**Assignment 4-ITC303**

Book.trade Version 1.0

**Project Management Assessment Stream (PMAS)**

**Project Manager/ Developer**

Noman Ahmed Khan - 11567969

**Project and Business Analyst**

Sandesh Raja Thapa - 11575549

**Developer/ Debugger**

Amankumar C. Parmar - 11569794

Table of Contents

Executive Summary 4

Work Breakdown Structure 5

Team Work Breakdown Structure Report for LCOM 6

Team Work Breakdown Structure Report for LCAM 7

Individual Work Breakdown Structure Report for Overall Project: 8

[Revised Vision 11](#_Toc484400301)

[Introduction 11](#_Toc484400302)

[Positioning 11](#_Toc484400303)

[Stakeholder Descriptions 14](#_Toc484400304)

[Product Overview 16](#_Toc484400305)

[Other Product Requirements 17](#_Toc484400306)

[Requirements Model 18](#_Toc484400307)

[Use Case Diagram: 18](#_Toc484400308)

[Use Case and Brief Description 20](#_Toc484400309)

[Full Use Case Descriptions of CCRD Use-Cases: 21](#_Toc484400310)

[Use-Case: Register 22](#_Toc484400312)

[Use-Case: Login 23](#_Toc484400313)

[Use-Case: Create a Book List 24](#_Toc484400314)

[Use-Case: Managing Book List 25](#_Toc484400315)

[Use-Case: Buying/Selling Books 26](#_Toc484400316)

[Use-Case: Trade Books 27](#_Toc484400317)

[Use-Case: Payment 28](#_Toc484400318)

[Activity Diagrams for the CCRD Use Case: 29](#_Toc484400319)

[Domain Model Diagram](#_Toc484400320) [33](#_Toc484400321)

[Revised Requirements Traceability Matrix 34](#_Toc484400322)

[Revised Non-Functional Requirements 36](#_Toc484400323)

[Introduction 36](#_Toc484400324)

[System wide Functional Requirements 36](#_Toc484400325)

[System Qualities 36](#_Toc484400326)

[System Interfaces 38](#_Toc484400327)

[Business Rules 39](#_Toc484400328)

[System Constraints 39](#_Toc484400329)

[System Compliance 39](#_Toc484400330)

[System Documentation 40](#_Toc484400331)

[Finalized Architecture Notebook 41](#_Toc484400332)

[Purpose 41](#_Toc484400333)

[Architectural goals and philosophy 41](#_Toc484400334)

[Assumptions and dependencies 41](#_Toc484400335)

[Architecturally significant requirements 42](#_Toc484400336)

[Decisions, constraints, and justifications 42](#_Toc484400337)

[Architectural Framework and Design: 43](#_Toc484400338)[43](#_Toc484400339)

[Executable Architecture: 44](#_Toc484400340)

[Technical Competency Demonstrator 50](#_Toc484400341)

[*Three Tier Architecture in Work 51*](#_Toc484400342)

[Revised Risk Register 56](#_Toc484400343)

UAT [Testing 68](#_Toc484400344)

[*Use Case : Register 69*](#_Toc484400345)

[*Use Case : Login 71*](#_Toc484400346)

[*Use Case : Add Books 72*](#_Toc484400347)

[*Use Case : Managing Books 74*](#_Toc484400348)

[*Use Case : Buy Books 75*](#_Toc484400349)

[*Use Case : Trading Books 76*](#_Toc484400350)

[*Use Case : Selling Books 78*](#_Toc484400351)

[*Use Case : Payment through online service 79*](#_Toc484400352)

[*Use Case : Payment through card 80*](#_Toc484400353)

[CCRD USE CASE TESTING 81](#_Toc484400354)

[List of Processes that are Core, Critical, Risky & Difficult for the chosen CCRD use cases 82](#_Toc484400355)

[Security 83](#_Toc484400356)

[Data Consistency 83](#_Toc484400357)

[Connectivity 83](#_Toc484400358)

[Script Execution 83](#_Toc484400359)

[Secure Payment Connection: 84](#_Toc484400360)

[Accurate and Consistent Billing System 84](#_Toc484400361)

[Payment Rollback 84](#_Toc484400362)

[Accurate Transaction Report 84](#_Toc484400363)

[Accessibility 85](#_Toc484400364)

[Usability 85](#_Toc484400365)

[System and Server Sync 85](#_Toc484400366)

[UAT Feedback Plan 86](#_Toc484400367)

[User on Desktop or Laptop Device 87](#_Toc484400368)

[User TO-DO 87](#_Toc484400369)

[User Feedback 87](#_Toc484400370)

[Revised Project Plan 88](#_Toc484400371)

[Introduction 88](#_Toc484400372)

[Project organization 88](#_Toc484400373)

[Project practices and measurements 88](#_Toc484400374)

[Deployment 88](#_Toc484400375)

[Project milestones and objectives 89](#_Toc484400376)

[Elaboration Phase Assessment 91](#_Toc484400377)

Executive Summary

The purpose of this document is to present the Work Breakdown Structure WBS for the overall work that has been done in this semester for Assessing Individual and Team Efforts. For this reason we have presented a few reports, which include WBS report of the team for LCOM, WBS report of the team for LCAM which highlights the assigned tasks to each team member and a WBS report for each individual, which highlights the task that they have done, brief details of what they tried to achieve with each task and what major milestone was being achieved with it. Lastly, it highlights whether the task was delivered on time and with what report it was delivered.

Below the WBS Reports is the last and final version of the LCAM report-Which also has LCOM integrated, which can be referred to in order to look at the work presented in this semester.

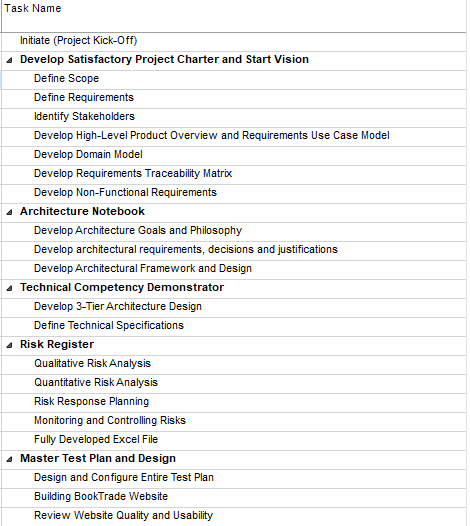
Work Breakdown Structure

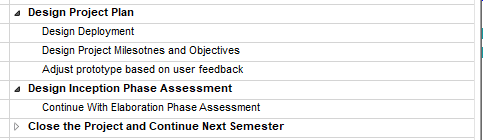
To make it easier for the lecturer to Assess individual and team efforts of each team member working on the project, the following work breakdown structures have been created:

* Team Work Breakdown Structure Report for LCOM: This explains the tasks that were given to the different team members as individuals as well as the breakdown of any task that was done together by the team in LCOM report.
* Team Work Breakdown Structure Report for LCAM: This explains the tasks that were given to the different team members for the LCAM report.
* Individual Work Breakdown Structure Report: This work breakdown structure is for each individual, explaining in details the tasks that were provided to them, what major milestone was being achieved with that task and whether it was delivered or not, and with what report it was delivered.

\*\* It is to be noted that our team has also uploaded a PowerPoint Presentation to turnitin, and a presentation was given within class in which each team member had to explain what work they have done throughout this semester.

The following WBS is just a list of tasks that were laid from the beginning of LCOM report till the end of LCAM report:





**Team Work Breakdown Structure Report for LCOM:**

|  |  |  |
| --- | --- | --- |
| **Task** | **Breakdown** | **Done by** |
| Executive Summary | - | Noman Ahmed Khan |
| Vision | - | Aman Parmar, Sandesh Raja Thapa, Noman Ahmed Khan |
| Requirements Model | 1a. Use Case + Description  2. Domain Model Diagram   1. Non-Functional Requirements | 1a. Noman Ahmed Khan  1b. Noman Ahmed Khan   1. Aman Parmar 2. Sandesh Raja Thapa |
| RTM | - | Sandesh Raja Thapa |
| Architecture Notebook | - | Noman Ahmed Khan |
| Technical Competency Demonstrator | 1. Theoretical Explanation of Teams Skills and Applications/ Languages to be used. 2. Setting up Integrated Application, Version Control Repository of all related documents in GitHub. | 1. Noman Ahmed Khan 2. Aman Parmar |
| Risk Register | - | Sandesh Raja Thapa |
| Master Test Plan | - | Aman Parmar |
| Project Plan | 1. Explanation of Different UP Stages. 2. Project Milestones and Objectives. | 1. Sandesh Raja Thapa 2. Noman Ahmed Khan |
| Inception Phase Assessment | - | Sandesh Raja Thapa. |

**Team Work Breakdown Structure Report for LCAM:**

|  |  |  |
| --- | --- | --- |
| **Task** | **Breakdown** | **Done by** |
| Executive Summary | - | Noman Ahmed Khan |
| Vision  Revised Vision | - | Aman Parmar, Sandesh Raja Thapa, Noman Ahmed Khan |
| Requirements Model | 1a. Use Case + Description  1b. Full Use Case Description + Activity Diagrams for CCRD Use Cases.  2. Domain Model Diagram   1. Non-Functional Requirements | 1a. Noman Ahmed Khan  1b. Noman Ahmed Khan   1. Aman Parmar 2. Sandesh Raja Thapa |
| RTM | - | Sandesh Raja Thapa |
| Architecture Notebook/ Final Architecture | - | Noman Ahmed Khan |
| Executable Architecture | - | Noman Ahmed Khan |
| Technical Competency Demonstrator | 1. Theoretical Explanation of Teams Skills and Applications/ Languages to be used. 2. Setting up Integrated Application, Version Control Repository of all related documents in GitHub. | 1. Noman Ahmed Khan 2. Aman Parmar |
| Risk Register  Revised Risk Register | - | Sandesh Raja Thapa |
| Master Test Plan  Revised Master Test Plan | - | Aman Parmar |
| User Acceptance Testing,  Functional Testing,  Feedback Plan,  CCRD Case Testing. | - | Aman Parmar |
| Project Plan | 1. Explanation of Different UP Stages. 2. Project Milestones and Objectives. | 1. Sandesh Raja Thapa 2. Noman Ahmed Khan |
| Inception Phase Assessment  Elaboration Phase Assessment | - | Sandesh Raja Thapa. |

**Individual Work Breakdown Structure Report for Overall Project:**

**Noman Ahmed Khan:** The following work has been done by Noman Ahmed Khan during this Semester.

|  |  |  |
| --- | --- | --- |
| **Task** | **Project Milestone** | **Delivery/Additional notes** |
| **Vision:** Completed the last iteration for the revision as well as the revised vision. | Vision/Revised Vision. | Delivered with LCOM and LCAM. |
| **Executive Summary:** written for LCOM and LCAM Reports. | No major milestone. Small summary of what is to be expected from document. | Delivered with LOM and LCAM |
| **Use Case Diagram with brief use case descriptions:** The use case diagram was presented and for each use case within the diagram a brief description chart was also presented. | Requirements Model in LCOM | Delivered with LCOM and committed in GitHub repository. |
| **Full Use Case Descriptions:** Full use case descriptions of the chosen CCRD Use cases along with Activity Diagrams to back the Full Use Case descriptions. | Revised Requirements Model in LCAM. | Delivered with LCAM and committed in GitHub repository. |
| **Architecture/Final Architecture:** Describe the proposed architecture that is to implemented, along with the goals that we wish to achieve, the constraints and justification for the chosen architecture. | Architecture Notebook in LCOM and Final Architecture in LCAM. | Delivered with LCOM and LCAM and committed in GitHub Repository. |
| **Executable Architecture:** The implementation of the finalized architecture was presented with the help of different UML Diagrams such as ERD, Architecture, Component and Deployment Diagram. | Executable Architecture in LCAM. | Delivered with LCAM and committed in GitHub Repository. |
| **Project Plan:** Lay out the Project Timeline with the Milestones, Major deliverables. | Project Plan/ Revised Project Plan. | Delivered with LCOM and LCAM. |

**Amankumar C. Parmar:** The following work has been done by Amankumar during this Semester.

|  |  |  |
| --- | --- | --- |
| **Task** | **Project Milestone** | **Delivery/Additional notes** |
| **Vision:** Completed the second iteration for the vision as well as revised vision. | Vision/Revised Vision | Delivered with LCOM and LCAM |
| **Domain Diagram:** Show the different classes to be used and the relationships between them. | Requirements Model/Revised Requirements Model. | Delivered with LCOM and LCAM and committed to GitHub repository. |
| **Technical Competency Demonstrator:** Set up GitHub repository where all the documents can be uploaded and kept track of, set up an integrated application- prototype website to showcase the skills of the team members. | Technical Competency Demonstration in LCOM | Delivered with LCOM. |
| **Master Test Plan:** Set out testing strategies to measure the outcome of the project, the testing environments and the different types of tests. | Master Test Plan/Revised Master Test Plan. | Delivered with LCOM and LCAM. |
| **Test Results:** The testing strategies that were set out in Master Test Plan along with the results with the help of User Acceptance Testing, CCRD Use Case testing, User Feedback Testing. | CCRD Use Case Test,  User Acceptance Test,  UAT Feedback Plan | Delivered with LCAM and committed to GitHub Repository. |

**Sandesh Raja Thapa:** The following work has been done by Sandesh Raja Thapa during this Semester.

|  |  |  |
| --- | --- | --- |
| **Task** | **Project Milestone** | **Delivery/Additional notes** |
| **Vision:** Completed the first iteration of the Vision as well as the revised vision. | Vision/Revised Vision. | Delivered with LCOM and LCAM. |
| **Non-Functional Requirements:** Explained the non-functional requirements that are set out in the vision and how the BookTrade wishes to achieve the successful implementation of those non-functional requirements. Those included, availability, reliability, accessibility, user interface, licensing etc. | Requirements Model/Revised Requirements Model. | Delivered with LCOM and LCAM. |
| **Requirements Traceability Matrix (RTM).** | Requirements Model/Revised Requirements Model. | Delivered with LCOM and LCAM. |
| **Risk Register:** Detail description of the risks that can impact the project and the plans to overcome those risks, the contingency plans and other risk related discussion. | Risk Register/Revised Risk Register | Delivered with LCOM and LCAM. |
| **Project Plan:** The basic description of the Project Plan and the different phases of the Project as per the Unified Process Software Development Model. | Project Plan/Revised Project Plan. | Delivered with LCOM and LCAM. |
| **Inception Phase Assessment:** Assessing the goals and the major milestone deliverables that were supposed to be achieved with the LCOM report at the end of the inception phase with the help of a description chart and Gantt chart. | Inception Phase Assessment. | Delivered with LCOM |
| **Elaboration Phase Assessment:** Assessing the goals and the major milestone deliverables that were supposed to be achieved with the LCAM report at the end of the Elaboration phase with the help of a description chart and Gantt chart. | Elaboration Phase Assessment | Delivered with LCAM |

REVISED VISION

### **Introduction**

Book Trade is an E-Commerce service started and sponsored by Noman, Sandesh and Aman as a final year software development project. We aim to provide a platform for users, especially students to have access to cheaper books. The goal of the project is to provide a website for users to efficiently buy/sell, trade and donate new or used books. This allows users to browse and compare prices of books from different users. Booktrade focuses on providing books in physical form.

