often. These Web portals has its own unique style of suggesting books and imparting knowledge to the students, and providing them with their own course materials, there is no way to generate reviews about how these online portals are performing in the market and what the students and other people thinks about such service.

Also, their exist no way that parents or students as well may give views about how they think about the portal. However, these reviews hold a high opinion for other new users who want to join the portal.

**Online Bookstore** is a user friendly web-portal which provides solution to all these above stated problems from providing one-stop solution to provision of online Book purchasing through the portal depending upon the availability of the books, to the provision of online course material which is authentic and up to the standards.

Provide commentary on the solution design highlighting appropriateness for problem domain, advantages, and potential trade-offs.

# Maintainability, Adaptability

Assess factors that will impact the ease with which the software may be maintained or altered over time.

|  |  |  |
| --- | --- | --- |
| **Area of Review** | **Observations** | **Suggestions** |
| Duplicated Code | The code was duplicated in Password class ,User.Comment class, User, Login class and User.Register class where it is reused for implementing various functions such as forgotP(), comment(), login(), and register().  The code is reused as it allows us to save effort and time to redesign the solution again and again. | Whenever you see code that looks like it was copied and pasted, even with a few minor changes, explore opportunities for reuse. You might consider implementing classes that encapsulate the logic, or creating helper classes. |
| Use of Short Methods | Short methods are used in User module for implementing functions such as login(), register(), comment() etc. This makes the code maintenance high. | Try to keep all methods short. Long methods become very hard to maintain over time |
| Variable Scoping | Since using Advanced Java as the platform for development, thus no global variables had been used. All the variables declared are local to the function where they are declared. | Try to keep variables scoped to the lowest level possible. Global variables should be avoided. |
| [Cohesion](http://en.wikipedia.org/wiki/Cohesion_(computer_science)) of Logic in Classes | The functions used in the User module have their functionality purely independent of the other and each uses unique parameters to implement the functionality, which are not passed by any other function. | Do everything you can to encourage high cohesion in all classes. |
| Coupling: Long Parameter Lists | Short parameter lists had been used to implement the designed functionality. This makes the code easy to maintain in future. | Generally, methods with long parameter lists create higher degrees of coupling and therefore decrease maintainability |
| Coupling: Control Coupling | Control coupling has been implemented in the form of flags used which return the assigned value when the function is called. Based upon these values the decision is taken whether the function has executed or not. It has been implemented in User module class in functions such as comment(), forgotP(), login(), register(), and search. | If you use “control flags” to drive the internal behavior of a method, explore opportunities for specialized classes or overloaded methods |
| Coupling: Global Data Coupling | The session variable is used as a type of global variable which stores the username of the User. The username is displayed on the home page of the User. | This typically occurs when global variables are used to drive the behavior of a group of classes |
| Coupling: Solution Sprawl Across Classes | Not applicable. | When you need to make changes across a large number of classes in order to implement a change in the application’s behavior, you’ve got solution sprawl. |
| Coupling: Inter-Layer Dependencies | The User module has no Inter-Layer Dependencies as single methods are  being called from class to implement a  particular functionality. | The more method calls you have from one class in a given assembly or “logical layer” to other classes in a different assembly or “logical layer”, the tighter the coupling, and the harder it will be to maintain over time. |
| Conditional Complexity, Level of Nesting, Use of Flags, use of switch statements | No deeply nested “If” statements have been used in implementing the logic. The Logic is directly implemented by defining each in a separate function for independent functionality. | Consider applying the “[Extract Method](http://refactoring.com/catalog/extractMethod.html)” refactoring to move code from within an “If Block” to a method that describes what that code does.  Whenever you see switch statements or “If” statements in a class, you might have an opportunity to use class specializations instead.  If a method has deeply nested “If” statements, uses flags (e.g. Booleans, etc.) to drive logic, it can become very difficult to read and maintain |
| [Encapsulation](http://en.wikipedia.org/wiki/Information_hiding), Information Hiding, Inappropriate Intimacy between Classes | Since every class of User module are independent and performs a specific functionality so no class know about the internals of other class and it is highly encapsulated. | When classes know too much about the internals of each other, they become very tightly coupled and hard to maintain. |
| Magic Numbers and Literals | No such magic numbers and literals used. | Try to replace magic numbers and literals with constants that have meaningful names.  Magic Numbers and literals are numeric or alpha-numeric values in the code whose meaning may not be self-explanatory. |
| Speculative Generality | The User module performs only those basic functions as mentioned in SRS. So every function is important and needed. | A.K.A. [You Aren’t Going to Need It](http://www.extremeprogramming.org/stories/simple2.html) |
| Versioning Approach | The logic of the module has been programmed by using Servlet and JSP as a logic of implementation leading to advanced version. | Has the developer produced an approach that may be easily versioned over time? |
| Use of Interfaces | The interfaces are reused leading to high reusability of the code. The interfaces are simple, user-friendly and full of instructions of how the operation is done. | Are interfaces used appropriately? |
| Simplicity of Solution | The designed solution is simple and user-friendly. |  |
| General readability and intuitive naming of fields, properties, variables, methods, etc. | The name of variables, methods used in the module have an appropriate name applying the same logic as to be used in the application of module. |  |
| [Appropriate Use of Comments](http://www.designpatternsfor.net/default.aspx?pid=24) | The comments have been used in the module. |  |
| General adherence to Microsoft Coding Standards | Not applicable. |  |
| Unit-Tests were created to support [regression testing](http://en.wikipedia.org/wiki/Regression_testing) | The various unit tests for the User module are implemented in the test case document and tested and is found working correct against these tests. |  |

