**Online news portel**

**by**

**A Report Submitted in partial fulfillment of the requirements of the degree of Bachelor in computer application**

**TINSUKIA COLLEGE**

**DEPARTMENT OF BCA**

**Committee Members**

**(Major Professor).**

1

**Acknowledgements**

**I would like to thank my major professor --- who persuaded and continuously guided me during the whole course of my project. I would also like to thank my friends for their assistance and insightful comments, and who willingly shared their expertise with me.**

**I sincerely acknowledge and thank my family members who gave moral support for me from my childhood.**

2

**Abstract**

**This news portel application can lead error free, reliable and better management system.**

**That means one need not distracted by information that is not relevant, while being able to reach the information. The aim behind this project is to automate the existing manual system of media houses by best uses of computer technology.**

3

**Table of Contents**

Section.1 Vision Document

|  |  |  |  |
| --- | --- | --- | --- |
| 1. Introduction | | | 7 |
| 1.1 Purpose and Motivation | | | 7 |
| 2. | Project Overview | | 7 |
| 2.1 Background | | | 7 |
| 3. | Requirement Specifications | | 9 |
| 3.1 Main Requirements | | | 9 |
| 3.2 Critical Use Case Requirements | | | 9 |
| 3.3 Environment | | | 11 |
| Section.2 Project Plan | | |  |
| 1. | Task Breakdown | | 15 |
|  | 1.1. | Inception Phase | 15 |
|  | 1.2. | Elaboration Phase | 15 |
|  | 1.3. | Production Phase | 16 |
| 2. | Cost Estimate | | 16 |
|  | 2.1. | COCOMO | 16 |
| 3. | Architecture Elaboration Plan | | 17 |
|  | 3.1. | Vision Document 1.0 Revision | 17 |
|  | 3.2. | Project Plan 1.0 Revision | 17 |
|  | 3.3. | Formal Requirement Specification | 18 |
|  | 3.4. | Architectural Design | 18 |
|  | 3.5. | Test Plan | 18 |
|  | 3.6. | Formal Technical Inspection | 18 |
|  | 3.7. | Architecture Prototype | 19 |
| Section 3.Architecture Design | | |  |
| 1. Introduction | | | 20 |
| 2. | Architecture | | 21 |
| 3. | Presentation Tier | | 23 |
| 4. | Middle Tier | | 26 |
| 5. | Data Tier | | 29 |
| 6. | References | | 31 |

4

|  |  |  |
| --- | --- | --- |
| Section.4 Formal Requirement Specification | | 32 |
| Section.5 Formal Technical Inspector Checklist | |  |
| 1. | Introduction | 37 |
| 2. | Items to be Inspected | 37 |
| 2.1. UML Diagrams | | 37 |
| 2.2. Formal Specification | | 38 |
| 3. | Formal Technical Inspectors | 38 |
| 4. | Formal Inspection Checklist | 39 |
| 5. | Formal Technical Inspection Report | 39 |

Section.6 Software Quality Assurance Plan

|  |  |  |
| --- | --- | --- |
| 1. | Purpose | 40 |
| 2. | Reference Documents | 40 |
| 3. | Management | 40 |
| 3.1 Organization and Responsibilities | | 40 |
| 3.2 Tasks | | 40 |
| 4. | Documentation | 41 |
| 5. Standards, Practices, Conventions, and Metrics | | 42 |
| 6. | Reviews and Audits | 43 |
| 7. | Test and Problem Reporting | 44 |
| 8. | Tools, Techniques, and Methodologies | 44 |
| 9. | Records collection, Maintenance, and Retention | 44 |

|  |  |  |  |
| --- | --- | --- | --- |
| 10. Deliverables | | | 44 |
| Section.7.Test Plan | | |  |
| 1. | Test plan identifier | | 45 |
| 2. Introduction | | | 45 |
| 3. | Features to be tested | | 45 |
| 5. | Approach | | 48 |
|  | 5.1 | Unit testing | 48 |
|  | 5.2 Load testing | | 49 |
|  | 5.3 | System Testing | 49 |
|  | 5.4 | Performance Testing | 49 |

5

|  |  |  |  |
| --- | --- | --- | --- |
| 6. | Pass/fail criteria | | 49 |
| 7. | Suspension criteria and resumption requirements | | 49 |
|  | 7.1 | Suspension criteria | 49 |
|  | 7.2 | Resumption requirements | 50 |
| 8. | Test deliverables | | 50 |

Section.8 Assessment Evaluation

|  |  |  |
| --- | --- | --- |
| 1. Introduction | | 51 |
| 2. | Manual Testing | 51 |
| 3. | Test Cases | 52 |
| 3.1 User | | 52 |
| 3.2 Admin | | 53 |
| 4. | Performance Testing | 54 |
| 5. | Observation | 55 |
| 6. | References | 56 |
| Section.9 User Manual | | 59 |
| Section.10 References | | 76 |

6

**Section.1 Vision Document**

**1. Introduction**

**1.1 Purpose and Motivation**

The main objective of the project is to create an online news portel that allows users to search the latest news of the day. The latest news are displayed in a tabular format and the user can be able to read the news instantly.

The motivation to create this project has many sources

* Interest to develop a good user friendly news portel with various topics.
* To increase my knowledge horizon in technologies like .NET, SQL, CSS, HTML, PHP.
* To gain good experience in .PHP before joining in a full time job.
* To gain expertise using Data Grid, Data Set, Data Table, Data Adapter and Data Readers.

1. **Project Overview 2.1 Background**

It may help collecting perfect management in detail. In a very short time, the collection will obvious, simple and sensible.it will help a person to know the management of past year perfectly and vividly.It will also reduce the cost of collecting the management and collection procedure will go on smoothly.

7

This project has the following functionalities:

**1) A Home page with product catalog**

It will display all the news categories and will have a search keyword option to search for the required news. It also includes some special sections like recommended national, weekly special news.

**2) Search**

A search by keyword option is provided to the user using a textbox .The keyword to be entered should be the book title.

**3) Administration**

The Administrator will be provided with special functionalities like

* Add or delete a latest news.
* Add or delete a member

8

**3. Requirement Specifications**

All the requirements are specified using OCL a software specification language in the second phase of my presentation.

**3.1 Main Requirements**

The Main Requirements include Microsoft Visual Studio 2005 and ASP.NET to develop the web application, SQL Server 2005 to design the database and Mozilla as a main web browser to run the website.



9

**Use Cases:**

**Browse Catalog**

1. **Search for a news**
   * **Purpose**: A user can search for a news of his choice by selecting category andtitle. Then a select query is used to retrieve data from the database and display the selected information.
   * **Actor:** User
   * **Input:** The user will select a category and enter title in a text box provided.
   * **Output:** The system will display the news which matches the selected searchcriteria. A dataset is created as a result of select query. Later the dataset is binded to the data repeater to display the selected data.

10

11

**Administrator**

1. **Login**
   * **Purpose**: If the Administrator wants to get access to all the functionalities ofOnline news portel he should login using his username and password.
   * **Actor:** Administrator
   * **Input:** The Administrator will enter his username and password.

12

* + **Output:** If it is a successful login the Administrator will be directed to his menupage. Else if the Administrator enters invalid information he will be asked to check the entered information.

1. **Add or Delete Category**
   * **Purpose**: If the Administrator wants to add or delete a news category then he caninsert or delete a news category using his administration rights and the category table will be updated in the database.
   * **Actor:** Administrator
   * **Input:** If the Administrator wants to add a news category the he should click theinsert link button in the category page else he can delete a particular selected news category.
   * **Output:** The updated categories list will be displayed in the main home page.
2. **Add or Delete news**
   * **Purpose**: If the Administrator wants to add or delete a news then he can insert ordelete a news using his administration rights and the book table will be updated in the database.
   * **Actor:** Administrator
   * **Input:** If the Administrator wants to add a news the he should click the insert linkbutton in the news page and fill the following fields related to the news.
     1. Title
     2. place
     3. Category
     4. Notes

If he wants to delete a news he can click the delete button to remove it from the database.

* + **Output:** The updated news list will be displayed in the main home page undertheir particular category.

1. **Manage Orders**
   * **Purpose**: If the Administrator wants to add or delete an order then he can insert ordelete an order using his administration rights.
   * **Actor:** Administrator
   * **Input:** If the Administrator wants to add an order the he should click the insertlink button in the orders page else he can delete a particular selected order **Output:** The updated orders list will be processed to the users.

13

1. **Logout**
   * **Purpose**: If the Administrator wants to end his session and sign out of the websitethen he can use the logout option.
   * **Actor:** Administrator
   * **Input:** The Administrator will click the logout button.
   * **Output:** The Administrator’s account session comes to an end and he shouldlogin again if he wants to enter into the website.

**3.3 Environment**

* The Online news portel will be developed in php environment.
* php will be used as the programming language.

14

**Section.2 Project Plan**

1. **Task Breakdown**
2. **Inception Phase**

The inception phase will define all the project’s requirements. This phase will include the production of a Vision Document, a Project Plan, a Software Quality Assurance Plan, and a Demonstration.

Vision Document will include the project’s requirements and overview. It includes overview of the project, its purpose, goals, risks, constraints, and direction. It gives a listing of the main requirements and their respective Use case models to illustrate the functionality. Project Plan will detail the phases, iterations, and milestones that will comprise the project. It will include a timeline for the project and a cost estimate for completing this project. It includes the Architecture Elaboration plan will define the activities and actions that must be accomplished before the Architecture Presentation.

Software Quality Assurance Plan describes the required documentation, standards and conventions test tracking and problem reporting, and tools used during the project. The plan will also identify the set of quality metrics used to assess product reliability.

Demonstration of at least one executable prototype is required. Projects with a graphical user interface will include an executable prototype of the user interface.

This phase will be complete once the supervisory committee has approved all the above work.

**1.2. Elaboration Phase**

The elaboration phase defines the project’s architecture. This phase will include the production of revisions to the Project Plan and the Vision Document, an Architecture Design Plan, a formal specification, Test Plan, Formal Technical Inspection and Architecture Prototype.

Revision of Vision Document will be an updated version to provide a complete representation of all requirements. These requirements will be ranked according to importance, and a set of critical requirements identified. Appropriate changes that were suggested by the committee at the end of phase one will also be updated in the updated version of Vision Document. Revision of Project Plan will include updated timeline and cost estimate for the project. It also includes The Implementation plan which will define the activities and actions that must be accomplished during implementation.Formal Specification will include OCL.Architecture Design Plan will include the complete architectural design documentation using appropriate diagrams such as class diagrams, sequence/collaboration diagrams, etc. It also includes documentation of reuse of commercial or pre-existing components. Test Plan will address the required tests to show that the product satisfies the requirements. The test plan will identify a set of test cases,

15

the types of tests that will be used for these test cases, the data that will be used for each case, and the requirement traces for each test case. Formal Technical Inspection will include inspections by two MSE students. One of the designs, formal requirement or executable prototype is subjected to inspection. Architecture Prototype will address all critical requirements identified in the vision document.

This phase will be complete once the supervisory committee has approved all the above work.

**1.3. Production Phase**

The production phase defines the project implementation and testing. This phase includes the user manual, component design, assessment evaluation, project evaluation, references, and formal technical inspection.

User Manual includes an overview and explanations of common usage, user commands, error messages, and data formats. Component Design The internal design of each component will be documented using sequence/collaboration diagrams and state chart/activity diagrams.Source Code which corresponds to architecture and component design will be submitted.Assessment Evaluation will include a document detailing the testing done on the project. And Project Evaluation includes evaluation of the project ideas and quality. References and Formal Technical Inspection Letters will also be documented.

This phase will be complete once the supervisory committee has approved all the above work.

**2. Cost Estimate**

The Cost Estimate is done using the COCOMO model**.**

**2.1. COCOMO**

Project effort and time will be estimated using the COCOMO estimation model (Barry Boehm).

The Online Book store has an average complexity and fair flexibility. Therefore, it is classified as an organic mode project under the COCOMO model. The following formula is the COCOMO model for cost estimation for organic mode projects: Effort = 3.2 \* EAF \* (Size) ^ 1.05

Time = 2.5 \* (Effort) ^ 0.38

Where Effort = number of staff months (PM)

EAF = effort adjustment factor

Size = number of lines of code for completed product. It is measured in KLOC (thousands of lines of codes)

Time = total number of months.

16

The Effort Adjustment Factor is the product of the 15 adjustment parameters. Each adjustment parameter is categorized as very low, low, nominal, high, or very high. All the adjustment parameters are listed below:

* RELY Required reliability 0.75 – 1.40
* DATA Database size 0.94 – 1.16
* CPLX Product complexity 0.70 – 1.65
* TIME Execution time constraint 1.00 – 1.66
* STOR Main storage constraint 1.00 – 1.56
* VIRT Virtual machine volatility 0.87 – 1.30
* TURN Computer turnaround time 0.87 – 1.15
* ACAP Analyst capability 1.46 – 0.71
* AEXP Applications experience 1.29 – 0.82
* PCAP Programmer capability 1.42 – 0.70
* VEXP Virtual machine experience 1.21 – 0.90
* LEXP Language experience 1.14 – 0.95
* MODP Use of modern practices 1.24 – 0.82
* TOOL Use of software tools 1.24 – 0.83
* SCED Required development schedule 1.23 – 1.10

Adjustment factors for the Online Book Store are listed below:

* RELY 1.00 Nominal
* DATA 1.00 Nominal
* CPLX 0.85 Low
* TIME 1.00Nominal
* STOR 1.00 Nominal
* VIRT 0.87 Low
* TURN 0.87 Low
* ACAP 1.00 Nominal
* AEXP 1.13 Low
* PCAP 1.00 Nominal
* VEXP 1.00 Nominal
* LEXP 1.00 Nominal
* MODP 0.91 High
* TOOL 0.91 High
* SCED 1.00 Nominal

The EAF value evaluated to 0.60. I have estimated the size to be around 3.00.

From the calculation I got

EFFORT = 6.08

TIME = 4.96

17

**3. Architecture Elaboration Plan**

All the activities to be done before the Architecture Presentation are defined here.

**3.1. Vision Document 1.0 Revision**

Vision Document 1.0 will be revised and updated to Vision Document 2.0. The revisions will contain a complete representation of project requirements and these requirements will be ranked according to importance, and a set of “critical” requirements identified. It also contain changes suggested by the committee members following presentation one. This revision will be approved by the Major Professor.

**3.2. Project Plan 1.0 Revision**

Project Plan 1.0 will be revised and updated to Project Plan 2.0. The revisions will contain Implementation plan. The Implementation plan will define the activities and actions that must be accomplished during implementation. It also contain changes suggested by the committee members following presentation one. This revision will be approved by the major professor.

**3.3. Formal Requirement Specification**

One part of the project will be formally specified using OCL. The specification will represent the formal requirements of the project, described in the Vision Document. This will be approved by the major professor.

**3.4. Architectural Design**

Architectural Design will be documented using UML diagrams such as class, sequence, and state chart diagrams. All architectural components will be documented at interface level. This will be approved by the major professor.

**3.5. Test Plan**

Test Plan will be developed. The document will follow the requirement listings found in Vision Document 2.0. It will also identify a set of test cases, the types of tests that will be used for these test cases, the data that will be used for each case, and the requirement traces for each test case. This will be approved by the major professor.

**3.6. Formal Technical Inspection**

18

Two MSE students will act as technical inspectors. The technical inspection will assess the project architecture. It will also include a formal checklist to be used by the inspectors. This will be approved by the major professor.

**3.7. Architecture Prototype**

An executable prototype will be built including all critical requirements described in the vision document.

19

**Section.3 Architecture Design**

**1. Introduction**

The purpose of this document is to provide an architectural design for the Online news portel. The design will show the presentation tier, the middle tier consisting of classes, sequence diagrams, and the data tier consisting of the database design diagram.

**2. Architecture**

Three-tier (layer) is a client-server architecture in which the user interface, business process (business rules) and data storage and data access are developed and maintained as independent modules or most often on separate platforms.

The Architecture of Online news portel is based on three-tier architecture. The three logical tiers are

* Presentation tier - ASP.NET Web forms, Master Pages, Images.
* Middle tier – C# classes.
* Data tier- Database

Fig.1 below shows the model of 3-tier architecture.

The main reason for considering three-tier architecture for the Online Book store is as follows:

***Flexibility:***

* Management of data is independent from the physical storage support,
* Maintenance of the business logic is easier,
* Migration to new graphical environments is faster.
* If there is a minor change in the business logic, we don’t have to install the entire system in individual user’s PCs.

**Reusability:**

* Reusability of business logic is greater for the presentation layer. As this component is developed and tested, we can use it in any other project and would be helpful for future use.

20

**Security:**

* More secured architecture since the client cannot access the database directly.

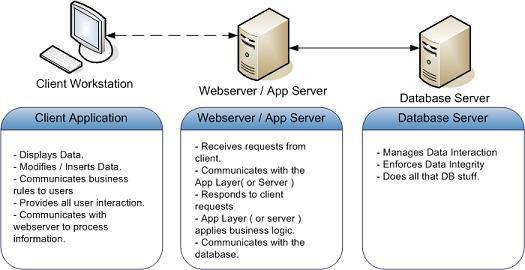


Fig.1 3-tier Architecture

21

22

**Fig.2 User-Page Flow**

23

24

Fig.3 Administrator-Page Flow

25

**4**. **Middle Tier**

**Class Diagram**

Fig.4 Class Diagram

26

**Sequence Diagrams**

**1) User Login**

Fig.5 User-Login Sequence Diagram

27

**2) Book Search**

28

Fig.8 User-Add to Cart Sequence Diagram

29

**4) Administrator**

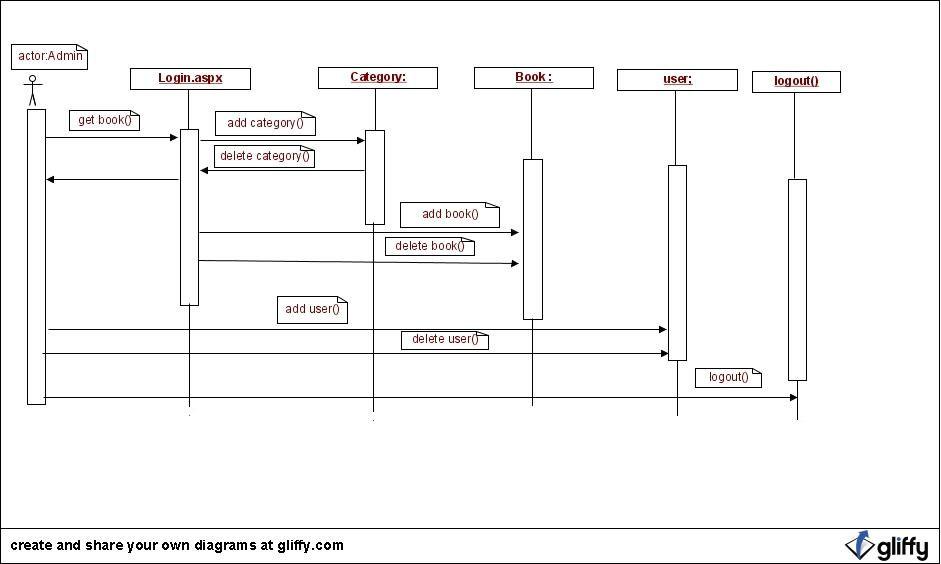
****

Fig.9 Adminstrator- Sequence Diagram

30

**4. Data Tier**

The system database has five tables Categories, Items, Orders, Members and Card types. The system database design is shown below.

|  |  |
| --- | --- |
| **Table Name** | **Definition** |
|  |  |
| Categories | Contains the Book Categories Information |
| Items | Contains the Book Information. |
| Members | Contains the Members Information |
| Orders | Contains the Book Orders Information |
| Card Types | Contains the Credit Card Information |

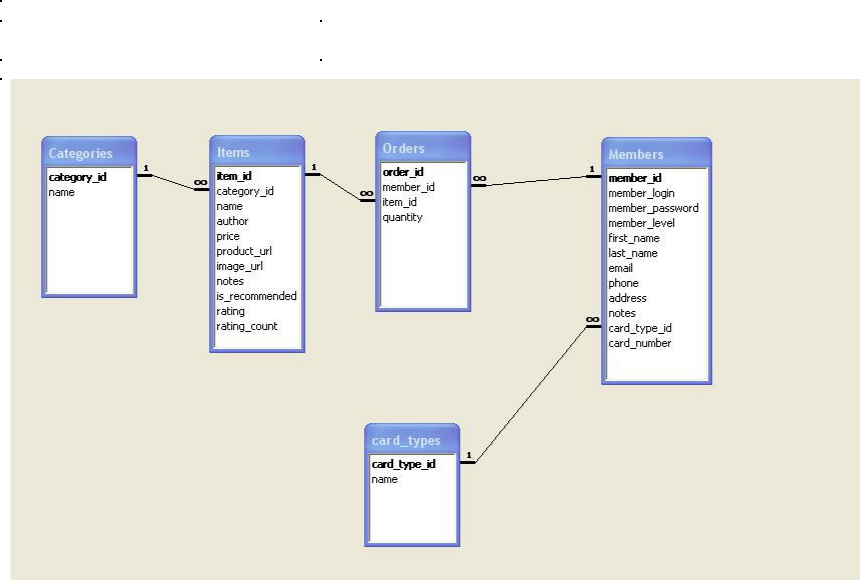


Fig.10 System Database Design

31

**Section.4 Formal Requirement Specification**

model OnlineBookstore

-- Classes

class User

attributes

UserID: string

password: string

LoginStatus: string

operations

Verifylogin (): Boolean

end

class Administrator < User

attributes

AdminID: string

password: string

Name: string

email: string

phoneNo: integer

operations

addCategory(): Boolean

deleteCategory:Boolean

addMember():Boolean

deleteMember():Boolean

addBook():Boolean

deleteBook():Boolean

addCCtype():Boolean

deleteCCtype():Boolean

end

class Customer < User

attributes

customerID:string

password:string

Name:string

address:string

email:string

phoneno:integer

CCInfo: string

operations

register(): Boolean

login(): Boolean

--updateprofile is used to update user information updateProfile(customerID: string, Name:string, address:string, email:string, phoneno: integer, CCInfo: string):Boolean

32

--Customer already has an customerID

pre: Customer.allInstances.customerID->includes (customerID)

post: Customer.allInstances.customerID = user.allInstances.customerID@pre --Customer name has been created

post: Customer.allInstances.Name =user.allInstances->select(C:Customer | C.customerID

<>customerID).Name@pre->includes (Name)

--Customer address has been created

post: Customer.allInstances.email= Customer.allInstances->select(C:Customer

* C.customerID<>customerID).email@pre->includes(email) post: Customer.allInstances.address = Customer.allInstances->select(C:Customer | C.customerID<>customerID).address@pre->includes(address) --Customer Phoneno has been created

post: Customer.allInstances.phoneno = Customer.allInstances->select(C:Customer | C.customerID<>customerID).phoneno@pre->includes(phoneno) post: Customer.allInstances.CCInfo = Customer.allInstances->select(C:Customer | C.customerID<>customerID). CCInfo @pre->includes(CCInfo)

end

class Category

attributes

categoryID:integer

categoryName:string

operations

getCategoryBooks(bookID:int):Set(Book) =

Book.allInstances->select(b:Book| b. bookID = bookID)

end

class Book

attributes

bookID:integer

categoryID:integer

bookName:string

authorName:string

notes:string

price:float

imageurl:string

producturl:string

rating:int

operations

getBook():Boolean

end

class ShoppingCart

attributes

orderID:integer

customerID:integer

price:float

33

operations

addCart():Boolean

deleteCart():Boolean

updateCart():Boolean

end

class BookOrder

attributes

orderID:integer

customerID:integer

price:float

quantity:integer

operations

placeOrder(BO:BookOrder):Boolean

--User authentication is verified

pre: BO.User. Verifylogin (BO.User. UserID, BO.User.password)=true

pre: BookOrder.allInstances->excludes(BO)

--Check whether customerid and customerid in Order is same

post: Customer.allInstances -> forAll(C:Customer | C.customerID=BO.customerID implies BO.orderID = C.orderID)

post: BookOrder.allInstances.orderID = BookOrder.allInstances.orderID@pre->includes(BO.orderID) end

class Search

attributes

bookTitle:String

categoryID:integer

operations

getBookset():Boolean

end

class AdvSearch

attributes

bookTitle:String

categoryID:integer

bookAuthor:String

bookLowCost:float

bookHighCost:float

operations

getBooksetbyAdv():Boolean

end

class BookSet

attributes

bookID:Int

bookName:String

end

-- Assosiations

--Each book should belong to only one Category

34

association bookCategory between

Category[1] role subCategory

Book[1..\*] role allBook

end

--Each BooksOrder should contain atleast one Book

association BooksOrderHasBook between

BooksOrder[1] role theBookOrder

Book[1..\*] role theBook

end

--Each BooksOrder should belong to exactly one Customer association CustomerHasOrder between Customer[1] role belongstocustomer

BooksOrder[0..\*] role thecustomerbook

end

--Each Shopping cart should belong to only one Customer association CustomerrelatedtoShoppingCart between Customer[1] role thecustomer ShoppingCart[0..\*] role thecart

end

--Each shoppingcart should have atleast one BooksOrder association ShoppingCartHasOrder between ShoppingCart[1] role thebookCart BooksOrder [1..\*] role theorder

end

--Each search should result some bookset

association searchhassomebookset between

Search[1] role thesearch

BookSet[0..\*] role thesearchset

end

--Each Advsearch should result some bookset

association Advsearchhassomebookset between

AdvSearch[1] role theAdvsearch

BookSet[0..\*] role theAdvsearchset

end

--Constraints

* Each user should have different userID context User

inv distinctuserID:

User.allInstances -> forAll(user1, user2 |user1 <> user2 implies user1.userID <> user2.userID)

35

* Each Book should belong to exactly one Category context Category

inv BookHasoneCategory:

Category.allInstances -> forAll(C1,C2 |C1<>C2 && C1. getCategoryBooks -> includes(book) implies C2. getCategoryBooks -> excludes(book))

--Each Book should have a different bookID

context Book

inv DistinctBookID:

Book.allInstances -> forAll(B1, B2 |B1 <> B2 implies B1.bookID<>

B2.bookID)

--The OrderID for each Order must be different

context BookOrder

inv DistinctOrderID:

BookOrder.allInstances -> forAll(BO1, BO2 |BO1 <> BO2 implies BO1.orderID<>

BO2.orderID)

--Each BookOrder should have some books

context BookOrder

inv BookOrderHasbooks

self.contains -> notEmpty()

* Each BookOrder belongs to exactly one customer context BookOrder

inv OrdertoOneCustomer

Order.allInstances -> forAll (BO1, BO2 | BO1.orderID <> BO2.orderID implies BO1.customerID <> BO2.customerID)

* Quantity should always be a positive value

context BookOrder

inv BookOrder Positive

self.quantity > 0

--Each Shopping cart belongs to only one customer

context ShoppingCart

inv CarthasOneCustomer

ShoppingCart.allInstances -> forAll (SC1, SC2 | SC1.orderID <> SC2.orderID

implies SC1.customerID <> SC2.customerID)

--In search Low price should be less than High Price

context AdvSearch

inv PriceCompare

self. bookLowCost < self. bookHighCost

* Price should always be a positive value context Book

inv BookPricePositive: self.price > 0

36

**Section.5 Formal Technical Inspectors Checklist**

**1. Introduction**

The document specifies checklist to be inspected by the technical inspectors of

Online news portel. The main purpose of the formal technical inspection process is to ensure quality of the software design. Two MSE students will perform the inspection and provide the formal report on the result of their inspection.

**2. Items to be inspected**

**2.1. UML Diagrams**

* Class diagrams
* Sequence diagrams

**2.2. Formal Specification**

* + USE Model

1. **Formal Technical Inspectors**
   * Phaninder Surapaneni
   * Snehal Monterio
2. **Formal Inspection Checklist**

|  |  |
| --- | --- |
| **Item to be Inspected** | **Pass/Fail/ Comments** |
|  | **Partial** |

All the symbols used in the class diagrams are according to the UML standards.

All the classes in the class diagram are clear as to what they represent in the architecture design document.

The symbols used in the sequence diagram correspond to UML standards.

Sequence diagram matches class diagram.

All the classes in the USE model are represented in the class diagram.

The multiplicities in the USE model have been depicted in the class diagram.

All the requirements in the Software Requirements

Specification have been covered in the

37

Architecture Design Document.

38

39

**Section.6 Software Quality Assurance Plan**

**1. Purpose**

The purpose of Software Quality Assurance Plan is to define all the techniques, procedures, and methodologies that will be used in the project to assure timely delivery of the software that meets specified requirements within project resources. Software Quality Assurance involves reviewing and auditing the software products and activities to verify that they comply with the applicable procedures and standards and providing the software project and other appropriate managers with the results of these reviews and audits.

1. **Reference Documents**
   * Project Plan 1.0
   * Vision Document 1.0
   * IEEE standard for Software Quality Assurance plans, IEEE STD 730-1998.
2. **Management**

**3.1 Organization and Responsibilities**

The organization consists of supervisory committee, major professor, developer and formal technical inspectors.

40

The committee will be responsible for attending the presentations and reviews at the end

of each phase. After each presentation, the committee will provide feedback and

suggestions regarding the project.

**Developer**

* Birinchi

The developer should submit all the deliverables and complete the project functionalities

on time. He should keep updating his weekly progress to the Major Professor.

**Formal Technical Inspectors**

The formal technical inspectors will be responsible for a technical review of the

architecture design artifacts and the formal requirements specifications and will also be

required to submit a formal report based on their findings.

Formal Technical Inspection Checklist

|  |  |
| --- | --- |
| **Item to be Inspected** | **Pass/Fail/ Comments** |
|  | **Partial** |

All the symbols used in the class diagrams are according to the UML standards.

All the classes in the class diagram are clear as to what they represent in the architecture design document.

The symbols used in the sequence diagram correspond to UML standards.

Sequence diagram matches class diagram.

All the classes in the USE model are represented in the class diagram.

The multiplicities in the USE model have been

41

depicted in the class diagram.

All the requirements in the Software Requirements

Specification have been covered in the

Architecture Design Document.

**3. 2 Tasks**

All tasks to be performed are mentioned in the Project Plan1.0.After the first phase if any changes are required the Major Professor will discuss with the developer.

**4. Documentation**

The documentation will consist of all the deliverables. They are vision document, project plan, software quality assurance plan, formal requirements specification, architecture design, test plan, formal technical inspection, prototype, user manual, component design, source code, assessment evaluation, project evaluation, references, and formal technical inspection letters. The committee members will review all documentation for final approval.

**5**. **Standards, Practices, Conventions, and Metrics**

* **Documentation Standard**s

The IEEE standards are used as reference for all the documents of Online Book Store project. IEEE Standard for Software Quality Assurance Planning is used for SQA 1.0.

* **Coding Standards**

The project coding standards follow C# .As the project is developed using the C# language.

* **Testing Standards**

The various types of testing performed would be mentioned in the Test Plan document at the end of second phase.

42

**6. Reviews and Audits**

All the deliverables produced in each phase of project development are reviewed by the major professor and supervising committee. These are evaluated by the committee, at the end of each phase and provide comments on the software prototype as well as suggestions for any changes or addition to the requirements specification. The two formal technical inspectors will assess the architecture design artifacts and submit a formal report based on their findings.

**7. Test and Problem Reporting**

All the testing procedures used for the project would be mentioned in the test plan document at the end of second phase. The results would be reviewed and all the unresolved problems will be reported to the committee members.

**8. Tools, Techniques and Methodologies**

The following tools, tecnniques and methods would be used for the project for the

specified purpose:

* + **Coding: -** C#, CSS, HTML, ASP.Net, Microsoft VisualStudio.Net 2005 IDE,Microsoft SQL Server 2005
  + **Testing:-**Junit
  + **Documentation:-**MS Word, Rational Rose, MS Project, OCL.

1. **Records collection, Maintenance, and Retention**

Three sets of design documentation would be produced and distributed to the University Library, Major Professor and developer. The source code, documentation and web pages are submitted to the Major Professor in the form of a CD.

**10. Deliverables**

The deliverables for all the three phases are listed below:

**Phase I**

* Vision Document 1.0
* Project Plan 1.0

43

• Software Quality Assurance Plan

**Phase II**

* Action Items –identified during phase I
* Vision Document 2.0
* Project Plan 2.0
* Formal Requirements Specification
* Architecture Design
* Test Plan
* Formal Technical Inspection
* Executable Architecture Prototype

**Phase III**

* Action Items - identified during phase I
* User Manual
* Component Design
* Source Code
* Assessment Evaluation
* Project Evaluation
* References
* Formal Technical Inspection

44

**Section.7 Test Plan**

**1. Test plan identifier**

CIS 895-MSE Project Test plan Online Book Store V1.0

**2. Introduction**

The goal of this document is to develop a test plan for the Online news portel design system. This document defines all the procedures and activities required to prepare for testing of the functionalities of the system which are specified in Vision document. The objectives of the test plan are to define the activities to perform testing, define the test deliverables documents and to identify the various risks and contingencies involved in testing.

**3. Features to be tested**

The following list describes the features to be tested:

**ADMIN:**

* Create and Delete news from Category
* Create and Delete a Category
* Manage news
* Manage Members

45

1. **Test Cases**

46

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

**4.2 ADMIN**

**Create and Delete a news from Category**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ID | TEST CASE | | |  |  | ADMIN INPUT | | | PASS CRITERIA | |  |
| AD\_CDB\_1 | Create | and | Delete | a | Admin adds a new news to | | | | news | should | be |
|  | news from Category | | |  | category | |  |  | updated | in Categories | |
|  |  |  |  |  |  |  |  |  | list |  |  |
| AD\_CDB\_2 | Create | and | Delete | a | Admin | deletes | a |  | news should be deleted | | |
|  | news from Category | | |  | from category | |  |  | in Categories list | |  |
| **Create and Delete a Category** | | | |  |  |  |  |  |  |  |  |
|  |  | | |  |  |  | | |  | |  |
| ID | TEST CASE | | |  |  | ADMIN INPUT | | | PASS CRITERIA | |  |
| AD\_CDC\_1 | Create | and | Delete | a | Admin | adds | a | new | Category | should | be |
|  | Category | |  |  | category | |  |  | updated to system | |  |
|  |  |  |  |  |  | | | |  |  |  |
| AD\_CDC\_1 | Create | and | Delete | a | Admin deletes a category | | | | Category | should | be |
|  | Category | |  |  |  |  |  |  | deleted from system | |  |
|  |  |  |  |  |  |  |  |  |  |  |  |

47

**Manage Members**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ID | TEST CASE |  | ADMIN INPUT | PASS CRITERIA | |  |
| AD\_MM\_1 | Manage Members | Admin accepts Members | | Member is accepted | |  |
|  |  |  | |  |  |  |
| AD\_MM\_2 | Manage Members | Admin deletes Members | | Member | is | not |
|  |  |  |  | accepted |  |  |

**5. Approach**

This section describes the overall approach of the testing which ensures that the each feature and the combination of the features are adequately tested. The major tasks that are used are

**5.1** **Unit testing**

Unit testing is a method of testing that verifies the individual units of source code are working properly. The goal of unit testing is to isolate each part of the program and show that the individual parts are correct. The NUnit a testing tool for C# will be used for unit testing.

48

**5.2 Load testing**

Load testing is the process of creating demand on a system or device and measuring its response. It generally refers to the practice of modeling the expected usage of a software program by simulating multiple users accessing the program concurrently. As such, this testing is most relevant for multi-user systems; often one built using a client/server model, such as web servers

**5.3 System Testing**

Once the entire system has been built then it has to be tested against the Software Requirement Specification and System Specification to check if it delivers the features required. System testing can involve a number of specialist types of test to see if all the functional and non-functional requirements have been met.

**5.4 Performance Testing**

The system should meet the performance requirements as mentioned in the Vision document. The performance will be evaluated based on the response time of the GUI and the database commands. Using JMETER tool performance testing will be done.

**5.5 Manual Testing**

Manual Testing will be done to ensure the correctness of various parts of the code using test cases generated by the tester.

**6. Pass/fail criteria**

The system should satisfy all the functional requirements, in the Vision document. Each feature to be tested will be evaluated against its requirement as stated in the Vision Document. The pass or fail of a test depends on whether the system meets with all the particular post conditions.

Test cases executed on the Online Book Store will pass if they meet the specific requirements as mentioned in the Vision Document.

1. **Suspension criteria and resumption requirements 7.1 Suspension criteria**

49

If the system contains one or more critical defects like the defects in the GUI editor which provides the editing features for one line diagrams and database locking, unlocking and sharing features which provides the environment for multiple users to work in parallel, the entire system should be suspended.

The testing may also be suspended if the hardware and software components required are not available on time.

The failed test cases should be recorded along with the description for failure.

**7.2 Resumption requirements**

When a new version of the system is transmitted to the test group after a suspension of testing has occurred, all previous tests will be rerun to ensure program changes have not inadvertently affected other portions of the program.

**8. Test deliverables**

The following documents are the available test deliverables:-

* Test plan
* Test case specifications
* Test input and output data
* Test procedure specifications
* Test logs

50

**Section.8 Assessment Evaluation**

**1. Introduction**

The goal of this document is to present all the test results using the test cases defined in the Test Plan Document. I have performed Manual and Performance testing for my Online Book store project.

**2. Manual Testing**

Manual testing is done to test the correctness of all the functionalities by manually entering the data.

Test case functionalities for manual testing include:

**ADMIN:**

* + Create and Delete book from Category
  + Create and Delete a Category
  + Manage Orders
  + Manage Members

1. **Test Cases**

52

**3.2 ADMIN**

**Create and Delete a news from Category**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ID | TEST CASE | | |  |  | ADMIN INPUT | RESULT | PASS CRITERIA |
| AD\_CDB\_1 | Create | and | Delete | a | Admin adds a new news to | | PASSED | news should be updated |
|  | news from Category | | |  | category | |  | in Categories list |
| AD\_CDB\_2 | Create | and | Delete | a | Admin deletesa news | | PASSED | news should be deleted |
|  | news from Category | | |  | from category | |  | in Categories list |



|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Create and Delete a Category** | | | |  |  |  |  |
|  |  |  |  |  |  | |  |
| ID | TEST CASE |  | ADMIN INPUT | RESULTS | PASS CRITERIA | |  |
| AD\_CDC\_1 | Create and Delete | Admin adds a new | | PASSED | Category | should | be |
|  | a Category | category | |  | updated to system | |  |
|  |  |  | |  |  |  |  |
| AD\_CDC\_1 | Create and Delete | Admindeletesa | | PASSED | Category | should | be |
|  | a Category | category | |  | deleted from system | |  |
|  |  |  |  |  |  |  |  |

53

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Manage Members** |  |  |  |  |
|  |  |  |  |  |  |
| ID | TEST CASE |  | ADMIN INPUT | RESULTS | PASS CRITERIA |
| AD\_MM\_1 | Manage Members | Admin accepts Members | | PASSED | Member is accepted |
|  |  |  | |  |  |
| AD\_MM\_2 | Manage Members | Admin deletes Members | | PASSED | Member is not accepted |
|  |  |  |  |  |  |

**4. Performance Testing**

The system should meet the performance requirements as mentioned in the Vision document. The performance will be evaluated based on the response time of the GUI and the database commands. Using JMETER tool performance testing will be done.

Apache JMETER is a 100% pure Java desktop application designed to load test functional behavior and measure performance*.* Originally designed to test web applications, it is now a general tool for testing client/server applications like database servers, FTP servers and others. It lets you test your applications under different work loads, and graphically represent the application's behavior.

I have tested the performance of three different pages of my Online Book Store Website as mentioned in the Test Plan document during Phase-2.

Pages Tested

* Home Page
* Search Page

The Performance testing has been done using a sample of 8 threads which are generated with a loop of 100.

54

**Home Page**

**http://localhost:3062/news/Default.aspx**

The graph below in Fig.1 shows the test results of Homepage of Online news portel Website.

Performance Graph



Fig.1

55

**Search Page**

**http://localhost/news/**

The graph results below in Fig.2 shows the test results of Search Page of Online news portel Website.

Performance Graph

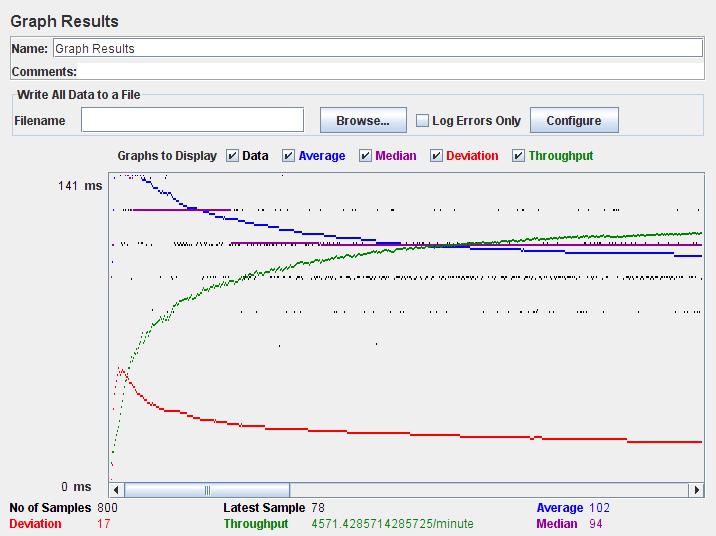


Fig.2

56

Fig.3

57

1. **Observations**
   * Average response time is more for search page when compared to the Home page.
   * Average response time is low for home page because it doesn’t many database interactions.
   * Average response time for Search page is more since it should wait for the results from the database.

58

70

**Section.10 Project Evaluation**

**1. Introduction**

This document evaluates the experience of the development of the Online news portel project. A brief description of the tools, process, techniques employed as well as the mistakes made is presented so that lessons are documented and learned.

**2. Problems faced**

The following are the problems faced during the design of Online news portel Website.

**2.1 C# language**

The main problem encountered during the process was my inexperience in the ASP.NET and C# Language and environment.

So I used the videos in asp.net website to get basic experience in C#.NET.I tried to solve some sample applications which helped me to design the website easily. It was difficult for me to format the home page of my website using the C#.NET code. So finally I choose to use the html div tags which made my work easier. I also didn’t have good experience of using the J-Meter testing tool for performance testing. So I learnt about it using some tutorials which I found using google.com.

**3. Metrics**

**3.1 Lines of Code**

My initial estimate was 3000 LOC during Phase-1.This was calculated in the COCOMO model in Project Plan 1.0 document.

But now I found the total to be 3203 LOC after the coding part is completed. I used the LOC metrics tool to count the total number of Lines of Code.

71

The reason for increase in the size of the project may be due to my inexperience in programming in C# and using extra functions for functionality. And it was difficult for me to estimate the exact number of lines of code during the initial phase of my Project.

The Table below shows the break down of time spent in each phase for Research, Design, Coding, Testing and Documentation.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Inception(Hours)** | **Implementation(Hours)** | **Elaboration(Hours)** | **Total** |
|  |  |  |  |  |
| **Research** | 20 | 25 | 5 | 50 |
|  |  |  |  |  |
| **Design** | 17 | 20 | 5 | 42 |
|  |  |  |  |  |
| **Coding** | 0 | 210 | 37 | 247 |
|  |  |  |  |  |
| **Testing** | 0 | 15 | 30 | 45 |
|  |  |  |  |  |
| **Documentation** | 20 | 28 | 20 | 68 |
|  |  |  |  |  |
| **Total** | 57 | 298 | 97 | 452 |
|  |  |  |  |  |

Table.2 Phase Breakdown

72

The Pie chart in Fig.1 shows the hour break down for each phase.

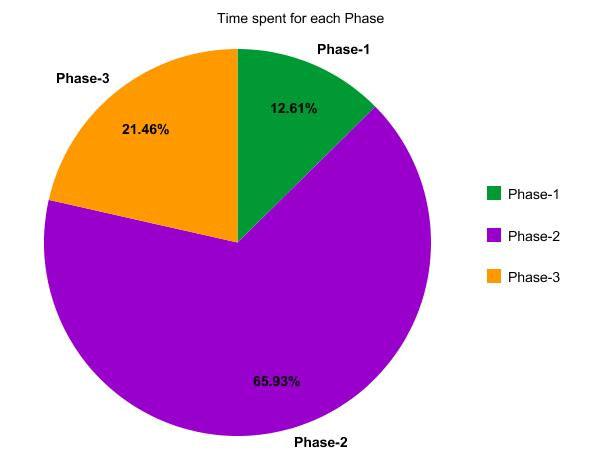


Fig.1 Project Phase Schedule

73

The pie chart in Fig.2 shows the Time allocated for various tasks during the Project Implementation.

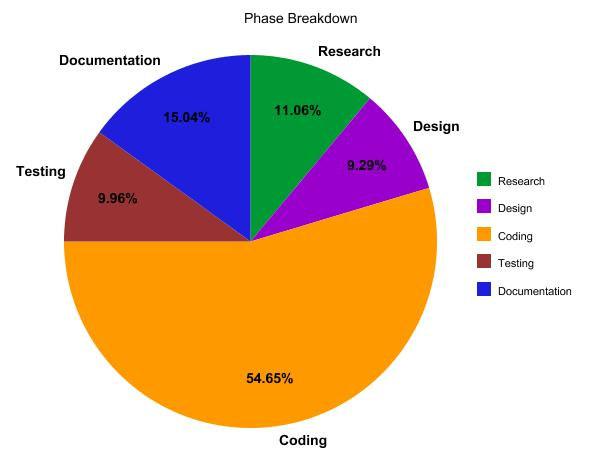


Fig.2 Task breakdown

1. **Lessons Learned Programming**

The Online Book Store Project helped me to improve my confidence level in C# Programming. Though I have made many mistakes during the initial phase I have learnt how to use user controls, master pages, data grid, data set and other data base functionalities.

**Time Management**

Since MSE Project is done as an individual I have learnt how to manage time during the Software Life Cycle Process. I have also learned how o face tense situations and meet the deadlines .This would add as a good experience for me for my future job prospective**.**

74

**UML and Software Lifecycle**

As software student though I have good knowledge in UML and Software LIFE cycle I never had any good practical experience regarding them. Through this project I have learnt how to develop a project following the various stages in Software Life Cycle.

**Documentation**

I always had a feeling that I am not good at documentation .But through this project and suggestions from my committee members I believe that I have improved my Documentation skills.

75

76