**Sean Cunningham**

**Isaac Dehora**

LAB WEEK 3 ASSIGNMENT 3 SCOPING AND REQUIREMENTS

Project management

Overview

1. Isaac Dehora – g00372043@gmit.ie / Scrum Master
2. Doha Kadour – g00390296@gmit.ie / Tester
3. Sean Cunningham -- g00381530@gmit.ie / Product Owner

Lecturer: Joseph Corr

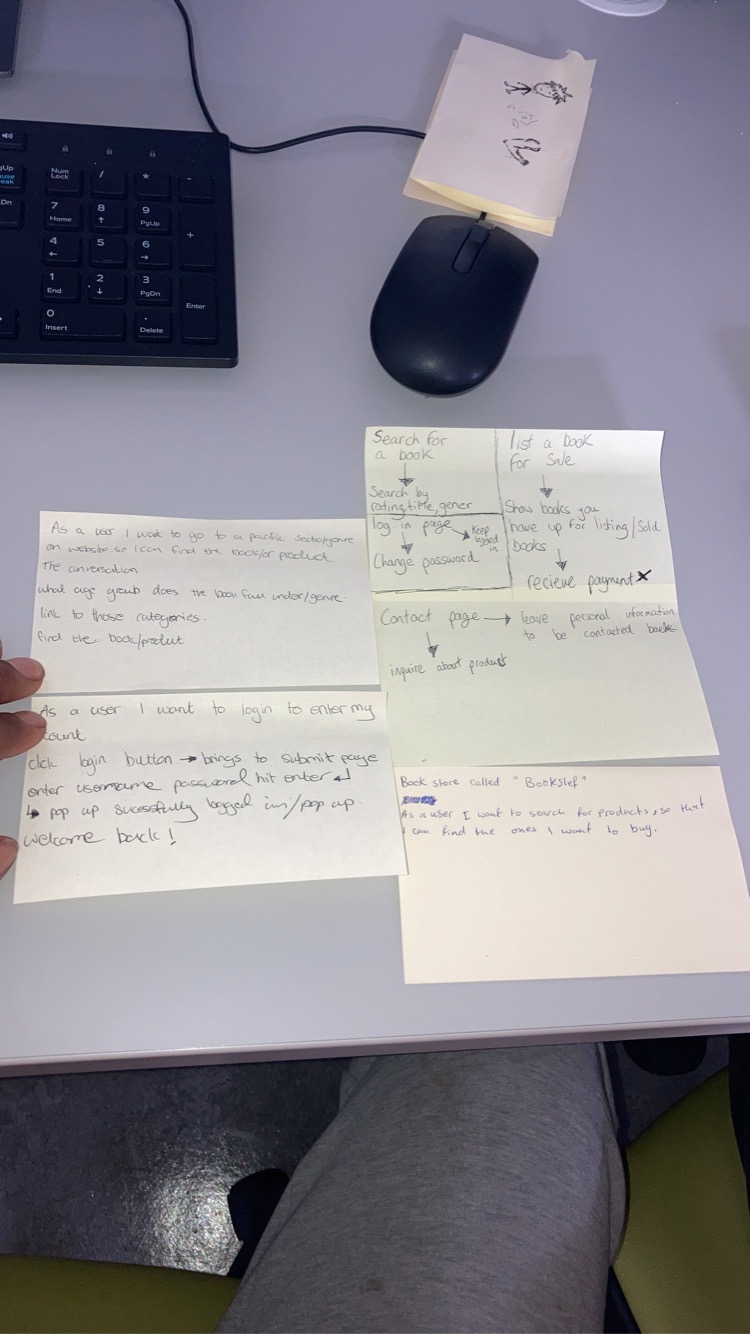
Project Name: The Bookshelf

Our three-person team’s name which was agreed upon is named The Bookshelf,

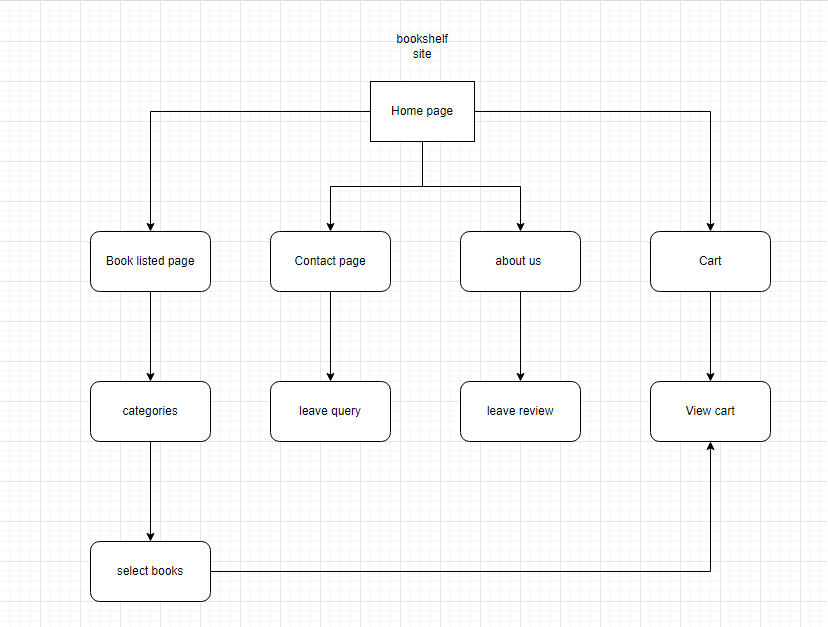
Through brainstorming and discussing ideas about the project we agreed on the project being an online bookstore for customers to visit and buy different books and other materials involving book i.e., book markers and highlighters. We chose this because we felt it was refreshing and I Isaac personally had never tried to do a bookstore before only sports shops, so I was intrigued by Doha’s idea about books. The goal of our project is to build an informative clean and accessible website that provides simple but enticing for our customers.

Brainstorming

We got together and created user stories on paper to see what we would like our website to do, we also went on diagrams.net to create flow charts to help us better understand how we want the project to flow



After this we then began to do an idea of how are website would flow using a flowchart diagram



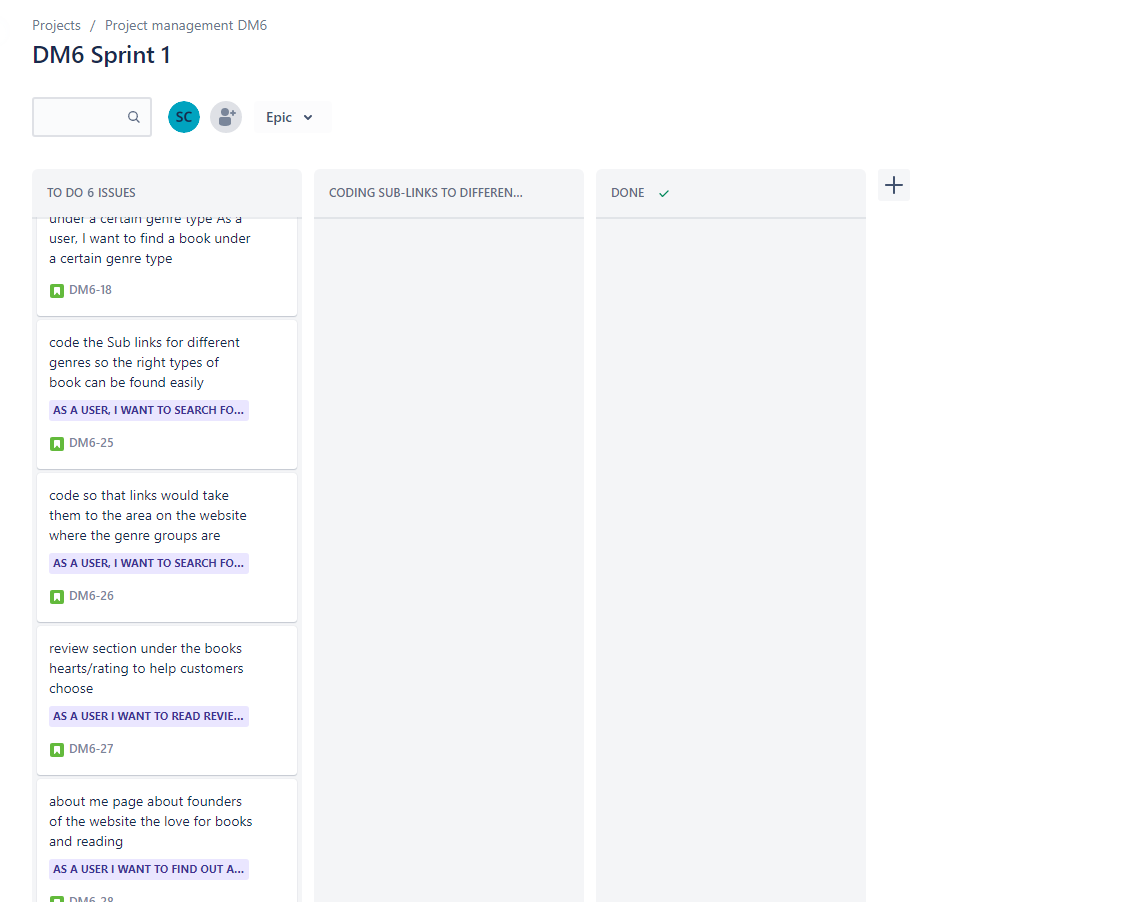
Applications

Jira: I created a Jira project and added all members to start scoping and working out the requirements to the project Story boards, road maps, issues and the backlog was made to give us an understanding od what the project would need to and where to start in the process of scoping and planning. The backlog was good it allowed me to see what we really would need and set a basis to our project to build on a good foundation moving forward.

GitHub: I created a GitHub project to allow the team to work in real time and share code and work together to help each other and have open communication in the team and help each other with any blockers if needed.

Epics

epics refer to a strategic focus for the project, a portfolio plan item, or another set of features that met a market requirement. It is not necessary to complete all user stories or functionalities. Rather, the epic might be enough to fill the void. Once acknowledged, the completed epic will be used in throughput calculations to see if supply and demand are in balance. Here are our epics Isaac created



Functional requirements

* Business rules (we had learnt from database management class)
* Performance
* Scalability
* Hosting

Nonfunctional requirements

* Useability
* Smoothness
* Security
* Eye catching

User stories

1. As a user I want to search for a book under a certain genre type.

Code/Create Sub links for different genres so the right types of books can be found easily.

1. As a user I want to read reviews on book.

review section under the books hearts/rating to help customers choose.

1. As a user I want to find out about the website.

Code/Create an About me page about founders of the website the love for books and reading for users’ assurance and trust.

1. As a user, I want to get in contact/interact with website users to ask questions.

Code a form to fill out with questions and remarks about our product automated so the company email.

The confirmation

1. User picks a list of genres that are links takes them to the genre of their choosing. From there they can view a list of books more to their liking.
2. Users have the option of rating a book out of 5 stars for public review then it’s saved to show. Average calculated together.
3. Through the navbar users can select the about us page to learn more about the3 shareholders of the website for trust and assurance.
4. Users have the option to fill out a form they type their name email and text box to write any question, request, and complaints for free.

**Development environment sketch**Chart, box and whisker chart

Description automatically generated

Graphical user interface

Description automatically generated

**Definition of done:**

Definition of done from what we have learnt in class is most used at the delivery team level. At this level, the user story has been reviewed and accepted by the Product Owner. The "done" user story will contribute to the team velocity after it has been accepted. The user story will not be completed unless all the set conditions are met. Here is a sample of user stories that would be used in definition of done

* The unit tests were successful.
* The code has been examined.
* Items needed for acceptance were met.
* The functional testing was successful.
* Requirements that are not functional are met.
* The User Story is accepted by the Product Owner.

**Definition of ready:**

A definition of ready refers to a user story that is ready to be implemented in a sprint. It does not have to be "completely clear" in terms of all acceptance criteria. It should only be "ready enough" when the team is confident in their ability to deliver the user story successfully. Here is a sample of user stories that would be used in the definition of ready

* Well-defined A user's perspective
* Acceptance criteria for user stories have been established.
* The delivery team sized the user story.
* User experiences are accepted by the Scrum team.
* Performance standards and how you want things to work are established.

**development environment and tool requirements**

Programming languages

* HTML (Hypertext Markup Language)
* CSS (Cascading Style Sheets)
* Java
* Java script

Tools

* Jira
* GitHub
* Microsoft teams
* Notepad++

Database

* Wamp
* SQL

**Testing strategy**

We have come together and agreed on the following steps for our testing strategy

* The first step is to define the scope and give an overview of the project.
* Then we will put it through testing.
* After we will Evaluate the environment.
* Testing Tools is the next step.
* Perform a risk assessment of any errors that have/could occur.
* Approval And Review.

**Deployment**

For our deployment we will be following a release plan burndown in which we will create a sprint plan on a scrum board and prioritize what we need to do and see if what we are able to put into one sprint is accomplishable

**Deployment tools**

* Jira
* In person stand-ups
* iteration plans
* In-person meetings