# 

# Robustness

Assess the primary factors that affect how well the software handles incorrect data or unforeseen scenarios.

|  |  |  |
| --- | --- | --- |
| **Area of Review** | **Observations** | **Suggestions** |
| Defensive Programming | The values received through the interface of the module are checked and validated through javavscript that they may be inserted into the table and no field can be left blank i.e no null value can be stored in the table. | Performs early parameter checking (e.g. boundary checks, type checks, assertions, etc.) before executing main body of logic  Checks return values received from service or method calls  Checks for nulls when appropriate  Avoids “Apocalypse Ready” designs; These are designs that handle exceptions that will probably never happen or “fringe case” issues |
| Proper use of Exception Handling | The try-catch statement has been used to implement the Exception Handling. In order to avoid errors t be handled by the programmer the exceptions are thrown to Java which handles these exceptions and displays the appropriate error message.  The use of try-catch enables the system to keep working even when one of its methods do not function correctly. | All exceptions are caught and handled at the “top of call-stack”  Lower in the call stack, exceptions are only caught to log or gather information, add information to the exception, perform cleanup, or attempt to recover  Prefer the use of standard framework-defined exceptions when possible  Exceptions are thrown only for clearly abnormal cases; Exceptions aren’t used to control application flow. |
| Exceptions are Logged to Facilitate Debugging | The thrown exceptions may be used by the developer to repair and maintain the code accordingly. |  |
| Parameters are strongly typed | The parameters used in the user module had been declared beforehand and are all defined data type of Java. And we know Java is strongly typed. |  |
| Assess Potential for Data Loss Due to “Shortening Conversions” | No conversion has been implemented for  “Shortening”, thus avoiding the data loss due to this factor. | This occurs when you attempt to cast larger types like longs to smaller types like integers. |

# Performance

Assess the areas that will typically have the greatest impact on application performance.

|  |  |  |
| --- | --- | --- |
| **Area of Review** | **Observations** | **Suggestions** |
| Style of Communication | The query or object retrieves only that much information from the database which is asked in the logic or requested by the user; the search function of the User module interface displays only the needed records of the tutor, no confidential data is displayed.  The code also has minimal cross-process class and finishes a logic in one function only. | Code has minimal cross-machine calls  Code has minimal cross-process calls  Code favors chunky vs. chatty communications to services  Avoidance of “Data Buffet” anti-pattern; This occurs when a query or object retrieves more data than it’s consumer will probably use |
| Evaluate use of [Boxing / Unboxing](http://msdn.microsoft.com/msdnmag/issues/1200/dotnet/) | Conversion from value type to object type and then again back to value type has been not implemented in the module. | Avoid conversion to/from value and reference types where possible |
| Loop considerations | The loops terminate as soon as the logic has been met, the expression are evaluated after the loop has terminated to store the final value only.  Logic getting the same results are not used within the body of the loop.  No logic as used in Parent module gets the same results. | Loops are exited as soon as conditions met  Expressions are not re-evaluated from within the loop controller statements  Logic that always gets same results does not occur within the body of loops  Use “For” instead of “For Each” when appropriate |
| String Handling | Not string concatenation and StringBuilder approaches are used as no appending i.e to join or add on to the end of something is done. | Assess string concatenation approaches  Assess use of StringBuilder |
| Resource Cleanup | The imports used in the module are essential for implementing the logic and are acquired late and released early.  The finalizers are not used in the module to avoid function overlapping i.e., whether the logic has executed or not the statements in the final block are implemented. | Resources are acquired late and released early  Assess potential for [Generation 2 garbage collection](http://msdn2.microsoft.com/en-us/library/ms973837.aspx)  Evaluate appropriate use of “using” statement  The [Dispose pattern](http://msdn2.microsoft.com/en-us/library/b1yfkh5e(VS.71).aspx) is implemented for managed resources  Finalizers are avoided |
| Appropriate Use of Caching | Since in User module each value is a different one thus nothing is cached and is not used frequently.  Every value used in the function is unique and is implemented only when the values are supplied via the interface. | Items that change frequently are not cached  Know your cache-hit ratio; don’t bother caching items that aren’t retrieved frequently |
| Appropriate use of ViewState and Postback checking | We have not stored the ViewState on the server side in a session and then passing the ViewState id to the client side via a hidden field. However when a user signs in, his session is maintained and when he logs out the session expires and he cannot roll back. |  |
| Consider opportunities for Asynchronous or Queued Operations | No need was found to perform the Queued operations. |  |
| Solution makes minimal or no use of Reflection | No use of reflection is done. |  |
| Use of Code Instrumentation | No use of Code Instrumentation is done in Parent module. |  |
| [Premature Optimization](http://en.wikipedia.org/wiki/Optimization_(computer_science))? | Premature optimization means if the value of core product is weak, doubling the percentage of users will not help much and it will hurt as every unit of effort put into optimization is one less unit that can be put into improving core product. We will not begin optimizing until users say that they are very disappointed with the product. Hence there is no premature optimization. |  |

# Supporting Documentation

Assess the design solution’s need for supporting documentation

|  |  |  |
| --- | --- | --- |
| **Area of Review** | **Observations** | **Suggestions** |
| Object Models | The Object Models are not implemented in this module. |  |
| Sequence Diagrams | Not mentioned in design templates. |  |
| Entity-Relationship Diagrams | The ER diagram used to depict the User module proved very useful as most of the functioning has been clearly stated in it and during coding proved very helpful in deciding the logic of the code. |  |
| Other Diagram | The class diagram also proved to be very useful in implementing the logic as all the name of functions, their return type, their variable names had been decided earlier; however the variables used to hold the values supplied through the interface has not been declared and defined in this diagram |  |
| Use of [nDoc](http://sourceforge.net/projects/ndoc/) or [XML Comments](http://msdn.microsoft.com/msdnmag/issues/02/06/XMLC/) | No use of XML comments .But comments are provided to make a clear understanding in User module. |  |
| Where will Documentation be Stored? How will it be Maintained? | The various templates holds the documentation and can be maintained easily. |  |

# Migration Considerations

Assess how current or future users (i.e. developers) dependent on this solution will migrate to the proposed approach. How will the risk of breaking changes be minimized?

The sole aim of Online Bookstore is to find a collection of good and reliable books for the students and applied to parents for their wards as to find a good and reliable information provider in contemporary world is a cumbersome task. The solution can be used by the Coaching Institutes, users from remote locations etc. The solutions originally developed under Advanced Java framework provides a base and can be migrated to other platforms by breaking this project module-wise.

* The individual source code files which were written in pure Java have to be rewritten in the proposed environment language.
* The ojdbc Jar files has to be suitably replaced by the libraries of the proposed new environment.
* The HTML, CSS and JavaScript files hardly will need any changes because they are independent of the platform in which they are used and are dependent on the browsers interpretation of them.
* If the migration is being done to a environment which supports Object Oriented Programming paradigms then classes which are defined in this project need not be redefined. Only certain names and appropriate statements need to be written.
* The pre requirement for portability is the generalized abstraction between the application logic and system interface.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| * **Version** | | | * **Description** | | * **Created By** | * **Date** |
| * ***1.0*** | | | * ***First version of the system*** | | * ***Shakti Singh, Shubham Goel, Shubham Kumar, Ammar Arfeeen, Md Imteyaz.*** | * ***30-08-2019*** |
| * **General Description** | | | | | | |
| * *A use case is a methodology used in system analysis to identify, clarify, and organize system requirements. The use case is made up of a set of possible sequences of interactions between systems and users in a particular environment and related to a particular goal. It consists of a group of elements (for example, classes and interfaces) that can be used together in a way that will have an effect larger than the sum of the separate elements combined. The use case should contain all system activities that have significance to the users. A use case can be thought of as a collection of possible scenarios related to a particular goal, indeed, the use case and goal are sometimes considered to be synonymous.* * *A use case (or set of use cases) has these characteristics:* * *Organizes functional requirements* * *Models the goals of system/actor (user) interactions* * *Records paths (called scenarios) from trigger events to goals* * *Describes one main flow of events (also called a basic course of action), and possibly other ones, called exceptional flows of events (also called alternate courses of action)* * *Is multi-level, so that one use case can use the functionality of another one.* * *Use cases can be employed during several stages of software development, such as planning system requirements, validating design, testing software, and creating an outline for online help and user manuals.* | | | | | | |
| * **Actors *<an entity that can interact with a system, invoking some behavior>*** | | | | | | |
| * Administrator * User * Student * Books. | | | | | | |
| * **Preconditions *<the state(s) the system can be in before this use case starts>*** | | | | | | |
| * The System is functioning properly * The System should be connected to the Internet * The System should have access to the database * The User should have knowledge of working on internet. * The User must have a valid E-mail id. * The User must have a valid Debit Card or an account in the bank. | | | | | | |
| * **Use Case Diagram <*a* *specific sequence of actions and interactions between actors and the system being discussed*>** | | | | | | |
| * **ONLINE BOOKSTORE System(Major) Use Case Diagram** | | | | | | |
| * **Basic flow of events: <*a* *specific sequence of actions and interactions between actors and the system being discussed (happy path Or basic course of action)*>** | | | | | | |
| * **Line** | | * **System Actor Action** | | * **System Response** | | |
|  | | * Administrator Manages(Deletes User, Modifies User Details) | | * Student, User, Books | | |
|  | | * Administrator Updates | | * Latest Books, Job Board, News & Update, Payments | | |
|  | | * Administrator Uploads | | * Online Materials | | |
|  | | * Administrator Moderates Comments on | | * Student, Books | | |
|  | | * Administrator Receives Payments to grant access from | | * Student, Books, Users | | |
|  | | * Student Searches | | * Books | | |
|  | | * Student Comments on | | * Books | | |
|  | | * Student Downloads | | * Online Materials | | |
|  | | * Student Makes Payments to | | * Administrator | | |
|  | | * User Searches | | * Books | | |
|  | | * User Comment on | | * Books | | |
|  | | * User Download | | * Online materials | | |
|  | | * User Makes Payment to | | * Administrator | | |
| * **Exceptional flow of events: *<an alternate sequence of actions and interactions between the actors and the system (alternate course of action)>*** | | | | | | |
| * 1 | * Administrator Updates for Student, Tutor, Parent. | | | * Payments, News & Updates, Latest Tutor, and Job Board. | | |
| * 2 | * Administrator manages | | | * Student, Books and Users. | | |
| * 3 | * Administrator manages(moderate’s comments) | | | * Comment User on Books , Student comment | | |
| * 4 | * Administrator receives payment for registering new user and giving access to online materials. | | | * Student, Books and User. | | |
| * **Post Conditions <*the state(s) the system can be in when this use case ends>*** | | | | | | |
| * The logout operation must enable the logout of the user from the current session. * The feedback provided by the user must be entertained in the system if applicable. * The maintenance of the system must be done regularly to keep the system running. * The system must be monitored regularly in order to avoid slow responsiveness and working of the system. | | | | | | |
| * **System Messages *<all system generated messages that the system will prompt for the user>*** | | | | | | |
| * 1 | * Login-Login successful or Login Failed: Re-enter your username and/or password. | | | | | |
| * 2 | * Search-Search Results not found. Please modify your search. | | | | | |
| * 3 * 4 * 5 * 6 | * Register-Successfully registered or you are already a member. * Upload-Successfully uploaded or failed to upload or slow internet connection. * Download-Successfully downloaded or failed to download due to slow internet connection. * Online Teaching- You have slow internet connection or tutor is not available or You appear to be offline. | | | | | |
|  | | | | | | |
| * **Appendix *<List of any supporting documents attached>*** | | | | | | |
| * <http://searchsoftwarequality.techtarget.com/definition/use-case> * <http://en.wikipedia.org/wiki/Use_case> | | | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **CHECKLIST FOR PROJECT REVIEW INTERVIEWS** | | | | |
|  | * **Yes** | * **No** | * **N/A** | * **Remarks** |
| * **Project Planning** | | | | |
| * Is it clear how the goals and objectives of this project fit within the overall goals and objectives of the organization (i.e., the bigger picture)? | * Yes |  |  |  |
| * Was the knowledge base used in planning the project (e.g., in accessing previous experience, defining activities, tasks, and deliverables, obtaining checklists and templates to kick-start project deliverables)? | * Yes |  |  | * Various templates like information gathering, project understanding, synopsis, SRS, design templates have been made before. |
| * Are the deliverables clearly defined (e.g., size, format, medium)? | * Yes |  |  | * The templates were well defined and provided a clear picture about the project. |
| * Has the process for acceptance been established for each deliverable? | * Yes |  |  |  |
| * Have the acceptance criteria been defined for each deliverable? | * Yes |  |  |  |
| * Is there a Risk Management Log listing the priority risks, and are Risk Management Action Plans in place for the seven to ten most significant risks? |  |  | * N/A |  |
| * Is the Work Breakdown Structure defined at an appropriate level of detail (i.e., tasks of manageable size - 5 to 10 days - at the level at which the work is managed)? | * Yes |  |  | * The Work Breakdown Structure template was clearly defined. |
| * Are appropriate estimating techniques being used (e.g., task-based, function points)? |  |  | * N/A |  |
| * Is there a detailed schedule in place? | * Yes |  |  | * The Work Breakdown Structure template clearly defined the schedule. |
| * Does the schedule clearly identify the major milestones and the dependencies between work components? | * Yes |  |  |  |
| * Is documentation available to substantiate the estimates and schedules (e.g., spreadsheets and working papers detailing the process followed and the assumptions made)? | * Yes |  |  |  |
| * Is the Project Organization appropriate? | * Yes |  |  | * The module has been divided within the Team members. |
| * Does the project have adequately skilled resources? | * Yes |  |  |  |
| * Is the project using the right level of resources? | * Yes |  |  |  |
| * Does the project team include an appropriate level of previous experience with this customer? |  | * No |  |  |
| * Does the project team include an appropriate level of previous experience with this application? |  | * No |  |  |
| * Does the project team include an appropriate level of previous experience with this technology? |  | * No |  |  |
| * Is there a staff rotation plan in place to rotate individuals off the project, or into new roles on the project, in a 12 to 18 month time frame? |  |  | * N/A |  |
| * Is there an approved financial plan that tracks the expected monthly cash flow, as the basis for monitoring the projectâ€™s financial performance? |  |  | * N/A |  |
| * Is there an appropriate level of cost and schedule contingency included in the plan? |  |  | * N/A |  |
| * Are standards and procedures put in place ahead of time for all deliverables and processes? | * Yes |  |  | * The conventions are followed. |
| * Is there a documented Project Management Plan that is consistent with the template defined in the knowledge base? |  |  | * N/A |  |
| * **Contract and Financial** | | | | |
| * Is there a signed contract in place (or a signed contract equivalent that allows work to proceed while the contract is being negotiated)? |  |  | * N/A |  |
| * Are the key contractual requirements clearly understood by the project team? |  |  | * N/A |  |
| * Are any necessary third party agreements signed and in place? |  |  | * N/A |  |
| * Is there a contract file that provides a comprehensive record of all dealings under the contract? |  |  | * N/A |  |
| * Is the project obtaining prompt customer payment of all moneys due under the contract? |  |  | * N/A |  |
| * Is the project meeting its targeted financial performance? |  |  | * N/A |  |
| * **Scope and Requirements** | | | | |
| * Does the scope statement clearly identify the boundaries of the project (i.e., will it be clear to all parties whether a proposed change is in or out of scope)? | * Yes |  |  | * The Project Scope has been clearly defined. |
| * Is the scope quantified and measurable (e.g., by identifying the expected size of the target system in function points)? |  |  | * N/A |  |
| * Are the requirements clearly defined and traceable to the project deliverables? | * Yes |  |  |  |
| * Have the requirements been signed-off by the Acceptor and formally baselined? |  |  | * N/A |  |
| * Are Change Control procedures in place and are they being followed for all changes to baselined deliverables? |  |  | * N/A |  |
| * Are Decision Request procedures in place and are they being followed? | * Yes |  |  |  |
| * Are the scope and requirements stable? | * Yes |  |  | * The scope and requirements are stable. |
| * **Customer Relationship** | | | | |
| * Is there a single overall Acceptor? |  | * No |  |  |
| * Is there frequent formal and informal contact with customer executives to review progress and maintain open communication? |  |  | * N/A |  |
| * Is a summary status report distributed to the customer on a regular basis (weekly or monthly)? |  |  | * N/A |  |
| * Is the customer performing adequate planning for their involvement? |  |  | * N/A |  |
| * Is there a Steering Committee in place that meets monthly? |  |  | * N/A |  |
| * Are key users involved and informed? |  |  | * N/A |  |
| * Are users contributing to the solution? |  |  | * N/A |  |
| * Is the customer organization committed to the success of the project? |  |  | * N/A |  |
| * Is the customer fair and reasonable? |  |  | * N/A |  |
| * Is the customer relationship good (i.e., there are no significant outstanding issues)? |  |  | * N/A |  |
| * **Team Performance** | | | | |
| * Are standards and procedures in place covering the general conduct of the team (e.g., project terminology, routine communications, meeting procedures)? | * Yes |  |  |  |
| * Have all team members received a full and sufficient orientation to the project? | * Yes |  |  |  |
| * Are all team members adequately trained to carry out their responsibilities? | * Yes |  |  | * The training has been given to all team members. |
| * Are all team member roles and responsibilities clearly defined and understood? | * Yes |  |  | * The role and responsibilities have been divided between the team members. |
| * Have project and personal objectives been defined for each team member? | * Yes |  |  |  |
| * Do team members interact effectively? | * Yes |  |  |  |
| * Does the team meet regularly to review status, share experiences, provide suggestions, and resolve problems and concerns? | * Yes |  |  |  |
| * Is project information posted/circulated? | * Yes |  |  | * The team leader sends email to every team member on the regular basis. |
| * Has the project implemented some means to recognize and reward quality service, creativity, innovation, and success (can be as simple as public acknowledgment and thanks for a job well-done)? | * Yes |  |  |  |
| * Is team morale high? | * Yes |  |  |  |
| * Is the productivity of the team at an optimum level? | * Yes |  |  |  |
| * Is project meeting time properly managed and are project meetings run effectively? | * Yes |  |  |  |
| * Are performance reviews conducted to record and communicate the accomplishments of team members, and to appraise their skills, to facilitate career development? | * Yes |  |  |  |
| * Is turnover within acceptable limits? |  |  | * N/A |  |
| * **Cost/Schedule Performance** | | | | |
| * Is the project on schedule and within budget? | * Yes |  |  |  |
| * Will the deliverables being produced meet the userâ€™s requirements? | * Yes |  |  |  |
| * Is there a current and reliable version of the project work plans that includes approved changes and current estimates to completion? | * Yes |  |  |  |
| * Is the schedule realistic now? | * Yes |  |  | * The schedule has been followed as mentioned in the Work breakdown Structure and directed by our Project Guide. |
| * Are team members clear about how each work product is to be produced (e.g., who is responsible for delivery, who will review, related or impacted deliverables, when the deliverable is required)? |  |  | * N/A |  |
| * Are team members tracking detailed progress against each task and reforecasting the remaining effort to completion and the expected completion date for each task, on a regular weekly basis? | * Yes |  |  | * The templates are reviewed regularly. |
| * Are team members preparing weekly status reports in a timely manner (narrative, tracking, and change/decision request logs)? |  |  | * N/A |  |
| * Does the Project Manager produce a Monthly Status Report to keep the projectâ€™s status visible to senior management in the customer and delivery organization? | * Yes |  |  |  |
| * Is corrective action taken to resolve significant cost or schedule variances, such as replanning the remaining work or taking action to improve performance? | * Yes |  |  |  |
| * Is project overtime within acceptable limits? |  | * No |  |  |
| * Is all time worked being recorded? | * Yes |  |  |  |
| * **Data Management** | | | | |
| * Are standards and procedures in place to control all of the data that support the project (e.g., document standards, configuration management standards and procedures)? |  |  | * N/A |  |
| * Are project records being established, maintained, and disposed of in accordance with the standards and procedures? | * Yes |  |  | * The templates and code backup has been maintained. |
| * Has an electronic and paper copy of all versions of all baselined deliverables been maintained? | * Yes |  |  |  |
| * Is a Project Book in place and is this being maintained? |  |  | * N/A |  |
| * Is an automated Project Index in place and is this being maintained (for projects of more than 15 people)? |  |  | * N/A |  |
| * Is all software controlled in accordance with the software configuration management standards and procedures? |  |  | * N/A |  |
| * Is a data management system used to record, communicate, and track the status of all data items to be delivered to the customer? |  |  | * N/A |  |
| * Are all action items identified, recorded, communicated and tracked to closure? |  |  | * N/A |  |
| * Are all project data backed up and are off-site back-ups in place in accordance with the company records management procedures? | * Yes |  |  | * The back-ups are maintained regularly. |
| * Is there an up-to-date record of the project in the project experience repository that provides data on the projectâ€™s processes, metrics, and lessons learned? |  |  | * N/A |  |
| * **Quality Management** | | | | |
| * Has the Quality Assurance role been identified and assigned, and does this have an appropriate level of authority? | * Yes |  |  | * The conventions are followed. |
| * Is there a documented Quality Management Plan that is consistent with the template defined in the knowledge base? |  |  | * N/A |  |
| * Are standards and procedures in place for all quality management activities and tasks (e.g., conducting walk-throughs, reviews, audits, inspections, handling and storing of products)? |  |  | * N/A |  |
| * Have the quality requirements been identified (i.e., the customerâ€™s expectations for quality, the quality factors, and the quality metrics that will be used to give visibility to the levels of quality being achieved)? | * Yes |  |  | * All the requirements have been identified. |
| * Are formal structured walk-throughs and reviews being conducted in accordance with the standards and procedures (e.g., the proper people attend, minutes are taken and distributed, and action items are assigned and followed up)? |  |  | * N/A |  |
| * Are appropriate test plans and procedures in place and are they being followed? | * Yes |  |  |  |
| * Is project data being analyzed to determine trends and norms and to identify when quality threshold or target levels are not being achieved? | * Yes |  |  |  |
| * Are group process techniques being used to involve all team members in the review and continual improvement of project work processes? | * Yes |  |  |  |
| * Is the established acceptance process being followed (including signed acceptance documents)? | * Yes |  |  |  |
| * Are the deliverables that are being produced consistent with the deliverable definitions in the Project Management Plan (e.g., size, format, medium)? | * Yes |  |  |  |
| * Are the deliverables being accepted by the customer, after they are submitted? |  |  | * N/A |  |
| * Are Fault Report procedures in place and are the procedures being followed as defined in the knowledge base? |  |  | * N/A |  |
| * **Administrative Support** | | | | |
| * Is the work space adequate? | * Yes |  |  |  |
| * Are the facilities and equipment adequate? | * Yes |  |  |  |
| * Is the general level of office support services adequate (e.g., document reproduction, office supplies, secretarial support)? |  |  | * N/A |  |
| * Is security in place that is appropriate for the security classification of the project (e.g., badges, sign-in registers, electronic lock accessibility)? |  |  | * N/A |  |
| * Is the system development environment adequate (e.g., availability of LANs, access to workstations, stability of the SDE)? | * Yes |  |  |  |
| * **Major Strengths** * Identify up to four positive aspects that represent major strengths of the project. | | | | |
| * 1. Student/Parent can search a qualified Tutor in location. * 2. Tutor can gain good student and money. * 3. Tutor can comment on his student’s performance which can be viewed by student and the parent for his ward. * 4. Administrator and Tutor can upload materials which can be downloaded by other users. * 5. The Tutor has the privilege to delete any student from his batch based on his will. | | | | |
| * **Risks** * Identify any future exposures or major areas of risk to successful project delivery. | | | | |
| * Database gets corrupted. * Website is hacked. * Scripting language is not working properly. * System gets crashed, for example with an infinite loop runs while registering. | | | | |
| * **Opportunities** * Identify the three areas for change that provide the biggest improvement opportunity for similar projects in future. | | | | |
| * Implementation of payment gateway. * Implementation of two-way chatting. * Uploading/Downloading of online materials. | | | | |
| * **Other** * Identify any other points you wish to discuss. | | | | |
| * Nothing. | | | | |