The Unified Process (UP) Software Development Model is going to be used by the team from start till end to complete the Project.

### **Positioning**

#### Problem Statement

There are students who want books for education, research or for hobby. Sometimes they are not able to find books in proper time at a proper price, or maybe the book is required only for a temporary period. People have books which they don’t require anymore and they want to sell it, trade it or donate it.

|  |  |
| --- | --- |
| **Problem** | The problem of:   * Finding cheaper used books that are not available from university or bookshops. * Trading books among student and other book readers. * Selling your old used books. * Buying used books by comparing different prices. * Donating unwanted books. |
| **Affects** | This problem affects:   * Users who want book for cheaper price. * Users who want a book that is unavailable at universities or public libraries. * Users who want books in physical form. * Users who can’t find proper way or platform to trade books. * Users who can’t find proper way or platform to buy books. * Users who can’t find proper way or platform to sell books. * Users who can’t find proper way or platform to donate books. |
| **Impact** | The problem where book is not available for rent:   * Users to buy new book.   The problem where book is not even available to purchase:   * Users to compromise and work with different or alternate version of book. * Users to work without a book or rely on various materials which requires a lot of research and time.   The problem where book is not in physical form:   * Users will have to use e-books which is not preferred by everyone. * Users will have to rely on different source.   The problem of where a proper platform is missing to trade, sell, buy, or donate books:   * Users are not able to find customers or someone who needs books. * Users will have to keep or throw away unwanted books. |
| **Solution** | A successful solution is to provide a platform to users where:   * Search for affordable books. * Find books that are unavailable in stores or library but accessible through other users. * Buy, sell, trade or donate books easily, efficiently, quickly, and effectively. * Users can reach more people with online website providing global exposure. * Which helps them get rid of unwanted books by utilizing them in trading or donating rather than throwing away. |

#### Product Position Statement

|  |  |
| --- | --- |
| **For** | Book users. |
| **Who** | Who want books in physical form or want to trade books. |
| **The Booktrade** | is ecommerce type website |
| **That** | Features trading and donating books. |
| **Unlike** | Other ecommerce sites that only providing platform to buy and sell. |
| **Our product** | Provides trading platform free of cost |

### **Stakeholder Descriptions**

#### Stakeholder Summary

| **Name** | **Description** | **Responsibilities** |
| --- | --- | --- |
| Users | * Students * Booklovers in need of physical book * Donors * Sellers * General Users | * They will create user account. * Provide and manage a list of books, * Give genuine information about books condition, * Provide trading requirements along with the price for the book. |
| Web Administrator | Web administrator manages network related stuff. | * Maintain network and servers * Update network and servers * Troubleshoot network and servers * Ensure safety and security of network and servers |
| Project Manager | Monitor the entire project | * Entrusted with the responsibility of having control over the entire project * Leading and managing the whole team * Making sure the project is In line with its intended deliverables * Monitor the performance and progress |
| Web Developer | Develop the webpage | * Designing and coding the website. * Making sure that the servers and database are connected. * Delivering the finished product. |
| System Analyst | The person who is in charge of analyzing the whole Project. | * Gathering functional and non-functional requirements. * Cost Analysis * Implementation of timelines * Analyzing design considerations |
| Third Party Services | * Postal Service * Retail Suppliers (Bookshop Keepers) * Paypal Service | * Delivery of Books from one user to another, or from users to the company warehouse. * Book keepers can sell books through our website * Paypal services used to conduct online payments. |
| Quality Assurance Team | * The person who looks after the quality of the website. | * Make sure server isn’t down. * Make sure all the processes are being done correctly. * Look after any complaints related to website and implement any changes. |

#### User Environment

User environment depends upon the different users using the website or working on the website.

A student sitting within the library will be using the website over the desktops, or their personal laptops, phones. The physical environment will be quiet. At first students will create accounts which will take about 5 minutes. Browsing, selecting and payment process should take about 10-20 mins depending on each user.

Users browsing the system while in public areas such as parks, stations will be either using laptops, tablets or phones. The physical environment will be quite noisy at times of rush hour, The environment also varies upon the weather for that particular day. Browsing and processing times may vary upon different conditions.

Web administrator, developers will be in a quieter environment working on either their laptops or PCs. Their process times depends on the task they are carrying out.

### 

### **Product Overview**

#### Needs and Features

|  |  |  |  |
| --- | --- | --- | --- |
| **Need** | **Priority** | **Features** | **Planned Release** |
| Create user account | High | A system that allows users to create account and manage it through website. | At the time of deployment |
| Login account | High | A system that allows users to login and manage. | At the time of deployment |
| Create list of books, | High | A system that manages list of books that user can provide. | At the time of deployment |
| Browse Books | Medium | A system that allows users to browse books provided by other users. | At the time of deployment |
| Trade Books | High | A system that provides a platform to set trading requirements and send proposals for trading. | At the time of deployment |
| Buy Books | High | A system that allows users to buy books, | At the time of deployment |
| Sell Books | High | A system that provides a platform for user to sell books. | At the time of deployment |
| Donate Books | Medium | A system that allows users to donate books. | At the time of deployment |
| Payment system | High | A system that allows users to make safe and secure payments either using Booktrade system or third party service. | At the time of deployment |
| Postal Service | High | A system that delivers only traded books to users. | At the time of deployment |

### 

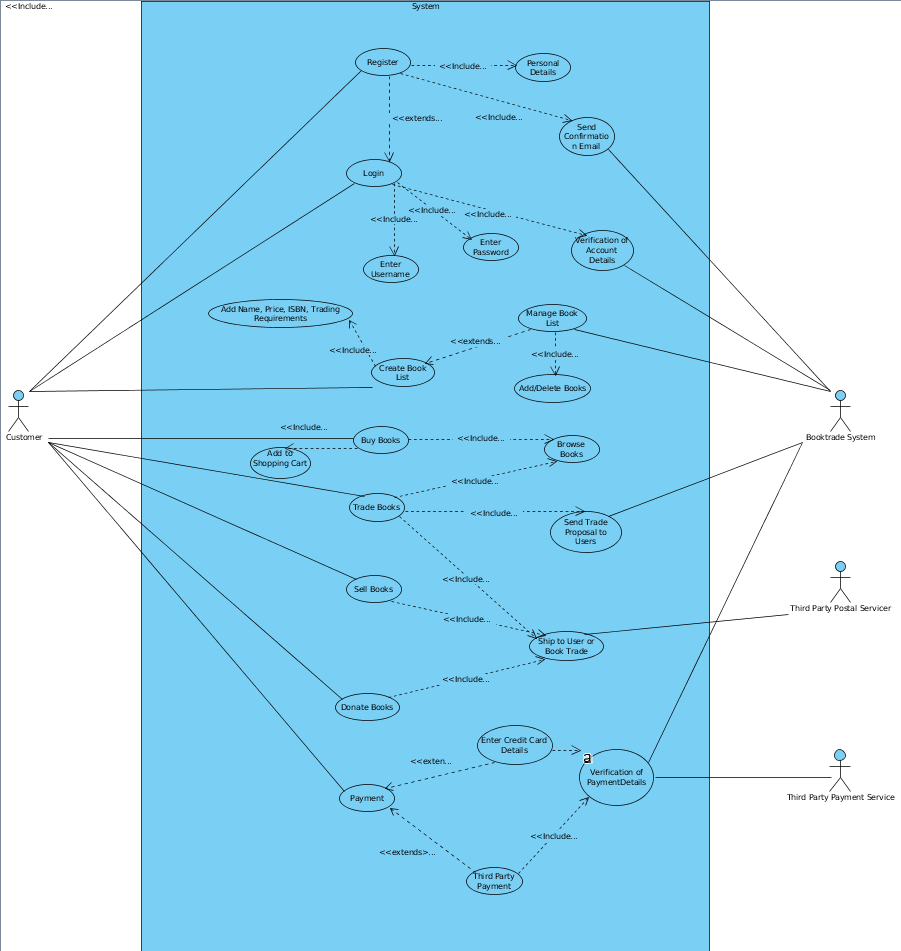
### **Other Product Requirements**

|  |  |  |
| --- | --- | --- |
| **Requirement** | **Priority** | **Planned Release** |
| Secure payment system | High | September 2017 |
| Reliability of servers to provide 24/7-hour service. | High | September 2017 |
| Internet to connect to website | Medium | April 2017 |
| A user-friendly layout and design for interaction | Medium | April 2017 |
| A FAQ (Frequently Asked Question) page for information. | Medium | April 2017 |

Requirements Model

### Unified Process (UP) Model is a use case driven model that is why we are presenting the requirements using the use case diagram, use case descriptions and supporting with UML diagram such as Activity Diagram.

### *Use Case Diagram:*



### *Use Case and Brief Description*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Use Case** | **Actors** | **Includes** | **Extends** | **Short Description** |
| Register | User System | Personal Details  Send Confirmation Email | Login | When a user registers for the Website, they have to provide their personal details, a confirmation email will be sent and they have an option to log in afterwards |
| Login | User System | Entering Username  Entering Password  Verification of Account Details |  | Logging in requires entering username and password. The account details will be verified by the system. |
| Create Book List | User System | Adding Name, Price, ISBN, Trading Requirements | Managing Book List | Creating book list includes adding the book lists. Privilege of Managing Books lists as well. |
| Managing Book List | User System | Add/ Delete Books |  | Managing book list allows you to Add or Delete Books |
| Buy Books | User System | Browsing Books  Adding to Shopping Cart |  | Buying books includes browsing the books and then adding them to the shopping cart |
| Trade Books | User System  Third Party Postal Service | Browsing Books  Send Trade Proposal to Users  Ship to User or Book Trade |  | Trading books requires browsing books, sending the trade proposal and shipping the book to Recipient Users or Book Trade |
| Sell Books | User System  Third Party Postal | Ship to User or Book Trade |  | Selling the books include shipping the books to Recipient Users or Book Trade |
| Donate Books | User System  Third Party Postal | Ship to User or Book Trade |  | Donating Books will include Shipping the books to Recipient Users or Book Trade |
| Payment | User System  Third Party Payment |  | Enter Credit Card Details  Third Party Payment | Users can pay either through the credit card or pay via third party services like PayPal and Bitcoin |
| Enter Credit Card Details | User System | Verification of Payment Details |  | When credit card details are entered, it will be verified by the system |
| Third Party Payment | User System  Third Party Payment Service | Verification of Payment Details |  | When Third Party Payment is done, Third Party Payment Service and Book Trade System verify it |

### **Full Use Case Descriptions of CCRD Use-Cases:**

The following use cases have been chosen as the CCRD Use-Case from the above Use-Case Diagram:

* Register
* Login
* Manage Book List
* Buy/Sell Books
* Trade Books
* Payment

# ­­­BookTrade

# Use-Case: Register

* Brief Description
  + When a user wants to create an account for the BookTrade website which will allow them to interact with other users, buy/sell/donate/trade books.
* Actors
  + User: Includes students, book readers who wish to buy/sell or trade/donate books.
* Pre-Conditions
  + Be using a laptop, personal computer, mobile phone.
  + Be connected to Internet.
* Normal Flow
* The use case begins when <actor> <does something>

|  |  |
| --- | --- |
| Actor | System |
| 1. Open BookTrade website and click on Register.  2. Enter credentials in the form. These include: username, Email, Password. | 1. Redirect the user to Register page.  2.1 System verifies the credentials.  2.2 System Registers the user into BookTrade Database.  2.3 System redirects to Log-In page. |

The use case ends.

* Alternate Flows
  + <alternate flow 1>
  + If at step <step number> of the normal flow <condition>, then

|  |  |
| --- | --- |
| Actor | System |
| 2. Invalid credentials entered. | 2.1 System allows the user to re-enter the credentials. |

* Post-conditions
  + User can now Log In to the website.
* Special Requirements
  + The password must be at least 4 characters long, be case sensitive, and contain special characters.

# Use-Case: Login

* Brief Description
  + When the user wants to login in order to carry out a transaction over BookTrade website.
* Actors
  + User: Include students, book readers who wish to buy/sell or trade/donate books.
* Pre-Conditions
  + Be registered with BookTrade.
  + Be using a laptop, personal computer, mobile phone.
  + Be connected to Internet
* Normal Flow
* The use case begins when <actor> <does something>

|  |  |
| --- | --- |
| Actor | System |
| 1. Open BookTrade website and click on Login.  2. Enter credentials in the form. These include: username, password. | 1. Redirect the user to Login page.  2.1 System verifies the credentials and logs in the user.  2.2 System redirects user to the homepage. |

The use case ends.

* Alternate Flows
  + <alternate flow 1>
  + If at step <step number> of the normal flow <condition>, then

|  |  |
| --- | --- |
| Actor | System |
| 2. Invalid credentials entered.  2.1 User re enters the credentails. | 2. System shows an error and tells the user to re enter the credentials.  2.1 System verifies the credentials and logs in the user.  2.2 System redirects the user to login page. |

* Post-conditions
* User can now buy, sell, trade or donate books.
* User can update their booklist.

# Use-Case: Create a Book List

* Brief Description
  + When the user wants to create a list of books they have to offer including book prices, trade conditions.
* Actors
  + User: Include students, book readers who wish to buy/sell or trade/donate books.
* Pre-Conditions
  + Be registered with BookTrade.
  + Be using a laptop, personal computer, mobile phone.
  + Be connected to Internet
* Normal Flow
* The use case begins when <actor> <does something>

|  |  |
| --- | --- |
| Actor | System |
| 1. Open BookTrade website and click on Login.  2. Enter credentials in the form. These include: username, password. | 1. Redirect the user to Login page.  2.1 System verifies the credentials and logs in the user.  2.2 System redirects user to the homepage. |

The use case ends.

* Alternate Flows
  + <alternate flow 1>
  + If at step <step number> of the normal flow <condition>, then

|  |  |
| --- | --- |
| Actor | System |
| 2. Invalid credentials entered.  2.1 User re enters the credentials. | 2. System shows an error and tells the user to re enter the credentials.  2.1 System verifies the credentials and logs in the user.  2.2 System redirects the user to login page. |

* Post-conditions
  + User can now buy, sell, trade or donate books.
  + User can update their booklist.

# Use-Case: Managing Book List

* Brief Description
* When a user wants to update their book list, either delete or add new books to the list and change prices.
* Actors
* User: Include students, book readers who wish to buy/sell or trade/donate books.
* Pre-Conditions
  + Be registered with BookTrade.
  + Be using a laptop, personal computer, mobile phone.
  + Be connected to Internet
  + Login to BookTrade website and access your book list.
* Normal Flow
* The use case begins when <actor> <does something>

|  |  |
| --- | --- |
| Actor | System |
| 1. Click on edit book list button.  2. Add or Delete the name, price, ISBN number of the book and click on submit button. | 1. Redirect the user to the book list page.  2.1 Verify the book list with user and make changes to the current book list. |

The use case ends.

* Alternate Flows
  + <alternate flow 1>
  + If at step <step number> of the normal flow <condition>, then

|  |  |
| --- | --- |
| Actor | System |
| 2. Incorrect name, ISBN number or price entered.  2.1 User checks that they have mistyped the details and corrects them.  2.2 User submits. | 2. As system verifies the list with the user.  2.1 System verifies the details once more with the user.  2.2 System updates the book list. |

* Post-conditions
  + User can continue browsing books.

# Use-Case: Buying/Selling Books

* Brief Description
  + User wants to buy a book using BookTrade website.
* Actors
  + User: Include students, book readers who wish to buy/sell or trade/donate books.
* Pre-Conditions
  + Be registered with BookTrade.
  + Be using a laptop, personal computer, mobile phone.
  + Be connected to Internet
  + Login to BookTrade.
* Normal Flow
  + The use case begins when <actor> <does something>

|  |  |
| --- | --- |
| Actor | System |
| 1. User Searches for the book using either the book title, author name, or ISBN number.  2. User compares the prices of books being offered by different users.  3. User selects the book they want to buy.  4. User enters their address details.  5. User waits for confirmation from BookTrade to continue with the payment.  6. User selects the payment method and enters the required details. | 1. The system shows list of books, along with their prices and different users offering those books.  3. System takes the user to the confirmation page, where the user enters their address.  4. System verifies the address with the user.  5. Book Trade sends confirmation email to user to continue with payment.  6. System confirms the payment. |

The use case ends.

* Alternate Flows
  + <alternate flow 1>
  + If at step <step number> of the normal flow <condition>, then

|  |  |
| --- | --- |
| Actor | System |
| 1. If at step 1, user is unable to find book that they were looking for, they email BookTrade.  2. If at step 4, user enters incorrect address details.  2b. User re-enters the correct address.  3a. If at step 6, user enters incorrect payment details.  3b. User enters the details again. | 1. BookTrade keeps an eye out for that book and lets the user know once its available.  2. System confirms the address before proceeding.  2b. System proceeds with confirmation and payment.  3a. System lets the user know that the payment could not be verified.  3b. System confirms the details and continues with the transaction. |

* Post-conditions
  + User waits for the book to arrive
  + User can continue browsing more books.
* Special Requirements
  + User must have PayPal Accounts.

# Use-Case: Trade Books

* Brief Description
  + User has books that they might want to trade, or other things that they are willing to trade for books with other users. They have trade requirements for each book they are willing to trade and send/receive trade proposals to/from interested users.
* Actors
  + User: Include students, book readers who wish to buy/sell or trade/donate books.
* Pre-Conditions
  + Be registered with BookTrade.
  + Be using a laptop, personal computer, mobile phone.
  + Be connected to Internet
  + Login to BookTrade.
  + Have written trade requirements with their book list.
* Normal Flow
  + The use case begins when <actor> <does something>

|  |  |
| --- | --- |
| Actor | System |
| 1. User Searches for the book using either the book title, author name, or ISBN number.  2a. User looks at if there are any trade requirements for the required book.  2b. User sends a trade proposal to the intended user offering the book.  3. The intended user (who is offering the book for trade) receives the trade proposal and replies back to the user who sent the proposal.  4. The user has agreed to trade and they proceed with trading.  5. The users (receiver and sender) enters their address details. | 1. The system shows list of books, along with their prices and different users offering those books.  2b. System sends the trade proposal to the intended user.  3. System sends the reply back to the user.  4. Book Trade redirects the user to address details page.  5. The system confirms the address details and sends the confirmation to the user who will be sending the book. |

The use case ends.

* Alternate Flows
  + <alternate flow 1>
  + If at step <step number> of the normal flow <condition>, then

|  |  |
| --- | --- |
| Actor | System |
| 1. If at step 1, user is unable to find book that they were looking for, they email BookTrade.  2. If at step 4, user enters incorrect address details.  2b. User re-enters the correct address. | 1. BookTrade keeps an eye out for that book and lets the user know once its available.  2. System confirms the address before proceeding.  2b. System proceeds with confirmation and trading. |

* Post-conditions
  + User waits for the book to arrive
  + User can continue browsing more books.

# Use-Case: Payment

* Brief Description
  + User has decided to buy a book and wish to make payment in order to continue with the transaction.
* Actors
  + User: Include students, book readers who wish to buy books and the who wish to sell books.
* Pre-Conditions
  + Be registered with BookTrade.
  + Be using a laptop, personal computer, mobile phone.
  + Be connected to Internet
  + Login to BookTrade.
  + Have searched and selected to buy the book they want.
  + Have received a confirmation from BookTrade to proceed with payment.
* Normal Flow
  + The use case begins when <actor> <does something>

|  |  |
| --- | --- |
| Actor | System |
| 1. User selects the payment method to make payment using their credit card over booktrade.  2. User enters their details which includes, AccountName, CardNumber, SecurityNumber, ExpirationDate and continues to make payment. | 1. System redirects user to a secured payment page.  2. System Verifies the account details with the user.  2a. System proceeds with transaction. |
|  |  |

The use case ends.

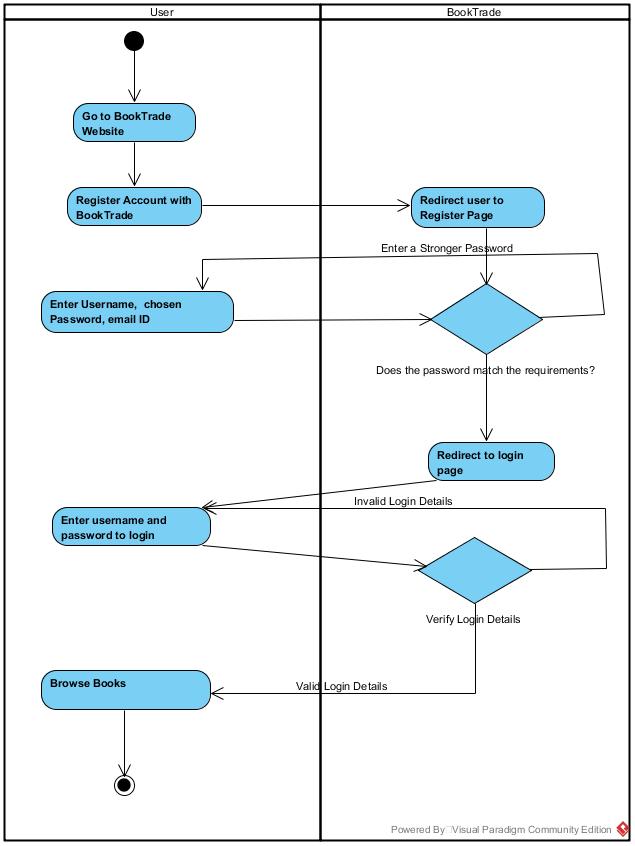
* Alternate Flows
  + <alternate flow 1>
  + If at step <step number> of the normal flow <condition>, then

|  |  |
| --- | --- |
| Actor | System |
| 1. If at step 1, user chooses to pay using third-party agent such as PayPal.  1b. User enters their credentials to login.  1c. User confirms that they wish to make the payment.  2. If at step 2, user enters incorrect card details.  2b. User will re-enter their credit card details. | 1. System redirects the user to the PayPal login page.  1b. System logs the user to their PayPal account.  1c. System proceeds with the transaction and sends the user a confirmation that they made payment with PayPal.  2. The payment is not authorized the user is asked to re-enter his/her credit card details.  2b. System verifies the details and continues with the transaction. |

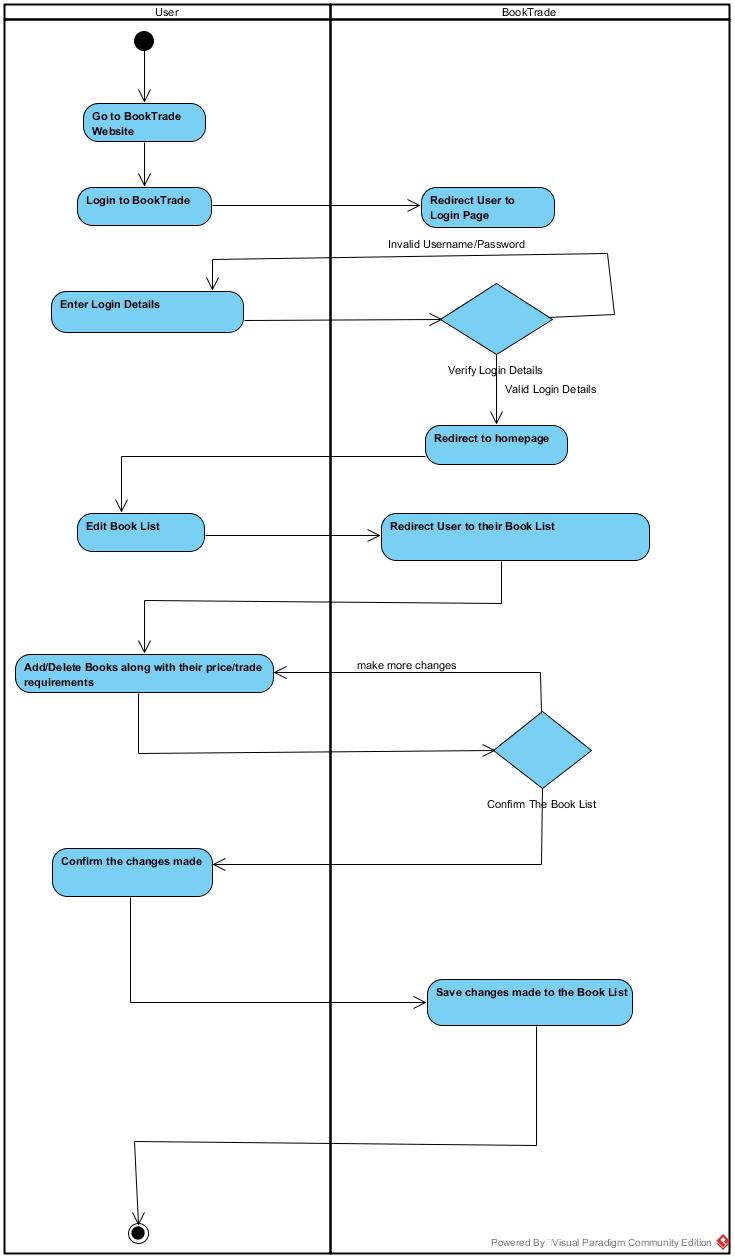
* Post-conditions
  + User waits for the book to arrive
  + User can continue browsing more books.

### **Activity Diagrams for the CCRD Use Case:**

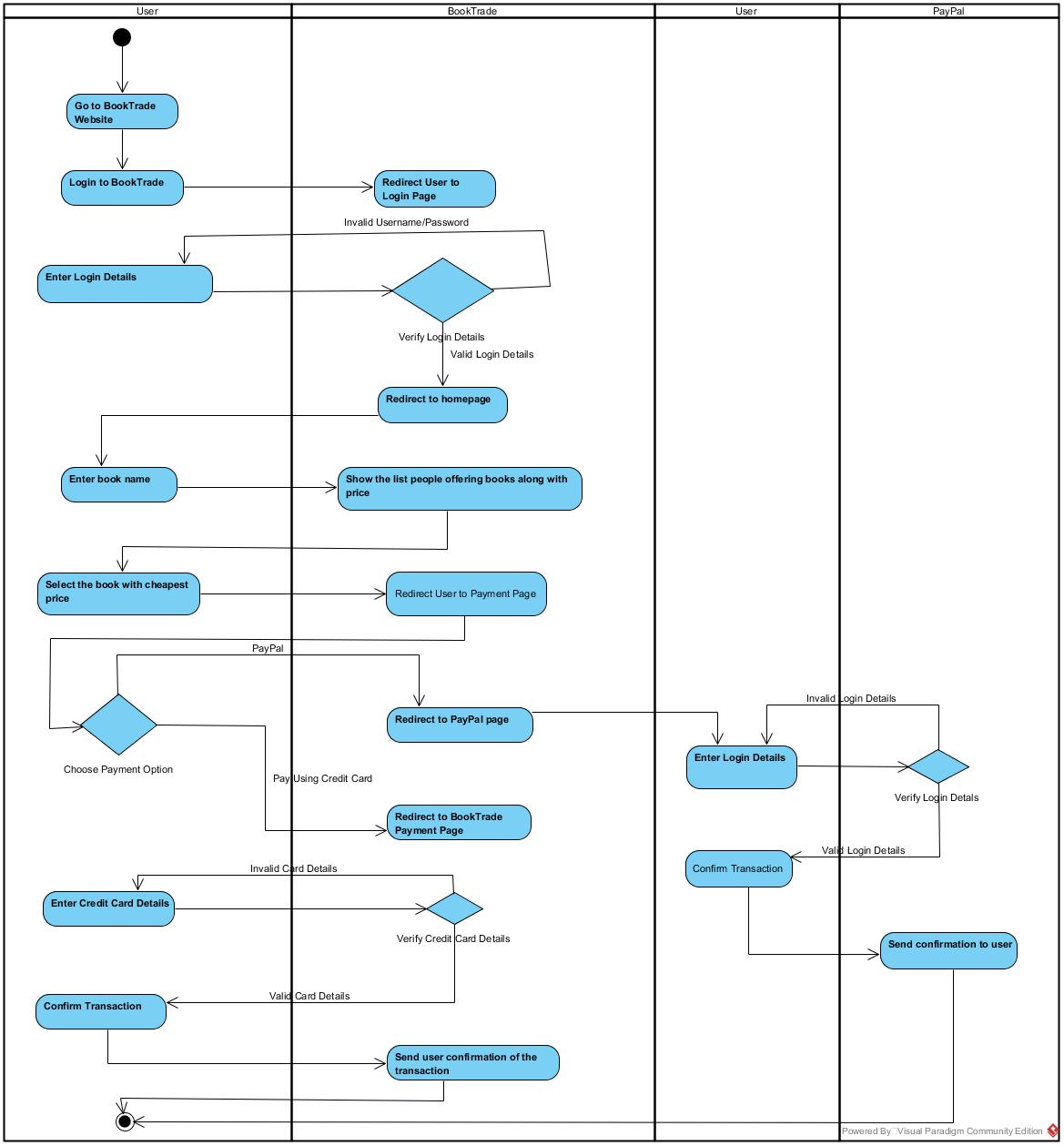
* **Register/Login:**



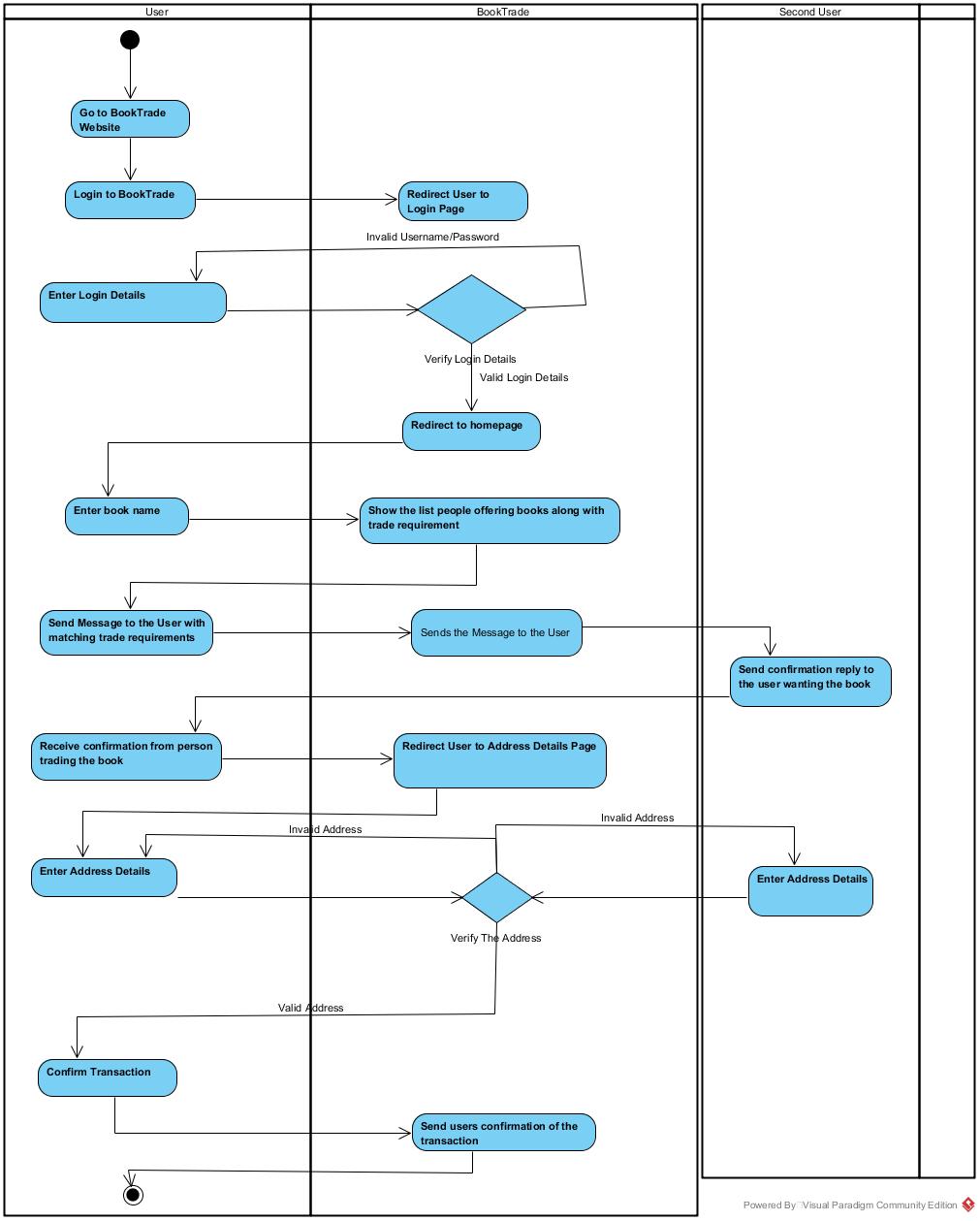
* **Manage Book List:**

****

* **Buy/Sell:**



* **Trade:**



\*Payment is already included within the activity diagrams for buy/sell.

### **Domain Model Diagram**

## 

Revised Requirements Traceability Matrix

The following trace matrix examples demonstrated are:

* BR1 – Business Requirement 1
* BRA = Business Requirement Area
* BRD1 – Business Requirement Deliverable1

| **Business Requirement** | **Area** | **Deliverables** | **Status** |
| --- | --- | --- | --- |
| BR1  The system should allow the new user to register into the system and then the system should successfully accept the personal details and hence send confirmation email | BRA | BRD1   * Register Design | Accepted |
| BR2  The system should allow the registered users to login and hence verify the credentials that support the login | BRA | BRD2   * Login Design * User Credentials * Account Details * Verification | Accepted |
| BR3  The system should allow the user to create the book list and accept the book specifications | BRA | BRD3   * Creating Books Design * Book Specifications Addition | Accepted |
| BR4  The system should allow the users to manage the book list by adding or deleting the books | BRA | BRD4   * Managing Books Design * Books Addition and Deletion | Accepted |
| BR5  The system should allow the users to buy books and hence facilitate them to browse books and add them to shopping cart | BRA | BRD5   * Books Design * Shopping cart | Accepted |
| BR6  The system should allow the users to trade books by browsing them and following trade requirements | BRA | BRD6   * Trade Design * Shipping Implementation | Accepted |
| BR7  The system should allow the users to sell books by collaborating with the third party if required | BRA | BRD7   * Selling Design * Shipping Implementation | Accepted |
| BR8  The system should allow the users to donate books by collaborating with the third party if required | BRA | BRD8   * Donating Design * Third Party Collaboration | Accepted |
| BR9  The system should facilitate payment by using credit card details or by using Third Party System and hence verify the payment details | BRA | BRD9   * Payment Design * Credit Card Authorization * Third Party Payment | Accepted |

| **BizReqID** | **BRD1** | **BRD2** | **BRD3** | **BRD4** | **BRD5** | **BRD6** | **BRD7** | **BRD8** | **BRD9** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| BR1 | X |  |  |  |  |  |  |  |  |
| BR2 |  | X |  |  |  |  |  |  |  |
| BR3 |  |  | X |  |  |  |  |  |  |
| BR4 |  |  |  | X |  |  |  |  |  |
| BR5 |  |  |  |  | X |  |  |  |  |
| BR6 |  |  |  |  |  | X |  |  |  |
| BR7 |  |  |  |  |  |  | X |  |  |
| BR8 |  |  |  |  |  |  |  | X |  |
| BR9 |  |  |  |  |  |  |  |  | X |

Revised Non-Functional Requirements

### *Introduction*

BookTrade consists of other supporting requirements which help the website run as intended. Most of the requirements which are mentioned below is for our Book Trade website. The following sections accurately identify, are important and help to achieve the basic working and functionalities of the system.

### **System wide Functional Requirements**

#### Data and Database Requirements

The system needs to be able to maintain and capture a set of characters and numbers that are in use. It will store, update, capture, manage and retrieve data whether is structured or unstructured like the free text and images. The database stores information about the books, buyers, sellers, donators, prices and all sorts of information that are necessary to be recorded. The system is also able to export the document in as a Word Document or a PDF Document.

#### **Privacy, Authentication and System Security Requirements (Data Security)**

In today’s modern world, data security is crucial to any business because of its contribution to protect and preserve the system components, information and integrity. Only the web administrator as described in our Vision document will be able to access the permissions and right controls. Only the web administrator is allowed to interact directly with the records in the database.

The login screen begins and grants authorization only to the users with a unique username and password. There are limited number of login attempts after which security codes will be sent to protect identity theft. The system facilitates automatic log off after a period of inactivity. The major things such as the user’s confidentiality, data integrity and data availability is ensured.

1. **Process Requirements**

The process controls which deal with recovery procedures and backup system will be present. A strong foundation free of bugs, viruses, spyware, malware, and other harmful codes will be created.

1. **Backup Requirements**

The vital information of the users in the Book Trade website need to be backed up early and often. It will be a very tedious and risky task in case of system failure. It is important to configure different backup settings and implement them so that all the important and other files stay safe and just in case if something happens, recovery processes can be brought to implementation immediately.

### **System Qualities**

#### Usability

The user interface will be relatively simple and will allow the beginners in the website to get through basic operation with a short duration of time. Also, ensuring that any type of information needed will be available in less than four clicks. If in need of any additional resources in the website, they will be easily added.

#### Reliability

The website needs to be reliable and downtime issues will be handled effectively as per the business procedures. There will be efforts to trying, implementing and experimenting with various servers to check reliability, The system shall be maintained during business hours. The system has enough local and remote server systems to continue provide the service unless a big disaster strikes. Effective run times and fast streaming speeds are ensured.

#### Availability

The Book Trade website should be available when needed

#### Privacy

The Book Trade website will ensure that the personal information to user’s information will remain private and personal space shall not be invaded

#### Scalability

The Book Trade website will be able to facilitate increasing number of users as time passes.

#### Usefulness

The Book Trade website will continue to offer valuable and worthwhile services helping the users carry out the intended features and functionalities of the website

#### Cost-Effectiveness

The Book Trade website will continue to operate with benefits and will try to outweigh the costs by providing valuable features to the students.

#### Performance

Whenever the home page is opened after the logging in process, the system should be able to display all the required information in less than 5 seconds. The system should continue to deliver fast and optimized performance in all sections of the websites and pages without issues. Efficient optimization with the RAM, operating system, bandwidth, and disk drive space with almost no lag issues will be implemented.

#### Supportability

The web site system will go through various tests which will ensure testability. The website should run smoothly and be portable in the most common devices and operating systems we use in daily life. The coding has been done effectively so as to reuse the components by just reviewing several changes. Web administrator will take care of any bugs that try to hamper with the system.

### **System Interfaces**

#### User Interfaces

The User Interface is essential because it outlines how the user interacts with the system. As per now, normal surfing interface in the Book Trade Website, Graphical User Interface and obviously touch interface for the devices that are compatible with it.

* Welcome screen to give the user a warm welcome
* Password-Protected login processes
* Simple and highly readable fonts, background color and overall interface.

#### Look & Feel

The interface will be simple and plain as possible so that even beginners can go through it without any hassle and trouble.

#### Layout and Navigation Requirements

The layout will be simple and represented like the web layout. The user will be able to navigate to various pages if they have the access. A normal and simple web page design will be implemented following the basics.

#### Consistency

Consistent fonts with appropriate screen density. The pages are designed to be clutter free and only the necessary information required will be presented. The Book Trade logo will be placed at the top of the website. The pages are print friendly and will deliver good pages when printed without issues.

#### User Personalization & Customization Requirements

The users can customize some of the features according to their needs. Being able to zoom in, change the alert notifications, changing the fonts in the pages are some of the customizations you can do.

#### **Interfaces to External Systems or Devices**

The system needs to interact and interface with the server application since the architecture is **3 Tier Architecture System**, where there are Users, Servers, and the Database System.

The system needs to interface with the server applications since it is a cloud-based system. The data travels from the server to the recipients while gathering the information from the database.

##### *Software Interfaces*

Object oriented language is used to design the software interfaces. It contains set of classes with an associated list of class methods. Various Protocols are used and an integration with the software is thought of for various Operating Systems.

##### *Hardware Interfaces*

The architecture that is used to connect devices with each other which includes plugs, sockets, cables, and electrical signals. The presence of web server and database server to facilitate users with all-time information access is present. There are strong security system measures to provide security and protect important data.

It is the architecture used to connect two devices to each other. It includes plugs, sockets, cables and electrical signals.

##### *Communications Interfaces*

The system connects to other systems using local area networks whereas the devices get connected through wireless network access. If the user wishes to access the site from a device that has no wireless connection, 3G or 4G data services need to be used.

### **Business Rules**

Business rules are generally procedures, conditions which define the roles and responsibilities that a certain individual must carry out.

#### **<Book Trade>**

##### **<Access Rights and Controls>**

* This business rule discusses about the access rights and controls of the Book Trade System.
* The Web Administrator has the overall responsibility of taking care of the website. The role extends to taking care of the Database and ensure its proper functioning.
* The information uploaded in the website must not be leaked to third-parties to prevent data breaches and leaked privacy.
* Ethical measures are expected from all users in order to allow the system to run smoothly and efficiently as expected.

### 

### **System Constraints**

To create such a complicated system where just 3 Students, act in a project, it requires a major investment in time, effort and diligent. The major constraint that website faces is that since it is a fresh website, users will have difficulty finding the website since it is difficult for start-ups to occupy the top results in search engine results. The compatibility of this website for mobile users remain a part of major discussion and whether it is going to be implemented or not depends on the circumstances. The website is completely dependent on the wireless connection and no facilities can be used without proper internet. Obviously, the website is being built by beginners so there can be constraints on database systems, processors, memory, firewall and operating system issues.

### 

### **System Compliance**

#### Licensing Requirements

The website has a certain licensing requirements allowing only the designated individual software is a licensed product which means that only the authorized individual has the license of the website.

#### Legal, Copyright, and Other Notices

The website has its own logo and third parties may not use the software logo. The processes and all the services are owned by Book Trade and some other third-party services to whom Book Trade might or may not give access to. No part of the website must be replicated distributed, modified, copied and displayed unless the permission has been given to do so. The word “BookTrade” has been trademarked.

#### Applicable Standards

The website system contains personal information about the individuals so it may be subject to legislation and consent. Privacy policy will be essential to grant users the idea of what will happen with their information. Standardization of such systems also benefits the health in the wider community. The processes and technologies involved in the system should also be standardized.

### 

### **System Documentation**

System documentation covers a wide variety of documents within any industry including Information Technology. This technical document will be created by a group of people to assure that the final output is accurate, complete, and readable. Some of them would be:

* Technical Website Knowledge
* Getting around and System Help Files
* Privacy Policy and Compliance
* Contribution and Donation
* Risk Management Procedures
* Change Management Procedures
* Data Security Measures

Finalized Architecture Notebook

There is guidance within this template that appears in a style named InfoBlue. This style has a hidden font attribute that allows you to toggle whether it is visible or hidden in this template. Use the Microsoft® Word® menu **Tools > Options > View > Hidden Text** check box to toggle this setting. There is also an option for printing: **Tools > Options > Print**.

### Purpose

The purpose of an Architecture Design is to allow the stakeholders to understand the overall design of the whole system in relation to the functional and non-functional requirements set out in the vision. To achieve a robust and flexible system, special emphasis will be given to producing an Architecture which builds upon the non-functional requirements of availability, reliability, usability, and performance. The below proposed Architecture design is complete, and will be the basis of deploying an executable architecture. The programmers are to keep in mind of this design which will make it easier for them to understand the requirements of the system and produce the website.

### Architectural goals and philosophy

No changes have been made to the goals and philosophy of the original proposed architecture which were as follows:

The main architectural goal is to implement a system that will be robust, and flexible. An architectural design is to be established that requires minimal maintenance but is also able to adapt to the additional requirements in the future. Since our project is a web-based e-commerce system and allows users to perform a variety of tasks, a layered/modular architecture is to be used. Security is to be one of the key issues that needs to be handled well as personal information such as credit card numbers and email id’s will be stored within the system. For this reason, the website will be hosted on a web-server established over our own personal device using a Linux operating system, which would ensure more control and security over the website. A modular and layered architecture would allow the functionalities to be independent of one another and will make it easier for the system to be implemented, tested and managed.

### 

### Assumptions and dependencies

Assumptions include:

* Not all users feel comfortable exposing their credit card details over websites and wish to use more secure payment services. For this reason, our website is going to implement third party payment services such as PayPal and bitcoin to allow users to make payments without giving away their credit card details.
* Using MySQL database server will provide higher quality database service and more security. This is critical as website needs a secure and well tested database to be running at all times. Also, provides a friendlier workbench interface to allow our team of young developers to work effectively.
* Using Apache II server on Linux operating system should provide greater security.

Dependencies:

* Any changes to the user requirements may require changes to be made within the database.

### 

### Architecturally significant requirements

Most of the significant requirements have been discussed in detail within the System Wide Requirements section. For a better understanding refer to the NFRs document above.

Some basic significant requirements include:

* The website should be quick and responsive and not take longer than 5 seconds to respond the users command.
* The users should be able to use the website on all devices and operating systems without any errors.
* A user-friendly interface is to be provided to allow users to navigate the website effectively and efficiently.

### Decisions, constraints, and justifications

Some key decisions along with their justifications provided below:

* Using MySQL as the default database server. MySQL is an industry certified database application which is a lot easier to install and setup. Providing a user-friendly workbench for even beginners to use, it allows our developers to be more effective. MySQL also provides better security as compared to other database applications.
* Third-party payment services: This is to make payments quicker and more secure for the users. May users have already established PayPal accounts and feel comfortable using such services over providing their credit card details to newer websites.
* Hosting the web-server over our own personal computer: For better security and easier maintenance we have decided to use Apache II server running on Linux Operating System. This should also give our developers and admin more authority over the server and website.
* Building website from scratch rather than using a third-party website building software: We want to do everything from front-end programming to back-end programming on our own so that at the end of project, our development team have learned newer skills and improved on their current skills.

### Architectural Framework and Design:

The following diagram represents the way we wish to apply the 3-tier architecture for the BookTrade System.

This architecture consists of:

1. The Client Side: This is any host computer/tablet/phone which will use the BookTrade website.
2. The server side: Apache II is used as the server, this is where the actual website will be deployed.
3. The database side: Back-end database server- MySQL is used to store and communicate data with the server.

Together the above three components make up the 3-tier architecture.

## 

## 

Executable Architecture

With Unified Process Model, it is important to develop an architecture as early as possible by the elaboration phase so that work can be started with the construction phase. Changes can always be made later on, but it is better have a detailed document (such as finalized architecture notebook) to support your executable architecture so minimal changes are needed later on.

This section of the report is to address how the finalized architecture discussed in the above finalized architecture notebook is to be deployed and implemented for BookTrade system.

The Executable Architecture will be discussed and implemented in accordance to the chosen CCRD Use-Cases. To understand why those Use-Cases were chosen below is a list of terms that apply to each use-case making them either core, critical, risky and/or difficult:

|  |  |
| --- | --- |
| Use-Cases | What makes it CCRD? |
| * Register * Login * Manage Book List | * Security: The register/login details need to be stored securely into the database. * Data Consistency: Data must be consistently presented with each user’s profile. * Connection: The connection of the servers must be well established at all times to allow users to use the website. * Script Execution: The script on the pages must execute to allow smooth running of the website. * System and Server Synchronization. |
| * Buy/Sell * Trade | * Communication: Both parties must be able to communicate with one another to ensure the transaction/trade is being carried on. * Accessibility: Users must be able to access the website. * Usability: Users must be able to use the website easily, must be able to perform the required tasks. * System and Server Synchronization. |
| * Payment | * Secure Connection: The connection is being carried out within secure environment, to prevent essential credit card details from being leaked. * Accurate and Consistent Billing: Only the amount that was supposed to be charged is charged. * Payment Rollback: If payment process is interrupted, the payment does not go through. * Accurate Transaction Report: Accurate receipts are presented to the users about the transaction. |

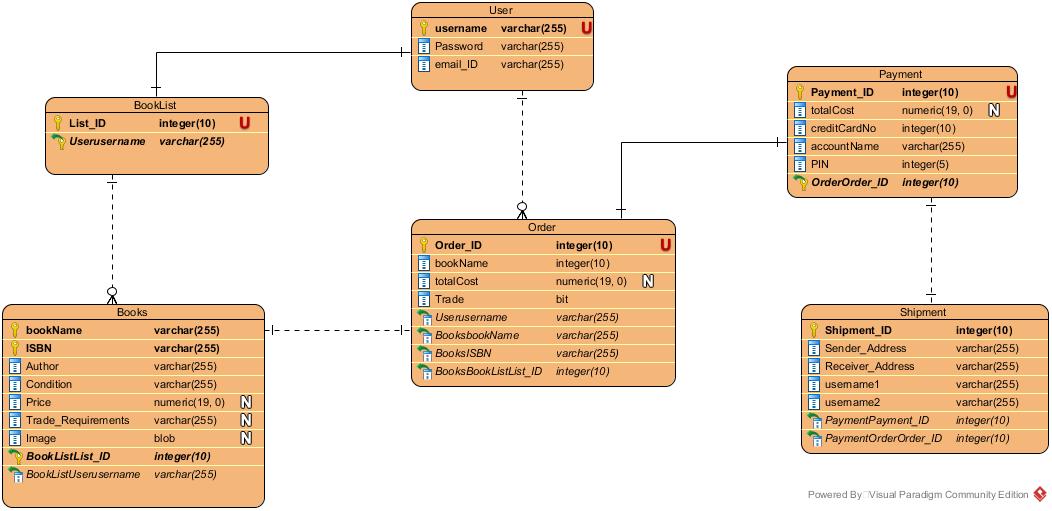
The Executable Architecture is to be presented with the help of the following diagrams:

* Entity Relationship Diagram (ERD)
* Architecture Diagram
* Component Diagram
* Deployment Diagram

**Entity Relationship Diagram (ERD):**

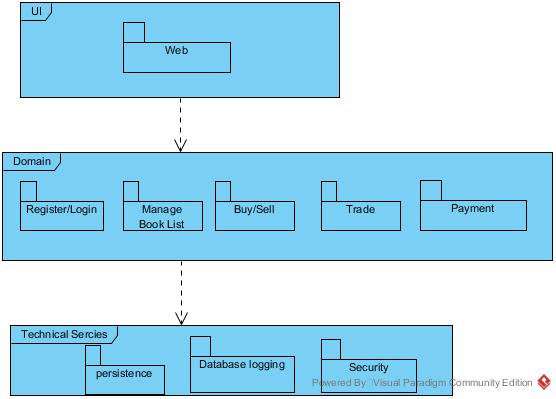
The purpose of entity relationship diagram is to show the final executable architecture of the database that is to be implemented at the back-end to store and retrieve data for the BookTrade website. We are using MySQL server, MySQL workbench to manage the BookTrade database.

Below is the Entity Relation Diagram for the proposed architecture:



**Architecture Diagram:**

The purpose of Architecture Diagram is to show the different layers that are being implemented within the proposed executable architecture:



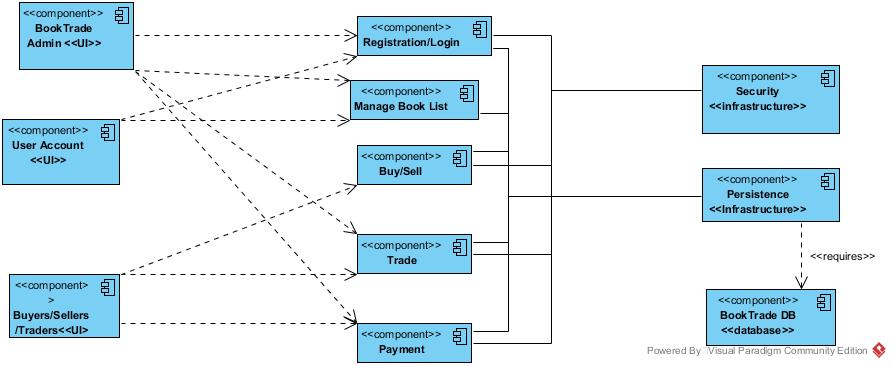
The three layers are:

* User Interface: This is the client side layer, which is the website and how it is presented to the user.
* Domain: This is the server side layer, all the main functions of the website that are executed on the server.
* Technical Services: This is the back-end database layer, the database logging, security and persistence is part of this layer.

**Component Diagram:**

The purpose of the component diagram is to show how the different components of the architecture are interacting with one another and what User-Interface is presented to the different type of people that are interacting with the BookTrade website.

The Component Diagram is below:



As you can see the above diagram shows that three main different type of people will be using the BookTrade website for different purposes:

* Admin: From the above diagram it is clear that the admin will have full control over all of the domain components of the BookTrade Website.
* User Account: This is the profile of the user where they manage their registration/login details as well as manage the book list that has been created.
* Buyers/Sellers/Traders: The users that will be interacting with the website in order to sell, buy, or trade books. They will have access to the Buy/Sell, Trade and Payment component of the Website.

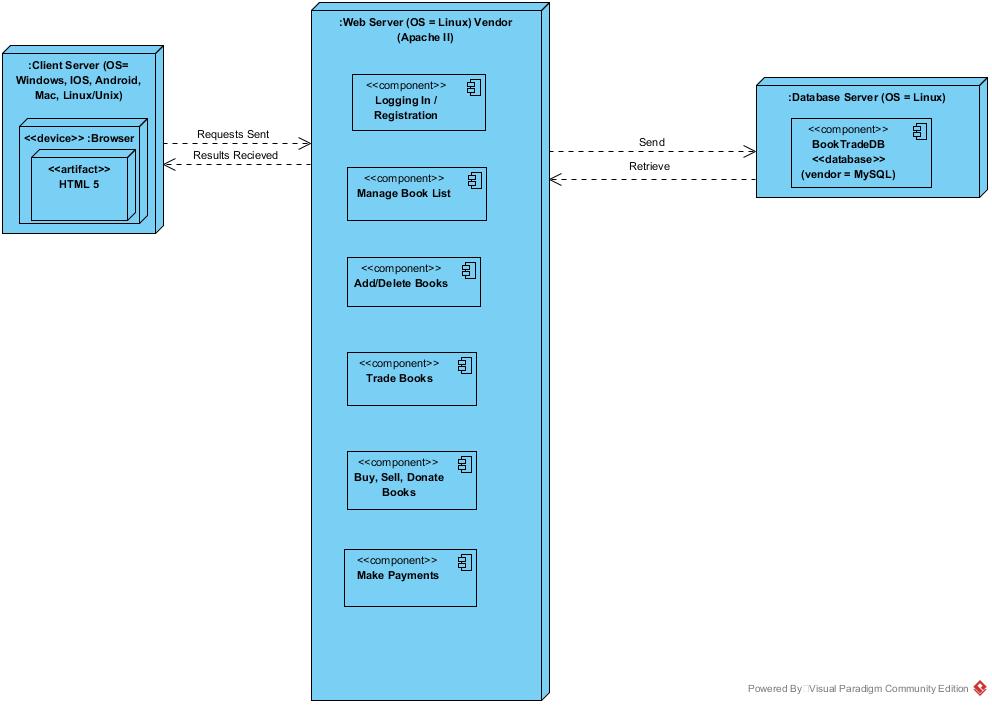
Also from the above component diagram it is visible how the different components of the system are interacting with one another- mostly independent, as well as with the database.

**Deployment Diagram:**

The purpose of the deployment diagram is to show the final overall architecture of the System. This will include:

* The different layers of the architecture.
* On what platform each layer is to be executed.
* How the connection is established between each layer.
* What are the components of each layer.

Below is the Deployment Diagram:



As seen in the above diagram there is a 3-tier architecture being implemented, the following are the layers and their execution setup:

* Client Server: This is the Website Interface that will be presented to users. It is to be executed on HTML 5 supporting browsers on any of the following platforms:
  + IOS Devices (Iphones, Ipads, Mac)
  + Android Devices (Tablets, Smartphones)
  + Windows OS (PC’s, Laptops, Tablets, Smartphones)
  + Linux/Unix OS (PC’s, Laptops)
* Web Server Side: This is the Web Server Apache II that is being run on Linux operating System. On the server, the website is being hosted which performs the following tasks:
  + Registering/Logging User
  + Allowing user to create/manage book list
  + Buy, sell, trade or donate books
  + Make payments for the books they have purchased.
* Database Server Side: This is the Database Server that is also being run on the Linux operating system. The vendor is MySQL. Here, all the data related to the user account, billing and payment is being recorded.

The communication between each layer is done through TCP/IP requests.

Technical Competency Demonstrator

Technical Competency is the combined skill and knowledge required by everyone within the organization to successfully complete a given task or a project. This is essential for the successful completion of the project especially when there is limited amount of time to work with. As our Project is to be deployed by the end of next semester, our team has carefully chosen the tools that we will work with, keeping in mind the knowledge and skills of each team member. The following are some of the technical competencies that our team possesses:

* Efficiency in HTML, CSS, PHP: Since our Project is a web-based project, our team has chosen to work with HTML, CSS, and PHP to build a dynamic website which implements a 3-tier architecture model. The front end will be handled using HTML and CSS programming languages, while the communication between the front-end and back-end will be handled using PHP. Our developers Aman and Noman are more than capable of Programming with the required languages as they have had experience working with above languages in previous courses, as well as learning on their own.
* Database Programming: Programming and managing a database is just as important as the website because database is the 3rd part of our 3-tier architecture model. For the current project, our team has chosen to work with MySQL database which is very commonly available and has features such as SQL Workbench that supports even the beginners. Our team is very much capable of producing a workable database that connects to our web server and allows for dynamic website to be built.

The above two technical competencies are shown in the links provided within this document. A live website has already been set-up which is running on Apache II server and just to show our competence we have also set-up a working log-in and registration page that allows for users to be added in to and loaded from the database.

Some other technical competencies include:

* Developing Talent: The aim of the project is not only to deploy a working software, but also to build upon the current skills of each team member and help other less capable team members to grow as better software developers. For this reason, it is extremely important that each team member communicates with others and help them understand any new concept or skill that they have acquired which is important for the completion of the project.
* Team Work and Team Leadership: Since our project team consists of only three people, each one of the members have an important leadership role to play to guide the project towards completion. Our team consists of capable leaders and professionals who work well with each other and will do what is required of them to complete their tasks on time.

## **Three Tier Architecture in Work**

##### *An Integrated Application:*

A website with working user registration and login is established as a prototype to demonstrate a working architecture with apache2 Server, Mysql Server database and Website(Client interface). Here is the link for website: [https://booktrade.duckdns.org](https://booktrade.duckdns.org/)

##### *Version Control Directory:*

For version control of Book.trade GitHub is being used. Book.trade source code can be found with version information here:

<https://github.com/in-fidle/booktrade/tree/master/website>

##### *Architecture*:

Diagrams explains architectures are displayed in directories at GitHub. Here is the link:

<https://github.com/in-fidle/booktrade/tree/master/diagrams>

##### *Server Specification:*

For server specifications access this link:

<https://httpd.apache.org/>

##### *Database Specification:*

MySql database is used for server database. More information on database can be found on following link:

<https://www.mysql.com/>

##### *Back-end Language:*

PHP is used as back-end language. It’s features and usability information can be found on this link:

<http://php.net/manual/en/intro-whatis.php>

##### *IDE used for developing source code:*

Open source editor Atom is used for developing website. More information about is in following link.

<https://atom.io/>

##### *Development environment for Website:*

Linux operating system is used for development of website. More information about the specific Linux distro can be found here:

<https://www.ubuntu.com/>

##### **Technical Specification:**

|  |  |
| --- | --- |
| System | Linux acp-HP-ENVY-15-Notebook-PC 4.8.0-46-generic #49~16.04.1-Ubuntu SMP Fri Mar 31 14:51:03 UTC 2017 x86\_64 |
| Server API | Apache 2.0 Handler |
| Virtual Directory Support | disabled |
| Configuration File (php.ini) Path | /etc/php/7.0/apache2 |
| Loaded Configuration File | /etc/php/7.0/apache2/php.ini |
| Scan this dir for additional .ini files | /etc/php/7.0/apache2/conf.d |
| PHP API | 20151012 |
| PHP Extension | 20151012 |
| Zend Extension | 320151012 |
| Zend Extension Build | API320151012,NTS |
| PHP Extension Build | API20151012,NTS |
| Debug Build | no |
| Thread Safety | disabled |
| Zend Signal Handling | disabled |
| Zend Memory Manager | enabled |
| Zend Multibyte Support | provided by mbstring |
| IPv6 Support | enabled |
| DTrace Support | enabled |
| Registered PHP Streams | https, ftps, compress.zlib, php, file, glob, data, http, ftp, phar |
| Registered Stream Socket Transports | tcp, udp, unix, udg, ssl, tls, tlsv1.0, tlsv1.1, tlsv1.2 |
| Registered Stream Filters | zlib.\*, string.rot13, string.toupper, string.tolower, string.strip\_tags, convert.\*, consumed, dechunk, convert.iconv.\*, mcrypt.\*, mdecrypt.\* |
|  | |

##### **Configuration** **apache2handler**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Apache Version | | Apache/2.4.18 (Ubuntu) | | |
| Apache API Version | | 20120211 | | |
| Server Administrator | | [booktrade17@gmail.com](mailto:booktrade17@gmail.com) | | |
| Hostname:Port | | 127.0.1.1:80 | | |
| User/Group | | www-data(33)/33 | | |
| Max Requests | | Per Child: 0 - Keep Alive: on - Max Per Connection: 100 | | |
| Timeouts | | Connection: 30 - Keep-Alive: 5 | | |
| Virtual Server | | Yes | | |
| Loaded Modules | | core mod\_so mod\_watchdog http\_core mod\_log\_config mod\_logio mod\_version mod\_unixd mod\_access\_compat mod\_alias mod\_auth\_basic mod\_authn\_core mod\_authn\_file mod\_authz\_core mod\_authz\_host mod\_authz\_user mod\_autoindex mod\_deflate mod\_dir mod\_env mod\_filter mod\_mime prefork mod\_negotiation mod\_php7 mod\_setenvif mod\_status | | |
| Directive | Local Value | | Master Value |
| Engine | 1 | | 1 |
| last\_modified | 0 | | 0 |
| Xbithack | 0 | | 0 |

##### 

##### **Apache Environment**

|  |  |
| --- | --- |
| Variable | Value |
| HTTP\_HOST | booktrade.duckdns.org |
| HTTP\_USER\_AGENT | Mozilla/5.0 (X11; Ubuntu; Linux x86\_64; rv:53.0) Gecko/20100101 Firefox/53.0 |
| HTTP\_ACCEPT | text/html,application/xhtml+xml,application/xml;q=0.9,\*/\*;q=0.8 |
| HTTP\_ACCEPT\_LANGUAGE | en-US,en;q=0.5 |
| HTTP\_ACCEPT\_ENCODING | gzip, deflate |
| HTTP\_REFERER | http://booktrade.duckdns.org/ |
| HTTP\_DNT | 1 |
| HTTP\_CONNECTION | keep-alive |
| HTTP\_UPGRADE\_INSECURE\_REQUESTS | 1 |
| PATH | /usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin |
| SERVER\_SIGNATURE | <address>Apache/2.4.18 (Ubuntu) Server at booktrade.duckdns.org Port 80</address> |
| SERVER\_SOFTWARE | Apache/2.4.18 (Ubuntu) |
| SERVER\_NAME | booktrade.duckdns.org |
| SERVER\_ADDR | 192.168.0.11 |
| SERVER\_PORT | 80 |
| REMOTE\_ADDR | 116.240.144.21 |
| DOCUMENT\_ROOT | /home/acp/booktrade/public\_html |
| REQUEST\_SCHEME | http |
| CONTEXT\_PREFIX | *no value* |
| CONTEXT\_DOCUMENT\_ROOT | /home/acp/booktrade/public\_html |
| SERVER\_ADMIN | webmaster@localhost |
| SCRIPT\_FILENAME | /home/acp/booktrade/public\_html/data/php/test.php |
| REMOTE\_PORT | 51420 |
| GATEWAY\_INTERFACE | CGI/1.1 |
| SERVER\_PROTOCOL | HTTP/1.1 |
| REQUEST\_METHOD | GET |
| QUERY\_STRING | *no value* |
| REQUEST\_URI | /data/php/test.php |
| SCRIPT\_NAME | /data/php/test.php |

##### 

##### **HTTP Headers Information**

|  |  |
| --- | --- |
| HTTP Request Headers | |
| HTTP Request | GET /data/php/test.php HTTP/1.1 |
| Host | booktrade.duckdns.org |
| User-Agent | Mozilla/5.0 (X11; Ubuntu; Linux x86\_64; rv:53.0) Gecko/20100101 Firefox/53.0 |
| Accept | text/html,application/xhtml+xml,application/xml;q=0.9,\*/\*;q=0.8 |
| Accept-Language | en-US,en;q=0.5 |
| Accept-Encoding | gzip, deflate |
| Referer | http://booktrade.duckdns.org/ |
| DNT | 1 |
| Connection | keep-alive |
| Upgrade-Insecure-Requests | 1 |
| HTTP Response Headers | |

##### 

##### **calendar**

|  |  |
| --- | --- |
| Calendar support | enabled |

##### 

##### **Core**

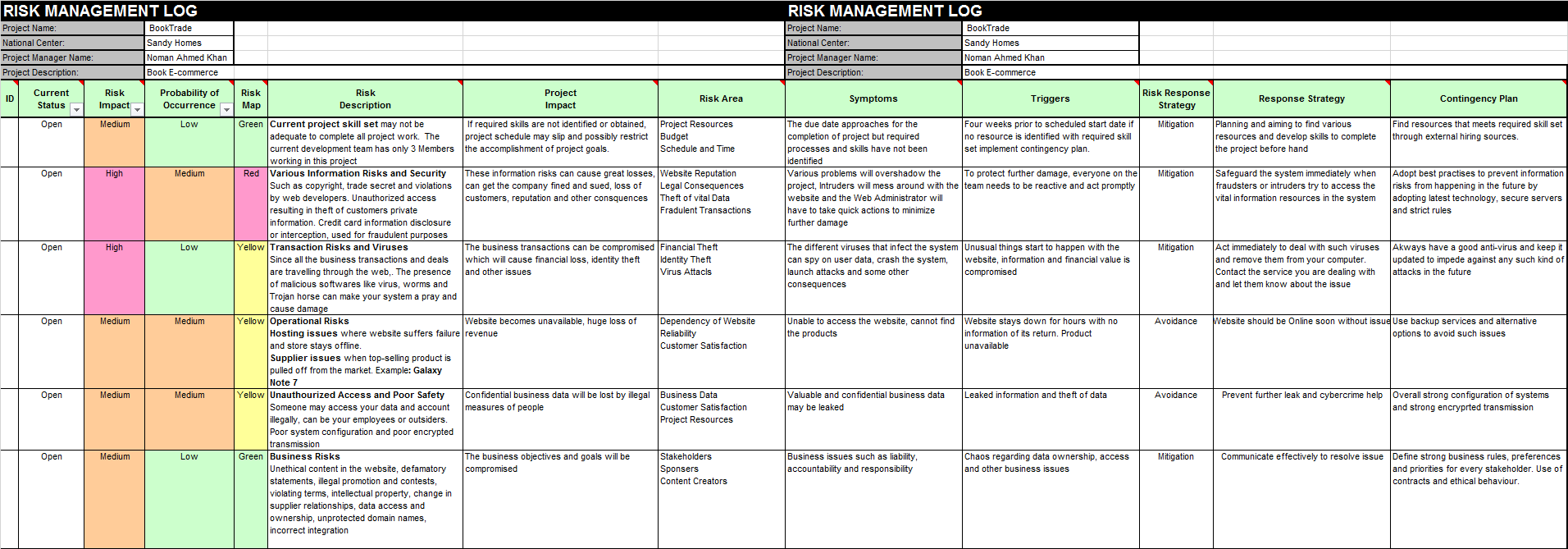
|  |  |
| --- | --- |
| PHP Version | 7.0.15-0ubuntu0.16.04.4 |

Revised Risk Register

The image below outlines the detailed risk management plant. Due to difficulty of embedding the excel file within the word document, the image had to be posted. Upon zooming in, the text can be read easily.

Risk register is an integral component in a risk management process and when this topic is understood properly it helps us to identify and correctly determine the risk management process with clear understanding. The major processes the members of the project went through making this risk register are outlined below:

1. **Qualitative Risk Analysis**
2. **Quantitative Risk Analysis**
3. **Risk Response Planning**
4. **Monitoring and Controlling Risks**



The detailed excel worksheet has been simplified for easy understanding:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Risk Description | Project Impact | Risk Areas | Triggers and Symptoms | Response and Contingency |
| Inadequate Skills in Project | Inability to achieve project goals | Budget, Resources, Schedule, and Time | Lack of skills and resources | Aiming and planning to find accurate skills and resources |
| Transaction Risks and Malwares | Compromised business Transactions | Identity theft and Virus Attacks | Personal Information Compromised | Prompt Action and Strict Security Rules |
| Operational Issues | Revenue loss and Down Time | Reliability and customer satisfaction decreases | Unavailable Website | Backup services and quick recovery |
| Poor Security | Business data breached | Vulnerable Information leakage | Confidential document and data are not safe | React quickly and Strong authenticated and encrypted system configuration |
| Ethical Issues | Compromised Business Objectives | Stakeholders, Customers not happy, | Liability, accountability and responsibility in question, business issues | Strong business rules, ethical behavior and effective communication required |

BOOK.TRADE   
VERSION 1.0

MASTER TEST PLAN

Version 2.0

Draft

May 2017

Parmar Aman C.

Senior Developer

Test Manager

*Test Plan Identifier*

*Book.trade Version 1.0 MTP 0.2*

*Introduction*

Book.trade is a book trading platform focused on physical books. Book.trade is for book lovers and students who likes to read physical books in this digital era. This project’s basic goal is to provide trading website for books called Book.trade and can be enhanced in future. Target audiences are people who wants books in physical form.

This version 1.0 features user registration and login. It works on Intel based machine running apache2 and MySql server on Linux operating system. These set up is for small scale implement and will be improved in near future.

Test phases are shown chronologically here:

* Integration level and unit testing – applying standard rules for coding and successful and efficient communication between units. (Unit and Integration Test)
* Assurance of quality of code – Code and Integration level tested to system level by successful repetition of small subsets of tests.
* System tests – Checking compatibility, usability, flexibility, performance of system.
* Assurance and acceptation of quality and product through proposal to production. (User Acceptance Test)
* Implementation.

*Test Items*

This testing process will use require:

* Book.trade software version 1.0 and supporting infrastructure.
* Client platforms or end devices.

This test process will not require nor include:

* Book.trade documentation e.g: Manual , Specification guide.
* Any legacy system that Book.trade integrates with except for interface.
* Supporting services like post and trading times.

*Test targets*

Following aspects will be tested in the process and each phase may or may not include all of them.

* Functional aspects of website
* Logical aspects of website
* Core Critical Risky Difficult (CCRD) cases
* Security
* Data Consistency
* Connectivity
* Successful Script Execution (in this particular project PHP, HTML, CSS, and MySQL)
* Secure Payment Connection (booktrade payment system not the third party system.)
* Accurate and Consistent Billing System (Charging process of payment e.g: sending bill)
* Accurate and Consistence Transaction Report (Receipt for transaction)
* Successful Payment Rollback (Reverting back to normal state without any trace of payment data in case of payment termination or cancellation )
* Accessibility (Ease of use of interface)
* Usability (Proper GUI to adopt by user e.g: no misleading or overlapping graphics)
* Working Back-end Architecture (Database, Server and end device should be in sync while session is running, without disruption)

*Functions Not to Test*

It is a pure intention to test every cases possible but it can vary due to following cases:

* Short on time for testing
* Short on resources for testing
* feature or functionality gets canceled in later phases.
* Too minor importance of case to test

*Approach*

Test cases will be prioritized according to risk level and will be processed accordingly. High risk cases will be tested first. There will be some exception lie:

* Low priority cases can be processed in groups while using minimum resources.
* Scheduling problems.
* Some low priority test are per-requisite for another high priority tests that might get dropped later.

**Note**: Source codes will be locked during testing times and won’t be updated unless there are any critical and major updates pending.

*Result Criteria*

There will be criteria before test to fulfill and then tests will be processed with the system administrator’s permission.

*Suspension Criteria*

Test will only stop if test subject is not available or connection disruption; here Book.trade website and tests will be resumed once the connection is reestablished.

*Test Documents*

Following docs will be generated for tests.

* Master Test Plan. (This document).
* Unit and system integration test plan and scripts.
* User acceptance test plan scripts.
* Test result report for above scripts.

Note: Daily incident reports will be generated and stored for some periods to give development team heads for any potential threats. In addition to that it will help team to recognize future features and functions as well as unnecessary features. That said, it will also introduce new limitation of proposed tests with new features to improve current processes.

All the documents will be provided in open document text (.odt) format.

*Remaining processes*

After the implementation of the system or website it will be considered that all the per-implementations  test are done and only post-implementation tests remains.

*Test Environments*

There will be primarily two parts of test environments which are: Client side environment that will be accessed by users and Book.trade server side environment that will be accessed by development and tech teams.

**Client Side Environments:**

* A high-end device – This includes a full featured personal computer, laptop, mobile, tablet, or phablet with maximum resources available in market.
* A low-end device – This includes personal computer, laptop, mobile, tablet, or phablet less with bare minimum specs to run program,
* Legacy devices – This includes such devices that are no more in use or discontinued but still being operated by Book.trade.

**Note:** All devices have different operating systems but since our product is a website which is accessed by a browser it doesn’t affect out test processes. Instead it will include all different kinds of browsers.

**Note:**Since there are lots of browsers available in market, its impossible to test each one of them. But there is a list of some chosen browser and all available version will be tested. These are the browsers that are available for client side environment but not limited:

* Lynx -Text only browser
* Internet Explorer – Browser by Microsoft
* Mosaic- A really old legacy browser
* Google Chrome – Browser by Google
* Safari – IOS web browser
* UC Browser -  Browser by Chinese Mobile Internet Company UCWeb
* Mozilla Firefox – Open Source Web Browser by Mozilla Foundation
* Opera – Browser by Opera Software (based on blink on layout version not the Presto layout one)
* Edge – Successor of Internet Explorer released by Microsoft in 2015.
* Android Browser – Open source browser introduced in earlier Android devices.

**Note:** The default installation configuration will be used, no additional add ons will be installed nor plug-ins.

*Server Side Environment:*

There is one server for production website and another exact same replica of it for testing and debugging. The replica will be used for all kind testing of units and integrity.

Even though that replica will be same as the production version it may differ in hardware set up and resources due to budget limits and goals.

*Available tools:*

* Site analysis – from site-see.com
* Website overview in detail- from nibbler.silktide.com
* W3C Link validation service – w3c.org
* W3C CSS validation service – w3c.org
* WAVE – website accessibility checker wave.webaim.org
* Performance test- from www.webpagetest.org

*Responsibilities*

* Noman Ahmed Khan, Sandesh Raja Thapa, Aman C. Parmar – Project sponsors, Final acceptance of the Web site and Legal matters.
* Noman Ahmed Khan(Project Manager)– Decisions requiring senior management approval.
* Aman C. Parmar(Senior Developer) – System and post implementation testing, Unit and integration testing and Configuration management.
* Sandesh Raja Thapa(Business Analyst) – Quality Assurance, International testing and Business operational matters.

*Risks*

Following scenarios presents the most likely contingencies:

* Website is unavailable for testing.
* Testing software is not available for testing.
* Testing people are not available.
* Large number of defects make it impossible to initiate tests.
* Not enough time to complete all the tests.

*Functions tests*

*Accessibility*

* The web page should have zero errors in WAVE check.

*Compatibility*

* The web page should render correctly on all above mentioned browsers.
* The web page text is clearly readable when printed in A4 size paper.
* The web page is clearly understandable when accessed through text only browser; Lynx.

*Coding Standards*

* All the server side code is written in PHP and HTML language with proper coding standards.
* All database server coding should be done in MySql database server.

*Content*

* Book.trade copyright information should be at the bottom of the website.
* All the spells are checked with US English spell checker.
* There should be no use of hidden text except for passwords.

*Functional*

* All sessions should be alive and not to be broken in any cases, especially for transactions.
* Pressing back and forward buttons on browser while the session is still in process.
* Clear cache during active session.
* Using history to check previous pages.
* Pressing reload button.
* Aborting session in mid way of process.
* Staying inactive for long breaks. More than 30 min.
* Changing the clock time of client side device.
* Opening same session on two different browsers or in two different instances of browser.

**Checking that form data on website is not lost during session in scenarios shown below:**

* User presses back or forward button in mid data entry of form.
* User goes to a previous visited page through history.
* User presses the book mark button.
* Checking different scenarios by
* Disabling cookie
* Activating Cookie
* Enable or disable cookie in mid session.

**All the client side validations are checked again at server side validation in case of:**

* User disabled scripts
* Browser defects

**Resources are used properly by:**

* Releasing memory when user session is expired.
* Releasing memory when user session is complete.
* Network connection is completely closed after successful transaction.
* Dates and times should be checked and kept in sync with server side clock time not the client side clock.
* Website should show all the T&C and disclosures
* Book.trade meets all the claims made on website.

*Legal*

* No copyrighted content is on the website unless it is with written permission from owner.
* Meta tags does not include other copyrighted phrases or words.

*Marketing*

* Meta tags are used on all public pages.
* Search engine query problems are to be checked by Scrubby.
* If ads exists, they don’t block users view or make webpage unusable.

*Navigation*

* All links should be working fine.
* External links opens in new browser instance.
* Internal links links should use existing browser instance.
* The website must be bookmarkable.
* Book mark should start from “Book.trade –”
* There should be no orphan files or pages that can’t be reached by home page.
* Links should not re-redirect itself in mid way.
* All pages can be browsed casually no need for site maps or search engine queries.
* Correct broken link reporting feature available.

*Performance*

* In webpagetest.org report the links should be accessible in less than 1 second.
* Redirecting should not take too long that it loads after session finishes.
* All webpages are accessible in seconds shouldn’t take more than 1 second to load.

*Reliability*

* Website should be able to handle worst case scenario for workload.
* When server restarts abnormally all the sessions and transactions are rolled back cleanly.
* Website should run for long periods of time without any distortion in resources.
* Recovery time should not exceed the defined time limit.
* System can handle unexpected server failure.
* Server should be able to work with minimum resources available.

*Security*

* Checks that user have chosen strong user id and password.
* User is locked for 30 min after multiple failed attempts at login.
* User is logged out after inactive period of 30 min.
* Client side changes won’t affect any security rules set by server.
* All the data sent and received is secured and encrypted.
* Checks that any field doesn’t have obvious input like putting user as user name or password and password.
* A separate database for any id verification or alternative login recovery data.
* Make sure that system is not connected to outside world directly.

*Usability*

* Website should go to minimum feature view without compromising clear and understandable layout.
* Layout is consistent for user to remember.
* All the requirements are clearly displayed on client screen with proper margin.

*Audit*

* All logs are verified for client’s and advertiser’s use.
* All management procedures are verified every month.
* Every year archive is stored and junk is purged regularly.

­­­­­

Book.trade  
Version 1.0

user acceptance test

functional test plan

Version 1.0

### Draft

May 2017

Parmar Aman C.

Senior Developer

Test Manager

## Use Case : Register

Actors: User

Priority: High

**Expected Input:**

**Field Username:**

* String doesn’t start with a space or a number.
* Should be 4 to 12 character long.
* Should be unique.

**Field Email:**

* Valid email type.
* Should be used only once per registration.

**Field Password:**

* Alphanumeric string with at least one symbol, one capital letter and one small letter.
* Should be 6 to 12 character long

**Expected Result:**

Successful registration.

**Invalid Input:**

**Field Username:**

* Username starts with a space or a number.
* Username is shorter than 4 characters.
* Username is longer than 12 characters.
* Username is already used.

**Field Email:**

* Invalid email type.
* Email is already used.

**Field Password:**

* Password is not Alphanumeric and doesn’t fulfill other requirements.
* Password is longer than 12 characters or shorter than 6 character.

**Expected Result:**

System displays appropriate error message and asks for resubmission.

## Use Case : Login

Actors: User, Administrators

Priority: High

**Expected Input:**

**Field Username:**

* Username should match the registered username.

**Field Password:**

* Password should match the registered password.

**Expected Result:**

Successful login.

**Invalid Input:**

**Field Username:**

* Username doesn’t match in the database.

**Field Password:**

* Password doesn’t match in the database.

**Expected Result:**

System asks for resubmission of credentials until 3 times. After 3 attempts user will locked for one hour

## Use Case : Add Books

Actors: User

Priority: High

**Expected Input:**

**Field Book Name:**

* String containing book name
* Can’t be empty.

**Field Book Author:**

* Author name
* Can be empty.

**Field Book ISBN:**

* only contains number.
* Should be valid ISBN number.
* ISBN will be checked on other sources and should match with the stated book.
* Can be empty.

**Field Book Condition:**

* Explanation should be only 1000 characters long.
* Can’t be empty.

**Expected Result:**

Book will be added to user account.

**Invalid Input:**

**Field Book Name:**

* Field is empty.

**Field Book Author:**

* Field is empty.

**Field Book ISBN:**

* ISBN doesn’t match any database.
* ISBN is not valid format

**Field Book Condition:**

* Field is empty.

**Expected Result:**

Error message will be displayed and user will be asked for resubmission.

## 

## Use Case : Managing Books

Actors: User

Priority: High

#### For Adding Book:

**Expected Input:**

**Book:**

* Book should be unique and different than other books of same user.
* Could be a different edition of existing book.
* ISBN should be unique

Field test is same as Adding Books use case.

**Expected Result:**

Book is added to user account.

#### For Deleting Books:

**Expected Input:**

**Book:**

* User selects all the books he wants to delete and confirms the decision.

**Expected Result:**

System hides books for 2 weeks. User can reverse the process in 2 weeks after that books will be deleted from database permanently.

## Use Case : Buy Books

Actors: User

Priority: High

**Expected Input:**

**User**

* User browses the books.
* Sends Proposal for book to other user.

If other user accepts the proposal

* User pays for the book.
* User provides shipping details.
* Other user sends the book after receiving payment.
* User receives the book.

**Expected Result:**

* User successfully makes the payment.
* User receives the Book.

**Invalid Input:**

**User:**

* User closes browser in mid process of buying, paying or browsing.
* Network Issues in mid session of buying books.

**Expected Result:**

System rollbacks to previous state without any traces of last process. Cancels the last process payment.

## Use Case : Trading Books

Actors: User

Priority: High

**Expected Input:**

**User:**

* User selects the books he wants to trade.
* Sends the proposals to the other user with whom user wants to trade.
* If other user accepts the proposal both party will exchange the shipping details and sends the book to each other.

If both user are with safe account:

* Both user will send the books to booktrade system.
* System will send the book to the users.

If one user is with safe account:

* The other user who doesn’t have the safe account will send his book to the system first.
* System will then notify the other user that it’s safe to send the book.
* Then user with safe account sends the book to the other user directly.
* And system will send the book from other user to the user with safe account.

**Expected Result:**

User with safe account successfully trades the book.

**Invalid Input:**

**User:**

* other user doesn’t send the book to the system

**Expected Result:**

If both user are with normal account they trade books with their own accords.

User is notified about the possible fraud and advised not to trade the book and leave the feedback for that user.

## Use Case : Selling Books

Actors: User

Priority: High

**Expected Input:**

**User:**

* User verifies account with ID proof (Optional)
* User selects the books that needs to be sold and system puts them on sell display.

**Expected Result:**

Successful display on sell list of books.

**Alternate Input:**

**User:**

* User removes books from sell display.

**Expected result:**

Book or books are removed from sell display and other users can’t see them.

## Use Case : Payment through online service

Actors: User

Priority: High

**Expected Input:**

**Field Username:**

* User is charged with proper bill of item or service.
* User uses PayPal or other online methods to make payments.

**Expected Result:**

User makes successful transaction.

**Invalid Input:**

**User:**

* Closes session in mid process.
* Network issue in payment process.
* Browser tab is closed.
* Transaction is canceled by user or system.
* Session time out occurs.
* Device failure or any other interruption in mid process.

**Expected Result:**

System successfully rollbacks the transaction without any traces.

## Use Case : Payment through card

Actors: User

Priority: High

**Expected Input:**

**Field Username:**

* User is charged with proper bill of item or service.
* User uses card details to make payments.

**Expected Result:**

User makes successful transaction.

**Invalid Input:**

**User:**

* Gives invalid card information.
* Card doesn’t have enough balance to make transaction.

**Expected Result:**

System successfully rollbacks the transaction without any traces.

Book.trade  
Version 1.0

CCRD test plan

Version 1.0

### Draft

May 2017

Parmar Aman C.

Senior Developer

Test Manager

## List of Processes that are Core, Critical, Risky & Difficult for the chosen CCRD use cases

* **Security**
* **Data Consistency**
* **Connectivity**
* **Script Execution**
* **Secure Payment Connection**
* **Accurate and Consistent Billing System**
* **Payment Rollback**
* **Accurate Transaction Report**
* **Accessibility**
* **Usability**
* **System and Server Sync**

### Security

**Expected Behavior:**

* Data send to server is secured.
* Data stored on database is secured.
* All the communication is secured and encrypted.
* All connection to server and server to database are secured.

### Data Consistency

**Expected Behavior:**

* Data format shouldn’t change when transferring from one place to another.
* All the form data should reach server without data leaks or changes in form.

### Connectivity

**Expected Behavior:**

* Connections should be secured.
* Connections shouldn’t get destroyed in mid session of user.
* Connections should follow their time limit policy.
* When terminated connections should remove all the traces of connection.

### Script Execution

**Expected Behavior:**

* All script are checked and debugged.
* Script maintain their language standards.
* Script should be runnable with expected output.

### Secure Payment Connection:

**Expected Behavior:**

* Online payment method are assumed be secured by the third party service provider as their statements.
* Card payment method accepted by system should open a secure session and connection.
* Payment data shouldn’t change or get leaked.
* Payment connection should follow their time limit policy.

### Accurate and Consistent Billing System

**Expected Behavior:**

* User should be charge accurately.
* Billing information shouldn’t change automatically.

### Payment Rollback

**Expected Behavior:**

* If payment session is terminated or disrupted in process, payment should rollback safely without any charges to user.
* No payment data should left are rollback.

### Accurate Transaction Report

**Expected Behavior:**

* Payment receipts should show accurate information.
* There should no exception for any cases where receipt is not giver to user.
* All receipts should show time and date of user’s location.

### Accessibility

**Expected Behavior:**

* Website should be available all the stated time.
* Contact methods should be online all the time.
* Website response time should as low as possible.
* Website should be operable by user with special needs.
* Website should contain options that increases the contrast of the page elements, which makes it easier to see.
* Website should make proper sound effect and show confirmation dialogs prior to any action.

### Usability

**Expected Behavior:**

* Website should be easy to use, it shouldn’t contain confusing or misleading elements.
* Website should respond to hardware changes like keyboard input and mouse click.
* Should be usable with same features on other devices like phone and tablet.
* Elements displayed on screen shouldn’t interfere with each other.

### System and Server Sync

**Expected Behavior:**

* Server and User session should be in sync all the time until session finishes.
* System should use server’s time and date for any operation.
* System shouldn’t do anything that is not requested by server.
* Server should always stay connected to the running user session and if disconnected it should reset session.

Book.trade  
Version 1.0

User acceptance test

Feedback plan

Version 1.0

### Draft

May 2017

Parmar Aman C.

Senior Developer

Test Manager

# User on Desktop or Laptop Device

### User TO-DO:

* Register with dummy credentials.
* Login with those credentials.
* Browse around the site with logged in account.
* Search for books.
* Add dummy books in account.
* Modify the book list.
* Delete books and reverse the process.
* Send proposal to trade books.
* Send proposal to buy books.
* Purchase dummy book.
* Sell the dummy books.
* Make payments with dummy accounts.
* Communicate with other user.

### User Feedback:

* Take feedback for User interface design.
* Take feedback for system process integrity and process response time.
* Take feedback for system usability.
* Take feedback for system ease of use.
* Take feedback for complaints.

Revised Project Plan

### *Introduction*

The project plan is of an E-commerce website BookTrade that serves as a platform for students to buy/sell trade and donate new or used books. This project plan outlines the organization of the project, different project practices and measurements, how the software is deployed and also showcases the project milestones and objectives in the sections below.

### *Project organization*

The project team comprises of three members and there are roles assigned to them. However, all three of them have collaborated and tried to help each other in different areas of their interest so that it feels like everybody is trying their best to make the project as informative and complete as possible.

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Student ID** | **Role** | **Email** |
| Noman Ahmed Khan | 11567969 | Project Manager, Programmer | [Noman-ahmed-khan@live.com](mailto:Noman-ahmed-khan@live.com) |
| Sandesh Raja Thapa | 11575549 | Project Researcher, Analyst | [Sandes.rtz96@gmail.com](mailto:Sandes.rtz96@gmail.com) |
| Aman Kumar Chandubhai Parmar | 11569794 | Tester, debugger. | [Aparmar62@gmail.com](mailto:Aparmar62@gmail.com) |

### *Project practices and measurements*

The project throughout makes use of the Unified Process Model Strategy. Practices in UP Model consists of four major phases which comprises of:

1. **Inception**:

The major goal of Inception is to create and establish the case of the system that we are trying to propose

1. **Elaboration**:

The major goal of Elaboration is to establish the ability to build the proposed system by carefully reviewing the financial constraints, time issues and various other types of constraints that a project faces

1. **Construction**

This process is all about building a system that is efficient and capable of operating successfully in various environments that a customer is expected to use it in.

1. **Transition**:

The process of rolling out a fully functional system to the people that need to use it or in simple words, customers.

### *Deployment*

Once we’ve created the website, the website will be deployed through the Apache 2 Server that we are using on Linux Operating System. For the updates, the update is to be made through the website and then we will be given the rights to get into the server and make any changes or updates required.

### *Project milestones and objectives*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Subject** | **Phase** | **Iteration** | **Dates** | **Primary objectives** (risks and use case scenarios) |
| **BookTrade Project** | Inception Phase | I-1 | 14/03 – 27/03 | Complete Version 1.0 of the Vision Document  Establish the Requirements UML Diagram   * Use Case Diagram * Domain Model Diagram   Set out the Architecture for the Project which compliments the non-functional requirements. |
| I-2 | 28/03 – 10/04 | Establish Risk Register  Create a Master Test Plan  Draft Version of Project Plan  Deliver Life Cycle Objectives Milestone (LCOM)  The Inception Phase Is Now Complete. |
| Elaboration Phase | E-1 | 11/04 – 24/04  (Session Break) | Mitigation of **First** Highest Priority Risks, and Implementing of Architecture to Support CCRD Use Case. Define Tests for the CCRD Use Case. |
| E-2 | 25/4 – 8/05 | Mitigation of **Second** Highest Priority Risks, and Implementing of Architecture to Support CCRD Use Case. Define Tests for the CCRD Use Case. |
| E-3 | 9/05 – 22/05 | Mitigation of **Third** Highest Priority Risks, and Implementing of Architecture to Support CCRD Use Case. Define Tests for the CCRD Use Case.  Deployment of the Executable Architecture in Trial Environment  Complete Internal User Acceptance Testing for CCRD Use Case in Trial Environment |
| E-4 | 23/05 – 3/06 | Contingency  Deliver Life Cycle Architecture Milestone (LCAM)  Completion of Elaboration Phase Project Assessment |
| Mid-year Semester Break | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Mid-year Semester Break | | | | |
| **BookTrade Project** | Construction Phase | C-1 | 11/07 – 24/07 | Completion of the First Set of Functional Requirements along with tests against the CCRD Use Cases. |
| C-2 | 25/07 – 7/08 | Completion of the Second Set of Functional Requirements along with tests against the CCRD Use Cases. |
| C-3 | 8/0 – 21/08 | Completion of the Third Set of Functional Requirements along with tests against the CCRD Use Cases. |
| C-4 | 22/08 – 4/09  (Session Break) | Contingency  Deliver Initial Operation Capability Milestone (IOCM)  Complete Construction Phase Project Assessment |
| Transition Phase | T-1 | 5/09 – 18/09 | Deployment of Application  Testing Each Part of Application against the Use Cases  Resolve any known errors |
| T-2 | 19/09 – 2/10 | Testing Each part of application against Use Cases  Resolve any known errors. |
| T-3 | 3/10 – 14/10 | Contingency  Deliver Product Release Milestone (PRM)  Complete Final Project Assessment |

Elaboration Phase Assessment

Introduction:

Whenever the elaboration phase starts, it signifies the end of the Inception phase and the completion of project funding. The various task descriptions related to the project management has been briefly outlined and described for the LCAM assessment.

|  |  |
| --- | --- |
| **Project Management:** | **Workflow Details** |
| Monitoring/Managing | Representation of the activities that are being carried in the project management, majorly the details of the workflow.   * Iteration Management * Controlling and Monitoring the Project   Since this project isn’t on a real big scale, the Project Manager doubles as the Test Manager and their workflow includes:  Acceptable Mission Achievement (workflow activities: [Reinforce, Access and Advocate Quality](http://www.cis.umassd.edu/%7Ejbergandy/process/activity/ac_tst_assadvqlt.htm), [Improve and Improvise Test Effort](http://www.cis.umassd.edu/%7Ejbergandy/process/activity/ac_tst_asststefr.htm))  The major document outputs are:   * Records of Review * [Iteration Assessment](http://www.cis.umassd.edu/%7Ejbergandy/process/artifact/ar_itass.htm) * [Status Assessment](http://www.cis.umassd.edu/%7Ejbergandy/process/artifact/ar_stass.htm) |
| Next Iteration Planning | This includes workflow details:   * [Next Iteration Planning](http://www.cis.umassd.edu/%7Ejbergandy/process/workflow/manageme/wfs_plan.htm) * [Developing Software Design Plan](http://www.cis.umassd.edu/%7Ejbergandy/process/workflow/manageme/wfs_sdp.htm) (Some scope change can be expected) * [Evaluation Mission](http://www.cis.umassd.edu/%7Ejbergandy/process/workflow/test/wfs_dfnevlmsn.htm) Definition (Activities related to Test Manager)   The major outputs obtained are:   * [Iteration Plan](http://www.cis.umassd.edu/%7Ejbergandy/process/artifact/ar_itpln.htm) – The overall second assignment has been updated based and relied upon the expected new functionalities during the upcoming iterations, improvising on the maturity of the product, mistakes realized, lessons learned from the previous assignment and trying to mitigate as many risks as possible. * [Software Design Plan](http://www.cis.umassd.edu/%7Ejbergandy/process/artifact/ar_sdp.htm) – Various plans under the software design are updated to reflect the latest requirements, changing scopes and risk issues. The Risk list has been revised to provide a simple description but covers all major risks. Additional risks have been identified and analysed * [Test Plan](http://www.cis.umassd.edu/%7Ejbergandy/process/artifact/ar_tstpl.htm) – Test Manager has done a significant work to reflect and update the mission test plan to add upon the testing for next iteration.   The various results obtained from the above processes, various iteration and status assessments are crucial in determining what change process will be required and the tools that will need to be updated if needed |
| **Requirements** |  |
| BookTrade Website Prototype Interface | This includes the BookTrade Website Design:  The user-interface of the Website BookTrade  The major output product is the BookTrade Website Design |
| Changing Requirements Management | Different discovery of various requirements and their improvisation and refinement goes on in elaboration.  The identified and required workflow detail is designated as: Changing Requirements Management |
| Refine the System Definition   * Set 1 Requirements * Set 2 Requirements * and it goes on | This includes the workflow detail [Refine the System Definition](http://www.cis.umassd.edu/%7Ejbergandy/process/workflow/requirem/wfs_refs.htm) (except for user-interface related activities, which are a separate task).  The major outputs obtained are:   * a [Use Case Model](http://www.cis.umassd.edu/%7Ejbergandy/process/artifact/ar_ucmod.htm) Requirements (Highest and prioritized detailing of the use cases) * Refined [Vision](http://www.cis.umassd.edu/%7Ejbergandy/process/artifact/ar_vsion.htm) and [Non-Functional Requirements and Specifications](http://www.cis.umassd.edu/%7Ejbergandy/process/artifact/ar_sspec.htm).   Requirements have been divided into various smaller sets to ease the processes as Ser 1 Requirements related to a topic and Set 2 Requirements related to some other topics. Organization of tasks are places in such a way that the use cases reflect the accurate scenario, composition and the detailed non-functional requirements. Specific issues and risks have been investigated. |
| **Definitive Architecture** |  |
| Candidate Architecture Definition | The workflow outlined is [Candidate Architecture Definition](http://www.cis.umassd.edu/%7Ejbergandy/process/workflow/ana_desi/wfs_and1.htm) |
| Structured Implementation Model | The workflow outlined is [Structure the Implementation Model](http://www.cis.umassd.edu/%7Ejbergandy/process/workflow/implemen/wfs_stim.htm). |
| Architecture Refine Process | The workflow outlined is Architecture Refine Process |
| **Support for Development** |  |
| Testing and Integration Support | Management and maintenance of the build environment, prioritization and selection of the builds test carries on.  The workflows outlined are   * [Build Stability](http://www.cis.umassd.edu/%7Ejbergandy/process/workflow/test/wfs_vldbldstb.htm) Validation * Baselines and Releases Management |
| Various Defect Fixings | Our Test Manager tries to fix the bad codes efficiently and effectively in a task that is difficult and ongoing and various development of features and components take place. |
| Components and Database Design and Development Components/Features   * Feature/Component 1 * Feature/Component 2 * and it goes on | A feature covers various tasks included in it and thus implements a scenario. Hence, the workflow details are outlined.   * [Components and Database Design and Development](http://www.cis.umassd.edu/%7Ejbergandy/process/workflow/ana_desi/wfs_and4.htm) * [Component](http://www.cis.umassd.edu/%7Ejbergandy/process/workflow/implemen/wfs_implc.htm) Implementation * [System Integration](http://www.cis.umassd.edu/%7Ejbergandy/process/workflow/implemen/wfs_insy.htm) * Evaluation and Testing * [Test Assets Improvement and Refinement](http://www.cis.umassd.edu/%7Ejbergandy/process/workflow/test/wfs_imptstast.htm)   Various other tasks are included as features, scenarios, or use-cases.  Various long and tedious tasks that consume times are divided as subtasks. For instance, into:   * Design – Prototyping may be included * Implement – Testing can be considered * Testing and Integration – Fixing various bugs in the codes can be included during Testing and Integration. |

The Gantt Chart Diagram has been presented below to represent our LCAM scenario performed by our various group members:

