

Title: Bookish Dine

Author: Mahmudur Rahman (00165683)

Centre: Daffodil International Academy

Table of Contents

| | |
|-------------------------------------|----|
| Abstract:..... | 3 |
| Acknowledgement:..... | 3 |
| Introduction..... | 4 |
| Analysis..... | 7 |
| Requirements..... | 7 |
| Functional Requirements | 7 |
| Non-Functional Requirements..... | 10 |
| Use Case..... | 10 |
| System Architecture..... | 11 |
| Design | 12 |
| Structural Model | 13 |
| Behavioral Model..... | 13 |
| Implementation | 15 |
| Choice of Programing Language | 15 |
| System Cutover | 16 |
| Data Migration..... | 18 |
| Training..... | 18 |
| Other Project Issues | 19 |
| Project Management..... | 19 |
| Risk Management | 19 |
| Configuration Management | 20 |
| Testing | 20 |
| Conclusion..... | 26 |
| Evolution of Work | 26 |
| Further Development of System | 27 |
| References..... | 28 |
| Appendices..... | 29 |
| Requirements Catalogue | 29 |
| Use Case Description..... | 31 |
| Test Scripts | 31 |
| System Code..... | 34 |

Abstract:

This project is about online book selling store. The main purpose of this project is people will be able to buy books from the website. And beside this user those who will register for the site will get some extra feature in the website.

Outcome of the system will visualize a website with multiple feature. And that site there will be three roles to access the site. Normal user will be able to visit the site. User will be able to visit book store and search feature to search books from the store. User can add book to the cart. After completing adding cart giving their own details user will confirm order placement. User will authenticate themselves by filling registration form and by logging with email and password can get access to story feature. These users will able to write, read and comment on story. And other one is admin role where admin will maintain the whole system. Admin's role is entry book, update books, delete books and see orders submitted by user.

Acknowledgement:

At the beginning I would thanks to Almighty Allah to give me strength to do the project and keep me healthy to the entire project duration. Secondly, I would like to thanks NCC Education for giving me this opportunity to do the project which teach me a lot of things from this project. Like how to brainstorm, solve problem, find solution and implement it. And the biggest thing I learn how to do a whole project becoming a project manager, coder and tester. I would thanks to my project supervisor to give so much motivation and courage to complete the project. I would also five thanks to my PHP course teacher for guiding me if I stuck on any error. He always makes me think clearly to solve the problem. Thanks all of them for helping and supporting me at the time of developing the whole project.

Introduction

Bookish Dine is an online book selling store website. People could place order from here to buy books. People will be able to read the blurb of the books before buying the books. By searching with name or author of the book user can find the books availability and read about its blurb. People those who register for the site will be able to read story and write story from their own creativity. On the other side of the website admin has a huge role of maintaining the website.

The Developed computing artefact is online book store website. Firstly, books store is developed. Books by different authors are shown in the BOOK STORE page. People are able to read the books blurb. And if anyone wants to buy books user can add the books to the cart by clicking ADD TO CART. One can choose the quantity of the books while adding books to the cart. After adding desired books to the cart user can show the summary of the cart by clicking CART in the navbar. Books Cart will show the chosen book of the customer with book price and total price. In Books Cart summary user can remove the unwanted books what the user doesn't want to take. After confirmation of chosen book user can place order by filling the Billing details properly. In any of the field of billing details is empty order will be not placed. There will be a warning if any field is empty. After completing all fields order will be placed and a confirmation page will appear showing greetings and the list of ordered summary of the books.

In developed computing artefact people can search by book or author name. User can find the books if book is available in the Book Store. From the searched result people will be able to read the blurb of the book. If the searched book or author name do not match then it will show a message that Book Not found.

People is able to register to the website and they can login with their email and password. For the registered member they can post story in STORY section. Those who wants to read can read the story. And if anyone wants to comments on the story can comment on particular story. Lastly. The admin features. Admin can manage the whole system. Admin can entry books, update books, post upcoming books for the customer, see the orders of the customer.

For developing a project, a methodology is used to get the most outcome of the project. In this project Agile Methodology have used. Agile work for getting the best outcome of whole project. It increases the work speed because of developing the major features of the project by prioritizing. For that reason, it divides the whole project in some small feature to develop and this is called iteration. (Rouse, 2018)And this iteration is done within two or four weeks. Continues feedback are taken from the user for each iteration and handed the feature when user is satisfied with the feature outcome. (tutorialspoint, 2019)For this project, there are some features for the online books store. These features are prioritized using MoSCoW prioritization method. For each iteration planning, requirements analysis, design, coding, unit testing and acceptance testing have done. While testing if any kind of error or feature needed to modify than again it put on the iteration to develop. By this process all features of books store project quality ensured within a limited amount of time given for project development.

The developed computing artefact will be online bookstore Named Bookish Dine. The developed computing artefact where people will place order for books beside this user will be able to read blurb of the particular book. User will search for any books with the name of the book or the name of the book author. User will able to add book to the cart and after giving their

information to the billing details they will be able to confirm their order of the chosen books they wanted to buy. Beside this there will be an authentication feature for reading and writing story on the website. And only authenticated people will be able to comment on the story which is written by the authenticated user. Lastly admin role. Admin will be able to entry books for selling. Post upcoming books list for the customer, will update the books quantity for the book store. And will see all the orders submitted by the customer.

For developing a project to get the best outcome it needs to set some aim. When all the aims are fulfilled then project can reach to its goal. Alike this project has some aims. For fulfilling the aims it need to follow some objectives. The main aims of the project are Authentication, Place Order, Search, Story, Comment and Admin. For fulfilling the aims there need to follow some objectives. Where user need to choose books for placing order with required information. User can search with particular book name or book author name. User need to register for authentication and with email and password they can authenticate themselves for the site. Only authenticated user can write story, read story and comment on particular story. And only Admin can manage the admin panel like update books, entry books, post new books and see orders.

For developing the whole project, the initial stage is analyzing the requirements of the project. And the requirements will be divided into two part- one is functional requirements and another is non-functional requirements. And functional requirements are prioritized with the help of MoSCoW Prioritization. And non-functional requirements are how functional requirements will perform in the system. Use case will model the whole project using use case diagram. Architecture of the project will be done drawing initial class diagram modeling the UML notation. And the project will be described with the help of diagram drawn for it. Design is another stage

of developing the whole project. It will model the whole project using structural model and behavior model. And in structural model whole project will be visualize using class diagram. By explaining the class, attributes and method which will describe the project. And behavioral model will be sequential diagram or other diagram which will tells about the behavior of class diagram and its activity. Implementation will be done by choosing the Language for the development. And after coding testing will be done for acceptance. And lastly evolution of the developed project will be discussed on the conclusion. And further development will be mention later on.

Analysis

Requirements

For developing project good analysis can make a good vision of the project. Analysis makes everyone clear what is going to be developed. Requirements and needs are cleared while analyzing. (Solutions, 2016). The requirements are divided into two sections i) Functional Requirements and ii) Non-Functional Requirements. Functional requirements are the functions that needed to the system and the non-functional requirements is the behavior of the functions that implemented on the system. (Eriksson, 2012) For this project functional requirements are prioritized using MoSCoW Prioritization. The Functional and non-functional requirements are discussed as follows: -

Functional Requirements

Functional Requirements is function that must have on the system. Without these functions, system will not made. Project functions also have functional dependencies to the other functions. Like user added book to the cart if user information is not put on the database than

how books can be delivered for this user details need to take. Like this functional dependency are seen in functional requirements. Functional Requirements are discussed as follow:

Add to cart: If people want to buy book, they need to add the book to the cart. And after adding books they have to proceed for placing the order. And after giving the personal information one can place order of their own.

Registration: User need to register with their information to the system. It will help user to login to the system and gives access to some specific feature that other user doesn't get.

Login: User can login to the using email and password which has given to the system while registering for the system.

Search: User can search for any books with book name or books author name. And system will show if it is available in the system database.

Authentication Level: In the system there will be two roles while authentication. One is for admin role and another one is user role. Which will give different access to the different role user.

Story: Authenticated registered user can post story to the system. And this authenticated user can able to read another authenticated people's story.

Comment: In the story authenticated people will be able to comment on the story that are published on the system.

Entry Book: Only authenticated admin can add books to the system. Upcoming books can also add by the admin.

Update Book: Admin will be able to update books in the system.

Delete Books: Admin Can delete books from the system if he needs to do that.

Order: Admin can see all the orders that submitted from the system.

MoSCoW Prioritization: MoSCoW Prioritization is a technique to prioritize the functions of the system going to developed. In this development of system MoSCoW Prioritization is used to priorities the feature that are going to develop. MoSCoW Prioritization are given below:

| Functional Requirements | Prioritization |
|--------------------------------|-----------------------|
| Add to Cart | Must Have |
| Registration | Must Have |
| Log In | Must Have |
| Story | Must Have |
| Post Story | Must Have |
| Comment on Story | Should Have |
| Search | Should Have |
| Upcoming Books Post | Should Have |
| Entry Book | Must Have |
| Update Book | Should Have |
| See Orders | Must Have |

Non-Functional Requirements

Non-functional requirements are the behavior of the functions that are developed for the system.

The non-functional requirements develop the quality, efficiency, reliability of the system. In this system people need to place order to buy books if they found it hard to place then project will not be user friendly. Or the data given by user cannot be protected the project will loss reliability.

Non-functional requirements should be in the development phase of project to ensure more efficiency, quality, reliability of the functions. More non-functional requirements are discussed below:

Availability: System should be available on the internet to use.

Reliability: People find failure free performance of the system.

Security: System should be secure. Only authenticated people can get access to the system. And system database is protected from all kinds of threat.

Data Integrity: In the system database data are maintained properly. Data values are stored according to the data types.

Usability: User can easily use the website. The system will be user friendly that user do not find any difficulty in using the system. (Eriksson, 2012)

Use Case

Use case diagram which shows a graphical portrayal interaction of the elements of a system.

Use case is like a methodology which will identify, clarify and organize the functional and non-functional requirements of the developing project. This use case has four components first one is boundary which only defines the system interest area. Secondly actors, according to their role they are involved with it which is defined by the system individually. Thirdly, use case which is specific role that are played by the actors within the system. Lastly the relation between actors

and the use case. The use case of the diagram is given below which will clarify, identify and organize the functions of the system.

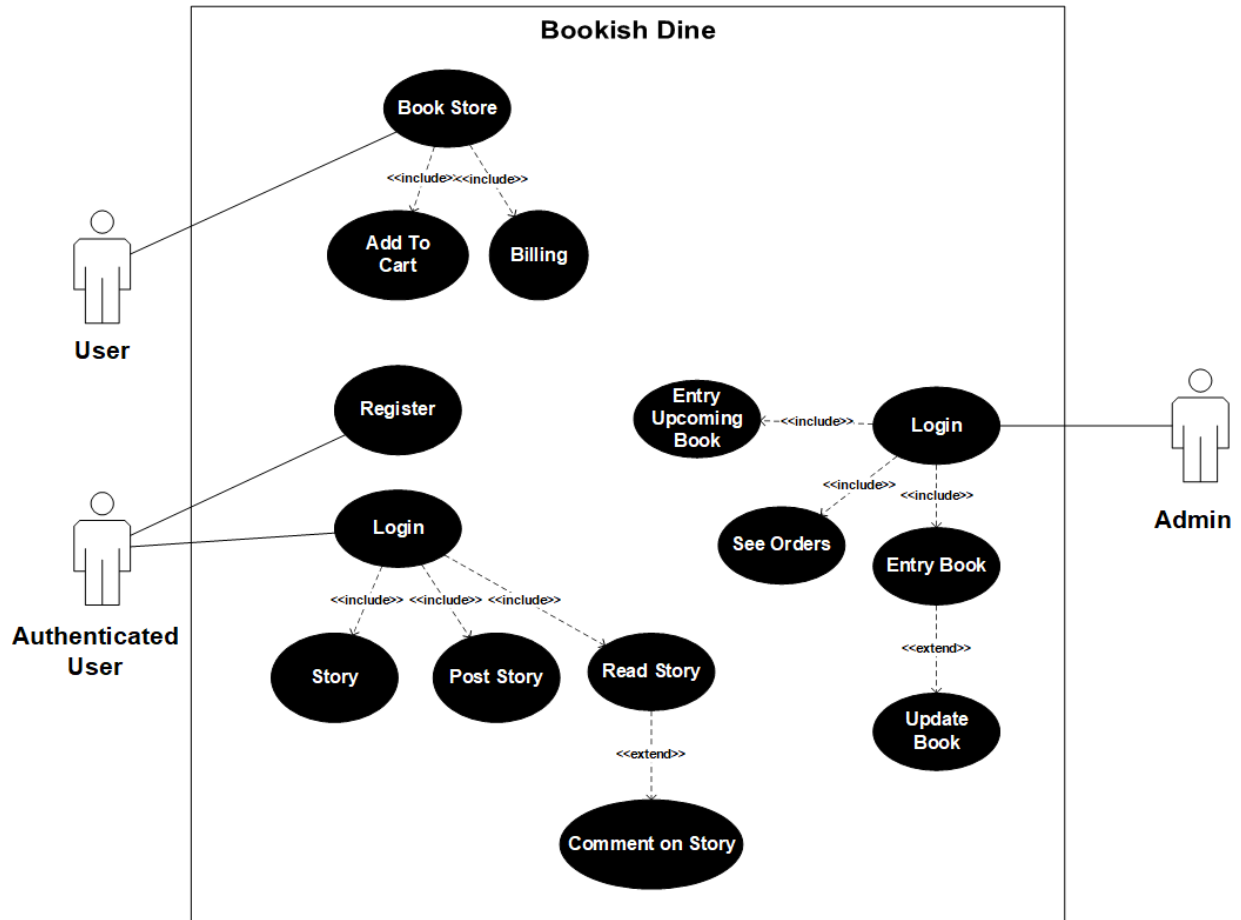


Figure 1: Use Case diagram of the system

System Architecture

System architecture will show the whole system, how it works. The system architecture of the system is given below:

Figure 2: Initial ERD

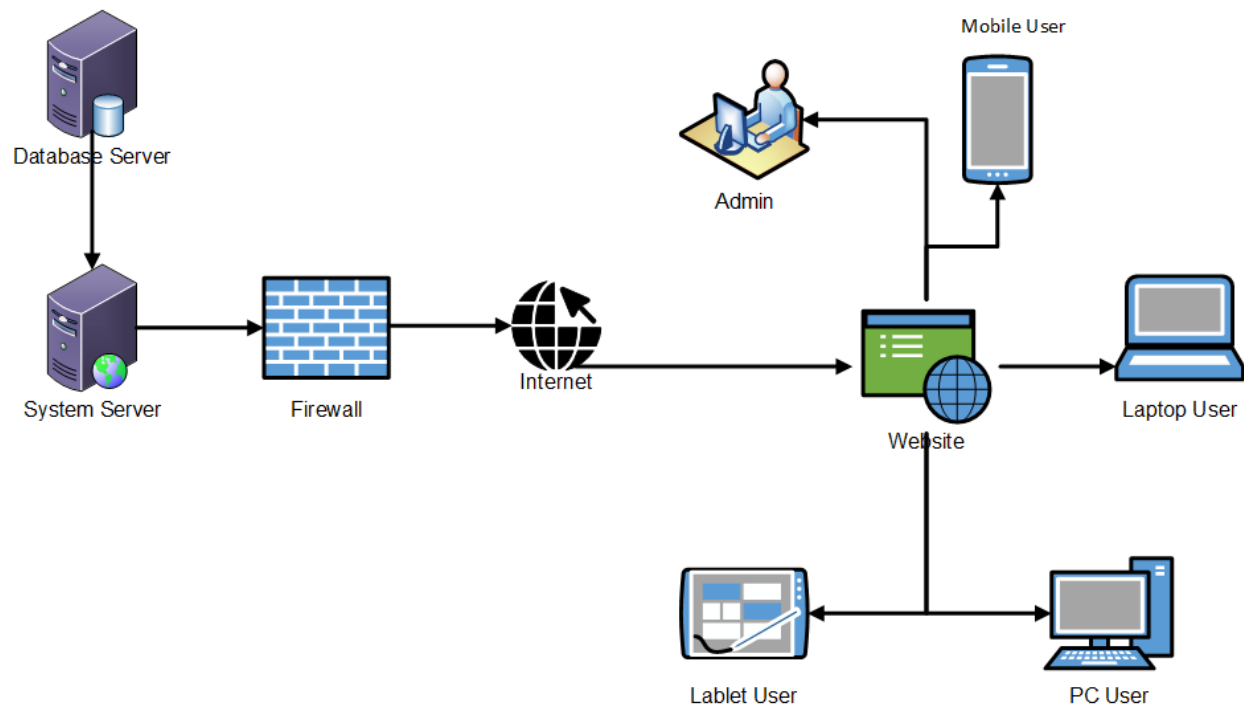


Figure 3: System Architecture of the System

Design

After completing the analyzing phase here, it comes the design phase. Designing phase will have to section one is structural model which will show the diagram of the structure of the system. How the system functions structure will look alike. And another one is behavioral model which shows the action of the function in a diagram. These diagrams need to draw at the beginning of the development of the system. The Entity Relation Diagram will be drawn to show the relation between entities and attributes which will show the database structure of the system. And a behavior model diagram will be drawn like sequential diagram to show the action of the functions of developing system.

Structural Model

Structural model a system is displaying the system components in an organized way and showing their relationship. Structural model will show the static model of the system. Initial ERD are given below:

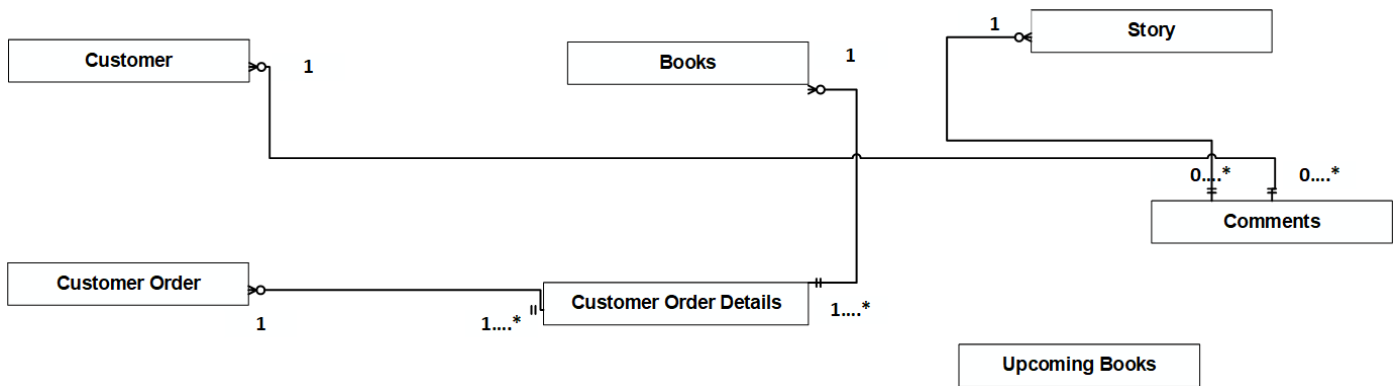


Figure 4: Initial Entity Relation Diagram

Behavioral Model

Behavioral Model is the dynamic model of the system. For the system a sequential diagram are given below to show the activity of user and admin.

Figure 5: Class Diagram of the system with ERD

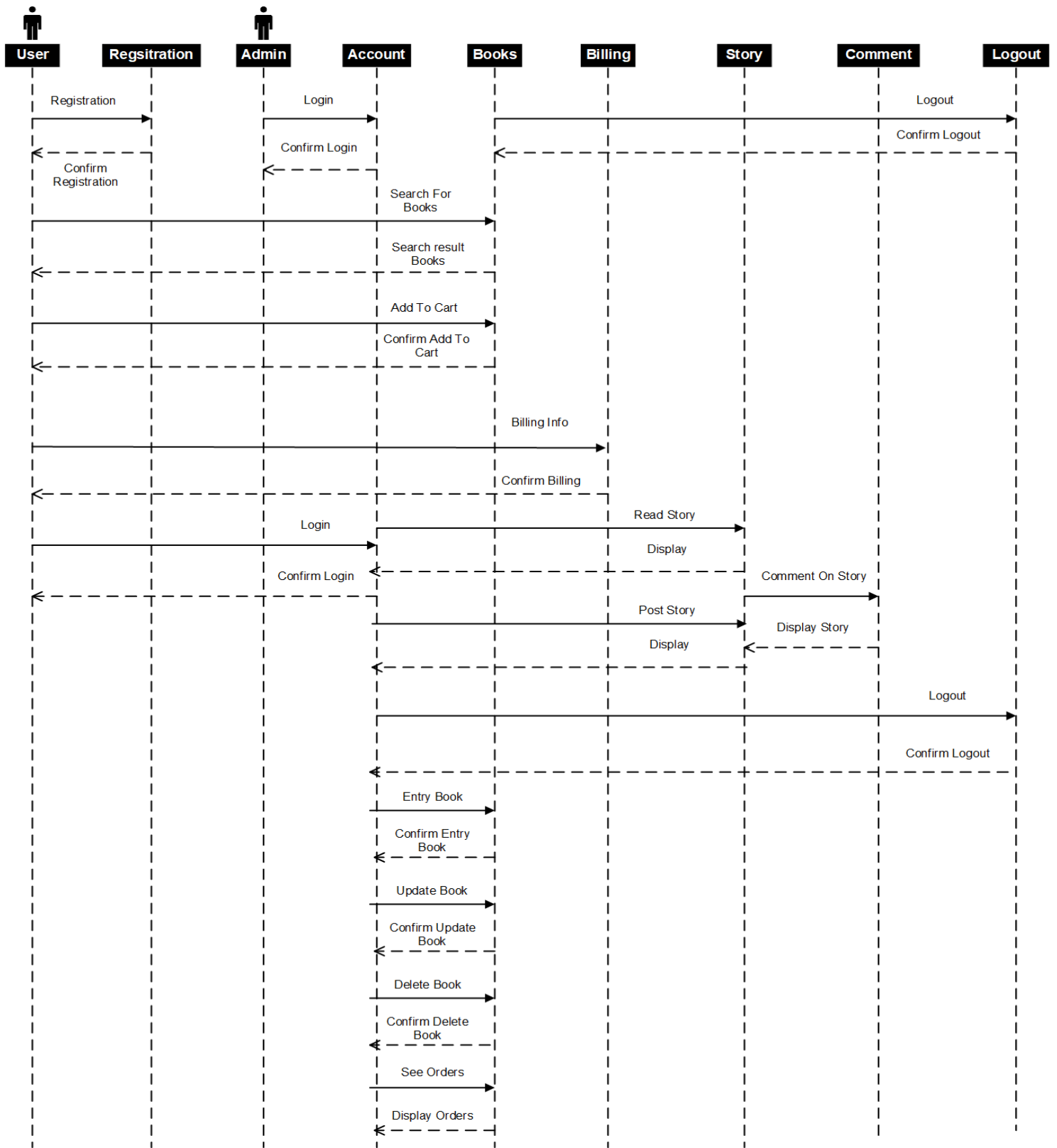


Figure 6: Sequence Diagram of the system

Implementation

Implementation phase, here the whole system will be implemented using the selected programming language. All the analysis, design for the system have done before, all will get visualization by implementation. In this project Agile methodology has used so the main functions of the project are prioritized using MoSCoW prioritization. Using MoSCoW core functions will be implemented first. In the project the main functions that come out with MoSCoW prioritization is Add to Cart. User can add book to the cart. So, this function will be implemented first. Like this admin panel is also an important function of the project so this will come with prioritization. And step by step all functions will be implemented for the project with in short period of time. Because the responsibility of agile is to implement the function in a short time.

Choice of Programming Language

For developing online book store PHP language has used. PHP is a server-side scripting language which is the most used language to develop website in the world. Around 79.0% of current websites are built with this language. (W3Techs, 2019) PHP is an open source language. It is easy to use with HTML. PHP code can easily write inside and outside of the HTML code. PHP will make interactive features to the website like messaging. PHP has huge guideline documentation for its functions. Lots of writing will be found about PHP on internet.

Any kind of query answer will be available by the PHP expert on internet. PHP works with the database also. PHP is so much user-friendly and easy to learn.(Bradley, 2019)

For this project MySQL used with PHP for the database. PHP and MySQL have a great combination of developing a website. MySQL is also open source database server. It can build a powerful database for the website. As it is open source people are adding new feature to it is becoming gaining power, functionality, scalability and speed. (Kohan, 2019)

Besides, in this project Bootstrap 4 Framework has used which helps to make the website responsive

And it helps in to decorate the website. A small portion of JavaScript have used in this project. JavaScript is client-side scripting language. It is used to make website more dynamic and interactive.

System Cutover

System cutover is migration plan of the system. It like a process to migrate one system into another system. In this project, system cutover is about development architecture to implementation architecture. In this phase, how much proposed system developed with the help of implementation will be specified.

With loading of the page there will be shown an interface which will contain a navbar which will have some nav item. And some nav item will be hidden which will be available for the registered or logged in user. When user click on the Book Store system will take user to a new page.

Where all the book will be shown for selling. In the bottom of the books there will be name of the book, author name, short blurb of the book. And lastly there will be two button one for adding books to the cart and another one is blurb. After clicking on blurb user will able to read books blurb which will increase the interest to buy that book. User will able to add multiple books to the

cart by clicking Add to Cart Button. When there adding books to the cart is finished then by clicking cart in the navbar they will see what books they added with the price and quantity and the total bill. If any user wanted to remove any book from the cart by clicking Remove book can be removed. Then for placing the order user have to fill a form with their information. Then a confirmation message will appear and the order summery will be shown. User will be able to register for the system to get some extra feature. For registration user need to click on SignUp from navbar dropdown it will direct user to the registration page contains a form with input field. After filling the input field will details of the user will redirect to the SignIn page automatically. After filling the login filed with email and password given in the registration field. If email and password matched in the back-hand database then user will be logged in if email password doesn't match then a message will be shown that Email or Password are wrong. After logging user will find a new nav item it he navbar that is story. Clicking on story it will redirect user to the story page where writing story of different user will be shown. By clicking any of the story user can read the whole story. After reading the story if user want to comment in the post then bottom of the post there will be a comment field by filling with the comment and clicking comment user can comment on story. If this user has any story to share that in the bottom of story page there will be a section where there will be a button of Post Story. By clicking that user will be taken to another page where a form of different fields will be shown. Filling the field and clicking on post user can post the story in the story page. Now user can logout by clicking dropdown named with user's user name and click log out. User will be able to search book with book name or book author name in search bar which will be in the home page showing a simple search icon. If book name or book author name matched with the database then that book will show on another page with its blurb. If doesn't matched than it will show a message Book Not Found.

The whole system will be managed by admin. Admin will have a great role in managing the website. Admin will be able to enter books to the website for selling. There will be an option of entering upcoming books to help people know which new books are coming. Update feature to update book price, book quantity. And lastly admin will see all the orders in the order option. Admin will manage the whole system.

Data Migration

A process of transferring data between data storage or format on a computer system. (Rouse, 2017) Data Migration can be performed by using software, or any data migration tools. In this project data migration can be done by setting up a MySQL database. At the beginning of the project for data structure we produce an ERD. From that ERD data will be migrated to the MySQL database using attributes. In the initial ERD we will get the Entity name that will be the table name in MySQL database. And for each table there will be some attributes. Like in the user table their attributes will be First Name, Last Name, Username, Email, Password and User Role. And this will be created in the MySQL database. Like this entity rest of the entities like Story, Customer Order, Customer Order Details, Comments, Books, Upcoming Books. By this data migration will take place from development architecture to implementation architecture in this project.

Training

Training is an important task which is needed for any project for the betterment of its maintenance. For this online bookstore a training session should be arranged for the admin to teach about the system developed so that admin can know how to manage the whole system. Admin should have minimum knowledge about the system. Training can make admin more efficient and active

in work. Admin will not turnover from his work. By training admin will have the knowledge about the system its project management, risk management, configuration management. Admin will be able to know about the programming language. If admin have minimum knowledge about all this thing for the system training admin will be able to maintain the system without any difficulty and admin's work flow will be efficient.

Other Project Issues

This phase is all about the issues that arise while implementing the project system. (Sinnaps, 2019) There are many phases like project management, risk management configuration management. All these phases come out at the time of implementation of the proposed system. With the help of iterative development, the raised issues are solved. And system gets its core functionality and become efficient in working.

Project Management

The approach by which whole management of a project is carried out that is called project management. In this project Agile approach have used for managing the project. For the consistency of work while developing the system are maintained with the help of iteration of agile methodology. After completing every iteration function was check if it giving the right function that wanted. Until the core functionality of the feature has done this iteration continues.

Risk Management

For developing a project, identification of possible threats to the project and managing that threat is risk management. Before implementation phase risk of the project need to identify. Then how to overcome the threats and how to reduce the threats should be discussed. And a

solution should make for the particular risk of the project. (Rouse, 2016) In this project for the form inputs user may left empty the fields that need to be filled like registration form or billing form. If they make the field empty than there will be problem in database in storing data and ensure authentication of the user. If the field empty this will notify the user for filling up the field. And by this risk had overcome. Beside this as the system has database system and it will be on online so any kind of hacking may occur to this site. For that reason, admin should have a good knowledge in security sector.

Configuration Management

Configuration Management is a process which works for the consistency of the performance of a system function. Configuration management main work is to keep track of the performance of the system functions. A system is performing the same way at every time when it is working. After a period of time the system need to develop. The system may update its interface, security, database. And configuration management compare the new updated functions are giving the same performance as it was giving early. In this online bookstore system after a period of time this system need to be updated. Further development is needed for needs and for the current trends. So that time configuration management will work to see that systems functions working as before it does. It will assure the new features have the consistency of the performance. It will compare the old version performance with the new version.

Testing

Testing is most important phase in development of system. Check the function of the system are working properly as they need to do or not is defined as testing. This testing done in two section one in unit testing which is testing the core unit of the function and another one is integration testing which is combination of multiple unit testing. Beside this there are other two

testing one is white box testing another one is black box testing. Black box testing the outer system working test. It will have no work with the internal structure or code. And white box testing is internal code testing of the functions. White Box testing and Black Box testing will be given at the appendices section. Now unit testing and the integration testing of some functions will be given below:

| | | | | |
|--------------------------------|----------------------|--|--|--|
| Unit Test No: 01 | | Test Class: Registration | Designed by Mahmudur Rahman | |
| Data Source: User Entry | | Objective: User Registration Process | Tester: Mahmudur Rahman | |
| Test Case No | Description | Task | Expected Result | Actual Result |
| 1 | Registration Process | A user inputs data for registration FirstName: John Last Name: Snow User Name: John_Snow Email: snow_john@gmail.com Password: 123 | When the user clicks on SIGN IN button then these data will be stored in the database. | The given information of user is stored in the database. |



CREATE YOUR ACCOUNT


SIGN IN

Already have an account ? [Login](#)

□ Edit Copy Delete 6 John Snow John_Snow snow_john@gmail.com 123 0

Figure 7: Registration form of the system & data saved in the database

| Unit Test No: 02 | | Test Class: Login | Designed by Mahmudur Rahman | |
|--------------------------------|--------------------|---|--|-------------------------------|
| Data Source: User Entry | | Objective: User Login Process | Tester: Mahmudur Rahman | |
| Test Case No | Description | Task | Expected Result | Actual Result |
| 2 | Login Process | A user inputs data for login Email: snow_john@gmail.com Password: 123 | When the user clicks on SIGN UP button then user will be entering to the system. | User can enter to the system. |



WELCOME BACK

LOGIN

New in this site ? Create New Account

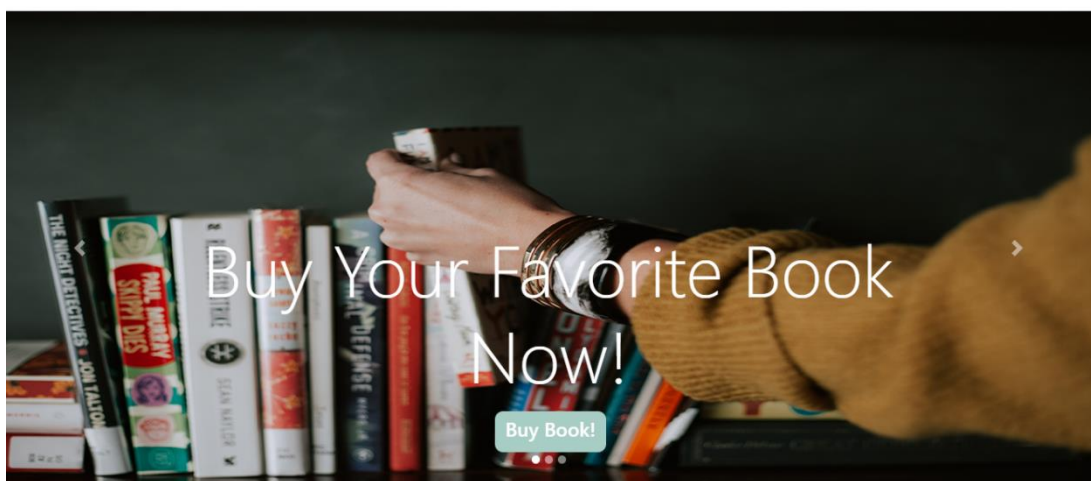


Figure 8: Log In with information and enter the system

| | | | | |
|--------------------------------|---------------------------|--|--|--|
| Integration Test No: 01 | | Tester: Mahmudur Rahman | Designed by Mahmudur Rahman | |
| | | Objective: Check user can see Story at nav bar after Logged in. | | |
| Test Case No | Description | Task | Expected Result | Actual Result |
| 2 | Check user functionality. | When a user will input information to logging in to the system. After logged in to the system then user will able to see story in nav bar. | User will be able to see story in nav bar after logged in. | User can see story in nav bar after logged in. |



Figure 9: Shows Story after logging to the system

Conclusion

In this phase, how much of the project proposal requirements are fulfilled will be discussed here. After implementation of the project requirements it has seen that almost every requirement is fulfilled. For fulfilling all the requirements, I faced so many problems in implementation phase. But I overcome that part doing research, finding solution for the error, surfing on the internet I manage and overcome the difficulties. The core functionalities that are wanted to develop that give shape to reality. And that functions are, user can add books to the cart and the chosen books are given for the placing the order. This was the most core functions of the project system. The registration and login. Users are able to register themselves and login to their account. The logged in user can read story, post story and comment on story. User can search book to see the availability of the book. All requirements are fulfilled. Beside this admin can manage the store by giving book entry, delete book and updating book. So, from this it assures that maximum number of proposed requirements are fulfilled in this project development. Now for adding more requirements or functions further development are needed.

Evolution of Work

After completing the whole project by doing development and implementation a system developed. In the system all core functions are developed. As all core functions are developed. Some lacking can be seen in the functions. Like:

Add to Cart: People will see the real time add to cart item number in the cart.

Registration: Registration email can be verified by sending mail to the email address this validation can perform for the registration.

Payment: An online payment system should be implemented for the system. Where people will pay for the book using card or other online payment system.

Story: Peoples comment should be counted. People would like to see the comments number.

Admin Panel: Admin can delete story from the system. Admin will able to make a chart of the sell. Can calculate the profit and loss of the selling.

All this thing needs to develop for increasing the feasibility, reliability and efficiency of the system. This will be developed in the further development.

Further Development of System

For adding more functions to the system there need further development in the developed system. Many functions are developed but further addition of function to this can make the functions more efficient. As for example after placing the order for books a secure payment method can be developed so that user can easily and safely pay the bill for the ordered books. Search can be improved using ajax which give live search results functionality. In add to cart if any one clicks on add to cart in navbar Cart count of item can be shown for tracking the books user added to the cart. Category of the books can be done so user find it easy in choosing and getting books quickly. Order Delivery Tracking can be implemented so that user can see the position of the order. Coupon or discount functions can be implemented. Discount will be gives in particular date. Admin can delete story which are not appropriate for the people. Admin can track daily selling books quantity. Calculate daily sells amount.

So, for making the system more efficient, reliable and user-friendly further development is Important.

References

Bradley, A., 2019. *Why Use PHP?*. [Online]

Available at: <https://www.thoughtco.com/why-use-php-2694006>

[Accessed 19 April 2019].

Eriksson, U., 2012. *Functional vs Non Functional Requirements*. [Online]

Available at: <https://reqtest.com/requirements-blog/functional-vs-non-functional-requirements/>

[Accessed 2019 April 2019].

Eriksson, U., 2012. *Functional vs Non Functional Requirements*. [Online]

Available at: <https://reqtest.com/requirements-blog/functional-vs-non-functional-requirements/>

[Accessed 17 April 2019].

Kohan, B., 2019. *PHP and MySQL Web Development*. [Online]

Available at: <https://www.comentum.com/php-and-mysql-web-development-definitions-history.html>

[Accessed 19 April 2019].

Rouse, M., 2016. *risk management*. [Online]

Available at: <https://searchcompliance.techtarget.com/definition/risk-management>

[Accessed 19 April 2019].

Rouse, M., 2017. *data migration*. [Online]

Available at: <https://searchstorage.techtarget.com/definition/data-migration>

[Accessed 19 April 2019].

Rouse, M., 2018. *Agile Project Management*. [Online]

Available at: <https://searchcio.techtarget.com/definition/Agile-project-management>

[Accessed 16 April 2019].

Sinnaps, 2019. *PROJECT MANAGEMENT APPROACH*. [Online]

Available at: <https://www.sinnaps.com/en/project-management-en/project-management-approach>

[Accessed 19 April 2019].

Solutions, J., 2016. *The Analysis Phase of a Project: An Important Piece of Any Microsoft Dynamics GP Implementation*. [Online]

Available at: <https://www.erpsoftwareblog.com/2016/02/analysis-phase-project-important-piece-microsoft-dynamics-gp-implementation/>

[Accessed 17 April 2019].

tutorialspoint, 2019. *SDLC - Agile Model*. [Online]

Available at: https://www.tutorialspoint.com/sdlc/sdlc_agile_model.htm

[Accessed 16 April 2019].

W3Techs, 2019. *PHP Market Report*. [Online]

Available at: <https://w3techs.com/technologies/report/pl-php>

[Accessed 19 April 2019].

Appendices

Requirements Catalogue

| Requirements No | Requirements | Priority of Requirements |
|-----------------|--|--------------------------|
| 1 | Create Add to Cart function | High |
| 2 | Create Registration feature for user to register with their info. | High |
| 3 | Create Login feature for the system | High |
| 4 | Show story for the authenticated user. | High |
| 5 | Post story interface for authenticated user to post story to the system. | High |
| 6 | Comment feature for the authenticated user for particular story | Medium |
| 7 | Search interface to search books from the system | Medium |
| 8 | Entry book feature to add book to the system | High |
| 9 | Update book of the existing book of the system | Medium |
| 10 | Post upcoming books for the user to show them | Medium |
| 11 | Feature of showing all orders of the system | High |

Use Case Description

Registration: User need to register to the system to get into the system.

Login: User need to login to the system to get extra feature using valid email and password.

Bookstore: User can see collection of books in bookstore section.

Add to cart: User can add book to the cart for buying book by clicking Add to Cart.

Billing: User can place order by filling the billing details field and confirm order for themselves.

Story: User can see story and read story after doing authentication using email and password.

Post Story: Authenticated user can post story to the system in story section.

Comment: Authenticated user can comment on a particular post that posted by any authenticated user.

Admin Login: Admin can login to the system with particular email and password to access for certain feature.

Entry Book: Admin can entry book to the system for selling them.

Update Book: Admin can update book information when update needed.

Entry Upcoming Books: Admin can entry upcoming book to the system for knowing user about upcoming books.

Orders: Admin can see all orders from the system that customers ordered.

Test Scripts

| Unit Test No: 02 | | Test Class: Entry Book | Designed by Mahmudur Rahman | |
|---------------------------------|-----------------------|--|---|---|
| Data Source: Admin Entry | | Objective: Admin entry Upcoming Book to the database | Tester: Mahmudur Rahman | |
| Test Case No | Description | Task | Expected Result | Actual Result |
| 2 | A basic functionality | Admin Will able to enter upcoming books by filling up the fields that are required to fill to entry book to the database | Admin will entry upcoming book to the database and it will store on the database. | Admin can entry upcoming book to the database and it store in the database. |

| | | | | |
|--------------------------------|--|---|------------------------------------|--|
| Unit Test No: 03 | | Test Class: Add to Cart | Designed by Mahmudur Rahman | |
| Data Source: User Entry | | Objective: User will add book to the cart | Tester: Mahmudur Rahman | |

| Test Case No | Description | Task | Expected Result | Actual Result |
|---------------------|---------------------|--|--|---|
| 3 | Add to cart process | User will click on Add to Cart for adding a single book to the cart. | User will click on Add to Cart for a particular book it will Add book to the cart. | User click on Add to Cart for a particular book it Adds book to the cart. |

| Unit Test No: 04 | | Test Class: Post Story | Designed by Mahmudur Rahman | |
|--------------------------------|--------------------|--|--------------------------------------|--------------------------------------|
| Data Source: User Entry | | Objective: User will Post story | Tester: Mahmudur Rahman | |
| Test Case No | Description | Task | Expected Result | Actual Result |
| 3 | Post Story process | user Will able to post story by filling up the fields that are required to fill to post story to | User will able to post story it will | User can post story, it saved in the |

| | | | | |
|--|--|--|---|---------------------------------------|
| | | the database. And it will show in the website. | save in the database and shows up in the website. | database and shows up in the website. |
|--|--|--|---|---------------------------------------|

System Code

```

<?php include_once 'template/head.php';?>
<?php include_once 'template/nav.php';?>

<?php

//To retriving cart data .....
$count=0;
if(isset($_POST["addtocart"])){
if(isset($_SESSION["books_cart"]))
{
    $books_arrayid = array_column($_SESSION["books_cart"], "bookID");
    if(!in_array($_GET["id"], $books_arrayid))
    {
        $count = count($_SESSION["books_cart"]);
        //$_SESSION['item'] = $count;
        $book_array = array(
            'bookID'          => $_GET["id"],
            'bookName'       => $_POST["bookName"],
            'authorName'    => $_POST["authorName"],
            'price'         => $_POST["price"],
            'quantity'      => $_POST["quantity"]
        );
        $_SESSION["books_cart"][$count] = $book_array;
    }
}
else
{
    //echo '<script>alert("Item Already Added")</script>';
    //echo '<script>>window.location="books-cart.php"</script>';
    for ($bid = 0; $bid<count($books_arrayid); $bid++){
        if ($books_arrayid[$bid] == $_GET["id"]){
            $_SESSION['books_cart'][$bid]['quantity'] += $_POST["quantity"];
        }
    }
}
}
}

```

Figure 10: Add to cart code of the system

```

    }
}
else
{
    $book_array = array(
        'bookID'           => $_GET["id"],
        'bookName'        => $_POST["bookName"],
        'authorName'      => $_POST["authorName"],
        'price'           => $_POST["price"],
        'quantity'        => $_POST["quantity"]
    );
    $_SESSION["books_cart"][0] = $book_array;
}

if(isset($_GET["action"]))
{
    if($_GET["action"] == "delete")
    {
        foreach($_SESSION["books_cart"] as $keys => $values)
        {
            if($values["bookID"] == $_GET["id"])
            {
                unset($_SESSION["books_cart"][$keys]);
                echo '<script>alert("Item Removed")</script>';
                header('location:bookscart.php');
            }
        }
    }
}
}

```

Figure 11: Add to cart code of the system

```
<?php
session_start();
$fName= $_POST['fName'];
$lName= $_POST['lName'];
$uName= $_POST['uName'];
$email= $_POST['eMail'];
$password= $_POST['password'];
$userRole= $_POST['userRole'];

if($fName == ''){
    header('location:signupForm.php?msg=Please type First Name');exit;
}

if($lName == ''){
    header('location:signupForm.php?msg=Please type Last Name');exit;
}

if($uName == ''){
    header('location:signupForm.php?msg=Please type User Name');exit;
}

if($email == ''){
    $_SESSION['msg'] = 'Please type Email';
    header('location:signupForm.php?msg=Please type Email');exit;
}

if($password == ''){
    header('location:signupForm.php?msg=Please type Password');exit;
}

include_once 'databaseConnect.php';
$conn=connect();
```

Figure 12: Signup code of the system

```
include_once 'databaseConnect.php';
$conn=connect();

$sql = "SELECT * FROM customer WHERE Email = '$eMail' OR UserName = '$uName'";
$output = $conn->query($sql);

if($output->num_rows>0){
    header('location:signupForm.php?msg=User Already Exists!');
    exit;
}
else{

    $sql = "INSERT INTO customer (FirstName, LastName, UserName, Email, Password, userRole)
        VALUES ('$fName', '$lName', '$uName', '$eMail', '$passWord', '$userRole')";

    $conn->query($sql);
    header('location:signInForm.php?msg= Registered Successfully');

}
?>
```

Figure 13: Signup code of the system

```
include_once 'databaseConnect.php';
$conn=connect();

$sql = "SELECT * FROM customer WHERE Email = '$eMail' OR UserName = '$uName'";
$output = $conn->query($sql);

if($output->num_rows>0){
    header('location:signupForm.php?msg=User Already Exists!');
    exit;
}
else{

    $sql = "INSERT INTO customer (FirstName,LastName,UserName,Email>Password,userRole)
    VALUES ('$fName','$lName','$uName','$eMail','$passWord','$userRole)";

    $conn->query($sql);
    header('location:signInForm.php?msg= Registered Successfully');

}
?>
```

Figure 14: Signup code of the system

```

<?php
    session_start();
    $eMail=$_POST['eMail'];
    $passWord=$_POST['passWord'];

    if($eMail == ''){
        header('location:signinForm.php?msg=Please type Valid Email');exit;
    }

    if($passWord == ''){
        header('location:signinForm.php?msg=Please type Password');exit;
    }

    include_once 'databaseConnect.php';
    $conn=connect();

    $sql = "SELECT * FROM customer WHERE Email = '$eMail' AND Password = '$passWord'";
    $output = $conn->query($sql);

    if($output->num_rows>0){
        foreach($output AS $row){
            $_SESSION['Customer_Id']=$row['Customer_Id'];
            $_SESSION['userRole'] = $row['userRole'];
            $_SESSION['userName'] = $row['UserName'];
        }
        $_SESSION['loggedin'] = true;
        header('location:index.php?msg=Loggedin successfully');
    }
    else{
        header('location:signinForm.php?msg=Your Email or Password is Wrong');
    }
?>

```

Figure 15: SignIn code of the system

```
<?php
    session_start();
    $b_name = $_POST['b_name'];
    $a_name = $_POST['a_name'];
    $b_price = $_POST['b_price'];
    $b_quan = $_POST['b_quan'];
    $b_blurb = $_POST['b_blurb'];
    $_SESSION['msg']='';

    if($b_name == ''){
        header('location:book_entryForm.php?msg=Please type Book Name');exit;
    }
    if($a_name == ''){
        header('location:book_entryForm.php?msg=Please type Author Name');exit;
    }
    if($b_price == ''){
        header('location:book_entryForm.php?msg=Please Enter Price');exit;
    }
    if($b_quan == ''){
        $_SESSION['msg'] = 'Please Enter Quantity';
        header('location:book_entryForm.php?msg=Please Enter Quantity');exit;
    }
    if($b_blurb == ''){
        header('location:book_entryForm.php?msg=Please type Blurb');exit;
    }
}
```

Figure 16: Book Entry code of the system


```

//File Upload

$target_dir = "img/";
$fileName= basename($_FILES["imagefileUpload"]["name"]);
$target_file = $target_dir . $fileName;

$uploadOk = 1; //Flag
$imageFileType = strtolower(pathinfo($target_file,PATHINFO_EXTENSION));

// Check if image file is a actual image or fake image
if(isset($_POST["submit"])) {
    $check = getimagesize($_FILES["imagefileUpload"]["tmp_name"]);
    if($check !== false) {
        echo "File is an image - " . $check["mime"] . ".";
        $uploadOk = 1;
    } else {
        header('location:book_entryForm.php?msg=File is not an image');
        $uploadOk = 0;
    }
}
// Check if file already exists
if (file_exists($target_file)) {
    header('location:book_entryForm.php?msg=Sorry, file already exists');
    $uploadOk = 0;
}
// Check file size
if ($_FILES["imagefileUpload"]["size"] > 500000) {
    header('location:book_entryForm.php?msg=Sorry, your file is too large');
    $uploadOk = 0;
}

```

Figure 17: Book Entry code of the system

```

}
// Allow certain file formats
if($imageFileType != "jpg" && $imageFileType != "png" && $imageFileType != "jpeg"
&& $imageFileType != "gif" ) {
    header('location:book_entryForm.php?msg=Sorry, only JPG, JPEG, PNG & GIF files are allowed');
    $uploadOk = 0;
}
// Check if $uploadOk is set to 0 by an error
if ($uploadOk == 0) {
    header('location:book_entryForm.php?msg=Sorry, your file was not uploaded');

// if everything is ok, try to upload file
} else {
    if (move_uploaded_file($_FILES["imagefileUpload"]["tmp_name"], $target_file)) {
        echo "The file ". basename( $_FILES["imagefileUpload"]["name"]). " has been uploaded.";
    } else {
        header('location:book_entryForm.php?msg=Sorry, there was an error uploading your file');
    }
}

//---End of File Upload

include_once 'databaseConnect.php';

$conn=connect();

```

Figure 18: Book Entry code of the system

```

include_once 'databaseConnect.php';

$conn=connect();

$sql = "INSERT INTO books(bookName,authorName,price,quantity,blurb,book_image)
VALUES ('$b_name','$a_name','$b_price','$b_quan','$b_blurb','$fileName')";

if($conn->query($sql)){
    $_SESSION['msg']='Added Successfully';
}
else{
    $_SESSION['msg']='Not Added'. $conn->Error;
}
header('location:book_entryForm.php');

?>

```

Figure 19: Book Entry code of the system

```

1 <?php
2 session_start();
3 include_once 'databaseConnect.php';
4 $conn=connect();
5
6     if(isset($_POST['btn_Comment'])) {
7         $cmnt = $_POST['comment'];
8         $C_userId = $_SESSION['Customer_Id'];
9         $C_username = $_SESSION['userName'];
10        $activeStoryId = $_SESSION['story_id'];
11
12        $sql = "INSERT INTO `comments` ( `Customer_Id`, `post_id`, `comments`, `Name` )
13        VALUES ( '$C_userId', '$activeStoryId', '$cmnt', '$C_username' )";
14        $conn->query($sql);
15        header('location:single_story.php?storyid='.$activeStoryId);
16    }?>

```

Figure 20: Comment code of the system

```

<?php
function connect() {
    $host='localhost';
    $user='root';
    $pass='';
    $database='BookishDine';

    $con = new mysqli($host,$user,$pass,$database);
    return $con;

    if($con->connect_error){
        die("connection failed:" . $con->connect_error);
    }
    else{
        echo "Connected";
    }
}
?>

```

Figure 21: Database Connection code of the system