project Understanding Document

*VERSION 1.0*

*30/07/2019*

TABLE OF CONTENTS

[1 introduction 4](#_toc218)

[1.1 Purpose of Project 4](#_toc219)

[2 project And Product Overview 4](#_toc223)

[3 Scope 4](#_toc226)

[3.1 Objectives 4](#_toc227)

[3.2 High-Level Requirements 5](#_toc234)

[3.3 Business Rules and Special considerations 5](#_toc263)

[3.4 Major Deliverables/Milestones 5](#_toc265)

[4 Duration 5](#_toc293)

[4.1 Timeline 5](#_toc294)

[6 project understanding approval 6](#_toc298)

# introduction

Many web portals claim to provide good books for the students and teachers who are looking for a one stop solution for their problems. The different web portals functioning now a days promise their user’s (the registered ones) to provide them with desired books for the subjects for which they are looking for, and lead them to these. The user can buy any book which he/she want by just searching by keywords for their desired book. Also there is a whole library of books from there they can select any or they can select by searching it from the category section.

Also, their exist a way that users as well may give views about how they think about the book. however, these reviews hold a high opinion for other new users who want to join the portal.

**ONLINEBOOKSTORE.COM** is a user friendly web-portal which provides solution to all these above stated problems from providing one-stop solution to provision of online study material through the portal depending upon the availability of the book, to the provision of online course material which is authentic and up to the standards.

## Purpose of Project

As stated above the sole aim of this project is to find users to use this website to purchase book online and get their desired books in just few days on their doorsteps by just clicking.

The main purposes of this project are:-

1. To provide a list of available books to the searching users
2. To provide them with categories of books such as fictions,science,thriller,etc
3. To charge a purchasing fees depending upon the payment to be made for buying a particular book.
4. To provide authentic study materials (whichever the administrator may find) available for download for the users by paying some fees.
5. To promote new books on the store by providing them place in a special category known as ”Latest books” .
6. Providing recent research and development news related to the subject by the administrator of the website.

**Benefits of the Project:**

The system includes every user who is willing to study. Benefits to each of these are described below.

**Benefit to Students:**

With availability of books online the student may find it a flexible mode to study and understand the subject in a better way, as he has guidance of the best author’s for the subject. If in offline mode the student doesn’t have to put in extra effort and may save precious time by finding a book for him/her in his/her city by going to the shop.

**Benefit to Teachers:**

The teachers may find a good source of books through the web-portal which they need to teach their students. Some books may not available on offline bookshop so the teachers can avail this facility too by buying through our portal.

# project And Product Overview

Onlinebookstore.com is a bookstore portal where students and teachers are entertained. Parents can find best & reliable tutors for their wards. Students can find total solution to their problems via study materials that we provide. Teachers can find the best source of study to teach their children.

With a simple registration, users can find good, reliable books with reasonable price, teachers can find good books as well as new study matterials which are promoted by the site administrator in the special category of “Latest books”. We keep updating the users with upcoming books of recent development and research in the field of subjects

The Onlinebookstore.com can be divided into two modules, namely:

**Module 1: Site Administrator**

This module is responsible for maintaining the overall system based on defined business rules and it perform the following functions:

1. It can delete user.
2. It can modify the user details.
3. It can update website details.
4. It can upload online materials.
5. It can add details to:

* Latest books

7. Respond to e-mail request.

8. It can receive payments.

9. It can moderate the comments.

**Module 2: Student**

The web portal as stated earlier assists them in finding them a suitable books and suggesting them their mode of payment whether online/offline as per they wish. In online mode the payment is to be paid my card or netbanking while in offline the payment is to be paid by cash to the delivery executive.

The area of concern is asked from the student at the time of searching for books and based on this information the criterions are matched with the category of books. Once the suitable book is found a buying price is shown and user can buy it by any mode of payment.

# Scope

## Objectives

1. The new user must register with the portal before availing any services from it.
2. For each registration maintain the record in the database for the new user.
3. For each new registration a profile is to be created if it is for the users then it will contain information about the user such as his/her name, email,phone no. and address.
4. When the book is found and selected by the student/parent payment for the price is to be done if user wants to purchase book.
5. The payment could be done in two ways by offline payment (cash on delivery) or through online payment system(card payment).
6. The user may comment on the quality and content of books and if it is appreciable or needs improvement. They can also upload their course materials on the portal which may be used later.
7. Admins may review these comments to know about the progress.

.

## High-Level Requirements

The following table presents the requirements that the project’s product, service or result must meet in order for the project objectives to be satisfied.

| Req. # | I Requirement Description |
| --- | --- |
| Registration | To Provide complete access to the users i.e.to become a valid user. |
| Transaction | Details of valid transaction and card details are retrieved from the  Bank |
| Availability Of book | Books may not be available after all copies will be sold. |
| Network Connectivity | The user should have fast internet connection and should be  supported by an advanced browser. |
| Availability Of Online  Material | Only valid member can access online material. Whenever he  Wants to access he should login along with the transaction id. |
| Searching For books | Only a registered user can search for books. |

## Business Rules and Special considerations

1. Students are eligible to access online books provided for buying.

2. Access will be provides within allotted time period after which payment has to be done again. In

exceptional cases the administrator has the rights to take any actions if necessary.

3. Any anonymous user to avail resources from the portal has to become a registered user.

4. The user should have a credit or a debit card to make online payments.

5. The user should have a valid email-id, postal address to register.

6. Registration can be only through a web-based channel.

7. No transaction is eligible for cash back.

## Major Deliverables/Milestones

[Provide a list of the major deliverables/milestones that will be completed by the end of this project. A deliverable is any unique and verifiable product, result or capability to perform a service that must be produced in order to complete a process, phase or project. A milestone is a key performance indicator that is typically reported to executives to indicate the project’s progress.]

| Major Deliverable | I Deliverable Description |
| --- | --- |
|  |  |
|  |  |
|  |  |

# 19.Duration

## Timeline

[Provide an estimate of the project duration (e.g., 18 months). You may provide a high-level timeline for the project if information is available at this time. This time estimate will be further refined in the Planning Phase of the project. If applicable, also state the expected life of the product. An example of a high-level timeline is provided below.]

System Development

Completed

Developed Prototype

Requirements Analysis

Completed

Project Plan Completed

System Development

Completed

Developed Prototype

Requirements Analysis

Completed

Project Plan Completed

21/07

23/07

27/07

29/07

# project understanding approval

The undersigned acknowledge they have reviewed the **Project Understanding Document** forthe ONLINEBOOKSTORE**.COM** project. Changes to this document will be coordinated with and approved by the undersigned or their designated representatives.

[List the individuals whose signatures are desired. Examples of such individuals are Business Steward, Project Manager or Project Sponsor. Add additional lines for signature as necessary. Although signatures are desired, they are not always required to move forward with the practices outlined within this document.]

|  |  |  |  |
| --- | --- | --- | --- |
| Signature: |  | Date: |  |
| Print Name: | SHAKTI SINGH |  |  |
| Title: |  |  |  |
| Role: |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Signature: |  | Date: |  |
| Print Name: | SHUBHAM KUMAR |  |  |
| Title: |  |  |  |
| Role: |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Signature: |  | Date: |  |
| Print Name: | SHUBHAM GOEL |  |  |
| Title: |  |  |  |
| Role: |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Signature: |  | Date: |  |
| Print Name: | MD. IMTEYAZ |  |  |
| Title: |  |  |  |
| Role: |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Signature: |  | Date: |  |
| Print Name: | AMMAR ARFEEN |  |  |
| Title: |  |  |  |
| Role: |  |  |  |

APPENDIX A: REFERENCES

[Insert the name, version number, description, and physical location of any documents referenced in this document. Add rows to the table as necessary.]

The following table summarizes the documents referenced in this document.

|  |  |  |
| --- | --- | --- |
| **Document Name and Version** | **Description** | **Location** |
| *<Document Name and Version Number>* | *[Provide description of the document]* | *<URL or Network path where document is located>* |

APPENDIX B: KEY TERMS

The following table provides definitions for terms relevant to this document.

|  |  |
| --- | --- |
| **Term** | **Definition** |
| **Credit/Debit Card** | means any credit/debit card issued on the account to the holder and must be a valid card. |
| **VISA/MasterCard** | types of cards issued by the banks which may be used for making payments. |
| **Transaction** | Transfer of monetary value from one bank account to another. |
| **Web Portal** | website bringing together information from diverse sources in a unified way. |
| **Authentic** | valid things, refers to the truthfulness of origins, attributions and commitment. |
| **Discretion** | depends upon the understanding of the user judgment power |
| **Contemporary** | relating to the current world |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |